WITIO-PCIe192
ULTRA

EDP-No.: A-864810

192 Inputs /outputs, interrupt-capable
digital input filters
32 counter 32Bit
2 timers 32Bit
8 OC units
2 IC units
interrupt capable
Board Identification

wasco®

user‘s guide

wasco®

Registered Trademarks

BASIC is a registered trademark of Dartmouth College.
Windows is a registered trademark of Microsoft.
wasco® is registered trademark.

Disclaimer

The company Messcomp Datentechnik GmbH assumes no liability for the use of the interface card WITIO-PCIe192ULTRA and this documentation, neither for direct nor indirect damages.

wasco®

Table of Contents

1. Description

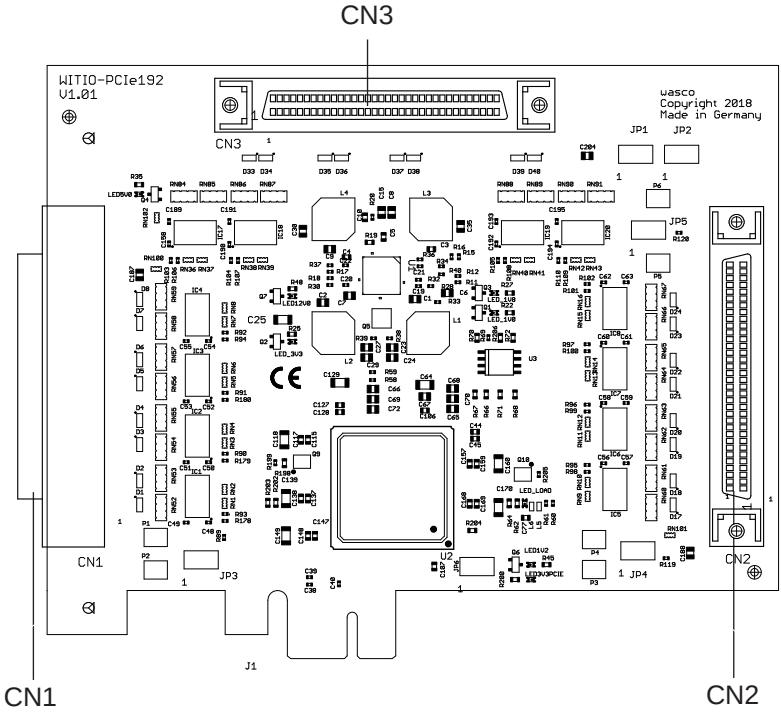
The WITIO-PCIe192ULTRA (Boardname: wasco-PCIe8296) features 192 digital inputs/outputs adjustable to input/output levels of 3,3V or 5V by jumpers. The channels can be programmed in groups of eight channels each as input or output. Every input channel is interrupt-capable and all of the 64 inputs of the connector CN1 have a programmable digital filter (0 up to 255µs). Additionally, one of the 32 counter (32Bit) or one of the two IC-Units (e.g. period or pulse duration measurement) can be linked to each input channel via software. For applications requiring a PWM or a simple, high-resolution pulse generation, eight OC-Units are provided. These can be linked to the first eight output channels of each connector. Additionally, the WITIO-PCIe192ULTRA features two programmable and interrupt-capable quartz-time-based timers.
The board is suitable for input and output applications not requiring galvanic isolation. The internal data bus of this board is organized 32 bit, each read or write access to the inputs and outputs is implemented as a 32-bit access. One 68-pin SCSI-II socket at the board‘s slot plate and two 68-pin SCSI-II sockets providing each 64 channels enable connection to the peripherals. The pin assignment of all connectors of the WITIO-PCIe192ULTRA are identical to the assignments of PCI-bus cards WITIO-PCI32STANDARD and WITIO-PCI64EXTENDED, a switch to PCIe is thus easily feasible. Furthermore, the card provides a jumper block for card identification in order to distinguish several identical cards in your system.

2. Installation of the WITIO-PCIe192ULTRA

2.1 Installation of the card into your system

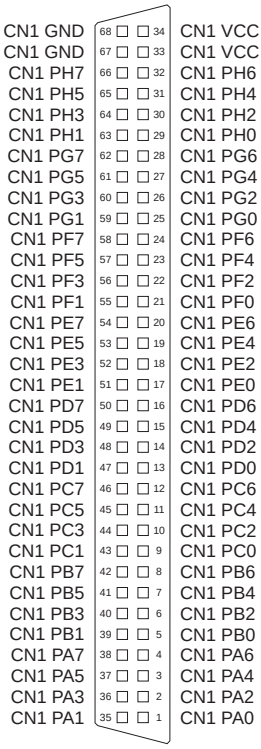
Before you insert the WITIO-PCIe192 unplug the power cord or make sure, there is no current to/in the computer. Inserting into a running system may cause damaging or destroying not only WITIO-PCIe192, but even other already inserted cards of your computer.
Select a free PCIe slot of your computer for then inserting the card. Please refer to the computer‘s manual for support. Screw the slot plate of the board to the computer housing to avoid a card‘s coming loose during operation by effects of the cables.

3. Connectors



CN1: 64 digital I/Os
CN2: 64 digital I/Os
CN3: 64 digital I/Os

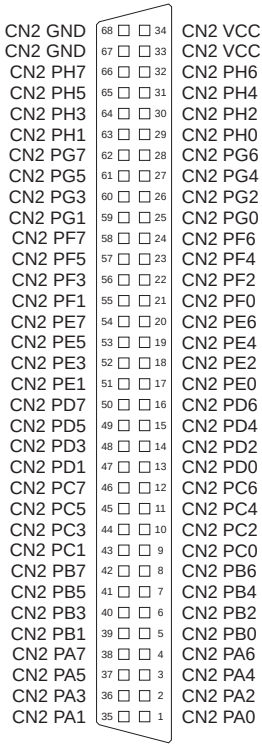
3.2 Pin assignment of CN1



Vcc:
Internal voltage supply (+ 5V / +3,3V) of the PCIe card (configurable by JP3),
Never apply an external voltage across this pin.

GND:
Ground connection

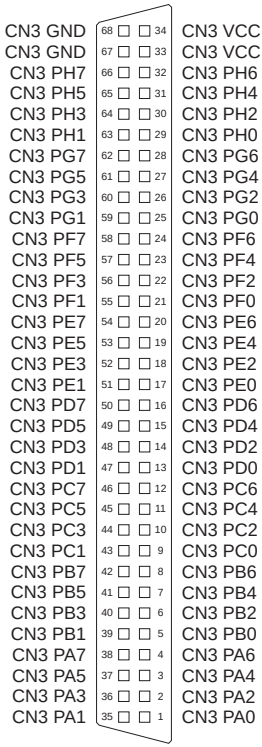
3.3 Pin assignment of CN2



Vcc:
Internal voltage supply (+ 5V / +3,3V) of the PCIe card (configurable by JP4),
Never apply an external voltage across this pin.

GND:
Ground connection

3.4 Pin assignment of CN3



VVcc:
Internal voltage supply (+ 5V / +3,3V) of the PCIe card (configurable by JP5),
Never apply an external voltage across this pin.

GND:
Ground connection

4. System Components

4.1 Block diagram

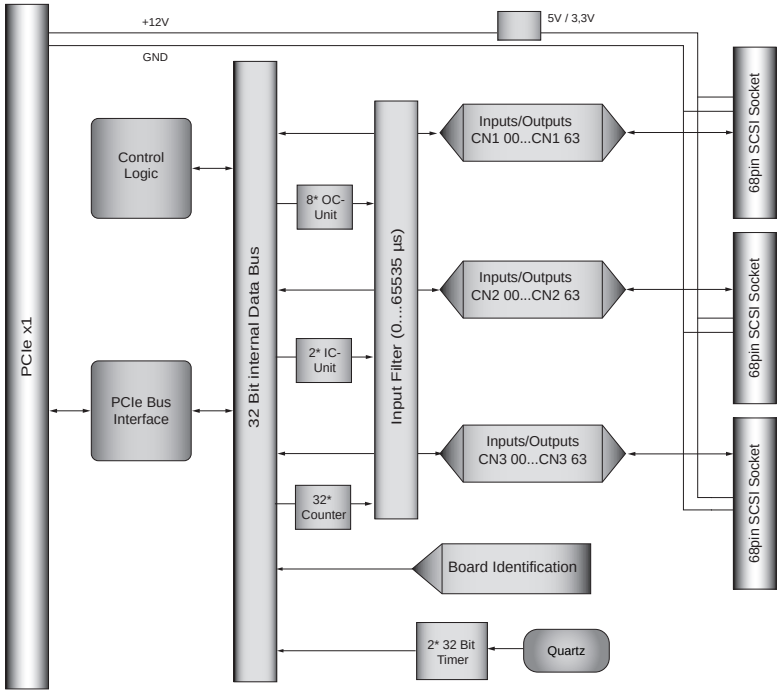


Fig. 4.1

4.2 Access to the system components

You can access to the hardware components of the WITIO-PCIe192 by reading from or writing to Memory Mapped I/O addresses using library functions. The addresses relevant to the WITIO-PCIe192 arise depending on the BIOS assigned base address. Access to the WITIO-PCIe192 is exclusively in double-word access. For reasons of compatibility the wasco driver features only process or allow for the least significant byte.
(You will find more information in the chapter Programming as well as in the sample programs on the supplied CD)

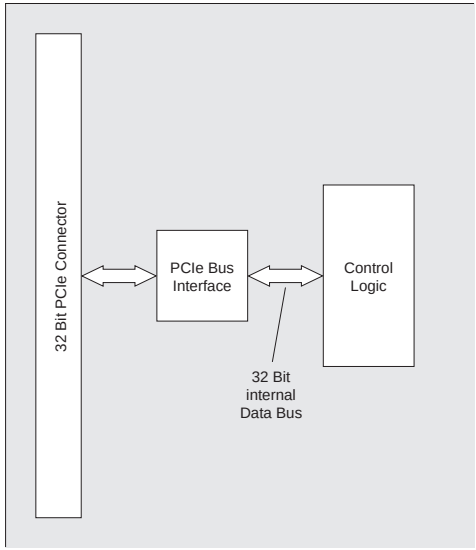


Fig. 4.2

5. 192 Digital Inputs / Outputs

For digital input / output, the WITIO-PCIe192ULTRA provides three identically constructed connectors with 64 digital IOs each.
These 64 digital IOs are arranged in groups of eight channels each (PortA to PortH). You can configure the direction (input or output) of each port individually via a register access. You can set the input / output level (3.3V/5V) of each connector by setting a jumper.

5.1 Selection of the direction

The port direction of a connector can be defined by writing to the corresponding direction register (DDIRCN1, DDIRCN2 or DDIRCN3). Each bit of the register represents a port of the connector. In the default state, all of the ports are defined to be input (except during compatibility mode).
If the corresponding bit is written 1, the corresponding port is configured as input port, with a 0 as output port.

5.2 Read inputs

For reading the inputs, always 32 IOs (4 ports) are aggregated in a 32-Bit register (DIN0CN1, DIN1CN1, DIN0CN2, DIN1CN2, DIN0CN3, DIN1CN3). If not all of the register ports are configured as input, the corresponding areas in the register are undefined and can be hidden in the application program using an AND operation.

0 = LOW at the input pin
1 = HIGH at the input pin

5.3 Read / write outputs

Writing to the register DOUTyCNx [(1)] sets the outputs. Each connector has two of these registers (DOUT0CNx und DOUT1CNx), in each of which 32 IO pins (4 ports) aggregate. If not all of the register ports are configured as output, then only the output ranges in the register are taken into account by the card.

0 = LOW at the output pin (if configured as output)
1 = HIGH at the output pin (if configured as output)

5.4 Deactivation of ports

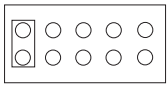
In order to minimize the power consumption of the card, and to prevent possible interferences on the connector, unused ports and their IOs can be deactivated. If a port is deactivated, all of the IOs are high-impedance. Ports always can be deactivated two by two, that is 16-channel-wise. To do this, set the respective bit of the DENCNx register.

0 = port group activated
1 = port group deactivated

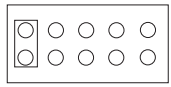
[1] (y = Register number, x = Connector number)

5.5 Level adjustment

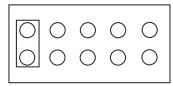
You can configure the voltage level for each of the connectors (IOs and VCC-PINs) with +3.3V and +5V. This is done by setting a jumper over the jumper block of the corresponding connector.



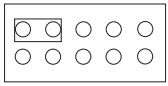
JP3 Voltage Level
CN1 +3.3V



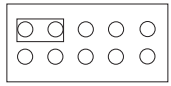
JP4 Voltage Level
CN2 +3.3V



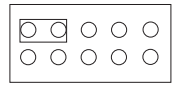
JP5 Voltage Level
CN3 +3.3V



JP3 Voltage Level
CN1 +5V



JP4 Voltage Level
CN2 +5V

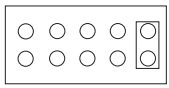


JP5 Voltage Level
CN3 +5V

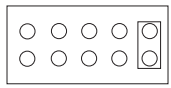
5.6 IOs in compatibility mode

In case of need, each one of the connectors can operate in compatibility mode. In this mode, the pinout corresponds to the pinout of the cards WITIO-PCI32$_{\text{Standard}}$ and WITIO-PCI64$_{\text{Extended.}}$ Then, the first 32 IO pins are configured as inputs and the second 32 IO pins as outputs.

