

WITIO-PCle192 ULTRA

EDV-Nr.: A-864810

192 interruptfähige Ein- bzw. Ausgänge
digitale Eingangsfiler
32 Zähler 32Bit
2 Timer 32Bit
3 OC-Einheiten
2 IC-Einheiten
interruptfähig
Board-Identifikation

wasco[®]

Handbuch

Copyright® 2018 by Messcomp Datentechnik GmbH

Diese Dokumentation ist urheberrechtlich geschützt. Alle Rechte sind vorbehalten.

Messcomp Datentechnik GmbH behält sich das Recht vor, die in dieser Dokumentation beschriebenen Produkte jederzeit und ohne Vorankündigung zu verändern.

Ohne schriftliche Genehmigung der Firma Messcomp Datentechnik GmbH darf diese Dokumentation in keinerlei Form vervielfältigt werden.

Geschützte Warenzeichen

Windows®, Visual Basic®, Visual C++®, Visual C#® sind eingetragene Warenzeichen von Microsoft.

wasco® ist ein eingetragenes Warenzeichen.

Linux® ist ein eingetragenes Warenzeichen.

Ubuntu® ist ein eingetragenes Warenzeichen.

LabVIEW® ist ein eingetragenes Warenzeichen.

Bei anderen genannten Produkt- und Firmennamen kann es sich um Warenzeichen der jeweiligen Inhaber handeln.

Haftungsbeschränkung

Die Firma Messcomp Datentechnik GmbH haftet für keinerlei durch den Gebrauch der Interfacekarte WITIO-PCIe192ULTRA und dieser Dokumentation direkt oder indirekt entstandene Schäden.

Inhaltsverzeichnis

1. Produktbeschreibung	5
2. Installation der WITIO-PCIe192_{ULTRA}	6
2.1 Installation der Karte in den Rechner	6
3. Anschlussstecker	7
3.1 Lage der Anschlussstecker auf der Platine	7
3.2 Steckerbelegung von CN1.....	8
3.3 Steckerbelegung von CN2.....	9
3.4 Steckerbelegung von CN3.....	10
4. Systemkomponenten	11
4.1 Blockschaltbild.....	11
4.2 Zugriff auf die Systemkomponenten.....	12
5. 192 Digitale Ein-/Ausgänge	13
5.1 Richtungsauswahl	13
5.2 Eingänge lesen.....	13
5.3 Ausgänge schreiben/lesen	14
5.4 Port deaktivieren.....	14
5.5 PegelEinstellung	15
5.6 IOs im Kompatibilitätsmodus	15
5.7 Portadressen	16
6. Erweiterte Funktionen digitaler Eingänge.....	21
6.1 Digitale Eingänge mit digitalem Filter	21
6.2 Interruptfunktionen der digitalen Eingänge.....	22
6.3 Portadressen	25
6.3 Portadressen	25
7. Erweiterte Funktionen digitaler Ausgänge	27
7.1 Grundfunktion	27
7.2 Digitale Ausgänge mit anderen Hardwarekomponenten belegen	27
7.3 Portadressen	28
8. Zähler.....	31
8.1 Grundfunktion	31
8.2 Interruptfunktion.....	31
8.3 Portadressen	32

9. Timer	42
9.1 Anwendung Intervall-Interruptauslösung	42
9.2 Portadressen	43
10. Input-Capture-Einheit	46
10.1 Kontinuierliche Messung periodischer Signale	46
10.2 Portadressen	47
11. Output-Compare-Einheit	51
11.1 PWM	51
11.2 Pulsausgabe	53
11.3 Portadressen	55
12. Interruptcontroller	62
12.1 Portadressen	64
13. Board-Identifikation	80
13.1 Portadressen	81
14. Programmierung unter Windows [®]	82
14.1 Installation des Windows [®] Treibers	82
14.2 Installation der Windows [®] Entwicklungsdateien	82
14.3 Programmierung der WITIO-PCle192 mit wasco [®] -Treiber	85
14.4 Zugriff auf die Karte WITIO-PCle192 ^{ULTRA}	86
14.5 Zuordnung der Memory Mapped I/O-Adressen.....	86
15. Programmierung unter Linux [®]	87
15.1 Installation des Linux [®] Treibers	87
15.2 Unterstützte Linux-Distributionen/Kernelversionen	87
15.3 Programmierung der WITIO-PCle192 mit wasco [®] -Treiber	87
15.4 Zugriff auf die Karte WITIO-PCle192	88
15.5 Zuordnung der Memory Mapped I/O-Adressen.....	88
16. Zubehör	89
16.1 Passendes wasco [®] -Zubehör	89
16.2 Anschlusstechnik (Anwendungsbeispiele).....	89
16.3 Einzelkomponenten zur Eigenkonfektionierung	93
17. Fehlersuche	94
18. Technische Daten	95
19. Produkthaftungsgesetz	96
20. EG-Konformitätserklärung	98

1. Produktbeschreibung

Die WITIO-PCle192_{ULTRA} (Boardname: WASCO-PCle8296) bietet 192 digitale Ein- bzw. Ausgangskanäle, welche mittels Jumper auf 3,3V oder 5V Ein- bzw. Ausgangspegel einstellbar sind. Die 192 Ein-/Ausgangskanäle sind in Gruppen zu je acht Kanälen als Ein- oder Ausgänge programmierbar. Jeder Eingangskanal ist interruptfähig und alle 64 Eingänge des Connectors CN1 besitzen einen programmierbaren digitalen Eingangsfiler (0 bis 255µs). Zudem kann an jeden Eingangskanal einer von zwei IC-Einheiten (z.B. Perioden- und Pulsdauermessung) per Software gekoppelt werden. Für Anwendungen, in welchen z.B. eine PWM oder eine einfache, hochauflösende Pulsgenerierung vonnöten ist, werden 3 OC-Einheiten zur Verfügung gestellt. Diese lassen sich an die jeweils ersten 8 Ausgangskanäle eines jeden Connectors koppeln. Außerdem besitzt die WITIO-PCle192_{ULTRA} zwei programmierbare, interruptfähige Timer mit Quarzzeitbasis.

Diese Karte eignet sich für Ein- und Ausgabeanwendungen, für die keine galvanische Trennungen notwendig sind. Der interne Datenbus dieser Karte ist 32Bit organisiert, jeder Lese- bzw. Schreibzugriff auf die Ein- und Ausgänge erfolgt als 32Bit-Zugriff. Eine 68polige SCSI-Buchse am Slotblech der Platine und zwei 68polige SCSI-Buchsen mit je 64 Kanälen ermöglichen den Anschluss der Peripherie. Die Steckerbelegungen aller Steckverbinder der WITIO-PCle192_{ULTRA} sind identisch zu den Belegungen der PCI-Bus-Karten WITIO-PCI32_{STANDARD} und WITIO-PCI64_{EXTENDED}, ein Umstieg auf PCIe ist dadurch einfach realisierbar. Des Weiteren besitzt die Karte einen Board-Identifikations-Jumperblock, um mehrere identische Karten im PC unterscheiden zu können.

2. Installation der WITIO-PCIe192_{ULTRA}

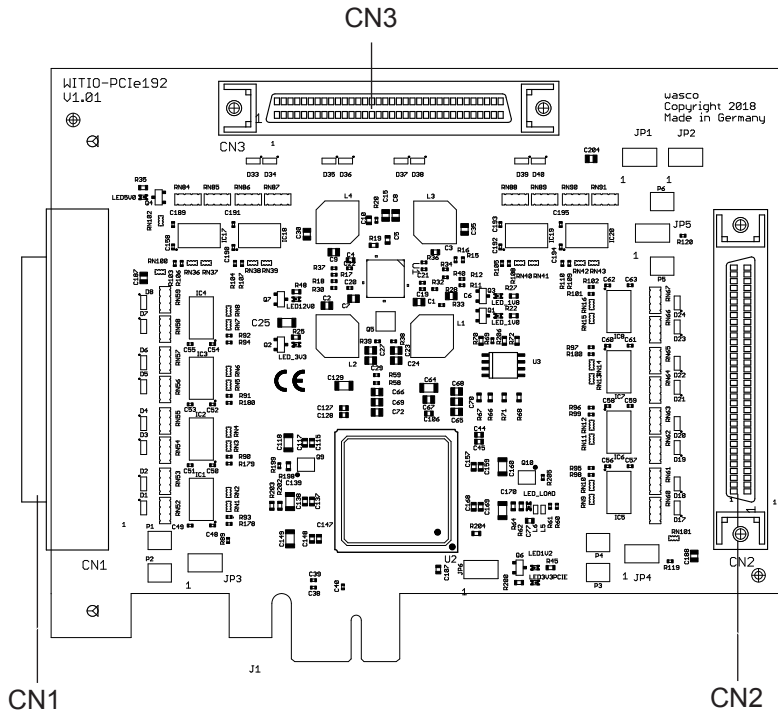
2.1 Installation der Karte in den Rechner

Achten Sie vor dem Einbau der WITIO-PCIe192 darauf, dass der Rechner vom Netz getrennt oder zumindest ausgeschaltet ist. Beim Einbau der Interface-Karte in den laufenden Rechner besteht die Gefahr, dass neben der WITIO-PCIe192 auch andere Karten des PCs oder Rechners beschädigt oder zerstört werden.

Wählen Sie in Ihrem Rechner einen freien PCIe-Steckplatz, in den Sie dann die Karte einsetzen. Nehmen Sie dazu auch das Benutzerhandbuch Ihres Computers zu Hilfe. Verschrauben Sie das Slotblech der Platine mit dem Rechnergehäuse, damit sich die Karte nicht während des Betriebs unter Einwirkung der Anschlusskabel aus dem Steckplatz lösen kann.

3. Anschlussstecker

3.1 Lage der Anschlussstecker auf der Platine



- CN1: 64 digitale I/Os
- CN2: 64 digitale I/Os
- CN3: 64 digitale I/Os

3.2 Steckerbelegung von CN1

CN1 GND	68	□	□	34	CN1 VCC
CN1 GND	67	□	□	33	CN1 VCC
CN1 PH7	66	□	□	32	CN1 PH6
CN1 PH5	65	□	□	31	CN1 PH4
CN1 PH3	64	□	□	30	CN1 PH2
CN1 PH1	63	□	□	29	CN1 PH0
CN1 PG7	62	□	□	28	CN1 PG6
CN1 PG5	61	□	□	27	CN1 PG4
CN1 PG3	60	□	□	26	CN1 PG2
CN1 PG1	59	□	□	25	CN1 PG0
CN1 PF7	58	□	□	24	CN1 PF6
CN1 PF5	57	□	□	23	CN1 PF4
CN1 PF3	56	□	□	22	CN1 PF2
CN1 PF1	55	□	□	21	CN1 PF0
CN1 PE7	54	□	□	20	CN1 PE6
CN1 PE5	53	□	□	19	CN1 PE4
CN1 PE3	52	□	□	18	CN1 PE2
CN1 PE1	51	□	□	17	CN1 PE0
CN1 PD7	50	□	□	16	CN1 PD6
CN1 PD5	49	□	□	15	CN1 PD4
CN1 PD3	48	□	□	14	CN1 PD2
CN1 PD1	47	□	□	13	CN1 PD0
CN1 PC7	46	□	□	12	CN1 PC6
CN1 PC5	45	□	□	11	CN1 PC4
CN1 PC3	44	□	□	10	CN1 PC2
CN1 PC1	43	□	□	9	CN1 PC0
CN1 PB7	42	□	□	8	CN1 PB6
CN1 PB5	41	□	□	7	CN1 PB4
CN1 PB3	40	□	□	6	CN1 PB2
CN1 PB1	39	□	□	5	CN1 PB0
CN1 PA7	38	□	□	4	CN1 PA6
CN1 PA5	37	□	□	3	CN1 PA4
CN1 PA3	36	□	□	2	CN1 PA2
CN1 PA1	35	□	□	1	CN1 PA0

Vcc:

Interne Versorgungsspannung (+ 5V / +3,3V) der PCIe-Karte (konfigurierbar durch JP3). Hier niemals eine externe Spannung anlegen.

GND:

Masse des Rechners

3.3 Steckerbelegung von CN2

CN2 GND	68	□	□	34	CN2 VCC
CN2 GND	67	□	□	33	CN2 VCC
CN2 PH7	66	□	□	32	CN2 PH6
CN2 PH5	65	□	□	31	CN2 PH4
CN2 PH3	64	□	□	30	CN2 PH2
CN2 PH1	63	□	□	29	CN2 PH0
CN2 PG7	62	□	□	28	CN2 PG6
CN2 PG5	61	□	□	27	CN2 PG4
CN2 PG3	60	□	□	26	CN2 PG2
CN2 PG1	59	□	□	25	CN2 PG0
CN2 PF7	58	□	□	24	CN2 PF6
CN2 PF5	57	□	□	23	CN2 PF4
CN2 PF3	56	□	□	22	CN2 PF2
CN2 PF1	55	□	□	21	CN2 PF0
CN2 PE7	54	□	□	20	CN2 PE6
CN2 PE5	53	□	□	19	CN2 PE4
CN2 PE3	52	□	□	18	CN2 PE2
CN2 PE1	51	□	□	17	CN2 PE0
CN2 PD7	50	□	□	16	CN2 PD6
CN2 PD5	49	□	□	15	CN2 PD4
CN2 PD3	48	□	□	14	CN2 PD2
CN2 PD1	47	□	□	13	CN2 PD0
CN2 PC7	46	□	□	12	CN2 PC6
CN2 PC5	45	□	□	11	CN2 PC4
CN2 PC3	44	□	□	10	CN2 PC2
CN2 PC1	43	□	□	9	CN2 PC0
CN2 PB7	42	□	□	8	CN2 PB6
CN2 PB5	41	□	□	7	CN2 PB4
CN2 PB3	40	□	□	6	CN2 PB2
CN2 PB1	39	□	□	5	CN2 PB0
CN2 PA7	38	□	□	4	CN2 PA6
CN2 PA5	37	□	□	3	CN2 PA4
CN2 PA3	36	□	□	2	CN2 PA2
CN2 PA1	35	□	□	1	CN2 PA0

Vcc:

Interne Versorgungsspannung (+ 5V / +3,3V) der PCIe-Karte (konfigurierbar durch JP4). Hier niemals eine externe Spannung anlegen.

