

ADIODA-PCle22

ISO16

EDV-Nr.: A-812230

16 A/D-Eingänge 22 Bit (single-ended) isoliert oder
8 A/D-Eingänge 22 Bit (differentiell) isoliert
8 Eingänge über Optokoppler
8 Ausgänge über Optokoppler
2*32-Bit-Zähler
Timer
OC- und IC-Einheiten
Board-Identifikation

wasco[®]

Handbuch

Copyright® 2024 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 ADIODA-PCle22 und dieser Dokumentation direkt oder indirekt entstandene Schäden.

Inhaltsverzeichnis

1. Produktbeschreibung	6
2. Installation der ADIODA-PCIe22_{ISO16}.....	7
2.1 Installation der Karte in den Rechner	7
3. Anschlussstecker	8
3.1 Lage der Anschlussstecker auf der Platine	8
3.2 Steckerbelegung von CN1.....	9
3.3 Steckerbelegung von CN2.....	10
3.4 Steckerbelegung von CN2 auf D-Sub 37 (Steckerverlegungsset)	11
4. Systemkomponenten	12
4.1 Blockschaltbild.....	12
4.2 Zugriff auf die Systemkomponenten.....	13
5. 16 A/D-Eingänge 22 Bit	14
5.1 Kanalkonfigurationstabelle	14
5.2 Konfiguration der Abtastrate	21
5.3 Konfiguration der Messauflösung und der Messdauer	21
5.4 Messmodus	23
5.5 FIFO	24
5.6 Transfer Function.....	24
5.7 Messbeispiele.....	25
5.8 ADC-Reset.....	29
5.9 Portadressen	30
6. 8 Optokopplereingänge	39
6.1 Pinbelegung der Eingangsoptokoppler.....	39
6.2 Eingangsspannungsbereiche	40
6.3 Eingangsbeschaltung	42
6.4 Eingangsstrom.....	42
6.5 Zugriff auf die Eingänge	43
6.6 Optokopplereingänge mit digitalem Filter	43
6.7 Interruptfunktionen der Optokopplereingänge	44
6.8 Portadressen	48

7. 8 Optokopplerausgänge	50
7.1 Pinbelegung der Ausgangsoptokoppler	50
7.2 Optokopplerdaten	50
7.3 Ausgangsbeschaltung	50
7.4 Funktionen der Optokopplerausgänge	51
7.5 Portadressen	52
8. Zähler	54
8.1 Grundfunktion	54
8.2 Interruptfunktion	54
8.3 Portadressen	55
9. Timer	60
9.1 Anwendung Interval-Interruptauslösung	60
9.2 Portadressen	61
10. Input-Capture-Einheit	64
10.1 Kontinuierliche Messung periodischer Signale	64
10.2 Portadressen	66
11. Output-Compare-Einheit	70
11.1 PWM	70
11.2 Pulsausgabe	72
12. Interruptcontroller	78
12.1 Portadressen	80
13. Board-Identifikation	94
13.1 Portadressen	95
14. Programmierung unter Windows[®]	96
14.1 Installation des Windows [®] Treibers	96
14.2 Installation der Windows [®] Entwicklungsdateien	96
14.3 Programmierung der ADIODA-PCIe22 mit wasco [®] -Treiber	99
14.4 Zugriff auf die Karte ADIODA-PCIe22 ^{ULTRA}	100
14.5 Zuordnung der Memory Mapped I/O-Adressen	100

15. Programmierung unter Linux[®]	101
15.1 Installation des Linux [®] Treibers	101
15.2 Unterstützte Linux-Distributionen/Kernelversionen	101
15.3 Programmierung der ADIODA-PCle22 mit wasco [®] -Treiber.....	101
15.4 Zugriff auf die Karte ADIODA-PCle22 ^{ULTRA}	102
15.5 Zuordnung der Memory Mapped I/O-Adressen.....	102
16. Zubehör	103
16.1 Passendes wasco [®] -Zubehör	103
16.2 Anschlusstechnik (Anwendungsbeispiele).....	104
16.2 Anschlusstechnik (Anwendungsbeispiele).....	104
16.3 Einzelkomponenten zur Eigenkonfektionierung	105
17. Fehlersuche	106
18. Technische Daten	107
19. Beschaltungsbeispiele	109
19.1 Beschaltung der A/D-Eingänge single-ended.....	109
19.2 Beschaltung der A/D-Eingänge differentiell	110
20. Produkthaftungsgesetz	111
21. EG-Konformitätserklärung	113

1. Produktbeschreibung

Die ADIODA-PCle22_{ISO16} (Boardname: WASCO-PCle8368) ist eine Multifunktionskarte mit galvanisch getrennten analogen Eingängen. Sie verfügt über 16 gemultiplexte single-ended oder 8 differentielle A/D-Eingangskanäle mit galvanischer Trennung, einem bipolaren Eingangsspannungsbereich von +/-10V und einer Auflösung bis zu 22 Bit. Ein 8K-FIFO (8192 Messwerte) ermöglicht dem Anwender ein zeitverzögertes Auslesen der Messwerte. Zusätzlich verfügt die Karte über jeweils 8 digitale, galvanisch getrennte Eingänge sowie Ausgänge. Die Potentialtrennung wird bei den Ein- und Ausgängen durch hochwertige Optokoppler gewährleistet. Spezielle, leistungsfähige Ausgangsoptokoppler bewältigen einen Schaltstrom von bis zu 150 mA. Jeder digitale Ein- und Ausgang ist zusätzlich durch Schutzdioden gegen schädliche Spannungsspitzen und Impulse geschützt. Über Jumperblöcke sind zwei verschiedene Eingangs-Spannungsbereiche einstellbar. Jedem digitalen 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 sämtliche Optokopplereingänge, Zähler, IC-Einheiten sowie dem A/D-FIFO möglich. Zudem ist eine Interruptauslösung zeitabhängig durch zwei 32-Bit-Timer möglich.

Die Anschlüsse der A/D-Eingänge sind der am Slotblech der Platine montierten 68-poligen SCSI-II-Buchse zugeführt. Die Optokoppler-Ausgänge sowie Optokoppler-Eingänge liegen am 40-poligen Pfostenstecker auf der Platine an. Über ein als Option erhältliches Steckerverlegungs-Set ist die Verlegung auf eine 37-polige D-Sub-Buchse mit Slotblech möglich.

Des Weiteren besitzt die Karte einen Board-Identifikations-Jumperblock, um mehrere identische Karten im PC per Software unterscheiden zu können.

2. Installation der ADIODA-PCle22_{ISO16}

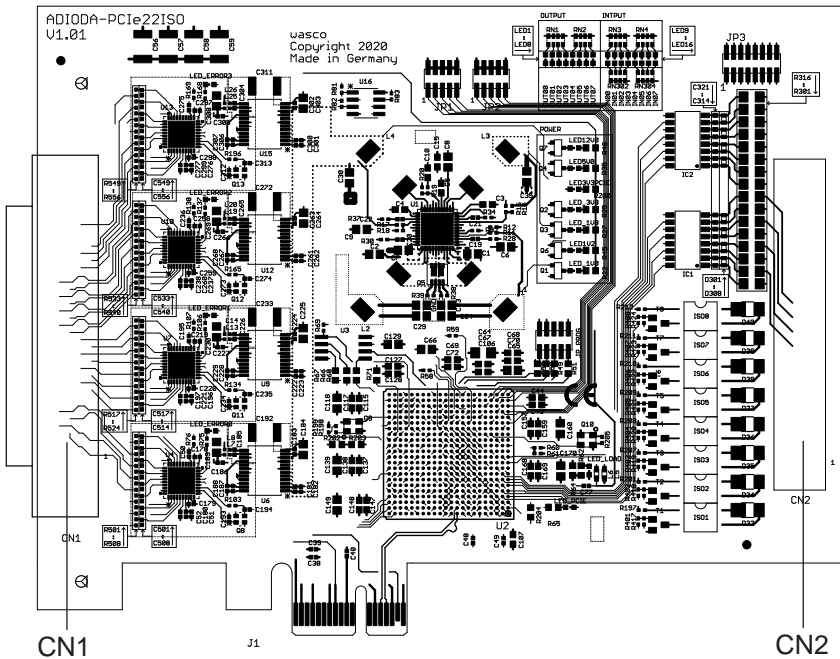
2.1 Installation der Karte in den Rechner

Achten Sie vor dem Einbau der ADIODA-PCle22 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 ADIODA-PCle22 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: A/D-Eingangskanäle AIN00...AIN15
- CN2: Optokoppler-Ausgänge OUT00...OUT07
Optokoppler-Eingänge IN00...IN07

3.2 Steckerbelegung von CN1

NC	68	□	□	34	NC
NC	67	□	□	33	NC
NC	66	□	□	32	NC
NC	65	□	□	31	NC
NC	64	□	□	30	NC
NC	63	□	□	29	NC
NC	62	□	□	28	NC
NC	61	□	□	27	NC
NC	60	□	□	26	NC
NC	59	□	□	25	NC
NC	58	□	□	24	NC
NC	57	□	□	23	NC
NC	56	□	□	22	NC
NC	55	□	□	21	NC
NC	54	□	□	20	NC
NC	53	□	□	19	NC
NC	52	□	□	18	NC
NC	51	□	□	17	NC
NC	50	□	□	16	NC
NC	49	□	□	15	NC
NC	48	□	□	14	NC
NC	47	□	□	13	NC
NC	46	□	□	12	NC
NC	45	□	□	11	NC
NC	44	□	□	10	NC
NC	43	□	□	9	GND_ISO_A
AIN15_A	42	□	□	8	AIN14_A
AIN13_A	41	□	□	7	AIN12_A
AIN11_A	40	□	□	6	AIN10_A
AIN09_A	39	□	□	5	AIN08_A
AIN07_A	38	□	□	4	AIN06_A
AIN05_A	37	□	□	3	AIN04_A
AIN03_A	36	□	□	2	AIN02_A
AIN01_A	35	□	□	1	AIN00_A

AINxx_A:

Analoge Eingangskanäle AIN00_A...AIN15_A

GND_ISO_A:

Masse der analogen Eingänge (isoliert gegen Rechner-Masse)

NC:

