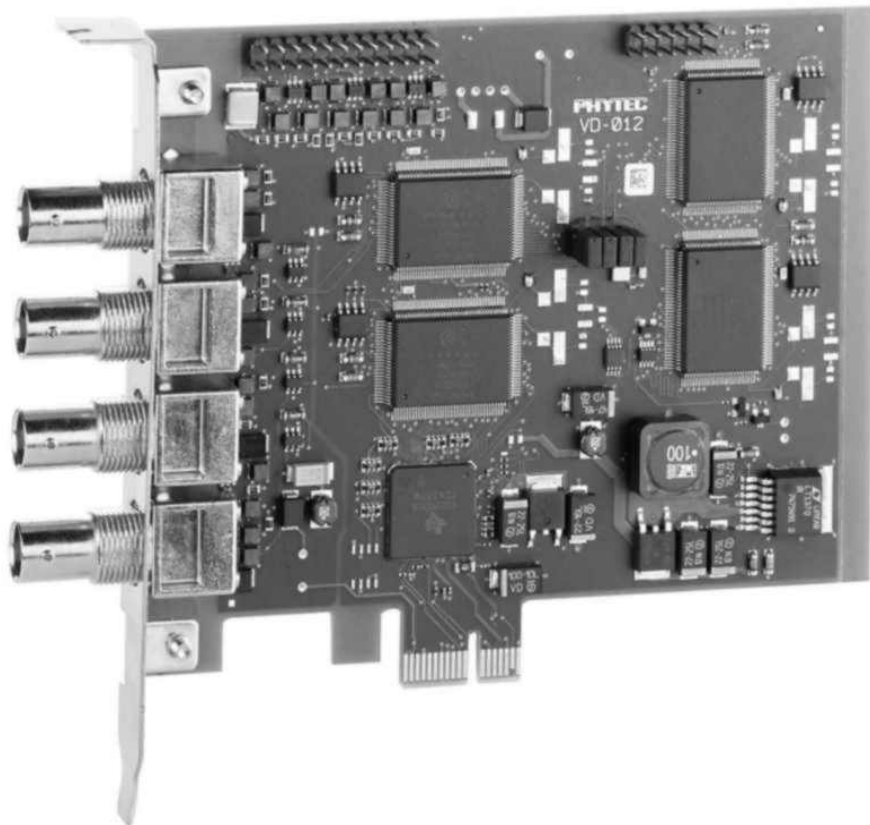


PCI Grabber-4x4



Hardware-Manual

Ausgabe November 2009

Im Buch verwendete Bezeichnungen für Erzeugnisse, die zugleich ein eingetragenes Warenzeichen darstellen, wurden nicht besonders gekennzeichnet. Das Fehlen der © Markierung ist demzufolge nicht gleichbedeutend mit der Tatsache, dass die Bezeichnung als freier Warenname gilt. Ebenso wenig kann anhand der verwendeten Bezeichnung auf eventuell vorliegende Patente oder einen Gebrauchsmusterschutz geschlossen werden.

Die Informationen in diesem Handbuch wurden sorgfältig überprüft und können als zutreffend angenommen werden. Dennoch sei ausdrücklich darauf verwiesen, dass die Firma PHYTEC Messtechnik GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf den Gebrauch oder den Inhalt dieses Handbuches zurückzuführen sind. Die in diesem Handbuch enthaltenen Angaben können ohne vorherige Ankündigung geändert werden. Die Firma PHYTEC Messtechnik GmbH geht damit keinerlei Verpflichtungen ein.

Ferner sei ausdrücklich darauf verwiesen, dass PHYTEC Messtechnik GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf falschen Gebrauch oder falschen Einsatz der Hard- bzw. Software zurückzuführen sind. Ebenso können ohne vorherige Ankündigung Layout oder Design der Hardware geändert werden. PHYTEC Messtechnik GmbH geht damit keinerlei Verpflichtungen ein.

© Copyright 2009 PHYTEC Messtechnik GmbH, D-55129 Mainz.

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form ohne schriftliche Genehmigung der Firma PHYTEC Messtechnik GmbH unter Einsatz entsprechender Systeme reproduziert, verarbeitet, vervielfältigt oder verbreitet werden.

Informieren Sie sich:

	EUROPA	NORD AMERIKA
Adresse:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 D-55129 Mainz GERMANY	PHYTEC America LLC 203 Parfitt Way SW, Suite G100 Bainbridge Island, WA 98110 USA
Angebots Hotline:	+49 (800) 0749832 order@phytec.de	+1 (800) 278-9913 sales@phytec.com
Technische Hotline:	+49 (6131) 9221-31 support@phytec.de	+1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	+1 (206) 780-9135
Web Seite:	http://www.phytec.de	http://www.phytec.com

2. Auflage November 2009

1	Einleitung	6
1.1	Übersicht über die Karte	6
1.2	Übersicht über die Modellvarianten.....	8
2	Lieferumfang / Technische Daten	10
2.1	Zubehör	11
2.2	Technische Daten VD-012(-X1)(-X2)	12
2.3	Anwendungsgebiete und Sicherheitshinweise	16
2.4	Hinweise zur CE-Konformität und Störsicherheit	17
2.5	Adressen und Ressourcen.....	19
2.6	Anschlussbelegung der Buchsen.....	21
2.6.1	Composite-Eingänge	23
2.6.2	S-Video-Anschluss.....	24
2.7	Jumper auf Stiftleiste X900.....	25
2.8	Der Erweiterungsanschluss Option-Port	26
2.9	I ² C-Schnittstelle.....	27
3	Einbau und Installation der Grabberkarte	28
3.1	Einsetzen der Grabberkarte	28
3.2	Installieren der Treiber	30
3.2.1	Zusätzliche Treiber (optional).....	32
3.3	Installation des Demoprogramms.....	33
4	Das Anschließen der Videoquellen	34
4.1	Anschlussmöglichkeiten von Videoquellen	36
4.1.1	Das S-Video Kabel.....	38
4.1.2	Das Composite Kabel.....	38
4.2	Zusatzplatine VZ-012.....	39
4.3	Übersicht aller Videoeingangskanäle.....	42
5	Inbetriebnahme des Grabbers mit Demoprogramm	44
5.1	Paralleler Bildeinzug mehrerer Bildquellen.....	51
5.2	Buslast bei Verwendung mehrerer Decoder	53
5.3	Beschreibung des Demoprogramms.....	56
5.4	Bildkontrolle.....	62
5.5	Weiter Funktionen unter <i>Image</i>	63
5.6	Fadenkreuze einblenden.....	64
5.7	Die Sonderfunktionen	64
5.8	Bilder speichern, Programm beenden	71
5.9	Inbetriebnahme unter Linux	71
6	Treiber-Software	73
6.1	Technische Grundlagen.....	74
6.1.1	Aufbau des pciGrabber-4x4	74
6.1.2	Videosignal und Digitalisierungsvorgang.....	78
6.1.3	Farübertragung und Farbspeicherung	81
6.1.4	Datenspeicherung durch DMA und RISC-Programm	83

6.2	Treiber für Microsoft Windows	89
6.2.1	Voraussetzungen	90
6.2.2	Einrichten des Gerätetreibers für Windows NT4.0	91
6.2.3	Einrichten des Gerätetreibers für Windows 2000 / XP / VISTA / 7.....	94
6.2.4	Anwendung der DLL	95
6.2.5	Anwenden der Windows NT4.0 TM / Windows 2000 TM / XP / VISTA / 7 - DLL	96
6.2.6	Programmierung unter Delphi	97
6.2.7	Beschreibung der in der DLL vorhandenen Funktionen	99
7	Störungssuche.....	156
Index	160

Bild 28:	Menü „Image Settings“	56
Bild 29:	Entstehung des Vollbildes aus zwei Halbbildern mit je 7 Zeilen	59
Bild 30:	Kammeffekt bei schnell bewegten Objekten	60
Bild 31:	Fenster zur Bildkontrolle	62
Bild 32:	Histogramm	64
Bild 33:	Color Meter	65
Bild 34:	Arithmetics Menü	67
Bild 35:	Normierungswert-Wahl	68
Bild 36:	Anzahl der Bilder	68
Bild 37:	Option Port Menü	69
Bild 38:	Option Port Menü (Jumper)	70
Bild 39:	Blockschaltbild VD-012, VD012-X1	74
Bild 40:	Blockschaltbild VD-012-X2	75
Bild 41:	Blockschaltbild eines Video-Decoders	75
Bild 42:	Zeilensprungverfahren (Beispiel mit 9 Zeilen)	78
Bild 43:	Halb- und Vollbilder	79
Bild 44:	Kammeffekt bei bewegten Objekten im Vollbild-Modus	80
Bild 45:	Übersicht über Pixel und Kontrolldatenfluß	86
Bild 46:	Verzeichnisstruktur der Windows Treiber	89
Bild 47:	Windows NT Registrierungs-Editor	92
Bild 48:	Hinzufügen eines Gerätetreiber-Eintrages	92
Bild 49:	Konfigurieren des Treibers	93
Bild 50:	Skalierung und Ausschnittsbildung	125
Bild 51:	Bild zur Skalierung: alle Werte gleich bis auf ppl	126
Bild 52:	Farbformate des pciGrabber-4x4	132
Bild 53:	Rückgabewert der Funktion data_present	138
Bild 54:	Timing Diagramm der Rückgabe-Parameters von data_present()	139

Tabellenverzeichnis

Tabelle 1: Belegung der BNC – Buchsen, Modell VD-012 23

Tabelle 2: Belegung der Mini-DIN – Buchsen, Modell VD-012-X1 23

Tabelle 3: Belegung der Buchsen, Modell VD-012-X2 24

Tabelle 4: Belegung des Option-Ports 26

Tabelle 5: Beschaltung der I²C-Schnittstelle am Option Port..... 27

Tabelle 6: Anzahl der möglichen Zusatzplatinen..... 40

Tabelle 7: Einstellung des Eingangskanals..... 49

Tabelle 8 Datenraten PAL 25fps..... 54

Tabelle 9 Datenraten NTSC 30fps 54

Tabelle 10: Speicherbedarf eines Pixel in den einzelnen Modi 134

1 Einleitung

Vielen Dank für den Erwerb eines pciGrabbers-4x4 der Firma PHYTEC Messtechnik GmbH. In diesem Benutzerhandbuch wird Ihnen die Installation des Framegrabbers beschrieben und auf die Treiber-Software eingegangen.

1.1 Übersicht über die Karte

Der pciGrabber-4x4 ist eine Framegrabber-Karte für den PCI-Express-Bus (x1). Er besitzt mehrere, voneinander unabhängige Video-Decoder, so dass mehrere Eingangskanäle parallel (gleichzeitig) digitalisiert werden können.

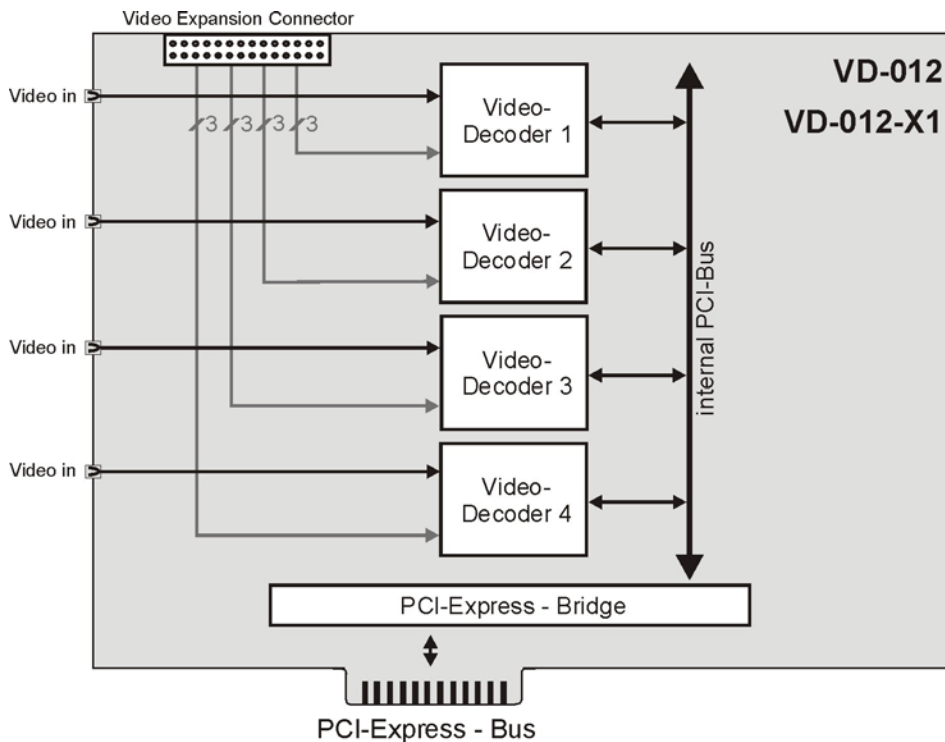


Bild 1: Übersicht pciGrabber-4x4, Modelle VD-012 und VD-012-X1

Jeder der Video-Decoder verhält sich wie eine eigene kleine Framegrabberkarte. Er kann ein Video-Signal in Echtzeit digitalisieren und im Speicher des Computers ablegen.

Der pciGrabber-4x4 ist wahlweise mit vier Composite-Eingängen (BNC-Buchsen) oder vier S-Video-Eingängen (Mini-DIN-Buchsen) erhältlich.

Hinweis:

Als Erweiterungsmöglichkeit können an jeden Decoder bis zu vier Kameras angeschlossen werden. Es kann jedoch pro Decoder nur eine Kamera gleichzeitig digitalisiert werden.

Zum Anschluss der zusätzlichen Kameras sind Adapterplattenen erhältlich.

Zusätzliche Kamera-Anschlüsse sind immer Composite-Eingänge (BNC-Buchsen), unabhängig davon, welche Eingangsvariante die Hauptanschlüsse auf der Grabberkarte haben.

Der pciGrabber-4x4 ist auch in einer Ausführung mit nur zwei Video-Decodern erhältlich (VD-012-X2):

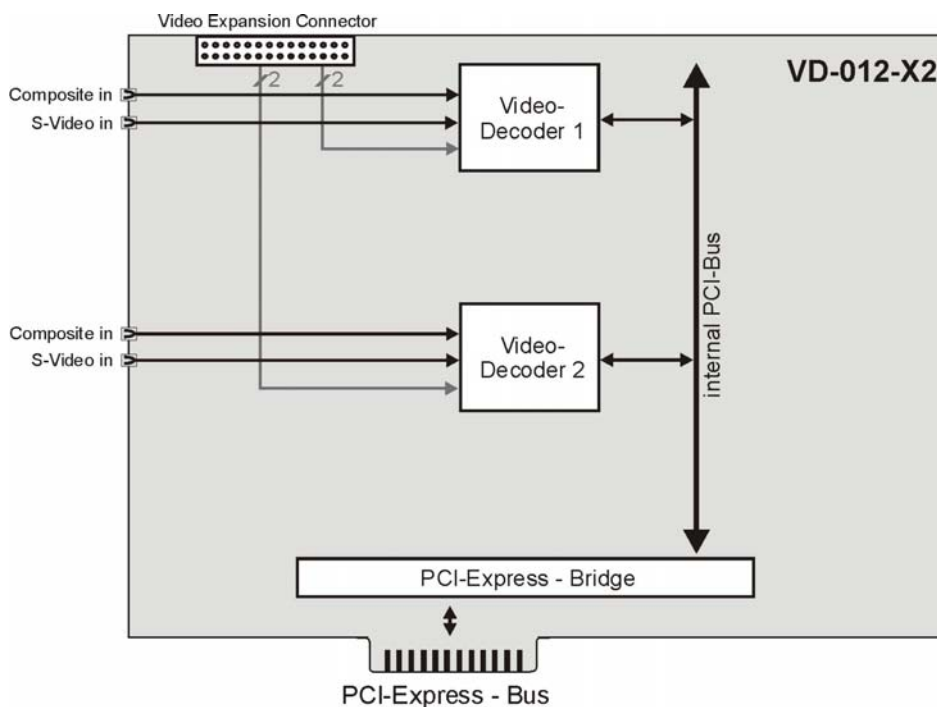


Bild 2: Übersicht pciGrabber-4x4, Modell VD-012-X2

Diese Ausführung besitzt pro Decoder je einen Composite- und einen S-Video-Eingang. Sie können per Software auswählen, welchen Eingangstyp Sie verwenden möchten.

Außerdem besitzt jeder Decoder zwei zusätzliche Composite-Eingänge (siehe Hinweis oben).

1.2 Übersicht über die Modellvarianten

Es existieren verschiedene Modellvarianten des pciGrabber-4x4. In der folgenden Tabelle ist eine Übersicht der Varianten zu sehen, für die dieses Handbuch geschrieben wurde:

Typ	Artikelnummer	Video-Decoder	Eingänge
pciGrabber-4x4	VD-012	4	4x BNC
pciGrabber-4x4	VD-012-X1	4	4x S-Video
pciGrabber-4x4	VD-012-X2	2	2x BNC 2x S-Video (umschaltbar)

Teil 1

Installation und Inbetriebnahme

2 Lieferumfang / Technische Daten

Zum Lieferumfang des pciGrabber-4x4 gehören:

- pciGrabber-4x4 (PCI Express-Steckkarte)
- Installations-CD mit
 - Demosoftware (Windows XP, NT4.0, 2000, VISTA und Windows 7)
 - Treiber-Software für Windows XP, NT4.0, 2000, VISTA und Windows 7
 - Twain-Treiber für Applikationen mit Twain-Schnittstelle
- dieses Manual

2.1 Zubehör

Bei PHYTEC können Sie folgendes Zubehör zu den pciGrabbern erhalten:

- S-Video-Anschlusskabel für den Anschluss von Farbkameras mit 4poliger Mini-DIN-Buchse (S-Video-Ausgang): Länge ca. 2 m – Bestell-Nr. WK051
- BNC-Anschlusskabel für den Anschluss von Kameras mit BNC-Steckern: Bestell-Nr.: WK058 (2m) / WK039 (10m)



Bild 3: Zubehörkabel

2.2 Technische Daten VD-012(-X1)(-X2)

Abmessungen: 125 x 90 x 20 mm zuzügl. Frontblende und Slot
120 x 90 x 20 mm ohne Nutzenrand

Datenbus: x1 PCI Express-Bus
(PCI Express Base Spec. Rev. 1.0a compliant)

Versorgung: +3,3 V (ca. 950 mA Idle, ca. 1300 mA Digitizing)
(werden dem PCI Express-Bus entnommen)

Anzahl

Decoder: Modell VD-012: 4 Decoder

Modell VD-012-X1: 4 Decoder

Modell VD-012-X2: 2 Decoder

Eingänge:

Modell VD-012:

4 Composite-Videoeingänge, 75 Ω , 1V_{ss})

*optional 12 Composite-Videoeingänge, 75 Ω , 1V_{ss}
(mit Erweiterungskarte VZ-012)*

Modell VD-012-X1:

4 S-Video-Eingang 75 Ω (0,7 V_{ss} / 0,3 V_{ss})

*optional 12 Composite-Videoeingänge, 75 Ω , 1V_{ss}
(mit Erweiterungskarte VZ-012)*

Modell VD-012-X2:

2 Composite-Videoeingänge, 75 Ω , 1V_{ss})

2 S-Video-Eingang 75 Ω (0,7 V_{ss} / 0,3 V_{ss})

*optional 4 Composite-Videoeingänge, 75 Ω , 1V_{ss}
(mit Erweiterungskarte VZ-012)*

Videoformate: PAL (B,G,H,I), NTSC (M)
bzw. entsprechende CCIR-Formate monochrom

Synchronisation: Composite-Sync bzw. Sync auf Y-Signal,
externe Synchronisation nicht möglich

Datenformate: 16 Mio. Farben: RGB32, RGB24, YCrCb 4:2:2,
YCrCb 4:1:1
64.000 Farben: RGB16
32.000 Farben: RGB15
256 Graustufen: Y8 Gray Scale

Bildauflösung: maximal 720 x 576 Pixel (PAL)
bzw. 640 x 480 Pixel (NTSC)
frei skalierbar in X- und Y-Richtung bis 14:1

**Bildeinzug
je Decoder:** Halbbild: 20 ms
Vollbild: 40 ms
Bildtransfer in den Arbeitsspeicher in Echtzeit
(Busmaster-Transfer)

**verwendete
Ressourcen:** 4 kByte Arbeitsspeicher (Registerbereich)
je Decoder
4 kByte Arbeitsspeicher
PCI Express-to-PCI-Bridge

Bildkorrektur: Gammakorrektur (zuschaltbar)
Helligkeit (+/- 50 %)
Kontrast (0 % ... 235 %)
Farbsättigung (U: 0...201 %, V: 0...283 %)
Farbton (+/- 90°, nur NTSC)

**Bildspeicher
je Decoder:** 630 Byte FIFO on-board,
Echtzeit-Speicherung im PC-Arbeitsspeicher
Even-/Odd-Speicher getrennt oder
gemeinsamer Vollbildspeicher (wählbar)

Ports: 8-bit Parallel-I/O, TTL-Pegel (Multipurpose)

Parameter	Symbol	Min	Max
Input High Voltage	V_{IH}	2,0 V	5 V
Input Low Voltage	V_{IL}	-0,5 V	0,8 V
Output High Voltage	V_{OH}	2,4 V	-
Output Low Voltage	V_{OL}	-	0,4 V
Input Low Current	I_{IL}	-	-70 μ A
Input High Current	I_{IH}	-	70 μ A

1 I²C-Schnittstelle (Master)

Parameter	Symbol	Min	Max
Übertragungsrate ¹	f_{I2C}	99,2 kHz	396,8 kHz
Input High Voltage	V_{IH}	3,5 V	5 V
Input Low Voltage	V_{IL}	-0,5 V	1,5 V
Hysteresis	V_{hys}	0,2 V	
Input High Current	I_{IH}	-	10 μ A
Input Low Current	I_{IL}	-	-10 μ A
Output Low Voltage	V_{OL}	-	0,4 V

¹ : die beiden Frequenzen sind softwaremäßig umschaltbar

Anschlüsse: Modell VD-012

- 4 x BNC-Buchse: Composite-Eingang
- Pfostenleiste 2x12:
(nicht an der Frontblende) 12 Composite-Eingänge
- Pfostenleiste 2x6:
(nicht an der Frontblende) GPIO-Port, 8 x TTL I/O
I²C-Schnittstelle
- Pfostenleiste 2 x 4:
(nicht an der Frontblende) Jumper

Modell VD-012-X1

- 4 x Mini-DIN-Buchse: S-Video-Eingang
- Pfostenleiste 2x12:
(nicht an der Frontblende) 12 Composite-Eingänge
- Pfostenleiste 2x6:
(nicht an der Frontblende) GPIO-Port, 8 x TTL I/O
I²C-Schnittstelle
- Pfostenleiste 2 x 4:
(nicht an der Frontblende) Jumper

Modell VD-012-X2

- 2 x Mini-DIN-Buchse: S-Video-Eingang
- 2 x BNC-Buchse: Composite-Eingang
- Pfostenleiste 2x12:
(nicht an der Frontblende) 4 Composite-Eingänge
- Pfostenleiste 2x6:
(nicht an der Frontblende) GPIO-Port, 8 x TTL I/O
I²C-Schnittstelle
- Pfostenleiste 2 x 4:
(nicht an der Frontblende) Jumper

2.3 Anwendungsgebiete und Sicherheitshinweise

Achten Sie beim Einsatz des pciGrabber-4x4 auf die Einhaltung der spezifizierten Betriebsbedingungen. Lesen Sie vor der Inbetriebnahme diese Anleitung sorgfältig.

- Der pciGrabber-4x4 dient zur Digitalisierung der Videosignale von Standard-TV-Kameras. Es können Signale von Composite-Videokameras verarbeitet werden, die den Normen CCIR B,G,H,I und Unternorm CCIR B,G,H,I/PAL entsprechen. Zusätzlich können Signale nach CCIR M/NTSC eingespeist werden. Die Kamerasignale können auch nach Luma- und Chromaanteil getrennt gemäß der S-Video-Norm zugeführt werden. Die Digitalisierung der Bilder erfolgt in Echtzeit. Die Übertragung der Bilddaten erfolgt über den PCI Express-Bus. Die effektive Transferrate muss ausreichend für die angeforderte Bilddatenmenge sein, ansonsten ist der Verlust von Bilddaten möglich. Auf die maximale Transferrate wird in Kapitel 5.1 eingegangen.
- Der pciGrabber-4x4 ist für die Verwendung in Standard-PCs bestimmt. Darunter werden Bürocomputer mit Gehäuse in handelsüblicher Ausführung verstanden. Der pciGrabber-4x4 muss in einen PCI Express-Slot eingesteckt werden. Der Grabber muss über die Frontblende / das Gehäuseblech eine zuverlässige Verbindung mit dem Gehäusemantel und PE (Schutzleiter) haben. Das Gerät ist für den Einsatz in trockener und sauberer Betriebsumgebung konzipiert. Bei Verwendung in Maschinen, industrieller Umgebung und Geräten abweichend von der üblichen PC-Konstellation ist zu prüfen, ob zusätzliche Maßnahmen zur Einhaltung von Sicherheits- und Funkstörnormen sowie zur Aufrechterhaltung der Betriebssicherheit erforderlich sind.
- Die Verwendung des Geräts in sicherheitsrelevanten Bereichen, in der Luft-/Raumfahrt und in kerntechnischen oder militärischen Anwendungen bedarf unserer Prüfung und Zustimmung.

- Bei gewerblicher Anwendung sind die Unfallverhütungsvorschriften des Verbandes der gewerblichen Berufsgenossenschaften für elektrische Anlagen und Betriebsmittel zu beachten.
- Vor der Inbetriebnahme ist generell zu prüfen, ob das Gerät für den vorgesehenen Anwendungsfall und Einsatzort geeignet ist. Im Zweifelsfall sind unbedingt Rückfragen bei Fachleuten, Sachverständigen oder dem Hersteller erforderlich.
- Das Produkt ist vor starken Erschütterungen und Vibrationen zu schützen. Erforderlichenfalls ist eine Federung oder Polsterung vorzusehen, die jedoch nicht die Belüftung des Geräts behindern darf.

Eine eventuell notwendige Reparatur darf nur vom Fachmann unter Verwendung von Originalbauteilen durchgeführt werden. Beim Anschluss des Geräts, dürfen nur zugelassene und geprüfte Anschlusskabel verwendet werden. Es muss auf korrekte Abschirmung und Entstörung der Kabel geachtet werden.

2.4 Hinweise zur CE-Konformität und Störsicherheit

Der pciGrabber-4x4 entspricht im Auslieferungszustand den CE-Bestimmungen für Haushalt-, Büro-, Gewerbe- und Industriebereich. Veränderungen des Grabbers, die nicht vom Hersteller autorisiert sind, führen zum Erlöschen der CE-Zulassung.

Achtung!

CE-Konformität wird nur bei Verwendung der zugelassenen Anschlusskabel gewährleistet. Diese Kabel sind im Zubehörprogramm zum pciGrabber-4x4 erhältlich (siehe *Kapitel 2*). Bei Benutzung anderer als den von PHYTEC spezifizierten Kabeln muss der Anwender CE-Konformität sicherstellen. Zur Verwendung der S-Videoleitungen WK-051 müssen Klappferrite vom Typ Würth Nr. 742 716 32 verwendet werden.

Möchten Sie eigene Kabel an den pciGrabber-4x4 anschließen, so beachten Sie bitte, dass diese Kabel mit einem Klappferrit bzw. einer gleichwertigen Entstörmaßnahme auszurüsten sind. Der Ferrit ist in einer Entfernung von ca. 5 cm zum Grabber anzubringen, und das Kabel ist zweimal durch ihn hindurchzuführen (eine Schleife).

Für Videoleitungen eignet sich z.B. auch der Klappferrit Nr. 742 716 32 der Fa. Würth, Kupferzell.

Um eine optimale Schirmwirkung zu erzielen, muss der Leitungsschirm mit dem Steckerkragen flächig verbunden werden.

Der pciGrabber-4x4 wurde in einer Standard-PC-Umgebung geprüft. Soll das Produkt in einer abweichenden Betriebsumgebung eingesetzt werden, ist gegebenenfalls zu prüfen, ob weitere Schirm- und Entstörmaßnahmen erforderlich sind.

Achtung!

Beachten Sie, dass starke Störspitzen auf den Videosignalen oder auf dem Kabelschirm die Eingangsstufe des pciGrabber-4x4 zerstören können.

In stark gestörten Industrieumgebungen und bei Verwendung langer Zuleitungen sind daher unbedingt zusätzliche Entstörmaßnahmen vorzusehen.

Bei Verlegung langer Videoleitungen oder dem Einbau der Bildverarbeitungskomponenten in Maschinen oder Anlagen können Mantel- bzw. Potentialausgleichsströme auftreten, die durch geeignete Vorrichtungen von den Eingängen des pciGrabber-4x4 fernzuhalten sind.

PHYTEC kann keine Haftung für Schäden übernehmen, die durch unsachgemäßen Anschluss der Signalquellen entstanden sind.

2.5 Adressen und Ressourcen

Je nach Variante belegen die Decoder der pciGrabber-4x4 einen Speicherbereich von 4x4 kByte (VD-012, VD-012-X1) oder 2x4 kByte (VD-012-X2) im Arbeitsspeicher des PCs für die lokalen Register. Der Adressbereich wird durch das BIOS des Rechners automatisch festgelegt, so dass keine Jumper o.ä. zu setzen sind.

Es können mehrere pciGrabber-4x4 in einem System installiert werden. Die Karten werden vom BIOS automatisch auf verschiedene Adressen konfiguriert.

Es kann nicht vorab bestimmt werden, welche Karte auf welche Adresse konfiguriert wird. Die Basisadresse jeder Karte kann über das PCI-BIOS ermittelt werden. Beim pciGrabber-4x4 ermittelt die Treibersoftware über das BIOS die Adresse und setzt sie in eine „Gerätenummer“ um. Es kann über den Treiber ermittelt werden, wie viel Karten sich im System befinden und dann jede Karte über ihre individuelle Gerätenummer angesprochen werden.

Jeder Decoder des pciGrabber-4x4 wird vom BIOS bzw. vom Betriebssystem als eigenständiges Videoaufnahmegerät behandelt. Somit wird also z.B. ein pciGrabber-4x4 (VD012) mit vier Decodern als vier Geräte abgebildet.

Jedes dieser virtuellen Geräte können Sie unabhängig von den anderen konfigurieren und betreiben.

Es kann nicht sicher vorhergesagt werden, welche Karte / welcher Decoder mit welcher Gerätenummer verbunden wird. Dies wird allein vom PCI-BIOS und der Architektur des PC-Motherboards bestimmt.

Der pciGrabber-4x4 kann bei einer ganzen Reihe von Ereignissen und Betriebszuständen einen Interrupt auslösen. Die Ursache des Interrupts kann über das Interrupt-Statusregister des Grabbers ermittelt werden.

Achtung!

Da mehrere Karten / Decoder den gleichen Interrupt /INTA benutzen, muss zunächst geprüft werden, welche Karte den /INTA ausgelöst hat.

2.6 Anschlussbelegung der Buchsen

Hinweis:

Die nachfolgende Beschreibung der Anschlüsse der Grabber ist als technische Referenz zu verstehen.

