

# **OPTOIO-PCle16 ULTRA**

**EDV-Nr.: A-829410**

16 Eingänge über Optokoppler  
16 Ausgänge über Optokoppler  
16\*32 Bit Zähler  
Timer  
OC- und IC-Einheiten  
Board-Identifikation

**wasco<sup>®</sup>**

Handbuch

## Copyright® 2017 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 OPTOIO-PCIe16<sup>ULTRA</sup> und dieser Dokumentation direkt oder indirekt entstandene Schäden.

## Inhaltsverzeichnis

<b>1. Produktbeschreibung .....</b>	<b>5</b>
<b>2. Installation der OPTOIO-PCIe16<sup>ULTRA</sup> .....</b>	<b>6</b>
2.1 Installation der Karte in den Rechner .....	6
<b>3. Anschlussstecker .....</b>	<b>7</b>
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 CN2 auf D-Sub 37 (Steckerverlegungsset) .....	10
<b>4. Systemkomponenten .....</b>	<b>11</b>
4.1 Blockschaltbild .....	11
4.2 Zugriff auf die Systemkomponenten .....	12
<b>5. 16 Optokopplereingänge .....</b>	<b>13</b>
5.1 Pinbelegung der Eingangsoptokoppler .....	13
5.2 Eingangsspannungsbereiche .....	14
5.3 Eingangsbeschaltung .....	16
5.4 Eingangsstrom .....	16
5.5 Zugriff auf die Eingänge .....	17
5.6 Optokopplereingänge mit digitalem Filter .....	17
5.7 Interruptfunktionen der Optokopplereingänge .....	18
5.8 Portadressen .....	22
<b>6. 16 Optokopplerausgänge .....</b>	<b>25</b>
6.1 Pinbelegung der Ausgangsoptokoppler .....	25
6.2 Optokopplerdaten .....	25
6.3 Ausgangsbeschaltung .....	25
6.4 Funktionen der Optokopplerausgänge .....	26
6.5 Portadressen .....	27
<b>7. Zähler .....</b>	<b>30</b>
7.1 Grundfunktion .....	30
7.2 Interruptfunktion .....	30
7.3 Portadressen .....	31
<b>8. Timer .....</b>	<b>37</b>
8.1 Anwendung Intervall-Interruptauslösung .....	37
8.2 Portadressen .....	38

<b>9. Input-Capture-Einheit</b> .....	41
9.1 Kontinuierliche Messung periodischer Signale.....	41
9.2 Portadressen .....	43
<b>10. Output-Compare-Einheit</b> .....	47
10.1 PWM.....	47
10.2 Pulsausgabe.....	49
<b>11. Interruptcontroller</b> .....	55
<b>12. Board-Identifikation</b> .....	69
<b>13. Programmierung unter Windows</b> <sup>®</sup> .....	71
13.1 Installation des Windows <sup>®</sup> Treibers .....	71
13.2 Installation der Windows <sup>®</sup> Entwicklungsdateien .....	71
13.3 Programmierung der OPTOIO-PCIe16 mit wasco <sup>®</sup> -Treiber .....	74
13.4 Zugriff auf die Karte OPTOIO-PCIe16 <sup>ULTRA</sup> .....	75
13.5 Zuordnung der Memory Mapped I/O-Adressen.....	75
<b>14. Programmierung unter Linux</b> <sup>®</sup> .....	76
14.1 Installation des Linux <sup>®</sup> Treibers .....	76
14.2 Unterstützte Linux-Distributionen/Kernelversionen .....	76
14.3 Programmierung der OPTOIO-PCIe16 mit wasco <sup>®</sup> -Treiber .....	76
14.4 Zugriff auf die Karte OPTOIO-PCIe16 <sup>ULTRA</sup> .....	77
14.5 Zuordnung der Memory Mapped I/O-Adressen.....	77
<b>15. Zubehör</b> .....	78
15.1 Passendes wasco <sup>®</sup> -Zubehör .....	78
15.2 Anschlusstechnik (Anwendungsbeispiele).....	78
15.3 Einzelkomponenten zur Eigenkonfektionierung .....	79
<b>16. Fehlersuche</b> .....	80
<b>17. Technische Daten</b> .....	81
<b>18. Produkthaftungsgesetz</b> .....	82
<b>19. EG-Konformitätserklärung</b> .....	84

## 1. Produktbeschreibung

Die OPTOIO-PCIe16<sup>ULTRA</sup> (Boardname: WASCO-PCIe8117) bietet 16 digitale Eingänge und 16 digitale Ausgänge mit galvanischer Trennung für jeden einzelnen Kanal. Die Potentialtrennung gewährleisten bei den Ein- wie Ausgängen hochwertige Optokoppler. Spezielle, leistungsfähige Ausgangsoptokoppler bewältigen einen Schaltstrom von bis zu 150 mA. Jeder Ein- und Ausgang ist zusätzlich durch Schutzdioden gegen schädliche Spannungsspitzen und Impulse geschützt. Über Jumperblöcke sind zwei verschiedene Eingangs-Spannungsbereiche einstellbar. Jedem Eingang kann ein programmierbarer Filter zugeordnet werden, um Eingangsimpulse unter einer einstellbaren Impulsdauer auszublenden.

Neben den galvanisch getrennten I/Os stehen mehrere Zähler, Output-Compare-Einheiten (z.B. PWM) und Input-Capture-Einheiten (z.B. Periodenmessung) zur Verfügung. Interruptauslösungen sind durch alle Optokopplereingänge, Zähler und IC-Einheiten sowie zeitabhängig durch zwei 32-Bit-Timer möglich. Die Anschlüsse der Ausgangsoptokoppler sind über am Slotblech der Platine montierten 37poligen D-Sub-Buchse zugeführt. Die Optokoppler-Eingänge liegen am 40poligen Pfostenstecker auf der Platine an. Über ein als Option erhältliches Steckerverlegungs-Set ist die Verlegung auf eine 37polige D-Sub-Buchse mit Slotblech möglich.

Die Steckerbelegung ist bezüglich der Optokoppler-Ein/Ausgänge identisch zu den ISA-Bus-Karten OPTOIO-16, zu den PCI-Karten OPTOIO-PCI16 und zu OPTOIO-PCIe16, ein Umstieg auf PCIe16<sup>ULTRA</sup> 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 OPTOIO-PCle16<sup>ULTRA</sup>**

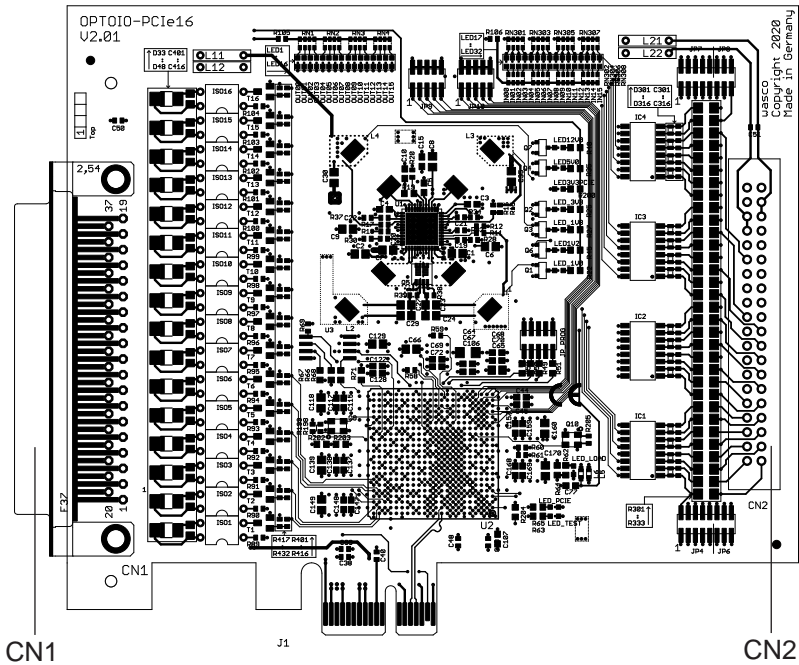
### **2.1 Installation der Karte in den Rechner**

Achten Sie vor dem Einbau der OPTOIO-PCle16 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 OPTOIO-PCle16 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.

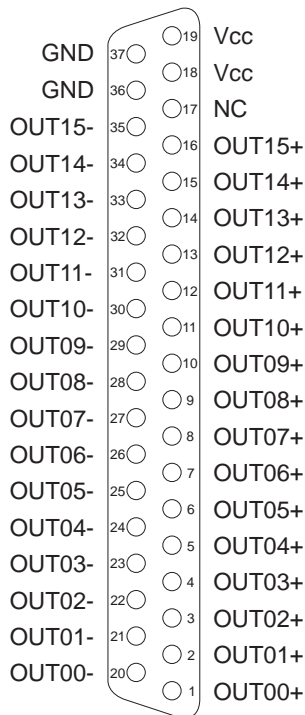
### 3. Anschlusstecker

#### 3.1 Lage der Anschlussstecker auf der Platine



- CN1: Optokoppler-Ausgänge OUT0...OUT15
- CN2: Optokoppler-Eingänge IN0...IN15

## 3.2 Steckerbelegung von CN1



### Vcc:

Interne Versorgungsspannung (+ 5V) der PCIe-Karte (nur durch Einlöten einer Drahtbrücke an L11). Hier niemals eine externe Spannung anlegen.