Pin nicht belegt

3.3 Steckerbelegung von CN2

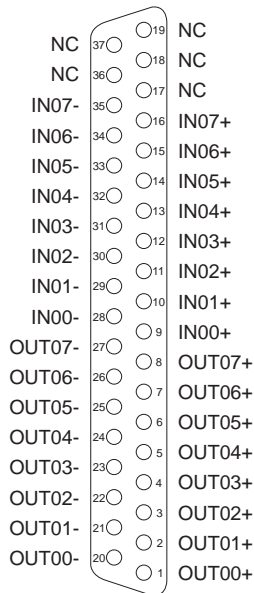
NC	40	○	○	39	NC
NC	38	○	○	37	NC
NC	36	○	○	35	NC
NC	34	○	○	33	NC
IN07-	32	○	○	31	IN07+
IN06-	30	○	○	29	IN06+
IN05-	28	○	○	27	IN05+
IN04-	26	○	○	25	IN04+
IN03-	24	○	○	23	IN03+
IN02-	22	○	○	21	IN02+
IN01-	20	○	○	19	IN01+
IN00-	18	○	○	17	IN00+
OUT07-	16	○	○	15	OUT07+
OUT06-	14	○	○	13	OUT06+
OUT05-	12	○	○	11	OUT05+
OUT04-	10	○	○	9	OUT04+
OUT03-	8	○	○	7	OUT03+
OUT02-	6	○	○	5	OUT02+
OUT01-	4	○	○	3	OUT01+
OUT00-	2	○	○	1	OUT00+

OUTxx:
Digitale Optokopplerausgänge

DINxx:
Digitale Optokopplereingänge

NC:
Pin nicht belegt

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



OUTxx:
Digitale Optokopplerausgänge

DINxx:
Digitale Optokopplereingänge

NC:
Pin nicht belegt

4. Systemkomponenten

4.1 Blockschaftbild

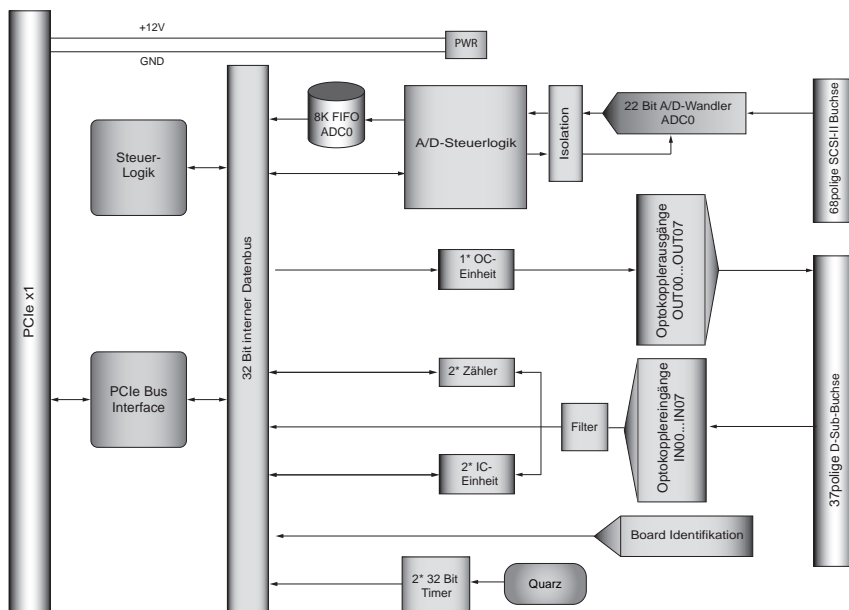


Abb. 4.1 Blockschaftbild

4.2 Zugriff auf die Systemkomponenten

Der Zugriff auf die Hardware-Komponenten der ADIODA-PCIe22 erfolgt durch das Lesen von bzw. Schreiben in Memory-Mapped I/O-Adressen mit Hilfe von Library-Funktionen. Die für die ADIODA-PCIe22 relevanten Adressen ergeben sich abhängig von einer vom Bios vergebenen Basisadresse. Der Zugriff auf die ADIODA-PCIe22 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, verfügbar auf unserer Homepage).

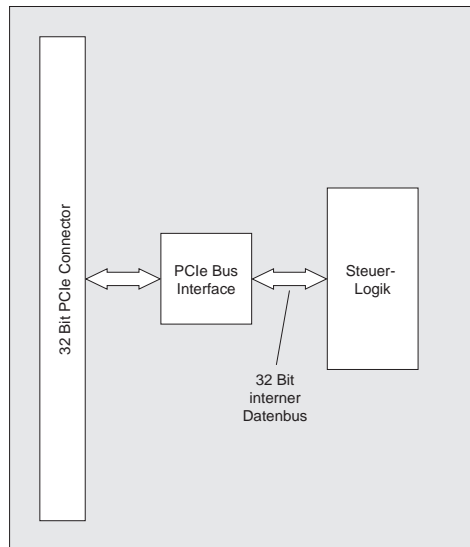


Abb. 4.2

5. 16 A/D-Eingänge 22 Bit

Die ADIODA-PCIe22 verfügt über 16 gemultiplexte single-ended oder 8 differentielle 22 Bit A/D-Eingangskanäle mit galvanischer Trennung und einem bipolaren Eingangsspannungsbereich von +/-10 V. Neben einfachen Einzelmessungen lassen sich dank einer Kanalkonfigurationstabelle und einer variablen Quarztaktung auch Mehrfach- und Dauerabtastungen über mehrere Kanäle durchführen. Neben der Kanalnummer kann auch die Messdauer der AD-Wandlung definiert werden, um zwischen einer schnellen oder einer möglichst genauen Wandlung zu wählen. Die Messwerte werden abgeglichen (Offset- und Gainfehler), in ein 8K-FIFO (8192 Messwerte) abgespeichert und stehen anschließend jederzeit zur Abholung bereit. Für ein möglichst genaues Messergebnis des ADCs wird neben der galvanischen Isolation ein jitterarmer Quarz für die interne Filterfunktion sowie eine Präzisions-Spannungsreferenz verwendet. Die Eingänge mit mind. 1 M Ω Eingangswiderstand sind mit einem Überspannungsschutz von +/-65 V geschützt.

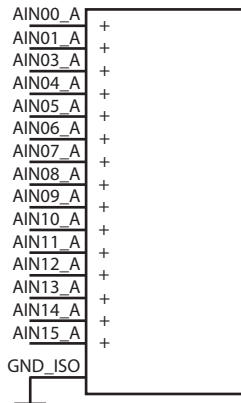
5.1 Kanalkonfigurationstabelle

Die aus 16 Registern bestehende Kanalkonfigurationstabelle (ADCCONFy) ermöglicht es dem Anwender, der PCIe-Karte genau mitzuteilen, welche Kanäle in welcher Reihenfolge abgetastet werden müssen. Im Default-Zustand steht in jedem Register der Wert 0xFFFFFFFF als Kanalkonfiguration. Bei einem Start des ADCs werden die Register mit der Kanalkonfiguration 0 beginnend nacheinander abgearbeitet, bis in einem Register der Wert 0xFFFFFFFF steht. Im Falle einer Einzelmessung wird hier die Abtastung beendet. Wird eine Mehrfach- oder eine Dauerabtastung durchgeführt, so wird nach dem Erreichen des Registers Kanalkonfiguration 15 oder dem Wert 0xFFFFFFFF von vorne begonnen. Bei einem Beschreiben des Registers Kanalkonfiguration 0 werden automatisch alle anderen Register auf 0xFFFFFFFF zurückgesetzt.

Registeradresse	
0x2040	Kanalkonfiguration 0
0x2044	Kanalkonfiguration 1
0x2048	Kanalkonfiguration 2
0x204C	Kanalkonfiguration 3
0x2050	Kanalkonfiguration 4
0x2054	Kanalkonfiguration 5
0x2058	Kanalkonfiguration 6
0x205C	Kanalkonfiguration 7
0x2060	Kanalkonfiguration 8
0x2064	Kanalkonfiguration 9
0x2068	Kanalkonfiguration 10
0x206C	Kanalkonfiguration 11
0x2070	Kanalkonfiguration 12
0x2074	Kanalkonfiguration 13
0x2078	Kanalkonfiguration 14
0x207C	Kanalkonfiguration 15

5.1.1 Single-Ended Betrieb

Im Single-Ended Betrieb stehen max. 16 Eingangskanäle zur Verfügung. Alle Eingangsspannungen in einem Eingangsspannungsbereich von +/-10 V werden gegen die Masse (GND_ISO) der A/D-Komponenten gemessen (siehe Grafik 5.1-1). Eine genauere Beschreibung der Beschaltung ist in Kapitel 19.1 zu finden.



Grafik 5.1-1 A/D-Wandler Single-ended

Wie zuvor erwähnt wird die Kanalauswahl über die Kanalkonfiguration in den entsprechenden Registern bestimmt. Welcher Wert für welchen Kanal einer single-ended Messung verwendet werden muss, ist aus der Tabelle 5.1-1 zu entnehmen.

Kanal- konfiguration	Single-ended Kanalauswahl (AINxx auf GND)																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	GND
0 _{hex}	+																-
1 _{hex}		+															-
2 _{hex}			+														-
3 _{hex}				+													-
4 _{hex}					+												-
5 _{hex}						+											-
6 _{hex}							+										-
7 _{hex}								+									-
8 _{hex}									+								-
9 _{hex}										+							-
A _{hex}											+						-
B _{hex}												+					-
C _{hex}													+				-
D _{hex}														+			-
E _{hex}															+		-
F _{hex}																+	-

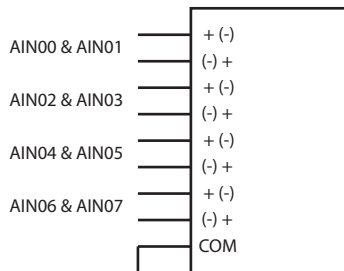
Tabelle 5.1-1 A/D-Wandler Single-ended Messung

So muss für eine single-ended Messung an Kanal 3 der Pluspol der Spannungsquelle an AIN02 und der Minuspol an ADGND angeschlossen werden. Die Kanalkonfiguration besitzt den Wert 2_{dez.}

5.1.2 Differenz-Betrieb

Im Differenz-Betrieb stehen max. 8 Eingangskanäle zur Verfügung. In der Differenz-Betriebsart gibt es für jeden Kanal jeweils einen Plus- und einen Minus-Eingang (siehe Grafik 5.1-2). Bitte beachten Sie, dass für alle Kanäle ebenfalls ein Bezug zur Masse (GND_ISO) hergestellt werden muss. Eine genauere Beschreibung der Beschaltung ist in Kapitel 19.2 zu finden. Durch die Differenzmessung können allgemein auftretende Störspannungen auf beiden Signalleitungen und der Analogmasse reduziert werden.