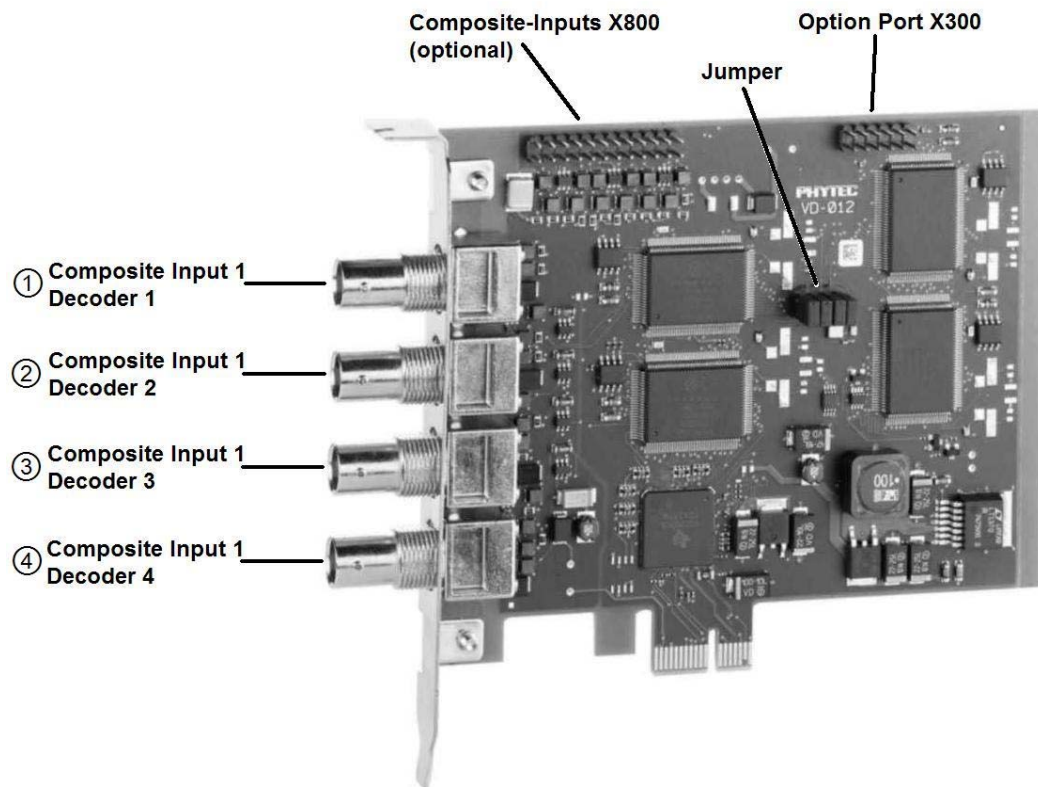


Bild 4: Anschlüsse des pciGrabber-4x4 (VD-012)

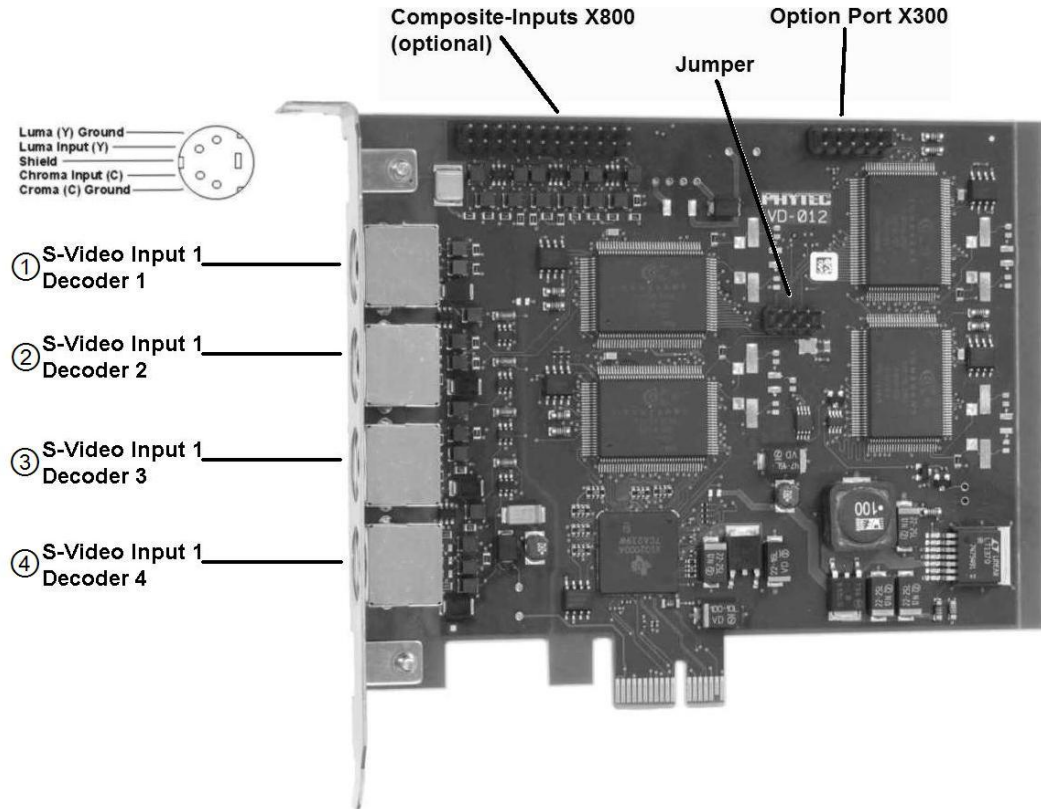


Bild 5: Anschlüsse des pciGrabber-4x4 (VD-012-X1)

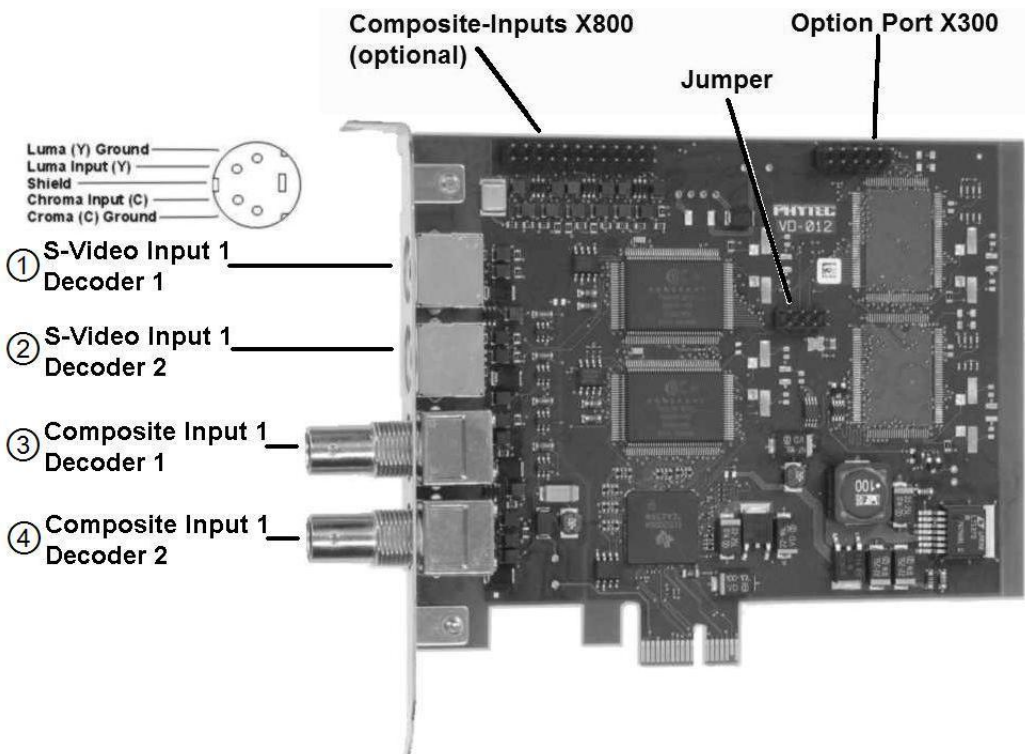


Bild 6: Anschlüsse des pciGrabber-4x4 (VD-012-X2)

2.6.1 Composite-Eingänge

Als Videoquellen eignen sich alle Composite-Quellen mit einem Signalhub von 1 V_{ss} und einem Abschlusswiderstand von 75 Ω. Die unterstützten Videonormen entnehmen Sie bitte *Kapitel 2.2*.

- **Version VD-012**

Es stehen vier Composite-Eingänge zur Verfügung, die über die BNC-Buchsen an der Front zugänglich sind (Buchse ①, ②, ③ und ④). Auf der Stiftleiste X800 sind zusätzlich weitere 12 Videoeingänge vorhanden. Diese Eingänge können über die Zusatzplatine VZ-012 benutzt werden (siehe *Kapitel 4.2*).

- **Version VD-012-X1**

Auf der Stiftleiste X800 sind 12 zusätzliche Composite-Eingänge vorhanden. Diese Eingänge können über die Zusatzplatine VZ-012 benutzt werden (siehe *Kapitel 4.2*).

- **Version VD-012-X2**

Es stehen zwei Composite-Eingänge zur Verfügung, die über die BNC-Buchsen an der Front zugänglich sind (Buchse ① und ②). Auf der Stiftleiste X800 sind zusätzlich weitere 4 Videoeingänge. Diese Eingänge können über die Zusatzplatine VZ-012 benutzt werden (siehe *Kapitel 4.2*).

Die Zuordnung der Eingänge zu den Kanalnummern ist wie folgt:

pciGrabber-4x4 (VD-012)			
BNC ①	BNC ②	BNC ③	BNC ④
Composite Input 1	Composite Input 1	Composite Input 1	Composite Input 1
Decoder 1	Decoder 2	Decoder 3	Decoder 4

Tabelle 1: Belegung der BNC – Buchsen, Modell VD-012

pciGrabber-4x4 (VD-012-X1)			
Mini-DIN ①	Mini-DIN ②	Mini-DIN ③	Mini-DIN ④
Composite Input 1	Composite Input 1	S-Video Input 1	S-Video Input 1
Decoder 1	Decoder 2	Decoder 3	Decoder 4

Tabelle 2: Belegung der Mini-DIN – Buchsen, Modell VD-012-X1

pciGrabber-4x4 (VD-012-X2)						
BNC ①		BNC ②		Mini-DIN ③		Mini-DIN ④
Composite	Input	Composite	Input	S-Video Input 1		S-Video Input 1
1		1		Decoder 1		Decoder 2
Decoder 1		Decoder 2				

Tabelle 3: Belegung der Buchsen, Modell VD-012-X2

PHYTEC bietet Ihnen als Zubehör fertig konfektionierte Anschlusskabel an, die die Signaleinspeisung über BNC- und Mini-DIN-Stecker ermöglichen (siehe Kapitel 2.1).

2.6.2 S-Video-Anschluss

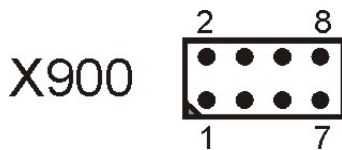
Der Vorteil des S-Video-Systems ist die getrennte Führung von Helligkeits- und Farbsignal. Dies verhindert Moiréstörungen an feinen Bildstrukturen und verbessert so die tatsächliche Auflösung bei Farbbildern.

S-Video-Quellen können an die Varianten VD-012-X1 und VD-012-X2 des pciGrabber-4x4 angeschlossen werden. Die Belegung kann der Tabelle 2 und Tabelle 3 entnommen werden.

- Die Buchsen sind entsprechend der *S-Video*-Norm beschaltet (vgl. Bild 5 und Bild 6). Der Anschluss einer Kamera kann mit einem handelsüblichen *S-Video*-Kabel erfolgen.

2.7 Jumper auf Stiftleiste X900

Sind in einem System mehrere Grabber der gleichen Variante installiert, werden diese unter der gleichen Bezeichnung geführt. Grabber/Decoder bekommen zwar vom Treiber unterschiedliche Adressen, aber es ist nicht möglich festzustellen, welche Adresse zu welchem Framegrabber gehört. Über die Jumper auf der Stiftleiste X900 können nun mehrere Framegrabber der gleichen Variante in einem System unterschieden werden.



Codierjumper (X900)

Pins	Jumper
1-2	Jumper JP1
3-4	Jumper JP2
5-6	Jumper JP3
7-8	Jumper JP4

2.8 Der Erweiterungsanschluss Option-Port

Der Erweiterungsport stellt dem Anwender 8 digitale I/O-Leitungen und ein I²C-Schnittstelle zur Verfügung. Die Signale sind auf einen Pfostenstecker mit 6 x 2 Pins geführt. Die Leiste ist mit X300 bezeichnet, Pin 1 befindet sich links unten. *Bild 7* zeigt die Belegung der Steckleiste.

Beachten Sie, dass der +5V Versorgungspin X300-1 mit max. 100 mA belastet werden darf.

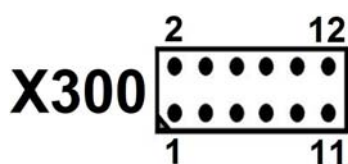


Bild 7: Belegung des Erweiterungssteckers

Erweiterungsanschluss (Option Port, X300)			
Pin	Funktion	Pin	Funktion
1	+5V out	8	I/O 6
2	I/O0	9	I/O 7
3	I/O1	10	I ² C SCL
4	I/O2	11	I ² C SDA
5	I/O3	12	GND
6	I/O4		
7	I/O5		

Tabelle 4: Belegung des Option-Ports

2.9 I²C-Schnittstelle

Über die I²C-Schnittstelle können externe Geräte abgefragt oder angesteuert werden, die eine I²C-Schnittstelle im Slave-Modus besitzen.

Die I²C-Schnittstelle steht an der internen Stiftleiste des *Option Ports* zur Verfügung (X300). Es können mehrere I²C-Geräte an den Bus angeschlossen werden, sie müssen sich jedoch in ihren Geräteadressen unterscheiden. *Tabelle 5* zeigt die relevante Anschlussbelegung Stiftleiste X300.

X300	
Pin	Funktion
10	I ² C Bus: SCL
11	I ² C Bus: SDA
12	Ground

Tabelle 5: Beschaltung der I²C-Schnittstelle am Option Port

Hinweis:

Die I²C-Schnittstelle wird mit TTL-Pegel betrieben. Dadurch ist die maximale Leitungslänge begrenzt. Bei einem angeschlossenen Gerät können je nach eingestellter Übertragungsrates Leitungslängen von ca. 1 m – 2 m erreicht werden.

Verwenden Sie zum Anschluss nur ausreichend geschirmte Kabel.

Informationen, wie Sie die I²C-Schnittstelle in Ihrer Software ansprechen können, finden Sie im *Kapitel 6.2.7* unter der Funktionsgruppe „Daten über die I²C-Schnittstelle übertragen“.

3 Einbau und Installation der Grabberkarte

Die Grabberkarte hat die Aufgabe, die analogen Bildsignale der Kamera umzuwandeln und dem Computer bzw. der Software in digitaler Form zu präsentieren.

Wenn Sie mit dem Einbau von Steckkarten noch keine Erfahrung haben, nehmen Sie sich dafür bitte etwas Zeit.

Es ist nicht schwer, sollte aber sorgfältig ausgeführt werden.

3.1 Einsetzen der Grabberkarte

Achtung!

Der Computer muss von der Netzspannungsversorgung getrennt sein. Es muss sichergestellt sein, dass das Gerät spannungsfrei ist.

- Entfernen Sie das Gehäuse des PCs (normalerweise verschraubt).
- Wählen Sie einen freien PCI Express-Steckplatz.

Ziehen Sie das Handbuch Ihres Rechners / Ihres Motherboards zur Rate, falls Sie nicht sicher sein sollten welchen Slot Sie benutzen müssen.

- Entfernen Sie die Blende am PC-Gehäuse, die sich vor dem ausgewählten Slot befindet (abschrauben oder ausbrechen).
- Setzen Sie nun den pciGrabber-4x4 wie im *Bild 8* dargestellt mit der Anschlussblende nach außen vorsichtig in den Steckplatz ein. Die Karte sollte sicher einrasten.
- Drücken Sie die Karte nicht mit Gewalt in den Slot, Motherboard bzw. Karte könnten Schaden nehmen!
- Achten Sie darauf, dass die goldenen Kontaktstreifen der Karte exakt mit den Kontaktfedern des Steckplatzes übereinstimmen.

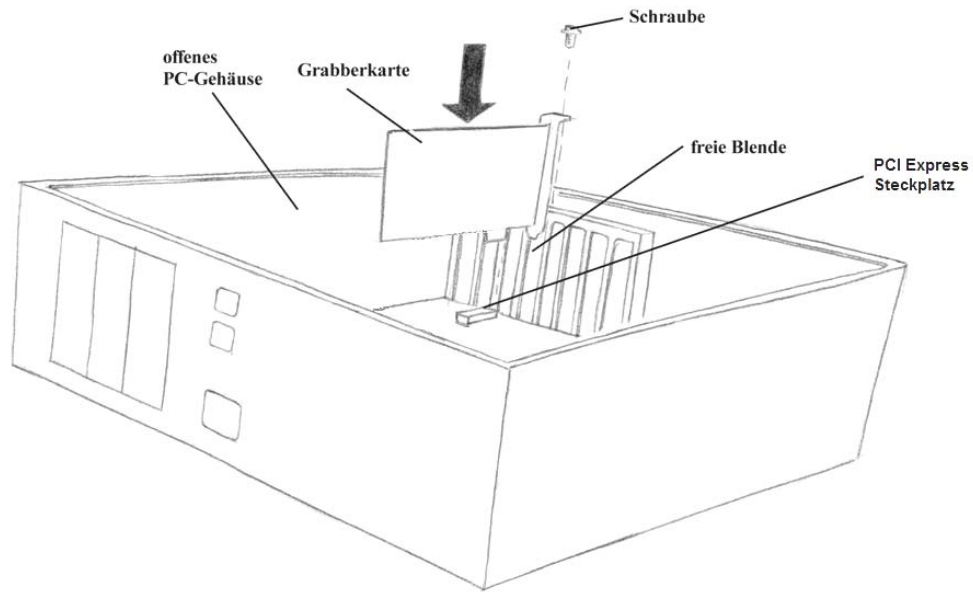


Bild 8: Einsetzen der Karte im PCI-Express-Steckplatz

- Vergewissern Sie sich anschließend, dass die Karte gerade sitzt und keine benachbarten Kontakte kurzgeschlossen sind.

Achtung!

Aus Stabilitätsgründen und damit die Karte eine sichere Masseverbindung zum Rechnergehäuse bekommt, befestigen Sie die Karte mit einer Schraube am Gehäuse (siehe Bild 8).

- Schließen Sie das Rechnergehäuse wieder.

Als nächstes müssen noch der Treiber und die Demosoftware der Karte installiert werden.

Die Installation der Treiber unterscheidet sich abhängig von dem Betriebssystem, das auf dem PC installiert ist.

Folgen Sie der Beschreibung in den folgenden Abschnitten, um die Treiber- und Demosoftware zu installieren.

3.2 Installieren der Treiber

- Schließen sie den Rechner wieder an das Stromnetz an und schalten Sie in ein.
Beim Hochfahren sollte das BIOS des Rechners die Karte automatisch erkennen.

Es gibt nun zwei Möglichkeiten:

1. Das Betriebssystem erkennt die Karte und fragt Sie nach Treibern oder
2. Das Betriebssystem erkennt die Karte nicht automatisch (z.B. Windows NT) und Sie müssen die Treiber manuell installieren.

Die Treiberinstallation läuft je nach Betriebssystem folgendermaßen ab:

- **Windows 2000/XP/VISTA/7™ :**

Nachdem die Karte erkannt wurde, werden Sie gebeten, den Treiber zu installieren.

Wählen Sie in dem Fenster *Hardware Assistent* die Option *Nach dem besten Treiber für das Gerät suchen* aus und bestätigen Sie mit *OK*.

Im darauffolgenden Fenster wählen Sie *Geben Sie eine Position an*. Legen Sie die **PHYTEC Vision Utilities** CD in Ihr CD-ROM Laufwerk. Wählen Sie *Durchsuchen* und im darauf erscheinenden Fenster das CD-ROM Laufwerk aus.

Wechseln Sie den Pfad auf *pciGrab4\driver\windows*. Bestätigen Sie mit *OK*.

Es erscheint eine Liste mit den gefundenen Treibern auf der CD. Wählen Sie aus dieser Liste den *PHYTEC PCI-Grabber* aus.

Der Treiber sollte nun automatisch von CD auf Ihren Rechner installiert werden

Mit diesem Schritt haben Sie den Treiber erfolgreich installiert.

Beachten Sie bitte noch *Kapitel 3.2.1*. Gehen Sie dann zu *Kapitel 3.3*, um die Demosoftware zu installieren.

- **Windows NT4.0TM (mit ServicePack 6):**

WindowsNT erkennt die Karte nicht automatisch. Der Treiber muss manuell installiert werden. Legen Sie die **PHYTEC Vision Utilites** CD in Ihr CD-ROM Laufwerk. Wählen Sie unter WindowsNT im Hauptverzeichnis der CD das Programm *Start.exe* aus.

Im erscheinenden Fenster wählen Sie *PCI-Grabber*, dann *Install drivers* und *WindowsNT4.0*.

Folgen Sie den Anweisungen des Installationsprogrammes. Die notwendigen Treiber werden dann automatisch installiert. Bestätigen Sie im folgenden Fenster den Neustart des Rechners.

Der Rechner sollte nun ganz normal bis zum Betriebssystem hochfahren.

Mit diesem Schritt haben Sie den Treiber erfolgreich installiert.

Beachten Sie bitte noch *Kapitel 3.2.1*. Gehen Sie dann zu *Kapitel 3.3*, um die Demosoftware zu installieren.

3.2.1 Zusätzliche Treiber (optional)

Sie können noch zusätzliche Treiber von der CD-ROM installieren. Diese sind nicht unbedingt notwendig, um die Karte wie in diesem Handbuch beschrieben, zu betreiben.

Der **Twain-Treiber** ist ein Standardtreiber, der z.B. in Grafikprogrammen oder Photo- bzw. Scannerprogrammen eingesetzt werden kann, um Bilder einzulesen, und mit dem jeweiligen Programm zu bearbeiten. So werden Grabber und Kamera quasi wie ein Scanner behandelt.

Für die weitere Bedienung des Programms in Verbindung mit dem Twain-Treiber konsultieren Sie bitte das Handbuch des verwendeten Grafikprogramms.

Wenn Sie den Treiber installieren möchten:

Legen Sie die **PHYTEC Vision Utilities** CD in Ihr Laufwerk und starten Sie aus dem Hauptverzeichnis der CD die Datei *start.exe*.

Im folgenden Fenster wählen Sie *pciGrabber* aus. Sie sehen in dem darauf erscheinenden Installationsfenster unter anderem die zwei Einträge:

- ***Install Twain***

3.3 Installation des Demoprogramms

Das Demoprogramm ermöglicht Ihnen, in Verbindung mit einer Kamera, die Karte zu testen, Bildparameter zu modifizieren, und einfache Bildoperationen durchzuführen.

Die Installation des Programms:

- Legen Sie die **PHYTEC Vision Utilities** CD in Ihr CD-ROM Laufwerk.
- Dann wird das CD-ROM Laufwerk ausgewählt und das Programm *start.exe* aus dem Hauptverzeichnis der CD gestartet.
- Im erscheinenden Installationsmenü (siehe *Bild 9*) wählen Sie *pciGrabber* aus.
- Klicken Sie danach auf *Demo installieren*.

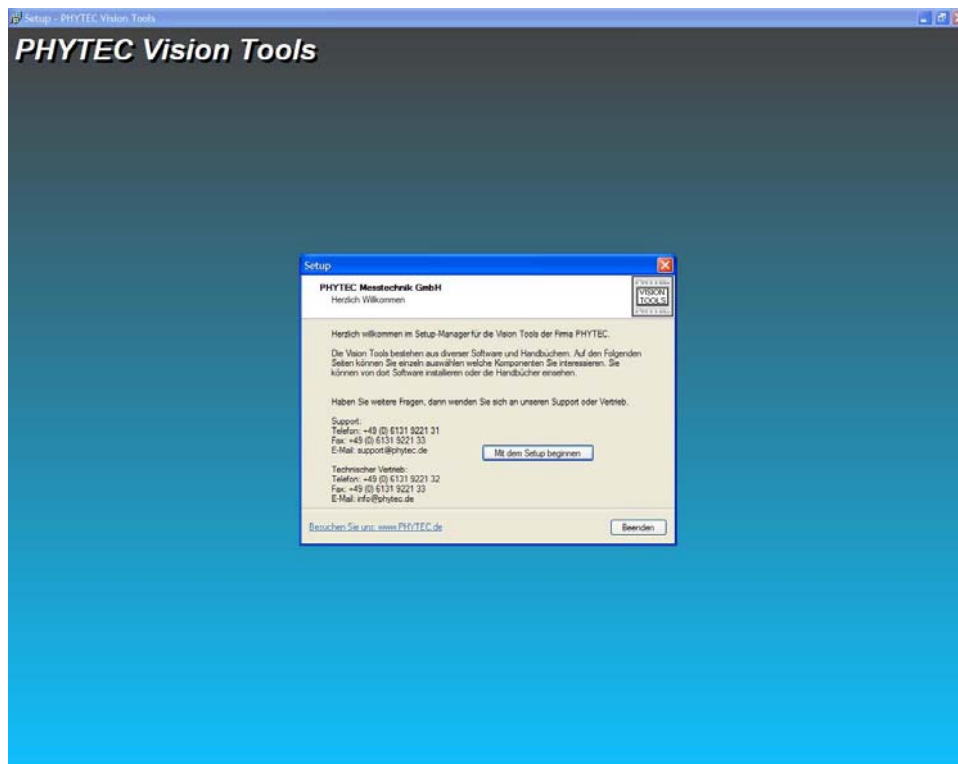


Bild 9: PHYTEC Installationsmenü

- Folgen Sie einfach den Installationsanweisungen und das Demoprogramm wird automatisch auf dem Rechner installiert.

4 Das Anschließen der Videoquellen

Beim pciGrabber-4x4 (siehe Bild 10, Bild 11 und Bild 12) haben Sie die Möglichkeit, eine oder mehrere Videoquellen anzuschließen. Dies können z.B. Videokameras oder auch Videorekorder mit den entsprechenden Ausgängen sein (Composite oder S-Video).

Je nach Grabber-Modell können bis zu 4 Composite (VD-012) oder bis zu 4 S-Video-Quellen (VD-012-X1) angeschlossen werden. An die Variante VD-012-X2 können bis zu zwei Composite und bis zu zwei S-Video-Quellen angeschlossen werden.

Die Umschaltung der einzelnen Kanäle erfolgt per Software bzw. durch das mitgelieferte Demoprogramm.

Es kann jedoch immer nur eine Kamera pro Decoder aktiv sein und Bilder liefern. Die Variante VD-012 und VD-012-X1 arbeiten mit vier Decodern und die Variante VD-012-X2 mit zwei Decodern.

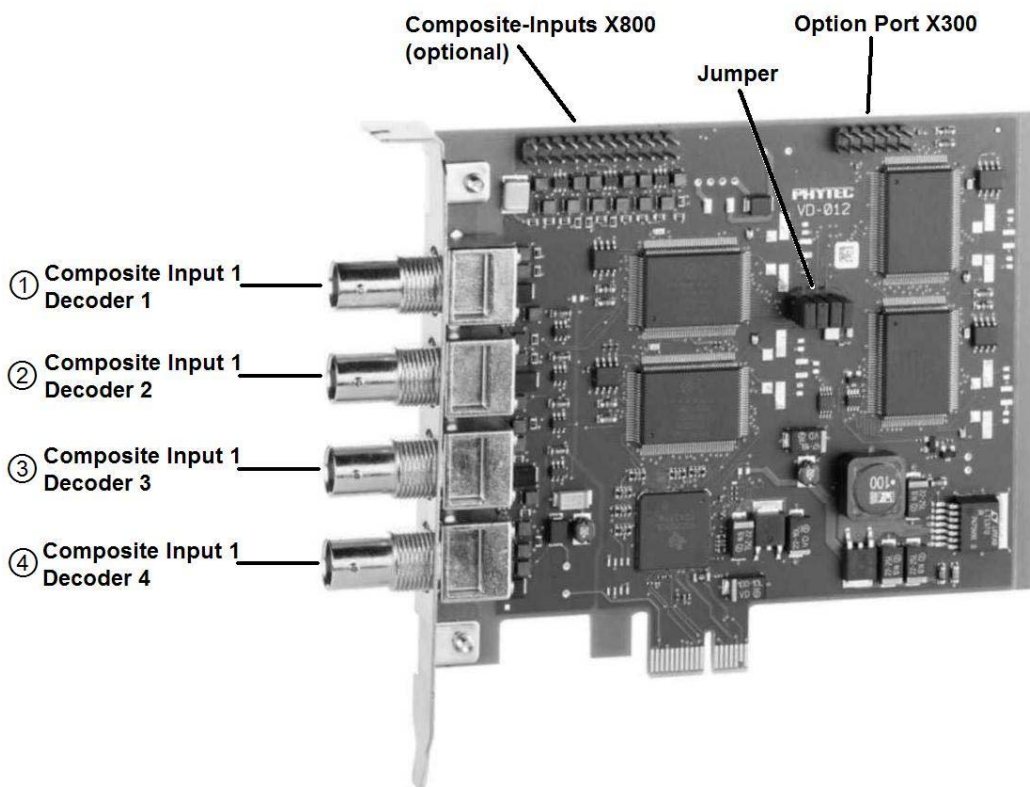


Bild 10: Übersicht der pciGrabber-4x4 Anschlüsse (VD-012)

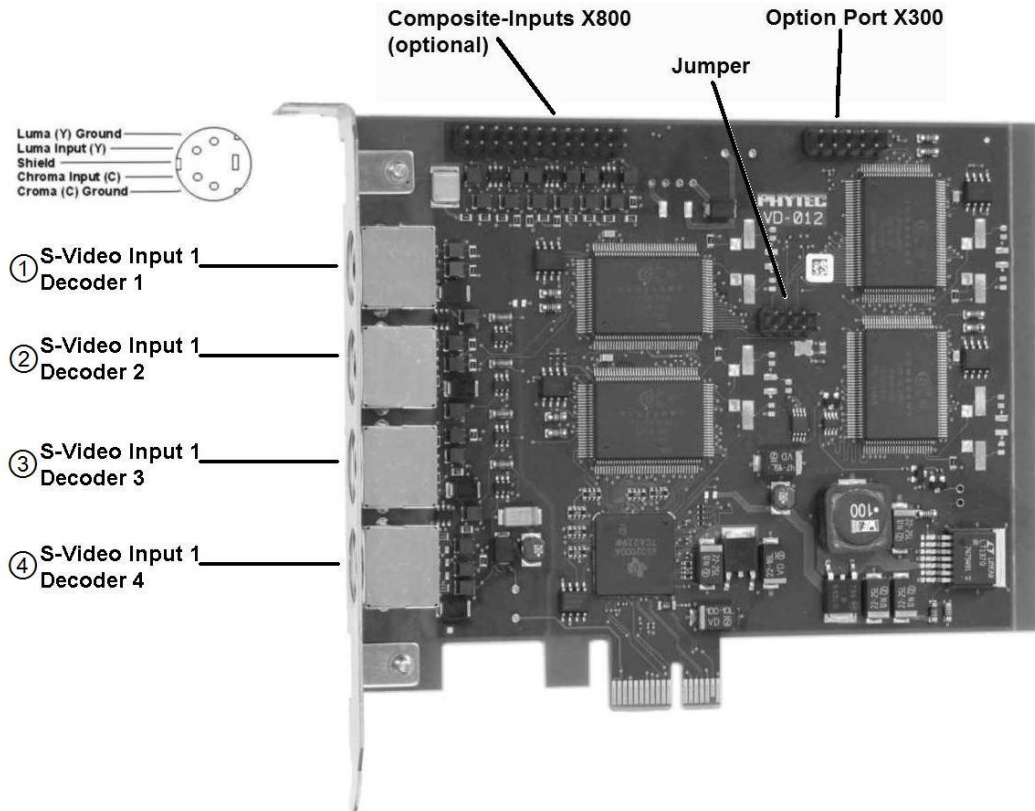


Bild 11: Übersicht der pciGrabber-4x4 Anschlüsse (VD-012-X1)

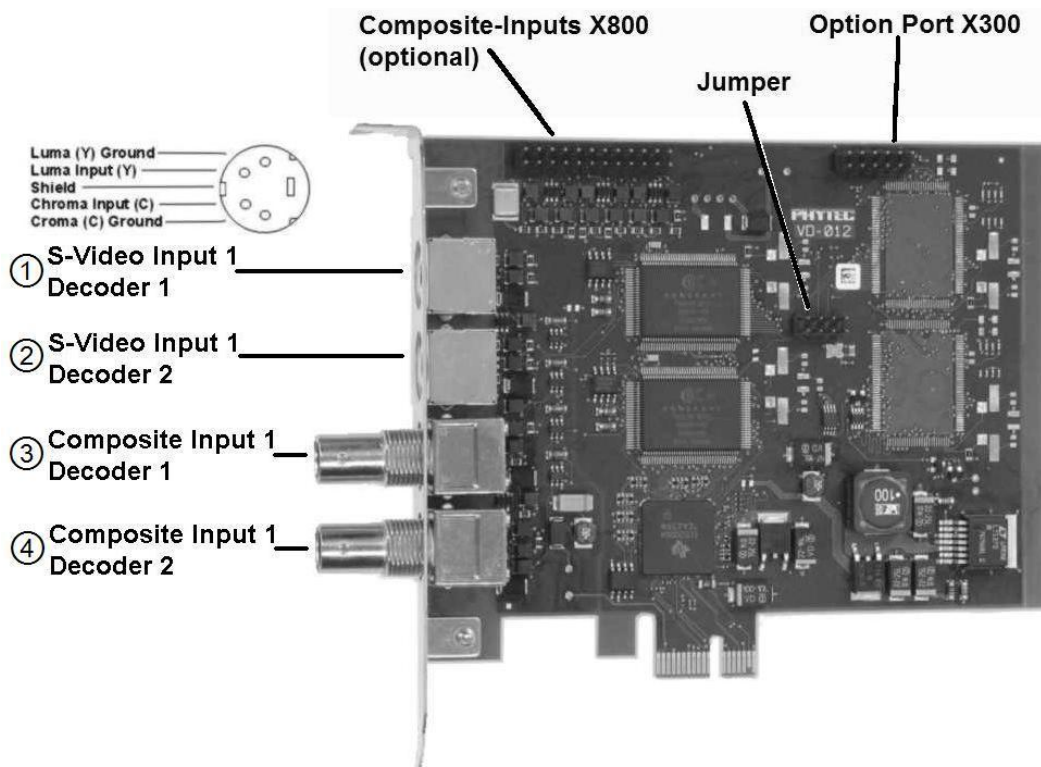


Bild 12: Übersicht der pciGrabber-4x4 Anschlüsse (VD-012-X2)

Die Composite-Eingänge liegen an den BNC-Buchsen.

S-Video-Signale können an den Mini-DIN-Buchse eingespeist werden.

Die dafür benötigten Kabel können Sie bei PHYTEC beziehen (siehe Kapitel 2.1, „Zubehör“).

Über die Stiftleiste X800 stehen weiter Videoeingänge zu Verfügung (siehe Kapitel 4.2).

Genaue Informationen zu den Anschlussbelegungen der Buchsen finden Sie vorne im Kapitel „Technische Daten“.

4.1 Anschlussmöglichkeiten von Videoquellen

Hier werden kurz verschiedene Anschlussmöglichkeiten der Videoquellen an den Grabber gezeigt.

Die gezeigten Kabel können Sie alle bei PHYTEC beziehen.

Die Legende der verwendeten Kabel beinhaltet die Bezeichnung und die PHYTEC-Bestellnummer (siehe Bild unten).



Bild 13: Video-Abschlusskabel – Legende: (Bezeichnung, PHYTEC Best.Nr.)

Ziehen Sie bitte die Anleitung ihrer Videoquelle zu Rate, um die Anschlusskompatibilität zu überprüfen.

Die Anschlussmöglichkeiten variieren abhängig vom Grabbermodell. In den beiden Bildern unten sind diese nach Modell sortiert dargestellt.

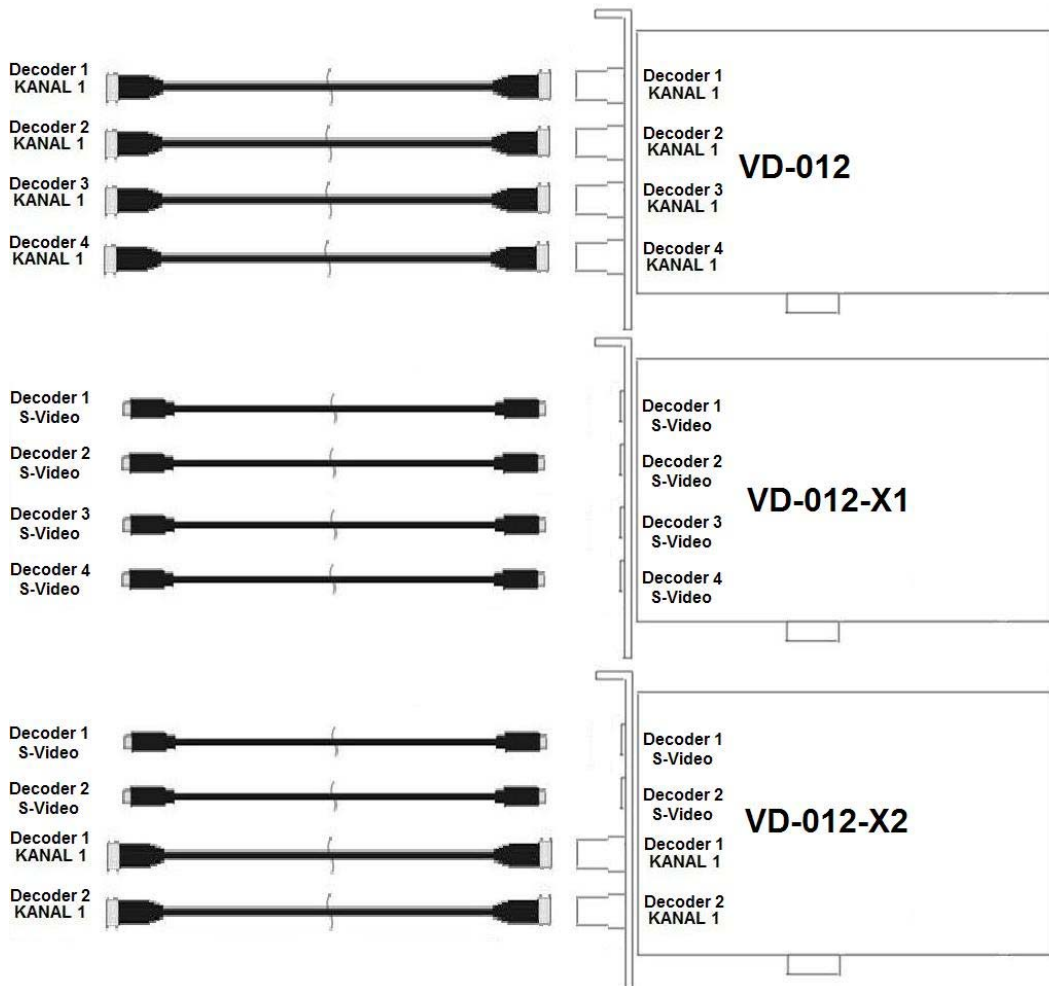


Bild 14: Anschlussmöglichkeiten an pciGrabber-4x4

Im folgenden Kapitel werden die verschiedenen gezeigten Kabelarten kurz erklärt.