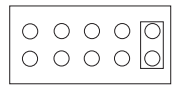
For using the compatibility mode of a connector, you have to set the jumper of the respective jumper block as shown below. The registers DDIRCNx are disabled in the case of the compatibility mode.



JP3 Compatibility Mode
CN1



JP4 Compatibility Mode
CN2



JP5 Compatibility Mode
CN3

5.7 Port addresses

Offset Address	Register Name	Bit Range	Bits																
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x140	DIN0CN1	31:16	DIN0CN1 <31:16>															
		15:0	DIN0CN1 <15:0>															
0x144	DIN1CN1	31:16	DIN1CN1 <31:16>															
		15:0	DIN1CN1 <15:0>															
0x148	DIN0CN2	31:16	DIN0CN2 <31:16>															
		15:0	DIN0CN2 <15:0>															
0x14C	DIN1CN2	31:16	DIN1CN2 <31:16>															
		15:0	DIN1CN2 <15:0>															
0x150	DIN0CN3	31:16	DIN0CN3 <31:16>															
		15:0	DIN0CN3 <15:0>															
0x154	DIN1CN3	31:16	DIN1CN3 <31:16>															
		15:0	DIN1CN3 <15:0>															
0x160	DOUT0CN1	31:16	DOUT0CN1 <31:16>															
		15:0	DOUT0CN1 <15:0>															
0x164	DOUT1CN1	31:16	DOUT1CN1 <31:16>															
		15:0	DOUT1CN1 <15:0>															
0x168	DOUT0CN2	31:16	DOUT0CN2 <31:16>															
		15:0	DOUT0CN2 <15:0>															
0x16C	DOUT1CN2	31:16	DOUT1CN2 <31:16>															
		15:0	DOUT1CN2 <15:0>															
0x170	DOUT0CN3	31:16	DOUT0CN3 <31:16>															
		15:0	DOUT0CN3 <15:0>															
0x174	DOUT1CN3	31:16	DOUT1CN3 <31:16>															
		15:0	DOUT1CN3 <15:0>															

Offset-Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x180	DDIRCN1	31:16	reserved (*)															
		15:0	reserved (*)											DDIRCN1 <7:0>				
0x184	DDIRCN2	31:16	reserved (*)															
		15:0	reserved (*)											DDIRCN2 <7:0>				
0x188	DDIRCN3	31:16	reserved (*)															
		15:0	reserved (*)											DDIRCN3 <7:0>				
0x190	ENCN1	31:16	reserved (*)															
		15:0	reserved (*)											ENCN1 <3:0>				
0x194	ENCN2	31:16	reserved (*)															
		15:0	reserved (*)											ENCN2 <3:0>				
0x198	ENCN3	31:16	reserved (*)															
		15:0	reserved (*)											ENCN3 <3:0>				

(*) reserved area has to be assigned 0

Register DIN0CNx (x = number of the connector):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R							
	DIN0CNx <31:24> (PD7 : PD0)							
23:16	R							
	DIN0CNx <23:16> (PC7 : PC0)							
15:8	R							
	DIN0CNx <15:8> (PB7 : PB0)							
7:0	R							
	DIN0CNx < 7:0> (PA7 : PA0)							

Bit 31 - 24 DIN0CNx <31:24> Port D Inputs PD7 to PD0 of the connector
Bit 23 - 16 DIN0CNx <23:16> Port C Inputs PC7 to PC0 of the connector
Bit 15 - 8 DIN0CNx <15:8> Port B Inputs PB7 to PB0 of the connector
Bit 7 - 0 DIN0CNx <7:0> Port A Inputs PA7 to PA0 of the connector

Register DIN1CNx (x = number of the connector):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R							
	DIN1CNx <31:24> (PH7 : PH0)							
23:16	R							
	DIN1CNx <23:16> (PG7 : PG0)							
15:8	R							
	DIN1CNx <15:8> (PF7 : PF0)							
7:0	R							
	DIN1CNx < 7:0> (PE7 : PE0)							

Bit 31 - 24 DIN1CNx <31:24> Port H Inputs PH7 to PH0 of the connector
Bit 23 - 16 DIN1CNx <23:16> Port G Inputs PG7 to PG0 of the connector
Bit 15 - 8 DIN1CNx <15:8> Port F Inputs PF7 to PF0 of the connector
Bit 7 - 0 DIN1CNx <7:0> Port E Inputs PE7 to PE0 of the connector

Register DOUT0CNx (x = Connector number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	DOUT0CNx <31:24> (PD7 : PD0)							
23:16	R/W							
	DOUT0CNx <23:16> (PC7 : PC0)							
15:8	R/W							
	DOUT0CNx <15:8> (PB7 : PB0)							
7:0	R/W							
	DOUT0CNx < 7:0> (PA7 : PA0)							

Bit 31 - 24 DOUT0CNx <31:24> Port D Inputs PD7 to PD0 of the connector
Bit 23 - 16 DOUT0CNx <23:16> Port C Inputs PC7 to PC0 of the connector
Bit 15 - 8 DOUT0CNx <15:8> Port B Inputs PB7 to PB0 of the connector
Bit 7 - 0 DOUT0CNx <7:0> Port A Inputs PA7 to PA0 of the connector

Register DOUT1CNx (x = Connector number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	DOUT1CNx <31:24> (PH7 : PH0)							
23:16	R/W							
	DOUT1CNx <23:16> (PG7 : PG0)							
15:8	R/W							
	DOUT1CNx <15:8> (PF7 : PF0)							
7:0	R/W							
	DOUT1CNx < 7:0> (PE7 : PE0)							

Bit 31 - 24 DOUT1CNx <31:24> Port H Inputs PH7 to PH0 of the connector
Bit 23 - 16 DOUT1CNx <23:16> Port G Inputs PG7 to PG0 of the connector
Bit 15 - 8 DOUT1CNx <15:8> Port F Inputs PF7 to PF0 of the connector
Bit 7 - 0 DOUT1CNx <7:0> Port E Inputs PE7 to PE0 of the connector

Register DDIRCNx (x = number of the connector):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	R/W							
	DDIRCNx <7:0>							

Bit 31 - 8 reserved (value 0 is to be written)
Bit 7 - 0 DDIRCNx <7:0> (default = 1)
0 = IO PIN configured as output
1 = IO PIN configured as input
DDIRCNx <0> = Port A, DDIRCNx <1> = Port B,
DDIRCNx <2> = Port C, DDIRCNx <3> = Port D,
DDIRCNx <4> = Port E, DDIRCNx <5> = Port F,
DDIRCNx <6> = Port G, DDIRCNx <7> = Port H

Register DENCNx (x = number of the connector):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U				R/W			
	reserved				DENCNx <3:0>			

Bit 31 - 4 reserved (value 0 is to be written)
Bit 3 - 0 DENCNx <3:0> (default = 0)
0 = enable port groups
1 = disable port groups
DENCNx <0> = Port A + Port B
DENCNx <1> = Port C + Port D
DENCNx <2> = Port E + Port F
DENCNx <3> = Port G + Port H

6. Advanced Functions of Digital Inputs

6.1 Digital inputs with digital filters

The digital inputs of the connector CN1 on the board WASCO-PCIe8296 have their own configurable digital filters to filter spurious pulses and transients of the input signal.
For that the filter checks whether or not a signal is applied long enough, as shown in fig. 6.2. If this is not the case, a too short pulse for example will be ignored. In the register DINFILyCN1[1] you can adjust the minimum time of how long the signal has to be applied to be considered. You can adjust a filter width of 0 - 255µs in steps of 1µs.
In state of default the filter is deactivated, say the filter duration is 0 µs.

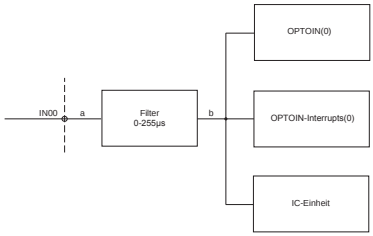


Fig. 6.1

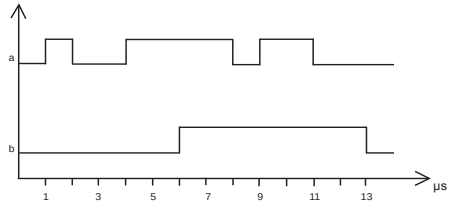


Fig 6.2

[1](y = Register number)

6.2 Interrupt functions of the digital inputs

To detect changes to the digital inputs without periodically querying the input state via PC, the WITIO-PCIe192ULTRA offers several interrupt options.
On one hand the card is able to trigger an interrupt on one of the inputs on a rising edge. On the other hand the card can signal to the PC a general change of the input states by an interrupt.
For further information please see the chapter Interrupt Controller.

6.2.1 Edge detection

In order to detect rising edges at the optocoupler inputs, each single input provides an edge detection with connectable interrupt function. For this purpose, a 32-bit interrupt register (DINIFyCNx)[1] is provided which makes available one bit per input channel for edge detection. As soon as the card detects a rising edge, the respective bit is set in register DINIFyCNx[1]. If at least one of the enabled bits is set, this will be passed to the interrupt controller over a line.

The interrupt function is enabled by writing to the 32bit register DINFyeCNx[1]. Each single bit represents one input. As shown in the table port addresses (chapter 12.1) the respective bit indicates an activation of the interrupt function with a 1 and deactivation with a 0. So, if the bit is 0, the corresponding bit in the register DINIFyCNx[1] will be set on a rising edge, but it will not be considered when the interrupt is triggered.
All of the interrupt channels are deactivated in default state.

[1](y = Register number, x = Connector number)

After the interrupt has been triggered, the source must be determined in the relevant interrupt service routine by reading the register DINIFyCNx[1]. Then the bit has to be cleared by setting the source channel bit in the register DINIFyrCNx[1]. After the card has executed the reset command, the bit is reset automatically.

Application example:
You want an edge detection with interrupt triggering on the channel PA1of the connector CN1. The following example lists each step of how to perform the configuration and what needs to be done in the interrupt service routine to re-enable the interrupt.
Please note that in this example the interrupt configuration of the driver is not indicated. For the discription of this please refer to the driver.
Additional, more program examples are made available for download on our homepage.

Configuration:

1. Activation of the card‘s interrupt function (see chapter Interrupt Controller)
2. Enable the required interrupt
   Before enabling an edge detection interrupt, please check whether or not the edge memory register DINIF0CN1 is reset completely. Otherwise, an interrupt might be triggered immediately after enabling the interrupt. If not all of the bits are reset in register DINIF0CN1, value 0 is writtenxffffffff(hex) into the register DINIF0rCN1.
   See the table how to set bit 1 in register DINIF0eCN1 to activate an edge detection interrupt on channel PA1. This way, with the help of the PCIe write command, the value 0x00000002(hex) resp. 2(dec) is written to this register.

[1](y = Register number, x = Connector number)

Interrupt Service Routine

1. To determine the interrupt source, the edge memory register DINIF0CN1 has to be read out (return value here 0x00000002(hex)). If other sources are available, such as timer etc., please check in INTCON register whether or not the interrupt received from the PC is derived from the DINIFyCNx register.
2. Once the source is identified, the source bit must be deleted.
For this purpose, in our case write 0x00000002(hex) to the register DINIF0rCN1.

Attention:
If in that time further interrupts were triggered (e.g. Timer), these must be deleted in their respective registers, too. Only after all of the activated interrupt registers have been reset to 0 again, another interrupt can be triggered.

6.2.2 Port changes

If the digital inputs often need to be queried to detect changes, another interrupt function can be used to relieve the PC. For this the WASCO-PCIe8296 provides the possibiltiy to trigger an interrupt in the event of a change at the inputs.
To enable this interrupt function on one hand the register DINICe has to be set to 0x00000001. On the other hand, the user can determine via the 32-bit register DINICCyeCNx(1), which one of the inputs should be considered for the detection. In the event of a change at the inputs, the corresponding bit is set in register DINICyCNx(1). To re-enable the interrupt after having been triggered, the corresponding bit in register DINICyrCNx(1) has to be set.
After the reset, the reset bit will be reset automatically.