Der maximale Eingangsspannungsbereich beträgt +/- 10 V, wobei eine maximale Spannungsdifferenz +/- 20 V möglich ist. (AIN0 = +10 V, AIN1 = -10 V)



Grafik 5.1-2 A/D-Wandler differenzielle Messung

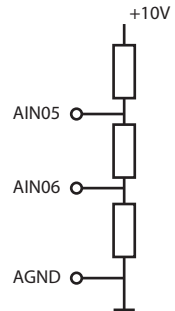
Auch hier findet die Kanalauswahl über die Kanalkonfiguration in den entsprechenden Registern statt. Die entsprechenden Werte sind aus der folgenden Tabelle zu entnehmen.

Kanal- konfiguration	Differentielle Kanalauswahl (AINxx auf AINyy)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10 _{hex}	+	-														
11 _{hex}	-	+														
12 _{hex}			+	-												
13 _{hex}			-	+												
14 _{hex}					+	-										
15 _{hex}					-	+										
16 _{hex}							+	-								
17 _{hex}							-	+								
18 _{hex}									+	-						
19 _{hex}									-	+						
1A _{hex}											+	-				
1B _{hex}											-	+				
1C _{hex}													+	-		
1D _{hex}													-	+		
1E _{hex}															+	-
1F _{hex}															-	+

Tabelle 5.1-2 A/D-Wandler differentielle Messung

Als Beispiel soll nun die Differenz zwischen zwei Spannungen an den Eingängen AIN05 und AIN06 gemessen werden. Hierfür schließen sie die erste Spannung an AIN05 und die zweite an AIN06 an (siehe Grafik 5.1-3).

Nun kann als Kanalbyte entweder der Wert 20_{dez} (AIN05+ / AIN06-) oder 21_{dez} (AIN05- / AIN06+) verwendet werden. Ergebnis ist eine negative Differenzspannung!



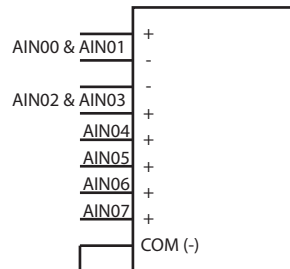
Grafik 5.1-3

Achtung:

Achten Sie darauf, dass die Differenz zwischen den Eingängen innerhalb des maximalen Eingangsspannungsbereiches +/-10 V liegen muss.

5.1.3 Kombination von Single-ended und Differenz Messung

Bei Bedarf können die Messvarianten wie in Grafik 5.1-4 auch von Kanal zu Kanal variiert werden oder sogar „on the fly“ zwischen den einzelnen Messungen geändert werden.



Grafik 5.1-4

5.2 Konfiguration der Abtastrate

Wird mehr als eine Messung durchgeführt, so wird ein mit einem Quarz getakteter 32-Bit-Timer für die regelmäßige, zeitgenaue Abtastung verwendet. Dieser wird mit einem 1MHz Takt gespeist und löst je nach Einstellung der Periodendauer zwischen 1 μ s bis $2^{32}-1$ μ s eine Wandlung aus. Die Einstellung erfolgt durch das Beschreiben des Registers ADCTCOM. Dabei entspricht ein LSB gleich 1 μ s. Die Default-Periodendauer des Timers liegt bei einer Sekunde (1000000 μ s).

Beachten Sie, dass Periodendauer (Reg. ADCTCOM) nicht kürzer als die eingestellte Messdauer (Reg. ADCRES) ist (siehe Kapitel 5.3 Konfiguration der Messauflösung und der Messdauer)

5.3 Konfiguration der Messauflösung und der Messdauer

Die ADIODA-PCle22 bietet die Möglichkeit bei den Abtastungen zwischen Schnelligkeit, Genauigkeit oder einem Mittel aus beidem zu wählen. Dies wird durch die Auswahl der Messdauer bzw. der Abtastdauer bestimmt. Zur Auswahl stehen 9 verschiedene Messgeschwindigkeiten mit ihren jeweiligen Messgenauigkeiten. Zwei der 9 Einstellmöglichkeiten verwenden zudem automatisch einen 50/60-Hz-Filter.

Die effektive Auflösung ist die maximale Messgenauigkeit in Bit mit einem Rauschanteil. Die typ. Auflösung bestimmt die typische Messgenauigkeit ohne eigenen Rauschanteil.

ADCRES-Value	effektive Auflösung ^(*)	typ. Auflösung	max. Messdauer	max. Abtastrate	Zusatzinfo
0	14 Bit	12 Bit	50µs	20 kSPS	
1	17 Bit	15 Bit	0.1ms	10 kSPS	
2	18 Bit	17 Bit	250µs	4 kSPS	
3	19 Bit	18 Bit	2ms	500 SPS	
4	20 Bit	19 Bit	8ms	125 SPS	
5	22 Bit	20 Bit	125ms	8 SPS	
6	22 Bit	20 Bit	500ms	2 SPS	
7		20 Bit	50ms	20 SPS	50&60Hz Filter
8		20 Bit	100ms	10 SPS	50&60Hz Filter

Tabelle 5.3 ADCRES-Werte mit jeweiliger Auflösung und Messdauer

^(*) Die typ. Auflösung wurde wie im Manual des verwendeten ADCs von Analog Devices durch das Messen von 1000 Messungen bei einer Messdauer von bis zu 2ms und 100 Messungen bei einer Messdauer unter 2ms gegen Masse bestimmt.

5.4 Messmodus

Insgesamt gibt es zwei Messmodi, um die AD-Eingänge abzutasten. Für Einzel- oder Mehrfachabtastungen kann der Messmodus Mehrfachabtastung (ADCMODE = 0x0) und für eine kontinuierliche Abtastung der Modus Dauerabtastung (ADCMODE = 0x1) verwendet werden.

Einzel-/Mehrfachabtastung

In diesem Modus wird die Konfigurationstabelle mit ihren aktiven Registern (alle Register bis zum Register mit dem Wert 0xFFFFFFFF) n mal abgearbeitet. Die Anzahl n entspricht dabei dem Wert des 16-Bit-Registers ADCCOUNT und kann einen Wert zwischen 1 und 65535 besitzen. Die Abtastung wird im vorgegebenen Takt (Abtastrate) durchgeführt. Jeder Takt entspricht einem Register der Tabelle. Bei einer Mehrfachmessung mit n=5 und 10 aktiven Registern in der Konfigurationstabelle würden z.B. 50 Abtastungen durchgeführt. Gestartet wird die Einzel- bzw. Mehrfachabtastung durch das Setzen des Startbits im Register ADCCON. Nach einer jeden Abtastung wird der Wert automatisch abgeglichen (Offset- und Gainfehler) und in das FIFO abgespeichert. Die Mehrfachabtastung kann auch vorzeitig durch das Setzen des Stopbits im Register ADCCON beendet werden.

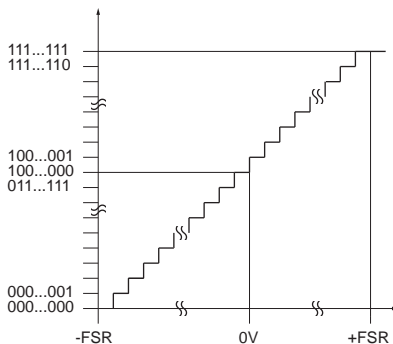
Dauerabtastung

Müssen mehr Abtastungen, als die Mehrfachabtastung ermöglicht, durchgeführt werden, kann die Dauerabtastung verwendet werden. Nach dem Setzen des Startbits im Register ADCCON wird die Konfigurationstabelle im vorgegebenen Takt (Abtastrate) solange abgearbeitet, bis das Stopbit im Register ADCCON gesetzt wird. Nach einer jeden Abtastung wird der Wert automatisch abgeglichen (Offset- und Gainfehler) und in das FIFO abgespeichert.

5.5 FIFO

Für die Zwischenspeicherung der konvertierten AD-Werte hat die ADIODA-PCle22 ein 8K FIFO (= 8192 Messwerte). Nach der erfolgreichen Abtastung eines Eingangs wird der Wert automatisch abgeglichen und in das FIFO als 24-Bit-Wert gespeichert. Ab hier kann durch das Auslesen des Registers ADCDATA der Abtastwert jederzeit vom Anwender abgeholt werden. Die Abtastwerte bleiben solange erhalten, bis sie ausgelesen wurden, ein Reset durch Setzen des Resetbits des ADCFCON Registers oder ein Reset der Karte erfolgt ist. Die genaue Anzahl an Abtastwerten im FIFO kann jederzeit durch das Einlesen des Registers ADCFCOUNT ermittelt werden. Bei einem Überlauf des FIFOs wird jeder neue Abtastwert verworfen und das Überlaufflag im ADC-Interruptregister gesetzt.

5.6 Transfer Function



5.7 Messbeispiele

5.7.1 Beispiel Einzel-/Mehrfachmessung

Der Programmaufbau einer Einzel- bzw. Mehrfachmessung ist weitestgehend identisch, weshalb diese Messmodi hier zusammengefasst werden. In diesem Beispiel wird eine Mehrfachmessung über 2 Kanäle (AIN0 & AIN4) mit jeweils 5 Messungen durchgeführt. Dabei wird eine Abtastrate von 10 Hz (jede 0.1s eine Messung) sowie einer effektiven Messauflösung von 17 Bit verwendet. Nach Abschluss der Gesamtmessung befinden sich insgesamt 10 Abtastwerte (2 Kanäle x 5 Messungen) im FIFO. Diese Werte liegen in folgender Reihenfolge vor (Messung1 AIN0, Messung1 AIN4, Messung2 AIN0,).