4.1.1 Das S-Video Kabel

Das S-Video Kabel wird grabberseitig mit der runden Mini-DIN Buchse verbunden. Die verwendete Videoquelle (z.B. Kamera mit S-Video Ausgang) sollte ebenfalls eine solche Buchse besitzen.

4.1.2 Das Composite Kabel

Die BNC-Stecker können mit den Composite-Ausgängen (BNC-Buchsen) einer Videoquelle verbunden werden.

Hinweis:

Sollte die Compositequelle eine Cinch-Buchse besitzen, muss ein Cinch-auf-BNC-Adapter (75 Ω) verwendet werden.

Um ein Bild einer Kamera darzustellen, ist in der Anwendungssoftware bzw. im Demoprogramm der richtige Decoder und der richtige Kanal zu wählen. Gegebenenfalls kann die mitgelieferte Software automatisch erkennen, welcher Kanal ein Signal führt (siehe *Kapitel 5*).

4.2 Zusatzplatine VZ-012

Zum pciGrabber-4x4 sind Zusatzplatinen erhältlich. Über diese Platinen werden die zusätzlichen Videoeingänge an X800 zur Verfügung gestellt. In Bild 15 ist diese Platine zu sehen.

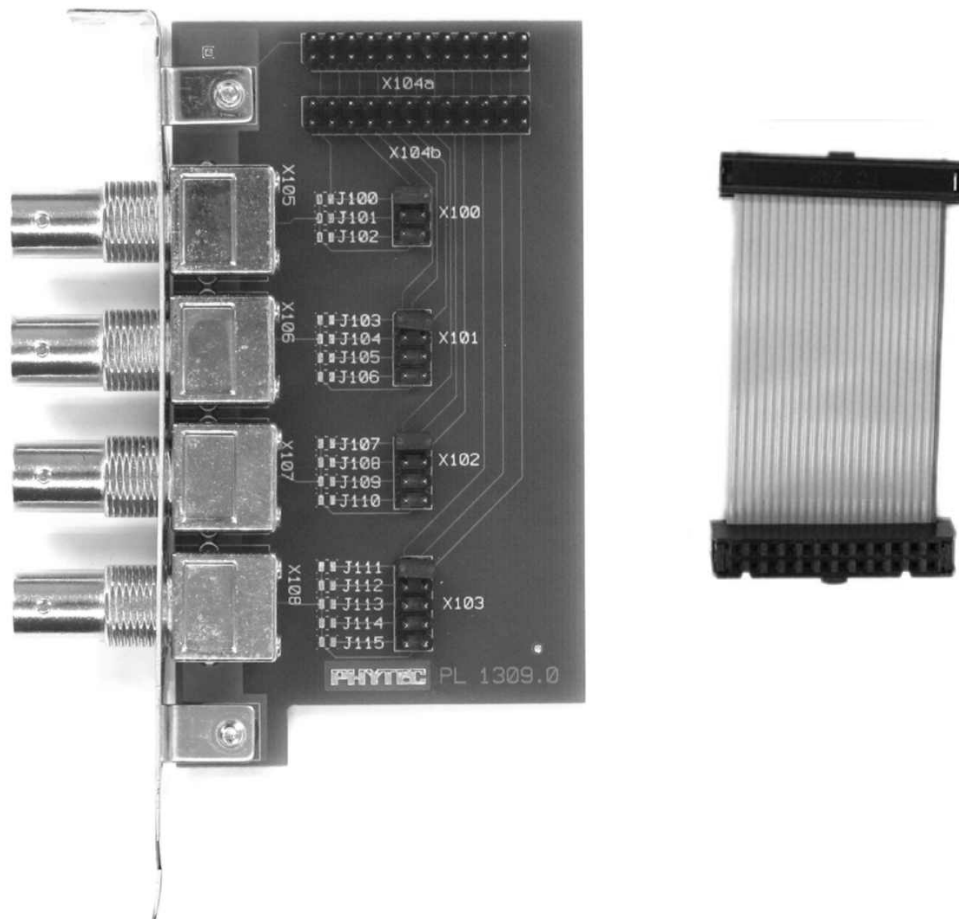


Bild 15: Zusatzplatine VZ-012 mit Flachbandkabel

Abmessungen: 55 x 90 x 20 mm zuzügl. Frontblende

Die Zusatzplatine wird an den Framegrabber angebunden. Das Flachbandkabel wird an die Stiftleiste X800 des Framegrabbers gesteckt. Auf der Zusatzplatine kann zwischen Stiftleiste X104a oder X104b gewählt werden. Es ist möglich, mehrere Zusatzplatinen parallel zu betreiben (Kaskadierung, siehe Bild 16). Dabei muss auf die richtigen Jumperstellungen geachtet werden.

Jede Zusatzplatine stellt vier Composite-Videoeingänge über BNC-Buchsen zur Verfügung. Die Platine selbst kann in einem freien PC-Slot installiert werden.

In Tabelle 6 ist zu sehen, wie viele Zusatzplatten bei welcher Variante von pciGrabber-4x4 möglich sind.

	Zusatzplatine 1	Zusatzplatine 2	Zusatzplatine 3
VD-012	•	•	•
VD-012-X1	•	•	•
VD-012-X2	•		

Tabelle 6: Anzahl der möglichen Zusatzplatten

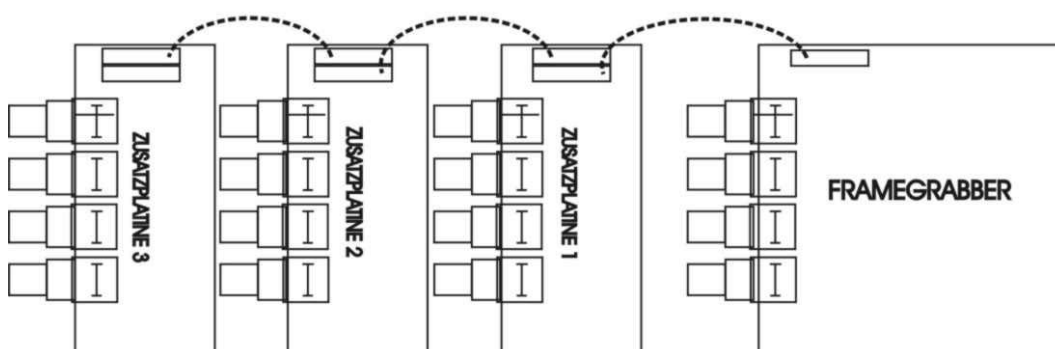


Bild 16: Anschluss der Zusatzplatten

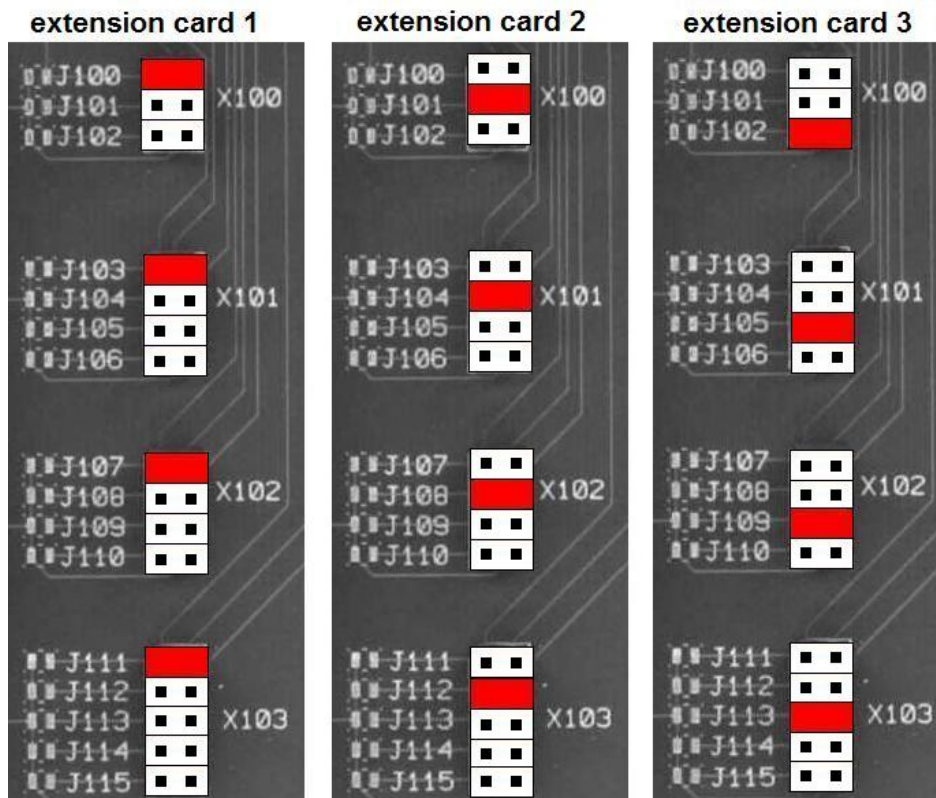
Die Zusatzplatten können wie in Bild 16 an den Grabber angebunden werden. Die erste Zusatzplatine wird mit dem Framegrabber verbunden und mit der nächsten Zusatzplatine verbunden. Die zweite Zusatzplatine wird wiederum mit der dritten verbunden.

Über Jumper muss nun eingestellt werden, welche Videokanäle an welchen Buchsen anliegen sollen.

Achtung!

Stecken Sie die Jumper genau so, wie in Bild 17 und Bild 18 dargestellt ist.

Dies ist wichtig, damit die Zuordnung der Videokanäle mit den Angaben in Kapitel 0 übereinstimmt.



Inbetriebnahme

Bild 17: Jumperstellungen der drei VZ-012 für VD-012 und VD-012-X1

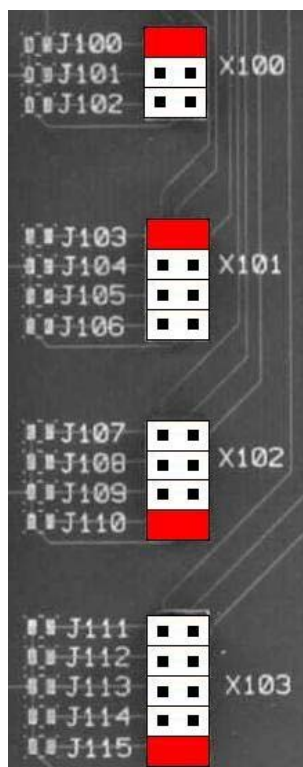


Bild 18: Jumperstellungen auf VZ-012 für VD-012-X2

4.3 Übersicht aller Videoeingangskanäle

In diesem Kapitel wird noch einmal eine komplette Übersicht über alle Eingänge gegeben, inklusive der Zusatzkanäle, die durch die Zusatzplatine VZ-012 zur Verfügung gestellt werden. Das Anschließen und die Jumperstellungen der Zusatzplatinen müssen Kapitel 4.2 entnommen werden, damit die Kanal- und Decoder-Zuordnung passen.

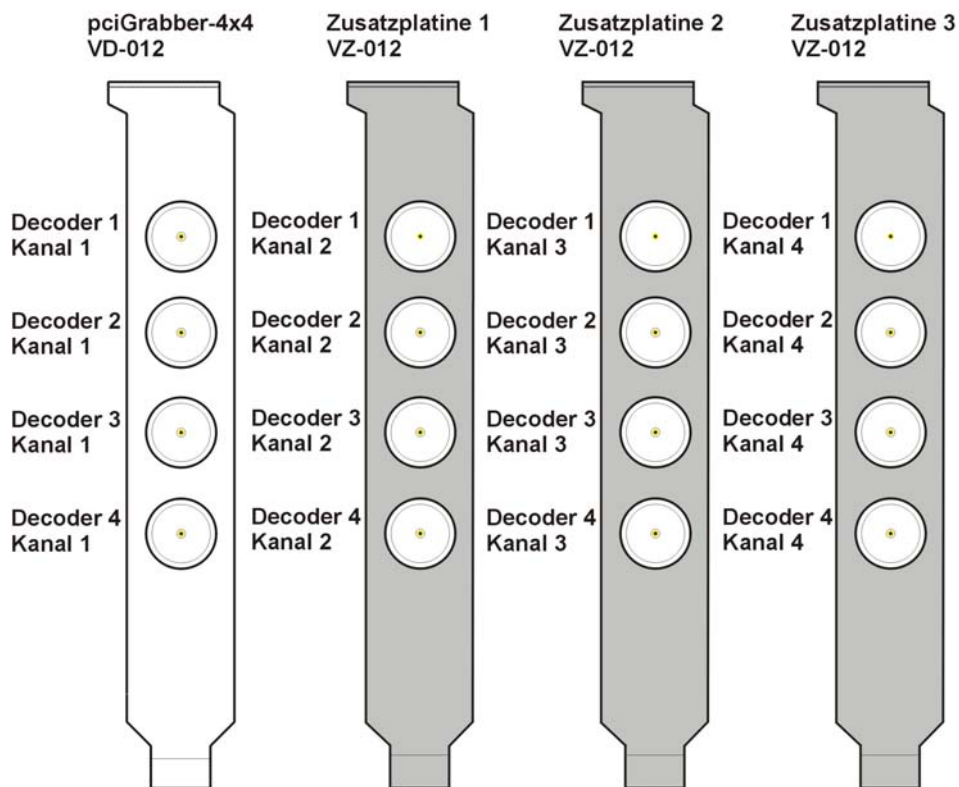


Bild 19: Video-Eingänge VD-012

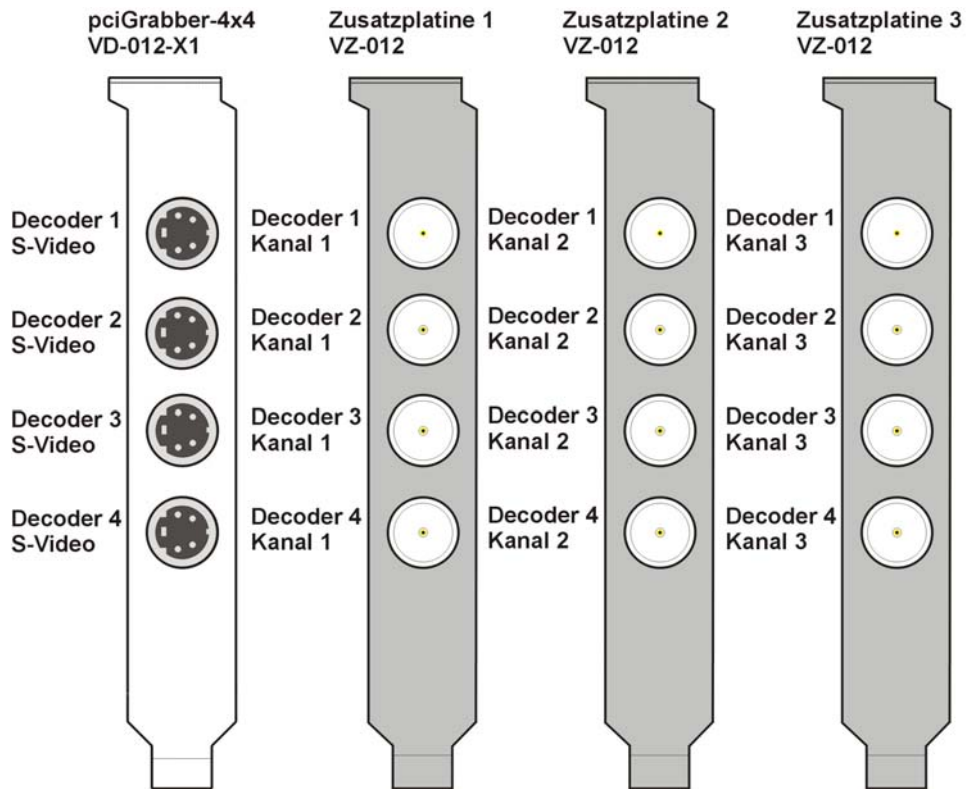


Bild 20: Video-Eingänge VD-012-X1

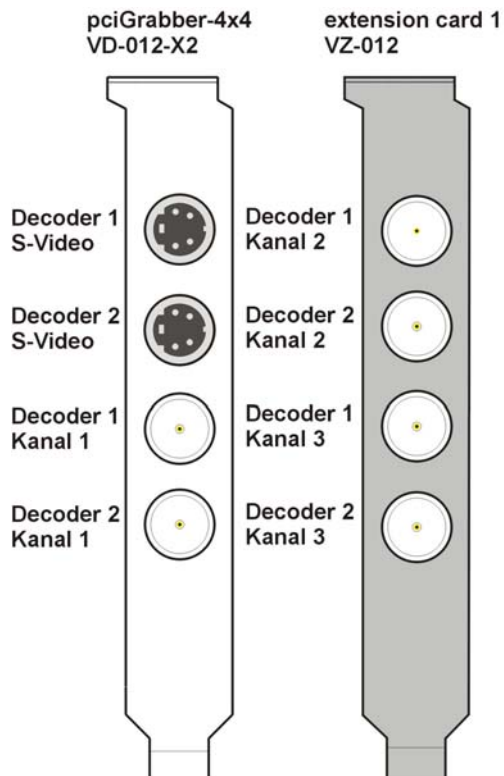


Bild 21: Video-Eingänge VD-012-X2

5 Inbetriebnahme des Grabbers mit Demoprogramm

Das Demoprogramm und der Grabbertreiber müssen korrekt installiert sein (siehe Kapitel 3), um in diesem Kapitel weiterarbeiten zu können.

Unter *START / Programme / Phytec/ pciGrabber4plus / Grab4PCI* finden Sie das Demoprogramm. Nach dem Start erscheint folgendes, zunächst noch leeres Programmfenster mit Menüleiste (siehe Bild 22).

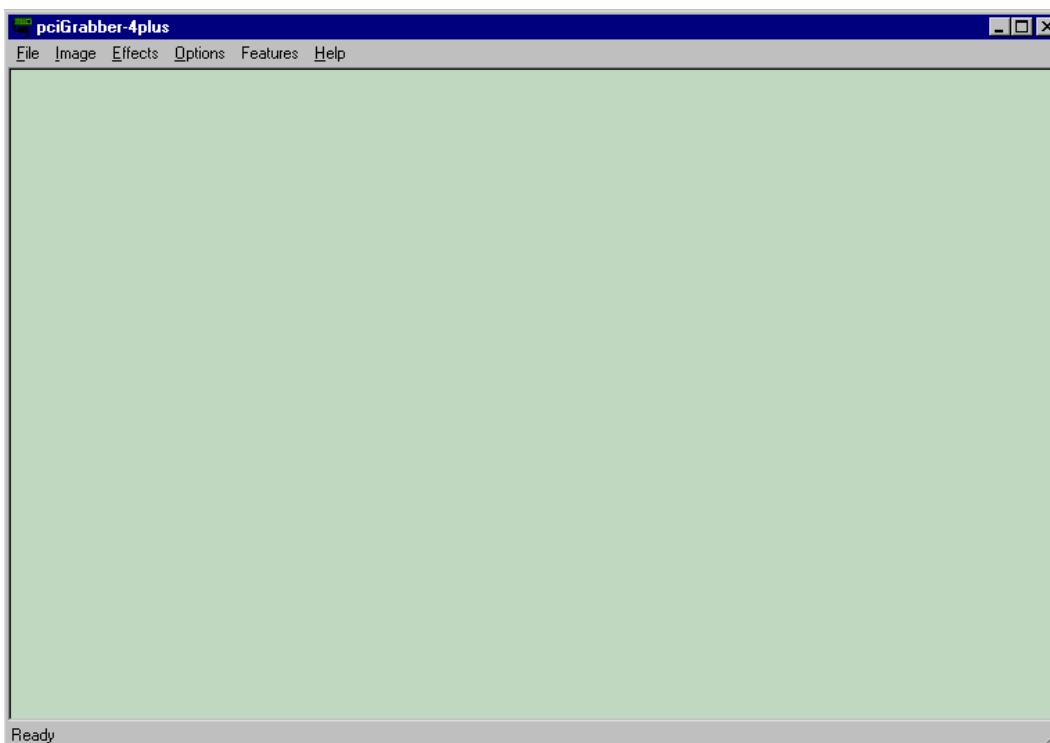


Bild 22: Oberfläche des Demoprogramms

Wir wollen zunächst ein bewegtes Live-Bild von einer Kamera darstellen. Stellen Sie deshalb zunächst sicher, dass eine Videokamera o.ä. an den Grabber angeschlossen ist und ein Bildsignal liefert.

Unter dem Menüpunkt *Options* befinden sich verschiedene Grundeinstellungen bezüglich des Grabbers und den arithmetischen Operationen.

Der Eintrag *Basic Settings* beinhaltet folgendes Menü:

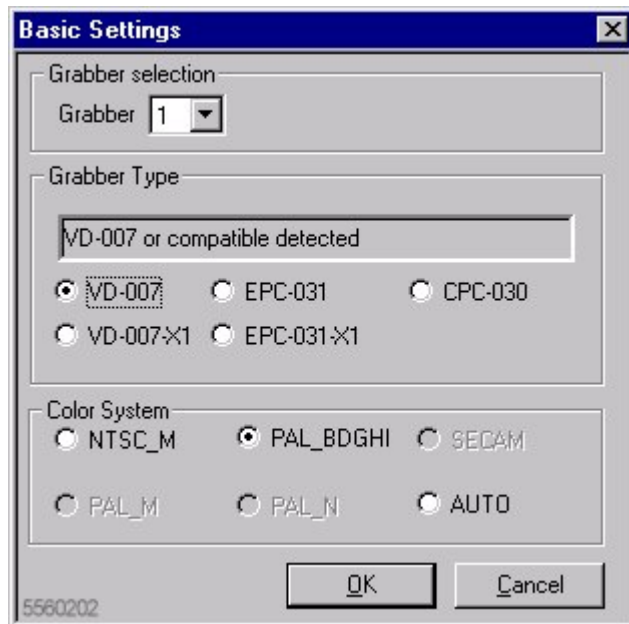


Bild 23: *Basic Settings Menü*

Sie können in diesem Menüpunkt auswählen, welcher Decoder des Grabbers verwendet werden soll. Es können auch mehrere Grabber in einem Rechner verwendet werden. Im dem Feld *Grabber selection* ist der entsprechende Decoder auszuwählen.

Im Bereich *Grabber Type* wird angezeigt, welches Grabbermodell im Rechner installiert ist.

Der pciGrabber-4x4 meldet sich mit:

- VD-012 Decoder 1
- VD-012 Decoder 2
- VD-012 Decoder 3
- VD-012 Decoder 4

oder

VD-012-X1 Decoder 1
VD-012-X1 Decoder 2
VD-012-X1 Decoder 3
VD-012-X1 Decoder 4

oder

VD-012-X2 Decoder 1
VD-012-X2 Decoder 2

Hinweis:

Mit diesem Demo-Programm können Sie mehrere PC-Framegrabbertypen von PHYTEC Messtechnik betreiben.

Zu diesen gehören:

pciGrabber-4plus
pciGrabber-4express

Diese Typen werden wie folgt in der Demosoftware erkannt:

pciGrabber-4plus
VD-009, VD-009-RS6, VD-009-X1 oder VD-009-X1-RS6
pciGrabber-4 express
VD-011 oder VD-011-RS6

Ältere, nicht mehr verfügbare Modelle werden mit der Bezeichnung *VD-007 or compatible detected* erkannt

Unter *Color System* wird das Farbsystem, mit dem der Grabber arbeiten soll, eingestellt.

PAL wird hauptsächlich in Europa, NTSC in USA eingesetzt.

Unter *Grabber Type* wird das erkannte Grabbermodell angezeigt.

Diese Parameter können bei Livebild-Betrieb nicht geändert werden.

Die Einträge *Addition Settings* und *Type Casting Settings* werden später im Zusammenhang mit den Punkten *Add Live Images* und *Arithmetics* im Menüpunkt *Features* erklärt.

In der Menüleiste finden Sie den Eintrag *Image*. Klicken Sie darauf und es öffnet sich folgendes Menü (siehe *Bild 22*).

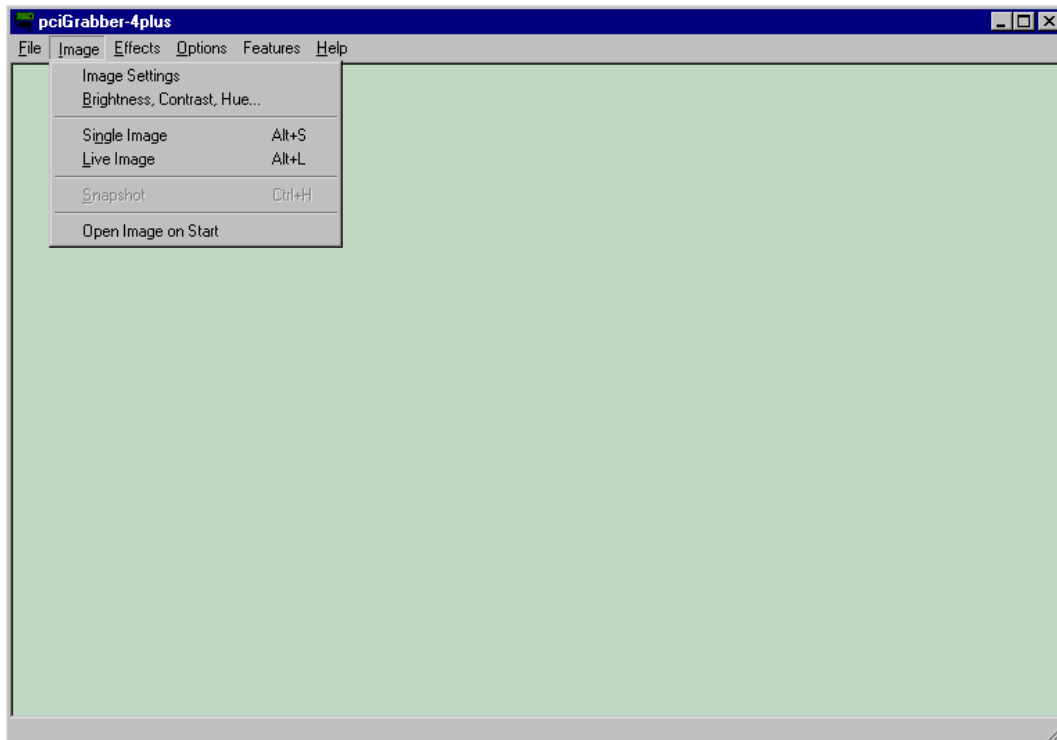


Bild 24: Menüauswahl Image

Wählen Sie den Eintrag *Image Settings*, um die Bildparameter des zu grabbenden Bildes einzustellen (siehe *Bild 24*):

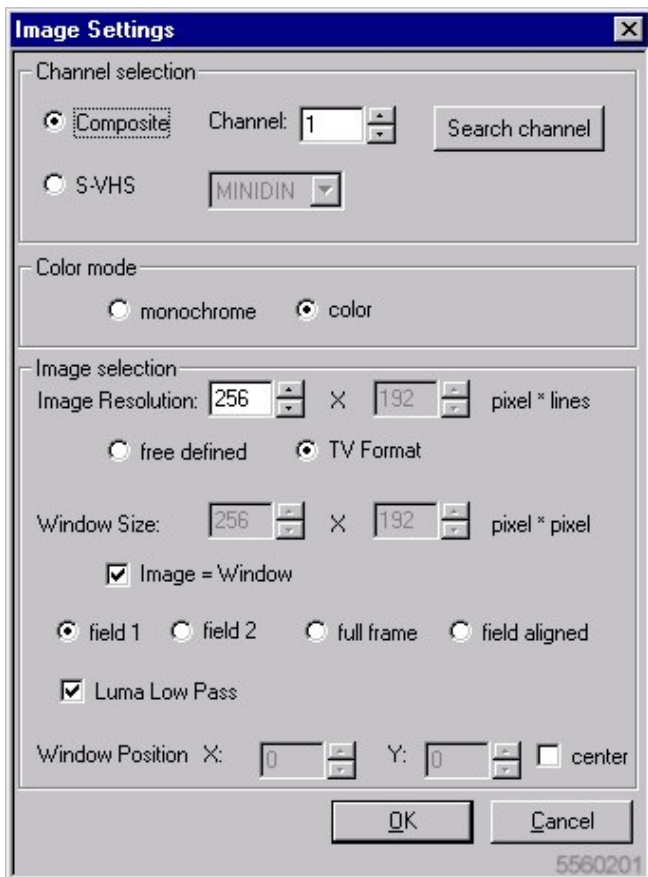


Bild 25: Oberfläche zum Einrichten der Bildausgabe

Die genaue Bedeutung der einzelnen Parameter wird später noch erklärt. Zunächst soll aber das Live-Bild auf den Bildschirm gebracht werden, um den Grabber zu testen. Dazu müssen nur wenige Angaben gemacht werden:

Es ist wichtig, den richtigen Videoeingang des Grabbers zu wählen. Im Bereich *Channel selection* geben Sie den Typ der Videoquelle (Composite / S-VHS) und den verwendeten Eingangskanal an.

Sie können den Eingangskanal manuell wählen oder auch automatisch suchen lassen. Für letzteres klicken Sie auf die Schaltfläche *Search channel* und der erste gefundene Kanal mit einer aktiven Videoquelle wird eingestellt.

Je nachdem, welchen Decoder Sie unter *Basic Settings* ausgewählt haben, müssen Sie die Kamera an den dazugehörigen Videoeingang anschließen, um ein Bild zu erhalten:

Kanalzuordnung VD-012

Grabber selection	Channel selection	Video-Eingang
VD-012 Decoder 1	Composite Channel 1	BNC-Buchse ①
VD-012 Decoder 2	Composite Channel 1	BNC-Buchse ②
VD-012 Decoder 3	Composite Channel 1	BNC-Buchse ③
VD-012 Decoder 4	Composite Channel 1	BNC-Buchse ④

oder

Kanalzuordnung VD-012-X1

Grabber selection	Channel selection	Video-Eingang
VD-012-X1 Decoder 1	S-VHS - MINIDIN	Mini-DIN-Buchse ①
VD-012-X1 Decoder 2	S-VHS - MINIDIN	Mini-DIN-Buchse ②
VD-012-X1 Decoder 3	S-VHS - MINIDIN	Mini-DIN-Buchse ③
VD-012-X1 Decoder 4	S-VHS - MINIDIN	Mini-DIN-Buchse ④

oder

Kanalzuordnung VD-012-X2

Grabber selection	Channel selection	Video-Eingang
	Composite Channel 1	BNC-Buchse ①
VD-012-X2 Decoder 1	S-VHS - MINIDIN	Mini-DIN-Buchse ③
	Composite Channel 1	BNC-Buchse ②
VD-012-X2 Decoder 2	S-VHS - MINIDIN	Mini-DIN-Buchse ④

Tabelle 7: Einstellung des Eingangskanals

Hinweis:

Wenn Sie die Erweiterungskarten VZ-012 verwenden, um mehrere Kameras an einen Decoder anschließen, entnehmen Sie die entsprechende Kanalzuordnung bitte Bild 19 bis Bild 21

Im Bereich *Color Mode* können Sie auch wählen, ob die Bilddarstellung in Farbe oder monochrom erfolgen soll.

Die restlichen Einträge unter *Image selection* können wir zunächst bei den Vorgabewerten belassen.

Durch ein Klick auf *OK* verlassen Sie das Menü.

Wählen Sie erneut den Eintrag *Image* und dann *Live Image*.

Es erscheint nun ein Livebild der gewählten Videoquelle in einem neuen Fenster (siehe *Bild 26*).

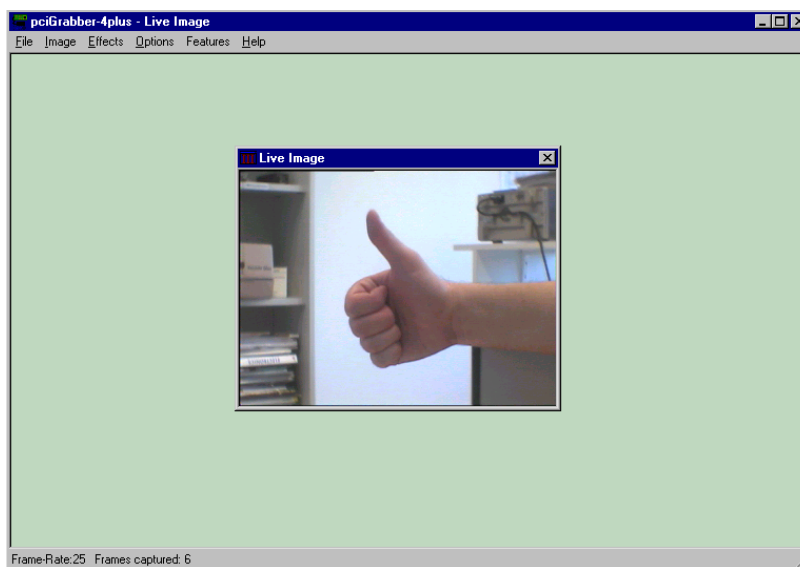


Bild 26: Livebild der Videoquelle

Wenn nur ein blaues Bild erscheint, überprüfen Sie bitte alle Verbindungen. Prüfen Sie bitte, ob die Kamera mit Spannung versorgt wird.

Kontrollieren Sie, ob der richtige Kanal und der richtige Decoder eingestellt wurden

Überprüfen Sie gegebenenfalls die Funktion der Signalquelle mit einem anderen Gerät (z.B. Video-Monitor).

Weitere mögliche Fehlerquellen sind im Anhang beschrieben.

In der unteren Leiste des Hauptfensters sehen Sie die Anzeige *Frame-Rate xx* (xx =Zahl). Die Zahl gibt die Bilder pro Sekunde an,

die im Live-Fenster aufgebaut werden. Der Wert ist abhängig von der Bildgröße und der Leistungsfähigkeit des Rechners, da das digitalisierte Bild vom RAM in die Grafikkarte übertragen werden muss, damit es schließlich auf dem Bildschirm erscheint.

Hinweis:

Die Bilddaten werden vom Grabber (unabhängig von der Rechenleistung des Prozessors) immer in Echtzeit im Arbeitsspeicher (RAM) des PCs abgelegt.

Die weitere Verarbeitung ist abhängig von der Rechenleistung des Prozessors (CPU) des PCs.

Weiterhin wird in der Statuszeile ein Zähler, der die Gesamtzahl der erfassten Livebilder (*Frames Captured*) anzeigt, dargestellt.

Wenn dieser Zähler bei 255 angelangt ist, fängt er wieder bei 0 an.

Dies kann z.B. als Indikator verwendet werden, ob der Grabber überhaupt aktiv ist.

5.1 Paralleler Bildeinzug mehrerer Bildquellen

Im letzten Kapitel haben wir einen Kanal des pciGrabber-4x4 in Betrieb genommen.

In diesem Abschnitt werden wir mehrere Bildquellen gleichzeitig digitalisieren und auf dem Bildschirm darstellen.