### GND:

Masse des Rechners (nur durch Einlöten einer Drahtbrücke an L12)

### NC:

Pin nicht belegt



### 3.3 Steckerbelegung von CN2

NC	40	○	○	39	NC
NC	38	○	○	37	Vcc
GND	36	○	○	35	Vcc
GND	34	○	○	33	NC
IN15-	32	○	○	31	IN15+
IN14-	30	○	○	29	IN14+
IN13-	28	○	○	27	IN13+
IN12-	26	○	○	25	IN12+
IN11-	24	○	○	23	IN11+
IN10-	22	○	○	21	IN10+
IN09-	20	○	○	19	IN09+
IN08-	18	○	○	17	IN08+
IN07-	16	○	○	15	IN07+
IN06-	14	○	○	13	IN06+
IN05-	12	○	○	11	IN05+
IN04-	10	○	○	9	IN04+
IN03-	8	○	○	7	IN03+
IN02-	6	○	○	5	IN02+
IN01-	4	○	○	3	IN01+
IN00-	2	○	○	1	IN00+

**Vcc:**

Interne Versorgungsspannung (+ 5V) der PCIe-Karte (nur durch Einlöten einer Drahtbrücke an L21). Hier niemals eine externe Spannung anlegen.

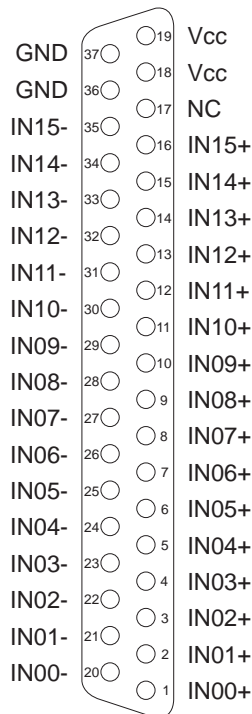
**GND:**

Masse des Rechners (nur durch Einlöten einer Drahtbrücke an L22)

**NC:**

Pin nicht belegt

## 3.4 Steckerbelegung von CN2 auf D-Sub 37 (Steckerverlegungsset)



### Vcc:

Interne Versorgungsspannung (+5V) der PCIe-Karte (nur durch Einlöten einer Drahtbrücke an L21). Hier niemals eine externe Spannung anlegen.

### GND:

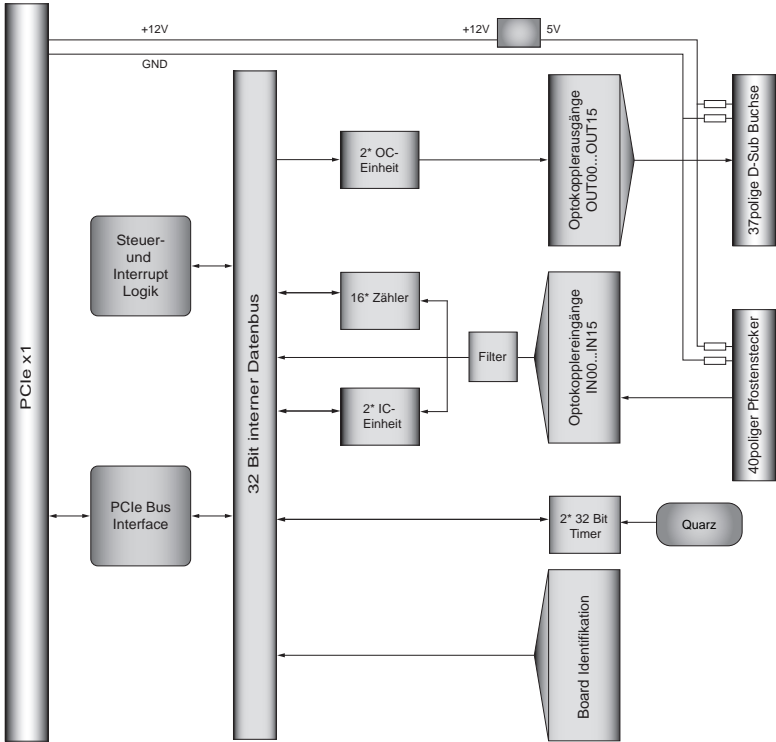
Masse des Rechners (nur durch Einlöten einer Drahtbrücke an L22)

### NC:

Pin nicht belegt

# 4. Systemkomponenten

## 4.1 Blockschaftbild



## 4.2 Zugriff auf die Systemkomponenten

Der Zugriff auf die Hardware-Komponenten der OPTOIO-PCIe16 erfolgt durch das Lesen von bzw. Schreiben in Memory-Mapped I/O-Adressen mit Hilfe von Library-Funktionen. Die für die OPTOIO-PCIe16 relevanten Adressen ergeben sich abhängig von einer vom Bios vergebenen Basisadresse. Der Zugriff auf die OPTOIO-PCIe16 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).

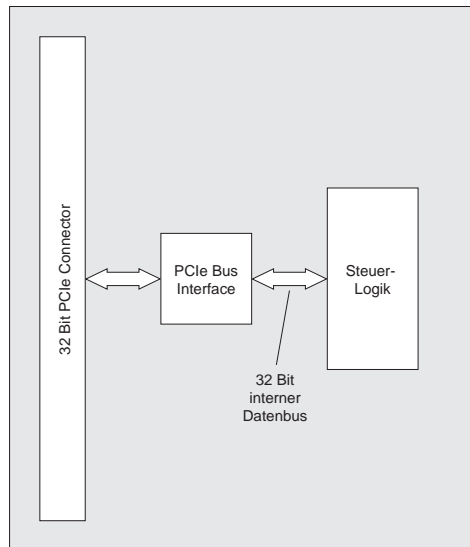


Abb. 4.1

## 5. 16 Optokopplereingänge

Die OPTOIO-PCIe16 verfügt über 16 Eingangskanäle, deren galvanische Trennung mittels Optokoppler erreicht wird. Die Isolationsspannung zwischen Masse des Computers und Eingang beträgt 500 Volt, während die Spannung zwischen den Eingangskanälen auf 50 Volt begrenzt ist.

### 5.1 Pinbelegung der Eingangsoptokoppler

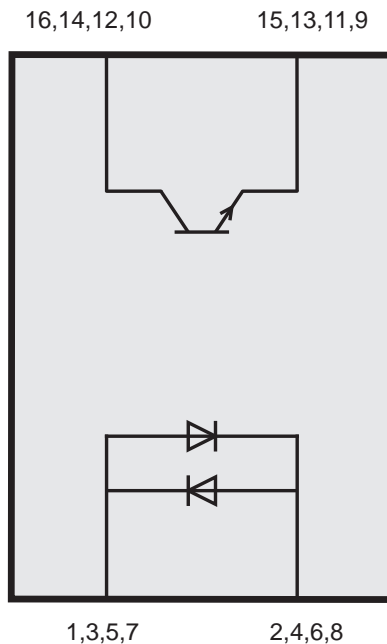
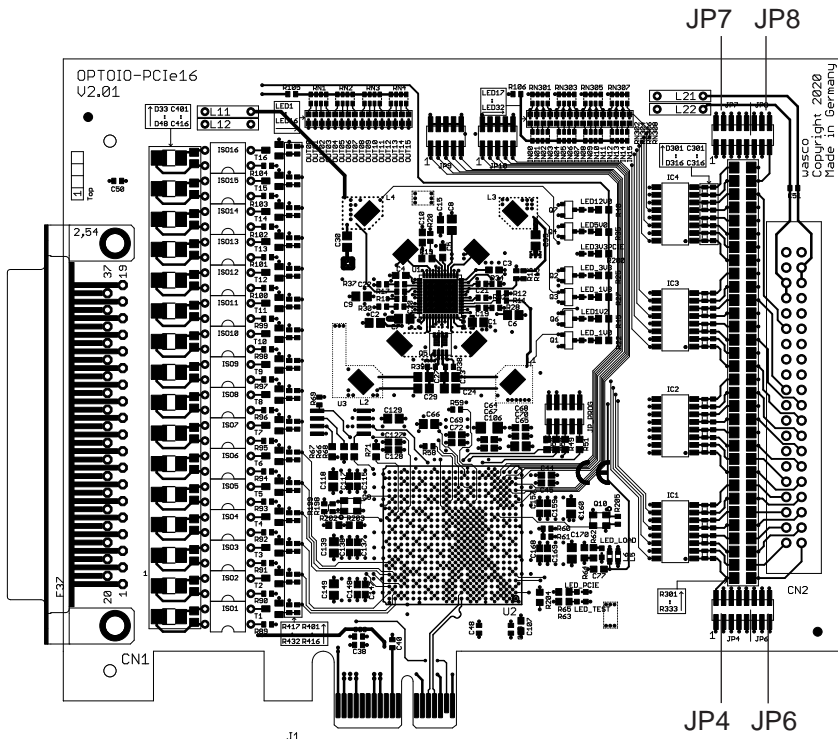


Abb. 5.1

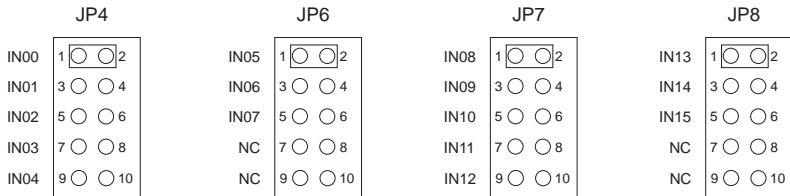
## 5.2 Eingangsspannungsbereiche

Durch das Setzen von Jumpers auf den Blöcken JP4, JP6, JP7 und JP8 kann für jeden Optokopplereingang zwischen zwei Eingangsspannungsbereichen gewählt werden.