1. Überprüfen Sie zu Beginn, ob die Messeinheit bereit ist. Dazu lesen Sie das Statusregister ADCSTAT ein. Ist das Busy-Bit (Bit0) gleich 0, so ist die Messeinheit bereit.
2. Ist die Messeinheit bereit, so bestimmen Sie die Messauflösung bzw. Messdauer. Dazu schreiben Sie den entsprechenden Wert aus Tabelle 5.3 in das Register ADCRES.
3. Als nächstes bestimmen Sie die Abtastrate bzw. den Zeitabstand zwischen den einzelnen Messungen. Hier muss berücksichtigt werden, dass der Zeitabstand zwischen den Messungen mindestens so lange wie die zuvor bestimmte Messdauer sein muss. Die Abtastrate wird durch das Beschreiben des Registers ADCTCOM bestimmt. Für unser Beispiel von 10 Hz muss der Wert 100.000 (= 100.000 μ s) in das Register geschrieben werden.
4. Durch das Beschreiben des Registers ADCMODE wird der Messmodus bestimmt. In dem Beispiel wird eine Einzel- bzw. Mehrfachmessung benötigt, weshalb das Register ADCMODE mit dem Wert 0 beschrieben wird.

5. Als letztes müssen nun durch Beschreiben der Konfigurationstabelle die Kanäle ausgewählt werden. Der zuerst abzutastende Kanal soll AIN0 im Single Mode sein. Daher wird der Wert 0 in das Register ADCONF0 geschrieben. Damit werden alle folgenden Register (ADCONF1 bis ADCONF15) automatisch auf 0xFFFFFFFF gesetzt. Als nächstes soll der Kanal AIN4 ebenfalls im Single Mode abgetastet werden, weshalb der Wert 4 in das Register ADCONF1 geschrieben werden muss.

6. Nachdem die AD-Messeinheit nun konfiguriert wurde, kann nun mit der Messung begonnen werden. Hierfür wird das Startbit (ADCCON(0) im Register ADCCON auf 1 gesetzt. Das Bit wird automatisch zurückgesetzt und die 10 Abtastungen werden durchgeführt.

7. Wurde eine Abtastung abgeschlossen wird Sie sofort in das FIFO geschrieben und kann sowohl während als auch nach dem gesamten Messvorgang durch das Lesen des Registers ADCDATA abgeholt werden. Wieviele Werte bereits im FIFO zur Verfügung stehen, kann durch das Einlesen des Registers ADCFCOUNT überprüft werden. Um den Messvorgang vorzeitig abubrechen, muss das Stopbit im Register ADCCON gesetzt werden. Das Stopbit wird automatisch zurückgesetzt und der Messvorgang beendet.

8. Ist das Busybit ADCSTAT(0) 0, ist der Messvorgang abgeschlossen.

5.7.2 Beispiel Dauerabtastung

Für die Zwischenspeicherung der konvertierten AD-Werte hat die ADIODA-PCle22 ein 8K FIFO (= 8192 Messwerte). Nach der erfolgreichen Abtastung eines Eingangs wird der Wert automatisch abgeglichen und in das FIFO gespeichert. Ab hier kann durch das Auslesen des Registers ADCDATA der Abtastwert jederzeit vom Anwender abgeholt werden. Die Abtastwerte bleiben solange erhalten bis sie ausgelesen wurden, ein Reset durch Setzen des Resetbits des ADCFCON Registers oder ein Reset der Karte erfolgt ist. Die genaue Anzahl an Abtastwerten im FIFO kann jederzeit durch das Einlesen des Registers ADCFCOUNT ermittelt werden. Bei einem Überlauf des FIFOs wird jeder neue Abtastwert verworfen und das Überlaufflag im ADC-Interruptregister gesetzt.

In diesem Beispiel wird eine Dauerabtastung über 2 Kanäle (AIN0 & AIN4) durchgeführt. Dabei wird eine Abtastrate von 10 Hz (jede 0.1 s eine Messung) sowie einer effektiven Messauflösung von 17 Bit verwendet. Nach Abschluss der Gesamtmessung befinden sich insgesamt 10 Abtastwerte (2 Kanäle x 5 Messungen) im FIFO. Diese Werte liegen in folgender Reihenfolge vor (Messung1 AIN0, Messung1 AIN4, Messung2 AIN0,).

1. Überprüfen Sie zu Beginn, ob die Messeinheit bereit ist. Dazu lesen Sie das Statusregister ADCSTAT ein. Ist das Busy-Bit (Bit0) gleich 0, so ist die Messeinheit bereit.
2. Ist die Messeinheit bereit, so bestimmen Sie die Messauflösung bzw. Messdauer. Dazu schreiben Sie den entsprechenden Wert aus Tabelle 5.3 in das Register ADCRES.
3. Als nächstes bestimmen Sie die Abtastrate bzw. den Zeitabstand zwischen den einzelnen Messungen. Hier muss berücksichtigt werden, dass der Zeitabstand zwischen den Messungen mindestens so lange wie die zuvor bestimmte Messdauer sein muss. Die Abtastrate wird durch das Beschreiben des Registers ADCTCOM bestimmt. Für unser Beispiel von 10 Hz muss der Wert 100.000 (= 100.000 μ s) in das Register geschrieben werden.
4. Durch das Beschreiben des Registers ADCMODE wird der Messmodus bestimmt. In dem Beispiel wird eine Dauerabtastung benötigt, weshalb das Register ADCMODE mit dem Wert 1 beschrieben wird.
5. Als letztes müssen nun durch Beschreiben der Konfigurationstabelle die Kanäle ausgewählt werden. Der zuerst abzutastende Kanal soll AIN0 im Single-Ended Mode sein. Daher wird der Wert 0 in das Register ADCONF0 geschrieben. Damit werden alle folgenden Register (ADCONF1 bis ADCONF15) automatisch auf 0xFFFFFFFF gesetzt. Als nächstes soll der Kanal AIN4 ebenfalls im Single-Ended Mode abgetastet werden, weshalb der Wert 4 in das Register ADCONF1 geschrieben werden muss.
6. Nachdem die AD-Messeinheit nun konfiguriert wurde, kann nun mit der Messung begonnen werden. Hierfür wird das Startbit (ADCCON(0) im Register ADCCON auf 1 gesetzt. Das Bit wird automatisch zurückgesetzt und die Abtastungen werden durchgeführt.

7. Wurde eine Abtastung abgeschlossen, wird Sie sofort in das FIFO geschrieben und kann sowohl während als auch nach dem gesamten Messvorgang durch das Lesen des Registers ADCDATA abgeholt werden. Wieviele Werte bereits im FIFO zur Verfügung stehen, kann durch das Einlesen des Registers ADCFCOUNT überprüft werden.
8. Soll der Messvorgang beendet werden, so muss das Stopbit im Register ADCCON gesetzt werden. Das Stopbit wird automatisch zurückgesetzt und der Messvorgang beendet.
9. Ist das Busybit ADCSTAT(0) 0, ist der Messvorgang abgeschlossen.

5.8 ADC-Reset

Für bestimmte Situationen kann es erforderlich sein, dass die gesamte ADC-Einheit resettet werden muss. Dies kann durch das Setzen des Resetbits ADCCON(2) durchgeführt werden. Dadurch werden alle ADC-Einheiten von Logik über FIFO und ADC zurückgesetzt und neu konfiguriert. Das Resetbit wird automatisch zurückgesetzt.

5.9 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
0x2000	ADCCON	31:16	reserviert (*)								ADCCON 2:0>							
		15:0	reserviert (*)															
0x2004	ADCSTAT	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x2008	ADCFCOUNT	31:16	reserviert (*)															
		15:0	ADCFCOUNT <13:0>															
0x200C	ADCDATA	31:16	reserviert (*)															
		15:0	ADCDATA <23:16>															
0x2080	ADCCOUNT	31:16	reserviert (*)															
		15:0	ADCCOUNT <15:0>															
0x2084	ADCRES	31:16	reserviert (*)															
		15:0	ADCRES <7:0>															
0x2048	ADCMODE	31:16	reserviert (*)															
		15:0	reserviert (*)								ADCMODE <3:0>							
0x204C	ADCTCOM	31:16	ADCTCOM<31:16>															
		15:0	ADCTCOM<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
0x2060	ADCCONF8	31:16	reserviert (*)															
0x2064	ADCCONF9	15:0	reserviert (*)															
0x2068	ADCCONF10	31:16	reserviert (*)															
0x206C	ADCCONF11	15:0	reserviert (*)															
0x2070	ADCCONF12	31:16	reserviert (*)															
0x2074	ADCCONF13	15:0	reserviert (*)															
0x2078	ADCCONF14	31:16	reserviert (*)															
0x207C	ADCCONF15	15:0	reserviert (*)															

(*) reservierter Bereich ist mit 0 zu belegen

Register ADCCON:

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							
	reserviert							
23:16	W							
	reserviert							
15:8	W							
	reserviert							
7:0	W							
	reserviert					RESET	STOP	START

Bit 31 - 3 reserviert (mit 0 belegen)

Bit 2 **ADCCON <2> RESET**

1 = Führt einen Reset der ganzen ADC-Einheit von Logik, FIFO und ADC aus. Das Bit wird automatisch auf 0 zurückgesetzt

Bit 1 1 = Beendet die laufende Abtastung. Das Bit wird automatisch auf 0 zurückgesetzt.

Bit 0 1 = Startet die Abtastung mit den zuvor eingestellten Konfigurationen. Das Bit wird automatisch auf 0 zurückgesetzt

Register ADCSTAT:

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							
	reserviert							
23:16	R							
	reserviert							
15:8	R							
	reserviert							
7:0	R							
	reserviert							BUSY

Bit 31 - 1 reserviert

Bit 0 **ADCCON <0> BUSY**

1 = ADC ist gerade beschäftigt.

0 = ADC ist gerade im Leerlauf

Register ADCFCOUNT:

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							
	reserviert							
23:16	R							
	reserviert							
15:8	R							
	reserviert	ADCFCOUNT<13:8>						
7:0	R							
	ADCFCOUNT<7:0>							

Bit 31 - 14 reserviert (mit 0 belegt)

Bit 13 - 0 **ADCFCOUNT <13:0>**

Hier steht die aktuelle Anzahl an Messwerten im FIFO

Register ADCDATA:

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							
	reserviert							
23:16	R							
	ADCDATA <23:16>							
15:8	R							
	ADCDATA <15:8>							
7:0	R							
	ADCDATA <7:0>							

Bit 31 - 24 reserviert (mit 0 belegt)

Bit 23 - 0 **ADCDATA <23:0>**

Aus diesem Register können die Abtastwerte aus dem FIFO ausgelesen werden

Register ADCCOUNT:

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							
	reserviert							
23:16	W							
	reserviert							
15:8	W							
	ADCCOUNT<15:8>							
7:0	W							
	ADCCOUNT<7:0>							

Bit 31 - 16 reserviert (mit 0 belegen)

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

Konfiguration der Anzahl an Messungen im Einzel-/Mehrfachmodus pro Kanal

1 = Einzelmessung

> 1 = Mehrfachmessung

Register ADCRES:

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							
	reserviert							
23:16	W							
	reserviert							
15:8	W							
	reserviert							
7:0	W							
	ADCCOUNT<7:0>							

Bit 31 - 8 reserviert (mit 0 belegen)

Bit 7 - 0 **ADCRES <7:0>**

Konfiguration Messauflösung und der Messdauer.