Wichtig zu wissen:

Die einzelnen Decoder des pciGrabber-4x4 verhalten sich wie völlig unabhängige Framegrabber.

Jede dieser Framegrabber-Instanzen können Sie individuell konfigurieren und steuern.

Das Demoprogramm zeigt jeweils die Einstellungen und das Bild *einer* dieser Framegrabber-Instanzen.

Um mehrere Bilder gleichzeitig darzustellen, öffnen Sie das Demoprogramm mehrmals. Somit kann bei den Varianten VD-012 und VD-012-X1 das Demoprogramm viermal gestartet werden.

Bei der Version VD-012-X2 können Sie das Programm zweimal parallel öffnen.

Als erstes muss nun unter *Basic Settings* für jedes Programm ein eigener Decoder zugewiesen werden.

Wichtig!

Achten Sie vor Start einer Bildaufnahme darauf, dass die Decoder-Zuordnung in den einzelnen Demo-Programmen unterschiedlich sind! Bekommen zwei Programme den gleichen Decoder zugewiesen, kann dies zu Problemen führen.

Wählen Sie nun in jedem Programm den Kanal unter *Image – Image Settings* aus und starten Sie dann die Bildaufnahme mit *Live Image*.

Hinweis:

Wenn die Bilddarstellung flackert oder verzerrt ist oder Sie ein schwarzes oder blaues Bild sehen, könnte eine fehlerhafte Einstellung der Decoder-Auswahl die Ursache sein.

Wenn mehrere Programm-Instanzen auf den gleichen Decoder zugreifen, können die Einstellungen dieses Decoders wechselseitig überschrieben werden und somit nicht mehr zu der vom jeweiligen Programm erwarteten erwarteten Einstellung passen.

Schliessen Sie in diesem Fall alle Demo-Programme bis auf eines.

Stoppen Sie die Bilddarstellung und rufen Sie *Image – Image Settings* erneut auf, um den Decoder neu zu konfigurieren.

Starten Sie dann ein weiteres Demo-Programmfenster erneut und stellen Sie unter *Basic Settings* die Decoder-Zuordnung richtig ein (keine Kollisionen mit anderen Programm-Instanzen).

5.2 Buslast bei Verwendung mehrerer Decoder

Die gleichzeitige Digitalisierung von mehreren Videoströmen erzeugt eine hohe Datenmenge auf dem internen Bus der Framegrabberkarte. Die Datenmenge ist dabei abhängig von den gewählten Bildeinstellungen und der Anzahl von aktiven Decodern.

Übersteigt die Summe der Datenmenge die Kapazität der Bussysteme, so können Bildstörungen auftreten.

In diesem Abschnitt wird beschrieben, worauf Sie beim gleichzeitigen Betrieb mehrerer Decoder achten müssen, damit dies vermieden wird.

Der pciGrabber-4x4 verwendet eine PCI Express-to-PCI-Bridge. Das bedeutet, dass alle Videodecoder zusammen an einem internen PCI-Bus betrieben werden.

Während der PCI Express-Bus bis zu 250 MB/s übertragen kann, ist auf dem PCI-Bus eine maximale Datenrate von 132 MB/s möglich.

Bei einer Digitalisierung von einem Decoder kann das Datenvolumen 44,24 MB/s betragen.

Es muss nun beachtet werden, dass die Summe der Datenrate aller Decoder die maximal möglichen Datenraten nicht übersteigt.

Die Datenrate eines Decoders hängt von dessen Konfiguration ab.

Maßgeblich sind folgende Parameter:

- Bildgröße (Anzahl der Pixel)
 - Anzahl der Byte pro Pixel (abhängig vom gewählten Farbformat, siehe *Bild 52*)
 - Anzahl der Bilder pro Sekunde (Framerate)
- Berechnung der maximalen Datenmenge eines Decoders (PAL):

Bildgröße(max): 768 x 576

Anzahl der Bytes pro Pixel (RGB32): 4

Anzahl der Bilder pro Sekunde: 25

$$\text{Bildgröße} \cdot \frac{\text{Byte}}{\text{Pixel}} \cdot \frac{\text{Bilder}}{s} = \frac{\text{MByte}}{s} \quad \text{Gleichung 1}$$

$$768 \cdot 576 \cdot 4\text{Byte} \cdot 25 \frac{1}{s} = 44,2368 \frac{\text{MByte}}{s}$$

Werden nun vier Decoder gleichzeitig verwendet, steigt das Datenvolumen um den Faktor vier:

$$44,2368 \frac{\text{MByte}}{s} \cdot 4 = 176,9 \frac{\text{MByte}}{s}$$

Um nun eine saubere Übertragung gewährleisten zu können, darf die maximale Datenrate des PCI-Busses nicht überschritten werden. Zusätzlich sollte noch ein Sicherheitsabstand von ~15 % berücksichtigt werden. Dies bedeutet, dass eine maximale Datenrate von 112 MB/s nicht überschritten werden sollte.

In *Tabelle 8* und *Tabelle 9* sind die Datenraten für einige Beispiel-Konfigurationen aufgeführt:

Anzahl der Decoder	RGB32 768 x 576	RGB32 640 x 480	RGB32 620 x 450	RGB16 768 x 576	Y8 Gray 768 x 576
1	44,2 MB/s	30,7 MB/s	27,9 MB/s	22,1 MB/s	11,1 MB/s
2	88,5 MB/s	61,4 MB/s	55,8 MB/s	44,2 MB/s	22,1 MB/s
3	132,7 MB/s	92,2 MB/s	83,7 MB/s	66,4 MB/s	33,2 MB/s
4	176,9 MB/s	122,9 MB/s	111,6 MB/s	88,5 MB/s	44,2 MB/s

Tabelle 8 Datenraten PAL 25fps

Anzahl der Decoder	RGB32 720 x 480	RGB32 640 x 480	RGB32 580 x 400	RGB16 720 x 480	Y8 Gray 720 x 480
1	41,5 MB/s	36,9 MB/s	27,8 MB/s	20,7 MB/s	10,4 MB/s
2	82,9 MB/s	73,7 MB/s	55,7 MB/s	41,5 MB/s	20,7 MB/s
3	124,4 MB/s	110,6 MB/s	83,5 MB/s	62,2 MB/s	31,1 MB/s
4	165,9 MB/s	147,5 MB/s	111,4 MB/s	82,9 MB/s	41,5 MB/s

Tabelle 9 Datenraten NTSC 30fps

Werden die 112 MB/s nicht überschritten, ist eine problemlose Datenübertragung möglich. Wie in den Tabellen zu sehen ist, existieren verschiedene Möglichkeiten das Volumen der Daten zu verringern:

Verringern der Auflösung

1. *Verringern der Farbtiefe*
2. *Verringern der Bildwiederholungsrate*

Mit der Hilfe von Gleichung 1 kann man das Datenvolumen berechnen und entscheiden, welche Einstellungen für die entsprechende Applikation passend sind.

Dabei ist mit Gleichung 1 die Datenrate für jeden Decoder entsprechend seiner Einstellungen zu berechnen und alle Datenraten zu addieren.

Die Summe darf 112 MB/s nicht überschreiten.



Bild 27: Überlasteter PCI-Bus

Bild 27 zeigt einen Effekt, welcher bei einer zu hohen Datenrate auftreten kann. Es ist gut zu sehen, dass das gesamte Bild streifig wird. Die Streifen entstehen dadurch, dass manche Bilddaten, mangels Kapazität des PCI-Busses, nicht schnell genug in den Arbeitsspeicher geschrieben werden können. Die Daten, die nicht übertragen werden können, gehen verloren, da der Videodecoder diese nicht zwischenspeichern kann. An diesen Stellen wird der Bildspeicher nicht aktualisiert. Hier sind dann die Pixel des letzten übertragenen Bilds zu sehen. In diesen Bereichen werden also „alte Bilddaten“ wiedergegeben und das Bild wird streifig.

5.3 Beschreibung des Demoprogramms

In diesem Abschnitt sehen wir uns nun die Funktionen bzw. Menüs des mitgelieferten Demoprogramms genauer an.

Sie können mit dem Demoprogramm sehr viele Parameter der Decoder einstellen und ausprobieren.

Das Menü *Image Settings* (siehe *Bild 28*) beinhaltet Parameter, welche die Bilderzeugung und Bilddarstellung beeinflussen:

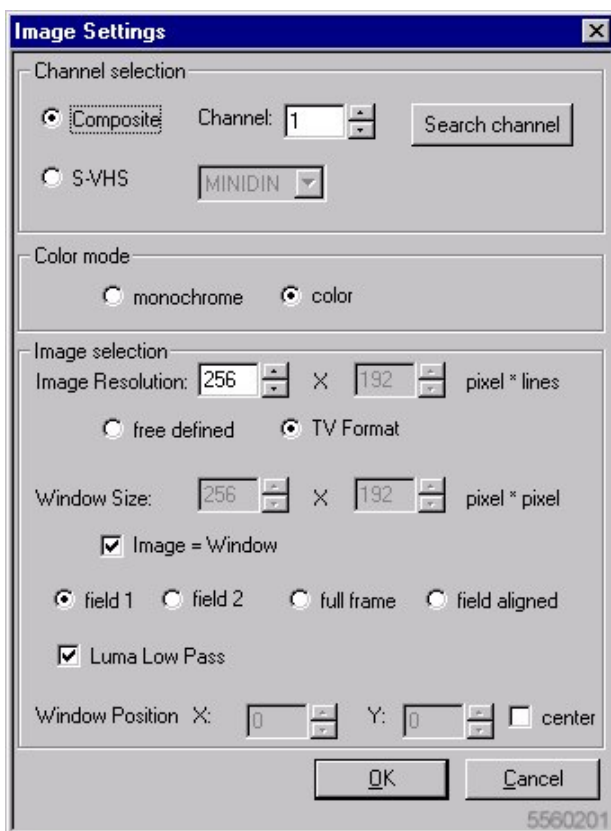


Bild 28: Menü „Image Settings“

Sie können hier Einstellungen vornehmen, bevor Sie ein Livebild sehen. Während des Livebildbetriebs können diese Einstellungen nicht geändert werden.

Der Abschnitt *Channel selection* ermöglicht die Einstellung des Videoquellentyps und die Kanalwahl.

Wählen Sie zunächst die richtige Signalart mit den Buttons *Composite* und *S-VHS*.

- **Composite-Quellen**

Composite bezieht sich auf die BNC-Stecker.

Wählen Sie unter dem Eintrag *Channel* den richtigen Eingangskanal für die angeschlossene Kamera. Beachten Sie auch, dass der richtige Decoder ausgewählt ist (Menü *Basic Settings*).

Sie können die Kamera auch suchen lassen, indem Sie auf *Search Channel* klicken. Das Programm stellt dann den ersten Kanal ein, auf dem ein Videosignal gefunden wurde. Diese Funktion kann immer nur auf einem Decoder suchen. Um die gesamte Karte prüfen zu lassen, muss unter *Basic Settings* jeder einzelne Decoder ausgewählt und mit *Search* durchsucht werden.

Bei Verwendung der Zusatzplatine VZ-012 stehen bis zu 16 Videokanäle zur Verfügung (VD-012).

- **S-Video - Quellen**

S-Video (oder „*S-VHS*“) – Quellen werden mit *MINIDIN* – bezeichnet. Die Bildquelle ist an der zugehörigen Mini-DIN-Buchse angeschlossen

Mit den Buttons *color* und *monochrom* kann das Bild in Farbe (bei Verwendung einer Farb-Videoquelle) oder monochrom wiedergegeben werden.

Im unteren Fensterbereich „*Image selection*“ kann Größe und Auflösung des gegrabten und dargestellten Bilds eingestellt werden.

Mit der Einstellung *Image Resolution* kann die Bildauflösung (= „Qualität“) eingestellt werden.

Die Einstellung teilt sich in x-Richtung für die Pixelzahl und in y-Richtung für die Zeilenzahl. Beide Werte können getrennt verändert werden, wenn der Button *free defined* selektiert wurde.

Dabei ist jedoch zu beachten, dass das Bild verzerrt (gedehnt oder gestaucht) wiedergegeben wird, wenn das Seitenverhältnis nicht 4:3 beträgt. (Dieses Verhältnis von Breite zu Höhe ist durch die Fernsehnorm vorgegeben).

Mit dem Button *TV Format* verhindern Sie, dass das Bild verzerrt dargestellt wird, da dann bei eingegebener Pixelzahl die Zeilenzahl im Verhältnis 4:3 (Breiten/Höhenverhältnis) automatisch berechnet wird.

Mit dem Button *Window Size* kann ein Ausschnitt aus dem gesamten digitalisierten Bildfeld herausgeschnitten und anstelle des gesamten Bildes auf dem Bildschirm angezeigt werden.

Dieser kann entsprechend kleiner als z.B. das Sichtfeld einer Kamera sein. Soll der gesamte digitalisierte Bereich dargestellt werden, setzen Sie das Häkchen bei *Image=Window*.

Die Einstellung *Window Size* verzerrt die Bildgeometrie nicht, da es kein Skalieren, sondern ein Ausschneiden ist.

Hinweis:

Beachten Sie, dass sowohl Skalierung als auch Ausschnittsbildung im Grabber in Echtzeit durchgeführt werden. Das dargestellte Bild wird also genau so vom Grabber im Speicher abgelegt, wie es auf dem Bildschirm erscheint. Das ist ein großer Vorteil, da keine CPU-Leistung für diese Aufgabe benutzt werden muss.

Ein kurzer Exkurs in die analoge Fernsehtechnik, nach der die Kamera arbeitet, trägt zum besseren Verständnis der Buttons *field1*, *field2*, *full frame* und *field aligned* bei:

Ein Fernsehbild (Norm-Videosignal), besteht aus zwei ineinander verschachtelten Teilbildern, den sog. Halbbildern (siehe *Bild 29*). Diese Halbbilder werden analog hintereinander erzeugt und nachfolgend im darstellenden Gerät (z.B. Fernseher) verschachtelt.

Die Verschachtelung dient der Verringerung des Flimmereffekts eines TV-Bildes.

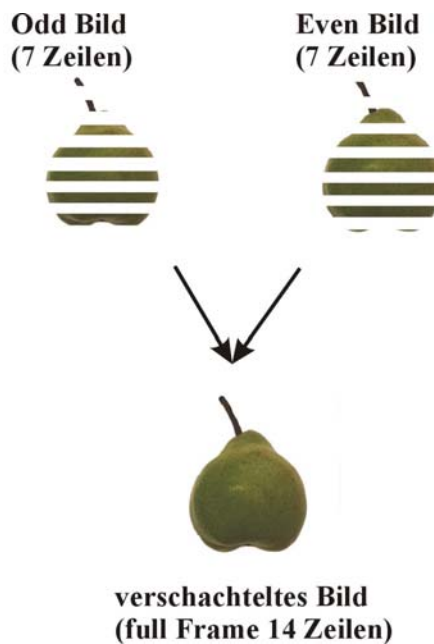


Bild 29: Entstehung des Vollbildes aus zwei Halbbildern mit je 7 Zeilen

Das Signal besteht bei der PAL-Norm aus 625 Zeilen, die in ein erstes Halbbild (*ungerade/odd, field1 mit Zeilen 1-625*) und ein zweites Halbbild (*gerade/even, field2 mit Zeilen 2-624*) aufgeteilt sind. Man kann die Bildfläche schon mit einem Halbbild vollständig erkennen, die vertikale Auflösung ist jedoch um die Hälfte reduziert, da dann nur noch 288 Zeilen vorhanden sind (abzüglich nicht sichtbarer Vor- und Nachlaufzeilen, sowie Prüf- und Datenzeilen bleiben ca. 576 sichtbare Zeilen von insgesamt 625 Zeilen).

Der Vorteil, wenn nur ein Halbbild digitalisiert wird, ist jedoch die Schnelligkeit von 20 ms im Gegensatz zu 40 ms bei einem Vollbild (*full frame*), also beiden Halbbildern.

Wenn man immer das gleiche (z.B. das erste) Halbbild digitalisieren möchte, entsteht allerdings immer eine Pause von 20 ms zwischen den Digitalisierungsvorgängen.

Ein weiterer Nachteil der Vollbilddigitalisierung ist, dass Objekte, die sich schnell bewegen, einen Versatz im Bild erzeugen können. Das Objekt ist im ersten Halbbild an einer anderen Position als im zweiten Halbbild, so dass eine Art Kammefekt entsteht und das Bild dann ähnlich wie *Bild 30* dargestellt aussieht.

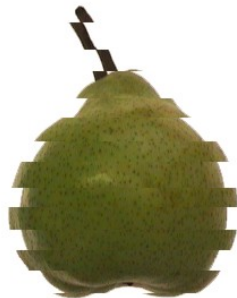


Bild 30: Kammefekt bei schnell bewegten Objekten

Sie haben im Demoprogramm die Möglichkeit, die oben erklärten Parameter zu beeinflussen.

Bei vertikalen Auflösungen kleiner 288 Zeilen ist es demnach günstiger, ein Halbbild zu digitalisieren. Dies kann mit den Buttons *field1* (erstes, ungerades Halbbild) oder *field2* (zweites, gerade Halbbild) ausgewählt werden.

Ist die Zeilenzahl größer 288, müssen beide Halbbilder digitalisiert werden, dazu klicken Sie *full frame* an. Wenn Sie bei *Image Resolution* eine Zeilenzahl größer 288 einstellen, wird diese Option automatisch gesetzt.

Der Button *field aligned* ermöglicht bei Halbbild-Darstellung die doppelte Anzahl an Halbbildern pro Sekunde. Dabei entfällt die 20 ms-Pause zwischen der Halbbilddigitalisierung.

Würde man einfach beide Halbbilder nacheinander darstellen, so würde der dargestellte Bildinhalt optisch immer eine halbe Zeile nach oben und unten springen.

Dies geschieht, da die räumliche Verschachtelung der Zeilen zu einem Vollbild nicht durchgeführt wird. Bei der Einstellung *field aligned* verschiebt der Grabber das zweite Halbbild automatisch um eine halbe Zeile, so dass es genau auf das erste Halbbild zu liegen kommt.

Dadurch tritt dieser Sprungeffekt nicht auf. Diese Einstellung *field aligned* ist also dann nützlich, wenn man Bilder mit höchstens 288 Zeilen so schnell wie möglich nacheinander digitalisieren will (1 Halbbild in jeweils 20 ms).

Bei horizontalen Auflösungen kleiner 360 Pixel sollten Sie das Häkchen bei *Luma Low Pass* setzen. Dieser Tiefpaß bewirkt quasi eine „Glättung“ des in der Größe reduzierten Videobildes. Es wird automatisch aktiviert bei Auflösungen kleiner 360 Pixel und deaktiviert bei größeren Auflösungen.

Mit *Window Position* können Sie innerhalb des oben gewählten Bildausschnitts die Position des darzustellenden Bildfensters vorgeben. Die Werte repräsentieren die Position der linken oberen Ecke. Soll der Ausschnitt genau in der Mitte des TV-Bildes liegen, kann der Haken bei *center* gesetzt werden.

Die Einstellung bewirkt also eine Verschiebung des Bildausschnitts im Gesamtbild. Sie kann natürlich nur dann vorgenommen werden, wenn der Bildausschnitt *Window Size* kleiner ist als das Gesamtbild *Image Resolution*.

5.4 Bildkontrolle

Während der Darstellung des Live-Bilds können Sie das Bildkontrollfenster öffnen, indem Sie unter *Image* auf *Brightness*, *Contrast*, *Hue...* klicken. Es öffnet sich der dargestellte Dialog.

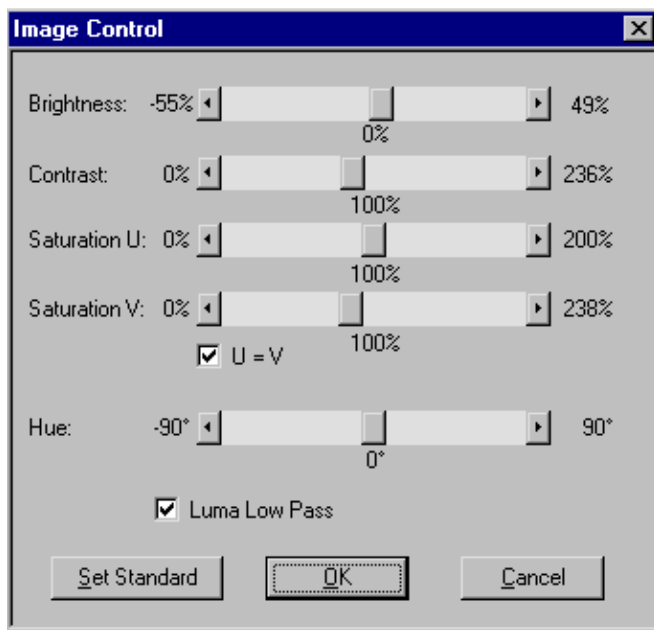


Bild 31: Fenster zur Bildkontrolle

Über Schieberegler lassen sich die Einstellungen von Helligkeit, Kontrast, Farbsättigung und Farbton verändern.

Die Werte werden sofort in den Grabber übertragen, so dass sich die Auswirkung der Justierung sofort im Live-Bild beurteilen lässt.

Über den Regler *Brightness* wird die Helligkeit eingestellt und kann an die Umgebungsbedingungen angepasst werden.

Der Regler *Contrast* verändert den Kontrast im Bild.

Für die Einstellung der Farbsättigung sind zwei Regler vorhanden: *Saturation U* und *Saturation V*. Dies ermöglicht eine getrennte Manipulation der Sättigung im Rot- und im Blauviolett-Bereich. Mit der Kontrollbox *U=V* lassen sich beide Regler zusammenschalten. Auf diese Weise kann man Farbsättigungsänderungen vornehmen, ohne den Farbton zu verändern.

Der Einstellregler für den Farbton, *Hue*, ist nur bei Verwendung des NTSC-Farbsystems sinnvoll. Er dient zur Korrektur von Farbtonverfälschungen durch Phasenfehler auf dem Übertragungsweg. Solche Störungen treten nur beim NTSC-System auf.

Das PAL-System korrigiert Farbtonfehler automatisch, so dass die Veränderung des Farbtonreglers keine Auswirkungen hat. Belassen Sie hier bitte den Regler in der Mittelposition.

Hinweis:

Korrekturen des Farbtons (z.B. Weißabgleich) können Sie bei PAL und NTSC dadurch durchführen, dass Sie die Farbsättigungsregler einzeln verschieben.

(Grundsätzlich ist es aber besser, den Weißabgleich nach Möglichkeit schon an der Kamera vorzunehmen.)

5.5 Weiter Funktionen unter *Image*

- Mit dem Eintrag *Single Image* wird ein Schnappschuss aufgenommen und auf dem Bildschirm dargestellt. In diesem Modus führt der Grabber nur eine Digitalisierung durch.
- Die Bilddarstellung wird durch die Parameter in *Image settings* bestimmt.
- Unter *Live Image* kann – wie zuvor beschrieben - das Livebild auf den Bildschirm gebracht werden. Auch hier bestimmen die Parameter in *Image settings* die Bilddarstellung.
- Mit der Option *Snapshot* können Schnappschüsse während des Livebetriebs erstellt werden.
- Das in diesem Moment gezeigt Bild wird dann in einem eigenen Fenster dargestellt. Es können mehrere Momentaufnahmen gemacht werden.
- Die erscheinenden Fenster werden automatisch durchnummeriert.
- Der Eintrag *Open Image on Start* ermöglicht es, mit den vorher unter *Image Settings* eingestellten Parametern, bei jedem Programmstart automatisch das Livebild der Videoquelle auf dem Bildschirm darzustellen.
- Wenn zusätzlich das Demoprogramm in der Autostart-Gruppe abgelegt wird, kann der Rechner nach dem Hochfahren ohne Eingreifen des Benutzers ein Livebild mit den vorher eingestellten Parametern auf den Bildschirm bringen.

5.6 Fadenkreuze einblenden

Wenn Sie Objekte in der Mitte des Bildes zentrieren wollen, können Sie verschiedene Fadenkreuze in das Livebild ein- und ausblenden. Die Einstellungen dafür befinden sich in der Menuleiste unter *Effects*. Es können auch Kombinationen der Kreuze bzw. alle auf einmal eingeblendet werden.

5.7 Die Sonderfunktionen

Das Demoprogramm bietet einige Sonderfunktionen, mit denen der Bildinhalt manipuliert und ausgewertet werden kann.

- **Histogramm anzeigen:**

Der Menüpunkt *Histogramm* ermöglicht es, von einem Standbild, das z.B. über die *Snapshot* Funktion erstellt wurde, eine Histogramm zu errechnen.

Ein Histogramm stellt die Farbwert- bzw. Grauwertverteilung dar. Die relative Häufigkeit der einzelnen Intensitätswerte wird über der Helligkeit bzw. der Intensität aufgetragen (siehe *Bild 32*).

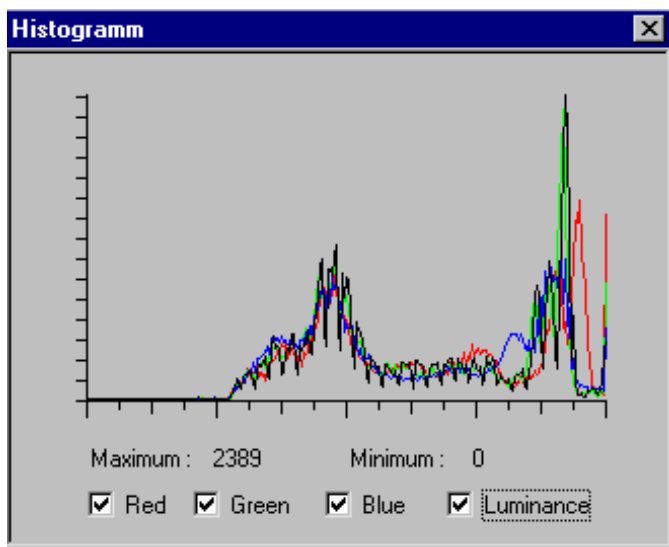


Bild 32: *Histogramm*

Der Wertebereich auf der x-Achse erstreckt sich von 0 bis 255. Die einzelnen darzustellenden Farben können über Checkboxes ausgewählt werden.

Achtung!

Das Histogramm kann nur von einem Standbild, nicht von einem Livebild dargestellt werden. Erzeugen Sie ein Standbild aus einem Livebild, indem Sie die Snapshot-Funktion verwenden.

- **Farben auswerten:**

Der Eintrag *Color Meter* öffnet die Oberfläche in *Bild 33*.

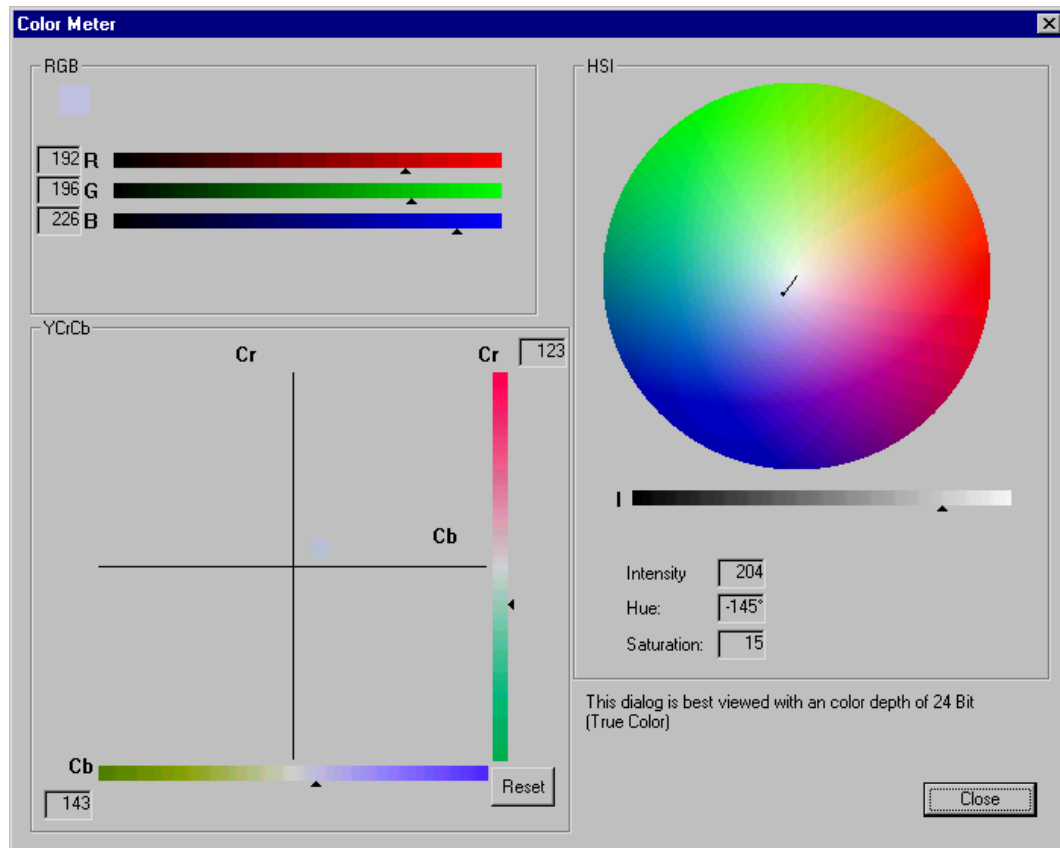


Bild 33: *Color Meter*

Das Color Meter funktioniert nur in der Livebilddarstellung.

In ihm wird der Farbwert des Pixels, das sich in der Bildmitte befindet, in verschiedenen Farbmodellen angezeigt.

Ein kleines im Livebild erscheinendes Fadenkreuz zeigt die Bildmitte an.

Das RGB-Modell zeigt die Farbwerte für Rot, Grün und Blau als Zeiger auf Intensitätsbalken sowie als Zahlenwert an.

Das YCrCb-Modell stellt den Farbwert als Farbbalken sowie zusätzlich in einem Koordinatensystem dar.

Der Farbverlauf kann dadurch über einen längeren Zeitraum beobachtet werden.

Mit dem Button *Reset* wird die Koordinatenanzeige gelöscht und es wird erneut aufgezeichnet.

Das HSI-Modell stellt den Farbwert auf einem Kreis dar. Der Vektor in dem Kreis zeigt durch seine Länge die Sättigung (Saturation), und durch seine Richtung den Farbton (Hue). Die Helligkeit wird auf einem Balken darunter angezeigt.

Die einzelnen Werte werden auch numerisch ausgegeben.

- **Farbbalken einblenden:**

Mit der Option *Color Bars* kann der Grabber getestet werden.

Die Farbbalken werden von der Hardware, nicht vom Demoprogramm generiert. Je nach gewähltem Bildausschnitt werden mehr oder weniger Balken zu sehen sein.

Alle Farbbalken werden bei einer horizontalen Auflösung von ca. 515 Pixeln sichtbar.

- **Arithmetische Operationen an Standbildern:**

Der Menüpunkt *Arithmetics* ermöglicht einfache arithmetische Operationen auf Standbildern (siehe *Bild 23*).

Bilder können pixelweise addiert, subtrahiert, multipliziert oder dividiert werden. Zusätzlich können Konstanten addiert oder multipliziert werden.

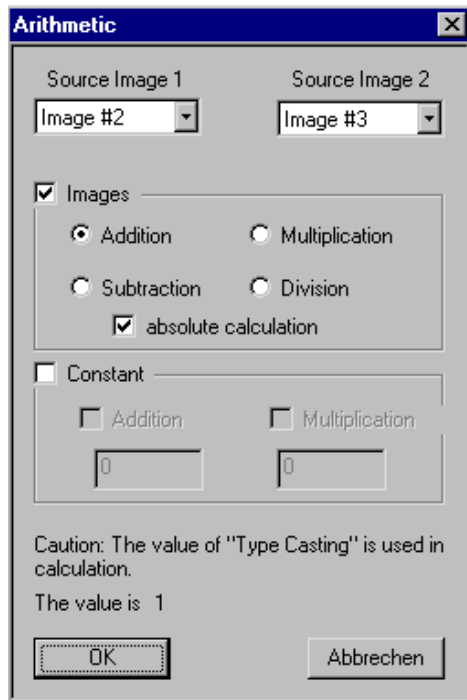


Bild 34: Arithmetics Menü

Die zu verarbeitenden Bilder werden unter *Source Image 1* und *2* ausgewählt. Die Nummern hinter *Image#* korrespondieren mit den Nummern der Bildfenster.

Unter dem Eintrag *Images* kann die arithmetische Operation gewählt werden. Ebenfalls kann angegeben werden, ob betragsmäßig gerechnet werden soll (*absolut calculation*).

Bei betragsmäßiger Berechnung können keine negativen Werte entstehen, die eventuell kein sinnvolles oder ein falsches Ergebnis darstellen.

Unter dem Punkt *Constant* kann eine Konstante zu jedem Pixel addiert (Helligkeitsänderung) oder multipliziert (Kontraständerung) werden.

Alle Operationen werden einer Normierung unterzogen.

Dies ist wichtig, da der Wertebereich des Ergebnisses weit außerhalb des Darstellungsbereiches von 0 bis 255 pro Farbkanal liegen kann.

Ergebnisse größer 255 werden auf 255, und Werte kleiner 0 auf 0 gesetzt.

Dabei wird verhindert, dass z.B. bei der Multiplikation weiße Bilder entstehen.

Der Normierungswert wird unter *Options/ Type Casting Settings* gewählt (siehe *Bild 35*).

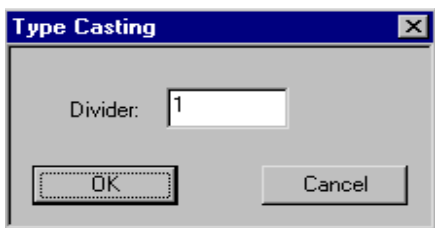


Bild 35: Normierungswert-Wahl

Der aktuelle Wert wird im unteren Fensterteil des *Arithmetic –* Menüs angezeigt.

Achtung!

Inkorrekte Einstellung der Normierungsvariablen führt zu unbefriedigenden Ergebnissen bei den arithmetischen Operationen (z.B. ganz schwarze oder weiße Bilder).

- Unter *Add Live Images* können bis zu 1000 aufeinanderfolgende Livebilder zu einem Bild summiert werden.
- Unter *Options / Addition Settings* kann die gewünschte Bilderanzahl gewählt werden (siehe *Bild 36*).

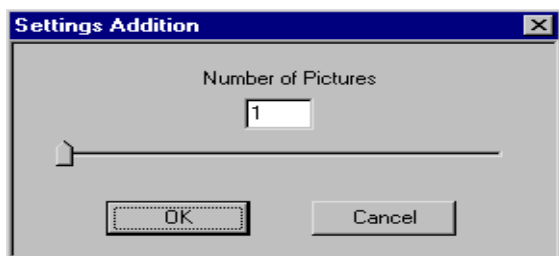


Bild 36: Anzahl der Bilder

So kann z.B. das Bildrauschen einer Aufnahme verringert oder bewegte Gegenstände gegenüber dem Hintergrund reduziert werden.

Nach der Addition wird das darzustellende Bild so normiert, dass die Gesamthelligkeit erhalten bleibt.

Der Vorgang dauert entsprechend der zu addierenden Bilder und Leistungsfähigkeit des Rechners.

Der Fortschritt des Prozesses wird am unteren Fensterrand in Prozent angezeigt.

Achtung!