Die Daten der zwei Eingangsspannungsbereiche entnehmen Sie bitte der folgenden Tabelle:

Jumper	LOW	HIGH
gesetzt	0...1 V	5...15 V
nicht gesetzt	0...2 V	14...30 V



Durch das Setzen eines Jumpers zwischen Pin1 und Pin2 des Jumperblocks JP4 wechselt der Eingangsspannungsbereich des IN00 von 0..2V (LOW) und 14..30V (HIGH) auf 0..1V (LOW) und 5..15V (HIGH). Die restlichen Eingangsspannungsbereiche bleiben unverändert.

### 5.3 Eingangsbeschaltung

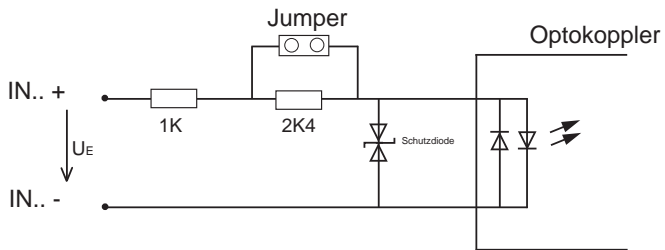


Abb. 5.3

### 5.4 Eingangsstrom

$$I_E \approx \frac{U_E - 1,1V}{3400\Omega} \quad (\text{Jumper nicht gesetzt})$$

$$I_E \approx \frac{U_E - 1,1V}{1000\Omega} \quad (\text{Jumper gesetzt})$$



## 5.5 Zugriff auf die Eingänge

Um den Zustand der Optokopplereingänge zu ermitteln, muss das Register OPTOIN ausgelesen werden. Jedes Bit des 32bit-Wertes steht für jeweils einen Eingang wie aus der Registertabelle zu entnehmen.

### Anwendungsbeispiel

Als Beispiel wird jeder dritte Optokopplereingang des Eingangssteckers auf HIGH und der Rest auf LOW gelegt. Wird das Register OPTOIN gelesen, so liefert die Karte den Wert 0x4444(hex)/17476(dec)/0b010001000100(bin) zurück. Aus diesem Wert können nun die Zustände der einzelnen Eingänge durch eine Und-Verknüpfung herausgefiltert werden.

## 5.6 Optokopplereingänge mit digitalem Filter

Jeder Optokopplereingang des Boards WASCO-PCIe8117 besitzt einen eigenen konfigurierbaren digitalen Filter, um Störungen oder Einschwingungen des Eingangssignals zu filtern. Dabei überprüft der Filter wie in Abb. 5.5 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 OPTOINFILx eingestellt. Hier kann eine Filterbreite von 0 - 65535 $\mu$ s in 1- $\mu$ s-Schritten eingestellt werden. Im Default-Zustand ist der Filter deaktiviert, sprich die Filterdauer beträgt 0 $\mu$ s.

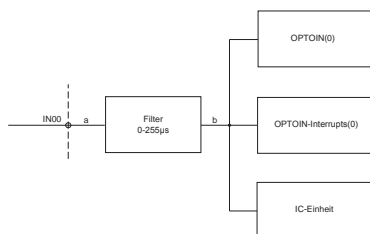


Abb. 5.4

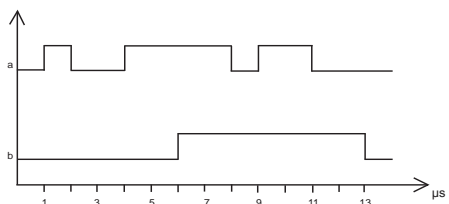


Abb. 5.5

Zu beachten ist bei der Verwendung der Filter, dass die Optokopplereingänge zum einen weit längere Schaltzeiten über  $1\mu\text{s}$  besitzen und zum anderen unterschiedlich steile Steigungs- und Fallkurven aufweisen. Diese werden bei der Filterkonfiguration aufgrund von Bauteiltoleranzen unter den Optokopplern nicht berücksichtigt und müssen vom Anwender selbst mit einberechnet werden. Werden diese Schaltzeiten nicht mit einberechnet, kann dies dazu führen, dass Signale am Optokopplereingang gefiltert werden, obwohl sie theoretisch lange genug anliegen.

### **5.7 Interruptfunktionen der Optokopplereingänge**

Um Änderungen an den Optokopplereingängen ohne regelmäßiges Abfragen der Eingänge durch den PC zu erkennen, bietet die OPTOIO-PCIe16 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.

### 5.7.1 Flankenerkennung

Um steigende Flanken an den Optokopplereingängen zu erkennen, besitzt jeder einzelne Eingang eine Flankenerkennung mit zuschaltbarer Interruptfunktion. Dafür wird ein 32bit Interruptregister (OPTOINIF) bereitgestellt, welches für jeden Eingangskanal ein Bit zur Flankenerkennung zur Verfügung stellt. Sobald an einem Eingang eine steigende Flanke durch die Karte erkannt wird, wird das jeweilige Bit im Register OPTOINIF 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 32bit Registers OPTOINIFe. Dabei repräsentiert jedes einzelne Bit einen Eingang. Wie aus der Tabelle Portadressen (Kapitel 11.1) 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 OPTOINIF zwar das entsprechende Bit gesetzt, jedoch wird dieses Bit bei der Interruptauslösung nicht berücksichtigt.

Im Defaultzustand sind alle Interruptkanäle deaktiviert.

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

## Anwendungsbeispiel:

Gewünscht wird eine Flankenerkennung mit Interruptauslösung am Kanal IN01. 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 OPTOINIF vollständig zurückgesetzt ist, da sonst eventuell sofort nach der Freigabe des Interrupts ein Interrupt ausgelöst wird. Sind nicht alle Bits im Register OPTOINIF zurückgesetzt, schreiben Sie den Wert 0xffffffff(hex) in das Register OPTOINIFr.

Aus der Tabelle ist zu entnehmen, dass zur Freischaltung des Flankenerkennungsinterruptes am Kanal IN01 Bit 1 im Register OPTOINIFe gesetzt werden muss. Mit Hilfe des PCIe-Schreibbefehls wird also der Wert 0x00000002(hex) bzw. 2(dec) in dieses Register geschrieben.

### Interrupt-Service-Routine

1. Um die Quelle des Interrupts zu ermitteln, muss das Flankenregister OPTOINIF 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 vom OPTOINIF-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 OPTOINIFr.

### 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.

### 5.7.2 Portänderungen

Müssen die Optokopplereingänge oft abgefragt werden, um Änderungen zu erkennen, so kann eine weitere Interruptfunktion verwendet werden, um den PC zu entlasten. So bietet die WASCO-PCIe8117 die Möglichkeit an, bei einer Änderung an den Eingängen einen Interrupt auszulösen. Zur Freischaltung dieser Interruptfunktion muss zum einen das Register OPTOINICe auf 0x00000001 gesetzt werden. Zum anderen kann der Anwender durch das 32bit Register OPTOINICCe bestimmen, welche Eingänge bei der Erkennung berücksichtigt werden sollen. Findet eine Änderung der Eingänge statt, so wird im Register OPTOINIC das entsprechende Bit gesetzt. Um nach einem Auslösen des Interrupts diesen wieder freizuschalten, muss das entsprechende Bit im Register OPTOINICr gesetzt werden. Nach dem Reset wird das Resetbit von selbst zurückgesetzt.