GND:

Masse des Rechners

3.4 Steckerbelegung von CN3

CN3 GND	88	□	□	34	CN3 VCC
CN3 GND	67	□	□	33	CN3 VCC
CN3 PH7	66	□	□	32	CN3 PH6
CN3 PH5	65	□	□	31	CN3 PH4
CN3 PH3	64	□	□	30	CN3 PH2
CN3 PH1	63	□	□	29	CN3 PH0
CN3 PG7	62	□	□	28	CN3 PG6
CN3 PG5	61	□	□	27	CN3 PG4
CN3 PG3	60	□	□	26	CN3 PG2
CN3 PG1	59	□	□	25	CN3 PG0
CN3 PF7	58	□	□	24	CN3 PF6
CN3 PF5	57	□	□	23	CN3 PF4
CN3 PF3	56	□	□	22	CN3 PF2
CN3 PF1	55	□	□	21	CN3 PF0
CN3 PE7	54	□	□	20	CN3 PE6
CN3 PE5	53	□	□	19	CN3 PE4
CN3 PE3	52	□	□	18	CN3 PE2
CN3 PE1	51	□	□	17	CN3 PE0
CN3 PD7	50	□	□	16	CN3 PD6
CN3 PD5	49	□	□	15	CN3 PD4
CN3 PD3	48	□	□	14	CN3 PD2
CN3 PD1	47	□	□	13	CN3 PD0
CN3 PC7	46	□	□	12	CN3 PC6
CN3 PC5	45	□	□	11	CN3 PC4
CN3 PC3	44	□	□	10	CN3 PC2
CN3 PC1	43	□	□	9	CN3 PC0
CN3 PB7	42	□	□	8	CN3 PB6
CN3 PB5	41	□	□	7	CN3 PB4
CN3 PB3	40	□	□	6	CN3 PB2
CN3 PB1	39	□	□	5	CN3 PB0
CN3 PA7	38	□	□	4	CN3 PA6
CN3 PA5	37	□	□	3	CN3 PA4
CN3 PA3	36	□	□	2	CN3 PA2
CN3 PA1	35	□	□	1	CN3 PA0

Vcc:

Interne Versorgungsspannung (+ 5V / +3,3V) der PCIe-Karte (konfigurierbar durch JP5). Hier niemals eine externe Spannung anlegen.

GND:

Masse des Rechners

4. Systemkomponenten

4.1 Blockschaftbild

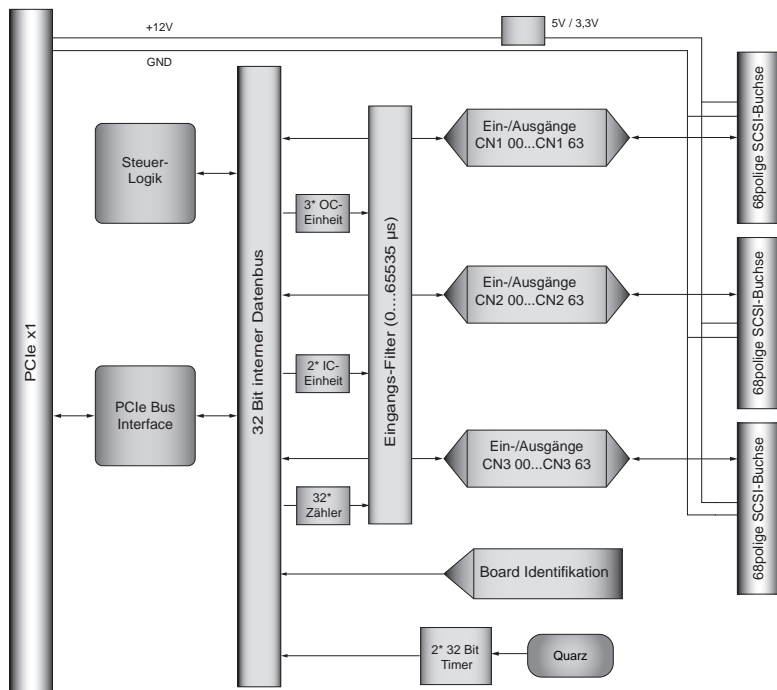


Abb. 4.1

4.2 Zugriff auf die Systemkomponenten

Der Zugriff auf die Hardware-Komponenten der WITIO-PCIe192 erfolgt durch das Lesen von bzw. Schreiben in Memory-Mapped I/O-Adressen mit Hilfe von Library-Funktionen. Die für die WITIO-PCIe192 relevanten Adressen ergeben sich abhängig von einer vom BIOS vergebenen Basisadresse. Der Zugriff auf die WITIO-PCIe192 erfolgt ausschließlich im Doppel-Word-Zugriff, wobei die WASCO-Treiber-Funktionen aus Kompatibilitätsgründen nur das niederwertigste Byte verarbeiten bzw. berücksichtigen. (Hinweise hierzu finden Sie im Kapitel Programmierung sowie in den Beispielpogrammen auf der mitgelieferten CD).

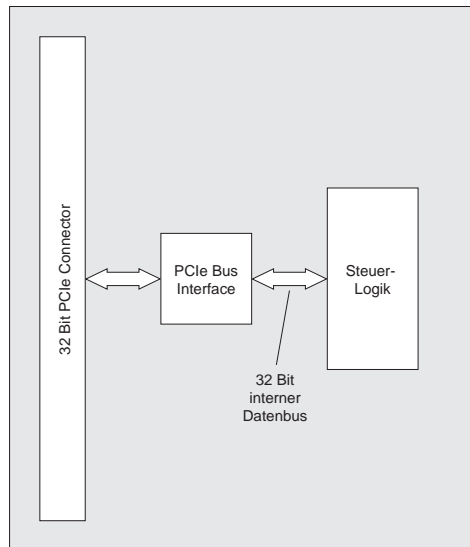


Abb. 4.2

5. 192 Digitale Ein-/Ausgänge

Zur digitalen Ein-/Ausgabe sind auf der WITIO-PCIe192_{ULTRA} drei identisch aufgebaute Connectoren mit jeweils 64 digitalen IOs. Diese 64 digitalen IOs sind in Gruppen (PortA bis PortH) mit jeweils 8 Kanälen zusammengefasst. Für jeden Port kann individuell dessen Richtung (Ein-/Ausgang) über einen Registerzugriff konfiguriert werden. Der Ein- bzw. Ausgangspegel (3.3V/5V) eines jeden Connectors kann durch Setzen eines Jumpers eingestellt werden.

5.1 Richtungsauswahl

Die Richtungen der Ports eines Connectors können durch Beschreiben des jeweiligen Direction-Registers (DDIRCN1, DDIRCN2 oder DDIRCN3) definiert werden. Dabei steht jedes Bit des Registers für einen Port des Connectors. Im Default-Zustand sind alle Ports als Eingänge definiert (mit Ausnahme während des Kompatibilitätsmodus).

Wird das entsprechende Bit mit einer 1 beschrieben, so ist der entsprechende Port als Eingangsport konfiguriert, mit einer 0 als Ausgangsport.

5.2 Eingänge lesen

Zum Einlesen der Eingänge werden immer 32 IOs (4 Ports) in einem 32Bit-Register (DIN0CN1, DIN1CN1, DIN0CN2, DIN1CN2, DIN0CN3, DIN1CN3) zusammengefasst. Sind nicht alle Ports des Registers als Eingang konfiguriert, sind die jeweiligen Bereiche im Register undefiniert und können im Anwenderprogramm durch eine AND-Verknüpfung ausgeblendet werden.

0 = LOW am Eingangspin

1 = HIGH am Eingangspin

5.3 Ausgänge schreiben/lesen

Das Setzen von Ausgängen erfolgt über das Beschreiben der DOUTyCNx⁽¹⁾ Register. Jeder Connector hat zwei dieser Register (DOUT0CNx und DOUT1CNx), in welchen jeweils 32 IO-Pins (4 Ports) zusammengefasst werden. Sind nicht alle Ports des Registers als Ausgänge konfiguriert, so werden nur die Ausgangsbereiche im Register von der Karte berücksichtigt.

0 = LOW am Ausgangspin (wenn als Ausgang definiert)

1 = HIGH am Ausgangspin (wenn als Ausgang definiert)

5.4 Port deaktivieren

Um den Stromverbrauch der Karte zu minimieren und eventuell mögliche Störungen auf dem Stecker zu verhindern, können nicht verwendete Ports und deren IOs deaktiviert werden. Ist ein Port deaktiviert, so sind alle IOs hochohmig.

Ports können immer in Zweiergruppen, also 16-Kanal-weise, deaktiviert werden. Dafür muss das jeweilige Bit des DENCNx-Registers gesetzt werden.

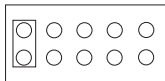
0 = Port-Gruppe aktiviert

1 = Port-Gruppe deaktiviert

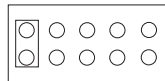
¹(y = Registernummer, x = Connector-Nummer)

5.5 Pegeleinstellung

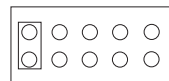
Für jeden Connector (IOs und VCC-PINs) kann der Spannungspegel mit +3.3V und +5V konfiguriert werden. Dies erfolgt durch Setzen eines Jumpers auf dem zum Connector gehörigen Jumperblock.



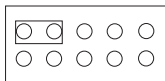
JP3 Spannungspegel
CN1 +3.3V



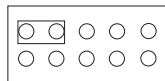
JP4 Spannungspegel
CN2 +3.3V



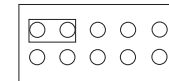
JP5 Spannungspegel
CN3 +3.3V



JP3 Spannungspegel
CN1 +5V



JP4 Spannungspegel
CN2 +5V

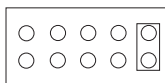


JP5 Spannungspegel
CN3 +5V

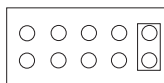
5.6 IOs im Kompatibilitätsmodus

Jeder Connector kann bei Bedarf im Kompatibilitätsmodus betrieben werden. In diesem Modus entspricht die Pinbelegung des Connectors der der Karten WITIO-PCI32_{Standard} und WITIO-PCI64_{Extended}. Dabei sind die ersten 32 IO-Pins als Eingänge konfiguriert und die zweiten 32 IO-Pins als Ausgänge.

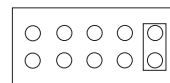
Um den Kompatibilitätsmodus eines Connectors zu verwenden, muss wie folgend dargestellt der Jumper des jeweiligen Jumperblocks gesetzt werden. Die DIRCNx Register sind im Falle des Kompatibilitätsmodus deaktiviert.



JP3 Kompatibilitätsmodus
CN1



JP4 Kompatibilitätsmodus
CN2



JP5 Kompatibilitätsmodus
CN3

5.7 Portadressen

Offset-Adresse	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-Adresse	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	DDIRC1	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x184	DDIRC2	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x188	DDIRC3	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x190	ENC1	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x194	ENC2	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x198	ENC3	31:16	reserviert (*)															
		15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

Register DIN0CNx (x = Connector-Nummer):

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 Eingänge PD7 bis PD0 des Connectors

Bit 23 - 16 DIN0CNx <23:16> Port C Eingänge PC7 bis PC0 des Connectors

Bit 15 - 8 DIN0CNx <15:8> Port B Eingänge PB7 bis PB0 des Connectors

Bit 7 - 0 DIN0CNx <7:0> Port A Eingänge PA7 bis PA0 des Connectors

Register DIN1CNx (x = Connector-Nummer):

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 Eingänge PH7 bis PH0 des Connectors

Bit 23 - 16 DIN1CNx <23:16> Port G Eingänge PG7 bis PG0 des Connectors

Bit 15 - 8 DIN1CNx <15:8> Port F Eingänge PF7 bis PF0 des Connectors

Bit 7 - 0 DIN1CNx <7:0> Port E Eingänge PE7 bis PE0 des Connectors

Register DOUT0CNx (x = Connector-Nummer):

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 Eingänge PD7 bis PD0 des Connectors

Bit 23 - 16 DOUT0CNx <23:16> Port C Eingänge PC7 bis PC0 des Connectors

Bit 15 - 8 DOUT0CNx <15:8> Port B Eingänge PB7 bis PB0 des Connectors

Bit 7 - 0 DOUT0CNx <7:0> Port A Eingänge PA7 bis PA0 des Connectors

Register DOUT1CNx (x = Connector-Nummer):

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 Eingänge PH7 bis PH0 des Connectors

Bit 23 - 16 DOUT1CNx <23:16> Port G Eingänge PG7 bis PG0 des Connector

Bit 15 - 8 DOUT1CNx <15:8> Port F Eingänge PF7 bis PF0 des Connectors

Bit 7 - 0 DOUT1CNx <7:0> Port E Eingänge PE7 bis PE0 des Connectors

Register DDIRCNx (x = Connector-Nummer):

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	R/W							
	DDIRCnx <7:0>							

Bit 31 - 8 reserviert (mit dem Wert 0 beschreiben)

Bit 7 - 0 DDIRCNx <7:0> (default = 1)

0 = IO-PIN als Ausgang geschaltet

1 = IO-PIN als Eingang geschaltet

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 = Connector-Nummer):

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U				R/W			
	reserviert				DENCnx <3:0>			

Bit 31 - 4 reserviert (mit dem Wert 0 beschreiben)

Bit 3 - 0 DENCnx <3:0> (default = 0)

0 = Portpaar aktivieren

1 = Portpaar deaktivieren

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. Erweiterte Funktionen digitaler Eingänge

6.1 Digitale Eingänge mit digitalem Filter

Die digitalen Eingänge auf dem Connector CN1 des Boards WASCO-PCIe8296 besitzen jeweils einen eigenen konfigurierbaren digitalen Filter, um Störungen oder Einschwingungen des Eingangssignals zu filtern. Dabei überprüft der Filter wie in Abb. 6.2 gezeigt, ob ein anliegendes Signal lange genug anliegt. Ist dies nicht der Fall, so wird z.B. ein zu kurzer Puls ignoriert. Die Mindestzeit, welche eine Signaländerung am jeweiligen Eingang anliegen muss, um nicht gefiltert zu werden, wird über das Register DINFILyCN1⁽¹⁾ eingestellt. Hier kann eine Filterbreite von 0 - 255µs in 1µs-Schritten eingestellt werden. Im Default-Zustand ist der Filter deaktiviert, sprich die Filterdauer beträgt 0µs.