Die Einstellung der Bildanzahl verändert gleichzeitig den Wert der Normierungsvariablen (Type Casting) so, dass das addierte Bild die gleiche Helligkeit hat wie ein Einzelbild.

Bei anschließender Verwendung der *Arithmetic* – Funktionen muss die Normierungsvariablen eventuell neu eingestellt werden.

• **Option Port Test:**

Der Menüpunkt *Features/Test Hardware* beinhaltet den Unterpunkt *Option Port*.

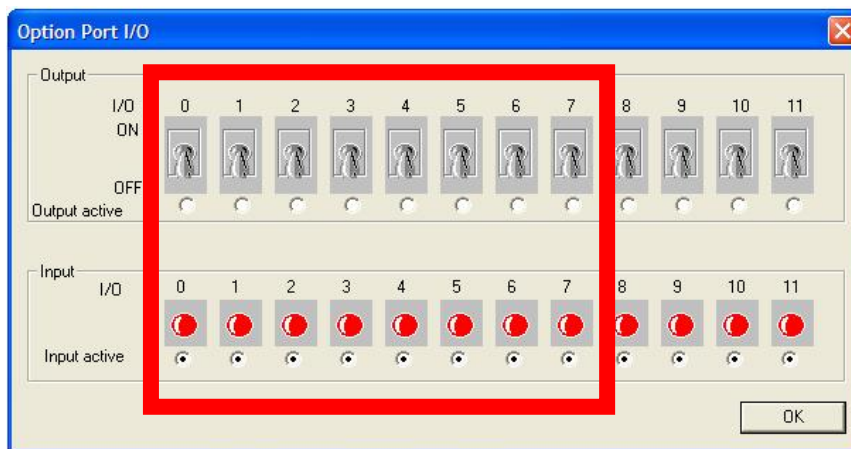


Bild 37: Option Port Menü

Mit diesem Menü ist es möglich die I/Os, welche auf dem Option Port (X300) zur Verfügung stehen, ein- oder auszuschalten. Es ist auch möglich, die Zustände an den I/Os einzulesen.

Nur die Kanäle 0 bis 7 sind den Pins des Option-Ports zugeordnet. Mit den Kanälen 8 bis 11 können Sie den Zustand der Jumper auf der Karte auslesen (siehe unten).

Wählen Sie zunächst aus, ob Sie einen bestimmten Pin als Ausgang (*Output active*) oder Eingang (*Input active*) verwenden möchten.

Bei Verwendung als Ausgang können Sie mit dem virtuellen Schalter den Signalzustand des Pins verändern.

Bei Verwendung als Eingang zeigen die roten Signallämpchen den Pegel des anliegenden Signals an.

Achtung!

Vergewissern Sie sich, dass kein externes Signal an einem Pin anliegt, den Sie auf „Ausgang“ schalten!

Dies kann den Pin oder den Decoder-Chip zerstören.

Beachten Sie beim Anlegen von externen Signalen die Spezifikationen der Eingänge. Zu hohe Spannungen oder das Anlegen von negativen Spannungen oder gestörten Signalen ohne Schutzmaßnahmen kann zur Zerstörung des Pins oder der Framegrabberkarte führen!

• **Jumper auslesen:**

Der Menüpunkt *Features/Test Hardware* beinhaltet den Unterpunkt *Option Port*.

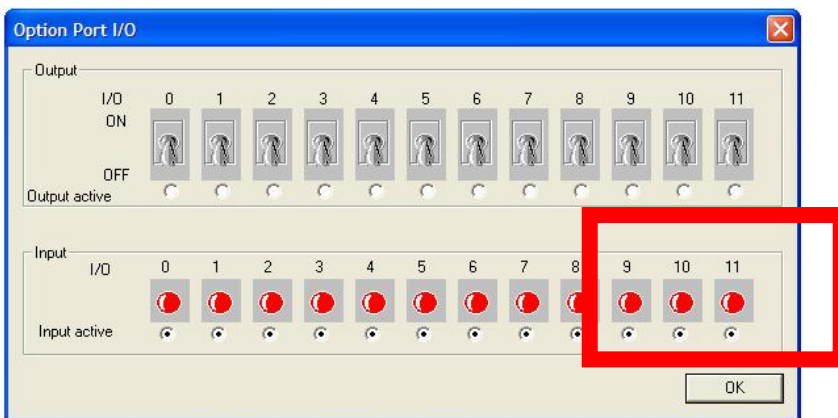


Bild 38: Option Port Menü (Jumper)

In diesem Menü ist es auch möglich, die Zustände der Jumper auszulesen.

Sind keine Jumper bestückt liegen I/O 8 bis 11 auf High-Pegel. Sie sind also rot markiert. Werden Jumper bestückt, erlischt die rote Markierung.

Hinweis:

Setzen Sie die I/O-Kanäle 8 bis 11 nicht auf „Output active“. Dies kann zur Zerstörung der Framegrabberkarte führen.

5.8 Bilder speichern, Programm beenden

Unter dem Eintrag *File* können Sie Livebilder (Momentaufnahme), Standbilder bzw. arithmetisch bearbeitete Bilder abspeichern. Mit *Save* oder *Save as* können die Bilder mit der vom Programm vergebenen Indexnummer bzw. mit einem eigenen vergebenen Namen abgespeichert werden.

Die Bilder werden im bmp-Format abgespeichert und können mit jedem Grafikprogramm betrachtet bzw. bearbeitet werden.

Mit *Close* kann das Standbild- sowie das Livefenster nacheinander geschlossen werden.

Mit *Exit* verlassen Sie das Programm.

5.9 Inbetriebnahme unter Linux

Zum Betrieb des pciGrabber-4x4 kann der BTTV-Grabbertreiber verwendet werden. Dieser Treiber ist bereits in den meisten Linux-Distributionen enthalten. Sie können also ohne besonderen Treiber arbeiten. Der Treiber muss zurzeit noch manuell an die pciGrabber-4x4 angepasst werden. Bitte lesen Sie hierzu die FAQs unter www.PHYTEC.de.

Hinweise:

- Durch die Verwendung der PCI Express-to-PCI-Bridge kann es vorkommen, dass die Buszuordnung für die einzelnen Geräte nicht mehr passt, da Linux diese nicht eigenständig aktualisiert. Die Geräte müssen in einem solchen Fall neu zugeordnet werden. Die Grafikkarte kann z.B. über den Befehl `SaX2` automatisch konfiguriert werden oder man nimmt manuell eine Änderung in der `xorg.conf` vor. Falls Sie weitere Informationen hierzu benötigen richten Sie sich bitte an den Anbieter ihrer Linux Distribution.

Teil 2

Programmierhandbuch

6 Treiber-Software

Dieser Kapitel erklärt, wie Sie den pciGrabber-4x4 aus eigenen Programmen heraus ansprechen können.

Die Treiber-Bibliotheken stellen Ihnen einen Satz von Funktionen zur Verfügung, mit denen Sie die Einstellungen des Grabbers verändern können, Statusinformationen abfragen und die Digitalisierung von Bildern einleiten können.

Softwaretreiber stehen Ihnen für verschiedene Betriebssysteme zur Verfügung.

In diesem Handbuch werden die Treiber für

- Windows XP/VISTA/7
- Windows NT4.0
- Windows 2000

erläutert.

Achtung!

Lesen Sie die readme.txt - Datei auf der Installations-CD, um neueste Informationen zu den Treibern und der Verfügbarkeit weiterer Treiber zu erhalten.

Im nächsten Kapitel werden einige technischen Grundlagen des Grabbers und der verwendeten Fernsehnormen erläutert, um Ihnen die Funktionsweise des Grabbers transparent zu machen.

6.1 Technische Grundlagen

6.1.1 Aufbau des pciGrabber-4x4

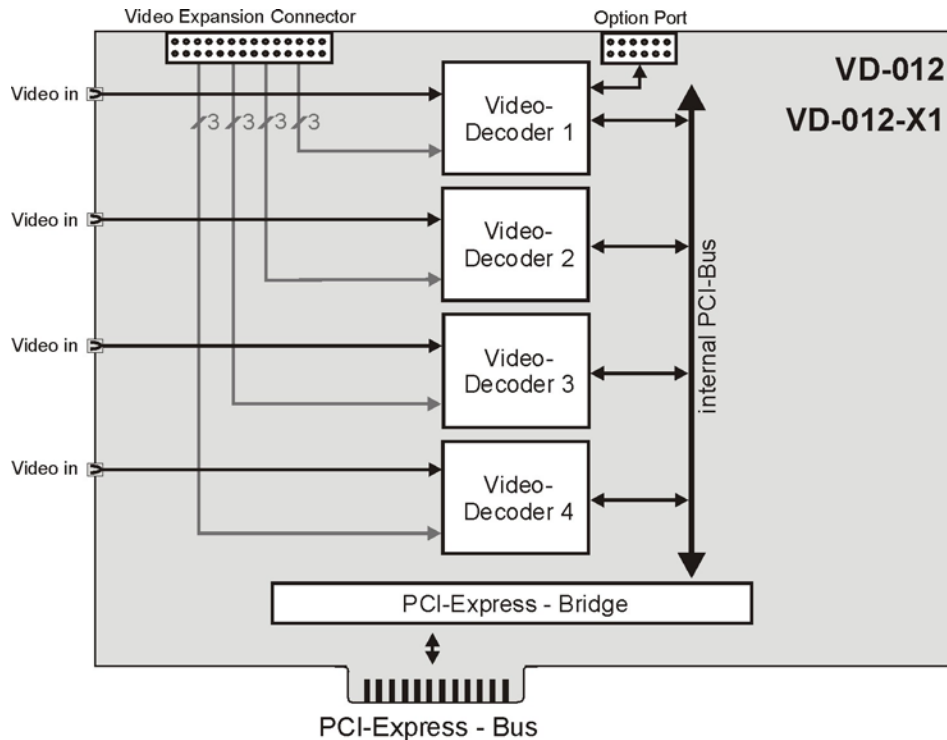


Bild 39: Blockschaltbild VD-012, VD012-X1

Bild 39 zeigt den Aufbau des VD-012 / VD-012-X1. Der pciGrabber-4x4 besteht aus vier identisch aufgebauten Video-Decodern, die mittels eines internen PCI-Busses mit der PCI-Express-Bridge verbunden sind. Jeder Decoder ist eine unabhängige Einheit.

Als Besonderheit stellt der erste Decoder die Funktionen des *Option Ports* zur Verfügung (GPIO, I²C).

Bei der Variante VD-012-X2 sind lediglich zwei Video-Decoder vorhanden. Da hier softwaremäßig jeweils zwischen Composite- und S-Video-Eingang ausgewählt bzw. umgeschaltet werden kann, unterscheidet sich das Signal-Routing geringfügig von den Modellen mit vier Decodern (Bild 40).

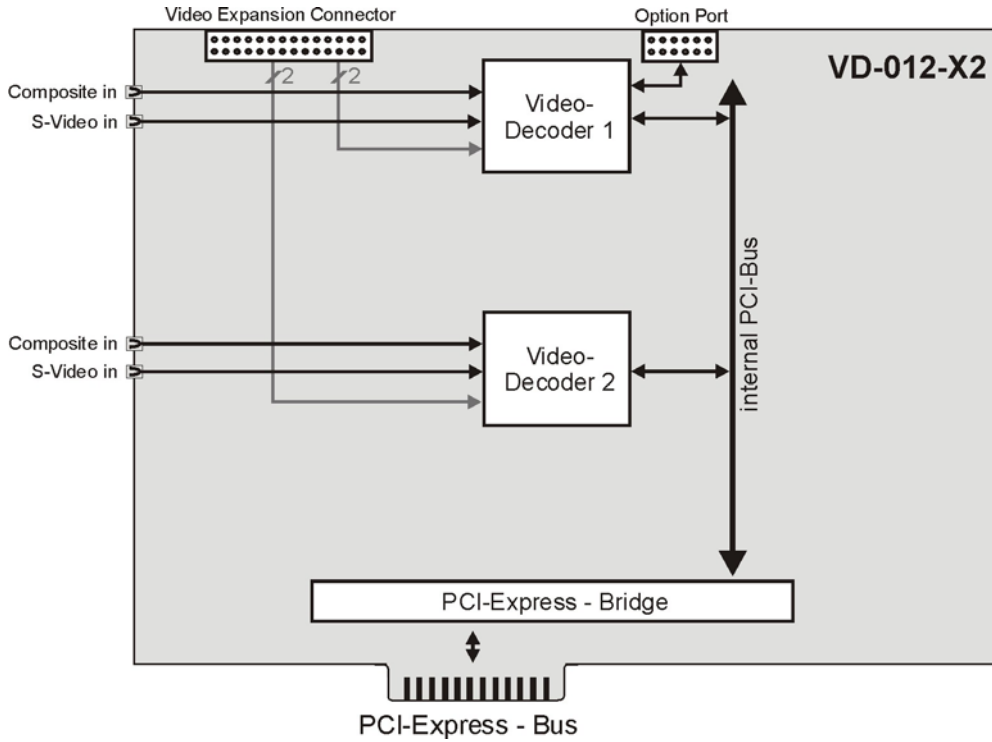


Bild 40: Blockschaltbild VD-012-X2

Bild 41 zeigt den Aufbau eines jeden Decoders. Alle Decoder sind gleich aufgebaut, außer dass lediglich der erste Decoder den *Option-Port* mit GPIOs und I²C-Bus zur Verfügung stellt.

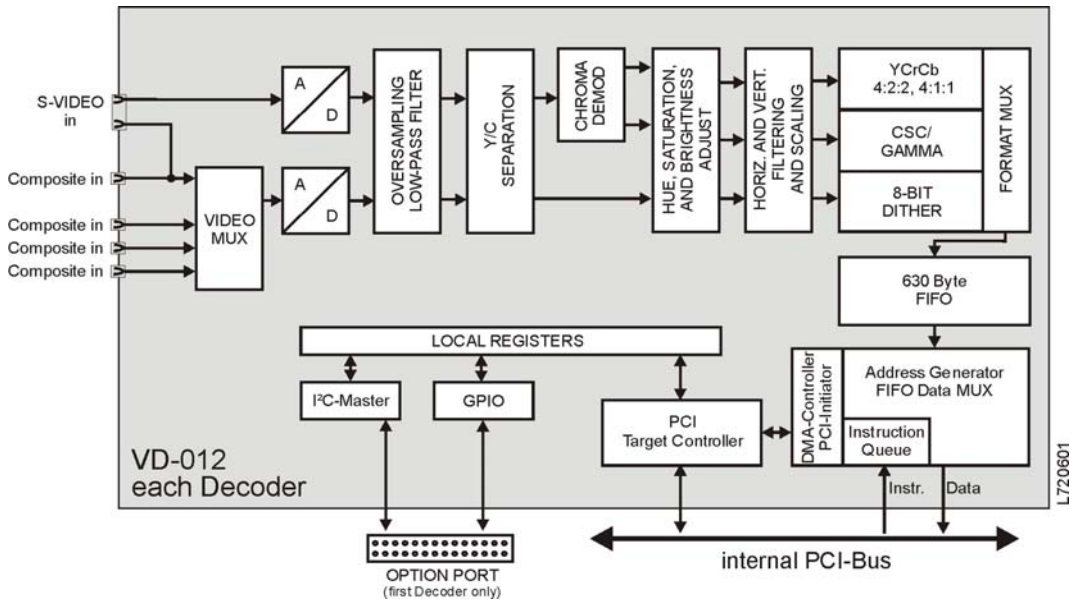


Bild 41: Blockschaltbild eines Video-Decoders

Das Composite-Eingangssignal wird zunächst über einen 4:1-Video-multiplexer geführt, den der Anwender über den PCI Express-Bus steuern kann. Der nachfolgende A/D-Wandler digitalisiert dieses Signal. Als Bildquellen eignen sich alle Geräte, die ein Farb-Videosignal gemäß den CCIR-Definitionen „PAL (B,G,H,I)“ und „NTSC (M)“.

Achtung!

In Europa benutzte Bildquellen liefern i.a. PAL-Fernsehsignale. In diesem Manual wird daher durchgehend die Verwendung von PAL-Quellen vorausgesetzt.

Über den *S-VIDEO*-Eingang können Luma- und Chroma-Signal getrennt eingespeist werden (z.B. von einer *S-Video*-Kamera oder einem S-VHS-Videorecorder). Für den Farbanteil wird hier ein getrennter A/D-Wandler benutzt, was die Bildqualität verbessert.

An den pciGrabber-4x4 lassen sich selbstverständlich auch Schwarzweiß-Bildquellen (*monochrom*) anschließen.

Die Verarbeitung von Grauwert-Bildern mit 256 Graustufen ist im Grabber vorbereitet und kann per Software aktiviert werden. Beim Einsatz von Schwarzweiß-Bildquellen können Sie die Bildschärfe verbessern, indem Sie die Farbfalle deaktivieren. Dies ist softwaremäßig möglich.

Nach den A/D-Wandlern folgen Funktionsgruppen, die den Bilddatenstrom in seine Komponenten zerlegen: Hinter dem Chroma-Demodulator liegen die Bilddaten getrennt nach Helligkeit (Y) und Farbanteilen (Cr/Cb) vor. Danach erfolgt die digitale Korrektur von Helligkeit, Kontrast und Farbsättigung und die Skalierung auf die gewünschte Bildgröße und -auflösung.

Im nachfolgenden *Video Format Converter* werden die vom pciGrabber-4x4 bereitgestellten Datenformate erzeugt. Über einen Datenmultiplexer wird das vom Anwender gewünschte Format ausgewählt und dem 630 Byte großen Pufferspeicher (FIFO) zugeführt.

Der FIFO dient als Bindeglied zum nachfolgenden PCI-Bus Interface, welches die Datenübertragung über den PCI-Bus organisiert.

Die Bilddaten werden per DMA direkt in den Arbeitsspeicher des PCs übertragen. Für jedes Halbbild wird ein eigener DMA-Kanal verwendet. Die Übertragung kann vom Anwender sehr flexibel gestaltet werden. Dazu dient eine *pixel instruction list* für jedes Halbbild, die im Folgenden als *RISC-Programm* für die im pciGrabber-4x4 integrierten PCI-Controller bezeichnet wird.

Das Prinzip der *pixel instruction list* wird in *Kapitel 6.1.2* näher erläutert.

Über den PCI-Controller wird des Weiteren der Zugriff zu den *lokalen Registern* abgewickelt. Diese ermöglichen die Einstellungen der Grabber-Parameter und das Rücklesen der aktuellen Statusinformation. Über diese Register erfolgt auch das Ansprechen der benutzerdefinierten I/O-Leitungen und der in den Grabber integrierten I²C-Schnittstelle.

Die PCI Express-to-PCI-Bridge wandelt den internen PCI-Bus in den PCI Express-Bus.

6.1.2 Videosignal und Digitalisierungsvorgang

Das vom Grabber verarbeitete analoge Norm-Videosignal besteht aus 625 Zeilen, die in zwei Halbbilder (fields) aufgeteilt sind und von der Videoquelle nacheinander geliefert werden. Das erste Halbbild (ungerade / odd) beinhaltet die Zeilen 1 bis 313, das zweite (gerade / even) die Zeilen 314 bis 625. Die Halbbilder sind ineinandergeschachtelt, um im TV-Bild den Flimmereffekt zu verringern. Räumlich gesehen folgt daher auf Zeile 1 die Zeile 314. Soll ein Vollbild dargestellt werden, müssen die beiden nacheinander empfangenen Halbbilder, entsprechend *Bild 42*, verschachtelt werden (Fernseher per elektrischer Schaltung, im PC-Graphikspeicher per Software).

Neben einigen Vor- und Nachlaufzeilen beinhaltet das Videosignal noch Prüf- und Datenzeilen sowie Zeilen mit Videotext-Information, so dass die effektive Bildfläche aus zwei Halbbildern mit je 288 Zeilen besteht.

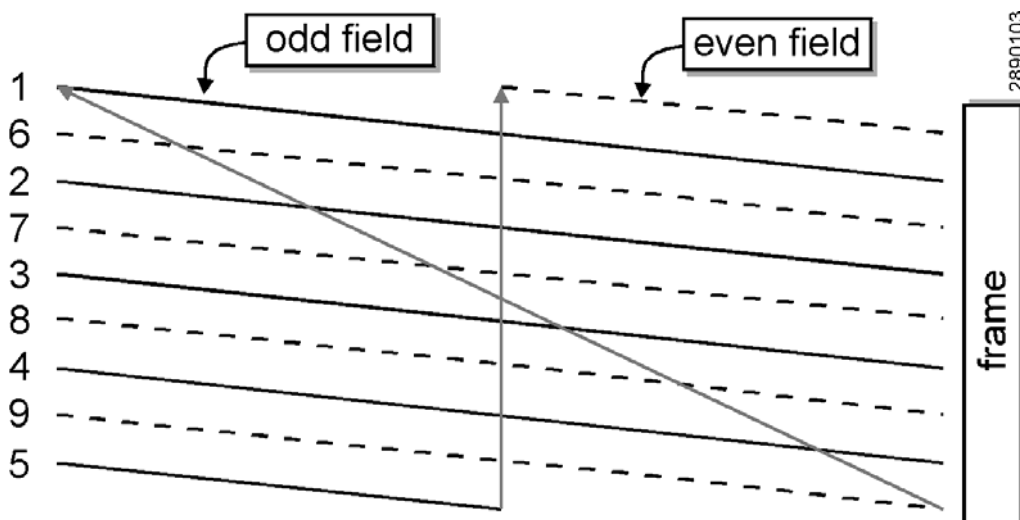


Bild 42: Zeilensprungverfahren (Beispiel mit 9 Zeilen)

Jedes Halbbild wird in 20 ms aufgebaut. In einem Halbbild ist bereits die vollständige Bildfläche zu erkennen, die vertikale Auflösung ist jedoch um die Hälfte reduziert. In vielen Anwendungen ist dies jedoch bereits ausreichend, so dass ein vollständiges Digitalisierungsergebnis bereits nach 20 ms vorliegt. Gegebenenfalls kann die Auflösung in X-Richtung ebenfalls verringert werden, um ein unverzerrtes Bild zu erhalten.

Eine Auflösungsverminderung in X-Richtung kann die Digitalisierung jedoch nicht weiter beschleunigen, da das Zeitraster des Bildaufbaus vorgegeben ist.

Wird die volle TV-Bildauflösung benötigt, so muss der Einzug beider Halbbilder abgewartet werden (40 ms). Die beiden Halbbilder folgen dabei zeitlich nacheinander.

Um die Verschachtelung der Halbbilder zu ermöglichen, ist die letzte Zeile des ungeraden (ersten) Halbbilds auf die Hälfte verkürzt. Demzufolge besteht die erste Zeile des zweiten Halbbilds nur aus der zweiten Zeilenhälfte.

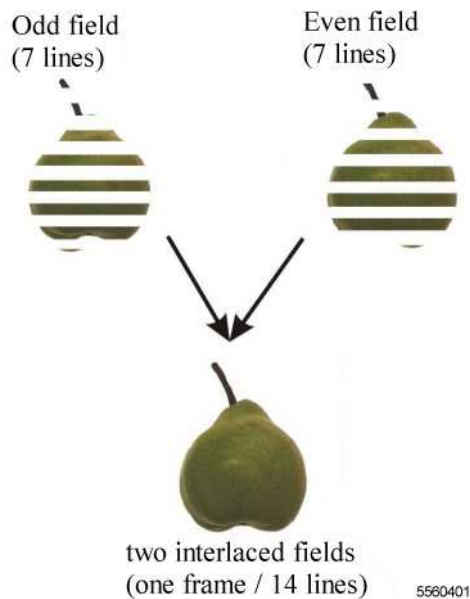


Bild 43: Halb- und Vollbilder

Ein Problem bei der Digitalisierung von TV-Bildern besteht darin, dass ein Objekt bei schneller Bewegung zwischen der Aufnahme des ersten und des zweiten Halbbilds schon ein merkliches Stück vorgerückt ist; die beiden Halbbilder passen also nicht mehr zueinander (Unschärfe / Verzerrungen). Auch aus diesem Grund benutzt man daher häufig - auf Kosten der vertikalen Auflösung - nur ein Halbbild.



Bild 44: Kammeffekt bei bewegten Objekten im Vollbild-Modus

6.1.3 Farbübertragung und Farbspeicherung

Farbinformation und Bildhelligkeit werden zur Bildübertragung von den TV-Systemen prinzipiell getrennt. Übertragen wird das Bildhelligkeitssignal (Luma-Signal, Y-Signal) und das Farbdifferenzsignal (Chroma-Signal). Letzteres charakterisiert die Farbe eines Bildpunkts durch die beiden Parameter Farbton und Farbsättigung.

Die Fernsehsysteme reduzieren die Übertragungsbandbreite des Farbsignals gegenüber dem Helligkeitssignal. Die Farbe wird also „unschärfer“ übertragen als die Helligkeit eines Bildpunkts. Dies entspricht anschaulich gesprochen dem Vorgehen eines Zeichners, der die Objektkonturen zunächst mit einer spitzen Feder zieht und das Bild anschließend mit einem groben Pinsel coloriert.

Die Y-Bandbreite beträgt beim PAL (B,G,H,I) - System 5 MHz, die des Chroma-Signals 1,5 MHz.

Das Chroma-Signal wird bei PAL auch als U/V-Signal, bei NTSC als Q/I-Signal bezeichnet. V- bzw. I-Signal charakterisieren Rottöne während das U- bzw. Q-Signal blauvioletten Farbtönen zugeordnet ist. Zusammenfassend spricht man vom Cr/Cb-Signal für (Chroma Red / Chroma Blue).

Über das Werte-Tripel (Y,Cr,Cb) werden Helligkeit und Farbe eines Bildpunkts vollständig definiert. Diese Werte lassen sich einem Bildverarbeitungssystem zur Farberkennung oder -kontrolle ohne weitere Vorverarbeitung zuführen. Häufig wird die Darstellung eines Bildpunkts in den Farbausügen Rot, Grün und Blau (R,G,B) gewünscht.

Die Umrechnung ist nach CCIR-Empfehlung für PAL durch folgende Matrix definiert:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1,371 & -191,45 \\ 1 & -0,338 & -0,698 & 116,56 \\ 1 & 1,732 & 0 & -237,75 \end{pmatrix} \cdot \begin{pmatrix} Y \\ U \\ V \\ 1 \end{pmatrix}$$

Der pciGrabber-4x4 kann die Bilder in das RGB-Format umrechnen und als RGB-Farbtripel im Bildspeicher ablegen. Bilder in diesem Format können gut weiterverarbeitet und angezeigt werden.

Zur Speicherung der Bildinformation oder zu Übertragungszwecken ist das YCrCb-Format oftmals günstiger, da es weniger Datenvolumen impliziert. Statt drei Byte pro Pixel sind nur zwei Byte (ein Wort) erforderlich. Die unteren acht Bit spezifizieren dabei die Helligkeit, die oberen acht den Farbanteil (Cr/Cb).

Als Farbanteil wird dabei zu jedem Y-Wert alternierend Cr und Cb geliefert. Man erhält so zu jedem Pixel nur die „halbe“ Farbinformation, entweder den Rot- oder den Blauanteil. Die jeweils fehlende Information kann durch Übernahme des Werts des Nachbarpixels gewonnen werden. Übertragen und gespeichert wird die Farbe also nur in der halben Auflösung der Helligkeit. Da die Farbinformation jedoch vom Fernsehsystem ohnehin bandbreitenreduziert übertragen wird (s.o.), wird durch diese Vorgehensweise kein Verlust verursacht. Dieses Datenformat wird als YCrCb 4:2:2 bezeichnet.

Mit dem ersten Pixel jeder Zeile wird $Y_1, Cr_{1/2}$ geliefert, mit dem zweiten $Y_2, Cb_{1/2}$ usw.

Achtung!

Zur richtigen Erkennung der Farbinformation sind vier aufeinanderfolgende Halbbilder erforderlich. Es ist deshalb nicht möglich, die Videoquelle nur zum Digitalisierungszeitpunkt kurz anzuschließen bzw. eine Videoquelle zu benutzen, die nur einzelne Halbbilder liefert. Ebenso arbeitet die Halbbildererkennung möglicherweise zunächst nicht korrekt, wenn auf ein anderes, unsynchronisiertes Kamerasignal umgeschaltet wird.

Beim schnellen Wechsel zwischen zwei Bildquellen muss daher damit gerechnet werden, dass ein direkt im Anschluss digitalisiertes Bild fehlerhaft ist. Besser ist es, eine entsprechende Wartezeit einzuhalten.

6.1.4 Datenspeicherung durch DMA und RISC-Programm

Dieses Kapitel befasst sich eingehender mit der Übertragung der Bilddaten in den Hauptspeicher und der Ablage der Pixel an den vom Anwender gewünschten Adressen.

Wie bereits oben erwähnt, erfolgt der Transfer der Bilddaten durch zwei DMA-Kanäle, einen für das Odd- und einen für das Even-Halbbild. Bezogen auf den internen PCI-Bus bedeutet das, dass im Augenblick der Bilddigitalisierung der DMA-Controller eines Decoders des pciGrabber-4x4 die Kontrolle über den PCI-Bus beansprucht, also *Master* auf dem Bus wird.

Die Bilddaten werden in Echtzeit über den PCI-Bus zur PCI Express-to-PCI-Bridge geschickt. Von dort aus werden die Daten über den PCI Express-Bus in den Arbeitsspeicher übertragen. Dies wird durch die hohe Datenübertragungsrate des PCI Express-/PCI-Busses ermöglicht.

Verzögerungen der Datenübertragung bzw. Zeiträume, in denen der PCI Express-/PCI-Bus dem Grabber nicht zur Verfügung steht, (er also die Kontrolle als Master an eine andere Karte abgeben muss) werden mit Hilfe des FIFO-Speichers überbrückt. Dies schließt jedoch längere Unterbrechungen des Datentransfers während eines Digitalisierungsvorgangs aus, da dann der FIFO-Speicher überläuft und Teile des Bildes verloren gehen. Im Allgemeinen ist dies jedoch unproblematisch (siehe Kapitel 5.1).

Mittel zur Kontrolle der Busvergabe sind die Parameter *Maximum_Latency* und *Minimum_Grant* der Karten. Diese werden vom Treiber automatisch eingestellt.

Im Bedarfsfall müssen sie an die zu übertragende Datenrate, die Systemkonfiguration und die Bus-Performance angepasst werden.

Der pciGrabber-4x4 weist eine hohe Flexibilität auf, was die Datenspeicherung betrifft. Der Anwender kann Ziel und Format der Bilddaten in gegebenen Rahmen frei bestimmen. Dazu ist ein

Mechanismus erforderlich, den kontinuierlichen Bilddatenstrom zu partitionieren und an die gewünschte Speicheradresse zu lenken.

Beim pciGrabber-4x4 wird dies mit Hilfe der *pixel instruction list* gelöst. Dabei handelt es sich um ein RISC-Programm, welches den DMA-Controller entsprechend steuert.

Dieses RISC-Programm muss von der Anwendung entsprechend der Aufgabenstellung, dem Daten- und Bildformat erstellt werden. Es ist also individuell an jedes Problem angepasst. Dies bedingt in den meisten Fällen, dass das RISC-Programm erst zur Laufzeit des Anwenderprogramms erstellt wird, denn oft sind Parameter, die das RISC-Programm beeinflussen, variabel (z.B. die Bildgröße).

Die mit dem pciGrabber-4x4 gelieferten Software-Treiber erstellen das jeweils passende RISC-Programm automatisch bei der Einstellung der Bildgröße. Dieser Vorgang ist für das Anwenderprogramm transparent. Dennoch sollte dem Programmierer auch bei der Anwendung der Treibersoftware dieser Zusammenhang bewusst sein.

Bild 45 zeigt die Verhältnisse im Überblick. Das Anwenderprogramm benutzt zur Einstellung von Bildgröße und Datenformat die Funktion **set_image()** des Treibers. Der Treiber führt daraufhin zwei Aktionen durch: Er stellt über einen PCI-Bus-Zugriff auf die lokalen Register des Grabbers die Bildgröße im *VideoScaler* ein. Dies sorgt dafür, dass der pciGrabber-4x4 die richtige Bildgröße erzeugt, der Datenstrom also richtig formatiert und am Ende einer Zeile bzw. eines Bildes mit entsprechenden Synchronzeichen versehen wird.

Auf dem gleichen Weg wird der *DataFormatConverter* auf das gewünschte Datenformat eingestellt. In das FIFO gelangt damit ein formatierter Pixeldatenstrom im gewünschten Format (z.B. RGB).

Die zweite Aktion, welche die Treibersoftware durchführt, ist die Erstellung eines auf den Bilddatenstrom angepassten RISC-Programms, das im Arbeitsspeicher des PCs abgelegt wird. Der DMA-Controller des pciGrabber-4x4 erhält Kenntnis von der Startadresse dieses Programms. Bei der Durchführung eines Digitalisierungsvorgangs holt sich der DMA-Controller sequentiell per DMA die RISC-Befehle aus dem Arbeitsspeicher, verarbeitet sie und speichert den Bilddatenstrom entsprechend diesen Anweisungen.

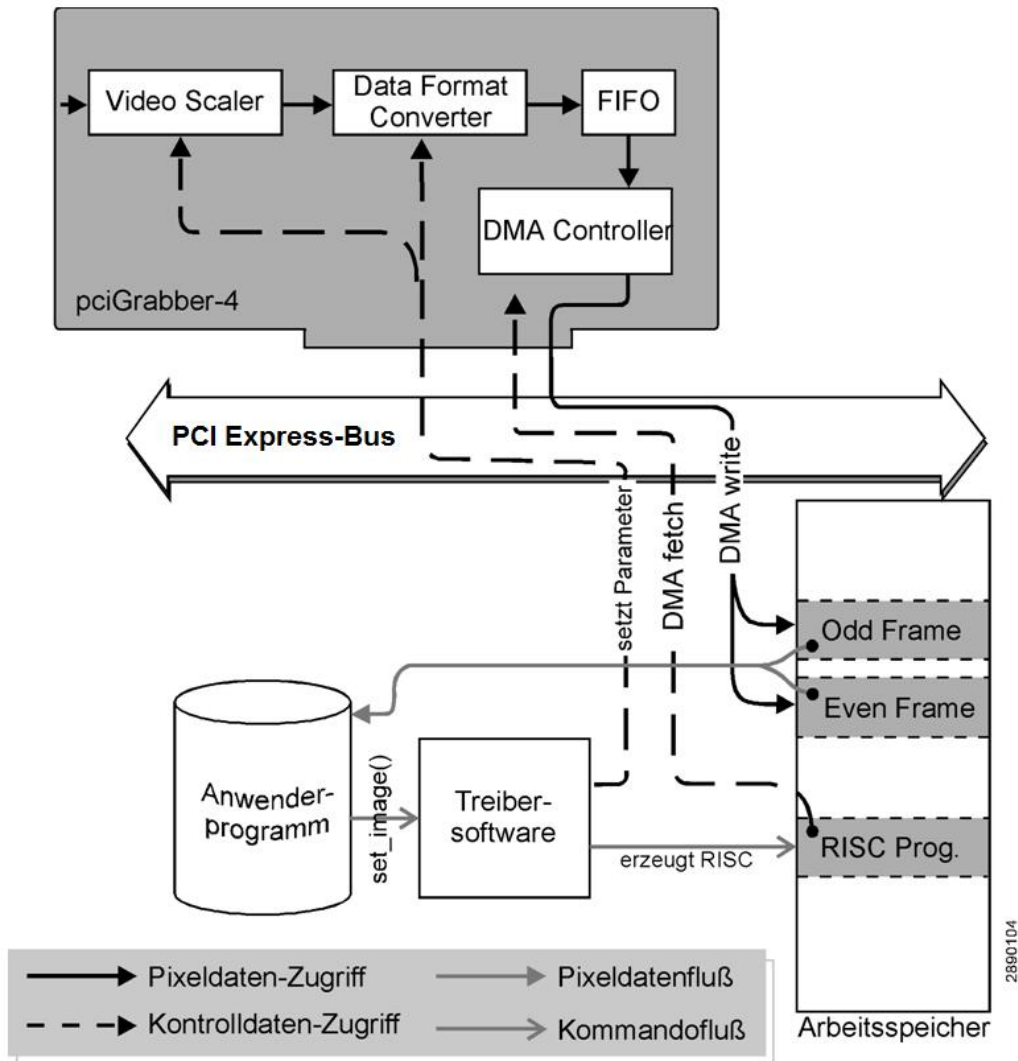


Bild 45: Übersicht über Pixel und Kontrolldatenfluß

Der Bilddatenstrom wird über DMA-Schreibzugriffe direkt in den Arbeitsspeicher geleitet und zwar an die Adressen, die im RISC-Programm spezifiziert sind. Dies sind Adressbereiche, die das Anwenderprogramm zuvor reserviert hat (z.B. durch die Definition von Arrays).