## 5.8 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																																																																																																																																																																																																																	
0x0000	OPTOIN	31:16	OPTOIN <31:16>																																																																																																																																																																																																																																
0x00A0	OPTOINFIL0	15:0	OPTOIN <15:0>																																																																																																																																																																																																																																
0x00A4	OPTOINFIL1	31:16	reserviert (*)																0x00A8	OPTOINFIL2	15:0	reserviert (*)															OPTOINFIL0 <7:0>	0x00AC	OPTOINFIL3	31:16	reserviert (*)															OPTOINFIL1 <7:0>	0x00B0	OPTOINFIL4	15:0	reserviert (*)															OPTOINFIL2 <7:0>	0x00B4	OPTOINFIL5	31:16	reserviert (*)															OPTOINFIL3 <7:0>	0x00C0	OPTOINFIL6	15:0	reserviert (*)															OPTOINFIL4 <7:0>	0x00BC	OPTOINFIL7	31:16	reserviert (*)															OPTOINFIL5 <7:0>	0x00C0	OPTOINFIL8	15:0	reserviert (*)															OPTOINFIL6 <7:0>	0x00C4	OPTOINFIL9	31:16	reserviert (*)															OPTOINFIL7 <7:0>	0x00C8	OPTOINFIL10	15:0	reserviert (*)															OPTOINFIL8 <7:0>				reserviert (*)															OPTOINFIL9 <7:0>				reserviert (*)															OPTOINFIL10 <7:0>
0x00A8	OPTOINFIL2	15:0	reserviert (*)															OPTOINFIL0 <7:0>																																																																																																																																																																																																																	
0x00AC	OPTOINFIL3	31:16	reserviert (*)															OPTOINFIL1 <7:0>																																																																																																																																																																																																																	
0x00B0	OPTOINFIL4	15:0	reserviert (*)															OPTOINFIL2 <7:0>																																																																																																																																																																																																																	
0x00B4	OPTOINFIL5	31:16	reserviert (*)															OPTOINFIL3 <7:0>																																																																																																																																																																																																																	
0x00C0	OPTOINFIL6	15:0	reserviert (*)															OPTOINFIL4 <7:0>																																																																																																																																																																																																																	
0x00BC	OPTOINFIL7	31:16	reserviert (*)															OPTOINFIL5 <7:0>																																																																																																																																																																																																																	
0x00C0	OPTOINFIL8	15:0	reserviert (*)															OPTOINFIL6 <7:0>																																																																																																																																																																																																																	
0x00C4	OPTOINFIL9	31:16	reserviert (*)															OPTOINFIL7 <7:0>																																																																																																																																																																																																																	
0x00C8	OPTOINFIL10	15:0	reserviert (*)															OPTOINFIL8 <7:0>																																																																																																																																																																																																																	
			reserviert (*)															OPTOINFIL9 <7:0>																																																																																																																																																																																																																	
			reserviert (*)															OPTOINFIL10 <7:0>																																																																																																																																																																																																																	

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
0x00CC	OPTOINFIL11	31:16 15:0	reserviert (*)															
0x00D0	OPTOINFIL12	31:16 15:0	OPTOINFIL11 <15:0>															
0x00D4	OPTOINFIL13	31:16 15:0	reserviert (*)															
0x00D8	OPTOINFIL14	31:16 15:0	OPTOINFIL12 <15:0>															
0x00DC	OPTOINFIL15	31:16 15:0	reserviert (*)															
			OPTOINFIL13 <15:0>															
			reserviert (*)															
			OPTOINFIL14 <15:0>															
			reserviert (*)															
			OPTOINFIL15 <15:0>															

(\*) reservierter Bereich ist mit 0 zu belegen

### Register OPTOIN:

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							
	OPTOIN <15:8>							
7:0	R							
	OPTOIN <7:0>							

Bit 31 - 16 undefiniert

Bit 15 - 0 **OPTOIN <15:0>**

jedes Bit entspricht einem Optokopplereingang

(Z.B. IN00 = OPTOIN<0>, IN13 = OPTOIN<13>)

Liegt an einem Eingang ein HIGH an, so steht in dem dazugehörigen Bit eine 1, ansonsten eine 0

### Register OPTOINFILx:

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							
	-							
23:16	R/W							
	-							
15:8	R/W							
	OPTOINFILx <15:8>							
7:0	R/W							
	OPTOINFILx <7:0>							

Bit 31 - 16 reserviert ( mit dem Wert 0 beschreiben)

Bit 15 - 0 **OPTOINFILx <15:0>** (default = 0)

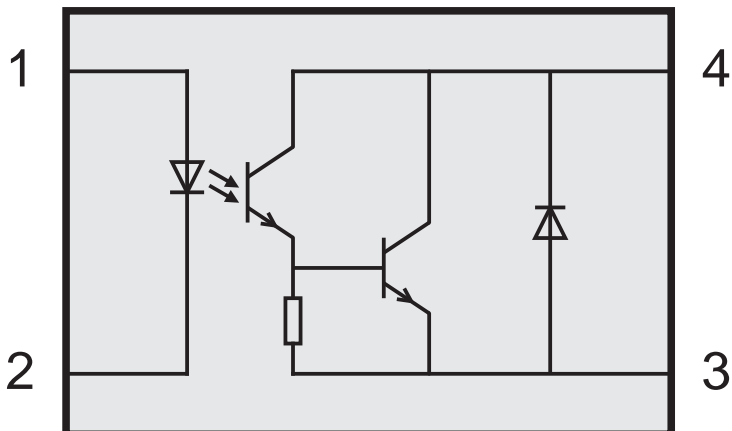
Der Wert bestimmt die Filterdauer des Filters x in  $\mu$ s



## 6. 16 Optokopplerausgänge

Die OPTOIO-PCIe16 verfügt über 16 Ausgangskanäle, deren galvanische Trennung ebenfalls mittels Optokoppler erreicht wird. Die Isolationsspannung zwischen Masse des Computers und Ausgang beträgt 500 Volt.

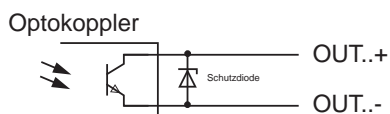
### 6.1 Pinbelegung der Ausgangsoptokoppler



### 6.2 Optokopplerdaten

Spannung-CE:	max. 50V
Spannung-EC:	0,1V
Strom-CE:	150 mA

### 6.3 Ausgangsbeschaltung



## 6.4 Funktionen der Optokopplerausgänge

### 6.4.1 Grundfunktion

Die Grundfunktion der Optokopplerausgänge ermöglicht das Sperren oder Durchschalten einzelner Ausgänge durch das Beschreiben des 32-Bit-Registers OPTOOUT. In dem Register steht, wie aus der Port-Adressen-Tabelle zu entnehmen ist, jedes einzelne Bit für einen Optokopplerausgang. Möchte man zum Beispiel jeden dritten Ausgang des Steckers durchschalten, so muss der Wert 0x4444(hex), 17476(dec) bzw. 0b0100010001000100(bin) in das Register OPTOOUT geschrieben werden.

### 6.4.2 Optokoppler mit anderen Hardwarekomponenten belegen

Neben der Grundfunktion, welche einen einfachen Zugriff auf die Optokopplerausgänge ermöglicht, können auch unterschiedliche Hardwarekomponenten wie z.B. ein PWM-Ausgang auf die einzelnen Optokopplerausgänge gelegt werden (siehe Abb. 6.1). Hierfür besitzt jeder Optokoppler einen Multiplexer mit einer 4bit-Adressierung (= bis zu 16 verschiedene Quellen). Als Default-Quelle ist nach einem Reset bzw. beim Hochfahren des PC's das OPTOOUT-Register als Peripherie festgelegt. Um die Quelle zu verändern, muss die Quelladresse (siehe Abb. 6.2) in das Register OPTOOUTMUXx geschrieben werden.

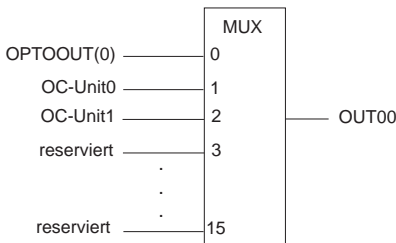


Abb. 6.1

Adresse	Peripherie
0x0 (default)	OPTOOUT(x)
0x1	OC-Unit0
0x2	OC-Unit1
0x3 - 0xF	reserviert

Abb. 6.2

## 6.5 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	
0x0008	OPTOOUT	31:16	reserviert (*)																
		15:0	OPTOOUT <15:0>																
0x03C0	OPTOOUTMUX0	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX0 [3:0]
0x03C4	OPTOOUTMUX1	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX1 [3:0]
0x03C8	OPTOOUTMUX2	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX2 [3:0]
0x03CC	OPTOOUTMUX3	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX3 [3:0]
0x03D0	OPTOOUTMUX4	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX4 [3:0]
0x03D4	OPTOOUTMUX5	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX5 [3:0]
0x03D8	OPTOOUTMUX6	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX6 [3:0]
0x03DC	OPTOOUTMUX7	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX7 [3:0]
0x03E0	OPTOOUTMUX8	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX8 [3:0]
0x03E4	OPTOOUTMUX9	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX9 [3:0]
0x03E8	OPTOOUTMUX10	31:16	reserviert (*)																
		15:0	reserviert (*)																OPTOOUTMUX10 [3:0]

(\*) 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
0x03EC	OPTOOOUTMUX11	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03F0	OPTOOOUTMUX12	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03F4	OPTOOOUTMUX13	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03F8	OPTOOOUTMUX14	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03FC	OPTOOOUTMUX15	31:16	reserviert (*)															
		15:0	reserviert (*)															

(\*) reservierter Bereich ist mit 0 zu belegen

## Register OPTOOUT:

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	R/W OPTOOUT <15:8>							
7:0	R/W OPTOOUT <7:0>							

Bit 31 - 16 reserviert ( mit dem Wert 0 beschreiben)

Bit 15 - 0 **OPTOOUT <15:0>** (default = 0)

Der Wert bestimmt den Zustand der Ausgangsoptokoppler. Dabei entspricht jedes Bit einem Optokopplerausgang.