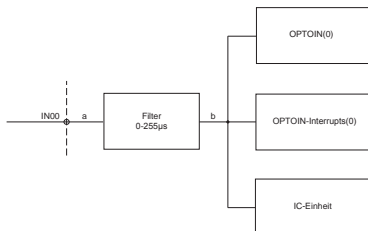


Abb. 6.1

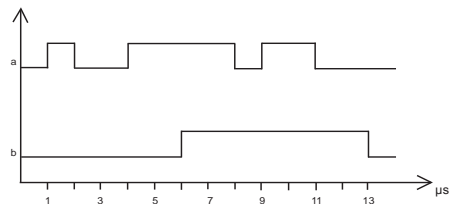


Abb. 6.2

¹(y = Registernummer)

6.2 Interruptfunktionen der digitalen Eingänge

Um Änderungen an den digitalen Eingängen ohne regelmäßiges Abfragen der Eingänge durch den PC zu erkennen, bietet die WITIO-PCIe192_{ULTRA} mehrere Interruptmöglichkeiten an. So gibt es zum einen die Möglichkeit, dass die Karte bei einer steigenden Flanke an einem der Eingänge einen Interrupt am PC auslöst, oder zum anderen, dass sie dem PC durch einen Interrupt eine generelle Änderung der Eingänge signalisiert.

Neben diesem Kapitel lesen Sie bitte das Kapitel Interruptcontroller durch.

6.2.1 Flankenerkennung

Um steigende Flanken an den digitalen Eingängen zu erkennen, besitzt jeder einzelne Eingang eine Flankenerkennung mit zuschaltbarer Interruptfunktion. Dafür werden 32Bit Interruptregister (DINIFyCNx)⁽¹⁾ bereitgestellt, welche für jeden Eingangskanal ein Bit zur Flankenerkennung zur Verfügung stellen. Sobald an einem Eingang eine steigende Flanke durch die Karte erkannt wird, wird das jeweilige Bit im Register DINIFyCNx⁽¹⁾ gesetzt. Ist mindestens eines der freigeschalteten Bits gesetzt, wird dies über eine Leitung an den Interruptcontroller weitergeleitet.

Das Freischalten der Interruptfunktion erfolgt durch das Beschreiben des jeweiligen 32Bit Registers DINFyCNx⁽¹⁾. Dabei repräsentiert jedes einzelne Bit einen Eingang. Wie aus der Tabelle Portadressen (Kapitel 12.1) zu entnehmen ist, bedeutet eine 1 im jeweiligen Bit eine Aktivierung und eine 0 eine Deaktivierung der Interruptfunktion. Steht im Bit also z.B. eine 0, so wird bei einer steigenden Flanke im Register DINIFyCNx⁽¹⁾ zwar das entsprechende Bit gesetzt, jedoch wird dieses Bit bei der Interruptauslösung nicht berücksichtigt.

Im Defaultzustand sind alle Interruptkanäle deaktiviert.

¹(y = Registernummer, x = Connector-Nummer)

Nach dem ausgelösten Interrupt muss in der entsprechenden Interrupt-Service-Routine durch das Einlesen der Register DINIFyCNx⁽¹⁾ die Quelle ermittelt und anschließend das Bit gelöscht werden, indem das zum Quellenkanal gehörige Bit im entsprechenden Register DINIFyCNx⁽¹⁾ gesetzt wird. Nachdem die Karte den Resetbefehl durchgeführt hat, wird das Bit automatisch zurückgesetzt.

Anwendungsbeispiel:

Gewünscht wird eine Flankenerkennung mit Interruptauslösung am Kanal PA1 des Connectors CN1. Im folgenden Beispiel werden die einzelnen Schritte aufgezählt, wie die Konfiguration durchgeführt werden muss und was in der Interrupt-Service-Routine zur erneuten Freigabe des Interrupts unternommen werden muss.

Bitte beachten Sie, dass in diesem Beispiel nicht die Interruptkonfiguration des Treibers aufgelistet ist. Eine Beschreibung hierzu liegt dem Treiber bei.

Neben diesem Beispiel sind zudem auf unserer Homepage Beispielprogramme zum Download bereitgestellt.

Konfiguration:

1. Freischaltung der Interruptfunktion der Karte (siehe Kapitel Interruptcontroller)
2. Freischalten des gewünschten Interrupts

Bevor der Flankenerkennungsinterrupt freigeschaltet wird, muss überprüft werden, ob das Flankenspeicherregister DINIF0CN1 vollständig zurückgesetzt ist, da sonst eventuell sofort nach der Freigabe des Interrupts ein Interrupt ausgelöst wird. Sind nicht alle Bits im Register DINIF0CN1 zurückgesetzt, schreiben Sie den Wert 0xfffffff(hex) in das Register DINIF0rCN1.

Aus der Tabelle ist zu entnehmen, dass zur Freischaltung des Flankenerkennungsinterruptes am Kanal PA1 Bit1 im Register DINIF0eCN1 gesetzt werden muss. Mit Hilfe des PCIe-Schreibbefehls wird also der Wert 0x00000002(hex) bzw. 2(dec) in dieses Register geschrieben.

¹(y = Registernummer, x = Connector-Nummer)

Interrupt-Service-Routine

1. Um die Quelle des Interrupts zu ermitteln, muss das Flankenregister DINIF0CN1 ausgelesen werden (Rückgabewert hier 0x00000002(hex)). Sollten noch andere Quellen wie Timer etc. möglich sein, muss im INTCON-Register überprüft werden, ob der vom PC empfangene Interrupt von einem der DINIFyCNx-Register stammt.
2. Ist die Quelle identifiziert, muss das Quellenbit gelöscht werden. Hierfür schreiben Sie in unserem Fall den Wert 0x00000002(hex) in das Register DINIF0rCN1.

Achtung:

Sollten in der Zeit noch weitere Interrupts geschehen sein (z.B. Timer), müssen diese in ihren jeweiligen Registern ebenfalls gelöscht werden. Erst nachdem alle aktivierten Interruptregister wieder auf 0 gesetzt wurden, kann ein weiterer Interrupt ausgelöst werden.

6.2.2 Portänderungen

Müssen die digitalen Eingänge oft abgefragt werden, um Änderungen zu erkennen, so kann eine weitere Interruptfunktion verwendet werden, um den PC zu entlasten. So bietet die WASCO-PCIe8296 die Möglichkeit an, bei einer Änderung an den Eingängen einen Interrupt auszulösen.

Zur Freischaltung dieser Interruptfunktion muss zum einen das Register DINICe auf 0x00000001 gesetzt werden. Zum anderen kann der Anwender durch die 32Bit Register DINICyCNx⁽¹⁾ bestimmen, welche Eingänge bei der Erkennung berücksichtigt werden sollen. Findet eine Änderung der Eingänge statt, so wird im Register DINICyCNx⁽¹⁾ das entsprechende Bit gesetzt. Um nach einem Auslösen des Interrupts diesen wieder freizuschalten, muss das entsprechende Bit im Register DINICyrCNx⁽¹⁾ gesetzt werden. Nach dem Reset wird das Resetbit von selbst zurückgesetzt.

¹(y = Registernummer, x = Connector-Nummer)

6.3 Portadressen

Offset-Adresse	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>															

(y = Registernummer, x = Connector-Nummer)

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)
 Der Wert bestimmt die Filterdauer des Filters y des Connectors x in µs

Offset-Adresse = 0x640 + DIN-Nummer * 4 (CN1)

¹(y = Registernummer, x = Connector-Nummer)

7. Erweiterte Funktionen digitaler Ausgänge

7.1 Grundfunktion

Die Grundfunktion der digitalen Ausgänge ermöglicht das Ausgeben von LOW- und HIGH-Signalen an einzelnen Ausgängen durch das Beschreiben von 32Bit-Registern DOUTyCNx⁽¹⁾. In den Registern steht, wie aus der Port-Adressen-Tabelle zu entnehmen ist, jedes einzelne Bit für einen digitalen Ausgang.

Möchte man zum Beispiel jeden dritten Ausgang der ersten zwei Ports des Steckers CN1 auf 3.3V bzw. 5V setzen, so muss der Wert 0x00004444(hex), 17476(dec) bzw. 0b0100010001000100(bin) in das Register DOUT0CN1 geschrieben werden.

7.2 Digitale Ausgänge mit anderen Hardwarekomponenten belegen

Neben der Grundfunktion, welche einen einfachen Zugriff auf die digitalen Ausgänge ermöglicht, können auch unterschiedliche Hardwarekomponenten wie z.B. ein PWM-Ausgang auf die einzelnen Ausgänge gelegt werden (siehe Abb. 7.1). Hierfür besitzen die ersten 8 digitalen Ausgänge eines jeden Connectors (PA0 bis PA7) einen Multiplexer mit einer 4-Bit-Adressierung (= bis zu 16 verschiedene Quellen). Als Default-Quelle ist nach einem Reset bzw. beim Hochfahren des PC's das DOUTyCNx-Register⁽¹⁾ als Peripherie festgelegt.

Um die Quelle zu verändern, muss die Quelladresse (siehe Abb. 7.2) in das Register DOUTMUXyCNx⁽¹⁾ geschrieben werden.

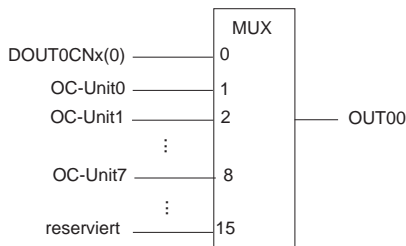


Abb. 7.1

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

Abb. 7.2

¹(y = Registernummer, x = Connector-Nummer)

7.3 Portadressen

Offset-Adresse	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 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX0CN1 <3:0>
0x03C4	DOUTMUX1CN1	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX1CN1 <3:0>
0x03C8	DOUTMUX2CN1	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX2CN1 <3:0>
0x03CC	DOUTMUX3CN1	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX3CN1 <3:0>
0x03D0	DOUTMUX4CN1	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX4CN1 <3:0>
0x03D4	DOUTMUX5CN1	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX5CN1 <3:0>
0x03D8	DOUTMUX6CN1	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX6CN1 <3:0>
0x03DC	DOUTMUX7CN1	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX7CN1 <3:0>
0x03E0	DOUTMUX0CN2	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX0CN2 <3:0>
0x03E4	DOUTMUX1CN2	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX1CN2 <3:0>
0x03E8	DOUTMUX2CN2	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX2CN2 <3:0>
0x03EC	DOUTMUX3CN2	31:16 15:0	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	reserviert (*)	DOUTMUX3CN2 <3:0>

DV03
(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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	reserviert (*)															
0x03F4	DOUTMUX5CN2	15:0	reserviert (*)															
0x03F8	DOUTMUX6CN2	31:16	reserviert (*)															
0x03FC	DOUTMUX7CN2	15:0	reserviert (*)															
0x0400	DOUTMUX0CN3	31:16	reserviert (*)															
0x0404	DOUTMUX1CN3	15:0	reserviert (*)															
0x0408	DOUTMUX2CN3	31:16	reserviert (*)															
0x040C	DOUTMUX3CN3	15:0	reserviert (*)															
0x0410	DOUTMUX4CN3	31:16	reserviert (*)															
0x0414	DOUTMUX5CN3	15:0	reserviert (*)															
0x0418	DOUTMUX6CN3	31:16	reserviert (*)															
0x041C	DOUTMUX7CN3	15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U				R/W			
	reserviert				DOUTMUXyCNx <3:0>			

Bit 31 - 4 reserviert (mit dem Wert 0 beschreiben)

Bit 3 - 0 **DOUTMUXyCNx <3:0>** (default = 0)

Bestimmt, welche Peripherie auf den digitalen Ausgang gelegt wird.

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 = reserviert

¹(y = Registernummer, x = Connector-Nummer)

8. Zähler

Die WASCO-PCIe8296 stellt insgesamt 32 32Bit-Ereigniszähler (steigende Flanken) zur Verfügung. Dabei kann jeder einzelne Zähler einem digitalen Eingang frei zugeordnet werden. Des weiteren besteht bei jedem Zähler die Möglichkeit, einen Interrupt bei Überlauf auszulösen.

8.1 Grundfunktion

1. Um einen Zähler zu verwenden, muss zu Beginn die Quelle ausgewählt werden. Hierfür hat jeder Zähler sein eigenes 32Bit Register (COUNTMUX⁽¹⁾).
2. Als nächstes muss der Counter über das Register COUNTLDx⁽¹⁾ vorgeladen werden. In der Regel wird hier der Wert 0 in das Register geschrieben.
3. Zuletzt wird durch das Setzen des ersten Bits im Register COUNTEx⁽¹⁾ der Zähler aktiviert. Ab nun beginnt der Zähler jede steigende Flanke zu zählen. Im Falle eines Überlaufs wird im Register COUNTIR das dem Zähler entsprechende Bit gesetzt. Um einen weiteren Überlauf zu erkennen, muss dieses Bit durch das Setzen des dem Zähler zugeteilten Bits im Register COUNTIRr gelöscht werden.
4. Der Zählerstand lässt sich durch das Auslesen des Registers COUNTx⁽¹⁾ ermitteln.