Es kann sich dabei - wie in *Bild 45* gezeigt - um zwei getrennte Bereiche für Odd- und Even-Bild handeln oder aber um einen Speicherbereich, in dem der pciGrabber-4x4 ein Vollbild entsprechend der oben diskutierten Verzahnung von Odd- und Even-Bild aufbaut. Selektiert werden diese Optionen durch einen Übergabeparameter von `set_image()`, der wiederum die Erstellung des RISC-Programms durch den Treiber entsprechend steuert.

Da das Anwenderprogramm die Adressbereiche, in denen die Bilddaten gespeichert werden, selbst erzeugt hat, ist ihm die Position dieser Bereiche im Arbeitsspeicher (z.B. als Zeiger darauf) bekannt. Es kann somit direkt und wahlfrei auf diese Daten zugreifen, ohne den Treiber benutzen zu müssen.

Der Treiber stellt Statusinformationen zur Verfügung, die dem Anwenderprogramm mitteilen, wann der Bildaufbau im Speicher beendet ist, die Daten also vollständig vorliegen.

Die realisierte Struktur stellt sicher, dass das Anwenderprogramm schnellstmöglich auf die Daten zugreifen kann. Die Geschwindigkeit ist Echtzeit, bezogen auf das Standard-TV-Format. Für einen Halbbildaufbau bedeutet dies eine Zeitdauer von 20 ms, ein abgeschlossener Vollbildaufbau dauert 40 ms. Zu dieser Zeitspanne, in der die Bildaufnahme erfolgt, sind gegebenenfalls Verzögerungszeiten hinzuzurechnen, um die tatsächliche Zeitdauer zwischen Bildanforderung und Abschluss des Digitalisierungsvorgangs zu bestimmen. Dies ist zunächst die Wartezeit bis zum Beginn des gewünschten Halbbilds.

Wird beispielsweise die Digitalisierung eines Even-Halbbilds gewünscht, von der Kamera jedoch gerade ein angebrochenes Even-Bild geliefert, so vergeht der Rest dieses Bildes und das darauffolgende Odd-Bild, bis wieder der Anfang eines Even-Bilds erkannt wird und die Digitalisierung beginnen kann.

Die Wartezeit beträgt in diesem Fall fast 40 ms (zwei Halbbilder, worst case). Darauf wird das Bild digitalisiert und liegt 20 ms später vollständig vor. Die nächsten 20 ms geschieht in diesem Speicherbereich nichts, da in dieser Zeit ein Odd-Bild angeliefert wird.

Wird eine kontinuierliche Digitalisierung gewünscht, so wird dann in den darauffolgenden 20 ms erneut der Speicherbereich des Even-Bilds mit Bildinformation beschrieben.

Es muss beachtet werden, dass in diesem Zeitraum die alte Bildinformation in diesem Speicherbereich kontinuierlich durch neue ersetzt wird. Dies kann zu Fehlinterpretationen des Bildinhalts durch die Software führen, insbesondere bei bewegten Szenen.

Abschließend sei der Sonderfall der Vollbild-Digitalisierung in einem Speicherbereich betrachtet. Hier tritt der gleiche Effekt auf, jedoch in anderer Ausprägung. Nach 20 ms Digitalisierungszeit liegt die Information des Odd-Bildes vollständig vor, alle ungeraden Bildzeilen sind also definiert, während die geraden undefiniert sind. In den folgenden 20 ms werden die Zeilen des geraden Halbbilds ergänzt. Da sofort darauf wieder ein Odd-Bild geliefert und bei kontinuierlicher Digitalisierung im Arbeitsspeicher abgelegt wird, gibt es in dieser Betriebsart keinen Zeitpunkt, in dem kein Einschreibvorgang in den Speicherbereich erfolgt (von den Austastlücken abgesehen). Es gibt hier also immer einen Punkt (X,Y) an dem alter Bildinhalt in neue Bildinformation übergeht und damit ein Bildversatz entstehen kann.

6.2 Treiber für Microsoft Windows

Bei der Ausführung des Installationsprogramms für das Windows Demoprogramm werden die Dateien entpackt und auf der Festplatte gespeichert. Die Verzeichnisstruktur nach der Installation sieht wie in *Bild 46* dargestellt aus. Der im linken Fensterbereich dargestellte Pfadname kann bei der Installation editiert werden und daher auf Ihrem System anders erscheinen. Die zur Compilierung erforderlichen Library- und Include-Dateien sind in den entsprechend bezeichneten Unterverzeichnissen enthalten.

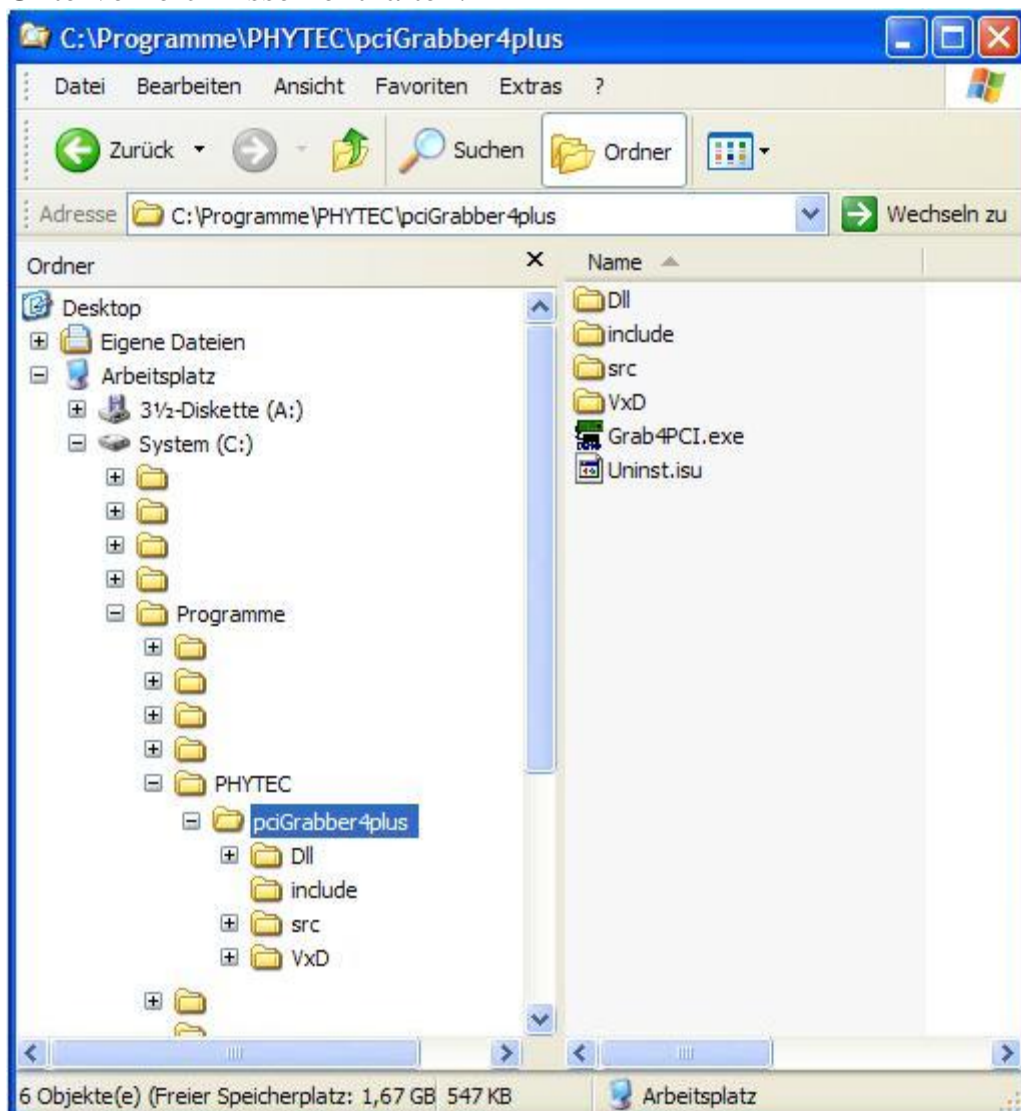


Bild 46: Verzeichnisstruktur der Windows Treiber

6.2.1 Voraussetzungen

Sie können Programme für den pciGrabber-4x4 mit Hilfe verschiedener Entwicklungsumgebungen erstellen.

Die erstellte Applikation ist unter den Betriebssystemen Microsoft Windows XP/VISTA/7 und Windows NT4.0/2000 einsetzbar.

Achtung!

Zum Betrieb des pciGrabber-4x4 unter Windows ist es erforderlich, dass der Gerätetreiber und die zugehörige DLL in das Windows-Systemverzeichnis kopiert werden. Zusätzlich muss der Treiber dem System durch Eintrag in die Registry-Table bekannt gemacht werden. Das von uns gelieferte Setup-Programm auf der CD führt diese Vorgänge auf Ihrem Rechner automatisch durch. Daher sind dort die Voraussetzungen für den Betrieb bereits geschaffen.

Um Ihre eigene Applikation auf anderen Rechnern betriebsbereit einzurichten, empfehlen wir Ihnen die Erstellung entsprechender Installations-Routinen

6.2.2 Einrichten des Gerätetreibers für Windows NT4.0

Da der pciGrabber-4x4 per DMA-Zugriff auf diesen Speicher zugreift, ist es erforderlich, dass es sich um physikalisch fixierten und kontinuierlich adressierbaren Speicher handelt. Deswegen muss der physikalisch zusammenhängenden Speicher allokiert werden, der als Bildspeicher fungiert. Der Treiber erlaubt außerdem den Zugriff auf die Register des pciGrabber-4x4.

Zur Auslieferung des Windows NT4.0 Treibers mit Ihrer eigenen Applikation können Sie die von uns ausgelieferte Installationsdiskette verwenden.

Diese befindet sich auf der Installations-CD im Verzeichnis **PCIGRAB4\DRIVER\WINNT40**. Sie können die Dateien in diesem Verzeichnis auf eine Diskette kopieren und zusammen mit Ihrer Applikation vertreiben.

Alternativ dazu können Sie auch ein eigenes Installationsprogramm erstellen. Dabei sind folgende Punkte zu beachten:
Dem Windows NT4.0-System muss der Gerätetreiber durch einen Eintrag in der Registry bekanntgemacht werden. Dazu gehen Sie folgendermaßen vor:

- Zunächst muss die Datei **PCIGRABBER4.SYS** in das Verzeichnis **<Windows>\System32\drivers** kopiert werden.
- Als nächstes wird der Treiber in die *Registry-Table* eingetragen: Benutzen Sie dazu das Programm REGEDIT im WINDOWSNT - Verzeichnis. Durchlaufen Sie den Registrierungsschlüsselbaum bis zu der Eintragsgruppe *Services* (siehe *Bild 47*).

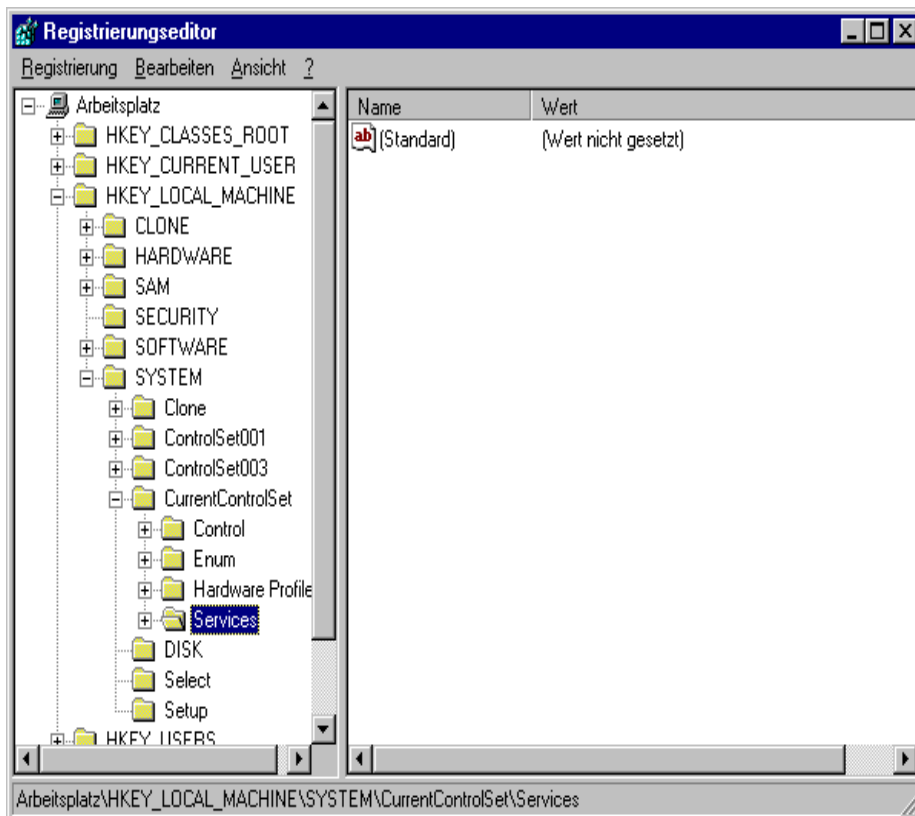


Bild 47: Windows NT Registrierungs-Editor

Erweitern Sie nun die Schlüsselgruppe *Services* um den Eintrag „pciGrabber4“, indem Sie über den Menüpunkt *Bearbeiten/Neu/Schlüssel* einen neuen Schlüssel erzeugen und ihm den Namen „pciGrabber4“ zuweisen, wie in Bild 48 gezeigt.

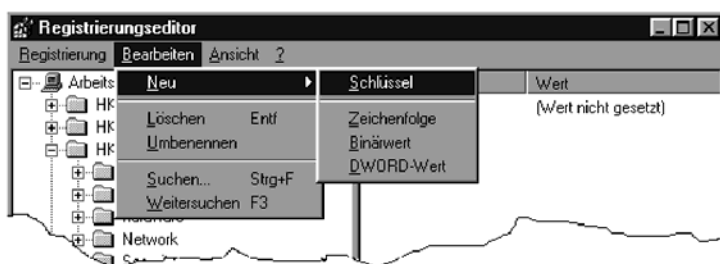


Bild 48: Hinzufügen eines Gerätetreiber-Eintrages

Jetzt konfigurieren Sie die neue Schlüsselgruppe: Markieren Sie den Eintrag „pciGrabber4“ wie in Bild 48.

Wählen Sie aus *Bearbeiten/Neu* den Menüpunkt *DWORD-Wert*. Innerhalb des Schlüssels „pciGrabber4“ wird ein Eintrag erstellt, der die Bezeichnung „Neuer Wert #1“ hat.

Ändern Sie diesen in „Start“. Klicken Sie dann mit der rechten Maustaste auf den erstellten Eintrag und wählen *Ändern* aus. In dem nun erscheinenden Dialog tragen Sie in dem Feld *Wert* die Zahl 2 ein. Wählen Sie aus *Bearbeiten/Neu* die Option *DWORD-Wert*. Ändern Sie die Bezeichnung in „Type“ und geben Sie dem Eintrag den Wert 1.

Anschließend wählen Sie aus *Bearbeiten/Neu* den Menüpunkt *DWORD-Wert* und erzeugen analog zu oben den DWORD Eintrag „ErrorControl“ mit dem Wert „1“. Das Endergebnis soll so wie in *Bild 49* dargestellt aussehen.

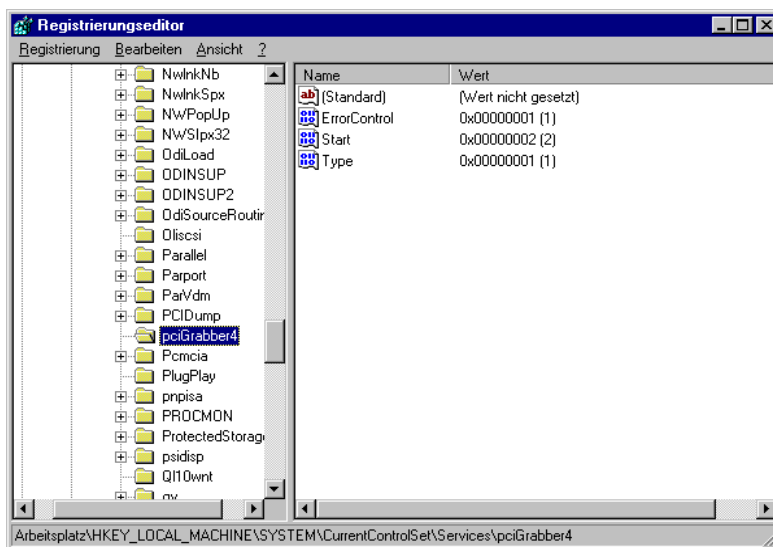


Bild 49: Konfigurieren des Treibers

Nach einem Neustart des Rechners wird jetzt der Treiber automatisch beim Start von Windows NT geladen.

Achtung!

Bei der Deinstallation des Anwenderprogramms sollte in jedem Fall der Gerätetreiber entfernt werden. Dazu ist er aus der Registry-Table auszutragen und aus dem System-Verzeichnis zu löschen.

Der Treiber belegt für den pciGrabber-4x4 einen Arbeitsspeicherbereich von ca. 1,2 MByte, der anderen Applikationen nicht zur Verfügung steht.

Achtung!

Bitte gehen Sie bei der Änderung des Registry-Eintrags mit der gebotenen Sorgfalt vor, da Sie bei fehlerhaften Änderungen die Konfiguration zerstören können. Dies kann den vollständigen Verlust der Funktionsfähigkeit Ihres Windows NT-Systems zur Folge haben. Für den Endanwender sollten Sie ein entsprechendes Install- und Uninstall-Programm vorsehen, das diese Aufgaben automatisiert.

6.2.3 Einrichten des Gerätetreibers für Windows 2000 / XP / VISTA / 7

Wie unter Windows NT4.0 dient der Gerätetreiber dazu, physikalisch zusammenhängenden Speicher zu allokiieren, der als Bildspeicher fungiert. Der Treiber erlaubt außerdem den Zugriff auf die Register des pciGrabber-4x4

Die Reservierung eines solchen Speichers ist in Windows 2000/XP/Vista/7 nur mittels eines Gerätetreibers möglich.

Durch den Treiber werden ferner die linearen Speicheradressen in physikalische Speicheradressen umgewandelt.

Der Treiber wird vom Anwenderprogramm nicht direkt angesprochen. Der gesamte Zugriff auf ihn erfolgt durch die DLL.

Zur Auslieferung des Windows 2000/XP/Vista/7 Treibers mit Ihrer eigenen Applikation können Sie die von uns ausgelieferte Installations-CD verwenden.

Diese befindet sich auf der Installations-CD im Verzeichnis **PCIGRAB4\DRIVER\WINDOWS**. Sie können die Dateien in diesem Verzeichnis auf eine Diskette kopieren und zusammen mit Ihrer Applikation vertreiben.

6.2.4 Anwendung der DLL

Aufgabe der DLL ist die Kommunikation zwischen Anwenderprogramm und pciGrabber-4x4. Über die DLL können die Einstellungen des Grabbers vorgenommen und Digitalisierungsvorgänge gesteuert werden. Sie ermöglicht weiterhin den Zugriff auf die digitalisierten Bilddaten im Arbeitsspeicher.

Achtung!

Die DLL wird nicht in das Anwenderprogramm eingebunden, sondern erst zur Laufzeit aufgerufen. Sie muss daher zur Programmlaufzeit im Windows-Systemverzeichnis vorhanden sein.

Zusätzlich zur *GR4CDLL.DLL* sind folgende weitere DLLs zum Betrieb erforderlich:

- **MSVCRT.DLL**
- **CTL3D32.DLL**
- **MFC42.DLL**

Die DLL wird dynamisch eingebunden. Dazu stellt Windows verschiedene API-Funktionen zur Verfügung. Zum Laden der DLL wird die Funktion **LoadLibrary(...)** benutzt. Man erhält einen Handle auf die DLL zurück. Zur Ermittlung der Einsprungadressen der verschiedenen DLL-Funktionen wird die API-Funktion **GetProcAddress(...)** verwendet. Die Freigabe der DLL am Programmende wird durch Aufruf der Funktion **FreeLibrary(...)** vorgenommen.

Weitere Informationen können Sie aus der Dokumentation Ihrer Entwicklungsumgebung oder den im beiliegenden SDK befindlichen Beispiel-Quellen entnehmen.

6.2.5 Anwenden der Windows NT4.0™ / Windows 2000™ / XP / VISTA / 7 - DLL

Zur Verwendung der DLL *Gr4CDLL.DLL* in eigenen Applikationen muss der Softwareentwickler zunächst für jede Funktion, die in der Applikation verwendet wird, einen Funktionszeiger deklarieren.

Beispiel:

- zu verwendende Funktion: **WORD Get_Error(void)**
- Deklaration des Funktionszeigers:

```
WORD (PASCAL * lpfm_GetError)(void);
```

Um den Bezug zwischen dem Funktionszeiger und der DLL herzustellen, wird die Funktion **GetProcAddress(...)** benutzt.

Beispiel:

```
lpfn_GetError = (WORD(PASCAL *)(void))  
                GetProcAddress(handle, „Get_Error“);
```

Nun kann die Funktion wie folgt aufgerufen werden:

```
WORD Fehlerstatus;  
...  
Fehlerstatus = lpfm_GetError();
```

Achtung!

Prüfen Sie **unbedingt** den Wert des Funktionszeigers (Rückgabewert von `GetProcAddress`) auf `= NULL`, um sicherzustellen, dass die auf dem (Anwender-) Rechner installierte Treiber-Version die zu verwendende Funktion unterstützt und Sie einen gültigen Handle zurückerhalten haben!

6.2.6 Programmierung unter Delphi

Um die Funktionen der DLL in Delphi bekanntzumachen, muss eine entsprechende *Unit* definiert werden. Beachten Sie dabei die Angabe der richtigen Aufrufkonvention, um die Kompatibilität der DLL sicherzustellen. Definieren Sie Funktionen vom Typ `stdcall`. Bei falscher Deklaration können ansonsten Stacküber- bzw. unterläufe auftreten, die zu einer Schutzverletzung führen.

Im Folgenden ist eine Beispiel-Unit definiert:

```
unit grab4dll;

interface

{ Die Aufrufkonvention 'stdcall' legt die Reihenfolge der
Parameterübergabe auf dem Stack fest und signalisiert Delphi, dass
die gerufene Funktion den für die Parameter benötigten
Stackbereich selbst wieder freigibt }

function Grab4_Get_Error: word;
    stdcall; external 'gr4cdll.dll' name 'Get_Error';

function Grab4_Max_Device_Number: word;
    stdcall; external 'gr4cdll.dll' name
'Max_Device_Number';

function Grab4_Data_Present(nDevNo: word): word;
    stdcall; external 'gr4cdll.dll' name 'Data_Present';

function Grab4_GetPictureBufferAdress(nDevNo: word;
    dwBitsSize: Cardinal):
cardinal;
    stdcall; external 'gr4cdll.dll' name 'Data_Present';

procedure Grab4_Initialize(nDevNo: word);
    stdcall; external 'gr4cdll.dll' name 'Initialize';

procedure Grab4_Set_Channel(nDevNo, nChannel: word);
    stdcall; external 'gr4cdll.dll' name 'Set_Channel';

procedure Grab4_Start_Grabber(nDevNo: word);
    stdcall; external 'gr4cdll.dll' name 'Start_Grabber';

procedure Grab4_Stop_Grabber(nDevNo: word);
    stdcall; external 'gr4cdll.dll' name 'Stop_Grabber';
```

pciGrabber-4x4

```
procedure Grab4_Set_Image(nDevNo: word;
                          nOhpos, nOvpos, nOhsize, nOvsize,
                          nOppl, nOlines, nOColformat :word;
                          nEhpos, nEvpos, nEhsize, nEvsize,
                          nEppl, nElines, nEColformat :word;
                          nColsystem:word;
                          nInterlaced:word;
                          nSingleShot:word);
    stdcall; external 'gr4cdll.dll' name 'Set_Image';

const
    NTSC_M:    word    = 0;
    PAL_BDGHI: word    = 1;
    SECAM:     word    = 2;
    PAL_M:     word    = 3;
    PAL_N:     word    = 4;
    AUTO:      word    = 5;

    RGB32:     word    = 0;
    RGB24:     word    = 1;
    RGB16:     word    = 2;
    RGB15:     word    = 3;
    YUY2:      word    = 4;
    BtYUV:     word    = 5;
    Y8:        word    = 6;
    RGB8:      word    = 7;

implementation

{ DLL Funktionen }

end.
```

6.2.7 Beschreibung der in der DLL vorhandenen Funktionen

Über die Funktionen der DLL steuern Sie alle Vorgänge im pciGrabber-4x4 und können den aktuellen Status sowie die eingestellten Werte zurücklesen. Im Folgenden werden diese Funktionen ausführlich beschrieben.

Wichtig!

Alle Funktionen beziehen sich immer auf **genau einen**, individuellen Video-Decoder.

Sie können somit jeden Decoder unterschiedlich von den anderen einstellen.

Beachten Sie umgekehrt, dass keine Einstellung sich auf die Funktion eines anderen Decoders auswirkt.

Möchten Sie beispielsweise bei einem VD-012 die Helligkeit aller Kamerabilder erhöhen, so müssen Sie die entsprechende Funktion vier mal aufrufen und jeweils die entsprechende Decoder-Nummer (nDevNo) angeben.

Zur besseren Übersicht sind die Funktionen in fünf Gruppen aufgeteilt. Die Nummer der jeweiligen Gruppe ist der Funktionsbeschreibung in einem schwarzen Kreis vorangestellt.

Die Funktionen sind wie folgt klassifiziert:

❶ Routinen zur Initialisierung / Hardwareabfrage

Diese Gruppe beinhaltet Routinen, die Sie ein Mal vor der Verwendung des Grabbers aufrufen müssen, damit der Grabber korrekt arbeitet. Des Weiteren finden Sie hier zwei Funktionen, die Ihnen Auskunft über die installierte Hardware und deren Fähigkeiten geben.

❷ Routinen, die den Grabber zur Bildaufnahme einstellen

Funktionen aus dieser Gruppe stellen den Grabber auf die angeschlossene Bildquelle (Kamera) ein und legen fest, wie das vom Grabber erzeugte Ergebnisbild im Speicher aussieht (Bildgröße, Farbformat usw.). Sie sollten für jede Funktion aus dieser Gruppe prüfen, ob sie verwendet werden darf und welche Parameter zutreffend sind. Gegebenenfalls werden diese Routinen

im Programmverlauf mehrfach aufgerufen (z.B. wenn der Eingangskanal umgeschaltet werden soll oder die Bildgröße verändert wird).

- ③ **Routinen zur Durchführung und Kontrolle des Grabbvorgangs**
Mit diesen Funktionen starten Sie die Bild-Digitalisierung, überwachen den Grabbvorgang und beenden die Digitalisierung.
- ④ **Routinen zur Einstellung von Bildparametern**
Funktionen aus dieser Gruppe ermöglichen die Einstellung von Parametern wie Bildhelligkeit, Kontrast, Farbsättigung usw. Sie müssen nicht verwendet werden, können aber jederzeit aufgerufen werden, um das Ergebnisbild an die Bedürfnisse anzupassen.
- ⑤ **Routinen zur Steuerung der Erweiterungsports**
Unter diese Kategorie fallen alle Funktionen, die nicht direkt mit der Bildaufnahme zu tun haben, sondern die weiteren Features des Grabbers wie I/O-Port, I²C-Schnittstelle usw. betreffen. Diese Funktionen müssen nur aufgerufen werden, wenn ein entsprechendes Grabber-Feature verwendet wird.

Achtung!

In allen nachfolgend beschriebenen Routinen wird der Parameter `nDevNo` verwendet. Dieser Parameter identifiziert den gewünschten Decoder, auf dem Sie die entsprechende Operation ausführen wollen.

Haben Sie mehrere pciGrabber-Karten im Rechner installiert, so werden alle vorhandenen Decoder durchnummeriert.

Dabei ist es unerheblich, auf welchen Einsteckkarten sich die Decoder verteilen.

Die Anzahl der vorhandenen Decoder kann mit der Funktion `Max_Device_Number()` bestimmt werden.

Kompatibilität zum pciGrabber-4 /plus / express

Der Treiber ist grundsätzlich abwärtskompatibel. Programme, die für den pciGrabber-4, pciGrabber-4plus oder pciGrabber-4 express geschrieben wurden, arbeiten also auch mit dem pciGrabber-4x4.

Zu beachten ist, dass die anderen Modelle einen anderen Option Port haben und dort der Code angepasst werden muss. Des Weiteren ist die Anzahl der Videoeingänge unterschiedlich.

Funktionen, die nicht zu alten Treiber-Versionen für den pciGrabber-4 kompatibel sind, sind im Folgenden mit einem Stern (☆) gekennzeichnet.

Diese Applikationen arbeiten grundsätzlich auch mit anderen Grabberkarten der pciGrabber-4 – Familie zusammen, allerdings können manche Funktionen nicht genutzt werden, weil die entsprechenden Hardware-Voraussetzungen nicht erfüllt sind. In jedem Fall sollten Sie mit einer neuen Applikation auch die neue Treiberversion verwenden.

● Fehlermeldungen auswerten

WORD Get_Error(void);

Returnwert:	0 = kein Fehler aufgetreten
	1 = Device Nummer nicht gefunden
	2 = Register nicht vorhanden
	3 = Initialisierung fehlgeschlagen
	4 = Grabber nicht gefunden
	5 = unbekannter Parameterwert
	6 = nicht unterstützt
	7 = neuere Treiberversion wird benötigt (Update)
	8 = kein PHYTEC-Grabber gefunden
	9 = kein Acknowledge
	10 = ungültige Adresse
	11 = Schreiben nicht möglich

Jede Ausführung einer Treiberfunktion sollte auf ihren Erfolg überprüft werden. Dazu dient die Funktion **Get_Error**. Direkt nach der Ausführung einer Funktion wird die treiberinterne Fehlervariable auf den aktuellen Status gesetzt.

Diese Variable wird über **Get_Error** dem Anwenderprogramm zugänglich gemacht, das auf diese Weise auf auftretende Fehler reagieren kann.

Die Überprüfung der Fehlervariable ist so lange möglich, bis ein neuer Aufruf einer Treiberfunktion erfolgt. Dadurch wird der Fehlerstatus auf das Resultat des neuen Aufrufs gesetzt.

❶ Versionsnummer der DLL bestimmen

DWORD GetVersionNumber(void);

Rückgabewert: Versionsnummer der Grab4CDLL
HighWord: Major_Version_Number
LowWord: Minor_Version_Number

Über sie kann die Versionsnummer der geladenen Grabber-DLL ermittelt werden.

Hinweis:

Prüfen Sie die Versionsnummer und stellen Sie bei Anwendungen für den pciGrabber-4x4 sicher, dass die Major-Version-Number größer/gleich 4 ist.

❶ Anzahl der vorhandenen Decoder bestimmen

WORD Max_Device_Number();

Rückgabewert: Anzahl der gefundenen, kompatiblen Decoder

Diese Funktion bestimmt, wie viele kompatible Video-Decoder im System vorhanden sind.

- VD-012 vier Decoder
- VD-012-X1 vier Decoder
- VD-012-X2 zwei Decoder

PCI- und PCI Express-Geräte bekommen durch das PCI-BIOS automatisch einen Adressbereich zugewiesen. Über das PCI-BIOS kann dann dieser Adressbereich für diese bestimmte Karte ermittelt werden. Befinden sich mehrere gleiche Karten im System, so werden deren Adressen nacheinander geliefert (siehe *auch Kapitel 2.5, "Adressen und Ressourcen"*).

Der Anwender braucht sich bei Verwendung des Treibers nicht um Adressen und Adressbereiche zu kümmern. Diese werden treiberintern in *Gerätenummern* (*device number = nDevNo*) umgesetzt.

Jeder Decoder einer im System befindliche pciGrabber-4x4 - Karte erhält im Treiber eine eindeutige Gerätenummer zugeordnet. Welche Nummer einem bestimmten Decoder zugeordnet wird, ist nicht unbedingt vorhersagbar, da dies von der Topologie des PCI-Busses und der Funktion des BIOS abhängt.

Damit die verschiedenen Decoder unabhängig voneinander über den Treiber angesprochen werden können, wird die Gerätenummer als Parameter bei jedem Funktionsaufruf übergeben.

Die Funktion **Max_Device_Number()** wird eingesetzt, um herauszufinden, wieviele Decoder sich im Rechner befinden. Es wird die höchste zulässige Gerätenummer zurückgegeben. Dies ist gleichzeitig die Anzahl der Grabber im System, da die niedrigste Gerätenummer = 1 ist. Wird der Wert 0 zurückgeliefert, so hat das PCI-BIOS keinen pciGrabber erkannt.

Achtung!

Für nDevNo dürfen im folgenden nur Werte

$0 < \text{nDevNo} \leq \text{Max_Device_Number}()$

angegeben werden.

❶ Decoder und Treiber nach dem Einschalten initialisieren

void Initialize(WORD nDevNo);

Diese Routine initialisiert den Video-Decoder und die Treibersoftware nach dem Einschalten. Sie **muss** vor dem ersten Zugriff auf einen Decoder einmal aufgerufen werden. Also bei der Variante VD-012 vier Mal. Bei der Variante VD-012-X2 zweimal. Die Initialisierung muss auch für jeden weiteren Decoder, der benutzt werden soll, durchgeführt werden. Das bedeutet, dass **Initialize** mehrfach mit allen zulässigen Werten für nDevNo aufgerufen werden muss.

Wichtig!

Die Funktion Initialize() **muss** für **jedem Decoder** zu Beginn eines Programms aufgerufen werden.

Ansonsten arbeitet der jeweilige Decoder nicht korrekt!