[1](y = Register number, x = Connector number)

6.3 Port addresses

Offset-Address	Register Name	Bit Range	Bits																
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x0640	DINFIL0CN1	31:16	DINFIL0CN1 <31:16>															
		15:0	DINFIL0CN1 <15:0>															
0x0640 + 4*y	DINFILyCN1	31:16	DINFILyCN1 <31:16>															
		15:0	DINFILyCN1 <15:0>															
0x073C	DINFIL63CN1	31:16	DINFIL63CN1 <31:16>															
		15:0	DINFIL63CN1 <15:0>															

Register DINFILyCNx[(1)]:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	DINFILyCNx < 31:24>							
23:16	R/W							
	DINFILyCNx < 23:16>							
15:8	R/W							
	DINFILyCNx < 15:8>							
7:0	R/W							
	DINFILyCNx < 7:0>							

Bit 31- 0 **DINFILyCNx <31:0>** (default = 0)
The value determines the filter duration of the filter y of the connector x in µs

Offset Address = 0x640 + DIN Number * 4 (CN1)

[1](y = Register number, x = Connector number)

7. Advanced Functions of the Digital Outputs

7.1 Basic function

The basic function of the digital outputs allows the output of LOW and HIGH signals at the single outputs by writing to the 32-bit register DOUTyCNx[(1)]. In these registers every single bit stands for one digital output, as shown in table Port Addresses.

For example, if you want to set every third output of the first two ports of the CN1 plug to the level 3.3V or. 5V, you have to write the value 0 x4444(hex), 17476(dec) resp. 0b0100010001000100(bin) to the register DOUT0CN1.

7.2 Assigning digital outputs with other hardware components

In addition to the basic function, which allows easy access to the digital outputs, it is possible to assign different hardware components to the individual outputs, such as a PWM output (see Fig. 7.1). For this purpose, the first 8 digital outputs of each connector (PA0 to PA7) have a multiplexer with a 4-bit addressing (= up to 16 different sources). As a default source, the register DOUTyCNx [(1)] is specified as peripheral after a reset or when booting the PC.

To change the source, the source address (see Fig. 7.2) has to be written to the register DOUTMUXyCNx[(1)].

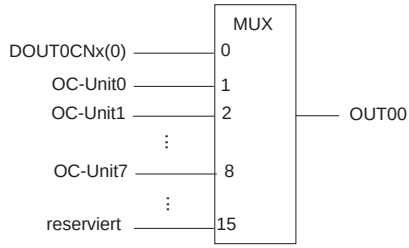


Fig. 7.1

Adresse	Peripherie
0x0 (default)	DOUT0CNx(z)
0x1	OC-Unit0
0x2	OC-Unit1
⋮	⋮
0x8	OC-Unit7
0x9 - 0xF	reserviert

Fig. 7.2

[1](y = Register number, x = Connector number)

7.3 Port addresses

Offset-Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x03C0	DOUTMUX0CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX0CN1 <3:0>	
0x03C4	DOUTMUX1CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX1CN1 <3:0>	
0x03C8	DOUTMUX2CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX2CN1 <3:0>	
0x03CC	DOUTMUX3CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX3CN1 <3:0>	
0x03D0	DOUTMUX4CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX4CN1 <3:0>	
0x03D4	DOUTMUX5CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX5CN1 <3:0>	
0x03D8	DOUTMUX6CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX6CN1 <3:0>	
0x03DC	DOUTMUX7CN1	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX7CN1 <3:0>	
0x03E0	DOUTMUX0CN2	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX0CN2 <3:0>	
0x03E4	DOUTMUX1CN2	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX1CN2 <3:0>	
0x03E8	DOUTMUX2CN2	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX2CN2 <3:0>	
0x03EC	DOUTMUX3CN2	31:16	reserved (*)															
		15:0	reserved (*)														DOUTMUX3CN2 <3:0>	

(*) reserved area has to be assigned with 0

Offset-Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x03F0	DOUTMUX4CN2	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX4CN2 <3:0>			
0x03F4	DOUTMUX5CN2	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX5CN2 3:0>			
0x03F8	DOUTMUX6CN2	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX6CN2 <3:0>			
0x03FC	DOUTMUX7CN2	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX7CN2 <3:0>			
0x0400	DOUTMUX0CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX0CN3 <3:0>			
0x0404	DOUTMUX1CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX1CN13<3:0>			
0x0408	DOUTMUX2CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX2CN3 <3:0>			
0x040C	DOUTMUX3CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX3CN3 <3:0>			
0x0410	DOUTMUX4CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX4CN3 <3:0>			
0x0414	DOUTMUX5CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX5CN3 <3:0>			
0x0418	DOUTMUX6CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX6CN3 <3:0>			
0x041C	DOUTMUX7CN3	31:16	reserved (*)															
		15:0	reserved (*)												DOUTMUX7CN3 <3:0>			

(*) reserved area has to be assigned with 0

Register DOUTMUXyCNx[(1)]:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U				R/W			
	reserved				DOUTMUXyCNx <3:0>			

Bit 31 - 4 reserved (assign value 0)

Bit 3 - 0 **DOUTMUXyCNx <3:0>** (default = 0)
Determines which peripherals are linked to the digital output
0 = DOUT-Register
1 = OC-Unit0
2 = OC-Unit1
3 = OC-Unit2
4 = OC-Unit3
5 = OC-Unit4
6 = OC-Unit5
7 = OC-Unit6
8 = OC-Unit7
9 - 15 = reserved

[1](y = Register number, x = Connector number)

8. Counter

The board WASCO-PCIe8296 provides a total of 32 32-bit event counters (rising edges). Every single counter can be freely assigned to one digital input. Furthermore, each counter can trigger an interrupt in the event of an overflow.

8.1 Basic function

1. For using a counter start by selecting a source. For this purpose every counter has its own 32-bit register (COUNTMUXx[(1)]).
2. Next, the counter has to be preloaded via the register COUNTLDx [(1)]. In general, the value 0 is written to the register.
3. Finally, the counter is activated by setting the first bit in the register COUNTex[(1)]. From this point, the counter starts to count every rising edge. In the event of an overflow the bit corresponding to the counter is set in the register COUNTIR. To detect another overflow, this bit has to be cleared by setting the bit allocated to the counter in the register COUNTIRr.
4. To determine the counter value read out the register COUNTx[(1)].

8.2 Interrupt function

Every overflow of a counter sets the bit allocated to the counter in the register COUNTIR. When the interrupt line has been enabled by setting the relevant bit in the register COUNTIRe, then the overflow will be passed on to the interrupt controller. In order to be able to reset the overflow bit, the bit assigned to the counter has to be set in the register COUNTIRr. After an internal reset of an overflow bit, the bit set in the register COUNTIRr will be reset automatically.

[(1)] x = Counter number)

8.3 Port addresses

Offset-Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1000	COUNT0e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1004	COUNT1e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1008	COUNT2e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x100C	COUNT3e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1010	COUNT4e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1014	COUNT5e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1018	COUNT6e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x101C	COUNT7e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1020	COUNT8e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1024	COUNT9e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1028	COUNT10e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x102C	COUNT11e	31:16	reserved (*)															
		15:0	reserved (*)															en

(*) reserved area has to be assigned with 0

Offset-Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1030	COUNT12e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1034	COUNT13e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1038	COUNT14e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x103C	COUNT15e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1040	COUNT16e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1044	COUNT17e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1048	COUNT18e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x104C	COUNT19e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1050	COUNT20e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1054	COUNT21e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1058	COUNT22e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x105C	COUNT23e	31:16	reserved (*)															
		15:0	reserved (*)															en

(*) reserved area has to be assigned with 0

Offset-Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1060	COUNT24e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1064	COUNT25e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1068	COUNT26e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x106C	COUNT27e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1070	COUNT28e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1074	COUNT29e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1078	COUNT30e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x107C	COUNT31e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1100	COUNT0	31:16	COUNT0<31:16>															
		15:0	COUNT0<15:0>															
0x1104	COUNT1	31:16	COUNT1<31:16>															
		15:0	COUNT1<15:0>															
0x1108	COUNT2	31:16	COUNT2<31:16>															
		15:0	COUNT2<15:0>															
0x110C	COUNT3	31:16	COUNT3<31:16>															
		15:0	COUNT3<15:0>															

(*) reserved area has to be assigned with 0

Offset-Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1110	COUNT4	31:16	COUNT4<31:16>															
		15:0	COUNT4<15:0>															
0x1114	COUNT5	31:16	COUNT5<31:16>															
		15:0	COUNT5<15:0>															
0x1118	COUNT6	31:16	COUNT6<31:16>															
		15:0	COUNT6<15:0>															
0x111C	COUNT7	31:16	COUNT7<31:16>															
		15:0	COUNT7<15:0>															
0x1120	COUNT8	31:16	COUNT8<31:16>															
		15:0	COUNT8<15:0>															
0x1124	COUNT9	31:16	COUNT9<31:16>															
		15:0	COUNT9<15:0>															
0x1128	COUNT10	31:16	COUNT10<31:16>															
		15:0	COUNT10<15:0>															
0x112C	COUNT11	31:16	COUNT11<31:16>															
		15:0	COUNT11<15:0>															
0x1130	COUNT12	31:16	COUNT12<31:16>															
		15:0	COUNT12<15:0>															
0x1134	COUNT13	31:16	COUNT13<31:16>															
		15:0	COUNT13<15:0>															
0x1138	COUNT14	31:16	COUNT14<31:16>															
		15:0	COUNT14<15:0>															
0x113C	COUNT15	31:16	COUNT15<31:16>															
		15:0	COUNT15<15:0>															

Offset-Address	Register Name	Bit Range	Bits																
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1140	COUNT16	31:16	COUNT16<31:16>															
		15:0	COUNT16<15:0>															
0x1144	COUNT17	31:16	COUNT17<31:16>															
		15:0	COUNT17<15:0>															
0x1148	COUNT18	31:16	COUNT18<31:16>															
		15:0	COUNT18<15:0>															
0x114C	COUNT19	31:16	COUNT19<31:16>															
		15:0	COUNT19<15:0>															
0x1150	COUNT20	31:16	COUNT20<31:16>															
		15:0	COUNT20<15:0>															
0x1154	COUNT21	31:16	COUNT21<31:16>															
		15:0	COUNT21<15:0>															
0x1158	COUNT22	31:16	COUNT22<31:16>															
		15:0	COUNT22<15:0>															
0x115C	COUNT23	31:16	COUNT23<31:16>															
		15:0	COUNT23<15:0>															
0x1160	COUNT24	31:16	COUNT24<31:16>															
		15:0	COUNT24<15:0>															
0x1164	COUNT25	31:16	COUNT25<31:16>															
		15:0	COUNT25<15:0>															
0x1168	COUNT26	31:16	COUNT26<31:16>															
		15:0	COUNT26<15:0>															
0x116C	COUNT27	31:16	COUNT27<31:16>															
		15:0	COUNT27<15:0>															

Offset-Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1170	COUNT28	31:16	COUNT28<31:16>															
		15:0	COUNT28<15:0>															
0x1174	COUNT29	31:16	COUNT29<31:16>															
		15:0	COUNT29<15:0>															
0x1178	COUNT30	31:16	COUNT30<31:16>															
		15:0	COUNT30<15:0>															
0x117C	COUNT31	31:16	COUNT31<31:16>															
		15:0	COUNT31<15:0>															
0x1180	COUNTMUX0	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX0 <7:0>				
0x1184	COUNTMUX1	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX1 <7:0>				
0x1188	COUNTMUX2	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX2 <7:0>				
0x118C	COUNTMUX3	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX3 <7:0>				
0x1190	COUNTMUX4	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX4 <7:0>				
0x1194	COUNTMUX5	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX5 <7:0>				
0x1198	COUNTMUX6	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX6 <7:0>				
0x119C	COUNTMUX7	31:16	reserved (*)															
		15:0	reserved (*)											COUNTMUX7 <7:0>				

(*) reserved area has to be assigned with 0

Offset-Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x11A0	COUNTMUX8	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX8 <7:0>							
0x11A4	COUNTMUX9	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX9 <7:0>							
0x11A8	COUNTMUX10	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX10 <7:0>							
0x11AC	COUNTMUX11	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX11 <7:0>							
0x11B0	COUNTMUX12	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX12 <7:0>							
0x11B4	COUNTMUX13	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX13 <7:0>							
0x11B8	COUNTMUX14	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX14 <7:0>							
0x11BC	COUNTMUX15	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX15 <7:0>							
0x11C0	COUNTMUX16	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX16 <7:0>							
0x11C4	COUNTMUX17	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX17 <7:0>							
0x11C8	COUNTMUX18	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX18 <7:0>							
0x11CC	COUNTMUX19	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX19 <7:0>							

(*) reserved area has to be assigned with 0

Offset-Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x11D0	COUNTMUX20	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX20 <7:0>							
0x11D4	COUNTMUX21	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX21 <7:0>							
0x11D8	COUNTMUX22	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX22 <7:0>							
0x11DC	COUNTMUX23	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMU23 <7:0>							
0x11E0	COUNTMUX24	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX24 <7:0>							
0x11E4	COUNTMUX25	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX25 <7:0>							
0x11E8	COUNTMUX26	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX26<7:0>							
0x11EC	COUNTMUX27	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX27 <7:0>							
0x11F0	COUNTMUX28	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX28 <7:0>							
0x11F4	COUNTMUX29	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX29 <7:0>							
0x11F8	COUNTMUX30	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX30 <7:0>							
0x11FC	COUNTMUX31	31:16	reserved (*)															
		15:0	reserved (*)								COUNTMUX31 <7:0>							

(*) reserved area has to be assigned with 0

Register COUNTxe: (x = Counter number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							R/W
	reserved							en

Bit 31 - 1 reserved (value 0 is written)

Bit 0 **COUNTxe<0>** (default = 0)
Deactivate or activate the counter
0 = deactivate (default)
1 = activate

Register COUNTx: (x = Counter number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	COUNTx <31:24>							
23:16	R/W							
	COUNTx <23:16>							
15:8	R/W							
	COUNTx <15:8>							
7:0	R/W							
	COUNTx <7:0>							

Bit 31 - 0 **COUNTx <31:0>** (default = 0)
This register allows to read out the current counter value of the counter x and to write to (for example for the initial state).