8.2 Interruptfunktion

Bei jedem Überlauf eines Zählers wird im Register COUNTIR das dem Zähler zugeordnete Bit gesetzt. Wurde der Interruptkanal durch das Setzen des entsprechenden Bits im Register COUNTIRe freigeschaltet, so wird der Überlauf an den Interruptcontroller weitergeleitet. Um das Überlaufbit rücksetzen zu können, muss im Register COUNTIRr das dem Zähler zugeordnete Bit gesetzt werden. Nach dem internen Rücksetzen des Überlaufbits wird automatisch das gesetzte Bit im Register COUNTIRr zurückgesetzt.

⁽¹⁾ x = Counter-Nummer)

8.3 Portadressen

Offset-Adresse	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	reserviert (*)															
		15:0	reserviert (*)															
0x1004	COUNT1e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1008	COUNT2e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x100C	COUNT3e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1010	COUNT4e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1014	COUNT5e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1018	COUNT6e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x101C	COUNT7e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1020	COUNT8e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1024	COUNT9e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1028	COUNT10e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x102C	COUNT11e	31:16	reserviert (*)															
		15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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	reserviert (*)															
		15:0	en															
0x1034	COUNT13e	31:16	reserviert (*)															
		15:0	en															
0x1038	COUNT14e	31:16	reserviert (*)															
		15:0	en															
0x103C	COUNT15e	31:16	reserviert (*)															
		15:0	en															
0x1040	COUNT16e	31:16	reserviert (*)															
		15:0	en															
0x1044	COUNT17e	31:16	reserviert (*)															
		15:0	en															
0x1048	COUNT18e	31:16	reserviert (*)															
		15:0	en															
0x104C	COUNT19e	31:16	reserviert (*)															
		15:0	en															
0x1050	COUNT20e	31:16	reserviert (*)															
		15:0	en															
0x1054	COUNT21e	31:16	reserviert (*)															
		15:0	en															
0x1058	COUNT22e	31:16	reserviert (*)															
		15:0	en															
0x105C	COUNT23e	31:16	reserviert (*)															
		15:0	en															

(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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	reserviert (*)															
		15:0	reserviert (*)															
0x1064	COUNT25e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1068	COUNT26e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x106C	COUNT27e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1070	COUNT28e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1074	COUNT29e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1078	COUNT30e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x107C	COUNT31e	31:16	reserviert (*)															
		15:0	reserviert (*)															
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>															

(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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-Adresse	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-Adresse	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	reserviert (*)															
		15:0	reserviert (*)															
0x1184	COUNTMUX1	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1188	COUNTMUX2	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x118C	COUNTMUX3	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1190	COUNTMUX4	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1194	COUNTMUX5	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1198	COUNTMUX6	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x119C	COUNTMUX7	31:16	reserviert (*)															
		15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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	reserviert (*)															
		15:0	reserviert (*)															
0x11A4	COUNTMUX9	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11A8	COUNTMUX10	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11AC	COUNTMUX11	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11B0	COUNTMUX12	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11B4	COUNTMUX13	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11B8	COUNTMUX14	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11BC	COUNTMUX15	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11C0	COUNTMUX16	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11C4	COUNTMUX17	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11C8	COUNTMUX18	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x11CC	COUNTMUX19	31:16	reserviert (*)															
		15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

Register COUNTx: (x = Counter Nummer)

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							R/W
	reserviert							en

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **COUNTx<0>** (default = 0)
 Zähler ein- bzw. ausschalten
 0 = ausschalten (default)
 1 = einschalten

Register COUNTx: (x = Counter Nummer)

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)

Aus dem Register kann der aktuelle Zählerstand des Zählers x ausgelesen sowie (z.B. für den Anfangszustand) beschrieben werden.

Register COUNTMUXx: (x = Counternummer)

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	R/W							
	COUNTMUXx <7:0>							

Bit 31 - 8 reserviert (mit dem Wert 0 beschreiben)

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

Registerwert bestimmt die Kartenperipherie, welche am Zähler anliegt

0 = digitaler Eingang 0 CN1 (default)

1 = digitaler Eingang 1 CN1

.

.

.

63 = digitaler Eingang 63 CN1

64 = digitaler Eingang 0 CN2

.

.

.

127 = digitaler Eingang 63 CN2

128 = digitaler Eingang 0 CN3

.

.

.

191 = digitaler Eingang 63 CN3

255 - 192 = reserviert -> mit 0 belegen

9. Timer

Die zur Verfügung stehenden 32Bit-Timer lassen sich als Zeitgeber oder zur konfigurierbaren Intervall-Interruptauslösung verwenden. Dabei können Intervalle zwischen 0 und 4294967295 μs in 1 μs -Schritten eingestellt werden.

9.1 Anwendung Intervall-Interruptauslösung

1. Zu Beginn muss der Timer x durch Löschen von Bit 0 des Registers $\text{TIMERx}^{(1)}$ deaktiviert und anschließend resetted werden. Der Reset wird durch das Beschreiben des Registers $\text{TIMERx}^{(1)}$ mit dem Wert 0 durchgeführt.
2. Anschließend muss der Intervall festgelegt werden. Die Intervalldauer wird im beschreibbaren 32Bit Register $\text{TIMERCOMPx}^{(1)}$ festgelegt.
$$\text{Intervalldauer} = (\text{TIMERCOMPx}^{(1)} + 1) * 1\mu\text{s}$$
3. Damit der Timer nach Ablauf des Intervalls einen Interrupt auslöst, muss dieser freigeschalten werden, was über das Setzen des entsprechenden Bits im Register TIMERIRe geschieht. (Achtung: auch der Interruptcontroller muss freigeschalten sein)
4. Ist der Timer fertig konfiguriert, kann er durch das Setzen von Bit 0 im Register $\text{TIMERx}^{(1)}$ aktiviert werden.
5. Wurde der Interrupt ausgelöst, kann dies im Register TIMERIR überprüft werden. Um einen erneuten Interrupt zu empfangen, muss das Quellenbit durch das Setzen des zugehörigen Resetbits im Register TIMERIRr gelöscht werden.

⁽¹⁾ x = Timer-Nummer)

9.2 Portadressen

Offset-Adresse	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	reserviert (*)															
		15:0	en															
0x1404	TIMER1e	31:16	reserviert (*)															
		15:0	reserviert (*)															
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>															

(*) reservierter Bereich ist mit 0 zu belegen

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							R/W
	reserviert							en

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **TIMERxe<0>** (default = 0)
 Timer starten bzw. stoppen
 0 = gestoppt (default)
 1 = gestartet

(x = Timer-Nummer)

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)

Aus dem Register kann der aktuelle Wert des Timers x ausgelesen sowie (z.B. für den Anfangszustand) beschrieben werden

(x = Timer-Nummer)

Register TIMERCMPx:

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							
	TIMERCMPx<31:24>							
23:16	R/W							
	TIMERCMPx<23:16>							
15:8	R/W							
	TIMERCMPx<15:8>							
7:0	R/W							
	TIMERCMPx<7:0>							

Bit 31 - 0 **TIMERCMPx<0>** (default = 0)
 Der Wert des TIMERCMP-Registers bestimmt die Intervalldauer des Timers.
 TIMERCMP = Intervalldauer - 1

(x = Timer-Nummer)

10. Input-Capture-Einheit

Die Input-Capture-Einheiten (IC-Unit) ermöglichen dem Anwender Pulsdauer und Periodendauer von eingehenden Signalen zu messen. Dabei hat jede Einheit einen eigenen 32Bit-Timer für die Zeitmessung in 1µs-Schritten und kann durch Programmierung jedem digitalen Eingang zugeordnet werden.

10.1 Kontinuierliche Messung periodischer Signale

In diesem Modus wird bei Aktivierung das Eingangssignal regelmäßig abgetastet und dabei Periodendauer sowie Pulsdauer ermittelt. Dabei beginnt die Einheit bei der ersten steigenden Flanke am Eingang mit der Messung und schließt diese bei folgender steigender Flanke ab. Nach Abschließen der Messung wird automatisch die Periodendauer und Pulsdauer berechnet und die Werte in die Register ICPERIODLx sowie ICPULSLx geschrieben. Bei der nächsten steigenden Flanke beginnt die Einheit von selbst mit der nächsten Messung.

10.1.1 Anwendung

1. Sorgen Sie dafür, dass die zu verwendende IC-Unit vor der Konfiguration deaktiviert ist. Die IC-Unit ist durch das Löschen des Bit 0 im Register ICUNITxe ausgeschaltet.
2. Ist die IC-Unit deaktiviert, führen Sie die Konfiguration im Register ICCONFIGx durch. Für die kontinuierliche Messung periodischer Signale schreiben Sie in dem Mode-Bereich den Wert b0000(bin).
3. Ist die Unit konfiguriert, muss die Quelle ausgewählt werden, indem diese in das Register ICMUXx geschrieben wird.
4. Um nun die Messung zu starten, setzen Sie das Bit 0 im Register ICUNITxe.

10.1.2 Interruptfunktion

Neben der Messung der Periodendauer sowie der Pulsdauer gibt es die Möglichkeit nach Abschluss dieser einen Interrupt auszulösen. Dafür aktivieren Sie die Interruptfunktion durch Setzen des entsprechenden Bits im Register ICUNITRe. Wird der Interrupt ausgelöst, so ist die Quelle im Register ICUNITIR auszulesen sowie durch das Setzen des entsprechenden Bits im Register ICUNITIRr wieder freischaltbar.

(x = IC-Nummer)

10.2 Portadressen

Offset-Adresse	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	reserviert (*)													en		
			reserviert (*)															
0x14C4	ICUNIT1e	31:16	reserviert (*)													en		
			reserviert (*)															
0x14E0	ICCONFIG0	31:16	reserviert (*)													ICMODE0		
			reserviert (*)															
0x14E4	ICCONFIG1	31:16	reserviert (*)													ICMODE1		
			reserviert (*)															
0x1500	ICMUX0	31:16	reserviert (*)													ICMUX0 <7:0>		
			reserviert (*)															
0x1504	ICMUX1	31:16	reserviert (*)													ICMUX1 <7:0>		
			reserviert (*)															
0x1540	ICPULS0	31:16	ICPULS0 <31:16>															
			ICPULS0 <15:0>															
0x1544	ICPULS1	31:16	ICPULS1 <31:16>															
			ICPULS1 <15:0>															
0x1560	ICPERIOD0	31:16	ICPERIOD0 <31:16>															
			ICPERIOD0 <15:0>															
0x1504	ICPERIOD0	31:16	ICPERIOD1 <31:16>															
			ICPERIOD1 <15:0>															

(*) reservierter Bereich ist mit 0 zu belegen

Register ICUNITx: (x = IC-Nummer)

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							R/W
	reserviert							en

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **ICUNITx<0>** (default = 0)
 IC-Unit starten bzw. stoppen
 0 = gestoppt (default)
 1 = gestartet (führt Messungen durch)

Register ICCONFIGx: (x = IC-Nummer)

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U				R/W			
	reserviert				ICMODEx <3:0>			

Bit 31 - 4 reserviert (mit dem Wert 0 beschreiben)

Bit 3 - 0 **ICMODEx<3:0>** (default = 0)
 Bestimmt den Modus, in welchem die IC-Einheit läuft
 0 = Mode 0 führt eine kontinuierliche Messung von Perioden- und Pulsdauer periodischer Signale durch (default)
 1 - 15 = reserviert (mit 0 belegen)

Register ICMUXx: (x = IC-Nummer)

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	R/W							
	ICMUXx <7:0>							

Bit 31 - 8 reserviert (mit dem Wert 0 beschreiben)

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

Registerwert bestimmt die Kartenperipherie, welche an der IC-Einheit anliegt

0 = digitaler Eingang 0 CN1 (default)

1 = digitaler Eingang 1 CN1

.

.

.

63 = digitaler Eingang 63 CN1

64 = digitaler Eingang 0 CN2

.

.

.

127 = digitaler Eingang 63 CN2

128 = digitaler Eingang 0 CN3

.

.

.

191 = digitaler Eingang 63 CN3

255 - 192 = reserviert -> mit 0 belegen

Register ICPULSx: (x = IC-Nummer)

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>**

Aus dem Register kann die zuletzt gemessene Pulsdauer in μ s ausgelesen werden

Register ICPERIODx: (x = IC-Nummer)

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>**

Aus dem Register kann die zuletzt gemessene Periodendauer in μ s ausgelesen werden

11. Output-Compare-Einheit

Mit der Output-Compare-Einheit (OC-Unit) ermöglicht die WASCO-PCIe8296-Karte dem Anwender, an digitalen Ausgängen eine PWM-Funktion zu koppeln oder einzelne Pulse auszugeben. Dabei können Rechtecksignale mit einer Periodendauer von 2 bis 2^{32} μs und einer Pulsdauer von 1 bis 2^{32} μs erzeugt werden.

11.1 PWM

Die OC-Einheit ermöglicht es dem Anwender, an den ersten 8 digitalen Ausgängen (PA0 bis PA7) eines jeden Connectors eine PWM anzulegen.

11.1.1 Funktionsweise

Um die PWM zu realisieren, verwendet die OC-Einheit einen beschreibbaren 32Bit-Timer mit einstellbarer Periodendauer in μs (OCPERIODx) und ein Zwei-Stufen-Compare-Register (OCUNITORx) zur Einstellung der Pulsdauer in μs . Ist die OC-Einheit deaktiviert, so liegt am Ausgang ein LOW an. Wird die OC-Einheit im PWM-Modus gestartet, beginnt der Timer im μs -Takt zu zählen und der OC-Ausgang bleibt auf LOW. Erreicht der Timer den Wert im Register OCPERIODx, so läuft er zum nächsten Takt über und beginnt mit dem Zählen wieder bei 0. Des Weiteren wird beim Überlauf die konfigurierte Pulsdauer aus dem OCUNITORx-Register in das am Timer anliegende Pipeline-Register übernommen, sowie der OC-Ausgang auf HIGH gesetzt (vorausgesetzt die Pulsdauer ist nicht $0\mu\text{s}$). Entspricht der Timerwert (OCTIMERx) dem Wert des am Timer anliegenden Pipeline-Registers, so wird der Ausgang bis zum nächsten Timer-Überlauf auf LOW gesetzt.