(siehe Tabelle 5.3 im Kapitel 5.3 Konfiguration der Messauflösung und der Messdauer)

Register ADCMODE:

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							
	reserviert							
23:16	W							
	reserviert							
15:8	W							
	reserviert							
7:0	W							
	reserviert				ADCCOUNT<3:0>			

Bit 31 - 4 reserviert (mit 0 belegen)

Bit 3 - 0 **ADCMODE <3:0>**
 Einstellung des Messmodus
 0 = Einzel-/Mehrfachabtastung
 1 = Dauerabtastung
 2 .. 7 = reserviert (mit 0 belegen)

Register ADCTCOM:

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							
	ADCTCOM<31:24>							
23:16	W							
	ADCTCOM<23:16>							
15:8	W							
	ADCTCOM<15:8>							
7:0	W							
	ADCTCOM<7:0>							

Bit 31 - 0 **ADCTCOM<3:0>**
 Register für Einstellung der Abtastrate von Mehrfach- und Dauerabtastung. Der Registerwert entspricht der Dauer zwischen den einzelnen Messungen in μs . Der Wert muss immer größer gleich der eingestellten Messdauer sein (siehe ADCRES).
 (default) = 100.0000 μs = 1 s

Register ADCCONF_y (y = ADCCONF-Registernummer):

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							
	ADCCONF<31:24>							
23:16	W							
	ADCCONF<23:16>							
15:8	W							
	ADCCONF<15:8>							
7:0	W							
	ADCCONF<7:5>				CHANNEL<4:0>			

Bit 31 - 0

- 0x00000000 = VIN0 Single-Ended
- 0x00000001 = VIN1 Single-Ended
- 0x00000002 = VIN2 Single-Ended
- 0x00000003 = VIN3 Single-Ended
- 0x00000004 = VIN4 Single-Ended
- 0x00000005 = VIN5 Single-Ended
- 0x00000006 = VIN6 Single-Ended
- 0x00000007 = VIN7 Single-Ended
- 0x00000008 = VIN8 Single-Ended
- 0x00000009 = VIN9 Single-Ended
- 0x0000000A = VIN10 Single-Ended
- 0x0000000B = VIN11 Single-Ended
- 0x0000000C = VIN12 Single-Ended
- 0x0000000D = VIN13 Single-Ended
- 0x0000000E = VIN14 Single-Ended
- 0x0000000F = VIN15 Single-Ended
- 0x00000010 = VIN0, VIN1 Differenz
- 0x00000011 = VIN1, VIN0 Differenz
- 0x00000012 = VIN2, VIN3 Differenz
- 0x00000013 = VIN3, VIN2 Differenz
- 0x00000014 = VIN4, VIN5 Differenz
- 0x00000015 = VIN5, VIN4 Differenz
- 0x00000016 = VIN6, VIN7 Differenz
- 0x00000017 = VIN7, VIN6 Differenz
- 0xFFFFFFFF = Register deaktivieren

Register ADCCONF_y (y = ADCCONF-Registernummer):

Bit 31 - 0	0x000000018 = VIN8, VIN9 Differenz
	0x000000019 = VIN9, VIN8 Differenz
	0x00000001A = VIN10, VIN11 Differenz
	0x00000001B = VIN11, VIN10 Differenz
	0x00000001C = VIN12, VIN13 Differenz
	0x00000001D = VIN12, VIN13 Differenz
	0x00000001E = VIN14, VIN15 Differenz
	0x00000001F = VIN15, VIN14 Differenz
	:
	reserviert
	:
	0xFFFFFFFF = Register deaktivieren

6. 8 Optokopplereingänge

Die ADIODA-PCIe22 verfügt über 8 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.

6.1 Pinbelegung der Eingangsoptokoppler

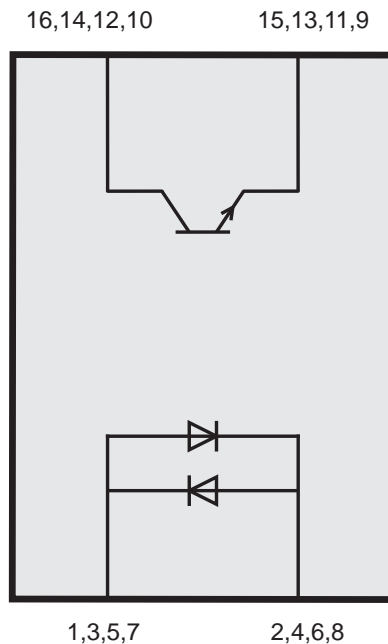
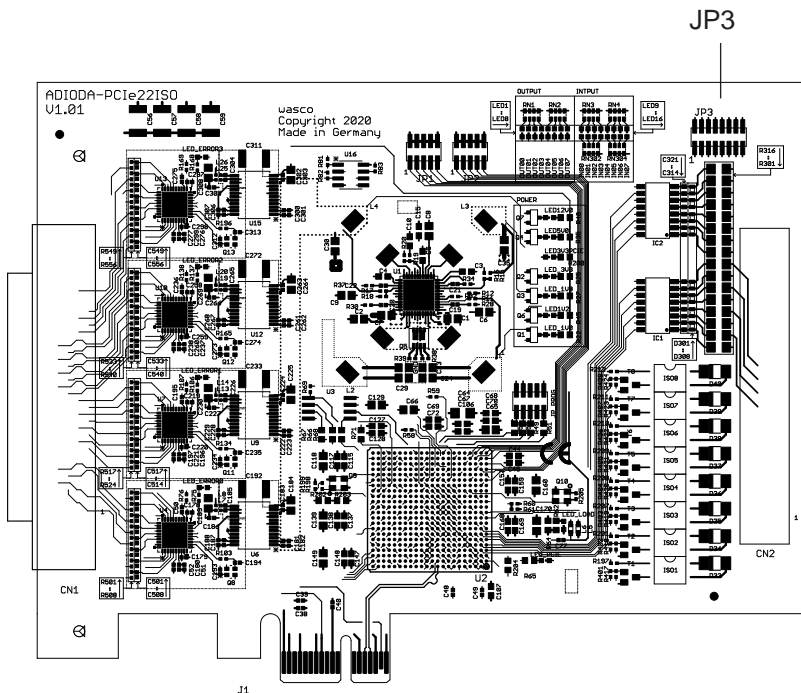


Abb. 6.1

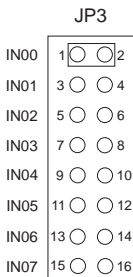
6.2 Eingangsspannungsbereiche

Durch das Setzen von Jumpers auf dem Block JP3 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 JP3 wechselt der Eingangsspannungsbereich des IN00 von 0..2 V (LOW) und 14..30 V (HIGH) auf 0..1 V (LOW) und 5..15 V (HIGH). Die restlichen Eingangsspannungsbereiche bleiben unverändert.

6.3 Eingangsbeschaltung

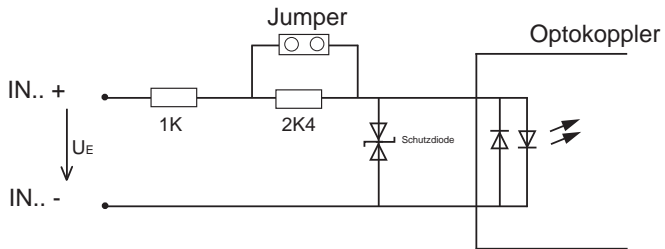


Abb. 6.3

6.4 Eingangsstrom

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

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

6.5 Zugriff auf die Eingänge

Um den Zustand der Optokopplereingänge zu ermitteln, muss das Register OPTOIN ausgelesen werden. Jedes Bit des 32-Bit-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 0x44(hex)/17476(dec)/0b01000100(bin) zurück. Aus diesem Wert können nun die Zustände der einzelnen Eingänge durch eine Und-Verknüpfung herausgefiltert werden.

6.6 Optokopplereingänge mit digitalem Filter

Jeder Optokopplereingang des Boards WASCO-PCIe8368 besitzt einen eigenen konfigurierbaren digitalen Filter, um Störungen oder Einschwingungen des digitalen Eingangssignals zu filtern. Dabei überprüft der Filter wie in Abb. 6.5 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 μ s in 1- μ s-Schritten eingestellt werden. Im Default-Zustand ist der Filter deaktiviert, sprich die Filterdauer beträgt 0 μ s.

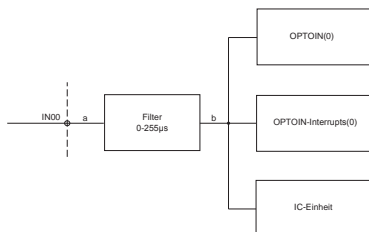


Abb. 6.4

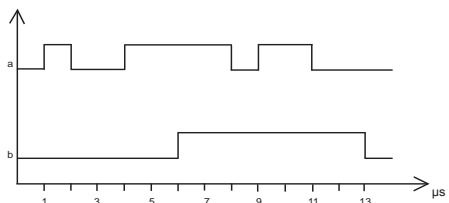


Abb. 6.5

Zu beachten ist bei der Verwendung der Filter, dass die Optokopplereingänge zum einen weit längere Schaltzeiten über 1 μ 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.

6.7 Interruptfunktionen der Optokopplereingänge

Um Änderungen an den Optokopplereingängen ohne regelmäßiges Abfragen der Eingänge durch den PC zu erkennen, bietet die ADIODA-PCle22 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.

6.7.1 Flankenerkennung

Um steigende Flanken an den Optokopplereingängen zu erkennen, besitzt jeder einzelne Eingang eine Flankenerkennung mit zuschaltbarer Interruptfunktion. Dafür wird ein 32-Bit-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 32-Bit-Registers OPTOINIFe. Dabei repräsentiert jedes einzelne Bit einen Eingang. Wie aus der Tabelle Portadressen (Kapitel Interruptcontroller 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 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.

6.7.2 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-PCIe8368 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 32-Bit-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.