Register COUNTMUXx: (x = Counter number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	R/W							
	COUNTMUXx <7:0>							

Bit 31 - 8 reserved (value 0 is written)

Bit 7 - 0 **COUNTMUXx <7:0>** (default = 0)

The register value determines the card peripheral applied to the counter

0 = digital input 0 CN1 (default)

1 = digital input 1 CN1

.

.

.

63 = digital input 63 CN1

64 = digital input 0 CN2

.

.

.

127 = digital input 63 CN2

128 = digital input 0 CN3

.

.

.

191 = digital input 63 CN3

255 - 192 = reserved -> assign with 0

9. Timer

The available 32-bit timers can be used as timers or as configurable interval interrupt triggers. For this, intervals between 0 and 4294967295µs can be adjusted in steps of 1µs.

9.1 Using as an interval interrupt trigger

1. Start with clearing the timer x by deleting the bit 0 of the register TIMERxe [(1)] and then reset the timer. This reset is executed by writing the value 0 to the register TIMERx [(1)].
2. Next, determinate the interval. The duration of the interval is set in the writable 32-bit register TIMERCOMPx [(1).]
Interval duration = (TIMERCOMPx [(1)] + 1)*1µs
3. In order to trigger an interrupt after the interval has elapsed, the timer has to be enabled. For this, set the corresponding bit in register TIMERIRe. (Attention: the interrupt controller has to be enabled, too)
4. The timer being configured completely, activate it by setting bit 0 in the register TIMERxe [(1)].
5. If the interrupt has been triggered, this can be checked in the register TIMERIR. To receive a new interrupt, the source bit must be cleared by setting the respective reset bit in register TIMERIRr.

([(1)] x = Timer number)

9.2 Port addresses

Offset-Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1400	TIMER0e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1404	TIMER1e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1420	TIMER0	31:16	TIMER0 <31:16>															
		15:0	TIMER0 <15:0>															
0x1424	TIMER1	31:16	TIMER1 <31:16>															
		15:0	TIMER0 <15:0>															
0x1430	TIMERCOMP0	31:16	TIMERCOMP0 <31:16>															
		15:0	TIMERCOMP0 <15:0>															
0x1434	TIMERCOMP1	31:16	TIMERCOMP1 <31:16>															
		15:0	TIMERCOMP1 <15:0>															

(*) reserved area has to be assigned with 0

Register TIMERxe:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							R/W
	reserved							en

Bit 31 - 1 reserved (value 0 is written)

Bit 0 **TIMERxe<0>** (default = 0)
start or stop the timer
0 = stopped (default)
1 = started

(x = Timer number)

Register TIMERx:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	TIMERx<31:24>							
23:16	R/W							
	TIMERx<23:16>							
15:8	R/W							
	TIMERx<15:8>							
7:0	R/W							
	TIMERx<7:0>							

Bit 31 - 0 **TIMERx<0>** (default = 0)
This register allows to read out the current value of the timer x and to write to (for example for the initial value).

(x = Timer number)

Register TIMERCOMPx:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	TIMERCOMPx<31:24>							
23:16	R/W							
	TIMERCOMPx<23:16>							
15:8	R/W							
	TIMERCOMPx<15:8>							
7:0	R/W							
	TIMERCOMPx<7:0>							

Bit 31 - 0 **TIMERCOMPx<0>** (default = 0)
The value of the register TIMERCOMP determines the interval duration of the timer.
TIMERCOMP = Interval duration - 1

(x = Timer number)

10. Input Capture Unit

The Input Capture Units (IC-Unit) allow the user to measure pulse duration and period duration of received signals. Each one of the units has its own 32-bit timer for time measurement in steps of 1µs, and can be assigned to any digital input by programming.

10.1 Continuous measurement of periodic signals

In this mode, the input signal is scanned regularly when the function is activated, and the period duration and pulse duration is determined. For this, the unit starts measuring at the first rising edge at the input and ends it at the following rising edge. Measurement completed, the period duration and the pulse duration is computed automatically and the values are written to the registers ICPERIODLx and ICPULSLx. At the next rising edge, the unit starts to measure by itself.

10.1.1 Application

1. Make sure, the intended unit to be deactivated before configuration. The IC Unit is disabled by clearing the bit 0 in the register ICUNITex.
2. When the IC Unit is deactivated, carry out the configuration in register ICCONFIGx. For the continuous measurement of periodic signals write the value b0000(bin) in the mode section.
3. When the unit is configured, then the source has to be selected by writing it into register ICMUXx.
4. Now to start the measurement, set bit 0 in the register ICUNITex.

10.1.2 Interrupt function

In addition to the measurement of the period and the pulse duration, it is possible to trigger an interrupt after having completed the measurement. For this you activate the interrupt function by setting the corresponding bit in register ICUNITRe. When the interrupt is triggered, read out the source in register ICUNITIR and again activate the source by setting the corresponding bit in register ICUNITIRr.

(x = IC number)

10.2 Port addresses

Offset-Address	Register Name	Bit Range	Bits																				
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x14C0	ICUNIT0e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x14C4	ICUNIT1e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x14E0	ICCONFIG0	31:16	reserved (*)															
		15:0	reserved (*)													ICMODE0		
0x14E4	ICCONFIG1	31:16	reserved (*)															
		15:0	reserved (*)													ICMODE1		
0x1500	ICMUX0	31:16	reserved (*)															
		15:0	reserved (*)											ICMUX0 <7:0>				
0x1504	ICMUX1	31:16	reserved (*)															
		15:0	reserved (*)											ICMUX1 <7:0>				
0x1540	ICPULS0	31:16	ICPULS0 <31:16>															
		15:0	ICPULS0 <15:0>															
0x1544	ICPULS1	31:16	ICPULS1 <31:16>															
		15:0	ICPULS1 <15:0>															
0x1560	ICPERIOD0	31:16	ICPERIOD0 <31:16>															
		15:0	ICPERIOD0 <15:0>															
0x1504	ICPERIOD0	31:16	ICPERIOD1 <31:16>															
		15:0	ICPERIOD1 <15:0>															

(*) reserved area has to be assigned with 0

Register ICUNITxe: (x = IC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							R/W
	reserved							en

Bit 31 - 1 reserved (value 0 is written)
Bit 0 **ICUNITxe<0>** (default = 0)
Start and stop the IC-Unit
0 = stopped (default)
1 = started (operating measuring)

Register ICCONFIGx: (x = IC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U				R/W			
	reserved				ICMODEx <3:0>			

Bit 31 - 4 reserved (value 0 is written)
Bit 3 - 0 **ICMODEx<3:0>** (default = 0)
Determines the mode the IC Unit is working with
0 = Mode 0 operates continuous measurement of pulse and period duration of periodic signals (default)
1 - 15 = reserved (assign with 0)

Register ICMUXx: (x = IC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	R/W							
	ICMUXx <7:0>							

Bit 31 - 8 reserved (value 0 is written)

Bit 7 - 0 **ICMUXx <7:0>** (default = 0)

The register value determines the card peripheral the IC unit is applied to

0 = digital input 0 CN1 (default)

1 = digital input 1 CN1

.

.

.

63 = digital input 63 CN1

64 = digital input 0 CN2

.

.

.

127 = digital input 63 CN2

128 = digital input 0 CN3

.

.

.

191 = digital input 63 CN3

255 - 192 = reserved -> assign with 0

Register ICPULSx: (x = IC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R							
	ICPULSx <31:24>							
23:16	R							
	ICPULSx <23:16>							
15:8	R							
	ICPULSx <15:8>							
7:0	R							
	ICPULSx <7:0>							

Bit 31 - 0 **ICPULSx<31:0>**
From this register read out the last measured pulse duration in µs

Register ICPERIODx: (x = IC number

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R							
	ICPERIODx <31:24>							
23:16	R							
	ICPERIODx <23:16>							
15:8	R							
	ICPERIODx <15:8>							
7:0	R							
	ICPERIODx <7:0>							

Bit 31 - 0 **ICPERIODx<31:0>**
From this register read out the last measured period duration in µs

11. Output Compare Unit

The board WASCO-PCIe8296 brings the option to the user to link a PWM function to the outputs or to return discrete pulses via the Output Compare Units. In this case, square-wave signals with a period duration of 2 to 2^{32} µs and a pulse duration of 1 to 2^{32} µs can be generated.

11.1 PWM

The Output Compare Units enable the user to apply a PWM to the first 8 digital outputs of each connector (PA0 bis PA7)

11.1.1 Operating principle

In order to realize the PWM, the OC Unit is using a writable 32-bit timer with adjustable period duration in µs (OCPERIODx) and a two-level compare register (OCUNITORx) to set the pulse duration in µs. If the OC Unit is deactivated, there is a LOW at the output. If the OC Unit is started in PWM mode, the timer starts counting in µs-clock and the OC output remains LOW. When the timer reaches the value in the register OCPERIODx, it will run over to the next cycle and start counting again at 0. Also at an overflow, the pipeline register connected to the timer will take-over the pulse duration configured in the register OCUNITORx, as soon as the OC output is set to HIGH (provided the pulse duration is not 0µs). When the timer value (OCTIMERx) matches the value of the pipeline register connected to the timer, the output will be set to LOW until the next timer overflow.
The application of the two-level pulse duration register ensures the complete return of each period prior to transfer, if the pulse duration is changed during OC operating.
If you want to skip the first period after starting the OC unit, in which no pulse at the output is emitted, you can preload the timer accordingly with another value than 0 (-> reduction of the period).

(x = OC number)

11.1.2 Calculation of the register values

OCPERIODx [2] = Period_duration_in_µs + 1 [µs]

OCUNITORx [2] = Pulse_duration_in_µs [µs]

OCTIMER = clocks [µs]

11.1.3 Application example

1. Deactivate the OC unit by clearing the corresponding bit in the register OCUNITxe[2].
2.Connect the OC unit to the required digital output. For this, select the source in the DOUTMUXyCNx[1] register allocated to the digital output (see chapter „Assigning digital outputs with other hardware components“)
3. Preload the OC timer of the OC unit. Here usually the value 0x00000000 is written to the register OCTIMERx [2].
4. Define the period duration of the PWM. For this, write the period duration to the register OCUNITORx [2] as follows:
OCPERIODx [2] = Period duration - 1 [µs]
5. Define the pulse duration. For this, write the pulse duration to the register OCPULSx [2] as follows:
OCPULSx [2] = Pulse duration [µs]
6. Select the mode of the OC unit. For using the PWM, the value 0 has to be written to the register OCCONFIGx [2]
7. Activate the OC unit by setting the corresponding bit in the register OCUNITxe [2].

[1] (y = Register number, x = Connector number)
[2] (x = OC number)

11.2 Pulse output

In addition to the PWM, the OC unit makes it possible to output discrete µs-accurate pulses at the digital outputs.

11.2.1 Functionality

To output discrete positive pulses you first have to configure the registers. Then, each time the en-bit is cleared and then set in the register OCUNITxe, you can issue a pulse as shown in the figure below:

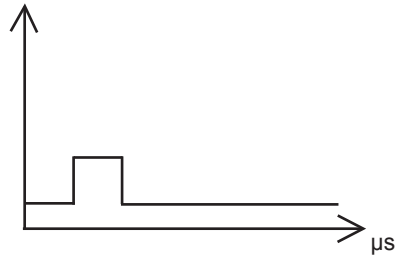


Abb. 11.1

To change the pulse duration, the OC unit always has to be deactivated (en-Bit in the OCUNITxe register cleared).

11.2.2 Calculation of the register values

OCPERIODx = Pulse_duration_in_µs

(x = OC number)

11.2.3 Application example

1. How to configure the OC unit

a) Deactivate the unit by clearing (= 0) the en-Bit in the register OCUNITxe [2]
b) Connect the OC unit to the required digital output. For this, select the source in the DOUTMUXyCNx [2] register (see chapter „Assigning digital outputs with other hardware components“)
c) Preload the OC timer of the OC unit with the value 0. For this, write 0x00000000 to the register OCTIMERx [2].
d) Load the pulse duration wanted to the register OCPERIODx [2] (see also chapter 11.2.2 Calculation of the register values)
e) Load the word 1 to the register OCUNITORx [2].
f) Select the mode Single Pulse by writing the value 1 to the mode section in the register OCCONFIGx [2].

2. How to output a pulse

a) Deactivate OC unit by clearing the enable bit in register OCUNITxe [2].
b) Activate OC unit by setting the enable bit in register OCUNITxe [2]. As a result the pulse is applied to the selected output.