Mit dem Einsatz des Zwei-Stufen-Pulsdauer-Registers wird gewährleistet, dass bei einer Änderung der Pulsdauer während des OC-Betriebs jede Periode vor der Übernahme vollständig ausgegeben wird. Möchte man die erste Periode nach dem Start der OC-Einheit, in welcher kein Puls am Ausgang ausgegeben wird, überspringen, so kann der Timer entsprechend mit einem anderen Wert als 0 vorgeladen werden (-> Verkürzung der Periode).

(x = OC-Nummer)

11.1.2 Berechnung der Registerwerte

OCPERIODx⁽²⁾ = Periodendauer_in_μs + 1 [μs]

OCUNITORx⁽²⁾ = Pulsdauer_in_μs [μs]

OCTIMER = Takte [μs]

11.1.3 Anwendungsbeispiel

1. OC-Einheit durch das Löschen des entsprechenden Bits im Register OCUNITxe⁽²⁾ deaktivieren.
2. OC-Einheit mit dem gewünschten digitalen Ausgang verbinden. Dafür muss in dem zum digitalen Ausgang gehörigen DOUTMUXyCNx-Register⁽¹⁾ die Quelle ausgewählt werden (siehe Kapitel „Digitale Ausgänge mit anderen Hardwarekomponenten belegen“).
3. Den OC-Timer der OC-Einheit vorladen. Hier wird in der Regel der Wert 0x00000000 in das Register OCTIMERx⁽²⁾ geschrieben.
4. Die Periodendauer der PWM festlegen. Hierfür muss die Periodendauer wie folgt in das Register OCUNITORx⁽²⁾ geschrieben werden:
OCPERIODx⁽²⁾ = Periodendauer - 1 [μs]
5. Die Pulsdauer festlegen. Hierfür muss die Pulsdauer wie folgt in das Register OCPULSx⁽²⁾ geschrieben werden:
OCPULSx⁽²⁾ = Pulsdauer [μs]
6. Den OC-Unit-Modus auswählen. Für das Verwenden der PWM muss der Wert 0 im Bereich Mode des Registers OCONFIGx⁽²⁾ geschrieben werden.
7. OC-Einheit durch das Setzen des entsprechenden Bits im Register OCUNITxe⁽²⁾ aktivieren.

¹ (y = Registernummer, x = Connector-Nummer)

² (x = OC-Nummer)

11.2 Pulsausgabe

Die OC-Einheit ermöglicht es dem Anwender neben der PWM auch einzelne μ s-genaue Pulse an den digitalen Ausgängen auszugeben.

11.2.1 Funktionsweise

Für die Ausgabe einzelner positiver Pulse müssen die Register zuerst konfiguriert werden. Anschließend kann mit jedem Löschen und anschließendem Setzen des en-Bits im Register OCUNITx ein Puls wie in folgender Grafik ausgegeben werden.

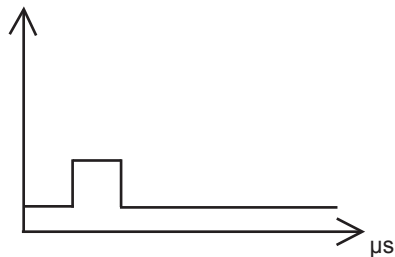


Abb. 11.1

Um die Pulsdauer zu ändern, muss die OC-Einheit immer deaktiviert sein (en-Bit im Register OCUNITx gelöscht).

11.2.2 Berechnung der Registerwerte

$$\text{OCPERIODx} = \text{Pulsdauer_in_}\mu\text{s}$$

(x = OC-Nummer)

11.2.3 Anwendungsbeispiel

1. OC-Einheit konfigurieren

- a) Deaktivieren Sie die Einheit durch Löschen (= 0) des en-Bits im Register OCUNITx⁽²⁾
- b) OC-Einheit mit dem gewünschten digitalen Ausgang verbinden. Dafür muss in dem zum digitalen Ausgang gehörigen DOUTMUXyCNx-Register⁽¹⁾ die Quelle ausgewählt werden (siehe Kapitel „Digitale Ausgänge mit anderen Hardwarekomponenten belegen“).
- c) Laden Sie den OC-Timer mit dem Wert 0 vor, indem Sie das Register OCTIMERx⁽²⁾ mit 0x00000000 beschreiben
- d) Laden Sie die gewünschte Pulsdauer in das Register OCPERIODx⁽²⁾ (siehe auch Kapitel 11.2.2 Berechnung der Registerwerte)
- e) Laden Sie das Register OCUNITORx⁽²⁾ mit dem Wert 1
- f) Wählen Sie den Single-Puls-Modus aus, indem Sie den Wert 1 in den Mode-Bereich im Register OCCONFIGx⁽²⁾ schreiben

2. Puls ausgeben

- a) OC-Einheit durch Löschen des Enable-Bits im Register OCUNITx⁽²⁾ deaktivieren
- b) OC-Einheit durch Setzen des Enable-Bits im Register OCUNITx⁽²⁾ aktivieren. Daraufhin wird am ausgewählten Ausgang der Puls angelegt.

¹ (y = Registernummer, x = Connector-Nummer)

² (x = OC-Nummer)

11.3 Portadressen

Offset-Adresse	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	reserviert (*)															
		15:0	en															
0x15C4	OCUNIT1e	31:16	reserviert (*)															
		15:0	en															
0x15C8	OCUNIT2e	31:16	reserviert (*)															
		15:0	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>															

Offset-Adresse	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
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>															

Offset-Adresse	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>															

Offset-Adresse	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	reserviert (*)															
		15:0	reserviert (*)															
0x1684	OCCONFIG1	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1688	OCCONFIG2	31:16	reserviert (*)															
		15:0	reserviert (*)															
			OCMODE0 <3:0>															
			OCMODE1 <3:0>															
			OCMODE2 <3:0>															

(*) reservierter Bereich ist mit 0 zu belegen

Register OCUNITx: (x = OC-Nummer)

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							R/W
	reserviert							en

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **OCUNITx<0>** (default = 0)
 OC-Unit starten bzw. stoppen
 0 = gestoppt (default)
 1 = gestartet

Register OCTIMERx: (x = OC-Nummer)

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)
 Aus dem Register kann der aktuelle Wert des OC-Timers x ausgelesen sowie (z.B. für den Anfangszustand) beschrieben werden.

Register OCUNITORx: (x = OC-Nummer)

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)
 Definiert die Pulsdauer der OC-Einheit x in μs
 Pulsdauer = OCUNITORx [μs]

Register OCPERIODx: (x = OC-Nummer)

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)
 Definiert die Periodendauer der OC-Einheit x in μs
 Periodendauer = OCPERIODx + 1 [μs]

Register OCONFIGx: (x = OC-Nummer)

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U				R/W			
	reserviert				OCMODEx <3:0>			

Bit 31 - 4 reserviert (mit dem Wert 0 beschreiben)

Bit 3 - 0 **OCMODEx<3:0>** (default = 0)

Bestimmt den Modus der OC-Einheit

0 = Mode 0 -> Pulsweitenmodulation (default)

1 = Mode 1 -> einzelne Pulsausgabe

2 - 15 = reserviert (mit 0 belegen)

12. Interruptcontroller

Um die einzelnen Interrupts aus den verschiedenen möglichen Quellen zu verarbeiten, wird der interne Interruptcontroller verwendet. Über ihn können einzelne Interruptquellen freigegeben oder die Quellen ausgelöster Interrupts identifiziert werden.

Wie Abb. 11.1 zeigt, stellt das 32Bit-Register INTCON die Zentraleinheit dar. Hier werden alle möglichen Interruptquellen (teilweise schon aufbereitet) zusammengeführt.

Wird ein Interrupt z.B. durch eine Flanke an einem digitalen Eingang ausgelöst, wird dieser an das dritte Bit im INTCON-Register weitergeleitet. Sobald der Registerwert von INTCON ungleich 0 ist (ein oder mehrere Interrupts liegen an), wird dieser an INT weitergeleitet. INT stellt dabei eine Art Torregister dar. Ist die Interruptfunktion der Karte aktiviert ($INTe = 1$) und das Register zurückgesetzt, so wird der Interrupt an den PC weitergegeben. Wurde der Interrupt ausgelöst, wird die Interruptleitung zum PC für weitere Interrupts gesperrt. Bevor diese wieder freigegeben werden kann, muss die Quelle bestimmt und der Auslöser behoben werden. Während dieser Zeit können zwar weitere Interrupts von anderen Quellen auf der Karte ausgelöst werden (z.B. durch andere Flankeneingänge oder Timer), jedoch werden diese nicht an den PC weitergeleitet. Wird ein Interruptauslöser bearbeitet und die jeweilige Quelle wieder freigegeben, wird das jeweilige Bit im Register INTCON automatisch auf 0 gesetzt. Sind alle Interruptauslöser bearbeitet und zurückgesetzt ($INT = 0$), kann das Register INT durch das Setzen des ersten Bits im Register $INTe$ gelöscht werden und ein weiterer Interrupt kann an den PC geschickt werden.

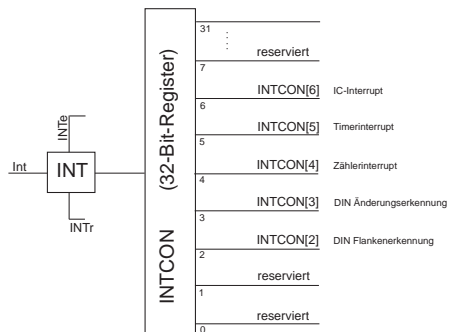


Abb. 12.1

Da z.B. bei einem digitalen Eingangsflankeninterrupt mehrere Quellen in Frage kommen, werden die Quellleitungen zum Register INTCON meist aufbereitet. Dies bedeutet, dass ein oder mehrere weitere 32Bit-Register an dem jeweiligen Bit des INTCON-Registers anliegen kann. Im Falle der Flankenerkennung der digitalen Eingänge sind dies die Register DINF0CN1, DINF1CN1, DINF0CN2, DINF1CN2, DINF0CN3 und DINF1CN3. In dem Register stellt jedes Bit einen digitalen Eingang dar (siehe Registerbeschreibung). Die einzelnen Eingänge können einzeln als Interruptquelle freigegeben werden (DINFyeCNx) und nach einem ausgelösten und bearbeiteten Interrupt wieder freigegeben werden (DINFyrCNx).

Ist dies geschehen, geht das jeweilige Bit im Register INTCON automatisch auf 0.

Anwendung

1. Konfiguration

- a) Überprüfen, ob alle Interruptquellen gelöscht sind (INTCON = 0)
- b) Einzelne Interruptquellen freigeben (siehe Dokumentation der entsprechenden Peripherie)
- c) Interruptfunktion freischalten (INTE = 1)

2. Interruptroutine

- a) Interruptquellenperipherie identifizieren durch das Lesen von INTCON und falls nötig entsprechender Peripherieregister
- b) Interrupt löschen
- c) Überprüfen, ob noch weitere Interrupts anliegen (INTCON = 0?)
- d) falls c) nicht zutrifft, alle anderen Interrupts ebenfalls löschen
- e) Interruptfunktion wieder freigegeben (INTR = 1)

12.1 Portadressen

Offset-Adresse	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	reserviert (*)															
		15:0	en															
0x0284	INTr	31:16	reserviert (*)															
		15:0	re															
0x0288	INTCON	31:16	reserviert (*)															
		15:0	reserviert (*)	INT7	INT6	INT5	INT4	INT3	INT2	reserviert (*)								
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>															

(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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>															
0x2F0	DINIF0rCN3	15:0	DINIF1rCN2 <15:0>															
0x2F4	DINIF1rCN3	31:16	DINIF0rCN3 <31:16>															
0x300	DINIF0CN1	15:0	DINIF0rCN3 <15:0>															
0x304	DINIF1CN1	31:16	DINIF1rCN3 <31:16>															
0x308	DINIF0CN2	15:0	DINIF0CN1 <15:0>															
0x30C	DINIF1CN2	31:16	DINIF1CN1 <31:16>															
0x310	DINIF0CN3	15:0	DINIF0CN2 <15:0>															
0x314	DINIF1CN3	31:16	DINIF1CN2 <31:16>															
0x4B0	DINICe	31:16	DINIF0CN3 <31:16>															
		15:0	DINIF0CN3 <15:0>															
			DINIF1CN3 <31:16>															
			DINIF1CN3 <15:0>															
			reserviert (*)															
			reserviert (*)															
			en															

(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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	reserviert (*)															
		15:0	reserviert (*)															
0x4D8	DINIC	31:16	reserviert (*)															
		15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

Offset-Adresse	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	reserviert (*)															
		15:0	COUNTIRe <15:0>															
0x0344	COUNTIRr	31:16	reserviert (*)															
		15:0	COUNTIRr <15:0>															
0x0348	COUNTIR	31:16	reserviert (*)															
		15:0	COUNTIR <31:16>															
		15:0	COUNTIR <15:0>															
0x0360	TIMERIRE	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x0364	TIMERIRr	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x0368	TIMERIR	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x036C	ICUNITIRE	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x0370	ICUNITIRr	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x0374	ICUNITIR	31:16	reserviert (*)															
		15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							R/W
	reserviert							en

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **INTe<0>** (default = 0)

Interruptfunktion der Karte freischalten bzw. sperren

0 = Interrupt gesperrt (default)

1 = Interrupt freigeben

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							W
	reserviert							en

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **INTr<0>** Mit dem Schreiben einer 1 wird das Register INTCON auf 0 gesetzt und es kann ein neuer Interrupt ausgelöst werden.