6.8 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	reserviert (*)								OPTOIN 7:0>							
0x00A0	OPTOINFIL0	15:0	reserviert (*)															
0x00A4	OPTOINFIL1	31:16	OPTOINFIL0 <15:0>															
0x00A8	OPTOINFIL2	15:0	reserviert (*)															
0x00AC	OPTOINFIL3	31:16	OPTOINFIL1 <15:0>															
0x00B0	OPTOINFIL4	15:0	reserviert (*)															
0x00B4	OPTOINFIL5	31:16	OPTOINFIL2 <15:0>															
0x00C0	OPTOINFIL6	15:0	reserviert (*)															
0x00BC	OPTOINFIL7	31:16	OPTOINFIL3 <15:0>															
		15:0	reserviert (*)															

(*) 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					U				
					-				
7:0					R				
					OPTOIN <7:0>				

Bit 31 - 8 undefiniert

Bit 7 - 0 **OPTOIN <7:0>**

jedes Bit entspricht einem Optokopplereingang

(Z.B. IN00 = OPTOIN<0>, IN7 = OPTOIN<7>)

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)

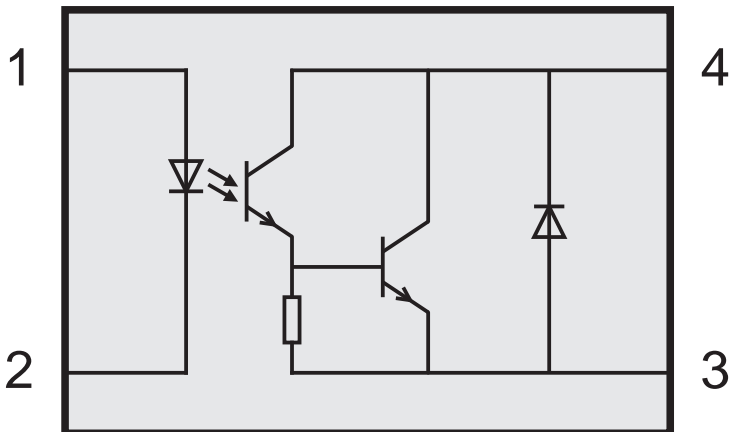
Bit 15 - 0 **OPTOINFILx <15:0>** (default = 0)

Der Wert bestimmt die Filterdauer des Filters x in µs

7. 8 Optokopplerausgänge

Die ADIODA-PCle22 verfügt über 8 Ausgangskanäle, deren galvanische Trennung ebenfalls mittels Optokoppler erreicht wird. Die Isolationsspannung zwischen Masse des Computers und Ausgang beträgt 500 Volt.

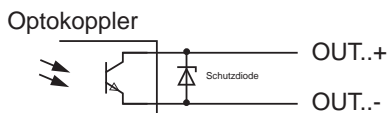
7.1 Pinbelegung der Ausgangsoptokoppler



7.2 Optokopplerdaten

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

7.3 Ausgangsbeschaltung



7.4 Funktionen der Optokopplerausgänge

7.4.1 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 0x44(hex), 68(dec) bzw. 0b01000100(bin) in das Register OPTOOUT geschrieben werden.

7.4.2 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. 7.1). Hierfür besitzt jeder Optokoppler 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 OPTOOUT-Register als Peripherie festgelegt. Um die Quelle zu verändern, muss die Quelladresse (siehe Abb. 7.2) in das Register OPTOOUTMUXx geschrieben werden.

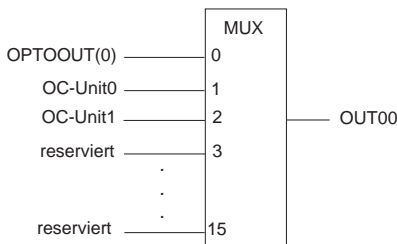


Abb. 7.1

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

Abb. 7.2

7.5 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	reserviert (*)															
0x03C0	OPTOOUTMUX0	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03C4	OPTOOUTMUX1	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03C8	OPTOOUTMUX2	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03CC	OPTOOUTMUX3	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03D0	OPTOOUTMUX4	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03D4	OPTOOUTMUX5	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03D8	OPTOOUTMUX6	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03DC	OPTOOUTMUX7	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	U							
	reserviert							
7:0	R/W							
	OPTOOUT <7:0>							

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

Bit 7 - 0 **OPTOOUT <7:0>** (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				R/W			
	reserviert				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

8. Zähler

Die WASCO-PCIe8368 stellt insgesamt 2 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.

8.1 Grundfunktion

1. Um einen Zähler zu verwenden, muss zu Beginn die Quelle ausgewählt werden. Hierfür hat jeder Zähler sein eigenes 32-Bit-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.

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.

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 (*)															
			en															
0x1004	COUNT1e	31:16	reserviert (*)															
		15:0	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
0x1080	COUNTc0	31:16 15:0	reserviert															
0x1100	COUNT0	31:16 15:0	COUNT0<31:16> COUNT0<15:0>															
0x1104	COUNT1	31:16 15:0	COUNT1<31:16> COUNT1<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
0x1180	COUNTMUX0	31:16	reserviert (*)															
		15:0	reserviert (*)															
			COUNTMUX0 <7:0>															
0x1184	COUNTMUX1	31:16	reserviert (*)															
		15:0	reserviert (*)															
			COUNTMUX1 <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 - 4 reserviert (mit dem Wert 0 beschreiben)

Bit 3 - 0 **COUNTMUXx <7:0>** (default = 0)

Registerwert bestimmt die Kartenperipherie, welche am Zähler anliegt

0 = Optokopplereingang IN00 (default)

1 = Optokopplereingang IN01

.

.

.

7 = Optokopplereingang IN7

255 - 8 = reserviert -> mit 0 belegen

9. Timer

Die zur Verfügung stehenden 32-Bit-Timer lassen sich als Zeitgeber oder zur konfigurierbaren Intervall-Interruptauslösung verwenden. Dabei können Intervalle zwischen 0 und 4.294.967.295 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 **TIMERx** deaktiviert und anschließend resettet werden. Der Reset wird durch das Beschreiben des Registers **TIMERx** mit dem Wert 0 durchgeführt.
2. Anschließend muss das Intervall festgelegt werden. Die Intervalldauer wird im beschreibbaren 32-Bit-Register **TIMERCOMPx** festgelegt.
$$\text{Intervalldauer} = (\text{TIMERCOMPx} + 1) \cdot 1\mu\text{s}$$
3. Damit der Timer nach Ablauf des Intervalls einen Interrupt auslöst, muss dieser freigeschaltet werden, was über das Setzen des entsprechenden Bits im Register **TIMERIR** geschieht. (Achtung: auch der Interruptcontroller muss freigeschaltet sein)
4. Ist der Timer fertig konfiguriert, kann er durch das Setzen von Bit 0 im Register **TIMERx** 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.

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

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

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 32-Bit-Timer für die Zeitmessung in 1- μ s-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 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.

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.

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

(*) 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

.

.

.

7 = Optokopplereingang IN7

255 - 8 = 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 μ 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 μ s ausgelesen werden

11. Output-Compare-Einheit

Mit der Output-Compare-Einheit (OC-Unit) ermöglicht die WASCO-PCle8368-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} μs und einer Pulsdauer von 1 bis 2^{32} μs erzeugt werden.

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

11.1 PWM

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

11.1.1 Funktionsweise

Um die PWM zu realisieren, verwendet die OC-Einheit einen beschreibbaren 32-Bit-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 (Optokopplerausgang gesperrt). 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 μs). 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).

11.1.2 Berechnung der Registerwerte

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

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

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

11.1.3 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.

11.2 Pulsausgabe

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

11.2.1 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 OCUNITxe ein Puls wie in folgender Grafik ausgegeben werden.

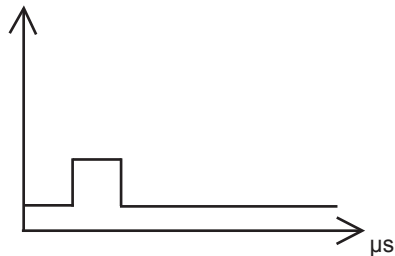


Abb. 11.1

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

11.2.2 Berechnung der Registerwerte

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

11.2.3 Anwendungsbeispiel

1. OC-Einheit konfigurieren

- a) Deaktivieren Sie die Einheit durch Löschen (= 0) des en-Bits im Register OCUNITx_e
- 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 11.2.2 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_e deaktivieren
- b) OC-Einheit durch Setzen des Enable-Bits im Register OCUNITx_e 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	OPERIOD0	31:16	OPERIOD0 <31:16>															
		15:0	OPERIOD0 <15:7>															
0x1664	OPERIOD1	31:16	OPERIOD1 <31:16>															
		15:0	OPERIOD1 <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 μs
 Pulsdauer = OCUNITORx [μs]

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 μs
 Periodendauer = OCPERIODx + 1 [μs]

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)

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. 12.1 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 INTR gelöscht werden und ein weiterer Interrupt kann an den PC geschickt werden.