[1] (y = Register number, x = Connector number)
[2] (x = OC number)

11.3 Port addresses

Offset-Address	Register Name	Bit Range	Bits																				
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x15C0	OCUNIT0e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x15C4	OCUNIT1e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x15C8	OCUNIT2e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x15CC	OCUNIT3e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x15D0	OCUNIT4e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x15D4	OCUNIT5e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x15D8	OCUNIT6e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x15DC	OCUNIT7e	31:16	reserved (*)															
		15:0	reserved (*)															en
0x1600	OCTIMER0	31:16	OCTIMER0 <31:16>															
		15:0	OCTIMER0 <15:0>															
0x1604	OCTIMER1	31:16	OCTIMER1 <31:16>															
		15:0	OCTIMER1 <15:0>															
0x1608	OCTIMER2	31:16	OCTIMER2 <31:16>															
		15:0	OCTIMER2 <15:0>															
0x160C	OCTIMER3	31:16	OCTIMER3 <31:16>															
		15:0	OCTIMER3 <15:0>															

(*) reserved area has to be assigned with 0

Offset-Address	Register Name	Bit Range	Bits																
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1610	OCTIMER4	31:16	OCTIMER4 <31:16>															
		15:0	OCTIMER4 <15:0>															
0x1614	OCTIMER5	31:16	OCTIMER5 <31:16>															
		15:0	OCTIMER5 <15:0>															
0x1618	OCTIMER6	31:16	OCTIMER6 <31:16>															
		15:0	OCTIMER6 <15:0>															
0x161C	OCTIMER7	31:16	OCTIMER7 <31:16>															
		15:0	OCTIMER7 <15:0>															
0x1620	OCUNITOR0	31:16	OCUNITOR0 <31:16>															
		15:0	OCUNITOR0 <15:0>															
0x1624	OCUNITOR1	31:16	OCUNITOR1 <31:16>															
		15:0	OCUNITOR1 <15:0>															
0x1628	OCUNITOR2	31:16	OCUNITOR2 <31:16>															
		15:0	OCUNITOR2 <15:0>															
0x162C	OCUNITOR3	31:16	OCUNITOR3<31:16>															
		15:0	OCUNITOR3 <15:0>															
0x1630	OCUNITOR4	31:16	OCUNITOR4 <31:16>															
		15:0	OCUNITOR4 <15:0>															
0x1634	OCUNITOR5	31:16	OCUNITOR5 <31:16>															
		15:0	OCUNITOR5 <15:0>															
0x1638	OCUNITOR6	31:16	OCUNITOR6 <31:16>															
		15:0	OCUNITOR6 <15:0>															
0x163C	OCUNITOR7	31:16	OCUNITOR7<31:16>															
		15:0	OCUNITOR7 <15:0>															

Offset-Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1660	OCPERIOD0	31:16	OCPERIOD0 <31:16>															
		15:0	OCPERIOD0 <15:7>															
0x1664	OCPERIOD1	31:16	OCPERIOD1 <31:16>															
		15:0	OCPERIOD1 <15:7>															
0x1668	OCPERIOD2	31:16	OCPERIOD2 <31:16>															
		15:0	OCPERIOD2 <15:7>															
0x166C	OCPERIOD3	31:16	OCPERIOD3 <31:16>															
		15:0	OCPERIOD3 <15:7>															
0x1670	OCPERIOD4	31:16	OCPERIOD4 <31:16>															
		15:0	OCPERIOD4 <15:7>															
0x1674	OCPERIOD5	31:16	OCPERIOD5 <31:16>															
		15:0	OCPERIOD4 <15:7>															
0x1678	OCPERIOD6	31:16	OCPERIOD6 <31:16>															
		15:0	OCPERIOD6 <15:7>															
0x167C	OCPERIOD7	31:16	OCPERIOD7 <31:16>															
		15:0	OCPERIOD7 <15:7>															

Offset-Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x1680	OCCONFIG0	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE0 <3:0>			
0x1684	OCCONFIG1	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE1 <3:0>			
0x1688	OCCONFIG2	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE2 <3:0>			
0x168C	OCCONFIG3	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE3 <3:0>			
0x1690	OCCONFIG4	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE4 <3:0>			
0x1694	OCCONFIG5	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE5 <3:0>			
0x1698	OCCONFIG6	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE6 <3:0>			
0x168C	OCCONFIG7	31:16	reserved (*)															
		15:0	reserved (*)												OCMODE7 <3:0>			

(*) reserved area has to be assigned with 0

Register OCUNITxe: (x = OC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							R/W
	reserved							en

Bit 31 - 1 reserved (value 0 is to be written)
Bit 0 **OCUNITxe<0>** (default = 0)
start or stop OC-Unit
0 = stopped (default)
1 = started

Register OCTIMERx: (x = OC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	OCTIMERx <31:24>							
23:16	R/W							
	OCTIMERx <23:16>							
15:8	R/W							
	OCTIMERx <15:8>							
7:0	R/W							
	OCTIMERx <7:0>							

Bit 31 - 0 **OCTIMERx<31:0>** (default = 0)
Read out the current value of the OC timer x from this register or write to (e.g. for the initial state)

Register OCUNITORx: (x = OC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R							
	OCUNITORx <31:24>							
23:16	R							
	OCUNITORx <23:16>							
15:8	R							
	OCUNITORx <15:8>							
7:0	R							
	OCUNITORx <7:0>							

Bit 31 - 0 **OCUNITORx<31:0>** (default = 0)
Detemines the pulse duration of the OC Unit x in µs
Pulse duration = OCUNITORx [µs]

Register OCPERIODx: (x = OC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R							
	OCPERIODx <31:24>							
23:16	R							
	OCPERIODx <23:16>							
15:8	R							
	OCPERIODx <15:8>							
7:0	R							
	OCPERIODx <7:0>							

Bit 31 - 0 **OCPERIODx<31:0>** (default = 0)
Detemines the period duration of the OC Unit x in µs
Period duration = OCPERIODx +1 [µs]

Register OCCONFIGx: (x = OC number)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U				R/W			
	reserved				OCMODEx <3:0>			

Bit 31 - 4 reserved (value 0 is to be written)

Bit 3 - 0 **OCMODEx<3:0>** (default = 0)
Determines the mode of the OC Unit
0 = Mode 0 -> Pulse width modulation (default)
1 = Mode 1 -> output of discrete pulses
2 - 15 = reserved (value 0 is to be written)

12. Interrupt Controller

The interrupt controller is used to process the single interrupts from the various possible sources. It can enable single interrupt sources or can detect the sources of triggered interrupts.
The 32 bit register INTCON is the Central Unit as shown in fig 12.1. Here all possible interrupt sources (partially already processed) are merged.
If an interrupt is triggered, e.g. by an edge on a digital input, this is passed to the first bit in the register INTCON. Whenever the register value of INTCON is nonzero (one or more interrupts are applied) this will be forwarded to INT. Thus, INT represents a type of gate register. The interrupt will be forwarded to the PC when the board‘s interrupt function is enabled (INTe = 1) and the register is reset. The interrupt line to the PC is blocked for any further interrupts when an interrupt has been triggered. To enable again the line, the source has to be determined and the trigger serviced. During this time it is possible to trigger further interrupts from other sources on the board (e.g. by other egde inputs or timers), but they will not be forwarded to the PC. When an interrupt trigger is serviced and the respective source is enabled again, the respective bit in register INTCON will be set to 0 automatically. All of the interrupt triggers being serviced and reset (INT = 0), the register INT can be cleared by setting the first bit in the INTr register and another interrupt can be forwarded to the PC.

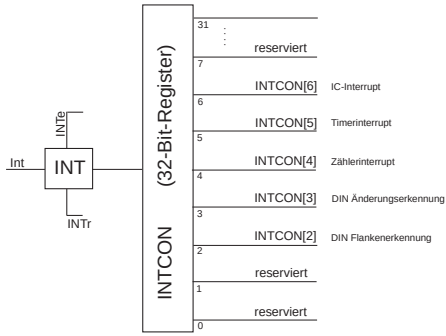


Fig. 12.1

As several sources of a digital input edge interrupt are to consider, the source lines to the register INTCON usually are conditioned. This means, an additional 32 bit register can be applied to the respective bit of the register INTCON. In case of an edge detection of the digital inputs, this are the registers DINF0CN1, DINF1CN1, DINF0CN2, DINF1CN2, DINF0CN3 and DINF1CN3. In these registers every bit represents one digital input (see register description). Every single input can be armed individually to be an interrupt source (DINFyeCNx) and enabled again after an interrupt is triggered and serviced (DINFyrCNx).
The process completed, the respective bit in the register INTCON is set to 0 automatically.

Application

1. How to configure

a) Check whether or not all interrupt sources are cleared (INTCON = 0)
b) Enable single interrupt sources (see documentations of the respective peripherals)
c) Enable interrupt function (INTe = 1)

2. Interrupt routine

a) Identify interrupt source peripherals by reading INTCON and corresponding peripheral register if required
b) Clear interrupt
c) Check if there are any outstanding interrupts (INTCON = 0?)
d) if c) is the case, clear all other interrupts as well
e) Enable again the interrupt function (INTr = 1)

12.1 Port addresses

Offset Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x0280	INTe	31:16	reserved (*)															
		15:0	reserved (*)															en
0x0284	INTr	31:16	reserved (*)															
		15:0	reserved (*)															re
0x0288	INTCON	31:16	reserved (*)															
		15:0	reserved (*)								INT7	INT6	INT5	INT4	INT3	INT2	reserved (*)	
0x2C0	DINIF0eCN1	31:16	DINIF0eCN1 <31:16>															
		15:0	DINIF0eCN1 <15:0>															
0x2C4	DINIF1eCN1	31:16	DINIF1eCN1 <31:16>															
		15:0	DINIF1eCN1 <15:0>															
0x2C8	DINIF0eCN2	31:16	DINIF0eCN2 <31:16>															
		15:0	DINIF0eCN2 <15:0>															
0x2CC	DINIF1eCN2	31:16	DINIF1eCN2 <31:16>															
		15:0	DINIF1eCN2 <15:0>															
0x2D0	DINIF0eCN3	31:16	DINIF0eCN3 <31:16>															
		15:0	DINIF0eCN3 <15:0>															
0x2D4	DINIF1eCN3	31:16	DINIF1eCN3 <31:16>															
		15:0	DINIF1eCN3 <15:0>															
0x2E0	DINIF0rCN1	31:16	DINIF0rCN1 <31:16>															
		15:0	DINIF0rCN1 <15:0>															
0x2E4	DINIF1rCN1	31:16	DINIF1rCN1 <31:16>															
		15:0	DINIF1rCN1 <15:0>															
0x2E8	DINIF0rCN2	31:16	DINIF0rCN2 <31:16>															
		15:0	DINIF0rCN2 <15:0>															

(*) reserved area has to be assigned with 0

Offset Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x2EC	DINIF1rCN2	31:16	DINIF1rCN2 <31:16>															
		15:0	DINIF1rCN2 <15:0>															
0x2F0	DINIF0rCN3	31:16	DINIF0rCN3 <31:16>															
		15:0	DINIF0rCN3 <15:0>															
0x2F4	DINIF1rCN3	31:16	DINIF1rCN3 <31:16>															
		15:0	DINIF1rCN3 <15:0>															
0x300	DINIF0CN1	31:16	DINIF0CN1 <31:16>															
		15:0	DINIF0CN1 <15:0>															
0x304	DINIF1CN1	31:16	DINIF1CN1 <31:16>															
		15:0	DINIF1CN1 <15:0>															
0x308	DINIF0CN2	31:16	DINIF0CN2 <31:16>															
		15:0	DINIF0CN2 <15:0>															
0x30C	DINIF1CN2	31:16	DINIF1CN2 <31:16>															
		15:0	DINIF1CN2 <15:0>															
0x310	DINIF0CN3	31:16	DINIF0CN3 <31:16>															
		15:0	DINIF0CN3 <15:0>															
0x314	DINIF1CN3	31:16	DINIF1CN3 <31:16>															
		15:0	DINIF1CN3 <15:0>															
0x4B0	DINICe	31:16	reserved (*)															
		15:0	reserved (*)															en

(*) reserved area has to be assigned with 0

Offset Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x4B4	DINICC0eCN1	31:16	DINICC0eCN1 <31:16>															
		15:0	DINICC0eCN1<15:0>															
0x4B8	DINICC1eCN1	31:16	DINICC1eCN1 <31:16>															
		15:0	DINICC1eCN1<15:0>															
0x4BC	DINICC0eCN2	31:16	DINICC0eCN2 <31:16>															
		15:0	DINICC0eCN2<15:0>															
0x4C0	DINICC1eCN2	31:16	DINICC1eCN2 <31:16>															
		15:0	DINICC1eCN2<15:0>															
0x4C4	DINICC0eCN3	31:16	DINICC0eCN3 <31:16>															
		15:0	DINICC0eCN3<15:0>															
0x4C8	DINICC1eCN3	31:16	DINICC1eCN3 <31:16>															
		15:0	DINICC1eCN3<15:0>															
0x4D4	DINICr	31:16	reserved (*)															
		15:0	reserved (*)															reset
0x4D8	DINIC	31:16	reserved (*)															
		15:0	reserved (*)															DINIC<0>

(*) reserved area has to be assigned with 0

Offset Address	Register Name	Bit Range	Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x0340	COUNTIRe	31:16	reserved (*)															
		15:0	COUNTIRe <15:0>															
0x0344	COUNTIRr	31:16	reserved (*)															
		15:0	COUNTIRr <15:0>															
0x0348	COUNTIR	31:16	COUNTIR<31:16>															
		15:0	COUNTIR<15:0>															
0x0360	TIMERIRe	31:16	reserved (*)															
		15:0	reserved (*)														TIMERIRe<1:0>	
0x0364	TIMERIRr	31:16	reserved (*)															
		15:0	reserved (*)														TIMERIRr<1:0>	
0x0368	TIMERIR	31:16	reserved (*)															
		15:0	reserved (*)														TIMERIR<1:0>	
0x036C	ICUNITIRe	31:16	reserved (*)															
		15:0	reserved (*)														ICUNITIRe<1:0>	
0x0370	ICUNITIRr	31:16	reserved (*)															
		15:0	reserved (*)														ICUNITIRr<1:0>	
0x0374	ICUNITIR	31:16	reserved (*)															
		15:0	reserved (*)														TIMERIR<1:0>	

(*) reserved area has to be assigned with 0

Register INTe:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							R/W
	reserved							en

Bit 31 - 1 reserved (value 0 is to be written)
Bit 0 **INTe<0>** (default = 0)
Enable or lock the card‘s interrupt function
0 = Interrupt locked (default)
1 = Interrupt enabled

Register INTr:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							W
	reserved							en

Bit 31 - 1 reserved (value 0 is to be written)
Bit 0 **INTr<0>** The register INTCON is set to 0 by writing a 1 and a new interrupt can be triggered.