(OPTOOUT<0> = OUT00, OPTOOUT<13> = OUT13)

Eine 1 im jeweiligen Bit verursacht ein Durchschalten des entsprechenden Optokopplers, eine 0 ein Sperren des Ausgangsoptokopplers.

## Register OPTOOUTMUXx:

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 OPTOOUTMUXx <3:0>			

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

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

Bestimmt, welche Peripherie auf den Ausgangsoptokoppler gelegt wird.

0 = OPTOOUT-Register

1 = OC-Unit0

2 = OC-Unit1

3 - 15 = reserviert

## 7. Zähler

Die WASCO-PCIe8117 stellt insgesamt 16 32-Bit-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.

### 7.1 Grundfunktion

1. Um einen Zähler zu verwenden, muss zu Beginn die Quelle ausgewählt werden. Hierfür hat jeder Zähler sein eigenes 32Bit Register (COUNTMUXxx).
2. Als nächstes muss der Counter über das Register COUNTLDxxx 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 COUNTExx 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 COUNTxx ermitteln.

### 7.2 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 COUNTIR zurückgesetzt.

### 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
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	reserviert (*)															
0x1034	COUNT13e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1038	COUNT14e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x103C	COUNT15e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1080	COUNTc0	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>															
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>															

(\*) 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
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>															
0x1180	COUNTMUX0	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1184	COUNTMUX1	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1188	COUNTMUX2	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
0x118C	COUNTMUX3	31:16	reserviert (*)															
0x1190	COUNTMUX4	15:0	COUNTMUX3 <7:0>															
0x1194	COUNTMUX5	31:16	reserviert (*)															
0x1198	COUNTMUX6	15:0	COUNTMUX4 <7:0>															
0x119C	COUNTMUX7	31:16	reserviert (*)															
0x11A0	COUNTMUX8	15:0	COUNTMUX5 <7:0>															
0x11A4	COUNTMUX9	31:16	reserviert (*)															
0x11A8	COUNTMUX10	15:0	COUNTMUX6 <7:0>															
0x11AC	COUNTMUX11	31:16	reserviert (*)															
0x11B0	COUNTMUX12	15:0	COUNTMUX7 <7:0>															
0x11B4	COUNTMUX13	31:16	reserviert (*)															
0x11B8	COUNTMUX14	15:0	COUNTMUX8 <7:0>															
0x11BC	COUNTMUX15	31:16	reserviert (*)															
		15:0	COUNTMUX9 <7:0>															
		15:0	COUNTMUX10 <7:0>															
		15:0	COUNTMUX11 <7:0>															
		15:0	COUNTMUX12 <7:0>															
		15:0	COUNTMUX13 <7:0>															
		15:0	COUNTMUX14 <7:0>															
		15:0	COUNTMUX15 <7:0>															

(\*) reservierter Bereich ist mit 0 zu belegen

## Register COUNTxe:

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

## Register COUNTx:

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 <315:8>							
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:

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 = Optokopplereingang IN00 (default)

1 = Optokopplereingang IN01

.

.

.

15 = Optokopplereingang IN15

255 - 16 = reserviert -> mit 0 belegen

## 8. Timer

Die zur Verfügung stehenden 32Bit-Timer lassen sich als Zeitgeber oder zur konfigurierbaren Intervall-Interruptauslösung verwenden. Dabei können Intervalle zwischen 0 und 4294967295  $\mu\text{s}$  in 1- $\mu\text{s}$ -Schritten eingestellt werden.

### 8.1 Anwendung Intervall-Interruptauslösung

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

## 8.2 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

## 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

## 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



## 9. 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 1 $\mu$ s-Schritten und kann durch Programmierung jedem digitalen Eingang zugeordnet werden.

### 9.1 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.

#### 9.1.1 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 ICUNITex 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 ICUNITex.

**Achtung:** Beachten Sie die unterschiedlichen Schaltverzögerungen bei der Verwendung der Optokopplereingänge. Diese verändern die Pulsbreite.

### **9.1.2 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.

## 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
0x14C0	ICUNIT0e	31:16	reserviert (*)													en		
0x14C4	ICUNIT1e	15:0	reserviert (*)															
0x14E0	ICCONFIG0	31:16	reserviert (*)													en		
0x14E4	ICCONFIG1	15:0	reserviert (*)															
0x1500	ICMUX0	31:16	reserviert (*)													ICMUX0 <7:0>		
0x1504	ICMUX1	15:0	reserviert (*)															
0x1540	ICPULS0	31:16	ICPULS0 <31:16>													ICMUX1 <7:0>		
0x1544	ICPULS1	15:0	ICPULS1 <31:16>															
0x1560	ICPERIOD0	31:16	ICPERIOD0 <31:16>													ICMUX1 <7:0>		
0x1504	ICPERIOD0	15:0	ICPERIOD0 <15:0>															
0x1504	ICPERIOD1	31:16	ICPERIOD1 <31:16>													ICMUX1 <7:0>		
0x1504	ICPERIOD1	15:0	ICPERIOD1 <15:0>															

(\*) reservierter Bereich ist mit 0 zu belegen

## Register ICUNITx:

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:

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:

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 = Optokopplereingang IN00 (default)

1 = Optokopplereingang IN01

.

.

.

15 = Optokopplereingang IN15

255 - 16 = reserviert -> mit 0 belegen

### Register ICPULSx:

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  $\mu$ s ausgelesen werden

### Register ICPERIODx:

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  $\mu$ s ausgelesen werden

## 10. Output-Compare-Einheit

Mit der Output-Compare-Einheit (OC-Unit) ermöglicht die WASCO-PCIe8117-Karte dem Anwender, an die Optokopplerausgänge eine PWM-Funktion zu koppeln oder einzelne Pulse auszugeben. Dabei können Rechtecksignale mit einer Periodendauer von 2 bis  $2^{32}$   $\mu\text{s}$  und einer Pulsdauer von 1 bis  $2^{32}$   $\mu\text{s}$  erzeugt werden.

Achtung: Während die OC-Einheit eine Auflösung im Mikrosekundenbereich ermöglicht, sind aufgrund der Ausgangsoptokopplerspezifikationen nur Staffelungen im Millisekundenbereich sinnvoll.

### 10.1 PWM

Die OC-Einheit ermöglicht es dem Anwender, an jedem beliebigen Optokopplerausgang eine PWM anzulegen.

#### 10.1.1 Funktionsweise

Um die PWM zu realisieren, verwendet die OC-Einheit einen beschreibbaren 32Bit-Timer mit einstellbarer Periodendauer in  $\mu\text{s}$  (OCPERIODx) und ein Zwei-Stufen-Compare-Register (OCUNITORx) zur Einstellung der Pulsdauer in  $\mu\text{s}$ . Ist die OC-Einheit deaktiviert, so liegt am Ausgang ein LOW an (Optokopplerausgang gesperrt). Wird die OC-Einheit im PWM-Modus gestartet, beginnt der Timer im  $\mu\text{s}$ -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).

### 10.1.2 Berechnung der Registerwerte

$OCPERIODx = \text{Periodendauer\_in\_}\mu\text{s} + 1 \text{ [}\mu\text{s]}$

$OCUNITORx = \text{Pulsdauer\_in\_}\mu\text{s} \text{ [}\mu\text{s]}$

$OCTIMER = \text{Takte [}\mu\text{s]}$

### 10.1.3 Anwendungsbeispiel

1. OC-Einheit durch das Löschen des entsprechenden Bits im Register OCUNITxe deaktivieren.
2. OC-Einheit mit dem gewünschten Optokopplerausgang verbinden. Dafür muss in dem zum Optokoppler gehörigen OPTOOUTMUXx-Register die Quelle ausgewählt werden (siehe Kapitel Optokopplerausgangsmultiplexer).
3. Den OC-Timer der OC-Einheit vorladen. Hier wird in der Regel der Wert 0x00000000 in das Register OCTIMERx geschrieben.
4. Die Periodendauer der PWM festlegen. Hierfür muss die Periodendauer wie folgt in das Register OCUNITORx geschrieben werden:  
 $OCPERIODx = \text{Periodendauer} - 1 \text{ [}\mu\text{s]}$
5. Die Pulsdauer festlegen. Hierfür muss die Pulsdauer wie folgt in das Register OCPULSx geschrieben werden:  
 $OCPULSx = \text{Pulsdauer [}\mu\text{s]}$
6. Den OC-Unit-Modus auswählen. Für das Verwenden der PWM muss der Wert 0 im Bereich Mode des Registers OCONFIGx geschrieben werden.
7. OC-Einheit durch das Setzen des entsprechenden Bits im Register OCUNITe aktivieren.