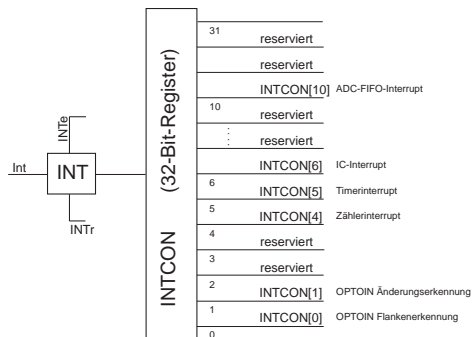


Abb. 12.1

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 (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) zutrifft, alle anderen Interrupts ebenfalls löschen
- e) Interruptfunktion wieder freigeben (INTr = 1)

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 (*)															
		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 (*)															
0x039C	ADCFIRE	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03A0	ADCFIRr	31:16	reserviert (*)															
		15:0	reserviert (*)															
0x03A4	ADCFIR	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 = Einer 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 OPTOINFe

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					R/W				
	OPTOINFe <7:0>								

Bit 31 - 8 undefiniert

Bit 7 - 0 **OPTOINFe<7:0>** 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 => OPTOINFe<0>, IN7 => OPTOINFe<7>). Eine 1 im Bit schaltet die Flankeninterrupt funktion 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					U				
					-				
7:0					W				
	OPTOINIFr <7:0>								

Bit 31 - 8 undefiniert

Bit 7 - 0 **OPTOINIFr<7:0>** Jedes Bit entspricht einem Optokopplereingang (z.B. IN00 => OPTOINIFr<0>, IN7 => OPTOINIFr<7>). 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	U							
	-							
7:0	R							
	OPTOINIF <7:0>							

Bit 31 - 8 undefiniert

Bit 7 - 0 **OPTOINIF<8:0>** zeigt, ob an einem der Optokopplereingänge eine steigende Flanke angelegen ist. Jedes Bit entspricht einem Optokopplereingang (z.B. IN00 => OPTOINIF<0>, IN17 => OPTOINIF<7>). 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	U							
	reserviert							
7:0	R/W							
	OPTOINICc<7:0>							

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

Bit 7 - 0 **OPTOINICc<7:0>** (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>, IN7 => OPTOINICc<7>)

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 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 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					U				
					reserviert				
7:0					R/W				
					COUNTIRE<7:0>				

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

Bit 7 - 0 **COUNTIRE<7:0>** (default = 0)
 Hiermit können die Interruptfunktionen der Zähler frei geschaltet werden. Jedes Bit entspricht einem Zähler
 (z.B. Zähler 0 => COUNTIRE<0>, Zähler 7 => COUNTIRE<7>)
 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	U							
	reserviert							
7:0	R/W							
	COUNTIRr<7:0>							

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

Bit 7 - 0 **COUNTIRr<7:0>**

Jedes Bit entspricht einem Zähler (z.B. Zähler 0 => COUNTIRr<0>, Zähler 7 => COUNTIRr<7>). 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	U							
	reserviert							
7:0	R/W							
	COUNTIR<7:0>							

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

Bit 7 - 0 **COUNTIR<7:0>** zeigt an, ob durch einen Zähler ein Interrupt ausgelöst wurde. Jedes Bit entspricht einem Zähler (z.B. Zähler 0 => COUNTIR<0>, Zähler 7 => COUNTIR<7>)
 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 Rücksetzen 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							TIMERIR <1:0>
	reserviert							

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							R/W
	reserviert							
								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<1:0>	

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

Register ADCFIRE:

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							ADCFIRE<0>

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

Bit 0 **ADCFIRE<0>** hiermit kann die Interruptfunktion des ADC-FIFOs bei einem OVERFLOW freigeschalten werden.

0 = Interrupt gesperrt (default)

1 = Interrupt freigegeben

Register ADCFIRr:

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							ADCFIRr<0>

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

Bit 0 **ADCFIRr<0>** Wurde durch einen Overflow des ADS-FIFOs ein Interrupt ausgelöst, muss dessen Signalbit im Register ADCFIR zurückgesetzt werden. Dies erfolgt durch das Setzen (= 1) des entsprechenden ADCFIRr-Bits. Die ADCFIRr-Bits werden nach dem Rücksetzen von selbst auf 0 zurückgesetzt.

Register ADCFIR:

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							ADCFIR<0>

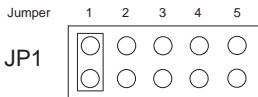
Bit 31 - 1 reserviert (mit dem Wert 0 beschreiben)

Bit 0 **ADCFIR<0>** zeigt an, ob durch einen Overflow des ADC-FIFOs ein Interrupt ausgelöst wurde.

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 15:0	reserviert (*)								reserviert (*)								Board ID	

(*) reservierter Bereich ist mit 0 zu belegen

14. Programmierung unter Windows[®]

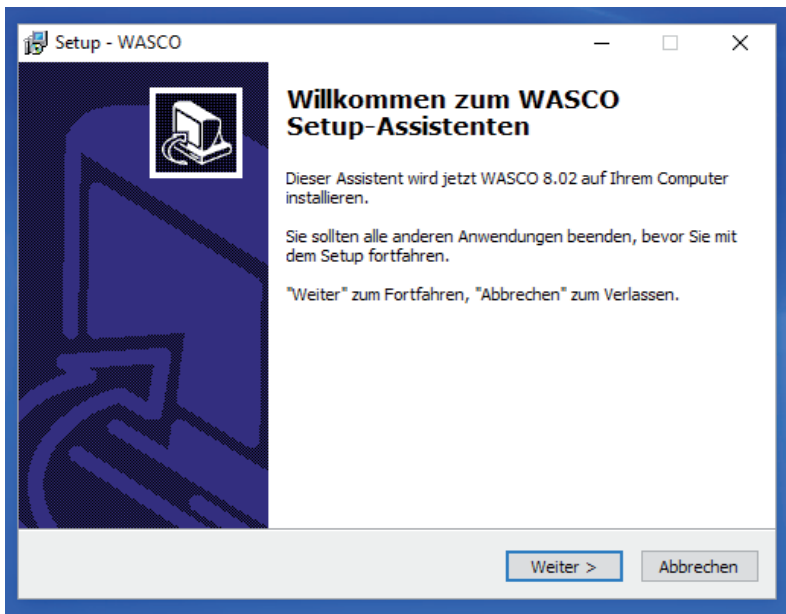
14.1 Installation des Windows[®] Treibers

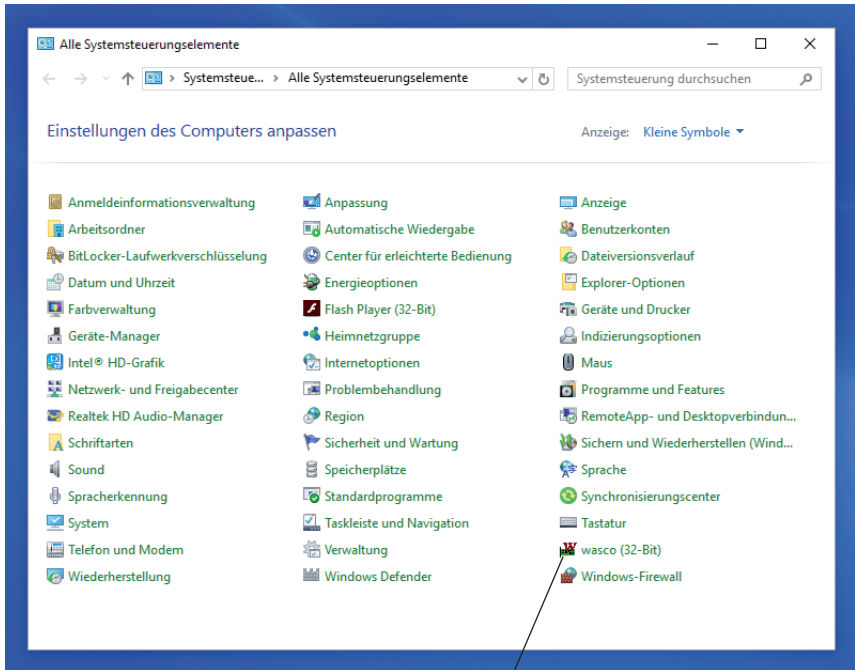
Für die Anwendung der Karte unter Windows[®] ist es notwendig, einen speziellen Treiber zu installieren, der den Zugriff auf die Karte ermöglicht. Unter Windows[®] 10, 8 und 7 meldet das Betriebssystem selbständig nach dem Einschalten des PCs, dass eine neue Hardware-Komponente gefunden wurde. Die Treiberdateien stehen auf unserer Homepage kostenfrei zum Download zur Verfügung.

Unter www.messcomp.com/php/software_passwort.php kann ein Passwort angefordert werden.

14.2 Installation der Windows[®] Entwicklungsdateien

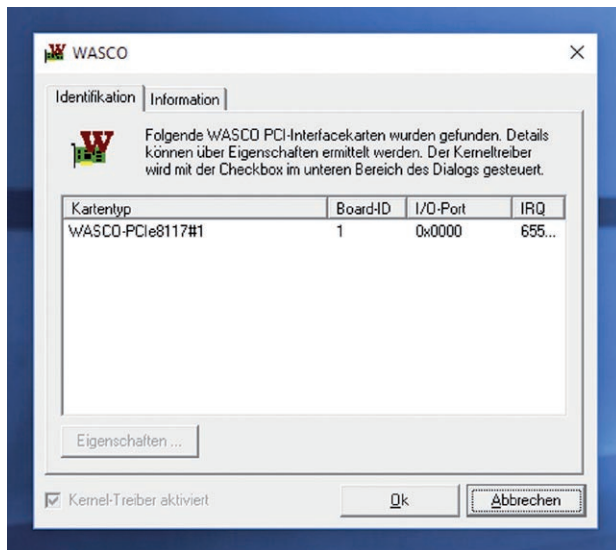
Zur Installation der Windows[®] Entwicklungsdateien führen Sie bitte die Datei "Setup.exe" und folgen Sie den Installationsanweisungen.





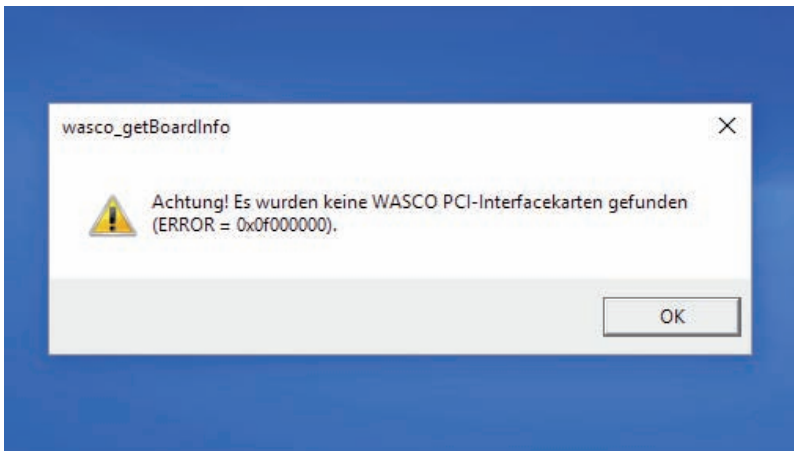
Wurden die Treiber- und Entwicklungsdateien vollständig installiert, finden Sie in der Systemsteuerung Ihres Rechners ein Icon zur Lokalisierung aller im System vorhandenen **wasco**[®] PCI- und PCIe-Karten.

Starten Sie die Kartenabfrage durch einen Doppelklick auf das "wasco®"-Icon. Folgendes Fenster erscheint: (Als Beispiel wurde hier eine ADIODA-PCle22^{ULTRA} verwendet)!