❶ Informationen über die installierten Decoder abfragen

SHORT Read_GrabberInfo(WORD nDevNo, WORD wInfoType)

wInfoType: spezifiziert, welche Grabbereigenschaft abgefragt wird

Returnwert: Wert der spezifizierten Eigenschaft

Damit sich die Applikation optimal an die im Rechner installierte Grabber-Hardware anpassen kann, ist es mit dieser Funktion möglich, Informationen über die Hardware abzurufen.

So kann z.B. bestimmt werden, wie viele Eingangskanäle der jeweilige Decoder unterstützt und der Kanalauswahl-Dialog kann eine entsprechende Bereichsprüfung vornehmen.

Die Eigenschaften werden grundsätzlich als numerischer Wert (vom Typ WORD) zurückgeliefert. Die Bedeutung der Schlüsselnummern können Sie aus den Einträgen des Header-Files entnehmen. Das Header-File finden Sie nach der Installation des Demoprogramms im Unterverzeichnis „include“ des während der Installation ausgewählten Verzeichnisses. Der Name lautet Grab4DLL.H.

Mit dem Parameter wInfoType wird ausgewählt, welche Information abgefragt werden soll. Die Bedeutung des Parameters kann ebenfalls aus der Header-Datei ersehen werden. Ist ein Parameter nicht definiert, so gibt die Funktion den Wert -1 zurück. Der Fehlerstatus ist in diesem Fall 6 = „NOT_SUPPORTED“.

Folgende Parameter können beispielsweise abgefragt werden:

GRABBER_TYPE : Spezifiziert die Typennummer des Grabbers. Der Rückgabewert ist numerisch, mittels der Header-Datei kann die Typenbezeichnung bestimmt werden. Zum Beispiel: `Read_GrabberInfo(1, GRABBERTYPE) = 11` so ist Grabber Nr. 11 ein VD-012 und dessen Decoder 1.

MAX_CHANNEL : Gibt die Anzahl der Composite-Eingangskanäle zurück. Beispielsweise für VD-012 = 4.

Hinweis:

Beachten Sie, dass die Ergebnisse der Abfrage für jedem einzelnen *Decoder* gelten.

Um die gesamten Möglichkeiten der Karte zu ermitteln, müssen Sie zunächst mit *Max_Device_Number()* die Anzahl der Video-Decoder ermitteln und dann die Fähigkeiten jedes einzelnen Decoders mit *Read_GrabberInfo()* ermitteln.

Die abfragbaren Features können sich mit neueren Kartenversionen erweitern. Welche Informationen zur Verfügung stehen, ist in der Header-Datei ersichtlich.

❶ Decodernamen als Klartextstring lesen

**WORD Read_OrderCode (WORD nDevNo,
unsigned char* sCodeString,
DWORD dwSizeOfString)**

*sCodeString: Zeiger auf einen zuvor deklarierten String
(25 Bytes min.), in das die Funktion den
Grabbernamen einschreibt.

dwSizeOfString: Größe des reservierten Arrays

Returnwert: Errorcode

Mit dieser Funktion kann die Typenbezeichnung des jeweiligen Decoders im Klartext ausgelesen werden. Die Übergabe des Namens erfolgt in einem nullterminierten String. Dazu muss zuvor ein Character Array reserviert werden und ein Zeiger auf dieses Array in *sCodeString übergeben werden. Die verfügbare Größe des Arrays wird der Funktion im Parameter SizeOfString mitgeteilt. Sie sollte mindestens 25 Zeichen betragen.

Falls die Typenbezeichnung nicht in das reservierte Array passt, liefert der Rückgabewert der Funktion den Fehlercode 5 zurück.

Die Typenbezeichnung entspricht der Bestellnummer. Sollte dem Treiber bei einer Karte keine Klartext-Information zur Verfügung stehen, so wird der String „TYPE CODE=xx“ zurückgeliefert mit xx = codierter Typnummer (siehe *Read_GrabberInfo*)

Hinweis:

Um die einzelnen Decoder eines pciGrabber-4x4 zu unterscheiden, liefert die Funktion im Anschluß an die Typenbezeichnung die Decodernummer, z.B. in der Form:

VD-012-X1 Decoder 1

Falls der abgefragte Grabber ein pciGrabber-4 (VD-007) oder ein verwandtes Produkt ist, wird als Fehlercode 6 und als String „VD-007 or compatible“ geliefert

② Decoder auf verwendetes Farbsystem einstellen

```
void Set_Color_System(WORD nDevNo, WORD nColSys);
```

nColSys: Code für Farbsystem

Über die Funktion **Set_Color_System** wird das Farbsystem eingestellt, nach dem der jeweilige Decoder arbeiten soll. Taktfrequenz und Eingangsregister des Videoprozessors werden entsprechend gesetzt. Der Anwender kann für *nColSys* vordefinierte Konstanten einsetzen:

- PAL_BDGHI stellt den Grabber für die Verwendung an PAL-Videoquellen ein
- NTSC_M konfiguriert für NTSC-Quellen

↔ ② Erkennung des Videoformats

```
WORD Get_Video_Status(WORD nDevNo);
```

< Returnwert: 0 = 525 Zeilen Format (NTSC / PAL-M)
 1 = 625 Zeilen Format (PAL / SECAM)

Diese Funktion erkennt das Videoformat der Kamera am eingestellten Kanal und liefert zurück, ob es sich um eine Bildquelle nach NTSC-oder PAL/SECAM-System handelt.

Als Erkennungsmerkmal wird die unterschiedliche Zeilenzahl der Fernsehsysteme herangezogen. Es dauert nach Anlegen eines Bildsignals 32 aufeinanderfolgende Halbbilder, bis die Erkennung abgeschlossen ist.

② Einstellen des Composite-Modus (Composite-Eingänge)

void Set_Composite(WORD nDevNo);

Der Aufruf dieser Routine schaltet den Decoder in den Composite-Modus. Der Chroma-ADC wird ausgeschaltet und die Farbfalle wieder aktiviert. Dieser Modus muss eingestellt werden, wenn Composite-Signale digitalisiert werden sollen.

Der Composite-Modus muss für alle Standard-Kameras gewählt werden (schwarzweiß oder Farbe), die keinen *S-Video*-Ausgang haben, über den sie mit dem Grabber verbunden sind. Bei der Standard-Initialisierung wird der Composite-Modus eingestellt.

Hinweis:

Nach Aufruf von *Set_Composite* muss noch mit *Set_Channel* angegeben werden, von welchem Eingangskanal das Bild digitalisiert werden soll.

Wird an dieser Buchse kein Signal festgestellt, wird der Grabber auf die Mini-DIN-Buchse eingestellt und der Rückgabewert ist NO_SIGNAL.

Hinweis:

- Die AUTO-Funktion arbeitet nicht, wenn die angeschlossene Bildquelle kein Videosignal liefert.

② ☆ Einstellen des Eingangskanals

void Set_ChannelEx(WORD nDevNo, WORD nChannel);

nChannel: Einzustellender Eingangskanal (1...4)

Über diese Funktion kann der Eingangskanal eines Video-Decoders gewählt werden. Die Signalführung erfolgt über einen Multiplexer. Dieser ist in den Video-Prozessor integriert.

Wertebereich:

VD-012 : zulässig sind Kanalnummern 1 bis 4

VD-012-X1: zulässig sind Kanalnummern 1 bis 3

VD-012-X2: zulässig sind Kanalnummern 1 bis 3

Hinweis:

Beachten Sie, dass das Umschalten zwischen den primären Eingangsbuchsen (an der Front der Grabberkarte) nicht durch Umschalten des Eingangskanals erfolgt.

Die Buchsen sind auf unterschiedliche Decoder geführt und die Eingangssignale können gleichzeitig digitalisiert werden. Sie können Decoder individuell starten und stoppen und z.B. die Anzeige zwischen verschiedenen Bildern „umschalten“, indem Sie Bilder aus unterschiedlichen Speicherbereichen auf dem Bildschirm anzeigen.

VD-012-X2: Die Umschaltung zwischen der BNC- und der Mini-DIN-Buchse eines Decoders können Sie mit Hilfe der Funktionen *Set_Composite()* und *Set_S_VideoEx()* durchführen.

Das Einstellen des Eingangskanals ist bei S-Video-Quellen *nicht* erforderlich! Bei *S-Video* erfolgt die Kanalschaltung automatisch mit Aufruf der Funktion *Set_S_VideoEx()*.

Achtung!

Beachten Sie, dass bei der Umschaltung des Eingangskanals Wartezeiten einzuhalten sind, bis ein Bild des neuen Kanals digitalisiert werden kann. Dies hat mehrere Ursachen:

- Es dauert aufgrund der Definition des Videosignals bis zu 4 Halbbilder, bis die Synchronisation eingerastet ist und das Farbsystem korrekt arbeitet.
- Aufgrund der Gleichspannungsentkopplung in Grabber und Kamera entstehen unterschiedliche mittlere Gleichspannungspegel auf den Signalleitungen. Dadurch können bei der Umschaltung Umladungseffekte auftreten.
- Die AGC des Video-Decoders muss sich erst wieder auf den neuen Signalpegel einstellen.
- Es ist daher nicht möglich, von Bild zu Bild den Kanal umzuschalten. Sie sollten Wartezeiten von mindestens 80 ms einhalten. Dies ist jedoch mit abhängig von den angeschlossenen Signalquellen und kann nur als Richtwert dienen.

② Ein-/Ausschalten der Farbfalle für s/w-Betrieb

void Set_BW(WORD nDevNo, WORD nOn);

nOn: 0 = Composite-Signal am Eingang (Farbfalle einschalten),
1 = s/w-Signal am Eingang (Farbfalle ausschalten)

Wird statt einer Farb- eine Schwarzweißkamera an den Decoder angeschlossen, so ist die Farbfalle, die störendes Farbmoiré aus dem Helligkeitssignal entfernt (*Cross-Color-Effekt*), überflüssig.

Mit dieser Funktion kann die Farbfalle softwaremäßig ein- und ausgeschaltet werden. Das Ausschalten der Farbfalle bei s/w-Signalen ist sinnvoll, da durch den Wegfall des Filters die Bildschärfe erhöht wird. Defaultmäßig ist die Farbfalle eingeschaltet.

② Ein-/Ausschalten des Interlaced-Modus'

void Set_Interlace(WORD nDevNo, WORD nInterlace);

nInterlace: 0 = Non-Interlace
 1 = Interlace
 2 = Field Aligned

Mit dieser Funktion kann dem Decoder angezeigt werden, ob es sich bei dem ankommenden Videosignal um ein Interlaced oder ein Non-Interlaced-Signal handelt. Dies beeinflusst die Funktion des vertikalen Skalierungsfilters. Bei Vollbildauflösung beispielsweise sollte *Interlace* eingestellt werden, bei Verwendung eines einzelnen Fields für die vertikale Skalierung sollte *Non-Interlace* gewählt werden, um die Bewegungs-Artefakte zu reduzieren.

Möchte man nur Halbbilder darstellen, so entsteht normalerweise zwischen zwei Digitalisierungsvorgängen eine Pause von 20ms, da das andere Halbbild wegen des Halbbildversatzes nicht angezeigt werden kann. Im Modus *Field Aligned* wird das zweite Halbbild intern um eine halbe Zeile verschoben, so dass es räumlich auf das erste Halbbild zu liegen kommt. Dadurch kann die Halbbild-Wiedergabe im 20ms – Rhythmus erfolgen.

Da das Halbbild durch die elektronische Filterung verändert wird, ist diese Funktion für Mess- und Automatisierungsaufgaben nur bedingt geeignet.

② Ein-/Ausschalten der AGC

```
void Set_AGC(WORD nDevNo,WORD nCAGC, WORD nAGC,  
WORD nCrush);
```

nCAGC: 0 = Chroma AGC ausschalten
 1 = Chroma AGC einschalten

nAGC: 0 = AGC einschalten
 1 = AGC ausschalten

nCrush: 0 = Nicht-adaptive AGC
 1 = Adaptive AGC

Jeder Decoder des pciGrabber-4x4 verfügt über zwei AGC-Regelkreise. Die allgemeine AGC überwacht die Signalpegel des Video-Signals und regelt die Eingangs-Amplitude entsprechend nach. Zusätzlich sorgt die Chroma-AGC für eine Anpassung der Farbträgeramplitude.

Für die AGC ist zusätzlich der Modus *Adaptive AGC* einstellbar. Dabei wird zusätzlich das Overflow-Bit des A/D-Wandlers überwacht. Bei einem festgestellten Overflow wird automatisch die A/D-Referenzspannung erhöht, wodurch sich der Eingangsspannungsbereich der Grabbers weiter erhöht.

Im Allgemeinen ist es ausreichend, den Decoder mit nicht adaptiver Verstärkungsregelung zu betreiben.

In Anwendungen, bei denen sich das Regelverhalten der **adaptiven** AGC störend auswirken kann (das ist besonders dann der Fall, wenn mit absoluten Helligkeitswerten gearbeitet wird), sollte die nicht-adaptive AGC verwendet werden.

Bei Applikationen, die häufig zwischen mehreren Kameras umschalten (z.B. Video-Überwachung), kann die Umschaltzeit durch die Verwendung der adaptiven AGC unter Umständen verringert werden.

Achtung!

Die AGC-Funktion muss immer eingeschaltet sein (nAGC=0)!
Es handelt sich bei dieser Funktion nicht um eine automatische Helligkeitsregelung, sondern um eine Anpassung des Decoders an die Signalamplitude der Kamera.

② Farbtöter Ein-/Ausschalten

void Set_CKill(WORD nDevNo, WORD nCKill);

nCKill: 0 = Ausschalten des Farbtöters
 1 = Einschalten des Farbtöters

Werden an ein Farbsystem Schwarz/Weiß-Bildquellen angeschlossen, so kann es zu einem leichten Farbrauschen kommen. Der Farbtöter verhindert diesen Effekt, indem er die Anwesenheit des Color-Bursts prüft und gegebenenfalls die Farbauswertung abschaltet.

Unter Umständen möchte man ein Farbsignal digitalisieren, dessen Farbträger sehr schwach ist. In diesem Fall ist es möglich, dass die Erkennungsschwelle des Farbtöters bereits unterschritten ist und das Bild nur in Graustufen digitalisiert wird.

Über *nCKill* = 0 kann der Farbtöter explizit ausgeschaltet werden, so dass das Bild dann farbig - möglicherweise verrauscht - digitalisiert wird.

Defaultmäßig ist der Farbtöter eingeschaltet, die Umschaltung zwischen Farb- und Schwarz/Weiß-Bildquellen erfolgt also automatisch.

② Auslassen von Fields, Frames aus dem Videosignal

```
void TemporalDect(WORD nDevNo,
                  WORD nDecField,
                  WORD nFldAlign,
                  WORD nDecRat);
```

nDecField: 0 = Frame(s) auslassen
 1 = Field(s) auslassen
 nAlign: 0 = odd Field wird als erstes ausgelassen
 1 = even Field wird als erstes ausgelassen
 nDecRat: Anzahl auszulassender Fields / Frames

Der pciGrabber-4x4 ermöglicht es, bei kontinuierlicher Digitalisierung, die Anzahl der pro Sekunde digitalisierten Bilder einzustellen. Defaultmäßig werden 50 bzw. 60 (NTSC) Bilder pro Sekunde digitalisiert (Video-Norm).

Mit Hilfe dieser Funktion kann bestimmt werden, wie viele Bilder von diesen 50 bzw. 60 Bildern bei der Digitalisierung *ausgelassen* werden sollen.

Über die beiden anderen Parameter wird die Art des Auslassungsvorgangs gesteuert:

nAlign gibt an, ob die Auslassung mit geraden oder ungeraden Halbbildern beginnen soll.

nDecField bestimmt, ob sich die Auslassung auf Voll- oder Halbbilder beziehen soll.

Beispiele (PAL):

- *nDecField=0, nDecRate=2*

Die Reduzierung wird auf Vollbild-Ebene durchgeführt. Pro 50 Bilder werden zwei Bilder ausgelassen. Die Bilder 1-24 werden normal ausgegeben, dann wird ein Bild ausgelassen. Die Bilder 26 bis 49 werden ausgegeben, gefolgt von einem ausgelassenen Bild.

- *nDecField=1, nDecRate=25*

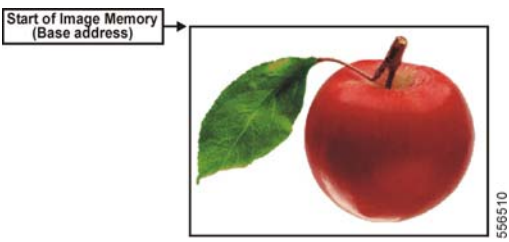
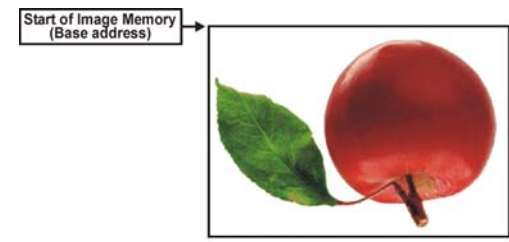
Die Reduzierung wird auf Halbbild-Ebene durchgeführt. Fünfundzwanzig Halbbilder werden pro 50 Halbbilder ausgelassen. Dies resultiert darin, dass jedes zweite Halbbild ausgegeben wird. Es werden also immer gleiche Halbbilder ausgelassen. Welches Halbbild zuerst ausgelassen wird, hängt von *nAlign* ab.

🗺️⭐️🔴 Bildorientierung einstellen

DWORD FlipPicture(WORD nDevNo, unsigned char flip)

nDevNo : System-Nummer des Grabbers
flip : 0 = Bild aufrecht
 : 1 = Bild vertikal gespiegelt (default)

Mit dieser Funktion wird festgelegt, wie das vom Decoder digitalisierte Bild im Speicher abgelegt wird:

flip = 0	flip=1 (default)
<p>Das Bild wird aufrecht dargestellt, an der niederwertigsten Bildspeicheradresse befindet sich die linke <u>obere</u> Ecke des Bilds:</p> 	<p>Das Bild wird <u>vertikal</u> gespiegelt geschrieben. An der niederwertigsten Bildspeicheradresse befindet sich die linke <u>untere</u> Ecke des Bilds:</p>  <p>Diese Variante ist z.B. nützlich, wenn Bilder im BMP-Format verarbeitet werden.</p>

Wichtig:

- Die Einstellung der Bildorientierung mit *FlipPicture()* muss vor Aufruf der *Set_Image()* – Funktion erfolgen. Die Einstellung wirkt sich erst nach Aufruf der *Set_Image()* – Funktion aus.
Wird die Einstellung im laufenden Betrieb geändert, so muss der Decoder angehalten werden, über *FlipPicture()* der gewünschte Wert gesetzt und dann *SetImage()* erneut aufgerufen werden.
- Die Default-Einstellung ist **flip=1**, das Bild wird also vertikal gespiegelt eingeschrieben (aus Kompatibilitätsgründen)

② Bildgröße und Skalierung einstellen

**void Set_Image (WORD nDevNo,
WORD nOhpos, WORD nOvpos,
WORD nOhsize, WORD nOvsize,
WORD nOppl, WORD nOlines,
WORD nOColformat,
WORD nEhpos, WORD nEvpos,
WORD nEhsize, WORD nEvsize,
WORD nEppl, WORD nElines,
WORD nEColformat,
WORD nColSystem,
WORD nInterlaced,
WORD nSingleShot);**

nOhpos, nOvpos : Position der linken oberen Ecke des Odd-Bildausschnitts im Videobild
(hpos = horizontal, vpos = vertikal)

nOhsize : Größe des Odd-Bildausschnitts in X-Richtung
nOvsize : Größe des Odd-Bildausschnitts in Y-Richtung
nOppl : gewünschte Größe des Odd-Videobilds in X-Richtung
(ppl = Pixel Per Line)

nOlines : gewünschte Zeilenzahl des Odd-Videobilds
nOColformat: gewünschtes Farbformat: (RGB32, RGB24, RGB16, RGB15, Y8, YCrCb 4:2:2, YCrCb 4:1:1)

nEhpos, nEvpos : Position der linken oberen Ecke des Even-Bildausschnitts im Videobild
(hpos = horizontal, vpos = vertikal)

nEhsize : Größe des Even-Bildausschnitts in X-Richtung
nEvsize : Größe des Even-Bildausschnitts in Y-Richtung
nEppl : gewünschte Größe des Even-Videobilds in X-Richtung
(Pixel Per Line)

nElines : gewünschte Zeilenzahl des Even-Videobilds
nEColformat: gewünschtes Farbformat: (RGB32, RGB24, RGB16, RGB15, Y8, YCrCb 4:2:2, YCrCb 4:1:1)

nColSystem : Code für Farbsystem (vgl. *Set_Color_System*)
nInterlace: 0 = Non-Interlace
1 = Interlace
2 = Field Aligned

nSingleShot : 0 = kontinuierliche Digitalisierung
1 = ein einzelnes Bild grabben

Mit der Routine **Set_Image ()** werden Größe, Position und Skalierung des vom Decoder gelieferten Bildausschnitts für Even- und Odd-Bilder getrennt festgelegt. Des Weiteren wird das Datenformat, in dem das Bild im Speicher abgelegt wird, definiert.

Achtung!

Der Aufruf der Funktion darf nur erfolgen, wenn sich der Decoder im Stop-Modus befindet. *Set_Image* darf **nicht** aufgerufen werden, wenn der Decoder digitalisiert oder eine Single-Shot – Digitalisierung noch nicht mit *Stop_Grabber* abgeschlossen wurde.

Die Einstellungen können grundsätzlich für beide Halbbilder getrennt vorgenommen werden. Der Parametersatz ist entsprechend gegliedert in Parameter mit vorgestelltem 'E' = Parameter für Even-Bild und vorgestelltem 'O' = Odd-Bild - Parameter. Parameter ohne diese Kennzeichnung gelten für beide Halbbilder.

Für die beiden Halbbilder können völlig verschiedene Bildgrößen angegeben werden. Sie werden im Bildspeicher hintereinander abgelegt und können getrennt voneinander bearbeitet werden.

Im *Interlaced-Modus* werden die beiden Halbbilder richtig verzahnt in einen gemeinsamen Speicherbereich geschrieben, wodurch sich die volle Auflösung von bis zu 720 x 576 Bildpunkten (PAL) bzw. 640 x 480 Bildpunkten (NTSC) ergibt.

Im Folgenden wird die Einstellung der Werte ohne Voranstellung der Halbbild-Spezifikation diskutiert (z.B. *hsize* für *nEhsize* / *nOhsize*). Es ist dann jeweils 'E' für 'Even' bzw. 'O' für 'Odd' voranzustellen. Ist die Festlegung auf einen bestimmten Halbbildtyp erforderlich, so wird der Parameter voll ausgeschrieben.

Bei der Einstellung der Parameter legt man zuerst ein Fenster mit der Größe des gewünschten Bilds (in Pixeln) fest. Dies geschieht mittels der Werte von *hsize* und *vsze*. *Hsize* gibt also die Anzahl der Pixel in x-Richtung an, die das aufgenommene Bild später hat, *vsze* die Anzahl der Pixel in y-Richtung.

Die Werte *ppl* und *lines* geben hingegen an, wieviel Pixel aus dem ankommenden Videobild generiert werden sollen. *ppl* definiert die Anzahl der Pixel pro Bildzeile, *lines* gibt an, wie viele Zeilen („Pixel in y-Richtung“) erzeugt werden.

Mit diesen beiden Werten kann also die Auflösung des Bilds festgelegt werden. Ein Vollbild im PAL-Format hat 720 Pixel x 576 Zeilen. Um es in höchstmöglicher Auflösung zu digitalisieren, wird *ppl* der Wert 720 zugewiesen und *lines* = 576 gesetzt. Die kleinsten Auflösungen liegen für PAL bei 50 x 40 Pixels pro Vollbild.

Werden einzelne Halbbilder benutzt, so liegt die maximale Auflösung bei jeweils 720 x 288 Zeilen.

Aufgrund der Tatsache, dass sich die Definition von *ppl* und *lines* stets auf *Halbbilder* bezieht, die tatsächliche Zeilenzahl des zu digitalisierenden TV-Bilds also doppelt so hoch ist, ist das Breiten-/Höhenverhältnis bei den digitalisierten *Halbbildern* (maximal dann 720 x 288 Pixel) um den Faktor zwei verzerrt. Um dies zu beheben, digitalisiert man entweder im Interlaced-Modus (wenn die hohe Auflösung erforderlich ist) oder man reduziert die Auflösung auf *ppl*=360, *lines*=288 und arbeitet mit der proportionsgerechten Auflösung 360 x 288.

Bei Vermessungs- und Automatisierungsanwendungen ist oft nicht unbedingt eine proportionsgerechte Darstellung nötig, sofern die Verzerrung im Algorithmus berücksichtigt wird oder nicht ins Gewicht fällt. So kann man z.B. die volle Halbbildauflösung 720 x 288 bei Vermessungen benutzen, die in X-Richtung genauer als in Y-Richtung arbeiten. Gegebenenfalls kann man die Kamera entsprechend der benötigten Vermessungsachse ausrichten.

Das mit *hsize* und *vsize* aufgespannte Fenster ist nun der Ausschnitt des Bildes, den man von dem digitalisierten Bild der Größe *ppl* x *lines* sieht. Ist *hsize* = *ppl* und *vsize* = *lines*, so sieht man das ganze digitalisierte Bild; sind sie kleiner, nur einen entsprechenden Ausschnitt. Das Verhältnis von *hsize* und *vsize* verändert nicht die Proportionen des Bildes, da hier keine Skalierung, sondern nur eine Ausschnittsbildung aus dem bereits skalierten Bild vorgenommen wird. *Bild 50* und *Bild 51* illustrieren die Bedeutung der Parameter.

Die Parameter *vsize* und *lines* sind in Bezug auf ein Halbbild anzugeben, wenn ein Halbbild digitalisiert und abgeholt wird.

Ist der mit *hsize* und *vsize* definierte Ausschnitt kleiner als die über *ppl* und *lines* festgelegte Bildgröße, so kann das Fenster mit den Werten von *hpos* und *vpos* in dem digitalisierten Bild verschoben werden. Bei *hpos*=0 und *vpos*=0 liegt es in der linken oberen Ecke des digitalisierten Bilds.

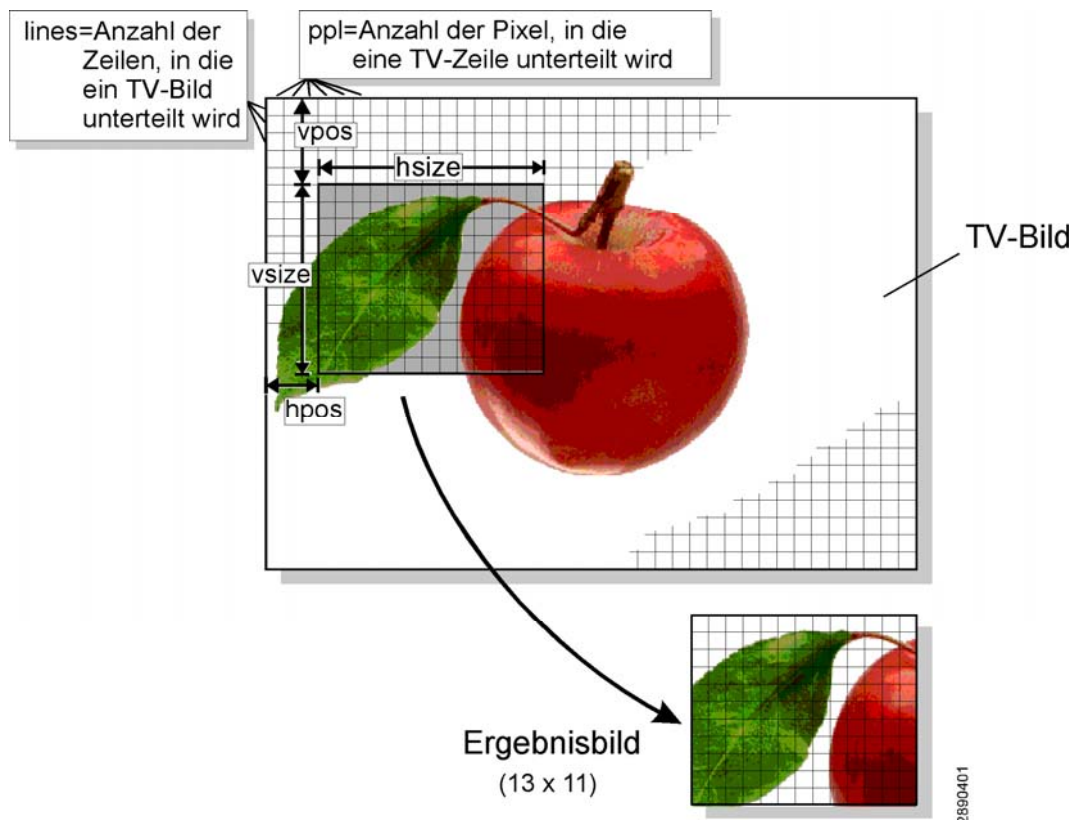


Bild 50: Skalierung und Ausschnittsbildung

Achtung!

- Der über *hsize* und *vsize* in der Größe und *hpos* und *vpos* in der Position definierte Fensterbereich darf an keiner Stelle den durch *ppl* und *lines* definierten Bereich des digitalisierten Bilds verlassen:

- hpos*=100, *hsize*=200, *ppl*=200 : nicht zulässig, da die letzten 100 Pixel nicht definiert sind
- hpos*=0, *hsize*=200, *ppl*=202 : zulässig; alle Pixel liegen im Bildbereich
- hpos*=100, *hsize*=100, *ppl*=200 : zulässig
- hpos*=100, *hsize*=200, *ppl*=300 : zulässig
- hpos*=300, *hsize*=300, *ppl*=800 : nicht zulässig: Bildbereich hat mehr Pixel als die TV-Norm liefert

(entsprechend in Y-Richtung)

- Alle Parameter in horizontaler Richtung müssen *gerade* Werte haben!
(*ppl*=123 ist nicht erlaubt, *ppl*=124 ist zulässig.)
- Werden die Parameter, die die Bildgröße beeinflussen, auf 0 gesetzt, so wird das entsprechende Halbbild nicht erzeugt.

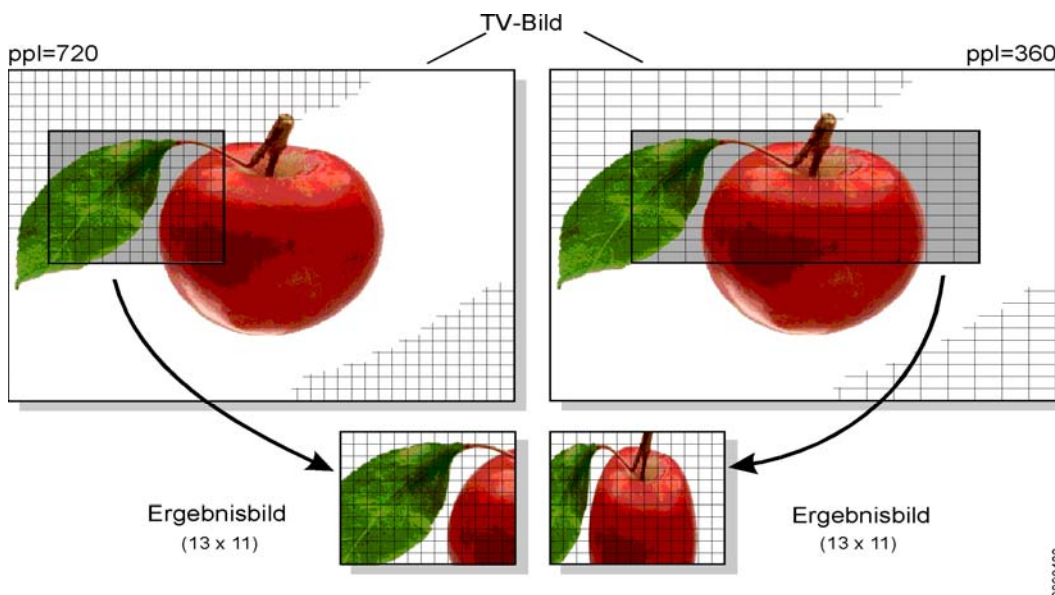


Bild 51: Bild zur Skalierung: alle Werte gleich bis auf ppl

Beispiele:• **Halbbild-Digitalisierung**

Es soll ein quadratisches Bild der Größe 256 x 256 Pixel digitalisiert werden, dessen Auflösung und Proportionen dem TV-Bild entsprechen.

(a) Auflösung und Skalierung

Zur Digitalisierung genügt ein Halbbild mit 288 Zeilen. Damit das Höhen-/Breitenverhältnis stimmt, muss die Auflösung in X-Richtung ebenfalls auf die Hälfte verringert werden (denn: Halbbild = halbe Höhe).

Also $720:2 = 360$.

Es ergeben sich $ppl = 360$, $lines = 288$

(b) Ausschnittsgröße

Das Bild soll quadratisch 256 x 256 Pixel groß sein. Daraus ergibt sich direkt:

$hsize = 256$, $vsize = 256$

(c) Positionierung

Es ist sinnvoll, den Bildausschnitt zu zentrieren.

In X-Richtung werden von 360 Pixeln nur 256 im Fenster dargestellt. Es entsteht ein Rand von $360-256=104$ Pixeln, der gleichmäßig auf beide Seiten verteilt werden soll, also jeweils 52 Pixel links und rechts. $hpos$ ist die Größe des linken Randes, also $hpos = 52$.

Entsprechend ist in Y-Richtung: $(288-256):2=16$; $vpos = 16$.

Hinweis:

Es wäre hier falsch, $ppl = 256$ und $lines = 256$ zu setzen. Dadurch würde das Breiten-/Höhenverhältnis verändert (TV-Norm = 4:3) zu 1:1 und das Bild verzerrt. Es wäre jedoch möglich, $lines = 256$ zu setzen und ppl über das Verhältnis von Bildbreite und -höhe zu berechnen. So wäre die Bildhöhe optimal ausgenutzt.

- **Halbbild-Digitalisierung mit Zoom**

Es soll ein Bild der Größe 120 x 100 geliefert werden, in dem das Originalbild um den Faktor 2 in X- und Y-Richtung vergrößert dargestellt wird.

(a) Auflösung / Skalierung

Ausreichend ist wieder ein Halbbild. Ohne Vergrößerung wären 360 x 288 Pixel anzufordern (s.o.). Um die Dehnung zu erzeugen, werden nun 180 x 144 Pixel definiert: $ppl = 180$, $lines = 144$. Das Breiten-/Höhenverhältnis wird gewahrt.

(b) Ausschnittsgröße

Entsprechend der Fenstergröße wird gesetzt:
 $hsize = 120$, $vsize = 100$.

(c) Positionierung

Mit $hpos$ und $vpos$ kann der Fensterausschnitt nach Bedarf maximal $180-120 = 60$ Pixel in X-Richtung und $144-100 = 44$ Pixel in Y-Richtung verschoben werden.

- **Vollbild-Digitalisierung**

Es soll ein Bild der Größe 700 x 500 geliefert werden, das in Auflösung und Proportionen dem TV-Bild entspricht.