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U	R			R		U	
	reserviert	INTCON <6:4>			INTCON <1:0>		reserviert	

Bit 31 - 7 reserviert (mit dem Wert 0 belegt)

Bit 6 **INTCON<6>**: signalisiert einen Interrupt von einer der IC-Einheiten
 0 = es wurde kein Interrupt von einer IC-Einheit ausgelöst
 1 = Eine der IC-Einheiten hat einen Interrupt ausgelöst

Bit 5 **INTCON<5>**: signalisiert einen Interrupt von einem der Timer
 0 = es wurde kein Interrupt von einem Timer ausgelöst
 1 = Eine der Timer hat einen Interrupt ausgelöst

Bit 4 **INTCON<4>**: signalisiert einen Interrupt von einem der Zähler
 0 = es wurde kein Interrupt von einem Zähler ausgelöst
 1 = Einer der Zähler hat einen Interrupt ausgelöst

Bit 3 **INTCON<3>**: signalisiert einen Interrupt ausgelöst durch eine Änderung des für diesen Interrupt frei geschalteten digitalen Eingang
 0 = es wurde kein Änderungsinterrupt durch die digitalen Eingänge ausgelöst
 1 = es wurde ein Änderungsinterrupt durch die digitalen Eingänge ausgelöst

Bit 2 **INTCON<2>**: signalisiert einen Interrupt ausgelöst durch eine steigende Flanke an einem des für diesen Interrupt frei geschalteten digitalen Eingang
 0 = es wurde kein Flankeninterrupt durch digitale Eingänge ausgelöst
 1 = es wurde ein Flankeninterrupt durch die digitalen Eingänge ausgelöst

Bit 1 - 0 reserviert (mit dem Wert 0 belegt)

Register DINIF0eCNx (x = Connector-Nummer):

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 dem Register können einzelne digitale Eingänge als Quelle freigeschaltet werden, um bei einer anliegenden positiven Flanke einen Interrupt auszulösen. Jedes Bit entspricht einem digitalen Eingang (z.B. PA0 => DINIF0eCNx<0>, PB5 => DINIF0eCNx<13>). Eine 1 im Bit schaltet die Flankeninterruptfunktion des digitalen Eingangs frei, eine 0 sperrt diese.

Register DINIF1eCNx (x = Connector-Nummer):

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 dem Register können einzelne digitale Eingänge als Quelle freigeschaltet werden, um bei einer anliegenden positiven Flanke einen Interrupt auszulösen. Jedes Bit entspricht einem digitalen Eingang (z.B. PE0 => DINIF1eCNx<0>, PBF => DINIF1eCNx<13>). Eine 1 im Bit schaltet die Flankeninterruptfunktion des digitalen Eingangs frei, eine 0 sperrt diese.

Register DINIF0rCNx (x = Connector-Nummer):

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>** Jedes Bit entspricht einem digitalen Eingang (z.B. PA0 => DINIF0rCNx<0>, PB5 => DINIF0rCNx<13>). Wurde an einem digitalen Eingang ein Flankeninterrupt ausgelöst, muss dessen Signalbit im Register DINIF0r zurückgesetzt werden. Dies erfolgt durch das Setzen (= 1) des entsprechenden DINIF0rCNx-Bits. Die DINIF0rCNx-Bits werden nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

Register DINIF1rCNx (x = Connector-Nummer):

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>** Jedes Bit entspricht einem digitalen Eingang (z.B. PE0 => DINIF1rCNx<0>, PF5 => DINIF1rCNx<13>). Wurde an einem digitalen Eingang ein Flankeninterrupt ausgelöst, muss dessen Signalbit im Register DINIF1r zurückgesetzt werden. Dies erfolgt durch das Setzen (= 1) des entsprechenden DINIF1r-Bits. Die DINIF1r-Bits werden nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

Register DINIF0CNx (x = Connector-Nummer):

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>** zeigt, ob an einem der digitalen Eingänge eine steigende Flanke angelegen ist. Jedes Bit entspricht einem digitalen Eingang (z.B. PA0 => DINIF0CNx<0>, PB5 => DINIF0CNx<13>). Eine 1 im jeweiligen Bit bedeutet, dass seit dem letzten Reset am Eingang eine steigende Flanke angelegt war, eine 0, dass keine Flanke angelegen ist.

Register DINIF1CNx (x = Connector-Nummer):

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>** zeigt, ob an einem der digitalen Eingänge eine steigende Flanke angelegen ist. Jedes Bit entspricht einem digitalen Eingang (z.B. PE0 => DINIF1CNx<0>, PF5 => DINIF1CNx<13>). Eine 1 im jeweiligen Bit bedeutet, dass seit dem letzten Reset am Eingang eine steigende Flanke angelegt war, eine 0, dass keine Flanke angelegen ist.

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							R/W
	reserviert							en

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **DINICe<0>** (default = 0) Interruptfunktion zur Erkennung von Änderungen an den digitalen Eingängen freischalten
 0 = Interrupt gesperrt (default)
 1 = Interrupt freigeben

Register DINICC0eCNx: (x = Connector-Nummer):

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)
 Einzelne digitale Eingänge für Interruptfunktion zur Erkennung von Änderungen an den digitalen Eingängen freischalten bzw. sperren. Jedes Bit entspricht einem digitalen Eingang (z.B. PA0 => DINICC0eCNx<0>, PB5 => DINICC0eCNx<13>)
 0 = Interrupt gesperrt (default)
 1 = Interrupt freigeben

Register DINICC1eCNx: (x = Connector-Nummer):

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)
 Einzelne digitale Eingänge für Interruptfunktion zur Erkennung von Änderungen an den digitalen Eingängen freischalten bzw. sperren. Jedes Bit entspricht einem digitalen Eingang (z.B. PE0 => DINICC1eCNx<0>, PF5 => DINICC1eCNx<13>)
 0 = Interrupt gesperrt (default)
 1 = Interrupt freigeben

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							W
	reserviert							re

Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)
 Bit 0 **DINICr<0>** (default = 0)
 Wurde ein Interrupt durch eine Änderung an den digitalen Eingängen ausgelöst, muss das Quellenregister DINIC zurück auf 0 gesetzt werden. Dies erfolgt durch das Setzen (=1) des DINICr-Bits. Das DINICr-Bit wird nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

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 undefiniert

Bit 0 **DINIC<0>** zeigt an, ob eine Änderung an einem freigegebenen digitalen Eingang stattgefunden hat.

0 = keine Änderung

1 = Änderung an einem freigegebenen digitalen Eingang

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)

Hiermit können die Interruptfunktionen der Zähler frei geschaltet werden. Jedes Bit entspricht einem Zähler

(z.B. Zähler 0 => COUNTIRE<0>, Zähler 13 => COUNTIRE<13>)

0 = Interrupt gesperrt (default)

1 = Interrupt freigeben

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>**

Jedes Bit entspricht einem Zähler (z.B. Zähler 0 => COUNTIRr<0>, Zähler 13 => COUNTIRr<13>). Wurde durch einen Zähler ein Interrupt ausgelöst, muss dessen Signalbit im Register COUNTIR zurückgesetzt werden. Dies erfolgt durch das Setzen (= 1) des entsprechenden COUNTIRr-Bits. Die COUNTIRr-Bits werden nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

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>** zeigt an, ob durch einen Zähler ein Interrupt ausgelöst wurde. Jedes Bit entspricht einem Zähler (z.B. Zähler 0 => COUNTIR<0>, Zähler 13 => COUNTIR<13>) 0 = kein Interrupt 1 = Interrupt ausgelöst

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U						R/W	
	reserviert						TIMERIRe <1:0>	

Bit 31 - 2 reserviert (mit dem Wert 0 beschreiben)

Bit 1 - 0 **TIMERIRe<1:0>** (default = 0) hiermit können die Interruptfunktionen der Timer freigeschaltet werden. Jedes Bit entspricht einem Timer (z.B. Timer 0 => TIMERIRe<0>, Timer 1 => TIMERIRe<1>) 0 = Interrupt gesperrt (default) 1 = Interrupt freigegeben

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U						R/W	
	reserviert						TIMERIRr <1:0>	

Bit 31 - 2 reserviert (mit dem Wert 0 beschreiben)

Bit 1 - 0 **TIMERIRr<1:0>** Jedes Bit entspricht einem Timer (z.B. Timer 0 => TIMERIRr<0>, Timer 1 => TIMERIRr<1>). Wurde durch einen Timer ein Interrupt ausgelöst, muss dessen Signalbit im Register TIMERIRr zurückgesetzt werden. Dies erfolgt durch das Setzen (= 1) des entsprechenden TIMERIRr-Bits. Die TIMERIRr-Bits werden nach dem Zurücksetzen von selbst auf 0 zurückgesetzt.

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	R							
	reserviert						TIMERIR <1:0>	

Bit 31 - 2 reserviert (mit dem Wert 0 beschreiben)

Bit 1 - 0 **TIMERIR<1:0>** zeigt an, ob durch einen der Timer ein Interrupt ausgelöst wurde. Jedes Bit entspricht dabei einem Timer (z.B. Timer 0 => TIMERIR<0>, Timer 1 => TIMERIR<1>)
 0 = kein Interrupt ausgelöst
 1 = Interrupt ausgelöst

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U							
	reserviert						R/W	
							ICUNITIRE<1:0>	

Bit 31 - 2 reserviert (mit dem Wert 0 beschreiben)

Bit 1 - 0 **ICUNITIRE<1:0>** (default = 0) hiermit können die Interruptfunktionen der IC-Einheiten freigeschaltet werden. Jedes Bit entspricht einer IC-Einheit (z.B. IC-Unit 0 => ICUNITIRE<0>, IC-Unit 1 => ICUNITIRE<1>)
 0 = Interrupt gesperrt (default)
 1 = Interrupt freigegeben

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	U						R/W	
	reserviert						ICUNITIRr<7:0>	

Bit 31 - 2 reserviert (mit dem Wert 0 beschreiben)

Bit 1 - 0 **ICUNITIRr<1:0>** Jedes Bit entspricht einer IC-Einheit (z.B. IC-Unit 0 => ICUNITIRr<0>, IC-Unit 1 => ICUNITIRr<1>). Wurde durch eine IC-Einheit ein Interrupt ausgelöst, muss dessen Signalbit im Register ICUNITIR zurückgesetzt werden. Dies erfolgt durch das Setzen (= 1) des entsprechenden ICUNITIRr-Bits. Die ICUNITIRr-Bits werden nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

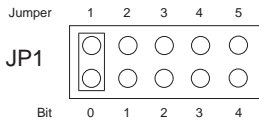
Register ICUNITIR:

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							
	reserviert							
23:16	U							
	reserviert							
15:8	U							
	reserviert							
7:0	R							
	reserviert						ICUNITIR <1:0>	

Bit 31 - 2 reserviert (mit dem Wert 0 beschreiben)

Bit 1 - 0 **ICUNITIR<1:0>** zeigt an, ob durch eine der IC-Einheiten ein Interrupt ausgelöst wurde. Jedes Bit entspricht dabei einer IC-Einheit (z.B. IC-Unit 0 => ICUNITIR<0>, IC-Unit 1 => ICUNITIR<1>).
 0 = kein Interrupt ausgelöst
 1 = Interrupt ausgelöst

13. Board-Identifikation



Die Board-Identifikation dient zur Unterscheidung mehrerer PC-Karten gleichen Typs im Computer. Sie erfolgt durch einen Jumperblock, welcher per Software gelesen werden kann.

Die zu lesende Board-Identifikation besteht aus einem Byte (8 Bit) und ist wie folgt aufgebaut:

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“ entspricht „1“, wenn der Jumper gesetzt ist, sonst „0“

Mittels des Lesebefehls kann die Jumperstellung des Jumperblocks JP1 ausgelesen werden. Die nicht benutzten Bits sind grundsätzlich „0“, ein gesetzter Jumper wird als „1“ gelesen.

Z.B.



Ergebnis des Lesebefehls: \$05

13.1 Portadressen

Offset-Adresse	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 (*)										reserviert (*)						Board ID
		15:0	reserviert (*)																

(*) reservierter Bereich ist mit 0 zu belegen

14. Programmierung unter Windows[®]

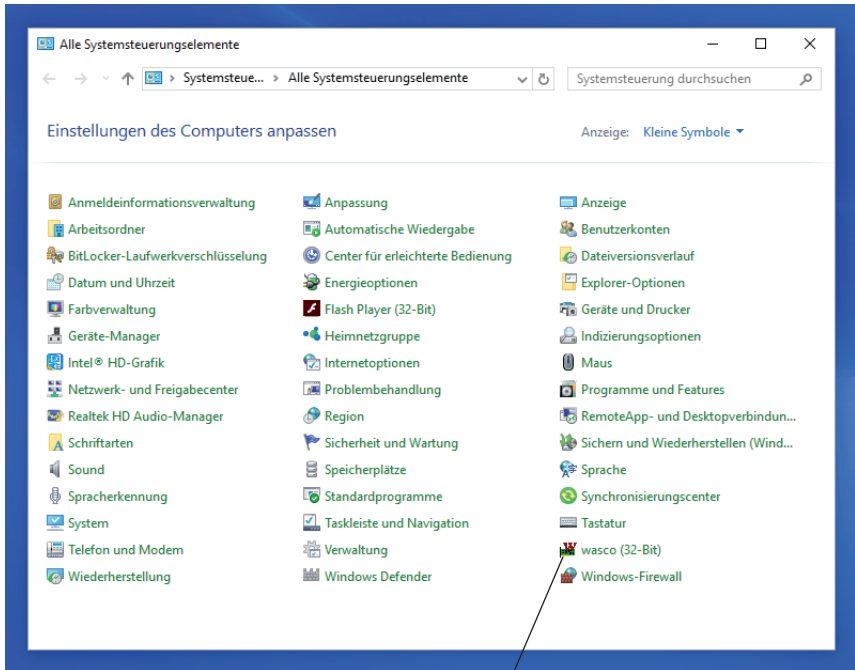
14.1 Installation des Windows[®] Treibers

Für die Anwendung der Karte unter Windows[®] ist es notwendig, einen speziellen Treiber zu installieren, der den Zugriff auf die Karte ermöglicht. Unter Windows[®] 10, 8 und 7 meldet das Betriebssystem selbständig nach dem Einschalten des PCs, dass eine neue Hardware-Komponente gefunden wurde. In diesem Fall legen Sie den Datenträger ein und weisen das System an, von diesem die Treiber-Dateien zu installieren. Sollte sich das Betriebssystem nicht melden, kann der Treiber auch im Gerätemanager installiert werden.