Wurde Ihre Karte im System erkannt, wird der Boardname WASCO-PCle8368, 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 ADIODA-PCIe22 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 ADIODA-PCIe22, befinden sich auf unserer Homepage.

Die Programmierung der Hardwarekomponenten der ADIODA-PCIe22 erfolgt durch den Zugriff auf Memory Mapped I/O-Adressen, die sich abhängig von der vom BIOS des Systems für die ADIODA-PCIe22 vergebenen Basisadresse ergeben. Eine genauere Beschreibung zur Programmierung befindet sich in der Treiberdokumentation.

14.4 Zugriff auf die Karte ADIODA-PCle22^{ULTRA}

Der Zugriff auf die ADIODA-PCle22^{ULTRA} erfolgt ausschließlich über den Boardnamen (Kartentyp) WASCO-PCle8368

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
Optokoppler-Eingangsport (IN00...IN07)	BA + \$0	RD
Optokoppler-Ausgangsport (OUT00...OUT07)	BA + \$8	RD/WR
Board Identifikation	BA+ \$FF8	RD

15. Programmierung unter Linux[®]

Für die Anwendung der Karte unter Linux[®] wird 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.

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 ADIODA-PCle22 mit **wasco**[®]-Treiber

Die Programmierung der Hardwarekomponenten der ADIODA-PCle22 erfolgt durch den Zugriff auf Memory Mapped I/O-Adressen, die sich abhängig von der vom BIOS des Systems für die ADIODA-PCle22 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 ADIODA-PCle22 befinden sich auf unserer Homepage.

15.4 Zugriff auf die Karte ADIODA-PCle22^{ULTRA}

Der Zugriff auf die ADIODA-PCle22^{ULTRA} erfolgt ausschließlich über den Boardnamen (Kartentyp) WASCO-PCle8368.

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:

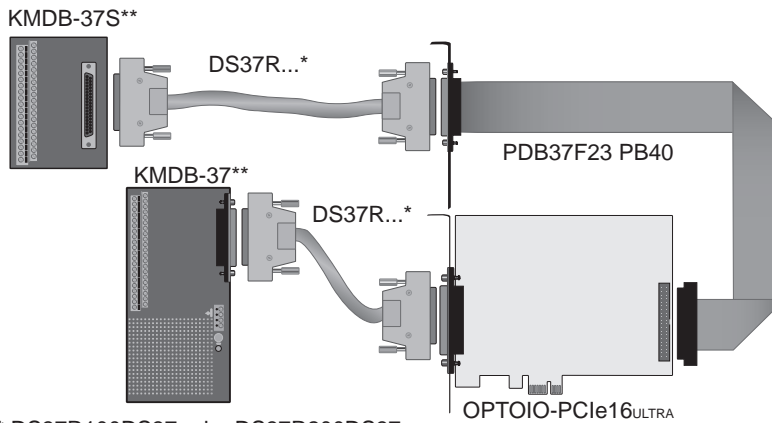
Port/Register	BA + Offset	RDWR
Optokoppler-Eingangsport (IN00...IN07)	BA + \$0	RD
Optokoppler-Ausgangsport (OUT00...OUT07)	BA + \$8	WR
Board Identifikation	BA+ \$FF8	RD

16. Zubehör

16.1 Passendes **wasco**[®]-Zubehör

Anschlusssteile	EDV-Nr.
PDB37F23PB40 Steckerverlegungsset	A-497500
DS68R100DS68 Verbindungsleitung	A-492200
DS68R200DS68 Verbindungsleitung	A-494200
DS68R500DS68 Verbindungsleitung	A-498200
DS37R100DS37 Verbindungsleitung	A-202200
DS37R200DS37 Verbindungsleitung	A-202400
DS37R500DS37 Verbindungsleitung	A-202800
KMDB-68 Klemm-Modul	A-494800
KMDB-37S Klemm-Modul (ohne Lötrasterfeld)	A-204910
KMDB-37 Klemm-Modul	A-2046

16.2 Anschlusstechnik (Anwendungsbeispiele)



* DS37R100DS37 oder DS37R200DS37
oder DS37R500DS37

** alternativ KMDB-37 mit oder ohne Lötasterfeld

16.3 Einzelkomponenten zur Eigenkonfektionierung

Anschlusssteile	EDV-Nr.
SCSI-II Stecker 68pol. für Flachbandleitung	A-553200
SCSI-II Buchse 68pol. für Flachbandleitung	A-557200
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 68pol. Stecker/Buchse	A-57780
Slotblech mit Ausschnitt für 37pol. Stecker/Buchse	A-5754
Pfostenbuchse 40pol. für Flachbandleitung	A-5642
Flachbandleitung 68pol.	A-572800
Flachbandleitung 37pol.	A-5718
Flachbandleitung 40pol.	A-5720

17. Fehlersuche

Nachfolgend finden Sie eine kurze Zusammenstellung der häufigsten bekannten Fehlerursachen, die während der Inbetriebnahme oder während der Arbeit mit der ADIODA-PCIe22 auftauchen können.

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

1. Sitzt die ADIODA-PCIe22 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

AD-Eingänge

16 Eingänge single-ended
oder 8 Eingänge differentiell (diff)
oder kombiniert se/diff per Software wählbar
ADC mit Potentialtrennung
Auflösung: bis zu 22 Bit
Eingangsspannungsbereich: +/-10V
Absoluter Eingangsspannungsbereich: +/-20V
FIFO: 8K (8192 Messwerte)
Eingangswiderstand: > 1 M Ω
Überspannungsschutz: +/-65 V
max. Abtastrate: 20 kHz
Präzisionsreferenz: 2 ppm/°C
Wandlungsauslösung per Software oder Timer

Digitale Eingänge über Optokoppler

Optokoppler
8 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
8 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

Timer

2*32-Bit Aufwärtszähler
Zählfrequenz 1MHz
Zeitabhängige Interruptauslösung
Takt von Quarzoszillator

Zähler

2*32-Bit Aufwärtszähler mit Überlaufbit
Interruptmöglichkeit bei Überlauf

Output-Compare-Einheit

2*32-Bit OC-Einheiten
Auflösung 1 µs
PWM-Erzeugung
Erzeugung einzelner Pulse

Input-Capture-Einheit

2*32-Bit IC-Einheiten
Auflösung 1 µs
Perioden- und Pulsdauermessungen

Quarzoszillator

4 MHz

Board-Identifikation

5-fach Jumperblock

Anschlusstecker

1 * 68-polige SCSI-Buchse
1 * 40-poliger Pfostenstecker

Bussystem

PCI Express 1.0 x1 (32 Bit Datenzugriff)

Abmessungen der Platine

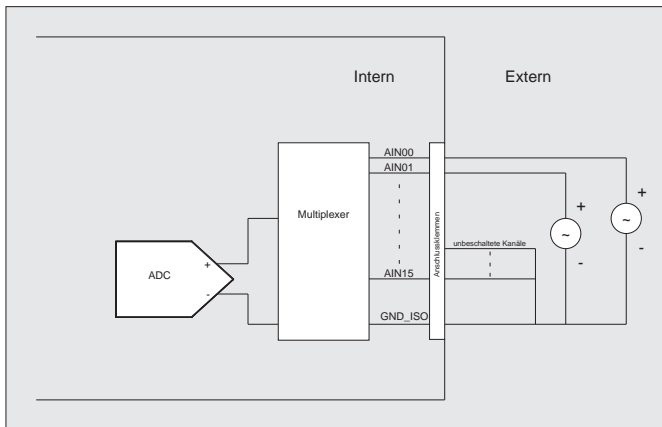
143 mm x 111 mm (l x b)
standard height, half length card
Multilayer-Platine

Sonstiges

Kontroll-LEDs für Spannungsversorgung

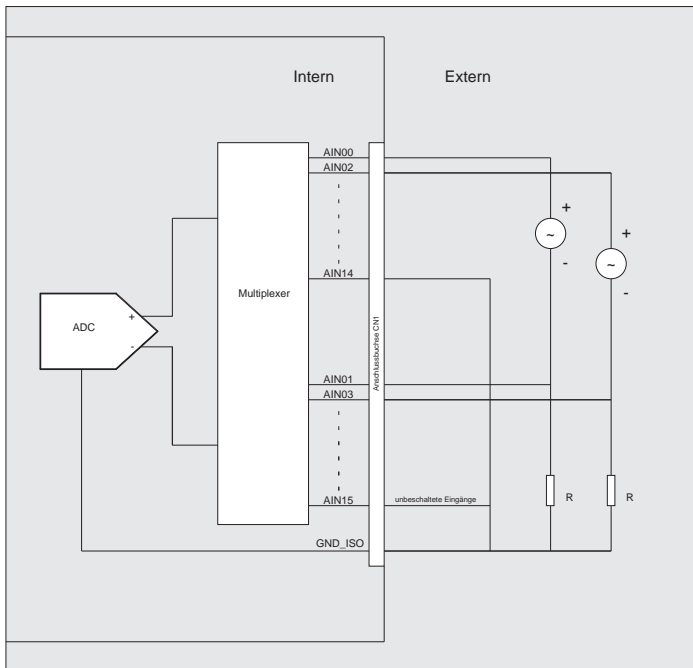
19. Beschaltungsbeispiele

19.1 Beschaltung der A/D-Eingänge single-ended



Grafik 19.1 Beschaltung der AD-Eingänge (Single-ended)

19.2 Beschaltung der A/D-Eingänge differentiell



Grafik 19.2 Beschaltung der AD-Eingänge (differentiell)

20. 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.

21. EG-Konformitätserklärung

Für das folgende mit CE-Kennzeichen gekennzeichnete Erzeugnis

ADIODA-PCle22^{ISO16}
EDV-Nummer A-812230

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, 06.02.2024



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:	ADIODA-PCIe22 _{ISO16}	A-812230

Die Firma Messcomp Datentechnik hat sich intensiv mit dem Thema Nachhaltigkeit und Ressourcenschonung beschäftigt und einige Prozesse verändert.

So haben wir uns in einem weiteren Schritt entschlossen, unseren Handbüchern keine CD mehr beizulegen.

Sie finden alle notwendigen Informationen und Dateien stets aktuell im Internet unter www.messcomp.com.

Der Software-Download der Treiber und Beispielprogramme ist kostenlos verfügbar, bitte fordern Sie das Passwort über die Seite www.messcomp.com/php/software_passwort.php an.

Das Passwort wird Ihnen umgehend vom Server zugewiesen.