## 10.2 Pulsausgabe

Die OC-Einheit ermöglicht es dem Anwender neben der PWM auch einzelne  $\mu$ s-genaue Pulse an den Optokopplerausgängen auszugeben.

### 10.2.1 Funktionsweise

Für die Ausgabe einzelner positiver Pulse (= Optokoppler leitet) 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.

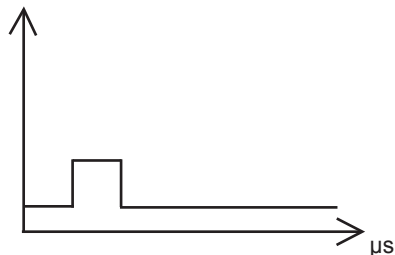


Abb. 10.1

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

### 10.2.2 Berechnung der Registerwerte

$\text{OCPERIODx} = \text{Pulsdauer\_in\_}\mu\text{s}$

### 10.2.3 Anwendungsbeispiel

#### 1. OC-Einheit konfigurieren

- a) Deaktivieren Sie die Einheit durch Löschen (= 0) des en-Bits im Register OCUNITx<sub>e</sub>
- b) OC-Einheit mit dem gewünschten Optokopplerausgang verbinden. Dafür muss in dem zum Optokoppler gehörigen OPTOOUTMUXx-Register die Quelle ausgewählt werden (siehe Kapitel Optokopplerausgangsmultiplexer).
- c) Laden Sie den OC-Timer mit dem Wert 0 vor, indem Sie das Register OCTIMERx mit 0x00000000 beschreiben
- d) Laden Sie die gewünschte Pulsdauer in das Register OCPERIODx (siehe auch Kapitel 10.2.2 Berechnung der Registerwerte)
- e) Laden Sie das Register OCUNITORx mit dem Wert 1
- f) Wählen Sie den Single-Puls-Modus aus, indem Sie den Wert 1 in den Mode-Bereich im Register OCCONFIGx schreiben

#### 2. Puls ausgeben

- a) OC-Einheit durch Löschen des Enable-Bits im Register OCUNITx<sub>e</sub> deaktivieren
- b) OC-Einheit durch Setzen des Enable-Bits im Register OCUNITx<sub>e</sub> aktivieren. Daraufhin wird am ausgewählten Ausgang der Puls angelegt.

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	reserviert (*)															
0x15C4	OCUNIT1e	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1600	OCTIMER0	31:16	OCTIMER0 <31:16>															
		15:0	OCTIMER0 <15:0>															
0x1604	OCTIMER1	31:16	OCTIMER1 <31:16>															
		15:0	OCTIMER1 <15:0>															
0x1620	OCUNITOR0	31:16	OCUNITOR0 <31:16>															
		15:0	OCUNITOR0 <15:0>															
0x1624	OCUNITOR1	31:16	OCUNITOR1 <31:16>															
		15:0	OCUNITOR1 <15:0>															
0x1660	OCPERIOD0	31:16	OCPERIOD0 <31:16>															
		15:0	OCPERIOD0 <15:7>															
0x1664	OCPERIOD1	31:16	OCPERIOD1 <31:16>															
		15:0	OCPERIOD1 <15:7>															
0x1680	OCCONFIG0	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x1684	OCCONFIG1	31:16	reserviert (*)															
		15:0	reserviert (*)															

(\*) reservierter Bereich ist mit 0 zu belegen

## Register OCUNITxe:

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

## Register OCTIMERx:

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:

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  $\mu\text{s}$   
 Pulsdauer = OCUNITORx [ $\mu\text{s}$ ]

## Register OCPERIODx:

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  $\mu\text{s}$   
 Periodendauer = OCPERIODx + 1 [ $\mu\text{s}$ ]

## Register OCONFIGx:

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)

# 11. 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 32-Bit-Register INTCON die Zentraleinheit dar. Hier werden alle möglichen Interruptquellen (teilweise schon aufbereitet) zusammengeführt.

Wird ein Interrupt z.B. durch eine Flanke an einem Optokopplereingang ausgelöst, wird dieser an das erste 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 INTTr gelöscht werden und ein weiterer Interrupt kann an den PC geschickt werden.

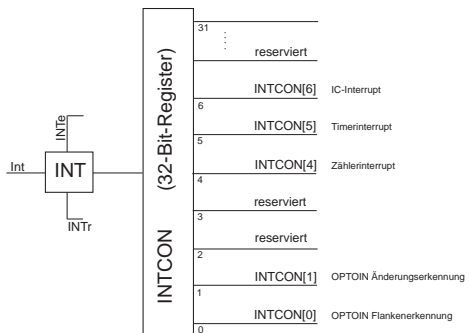


Abb. 11.1

Da z.B. bei einem Optokopplereingangsflankeninterrupt mehrere Quellen in Frage kommen, werden die Quelleitungen zum Register INTCON meist aufbereitet. Dies bedeutet, dass ein weiteres 32-Bit-Register an dem jeweiligen Bit des INTCON-Registers anliegen kann. Im Falle der Flankenerkennung der Optokopplereingänge ist dies das Register OPTOINF. In dem Register stellt jedes Bit einen Optokopplereingang dar (siehe Registerbeschreibung). Die einzelnen Eingänge können einzeln als Interruptquelle freigegeben werden (OPTOINFe) und nach einem ausgelösten und bearbeiteten Interrupt wieder freigegeben werden (OPTOINFr). Ist dies geschehen, geht das jeweilige Bit im Register INTCON automatisch auf 0.

## Anwendung

### 1. Konfiguration

- a) Überprüfen, ob alle Interruptquellen gelöscht sind ( $\text{INTCON} = 0$ )
- b) Einzelne Interruptquellen freigeben (siehe Dokumentation der entsprechenden Peripherie)
- c) Interruptfunktion freischalten ( $\text{INTe} = 1$ )

### 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 ( $\text{INTCON} = 0?$ )
- d) falls c) zutrifft, alle anderen Interrupts ebenfalls löschen
- e) Interruptfunktion wieder freigeben ( $\text{INTr} = 1$ )



## 11.1 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 (*)													en							
0x0284	INTr	31:16	reserviert (*)													re							
0x0288	INTCON	31:16	reserviert (*)													INT7		INT6	INT5	INT4	reserviert (*)	INT1	INT0
0x028C	OPTONIFe	15:0	reserviert (*)																				
0x0294	OPTONIFr	15:0	OPTONIFe <15:0>																				
0x029C	OPTONIF	15:0	reserviert (*)																				
0x02A8	OPTONICe	15:0	OPTONIFr <15:0>																				
0x02AC	OPTONICCe	15:0	reserviert (*)																				
0x02B4	OPTONICr	15:0	OPTONICe <15:0>																				
0x02B8	OPTONIC	31:16	reserviert (*)													reset							
0x0340	COUNTRe	15:0	reserviert (*)													reserviert (*)		OPTONICa					
0x0344	COUNTRr	15:0	reserviert (*)													COUNTRe <15:0>		reserviert (*)		COUNTRr <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
0x0348	COUNTIR	31:16	reserviert (*)															
0x0360	TIMERIRe	15:0	COUNTIR<15:0>															
0x0364	TIMERIRr	31:16	reserviert (*)															
0x0368	TIMERIR	15:0	reserviert (*)															
0x036C	ICUNITIRe	31:16	reserviert (*)															
0x0370	ICUNITIRr	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			U		R	
	reserviert	INTCON <6:4>			reserviert		INTCON <1:0>	

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 - 2 reserviert ( mit dem Wert 0 belegt)

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

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

## Register OPTOINIFe

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/W							
	OPTOINIFe <15:8>							
7:0	R/W							
	OPTOINIFe <7:0>							

Bit 31 - 16 undefiniert

Bit 15 - 0 **OPTOINIFe<15:0>** In dem Registerabschnitt können einzelne Optokopplereingänge als Quelle freigeschaltet werden, um bei einer anliegenden positiven Flanke einen Interrupt auszulösen. Jedes Bit entspricht einem Optokopplereingang (z.B. IN00 => OPTOINIFe<0>, IN13 => OPTOINIFe<13>). Eine 1 im Bit schaltet die Flankeninterruptfunktion des Optokopplereingangs frei, eine 0 sperrt diese.