14.2 Installation der Windows[®] Entwicklungsdateien

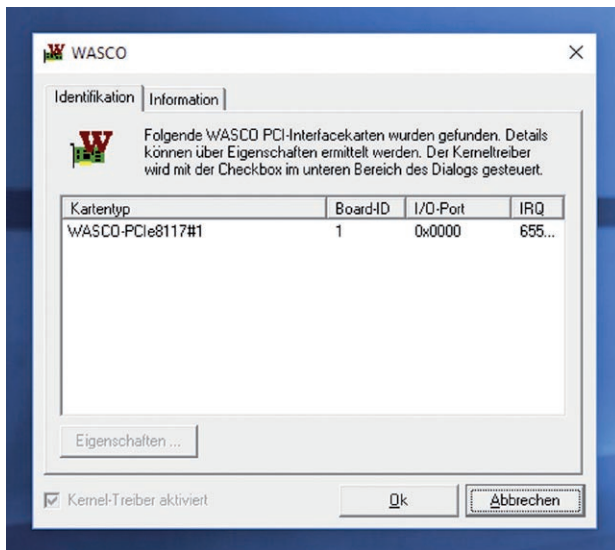
Zur Installation der Windows[®] Entwicklungsdateien führen Sie bitte die Datei "Setup.exe" im Ordner Treiber auf der mitgelieferten CD aus und folgen Sie den Installationsanweisungen.





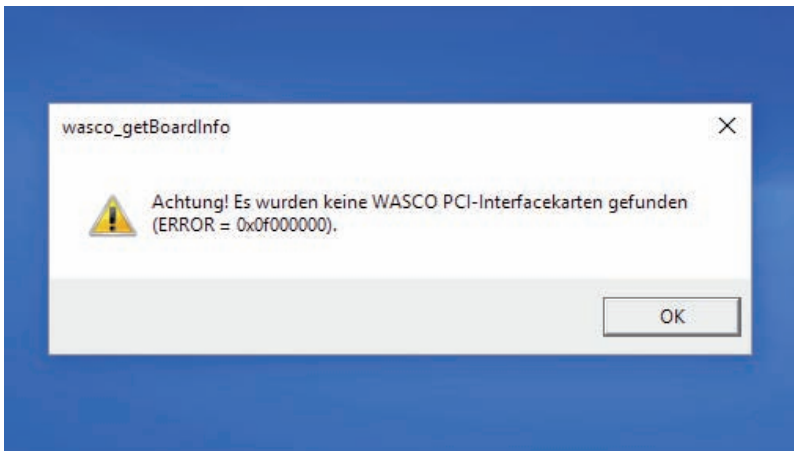
Wurden die Treiber- und Entwicklungsdateien vollständig installiert, finden Sie in der Systemsteuerung Ihres Rechners ein Icon zur Lokalisierung aller im System vorhandenen **wasco**[®] PCI- und PCIe-Karten.

Starten Sie die Kartenabfrage durch einen Doppelklick auf das "wasco®"-Icon. Folgendes Fenster erscheint: (Als Beispiel wurde hier eine WASCO-PCIe8117 verwendet)



Wurde Ihre Karte im System erkannt, wird der Boardname WASCO-PCIe8296, Board-ID, I/O-Adresse sowie die mögliche Interruptnummer für die jeweilige Karte in diesem Fenster angezeigt. Des Weiteren kann über den Reiter "Information" die Treiber-Version sowie der Standort der Treiberdatei abgefragt werden.

Wurde Ihre Karte im System nicht erkannt, wird folgende Fehlermeldung angezeigt:



Informieren Sie sich im Kapitel Fehlersuche über die möglichen Ursachen!

14.3 Programmierung der WITIO-PCIe192 mit **wasco**[®]-Treiber

Nach Installation der Entwicklungsdateien von Kithara mittels des Setup-Programms befinden sich in dem Ordner `.../wasco/` die entsprechenden Entwicklungsdateien sowie die Beispielprogramme. Weitere Beispielprogramme, speziell für den Zugriff auf die WITIO-PCIe192, befinden sich auf der beiliegenden CD sowie auf unserer Homepage.

Die Programmierung der Hardwarekomponenten der WITIO-PCIe192 erfolgt durch den Zugriff auf Memory Mapped I/O-Adressen, die sich abhängig von der vom BIOS des Systems für die WITIO-PCIe192 vergebenen Basisadresse ergeben. Eine genauere Beschreibung zur Programmierung befindet sich in der Treiberdokumentation.

14.4 Zugriff auf die Karte WITIO-PCIe192_{ULTRA}

Der Zugriff auf die WITIO-PCIe192_{ULTRA} erfolgt ausschließlich über den Boardnamen (Kartentyp) WASCO-PCIe8296

14.5 Zuordnung der Memory Mapped I/O-Adressen

Die Memory Mapped I/O-Adressen der einzelnen Hardware-Komponenten ergeben sich abhängig von der Basisadresse wie folgend anhand einiger Beispiele gezeigt:

Port/Register	BA + Offset	RD/WR
erste 32 Eingänge von CN1 lesen (PD7..PD0, PC7..PC0, PB7..PB0, PA7..PA0)	BA + \$140	RD
zweite 32 Eingänge von CN1 lesen (PH7..PH0, PG7..PG0, PF7.F0, PE7..PE0)	BA + \$144	RD
erste 32 Ausgänge von CN1 lesen/schreiben (PD7..PD0, PC7..PC0, PB7..PB0, PA7..PA0)	BA + \$160	RD/WR
zweite 32 Ausgänge von CN1 lesen/schreiben (PH7..PH0, PG7..PG0, PF7.F0, PE7..PE0)	BA + \$164	RD/WR
Board Identifikation	BA+ \$FF8	RD

15. Programmierung unter Linux[®]

Für die Anwendung der Karte unter Linux[®] wird auf der mitgelieferten CD oder auf unserer Webseite ein Linux wasco[®]-Treiber zur Verfügung gestellt. Dieser liegt in Code-Form vor und kann daher auch jederzeit vom Kunden geändert und angepasst werden.

Der Treiber beinhaltet kein Interrupthandling, welches jedoch bei Bedarf jederzeit vom Kunden hinzugefügt werden kann. Grund dafür ist, dass das individuelle Interrupthandling im Kernel-Modul bearbeitet werden sollte.

15.1 Installation des Linux[®] Treibers

Unter Linux[®] ist es für den Zugriff auf die Karte notwendig, einen speziellen Treiber zu installieren, der den Zugriff auf die Karte ermöglicht. Dafür legen Sie den Datenträger ein und kopieren den Ordner des Linux-Treibers auf ihr System. Zur Installation folgen Sie den Angaben im readme-File

15.2 Unterstützte Linux-Distributionen/Kernelversionen

Der wasco[®]-Treiber wurde in folgenden Umgebungen getestet:

Ubuntu[®] 18.04.4 LTS (Kernel: 5.3.0)

15.3 Programmierung der WITIO-PCIe192 mit wasco[®]-Treiber

Die Programmierung der Hardwarekomponenten der WITIO-PCIe192 erfolgt durch den Zugriff auf Memory Mapped I/O-Adressen, die sich abhängig von der vom BIOS des Systems für die WITIO-PCIe192 vergebenen Basisadresse ergeben.

Der Zugriff findet über die Funktionen pread und pwrite statt. Dabei werden unter der Programmiersprache C und C/C++ keine weiteren externen Libraries benötigt. Beispiele für den genauen Zugriff auf die WITIO-PCIe192 befinden sich auf der beiliegenden CD sowie auf unserer Homepage.

Der Linux[®] wasco[®]-Treiber unterstützt kein Interrupthandling, welches jedoch bei Bedarf jederzeit durch den Kunden hinzugefügt werden kann.

15.4 Zugriff auf die Karte WITIO-PCIe192

Der Zugriff auf die WITIO-PCIe192^{Ultra} erfolgt ausschließlich über den Boardnamen (Kartentyp) WASCO-PCIe8182

15.5 Zuordnung der Memory Mapped I/O-Adressen

Die Memory Mapped I/O-Adressen der einzelnen Hardware-Komponenten ergeben sich abhängig von der Basisadresse wie folgt:

32-Bit-Modus

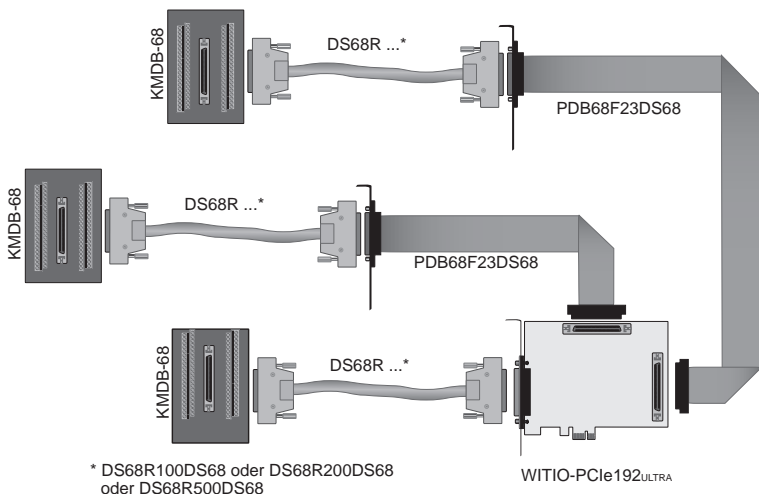
Port/Register	BA + Offset	RD/WR
erste 32 Eingänge von CN1 lesen (PD7..PD0, PC7..PC0, PB7..PB0, PA7..PA0)	BA + \$140	RD
zweite 32 Eingänge von CN1 lesen (PH7..PH0, PG7..PG0, PF7.F0, PE7..PE0)	BA + \$144	RD
erste 32 Ausgänge von CN1 lesen/schreiben (PD7..PD0, PC7..PC0, PB7..PB0, PA7..PA0)	BA + \$160	RD/WR
zweite 32 Ausgänge von CN1 lesen/schreiben (PH7..PH0, PG7..PG0, PF7.F0, PE7..PE0)	BA + \$164	RD/WR
Board Identifikation	BA+ \$3E0	RD

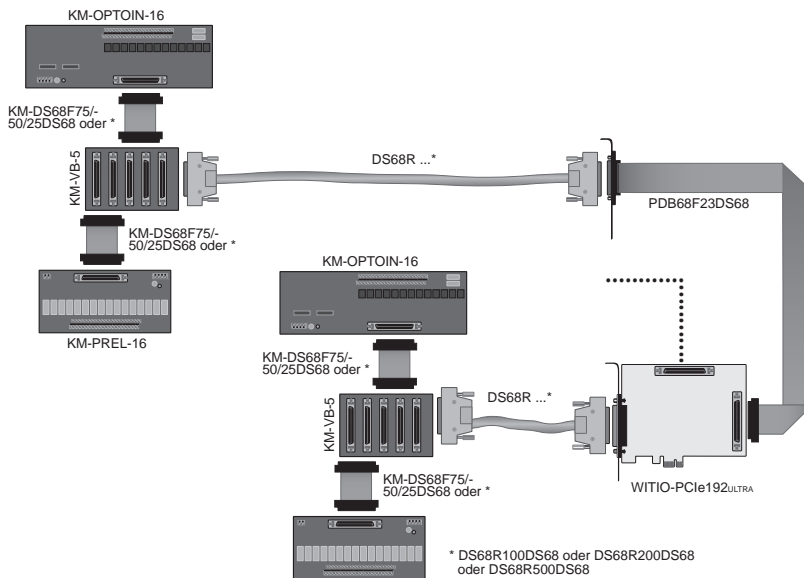
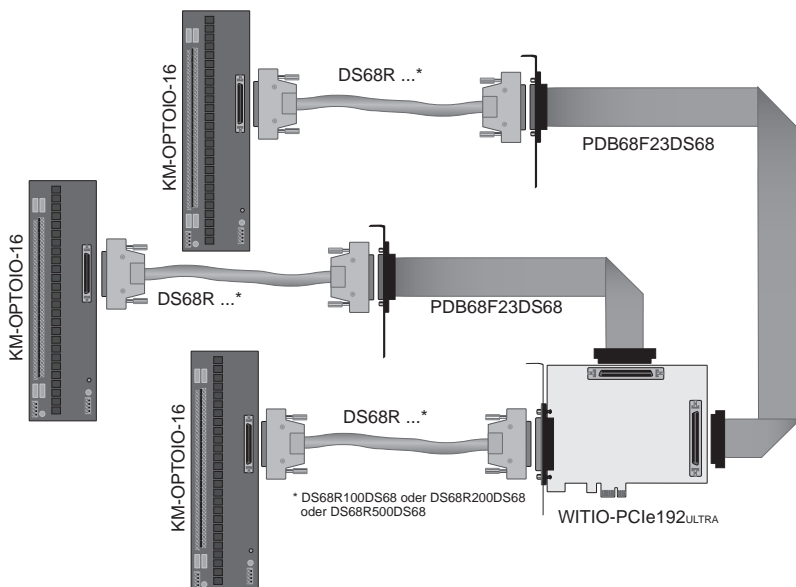
16. Zubehör