Register INTCON:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U	R			R		U	
	reserved	INTCON <6:4>			INTCON <1:0>		reserved	

Bit 31 - 7 reserved (value 0 is written)

Bit 6 **INTCON<6>**: indicates an interrupt from one of the OC Units
0 = no interrupt was triggered by an OC Unit
1 = one of the OC Units has triggered an interrupt

Bit 5 **INTCON<5>**: indicates an interrupt from one of the timers
0 = no interrupt was triggered by a timer
1 = one of the timers has triggered an interrupt

Bit 4 **INTCON<4>**: indicates an interrupt from one of the counters
0 = no interrupt was triggered by a counter
1 = one of the counters has triggered an interrupt

Bit 3 **INTCON<3>**: indicates an interrupt triggered by a change at the digital input enabled for this interrupt
0 = no change interrupt was triggered by the digital inputs
1 = the digital inputs triggered a change interrupt

Bit 2 **INTCON<2>**: indicates an interrupt triggered by a rising edge at the digital input enabled for this interrupt
0 = no edge interrupt was triggered by the digital inputs
1 = the digital inputs triggered an edge interrupt

Bit 1 - 0 reserved (value 0 is written)

Register DINIF0eCNx (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	DINIF0eCNx <31:24>							
23:16	R/W							
	DINIF0eCNx <23:16>							
15:8	R/W							
	DINIF0eCNx <15:8>							
7:0	R/W							
	DINIF0eCNx <7:0>							

Bit 31 - 0 **DINIF0eCNx**<31:0> (default = 0) In this register section single digital inputs can be enabled as a source to trigger an interrupt on a positive edge. Every bit corresponds to a digital input (e.g. PA0 => DINIF0eCNx<0>, PB5 => DINIF0eCNx<13>). If a bit is 1, the function for an edge interrupt of the digital input is enabled, if it is 0 the function is blocked.

Register DINIF1eCNx (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	DINIF1eCNx <31:24>							
23:16	R/W							
	DINIF1eCNx <23:16>							
15:8	R/W							
	DINIF1eCNx <15:8>							
7:0	R/W							
	DINIF1eCNx <7:0>							

Bit 31 - 0 **DINIF1eCNx**<31:0> (default = 0) In this register section single digital inputs can be enabled as a source to trigger an interrupt on a positive edge. Every bit corresponds to a digital input (e.g. PE0 => DINIF1eCNx<0>, PF5 => DINIF1eCNx<13>). If a bit is 1, the function for an edge interrupt of the digital input is enabled, if it is 0 the function is locked.

Register DINIF0rCNx (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	W							
	DINIF0rCNx <31:24>							
23:16	W							
	DINIF0rCNx <32:16>							
15:8	W							
	DINIF0rCNx <15:8>							
7:0	W							
	DINIF0rCNx <7:0>							

Bit 31 - 0 **DINIF0rCNx<31:0>** Each bit corresponds to a digital input. (e.g PA0 => DINIF0rCNx<0>, PB5 => DINIF0rCNx<13>). If an edge interrupt has been triggered at a digital input, its signal bit in the DINIF0r register has to be reset. This is done by setting (= 1) the corresponding DINIF0rCNx bit. The DINIF0rCNx bits are set to 0 automatically after reset.

Register DINIFr1CNx (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	W							
	DINIF1rCNx <31:24>							
23:16	W							
	DINIF1rCNx <32:16>							
15:8	W							
	DINIF1rCNx <15:8>							
7:0	W							
	DINIF1rCNx <7:0>							

Bit 31 - 0 **DINIF1rCNx<31:0>** Each bit corresponds to a digital input. (e.g PE0 => DINIF1rCNx<0>, PF5 => DINIF1rCNx<13>). If an edge interrupt has been triggered at a digital input, its signal bit in the DINIF0r register has to be reset. This is done by setting (= 1) the corresponding DINIF0rCNx bit. The DINIF0rCNx bits are set to 0 automatically after reset.

Register DINIF0CNx (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	-							
23:16	U							
	-							
15:8	R							
	DINIF0CNx <15:8>							
7:0	R							
	DINIF0CNx <7:0>							

Bit 31 - 0 **DINIF0CNx<31:0>** indicates whether or not a rising edge is applied to one of the digital inputs. Each bit corresponds to a digital input (e.g. PA0 => DINIF0CNx<0>, PB5 => DINIF0CNx<13>). A 1 in the respective bit means, that a rising edge was applied to the input since the last reset, if it is 0 no edge was applied.

Register DINIF1CNx (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	-							
23:16	U							
	-							
15:8	R							
	DINIF0CNx <15:8>							
7:0	R							
	DINIF0CNx <7:0>							

Bit 31 - 0 **DINIF1CNx<31:0>** indicates whether or not a rising edge is applied to one of the digital inputs. Each bit corresponds to a digital input (e.g. PE0 => DINIF1CNx<0>, PF5 => DINIF1CNx<13>). A 1 in the respective bit means, that a rising edge was applied to the input since the last reset, if it is 0 no edge was applied.

Register DINICe:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							R/W
	reserved							en

Bit 31 - 1 reserved (value 0 is written)

Bit 0 **DINICe<0>** (default = 0) Enable interrupt function to detect changes to the digital inputs
0 = Interrupt disabled (default)
1 = enable interrupt

Register DINICC0eCNx: (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	DINICC0eCNx<31:24>							
23:16	U							
	DINICC0eCNx<23:16>							
15:8	R/W							
	DINICC0eCNx<15:8>							
7:0	R/W							
	DINICC0eCNx<7:0>							

Bit 31 - 0 **DINICC0eCNx<15:0>** (default = 0)
Enable or disable single digital inputs for the interrupt function to detect changes to the digital inputs. Each bit corresponds to a digital input (e.g. PA0 => DINICC0eCNx<0>, PB5 => DINICC0eCNx<13>)
0 = Interrupt disabled (default)
1 = enable interrupt

Register DINICC1eCNx: (x = Connector Number):

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	DINICC0eCNx<31:24>							
23:16	U							
	DINICC0eCNx<23:16>							
15:8	R/W							
	DINICC0eCNx<15:8>							
7:0	R/W							
	DINICC0eCNx<7:0>							

Bit 31 - 0 **DINICC1eCNx<15:0>** (default = 0)
Enable or disable single digital inputs for the interrupt function to detect changes to the digital inputs. Each bit corresponds to a digital input (e.g. PE0 => DINICC1eCNx<0>, PF5 => DINICC1eCNx<13>)
0 = Interrupt disabled (default)
1 = enable interrupt

Register DINICr:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U							W
	reserved							re

Bit 31 - 1 reserved (value 0 is written)
Bit 0 **DINICr<0>** (default = 0)
If an interrupt was triggered by a change to the digital inputs, the source register DINIC has to be reset to 0. This is done by setting (=1) the DINICr bit. The DINCr bit will be reset to 0 automatically after resetting.

Register DINIC:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	-							
23:16	U							
	-							
15:8	U							
	-							
7:0	U							R
	-							DINIC<0>

Bit 31 - 1 undefined

Bit 0 **DINIC<0>** indicates whether a change has occurred to an enabled digital input
0 = no change
1 = change to an enabled digital input

Register COUNTIRe:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W							
	COUNTIRe<31:24>							
23:16	R/W							
	COUNTIRe<23:16>							
15:8	R/W							
	COUNTIRe<15:8>							
7:0	R/W							
	COUNTIRe<7:0>							

Bit 31 - 0 **COUNTIRe<31:0>** (default = 0)
This enables the interrupt functions of the counters. Each bit corresponds to one counter (e.g. Counter 0 => COUNTIRe<0>, Counter 13 => COUNTIRe<13>)
0 = Interrupt disabled (default)
1 = enable interrupt

Register COUNTIRr:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	COUNTIRr<31:24>							
23:16	U							
	COUNTIRr<23:16>							
15:8	R/W							
	COUNTIRr<15:8>							
7:0	R/W							
	COUNTIRr<7:0>							

Bit 31 - 0 **COUNTIRr<31:0>**
Each bit corresponds to one counter (e.g. Counter 0 => COUNTIRr<0>, Counter 13 => COUNTIRr<13>). If an interrupt has been triggered by a counter, its signal bit has to be reset in register COUNTIR. This is done by setting (=1) the respective COUNTIRr bit. The COUNTIRr bits are reset to 0 automatically after resetting.

Register COUNTIR:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	COUNTIR<31:24>							
23:16	U							
	COUNTIR<23:16>							
15:8	R/W							
	COUNTIR<15:8>							
7:0	R/W							
	COUNTIR<7:0>							

Bit 31 - 0 **COUNTIR<31:0>** indicates whether an interrupt has been triggered by a counter. Each bit corresponds to one counter (e.g. Counter 0 => COUNTIR<0>, Counter 13 => COUNTIR<13>)
0 = no interrupt
1 = Interrupt triggered

Register TIMERIRe:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U						R/W	
	reserved						TIMERIRe <1:0>	

Bit 31 - 2 reserved (value 0 is written)

Bit 1 - 0 **TIMERIRe<1:0>** (default = 0) this enables the interrupt functions of the timers to be enabled. Each bit corresponds to a timer (e.g. Timer 0 => TIMERIRe<0>, Timer 1 => TIMERIRe<1>)
0 = Interrupt disabled (default)
1 = enable interrupt

Register TIMERIRr:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U						R/W	
	reserved						TIMERIRr <1:0>	

Bit 31 - 2 reserved (value 0 is written)

Bit 1 - 0 **TIMERIRr<1:0>** Each bit corresponds to a timer (e.g. Timer 0 => TIMERIRr<0>, Timer 1 => TIMERIRr<1>). If a timer has triggered an interrupt its signal bit has to be reset in the TIMERIR register. This is done by setting (= 1) the corresponding TIMERIRr bit. The TIMERIRr bits are reset to 0 automatically after resetting.

Register TIMERIR:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	R							
	reserved						TIMERIR <1:0>	

Bit 31 - 2 reserved (value 0 is written)

Bit 1 - 0 **TIMERIR<1:0>** indicates whether an interrupt has been triggered by one of the timers. Each bit corresponds to one timer (e.g. Timer 0 => TIMERIR<0>, Timer 1 => TIMERIR<1>)
0 = no Interrupt triggered
1 = Interrupt triggered

Register ICUNITIRe:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U						R/W	
	reserved						ICUNITIRe<1:0>	

Bit 31 - 2 reserved (value 0 is written)

Bit 1 - 0 **ICUNITIRe<1:0>** (default = 0) here you can enable the interrupt functions of the IC Units. Each bit corresponds to one IC Unit (e.g. IC-Unit 0 => ICUNITIRe<0>, IC-Unit 1 => ICUNITIRe<1>)
0 = Interrupt disabeld (default)
1 = Interrupt enabled

Register ICUNITIRr:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	U						R/W	
	reserved						ICUNITIRr<7:0>	

Bit 31 - 2 reserved (value 0 is written)

Bit 1 - 0 **ICUNITIRr<1:0>** Each bit corresponds to one IC Unit (e.g. IC-Unit 0 => ICUNITIRr<0>, IC-Unit 1 => ICUNITIRr<1>). If an IC Unit has triggered an interrupt its signal bit has to be reset in the ICUNITIR register. This is done by setting (= 1) the corresponding ICUNITIRr bit. The ICUNITIRr bits are reset to 0 automatically after resetting.