## Register OPTOINIFr

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	W							
	OPTOINIFr <15:8>							
7:0	W							
	OPTOINIFr <7:0>							

Bit 31 - 16 undefiniert

Bit 15 - 0 **OPTOINIFr<15:0>** Jedes Bit entspricht einem Optokopplereingang (z.B. IN00 => OPTOINIFr<0>, IN13 => OPTOINIFr<13>). Wurde an einem Optokopplereingang ein Flankeninterrupt ausgelöst, muss dessen Signalbit im Register OPTOINIF zurückgesetzt werden. Dies erfolgt durch das Setzen (= 1) des entsprechenden OPTOINIFr-Bits. Die OPTOINIFr-Bits werden nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

## Register OPTOINIF

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							
	OPTOINIF <15:8>							
7:0	R							
	OPTOINIF <7:0>							

Bit 31 - 16 undefiniert

Bit 15 - 0 **OPTOINIF<15:0>** zeigt, ob an einem der Optokopplereingänge eine steigende Flanke angelegen ist. Jedes Bit entspricht einem Optokopplereingang (z.B. IN00 => OPTOINIF<0>, IN13 => OPTOINIF<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 OPTOINICe:

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 **OPTOINICe<0>** (default = 0) Interruptfunktion zur Erkennung von Änderungen an den Optokopplereingängen freischalten  
 0 = Interrupt gesperrt (default)  
 1 = Interrupt freigeben

## Register OPTOINICc:

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	R/W							
	OPTOINICc<15:8>							
7:0	R/W							
	OPTOINICc<7:0>							

Bit 31 - 15 reserviert ( mit dem Wert 0 beschreiben)

Bit 15 - 0 **OPTOINICc<15:0>** (default = 0)

Einzelne Optokopplereingänge für Interruptfunktion zur Erkennung von Änderungen an den Optokopplereingängen freischalten bzw. sperren. Jedes Bit entspricht einem Optokopplereingang (z.B. IN00 => OPTOINICc<0>, IN13 => OPTOINICc<13>)

0 = Interrupt gesperrt (default)

1 = Interrupt freigeben

## Register OPTOINICr:

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 - 15 reserviert ( mit dem Wert 0 beschreiben)

Bit 15 - 0 **OPTOINICr<0>** (default = 0)

Wurde ein Interrupt durch eine Änderung an den Optokopplereingängen ausgelöst, muss das Quellenregister OPTOINIC zurück auf 0 gesetzt werden. Dies erfolgt durch das Setzen (=1) des OPTOINICr-Bits. Das OPTOINICr-Bit wird nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

## Register OPTOINIC:

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	
								OPTOINIC<0>	

Bit 31 - 1 undefiniert

Bit 0 **OPTOINIC<0>** zeigt an, ob eine Änderung an einem freigegebenen Optokopplereingang stattgefunden hat.  
 0 = keine Änderung  
 1 = Änderung an einem freigegebenen Optokopplereingang

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

Bit 31 - 16 reserviert ( mit dem Wert 0 beschreiben)

Bit 15 - 0 **COUNTIRE<15:0>** (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							
	reserviert							
23:16	U							
	reserviert							
15:8	R/W							
	COUNTIRr<15:8>							
7:0	R/W							
	COUNTIRr<7:0>							

Bit 31 - 16 reserviert ( mit dem Wert 0 beschreiben)

Bit 15 - 0 **COUNTIRr<15:0>**

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

Bit 31 - 16 reserviert ( mit dem Wert 0 beschreiben)

Bit 15 - 0 **COUNTIR<15:0>** 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>** 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>** 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.

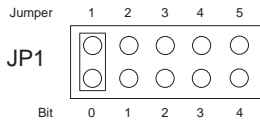
### Register ICUNITIR:

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

## 12. 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

## 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	
0xOFF8	BOARDID	31:16	reserviert (*)										reserviert (*)						Board ID
		15:0	reserviert (*)																

(\*) reservierter Bereich ist mit 0 zu belegen

## 13. Programmierung unter Windows<sup>®</sup>

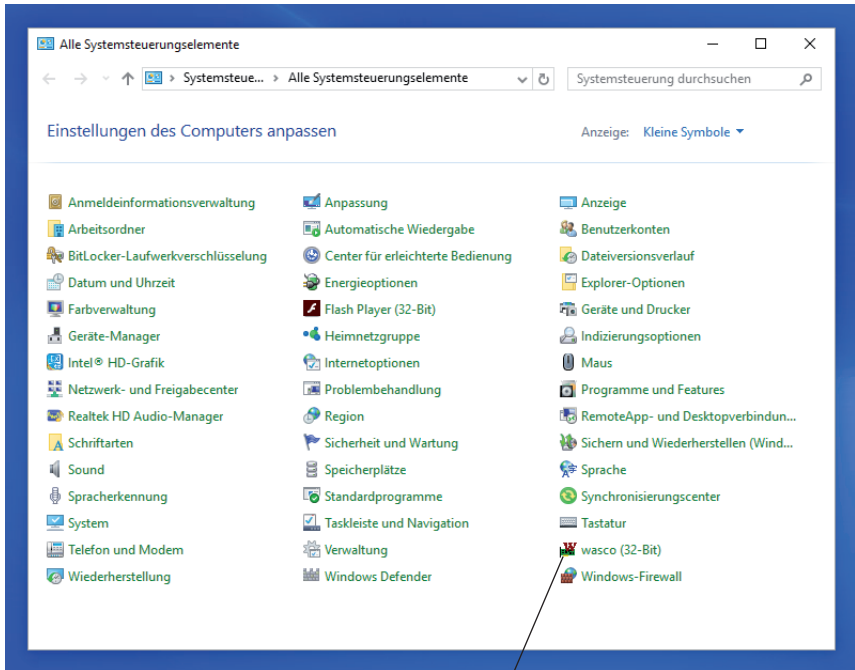
### 13.1 Installation des Windows<sup>®</sup> Treibers

Für die Anwendung der Karte unter Windows<sup>®</sup> ist es notwendig, einen speziellen Treiber zu installieren, der den Zugriff auf die Karte ermöglicht. Unter Windows<sup>®</sup> 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.

### 13.2 Installation der Windows<sup>®</sup> Entwicklungsdateien

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

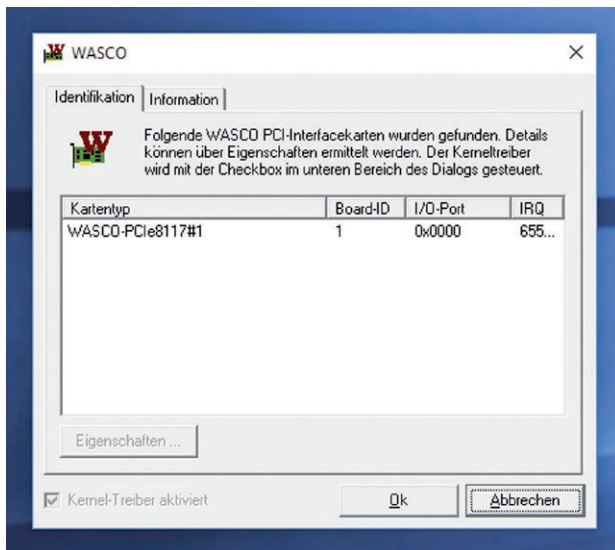




Wurden die Treiber- und Entwicklungsdateien vollständig installiert, finden Sie in der Systemsteuerung Ihres Rechners ein Icon zur Lokalisierung aller im System vorhandenen **wasco®** PCI- und PCIe-Karten.

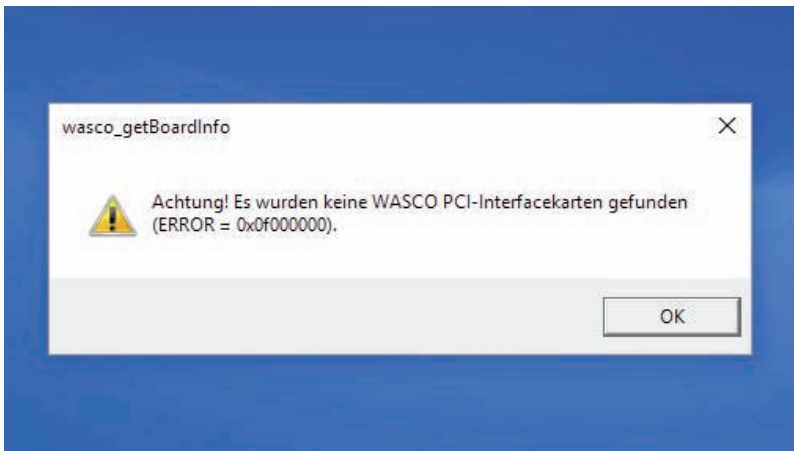


Starten Sie die Kartenabfrage durch einen Doppelklick auf das "wasco®"-Icon. Folgendes Fenster erscheint: (Als Beispiel wurde hier eine OPTOIO-PCIe16ULTRA verwendet)!