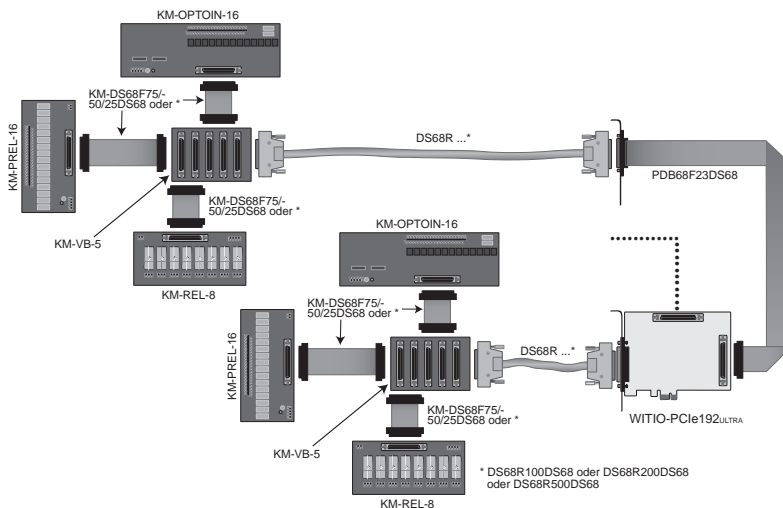
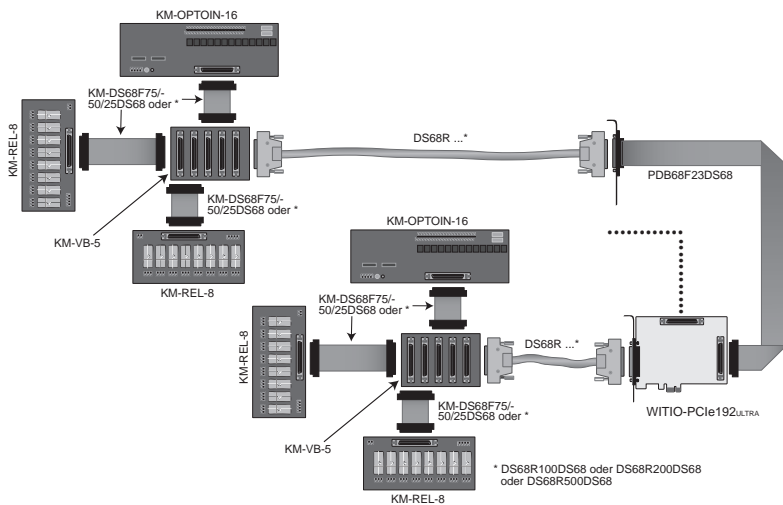
16.1 Passendes **wasco**[®]-Zubehör

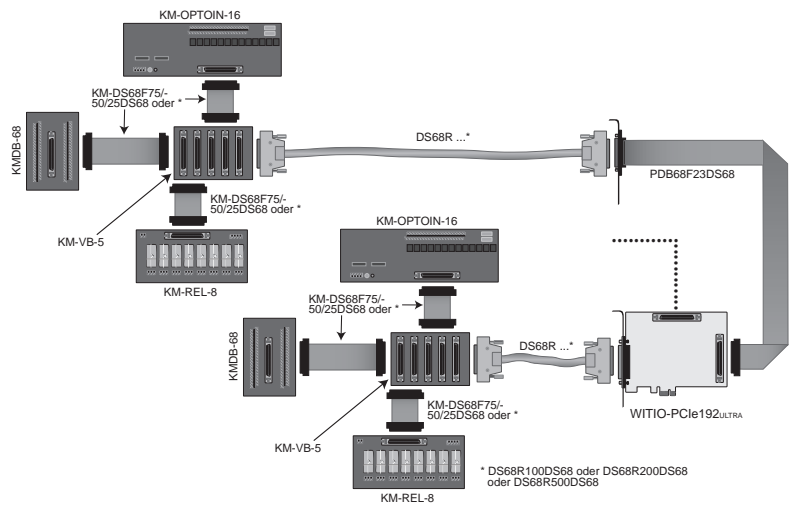
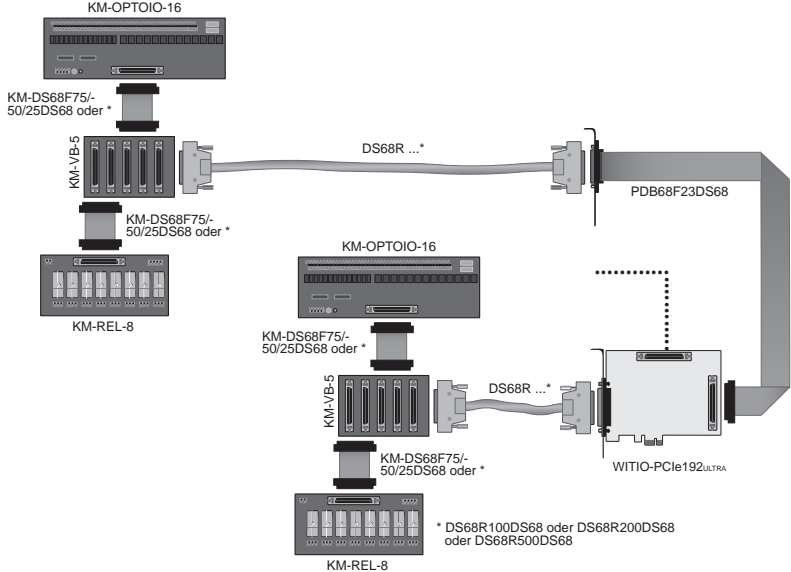
Anschlusssteile	EDV-Nr.
PDB68F23DS68 Steckerverlegungsset	A-498500
DS68R200DS68 Verbindungsleitung	A-492400
DS68R500DS68 Verbindungsleitung	A-492800
KMDB-68 Klemm-Modul	A-494800
KM-OPTOIN-32 Optokoppler-Modul	A-483600
KM-OPTOOUT-32 Optokoppler-Modul	A-484600
KM-PREL-16 Relais-Modul	A-485400
KM-REL-8 Relais-Modul	A-486200
KM-VB-5 Verbindungs-Modul	A-488200

16.2 Anschlussstechnik (Anwendungsbeispiele)









16.3 Einzelkomponenten zur Eigenkonfektionierung

Anschlusssteile	EDV-Nr.
SCSI-II Stecker 68pol. für Flachbandleitung	A-553200
SCSI-II Buchse 68pol. für Flachbandleitung	A-557200
SCSI-II Stecker 68pol. für Lötanschluss	A-555340
Slotblech mit Ausschnitt für 68pol. Stecker/Buchse	A-577800
Flachbandleitung 68pol.	A-572800

17. Fehlersuche

Nachfolgend finden Sie eine kurze Zusammenstellung der häufigsten bekannten Fehlerursachen, die während der Inbetriebnahme oder während der Arbeit mit der WITIO-PCIe192 auftauchen können.

Prüfen Sie bitte zunächst folgende Punkte, bevor Sie mit Ihrem Händler Kontakt aufnehmen.

1. Sitzt die WITIO-PCIe192 richtig in der Steckverbindung?
2. Sind alle Kabelverbindungen in Ordnung?
3. Wurde die Karte im System richtig erkannt?
Prüfen Sie hierzu alle Einstellungen in Ihrem Rechner oder wenden Sie sich an Ihren Systemadministrator.
(Da es sich hierbei um Einstellungen im BIOS des Rechners handelt, können wir hier nicht näher darauf eingehen und verweisen hierzu auf Ihr Systemhandbuch)!
4. Wurde die neueste Treiberversion des **wasco**[®] Treibers installiert?
Updates finden Sie unter: <http://www.messcomp.com>

18. Technische Daten

Ein- /Ausgänge

Kanäle: 192

Ausgangspegel 3.3V/5V, durch Jumper einstellbar.

Alle Eingänge mit programmierbaren Eingangsfiler, Flankeninterrupt und Change-Interrupt

Ausgangsstrom: 5 mA pro Kanal

In 8-Bit-Gruppen als Ein- oder Ausgänge programmierbar

Output-Compare-Einheit

8 OC-Einheiten zu den ersten 8 Ausgangskanälen jeden Connectors schaltbar

Auflösung 32Bit [1µs]

PWM-Erzeugung

Erzeugung einzelner Pulse

Input-Capture-Einheit

2 IC-Einheiten an allen Eingängen schaltbar

Auflösung 32Bit [1µs]

Perioden- und Pulsdauermessungen

Zähler

32 Zähler an allen Eingängen schaltbar

Auflösung 32 Bit

Timer

2 Timer

Auflösung 32Bit [1µs]

Quarzoszillator

4 MHz

Board-Identifikation

5-fach Jumperblock

Anschlusstecker

3 * 68polige SCSI-Buchse

Bussystem

32 Bit PCIe-Bus

(Interner Datenzugriff 32Bit)

Abmessungen

137 mm x 111 mm (l x b)

Sonstiges

Sicherung und Kontroll-LEDs für Spannungsversorgung

19. Produkthaftungsgesetz

Hinweise zur Produkthaftung

Das Produkthaftungsgesetz (ProdHaftG) regelt die Haftung des Herstellers für Schäden, die durch Fehler eines Produktes verursacht werden.

Die Verpflichtung zu Schadenersatz kann schon gegeben sein, wenn ein Produkt aufgrund der Form der Darbietung bei einem nichtgewerblichen Endverbraucher eine tatsächlich nicht vorhandene Vorstellung über die Sicherheit des Produktes erweckt, aber auch wenn damit zu rechnen ist, dass der Endverbraucher nicht die erforderlichen Vorschriften über die Sicherheit beachtet, die beim Umgang mit diesem Produkt einzuhalten wären.

Es muss daher stets nachweisbar sein, dass der nichtgewerbliche Endverbraucher mit den Sicherheitsregeln vertraut gemacht wurde.

Bitte weisen Sie daher im Interesse der Sicherheit Ihre nichtgewerblichen Abnehmer stets auf Folgendes hin:

Sicherheitsvorschriften

Beim Umgang mit Produkten, die mit elektrischer Spannung in Berührung kommen, müssen die gültigen VDE-Vorschriften beachtet werden.

Besonders sei auf folgende Vorschriften hingewiesen:
VDE0100; VDE0550/0551; VDE0700; VDE0711; VDE0860.
Sie erhalten VDE-Vorschriften beim
vde-Verlag GmbH
Bismarckstraße 33, 10625 Berlin.

- * Vor Öffnen eines Gerätes den Netzstecker ziehen oder sicherstellen, dass das Gerät stromlos ist.
- * Bauteile, Baugruppen oder Geräte dürfen nur in Betrieb genommen werden, wenn sie vorher in ein berührungssicheres Gehäuse eingebaut wurden. Während des Einbaus müssen sie stromlos sein.
- * Werkzeuge dürfen an Geräten, Bauteilen oder Baugruppen nur benutzt werden, wenn sichergestellt ist, dass die Geräte von der Versorgungsspannung getrennt sind und elektrische Ladungen, die in im Gerät befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- * Spannungsführende Kabel oder Leitungen, mit denen das Gerät, das Bauteil oder die Baugruppe verbunden sind, müssen stets auf Isolationsfehler oder Bruchstellen untersucht werden. Bei Feststellen eines Fehlers in der Zuleitung muss das Gerät unverzüglich aus dem Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- * Bei Einsatz von Bauelementen oder Baugruppen muss stets auf die strikte Einhaltung der in der zugehörigen Beschreibung genannten Kenndaten für elektrische Größen hingewiesen werden.
- * Wenn aus den vorgelegten Beschreibungen für den nichtgewerblichen Endverbraucher nicht eindeutig hervorgeht, welche elektrischen Kennwerte für ein Bauteil gelten, so muss stets ein Fachmann um Auskunft ersucht werden.

Im Übrigen unterliegt die Einhaltung von Bau und Sicherheitsvorschriften aller Art (VDE, TÜV, Berufsgenossenschaften usw.) dem Anwender/Käufer.

20. EG-Konformitätserklärung

Für das folgende mit CE-Kennzeichen gekennzeichnete Erzeugnis

WITIO-PCIe192^{ULTRA}
EDV-Nummer A-864810

wird hiermit bestätigt, dass es den Anforderungen der betreffenden EMC-Richtlinien 2014/30/EU entspricht. Bei Nichteinhaltung der im Handbuch angegebenen Vorschriften zum bestimmungsgemäßen Betrieb des Produktes verliert diese Erklärung Ihre Gültigkeit.

Folgende Normen wurden berücksichtigt:

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)

Diese Erklärung wird verantwortlich für den Hersteller

Messcomp Datentechnik GmbH
Neudecker Str. 11
83512 Wasserburg

abgegeben durch

Dipl.Ing.(FH) Hans Schnellhammer

Wasserburg, 19.07.2018



Referenzsystem-Bestimmungsgemäßer Betrieb

Die PC-Erweiterungskarte ist ein nicht selbständig betreibbares Gerät, dessen CE-Konformität nur bei gleichzeitiger Verwendung von zusätzlichen Computerkomponenten beurteilt werden kann. Die Angaben zur CE-Konformität beziehen sich deshalb ausschließlich auf den bestimmungsgemäßen Einsatz der PC-Erweiterungskarte in folgendem Referenzsystem:

Schaltschrank:	Vero IMRAK 3400	804-530061C 802-563424J 802-561589J
19" Gehäuse:	Vero PC-Gehäuse	145-010108L
19" Gehäuse:	Zusatzelektronik	519-112111C
Motherboard:	ASUS P5G41-M LE	
Schnittstellen:	WITIO-PCle192 _{ULTRA}	A-864810