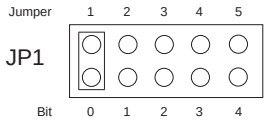
Register ICUNITIR:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U							
	reserved							
23:16	U							
	reserved							
15:8	U							
	reserved							
7:0	R							
	reserved						ICUNITIR <1:0>	

Bit 31 - 2 reserved (value 0 is written)

Bit 1 - 0 **ICUNITIR<1:0>** indicates whether an interrupt has been triggered by one of the IC Units. Each bit corresponds to one IC Unit (e.g. IC-Unit 0 => ICUNITIR<0>, IC-Unit 1 => ICUNITIR<1>).
0 = no Interrupt triggered
1 = Interrupt triggered

13. Board Identification



The board identification is used to differentiate between several PC cards of the same type on the computer. This is done by a jumper block, which can be read by software.

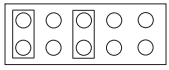
The board identification to be read consists of one Byte (8 Bit) and is structured as follows:

Bit	7	6	5	4	3	2	1	0
Jumper				5	4	3	2	1
Board ID Register	0	0	0	x	x	x	x	x

„x“ is „1“, if the jumper is set, otherwise „0“

The jumper setting of the jumper block JP1 can be read out by means of the read command. The unused bits are basically „0“, a set jumper is read as „1“.

E.g.

JP1  (Jumper 1 and 3 set)

Result of the read command: $05

13.1 Port Addresses

Offset Address	Register Name	Bit Range	Bits																	
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0xOFF8	BOARDID	31:16	reserviert (*)															
		15:0	reserviert (*)													Board ID		

(*) reserved area has to be assigned with 0

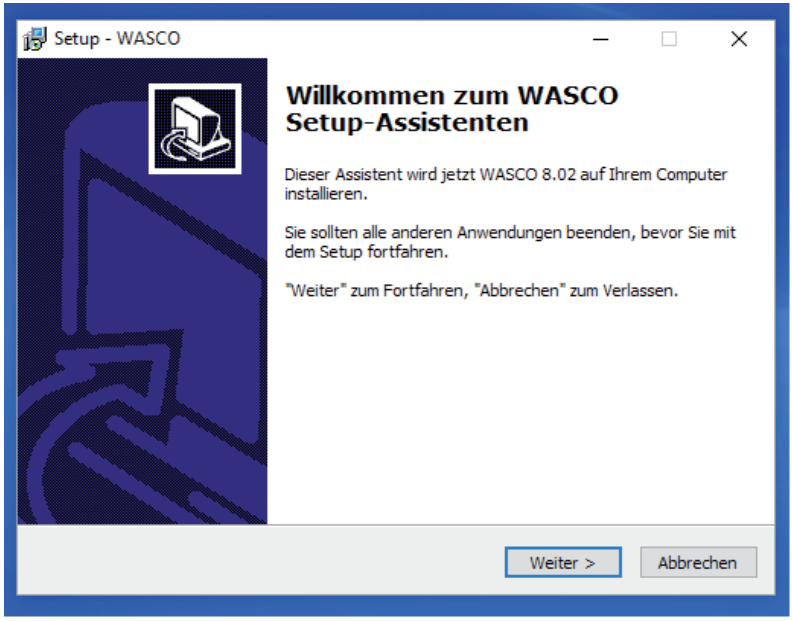
14. Programmierung unter Windows®

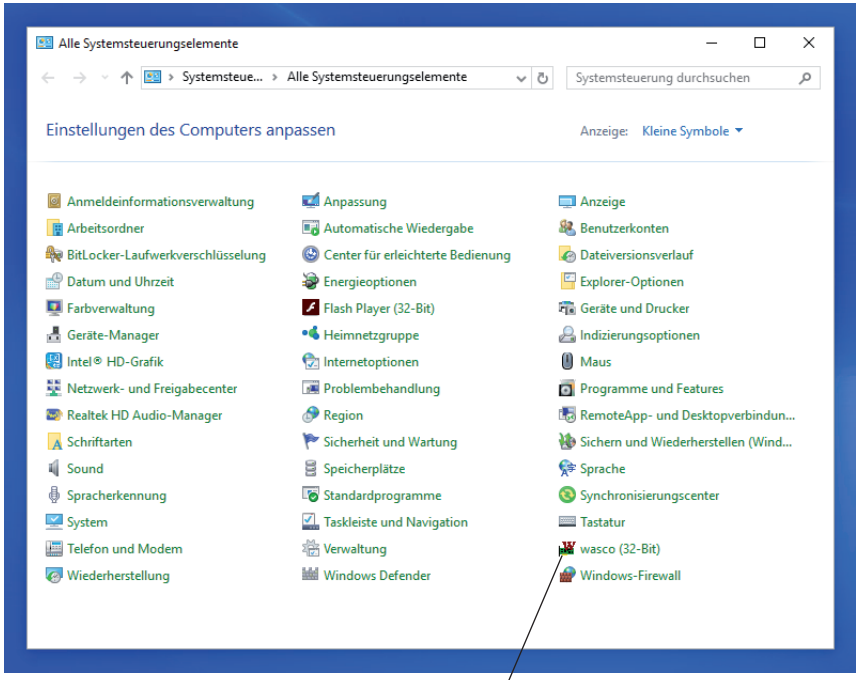
14.1 Installation of the Windows® driver

In order to implement the card under Windows®, it is necessary to install a special driver, which allows access to the card. The operating system under Windows® 10, 8 and 7 automatically reports after starting the PC, that a new hardware component has been found. In this case, insert the data medium and instruct to the system to install the driver files herefrom. If the operating system does not respond, the driver also can be installed in the Device Manager.

14.2 Installation of the Windows® development files

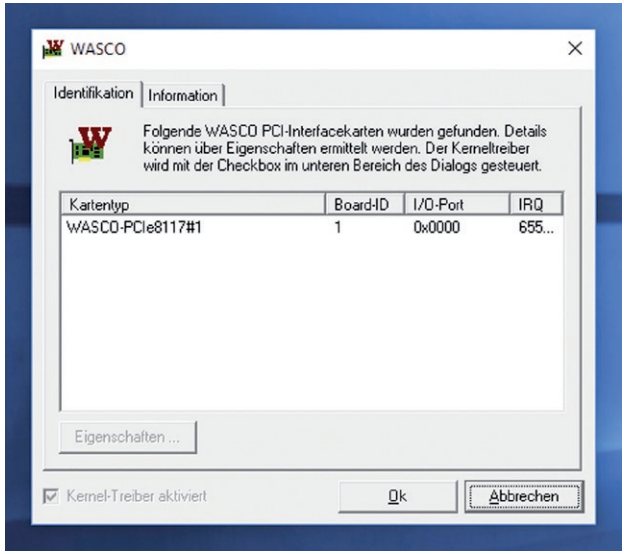
For installation of the development files, please run the file "Setup.exe" in the folder driver on the accompanying CD and follow the installation instructions.





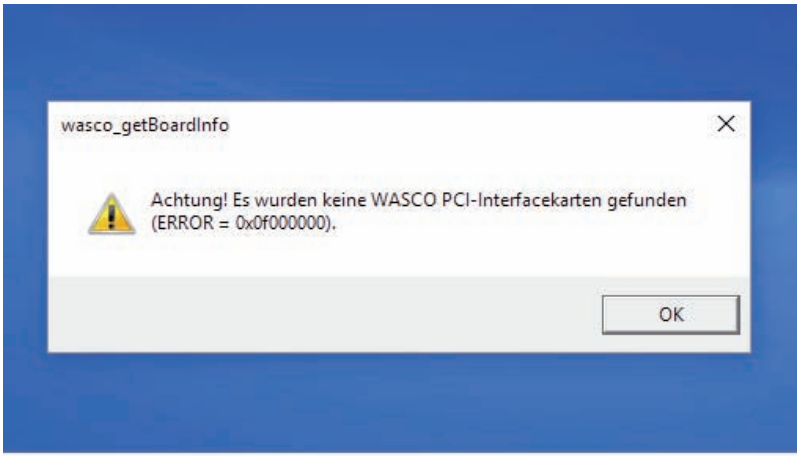
Once the driver and development files have been installed completely, you will find an icon in the control panel of your computer to localize all **wasco**® PCI and PCIe cards available in the system.

Start the card query by double-clicking the "wasco®"- Icon. Following window appears: (A WASCO-PCIe8117 is used as an example)



If your card has been detected in the system, the board name WASCO-PCIe8296, Board-ID, I/O address as well as the possible interrupt number of the respective card are displayed in this window. Furthermore, the driver version and the location of the driver file can be queried via the „Information“ tab.

If your card was not detected, following error message will be displayed:



Please find more about the possible causes in the chapter Troubleshooting.

14.3 Programming the WITIO-PCIe192 with wasco® drivers

After installing development files of Kithara by means of the setup program the folder **wasco** contains of the relevant development files and the sample programs. Further sample programs specified for access to the WITIO-PCIe192 you can find on the enclosed CD or please visit our homepage.

Programming the hardware components of the WITIO-PCIe192 is realized by access to Memory Mapped I/O addresses which depend on the base address assigned by the system‘s BIOS for the WITIO-PCIe192. Find a more detailed description for programming in the driver documentation.

14.4 Access to the WITIO-PCIe192ULTRA

The acess to the WITIO-PCIe192ULTRA is done exclusively via the board name (card type) WASCO-PCIe8296

14.5 Assignment of the Memory Mapped I/O Addresses

The Memory Mapped I/O addresses of the single hardware components depend on the base address, as shown in following table using a few examples:

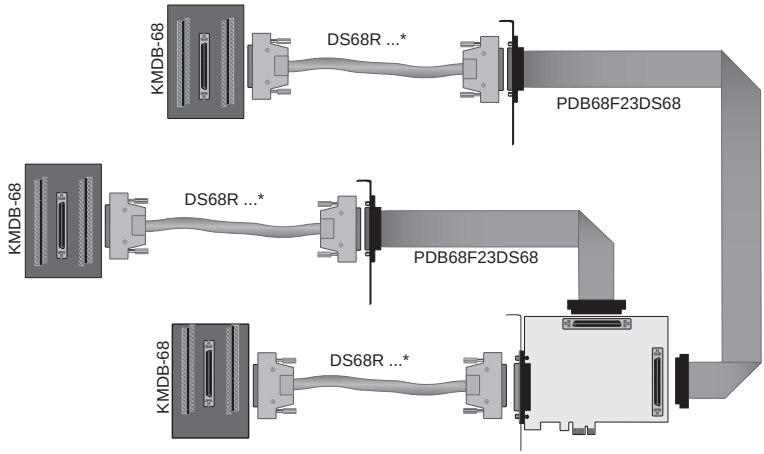
Port/Register	BA + Offset	RD/WR
read first 32 inputs of CN1 (PD7..PD0, PC7..PC0, PB7..PB0, PA7..PA0)	BA + $140	RD
read second 32 inputs of CN1 (PH7..PH0, PG7..PG0, PF7.F0, PE7..PE0)	BA + $144	RD
read/write first 32 outputs of CN1 (PD7..PD0, PC7..PC0, PB7..PB0, PA7..PA0)	BA + $160	RD/WR
read/write second 32 outputs of CN1 (PH7..PH0, PG7..PG0, PF7.F0, PE7..PE0)	BA + $164	RD/WR
Board Identification	BA+ $3E0	RD