Wurde Ihre Karte im System erkannt, wird der Boardname WASCO-PCIe8117, 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!

### 13.3 Programmierung der OPTOIO-PCIe16 mit **wasco**<sup>®</sup>-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 OPTOIO-PCIe16, befinden sich auf der beiliegenden CD sowie auf unserer Homepage.

Die Programmierung der Hardwarekomponenten der OPTOIO-PCIe16 erfolgt durch den Zugriff auf Memory Mapped I/O-Adressen, die sich abhängig von der vom BIOS des Systems für die OPTOIO-PCIe16 vergebenen Basisadresse ergeben. Eine genauere Beschreibung zur Programmierung befindet sich in der Treiberdokumentation.

### 13.4 Zugriff auf die Karte OPTOIO-PCIe16<sup>ULTRA</sup>

Der Zugriff auf die OPTOIO-PCIe16<sup>ULTRA</sup> erfolgt ausschließlich über den Boardnamen (Kartentyp) WASCO-PCIe8117

### 13.5 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:

<b>Port/Register</b>	<b>BA + Offset</b>	<b>RD/WR</b>
Optokoppler-Eingangsport (IN00...IN15)	BA + \$0	RD
Optokoppler-Ausgangsport (OUT00...OUT15)	BA + \$8	RD/WR
Board Identifikation	BA+ \$FF8	RD

## 14. Programmierung unter Linux<sup>®</sup>

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

### 14.1 Installation des Linux<sup>®</sup> Treibers

Unter Linux<sup>®</sup> 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

### 14.2 Unterstützte Linux-Distributionen/Kernelversionen

Der wasco<sup>®</sup>-Treiber wurde in folgenden Umgebungen getestet:

Ubuntu<sup>®</sup> 18.04.4 LTS (Kernel: 5.3.0)

### 14.3 Programmierung der OPTOIO-PCIe16 mit **wasco**<sup>®</sup>-Treiber

Die Programmierung der Hardwarekomponenten der OPTOIO-PCIe16 erfolgt durch den Zugriff auf Memory Mapped I/O-Adressen, die sich abhängig von der vom BIOS des Systems für die OPTOIO-PCIe16 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 OPTOIO-PCIe16 befinden sich auf der beiliegenden CD sowie auf unserer Homepage.

#### 14.4 Zugriff auf die Karte OPTOIO-PCle16<sup>ULTRA</sup>

Der Zugriff auf die OPTOIO-PCle16<sup>ULTRA</sup> erfolgt ausschließlich über den Boardnamen (Kartentyp) WASCO-PCle8117.

#### 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 folgt:

32-Bit-Modus

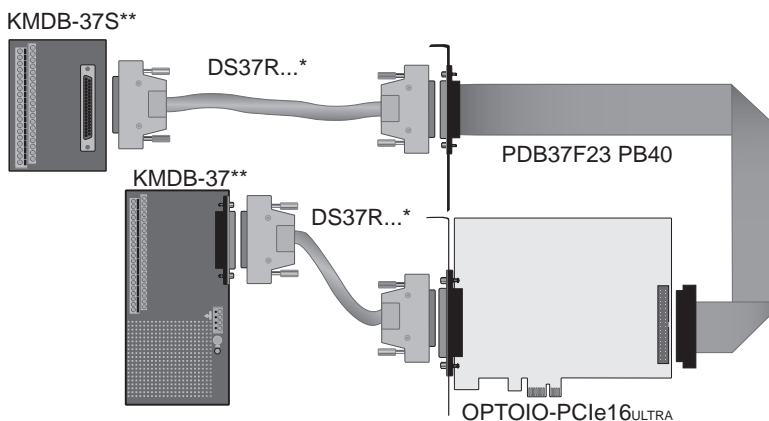
Port/Register	BA + Offset	RDWR
Optokoppler-Eingangsport A (IN00...IN07)	BA + \$0	RD
Optokoppler-Eingangsport B (IN08...IN15)	BA + \$4	RD
Optokoppler-Ausgangsport A (OUT00...OUT07)	BA + \$8	WR
Optokoppler-Ausgangsport B (OUT08...OUT15)	BA + \$C	WR
Board Identifikation	BA+ \$3E0	RD

## 15. Zubehör

### 15.1 Passendes **wasco**<sup>®</sup>-Zubehör

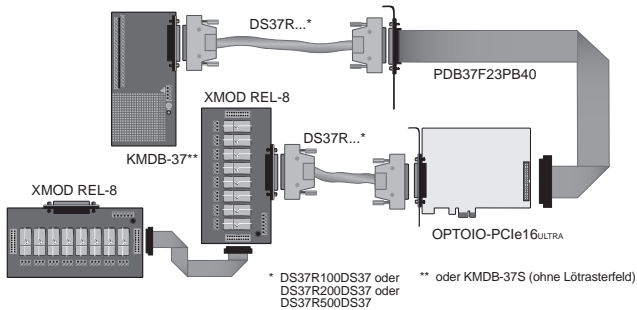
Anschlusssteile	EDV-Nr.
PDB37F23PB40 Steckerverlegungsset	A-497500
DS37R100DS37 Verbindungsleitung	A-202200
DS37R200DS37 Verbindungsleitung	A-202400
DS37R500DS37 Verbindungsleitung	A-202800
KMDB-37S Klemm-Modul (ohne Lötrasterfeld)	A-204910
KMDB-37 Klemm-Modul	A-2046
XMOD REL-4 Relais-Modul	A-3264
XMOD REL-8 Relais-Modul	A-3268

### 15.2 Anschlusstechnik (Anwendungsbeispiele)



\* DS37R100DS37 oder DS37R200DS37  
oder DS37R500DS37

\*\* alternativ KMDB-37 mit oder ohne Lötrasterfeld



### 15.3 Einzelkomponenten zur Eigenkonfektionierung

Anschlusssteile	EDV-Nr.
D-Sub-Stecker 37pol. für Lötanschluss	A-5506
D-Sub-Haube 37pol. Stecker (Lötanschluss)	A-5586
D-Sub-Stecker 37pol. für Flachbandleitung	A-5526
D-Sub-Buchse 37pol. für Flachbandleitung	A-5566
Slotblech mit Ausschnitt für 37pol. Stecker/Buchse	A-5754
Pfostenbuchse 40pol. für Flachbandleitung	A-5642
Flachbandleitung 37pol.	A-5718
Flachbandleitung 40pol.	A-5720

## 16. Fehlersuche

Nachfolgend finden Sie eine kurze Zusammenstellung der häufigsten bekannten Fehlerursachen, die während der Inbetriebnahme oder während der Arbeit mit der OPTOIO-PCIe16 auftauchen können.

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

1. Sitzt die OPTOIO-PCIe16 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**<sup>®</sup> Treibers installiert?  
Updates finden Sie unter: <http://www.messcomp.com>



## 17. Technische Daten

### Digitale Eingänge über Optokoppler

Optokoppler

16 Kanäle, galvanisch getrennt

Galvanische Trennung auch zwischen den einzelnen Kanälen mit zwei separaten Anschlüssen für jeden Kanal

Überspannungsschutz durch Schutzdioden

Zwei Eingangsspannungsbereiche durch Jumper wählbar:

Bereich 1      HIGH = 14..30 Volt

LOW = 0..2 Volt

Bereich 2:     HIGH = 5..15 Volt

LOW = 0..1 Volt

Eingangsfrequenz: max. 10 kHz

### Digitale Ausgänge über Optokoppler

Optokoppler

16 Kanäle, galvanisch entkoppelt, gesockelt

Galvanische Trennung auch zwischen den einzelnen Kanälen mit zwei separaten Anschlüssen für jeden Kanal

Überspannungsschutz durch Schutzdioden

Ausgangsstrom: max. 150 mA

Ausgangsfrequenz ca 1 KHz

Spannung-CE: max. 50 V

Spannung-EC: max. 0,1 V

### Board-Identifikation

5-fach Jumperblock

### Anschlussstecker

1 \* 37polige D-Sub-Buchse

1 \* 40poliger Pfostenstecker

### Bussystem

32 Bit PCIe-Bus (8 Bit Datenzugriff)

### Abmessungen der Platine

129 mm x 111 mm (l x b)

standard height, half length card

Multilayer-Platine

### Sonstiges

Kontroll-LEDs für Spannungsversorgung

## 18. 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.

## 19. EG-Konformitätserklärung

Für das folgende mit CE-Kennzeichen gekennzeichnete Erzeugnis

**OPTOIO-PCle16<sup>ULTRA</sup>**  
**EDV-Nummer A-829410**

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, 19.07.2017



## 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:	OPTOIO-PCIe16 <sup>ULTRA</sup>	A-829410