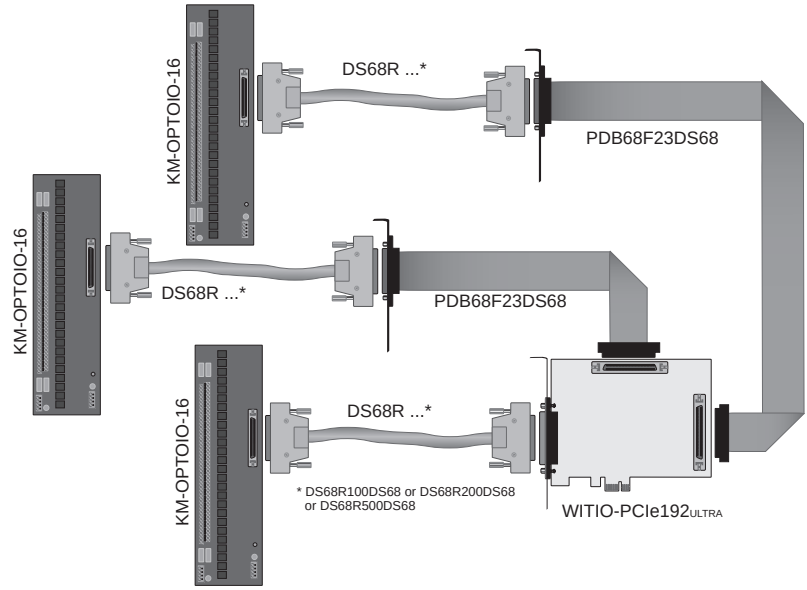
15. Accessories

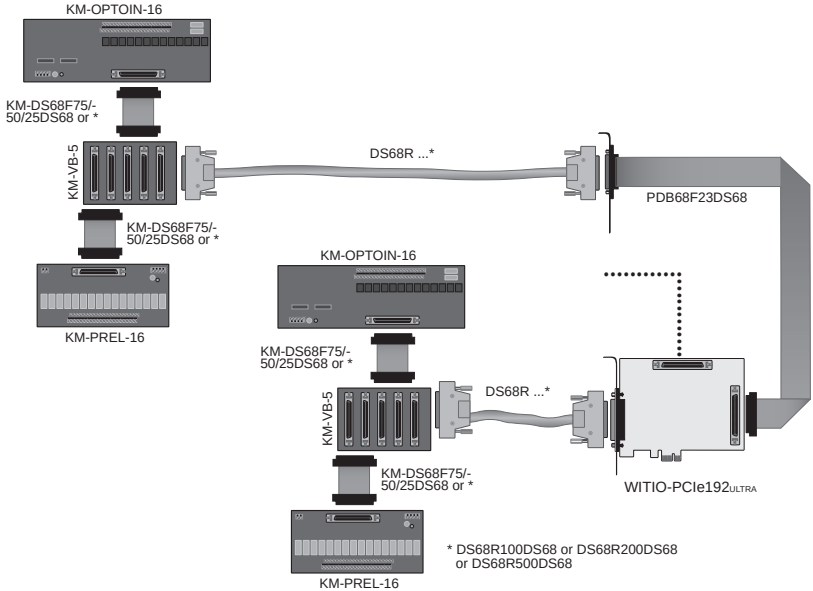
15.1 Compatible wasco® accessories

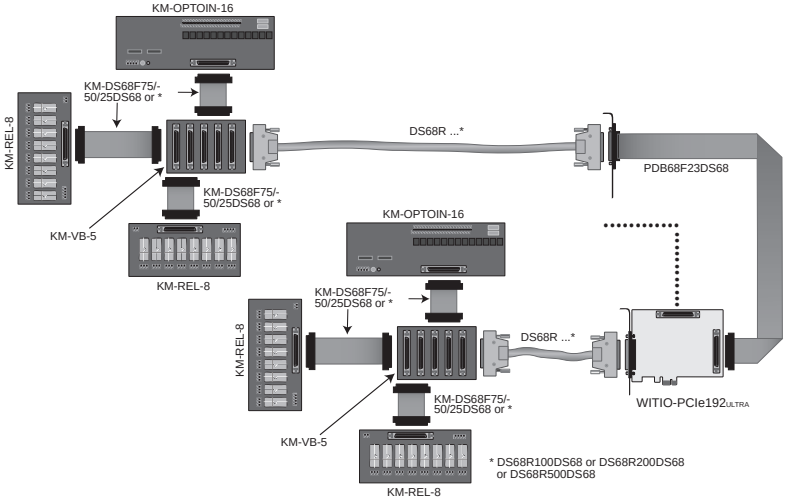
Connecting parts	EDP-No.
PDB68F23DS68 Ribbon cable	A-498500
DS68R200DS68 Connecting wire (2 meter)	A-492400
DS68R500DS68 Connecting wire (5 meters)	A-492800
KMDB-68 Connecting Board	A-494800
KM-OPTOIN-32 Optocoupler Module	A-483600
KM-OPTOOUT-32 Optocoupler Module	A-484600
KM-PREL-16 Relais Module	A-485400
KM-REL-8 Relais Module	A-486200
KM-VB-5 Connection Module	A-488200

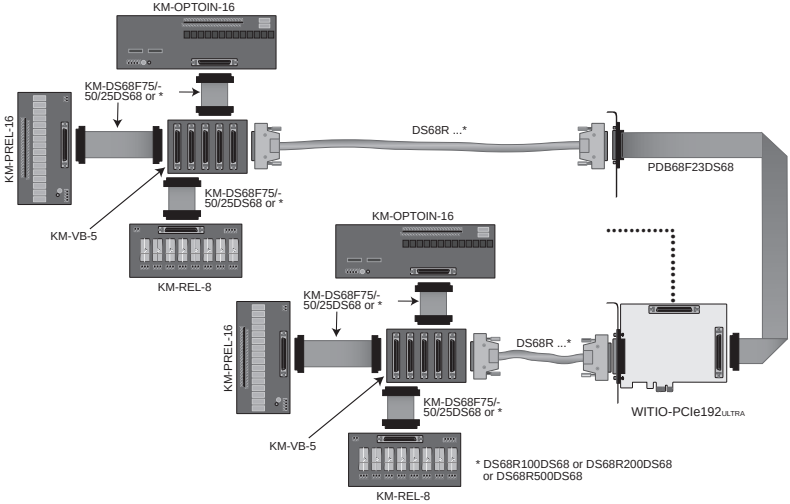
15.2 Connection Technique (application examples)

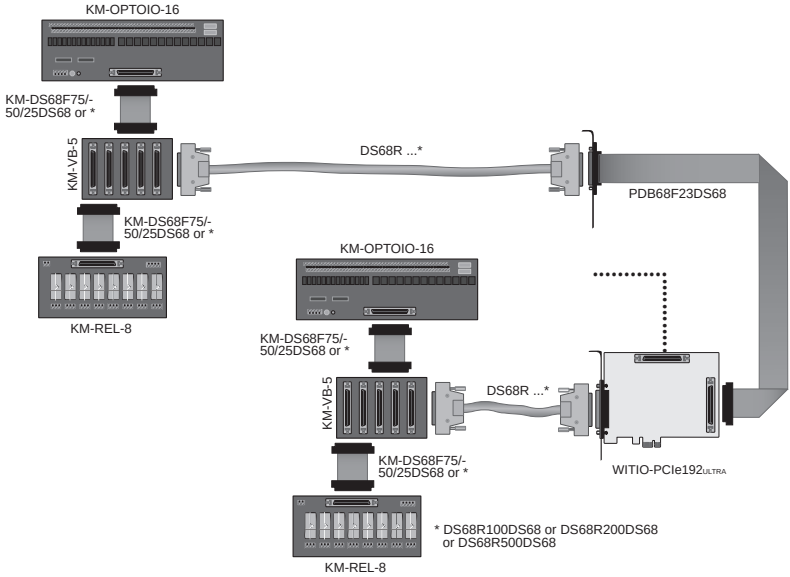


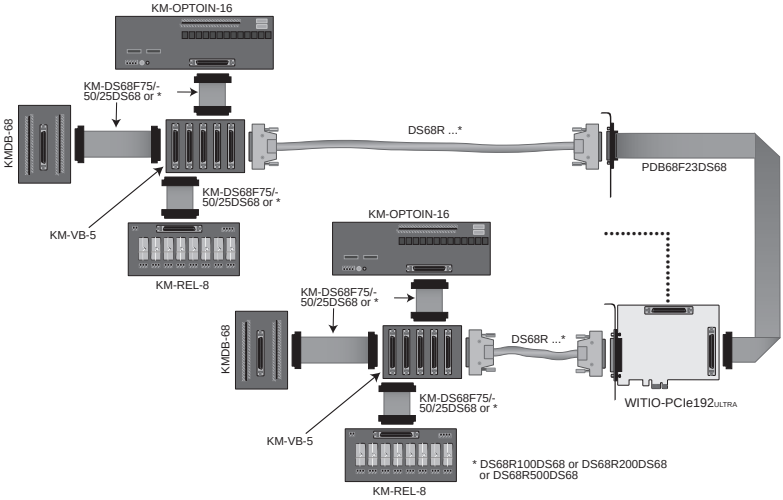

KM-OPTOIO-16
DS68R ...*
PDB68F23DS68
KM-OPTOIO-16
DS68R ...*
PDB68F23DS68
KM-OPTOIO-16
DS68R ...*
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68
WITIO-PCIe192ULTRA


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-PCIe192ULTRA
KM-PREL-16
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68


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 *
KM-VB-5
WITIO-PCIe192ULTRA
KM-REL-8
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68


KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KM-PREL-16
DS68R ...*
PDB68F23DS68
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
KM-REL-8
KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KM-PREL-16
DS68R ...*
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
WITIO-PCIe192ULTRA
KM-REL-8
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68


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-PCIe192ULTRA
KM-REL-8
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68


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-PCIe192ULTRA
KM-VB-5
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68
KM-REL-8

15.3 Single components for self-assembly

Connection parts	EDP-No.
SCSI-II Connector 68-pin for flat ribbon cable	A-553200
SCSI-II Socket 68-pin for flat ribbon cable	A-557200
SCSI-II Connector 68-pin for Solder Connection	A-555340
Slot bracket with cutout for 68-pin connector male/female	A-577800
Flat ribbon cable 68-pin	A-572800

16. Troubleshooting

Following you can find a brief compilation of the most common known causes of errors that may occur during starting-up or while running the WITIO-PCIe192.
Please check these points before you contact your dealer or distributor to solve your problem:

1st	Is WITIO-PCIe192 properly inserted to the connector ?
2nd	Are all cable connections all right?
3rd	Did your system detect the card correctly? Please check all settings of your computer or contact your system administrator. (As this are BIOS settings of the computer we cannot expand on this issue. We refer to your system manual.)
4th	Did you install the latest driver version for the **wasco**® drivers? Updates you can find here: http://www.messcomp.com

17. Specifications

Inputs / Outputs
Channels: 192
Output level 3.3V/5V, adjustable by jumper
All of the inputs with programmable input filter, edge interrupt and change interrupt
Output current: 5 mA per channel
Programmable in 8-Bit-groups as inputs or outputs

Output-Compare-Unit
8 OC Units connectable to the first eight output channels of each connector
Resolution 32Bit [1µs]
PWM generation
Generation of discrete pulses

Input-Capture-Unit
2 IC Units connectable at all of the inputs
Resolution 32Bit [1µs]
Period and pulse duration measurements

Counter
32 counters connectable at all of the inputs
Resolution 32 Bit

Timer
2 Timers
Resolution 32Bit [1µs]

Quartz crystal oscillator
4 MHz

Board identification
Jumper block with five pairs of contact pins

Connection plug
3 * 68pin SCSI socket

Bus system
32 Bit PCIe-Bus
(Internal Data access 32Bit)

Dimensions of the board
137 mm x 111 mm (l x h)

Others
Protection and Control LEDs for power supply

18. Product Liability Act

Information on 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 may already exist, 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 comply with the required safety instructions when 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 customers of the following safety instructions:

Safety Instructions

The valid VDE regulations must be observed, when handling products that come into contact with electrical voltage.

Particular attention must be paid to the regulations:
VDE100; VDE0550/0551; VDE0700; VDE0711; VDE0860.

You recieve the regulations at:
Vde-Verlag GmbH
Bismarckstr. 33
10625 Berlin
Germany

* unplug the mains plug before you open the device or make sure, there is no current to/in the unit.

* You only may start up any components, boards or equipment, if they have been installed in a touch-proof casing before. During installation, the the equipment must be de-energized.

* Make sure that the device is disconnected from the supply voltage before using any tools on any devices, components or assemblies. Any electric charges stored in components in the device are to be discharged prior.

* Live cables or wires, which are connected with the device, the components or the assemblies, must be inspected for insulation faults or breaks. In case of any defect the device must be taken out of service immediately, until the defective lines have been replaced.

* When using components or circuit boards you must strictly comply with the characteristic specifications for electrical parameters specified in the relevant description.

* As a non-commercial end-user, if it is not clear whether or not the electrical characteristic specifications given in the provided description apply to a component, you must consult a specialist.

Furthermore, compliance with construction and safety regulations of all kinds (VDE, TÜV, industrial injuries corporation, etc.) is subject to the user/customer.

19. Declaration of Conformity

This is to certify, that the CE marked product

WITIO-PCIe192ULTRA
EDP Number A-864810

comply with the requirements of the relevant EMC directives 2014/30/EU. 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.

The following standards were considered:

EN 55011: 2009 + A1. 2010 (Group 1, Class A)
EN 55022: 2010 / AC: 2011
EN 55024: 2010
EN 61000-6-4: 2007 + A1: 2011
EN 61000-6-2: 2005 / AC: 2005
(EN 6100-4-2: 2008; EN 6100-4-3: 2006 + A1: 2007 + A2; EN 6100-4-4: 2012; EN 6100-4-5: 2014; EN 6100-4-6: 2013; EN 6100-4-8: 2009; EN 6100-4-11: 2004)

The following manufacturer is responsible for this declaration:

Messcomp Datentechnik GmbH
Neudecker Str. 11
83512 Wasserburg

submitted by

Dipl.Ing.(FH) Hans Schnellhammer

Wasserburg, 19.07.2018

Reference system for intended use

This PC expansion card is not a stand-alone device. The CE-conformity only can be assessed when using additional computer components simultaneously. Thus the information to the CE conformity exclusively refers to the following reference system for the intended use of the PC expansion card:

Control Cabinet:	Vero IMRAK 3400	804-530061C 802-563424J 802-561589J
19" Casing:	Vero PC Casing	145-010108L
19" Casing:	Additional Electronics	519-112111C
Motherboard:	ASUS P5G41-M LE	
Interface:	WITIO-PCIe192ULTRA	A-864810