(a) Auflösung und Skalierung

Zur Digitalisierung wird hier ein Vollbild benötigt. Da also als Grundlage ein volles TV-Bild dienen soll, beträgt die zugrundeliegende Bildauflösung $ppl = 720$ und $lines = 576$. In vertikaler Richtung muss der lines-Wert nun auf beide Halbbilder verteilt werden:

$$nOlines = nElines = \frac{1}{2} lines$$

$$nOlines = \frac{1}{2} 576 = 288$$

$$nElines = \frac{1}{2} 576 = 288$$

Da das erzeugte Vollbild insgesamt 576 Zeilen hat, werden in X-Richtung 720 Pixel benötigt, damit das Höhen-/Breitenverhältnis stimmt:

$$nOppl = nEppl = 720$$

(b) Ausschnittsgröße

Das Bild soll 700 x 500 Pixel groß sein. Daraus ergibt sich direkt: $hsize=700$.

Für die vertikale Ausschnittsgröße muss wieder die Aufteilung in zwei Halbbilder erfolgen:

$$nOvsize = nEvsize = \frac{1}{2} 500 \text{ Pixel} = 250$$

(c) Positionierung

Es ist sinnvoll, den Bildausschnitt zu zentrieren.

In X-Richtung werden von 720 Pixeln nur 700 im Fenster dargestellt. Es entsteht ein Rand von $720-700 = 10$ Pixeln, der gleichmäßig auf beide Seiten verteilt werden soll, also jeweils 5 Pixel links und rechts. $hpos$ ist die Größe des linken Randes, also $hpos=5$:

$$nOhpos = nEhpos = 4 \text{ (gerader Wert)}$$

Entsprechend ist in Y-Richtung: $(288-250):2 = 19$;

$$nOvpos = nEvpos = 18 \text{ (gerader Wert)}.$$

Der Parameter $nInterlaced$ sollte auf 1 gesetzt werden, damit die Bilder automatisch verzahnt werden.

Nachdem nun Größe und Auflösung des Bilds definiert sind, muss im nächsten Schritt das Datenformat ausgewählt werden. Der Parameter Colformat beschreibt, in welcher Form ein Pixel im Arbeitsspeicher der CPU abgelegt wird und damit auch, wie viel Bytes von einem Pixel belegt werden.

Das Datenformat ist in erster Linie von der Anwendung bestimmt. Grundsätzlich sind drei Formate zu unterscheiden, die teilweise nochmals in Unterformate unterteilt sind. *Bild 52* zeigt, wie die Pixel in den einzelnen Formaten im Speicher abgelegt werden.

- **RGB:** Die Helligkeitsinformation wird für die drei Farbkanäle Rot, Grün und Blau getrennt abgelegt. Dies ist die Form, in der allgemein Farbinformation betrachtet und verarbeitet wird.
 - Bei *RGB32* werden 32-bit, also ein Doppelwort pro Pixel benutzt. Das unterste Byte eines jeden Doppelworts enthält die Blau-Information (8-bit breit), das zweite Grün- und das dritte die Rotwerte. Das oberste Byte enthält keine Information. Es wird lediglich eingefügt, damit die Zuordnung von einem Pixel zu einem Doppelwort erreicht wird: Mit jeder DWort-Grenze beginnt auch ein neues Pixel (zusammengehörige Information ist in *Bild 52* in gleicher Weise schraffiert). Das Alignment auf DWords hat gegebenenfalls den Vorteil, dass schnellere Zugriffsbefehle benutzt werden können. Die Farbtiefe beträgt 16 Mio. Farben ($2^{38} = 16777 \cdot 10^6$).
 - *RGB24* überträgt den gleichen Informationsgehalt wie *RGB32*, es entfällt jedoch das Füllbyte. Das Bild wird bei gleicher Farbauflösung kompakter im Speicher dargestellt.
 - *RGB16* hat eine geringere Farbauflösung. In diesem Format werden jeweils fünf Bit für Blau und Rot-Kanal verwendet, der Grün-Kanal verfügt über sechs Bit. Die Farbtiefe beträgt damit 65.536 Farben ($2^5 \cdot 2^5 \cdot 2^6 = 65536$). Der Speicherplatzbedarf eines Pixels beträgt dementsprechend 16-bit = 1 Wort. In *Bild 52* ist die Aufteilung des Worts in die drei Farbkanäle gezeigt.

Die Farbinformation wird dabei „linksbündig“ ausgerichtet; die niederwertigen Bits fallen also weg, wodurch die Farbtiefe reduziert wird.

Die Farbtiefe des Grün-Kanals ist doppelt so hoch wie die der beiden anderen Kanäle. Das RGB16-Format entspricht dem Farbsystem von Grafikkarten mit 65.535 Farbstufen.

- **RGB15:** Die Aufteilung entspricht dem RGB16-System, jedoch ist die Farbtiefe für alle drei Farbkanäle gleich (je 5-bit = 32 Stufen). Daraus ergeben sich 32.818 darstellbare Farben. Insgesamt werden nur 15-bit benötigt, wodurch das oberste Bit eines jeden Wortes unbenutzt bleibt. Es enthält den Wert 0 (siehe *Bild 52*).
- **YCrCb:** In diesem Format liegen Grauwert und Farbinformation getrennt vor. Der Parameter Y beschreibt die Helligkeit des Pixels (Grauwert), der Wertetupel (Cr,Cb) enthält die Farbinformation. (Cr,Cb) ist als Vektor im Farbkreis zu verstehen. Der Farbton entspricht dem Winkel des Zeigers, die Sättigung wird durch den Betrag dargestellt. Anwendung findet dieses Format z.B. beim Weiterleiten der Daten an eine für das YCrCb-System ausgelegte Grafikkarte, beim getrennten Verarbeiten von Helligkeit und Farbwert und - aufgrund seiner Kompaktheit - beim (Zwischen-)Speichern von Bildinformation.
 - **YUY2:** Dieses Format entspricht dem Format YCrCb 4:2:2. In einem DWord befindet sich die Information zweier Pixel. Y0 und Y1 ist die Helligkeitsinformation der benachbarten Pixel, Cb0 und Cr0 die Farbinformation des ersten Pixels, die für beide Pixel genutzt wird. Die Farbinformation des zweiten Pixels wird nicht genutzt.
 - **BtYUV :** Entspricht YCrCb 4:1:1. Vier Pixel teilen sich eine Farbinformation. Die Anordnung der Information im Speicher ist aus *Bild 52* zu ersehen. Jeweils drei DWords sind logisch zusammengefasst. Sie enthalten die Information für acht Pixel. Dies ist die Helligkeit für jedes Pixel (Y0..Y7) und die Farbinformation des ersten (Cb0/Cr0) und fünften Pixels (Cb4/Cr4).

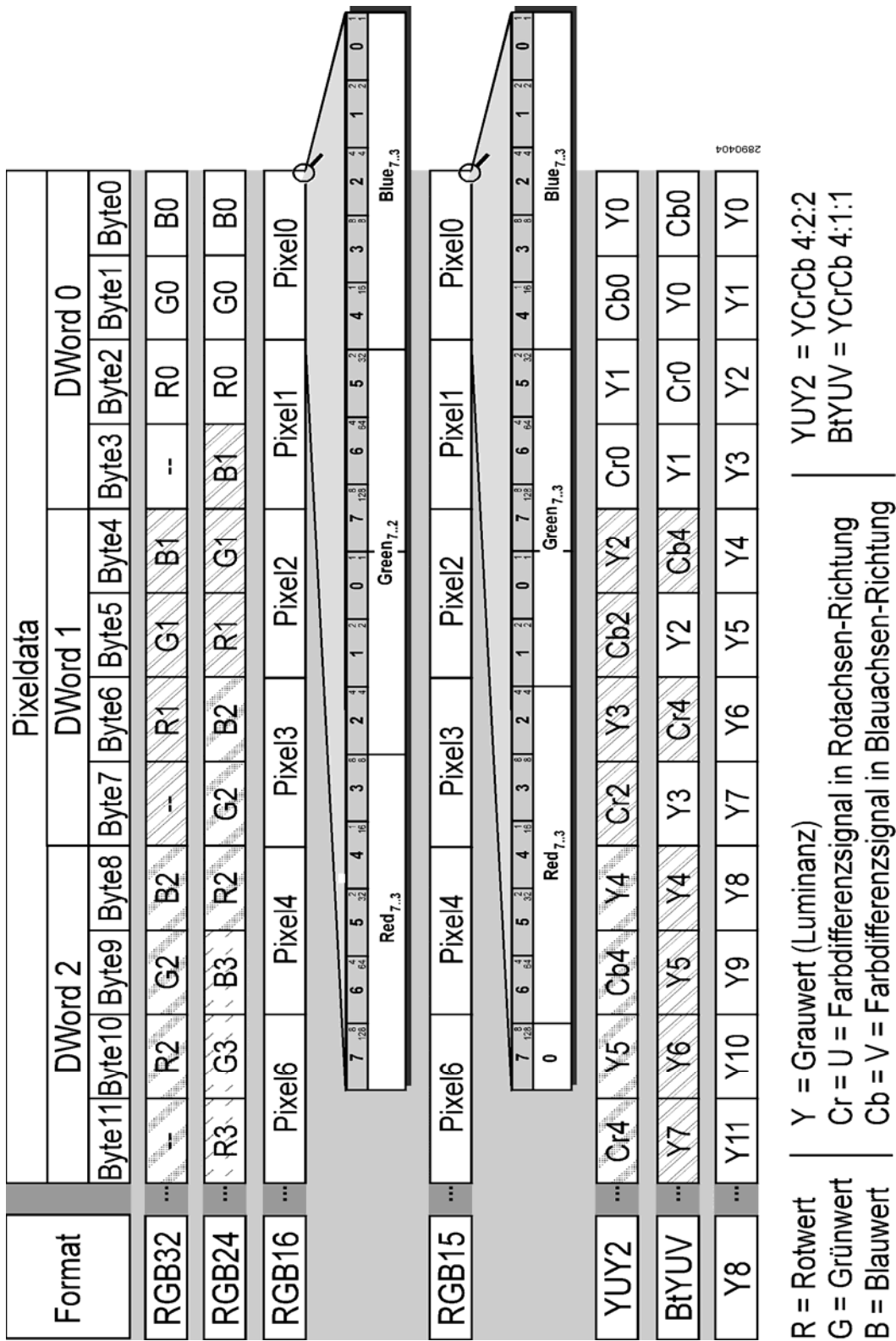


Bild 52: Farbformate des pciGrabber-4x4

- **Y:** Im Grayscale-Format wird nur der Grauwert, also die Helligkeitsinformation eines Pixels abgespeichert. Die Farbinformation wird nicht berücksichtigt. Dieses Format empfiehlt sich, wenn bei der Auswertung Farbe keine Rolle spielt.
 - **Y8:** Im Y8-Format ist die Grauwertinformation jedes Pixels als 8-bit-Wert hintereinander im Speicher abgelegt. Ein Byte entspricht also einem Pixel.

Nun ist das Format des zu digitalisierenden Bildes genau definiert. Der Decoder muss jetzt angewiesen werden, wo und in welcher Weise die erzeugten Bilddaten im Arbeitsspeicher abgelegt werden. Dazu ist es zunächst erforderlich, einen ausreichend großen Arbeitsspeicherbereich zu reservieren.

Dies geschieht mit den bekannten Befehlen zur Allokation von Speicher in der jeweiligen Programmiersprache (z.B. *malloc(...)*).

Der Windows-Treiber reserviert selbständig einen entsprechenden Bildspeicher, in dem das Bild abgelegt wird.

Wie viel Speicherplatz wird benötigt? Dies errechnet sich aus der Größe des Bilds (Anzahl der Pixel) und der pro Pixel benötigten Byte (Farbauflösung):

$$\text{Speicherbedarf pro Halbbild} = \text{hsize} \cdot \text{vsize} \cdot \text{pixelgröße} \text{ [Byte]}$$

Der Wert *pixelgröße* bestimmt sich wie in *Tabelle 10* gezeigt.

Bei den Formaten YUV2 und BtYUV muss beachtet werden, dass 2 bzw. 8 Pixel logisch zusammengehören und die Bildauflösung entsprechend passend gewählt wird. Bei der Berechnung ist zu beachten, dass die Pixelzahl *hsize* · *vsize* für jedes Halbbild einzeln zu berechnen und gegebenenfalls bei Vollbildern zu addieren ist.

Format	<i>pixelgröße</i> [Byte]
RGB32	4
RGB24	3
RGB16, RGB15	2
YUY2	4 Byte pro 2 Pixel
BtYUV	12 Byte für 8 Pixel
Y8	1

Tabelle 10: Speicherbedarf eines Pixel in den einzelnen Modi

Werden Vollbilder benötigt, weil die vertikale Auflösung mehr als 288 Zeilen betragen soll, so kann der Decoder angewiesen werden, dieses Vollbild in einem Speicherbereich komplett aufzubauen. Er nimmt dann die erforderliche Verzahnung der beiden Halbbilder automatisch vor. Wird diese Option gewünscht, so setzt man $nInterlaced = 1$.

Abschließend wird angegeben, in welcher Weise die Bildaufnahme vorgenommen werden soll: Mit $nSingleShot = 0$ wird der Decoder angewiesen, kontinuierlich zu digitalisieren. Das bedeutet, dass ab dem Moment, in dem der Decoder den Befehl zum Start erhält, andauernd in Echtzeit (50 Halbbilder pro Sek.) die angegebenen Speicherbereiche mit aktuellen Bilddaten beschrieben werden. Im Halbbild-Modus ($nInterlaced = 0$) wird ein Speicherbereich beschrieben

(20 ms), dann der andere (wieder 20 ms) und weiter wechselweise. Dies bedeutet, dass jeder Halbbildspeicher mindestens 20 ms lang nicht vom Decoder belegt wird.

Im Vollbildmodus ($nInterlaced = 1$) beschreibt der Decoder den gemeinsamen Bildspeicher ständig, 20 ms die ungeraden, dann 20 ms die geraden Zeilen.

Bei der Auswertung des Bildinhalts kann es störend sein, wenn man in dem Bildfeld arbeitet, das der Decoder gerade aktualisiert, da ein Versatz von bewegten Objekten ab einer zufälligen Bildzeile auftreten kann. In diesem Fall ist es ratsam, im Start/Stop-Betrieb zu arbeiten oder die Halbbilder in zwei getrennte Bildspeicher zu digitalisieren, die man entsprechend abwechselnd bearbeitet.

$nSingleShot = 1$ bewirkt, dass nur genau ein Digitalisierungsvorgang durchgeführt wird. Es werden zwei Halbbilder (ein Odd- und ein Even-Bild) aufgenommen, bzw. ein Vollbild.

Der Anwender kann die Bilder aufnehmen und dann per erneuten Start-Befehl die Bildspeicher aktualisieren. Diese Betriebsart ist empfehlenswert, wenn nur gelegentlich Bilder digitalisiert werden müssen und es nicht auf Echtzeit-Darstellung ankommt.

Egal, ob kontinuierlich gegrabbt wird oder nur Einzelbilder geschossen werden: **Set_Image()** stellt nur die Parameter ein, legt also fest, *wie* die Bilder aufgenommen werden sollen. Der Grabbvorgang selbst wird dadurch noch nicht gestartet. Dies geschieht durch den Befehl **Start_Grabber()** (siehe *unten*).

③ Bildaufnahme starten

void Start_Grabber(WORD nDevNo);

Die Funktion **Start_Grabber()** startet den Grabvorgang für den über $nDevNo$ spezifizierten Grabber/Decoder. Dies bedeutet, dass der Digitalisierungsvorgang mit dem Beginn des nächsten verfügbaren Bildes beginnt. Ist kontinuierliches Grabben gewählt worden, so wird fortgesetzt digitalisiert, bis **Stop_Grabber()** aufgerufen wird. Bei Einzelbild-Betrieb wird die Digitalisierung nach dem ersten vollständigen Bild beendet.

Wann wird die erste Digitalisierung durchgeführt?

Der Treiber arbeitet auf Vollbild-Basis. Es werden immer zusammenhängende Even/Odd-Bildkombinationen betrachtet. Das führt zu einem Zeitverhalten, das abhängig ist vom gewünschten Halbbild und dem zum Startzeitpunkt gerade aktuell am Videoeingang anliegenden Halbbild. Folgende Fälle müssen unterschieden werden:

(1) Es soll ein Even-Bild digitalisiert werden

a) Am Eingang liegt gerade ein Even-Bild an

⇒ Da das laufende Bild nicht mehr ausgewertet werden kann (der Beginn fehlt), läuft der Rest des Even-Bilds und das folgende Odd-Bild durch, danach beginnt die Digitalisierung. Es wird also das nächste vollständige Even-Bild digitalisiert. Die Zeitverzögerung Startbefehl bis Digitalisierungsbeginn beträgt < 40 ms.

b) Am Eingang liegt gerade ein Odd-Bild an

⇒ Das auf das Odd-Bild folgende Even-Bild wird digitalisiert. Maximale Verzögerung: 20 ms.

(2) Es soll ein Odd-Bild digitalisiert werden

a) Am Eingang liegt gerade ein Even-Bild an

⇒ Da der Treiber zuerst Even-Bilder verarbeitet, vergeht zunächst der Rest des unvollständigen Even-Bilds und das folgende Odd-Bild, bevor der Grabber auf dem nächsten Even-Bild einrastet. Das auf dieses Even-Bild folgende Odd-Bild wird digitalisiert. Die maximale Verzögerung beträgt < 60 ms.

b) Am Eingang liegt gerade ein Odd-Bild an

⇒ Es vergeht das unvollständige Odd-Bild, danach rastet der Grabber im Even-Bild ein und digitalisiert das folgende Odd-Bild. Verzögerung: < 40 ms.

③ Bildaufnahme stoppen

```
void Stop_Grabber(WORD nDevNo);
```

Die Funktion **Stop_Grabber** stoppt den Grabbvorgang. Der Bildeinzug wird sofort abgebrochen, das aktuell digitalisierte Bild ist eventuell unvollständig (siehe *Bild 49*).

Achtung!

Stop_Grabber muss auch dann aufgerufen werden, wenn die Digitalisierung bei Single-Shot-Betrieb (*nSingleShot* = 1 in *Set_Image*) automatisch beendet wird. Der Decoder ist so lange verriegelt (in Wartestellung), bis **Stop_Grabber** aufgerufen wird. Danach kann ein neues Einzelbild mit **Start_Grabber** angefordert werden.

③ Digitalisierungsfortschritt erkennen

WORD Data_Present(WORD nDevNo);

Returnwert: zeigt den Digitalisierungsfortschritt an
(kann Werte zwischen 0 und 15 (4-bit) annehmen)

Die Funktion **Data_Present** signalisiert, ob ein Even oder Odd Bild digitalisiert und im Speicher abgelegt wurde.

In den einzelnen Bits des Rückgabewerts ist codiert, wie weit die Digitalisierung bei Einzelbild- oder kontinuierlichem Betrieb fortgeschritten ist (siehe *Bild 53*).

Bit 0 und 2 zeigen an, dass ein Even-Bild eingelesen wurde, Bit 1 und 3 ein Odd-Bild. Die Bits 0 und 1 wechseln bei jedem Einlaufen eines Even- bzw. Odd-Bildes ihren Wert.

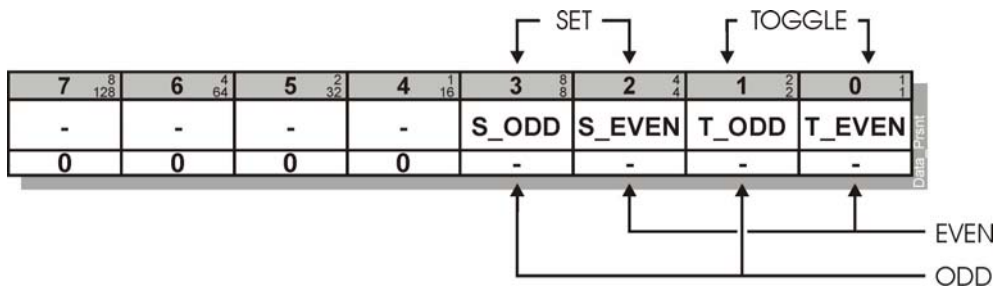


Bild 53: Rückgabewert der Funktion data_present

Bei kontinuierlichem Grabben kann damit bestimmt werden, wann der Inhalt des entsprechenden Bildspeichers mit einem neuen Bild vollständig ersetzt wurde. Jeder *Wechsel* des Zustands (0 auf 1 und 1 auf 0) zeigt an, dass ein neues Halbbild eingeschrieben wurde.

Die Bits 2 und 3 werden auf 1 gesetzt, sobald ein Even- bzw. Odd-Bild vollständig eingelesen wurde. Diese Bits werden verwendet, wenn nur ein Bild eingelesen werden soll (Single-Shot, durch **Set_Image** festgelegt). Sie bleiben dann so lange gesetzt, bis ein neuer Digitalisierungsvorgang eingeleitet wird.

Achtung!

Fragen Sie den Status während der Digitalisierung nicht zu häufig ab, da jede Abfrage den PCI- / PCI Express-Bus belegt und dadurch die Decoder an der Datenübertragung hindert. Fügen Sie gegebenenfalls Wartezeiten zwischen den Abfragen ein, um die Digitalisierung nicht unnötig zu verlangsamen.

Achtung!

Achten Sie darauf, die richtigen, dem aktuellen Modus entsprechenden Bits auszuwerten, da ansonsten Ihr Programm möglicherweise zum falschen Zeitpunkt auf die Bilddaten zugreift.

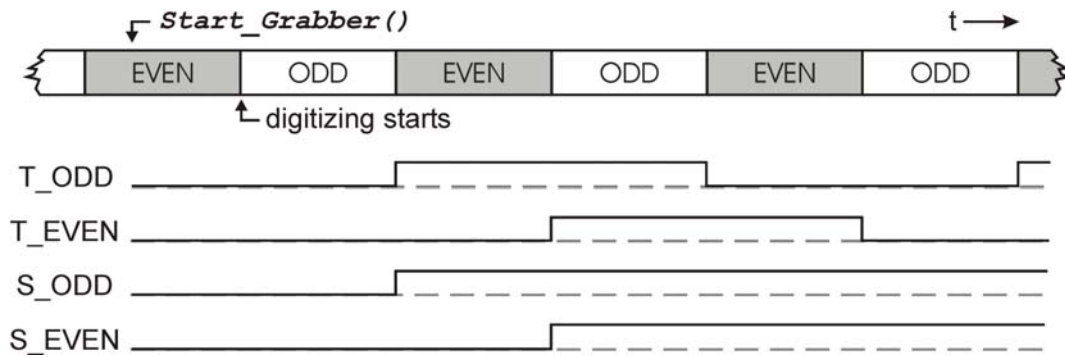


Bild 54: Timing Diagramm der Rückgabe-Parameters von data_present()

③ Bilddaten auslesen

**DWORD GetPictureBufferAddress(WORD nDevNo,
DWORD dwBitsSize) ;**

dwBitsSize: Größe des für das Bild benötigten Speicherbereichs

Returnwert: Adresse des Beginns des Speicherbereichs

Um die digitalisierte Bildinformation aus dem vom Gerätetreiber verwalteten Bildspeicher auslesen zu können, muss dem Anwenderprogramm die Startadresse dieses Bildspeichers bekannt sein.

Die Funktion **GetPictureBufferAddress** liefert dem Anwenderprogramm die Adresse zurück, an welcher der vom Treiber reservierte Speicherbereich beginnt. In diesem Speicher legt der jeweilige Decoder die digitalisierten Bilder wie folgt ab:

(a) Nur Even- oder nur Odd-Bilder werden digitalisiert

(d.h. die Dimensionsparameter des anderen Halbbilds sind Null)

⇒ Das digitalisierte Halbbild steht am Beginn des Bildspeicherbereichs.

(b) Even- und Odd-Bilder werden digitalisiert

(nInterlaced=0)

⇒ Das Even-Bild steht am Beginn des Bildspeicherbereichs. Bündig hinter dem Even-Bild beginnt das Odd-Bild. Die Startadresse des Odd-Bilds ist also die Startadresse des Bildspeichers plus $nEhsiz$ e $nEvsiz$ e Anzahl der Byte pro Pixel.

(c) Vollbilder werden im Interlace-Modus digitalisiert

(nInterlaced=1)

⇒ Das digitalisierte Vollbild besteht aus richtig ineinandergeschachtelten Halbbildern. Es beginnt an der Startadresse des Bildspeichers.

Der Parameter *dwBitsSize* gibt die Gesamtgröße des benötigten Bildspeichers an. Diese errechnet sich für ein Bild aus der Anzahl der Pixel und der pro Pixel benötigten Bytes: *hsize* *vsize* Pixelgröße.
Beachten Sie, dass *dwBitsSize* die Größe des **insgesamt** benötigten Bildspeichers angibt. Werden Even- und Odd-Halbbild getrennt digitalisiert, so müssen Sie den Speicherplatzbedarf beider Bilder addieren. Mehr zur Berechnung des Speicherplatzbedarfs erfahren Sie bei dem Befehl **SetImage()**.

③ ☆ Interrupt aktivieren

**DWORD ActivateFieldInterrupt (WORD nDevNo,
HANDLE *hEvent);**

nDevNo : System-Nummer des Decoders

HANDLE: Pointer auf ein Event, dass beim triggernden Ereignis ausgelöst wird.

Jeder Decoder kann über einen Interrupt das Ende eines Halbbild-Digitalisierungsvorgangs anzeigen. Dazu wird ein Event definiert, dass der Decoder beim Auftreten des Ereignisses (Halbbild digitalisiert) auslöst. Damit kann sehr schnell auf dieses Ereignis reagiert werden und das Polling (siehe *DataPresent()*) entfallen.

Die Funktion *ActivateFieldInterrupt()* dient zur Aktivierung der Interrupt-Funktion des Decoders. Gleichzeitig wird ein Zeiger auf ein Event übergeben, dass durch das triggernde Ereignis ausgelöst wird. Dieser Event wird vom Treiber angelegt.

Sie können in Ihrer Applikation auf diesen Event triggern, indem Sie die Windows-API-Funktion *WaitForSingleObject()* verwenden. Beachten Sie, dass Sie dem Event nach der Auswertung durch Aufruf der Funktion *ResetEvent()* zurücksetzen müssen, bevor Sie erneut auf den Event triggern können.

Wichtig:

- Das Event tritt immer ein, wenn die Digitalisierung eines Halbbilds abgeschlossen ist.

Dies ist nur der Fall, wenn der Decoder tatsächlich gestartet wurde.

Nach Eintreten des Ereignisses bleibt das Event bestehen (aktiv).

Sie können also auch zu einem späteren Zeitpunkt erkennen, ob eine Digitalisierung abgeschlossen wurde.

Um das Event zurückzusetzen, muss explizit die Funktion *ResetEvent()* aufgerufen werden. Ansonsten würden Sie beim nächsten Aufruf von *WaitForSingleObject()* sofort auf die noch anstehende Signalisierung des letzten Events reagieren.

Sie sollten *ActivateFieldInterrupt()* daher nur bei gestopptem Decoder aufrufen. Gegebenenfalls kann es sinnvoll sein,

sicherheitshalber ein `ResetEvent()` durchzuführen, um alte Signalisierungen zu löschen.

- Sofort nach Eintreffen des Events können Sie mit Hilfe der `DataPresent()` – Funktion herausfinden, welches Halbbild digitalisiert wurde. Bei kontinuierlicher Digitalisierung geschieht dies durch Abfrage der `T_EVEN / T_ODD` – Flags. Beachten Sie, dass diese Flags ihren Zustand nur für 20 ms nach dem Ende der Digitalisierung behalten, da dann schon wieder ein neues Halbbild eingelaufen ist.
Hinweis: Die Exklusiv-Oder-Verknüpfung von `T_EVEN` und `T_ODD` ergibt die Parität des fertig digitalisierten Halbbilds (1=ODD, 0=EVEN): `field_ready := T_EVEN ^ T_ODD ;`
- Der Event wird vom Grabber-Treiber mit einer nur sehr geringen Verzögerung erzeugt. Beachten Sie jedoch, dass die Signalisierung durch das System mit einer (variablen) Zeitverzögerung erfolgen kann.

③ Erkennung eines Videosignals

WORD Get_Signal_Status(WORD nDevNo);

Returnwert: 0 = kein Videosignal am Eingang vorhanden
 1 = undefiniert
 2 = Videosignal liegt an
 3 = Videosignal liegt an und Zeile ist eingerastet

Mit Hilfe dieser Funktion kann festgestellt werden, ob am eingestellten Kanal des Decoders eine Kamera angeschlossen ist oder nicht. Des Weiteren kann die Signalqualität beurteilt werden.

Der Returnwert 0 zeigt an, dass keine Videoquelle am aktuell ausgewählten Kanal anliegt. Werden für 31 Zeilen keine Synchronimpulse detektiert, so wird angenommen, dass kein Signal anliegt.

Wird 2 zurückgegeben, so ist eine Videoquelle erkannt worden. Zufällige Störungen am Eingang können jedoch zu einer fehlerhaften Aussage führen.

Der Rückgabewert 3 zeigt an, dass ein stabiles Videosignal anliegt, das mit sehr großer Wahrscheinlichkeit von einer Bildquelle kommt.

Der Wert wird generiert, wenn der Horizontal-Synchronimpuls innerhalb von ± 1 Clock von der erwarteten Position gefunden wird. Dies muss für 32 aufeinanderfolgende Zeilen gegeben sein. Umgekehrt wird der „HLOCK“-Zustand verlassen, wenn das Kriterium für 32 Zeilen nicht erfüllt wird. Es handelt sich also um ein sehr zuverlässiges Kriterium. Einige Bildquellen, wie z.B. Videorecorder, liefern von Natur aus kein stabiles Zeilentiming. Bei diesen Geräten kann es möglich sein, dass der Return-Wert 3 nie erreicht wird. Die Bilddigitalisierung erfolgt wegen Verwendung des ULTRALOCK™-Synchronisationsverfahrens in der Regel trotzdem einwandfrei.

③ ☆ **Anzahl der digitalisierten Bilder verfolgen**

Beim pciGrabber-4x4 ist es möglich, die Anzahl der digitalisierten Bilder zu verfolgen. Dazu stehen die folgenden zwei Funktionen zur Verfügung.

Diese Funktionen sind nicht kompatibel zum pciGrabber-4.

BYTE Get_CaptureCounter (WORD nDevNo)

Rückgabewert: Anzahl der gegrabten Fields modulo 256

Die Funktion gibt die Anzahl der digitalisierten Halbbilder eines Decoders wieder. Das Ergebnis ist ein Byte-Wert, der Zähler springt also von 255 wieder auf 0 um.

void Reset_CaptureCounter (WORD nDevNo)

Der Aufruf dieser Routine setzt den Halbbildzähler auf Null zurück.

④ Einstellen der Bildhelligkeit

void Set_Brightness(WORD nDevNo, short nBright);

nBright: Bildhelligkeit (-128..127)

Stellt das Register für die Bildhelligkeit im Videoprozessor ein. Der Wert bestimmt eine Konstante, die zu dem Helligkeitswert eines Pixels im Videoprozessor addiert wird. Die Helligkeit lässt sich so im Bereich $- 50 \%$ bis $+ 49,6 \%$ variieren. Ein LSB entspricht einer Helligkeitsänderung um $0,39 \%$:

$nBright = \text{Helligkeit } [\%] \cdot 2,5601 [1/\%]$

↔ ④ Helligkeitseinstellung zurücklesen

short Get_Brightness(WORD nDevNo);

Returnwert: Inhalt des Helligkeitsregisters des Videoprozessors

Diese Funktion ermöglicht das Auslesen des aktuellen Helligkeitswerts aus dem Videoprozessor.

④ Einstellen des Kontrasts

void Set_Contrast(WORD nDevNo, WORD nContr);

nContr: Bildkontrast (0..511)

Stellt den Kontrast des aufgenommenen Bilds ein. Der Kontrastwert stellt einen konstanten Faktor dar, mit dem der Helligkeitswert der Pixeldaten im Videoprozessor (entsprechend skaliert) multipliziert wird. Die Einstellung ist im Bereich 0% bis $236,57 \%$ möglich:

$nContr = \text{Kontrast } [\%] \cdot 2,1598 [1/\%]$

④ Kontrasteinstellung zurücklesen

WORD Get_Contrast(WORD nDevNo);

Returnwert: momentan eingestellter Kontrastwert

Die Funktion liest den Inhalt der Kontrast-Register aus dem Video-prozessor und liefert sie als Ganzzahl-Wert zurück.

④ Einstellen der Farbsättigung

**void Set_Saturation(WORD nDevNo,WORD nSat_U,
WORD nSat_V);**

nSat_U: Sättigung des U-Farbanteils (0..511, Default = 254)

nSat_V: Sättigung des V-Farbanteils (0..511, Default = 180)

Über diese Funktion wird die Farbsättigung des Bilds getrennt für U- und V-Farbanteil eingestellt. Es handelt sich dabei um eine Veränderung des Verstärkungsfaktors getrennt für die beiden Farbanteile.

Normalerweise ist das Verhältnis Werte *nSat_U* und *nSat_V* gleich. Durch eine Differenz von U- und V-Anteil können Farbstiche (z.B. von nicht abgeglichenen Farbkameras) ausgeglichen werden, also die Farbe des Bilds verändert werden.

$nSat_U = \text{U-Sättigung [\%]} \cdot 2,5400 [1/\%] ; 0\% \text{ bis } 201,18 \%$

$nSat_V = \text{V-Sättigung [\%]} \cdot 2,1396 [1/\%] ; 0\% \text{ bis } 238,83 \%$

④ Inhalte der Farbsättigungsregister lesen

WORD Get_Sat_U(WORD nDevNo);

WORD Get_Sat_V(WORD nDevNo);

Returnwert: Wert der aktuellen U- bzw. V-Farbsättigung

Der Inhalt der Farbsättigungsregister kann durch Aufruf dieser Routinen ermittelt werden.

④ Farbton korrigieren (nur bei NTSC)

void Set_Hue(WORD nDevNo, short nHue);

nHue: Farbton, Phasenlage des Farbträgers (-128..127)

Mit dieser Funktion wird bei Digitalisierung von NTSC-Farbbildern der Farbton eingestellt, indem die Phasenlage des Farbträgers variiert wird. Bei PAL hat dieser Wert keine Bedeutung, da hier durch das Farbsystem selbst Phasenfehler automatisch ausgeglichen werden. Ein LSB entspricht einer Phasenwinkelkorrektur um $0,7^\circ$, der Farbträger kann somit im Bereich $- 89,3^\circ$ bis $+ 90^\circ$ nachgestellt werden. Damit der Farbdekoder unter PAL korrekt arbeitet, muss dieser Wert auf 0 gesetzt werden.

④ Inhalt des Farbton-Registers zurücklesen

short Get_Hue(WORD nDevNo);

Returnwert: äquivalent der eingestellten Phasenlage des Farbträgers

Diese Funktion dient dem Rücklesen des eingestellten Hue-Wertes.

④ Ein-/Ausschalten des Luma-Tiefpassfilters

void Set_LDec(WORD nDevNo, WORD nOn, WORD nHFilt);

nOn: 1 = Luma Decimation einschalten
0 = Luma Decimation ausschalten

nHFilt: 0 = Automatische Filterauswahl
1 = CIF Filter
2 = QCIF Filter
3 = ICON Filter

Bei kleinen Bildformaten erzielt man eine bessere Bildwiedergabe, wenn die Auflösung des Eingangssignals reduziert wird (man also die Schärfe des Bilds an die spätere Auflösung anpasst). Dazu kann mit dieser Routine optional ein Tiefpaßfilter in den Luma-Pfad eingeschaltet werden.

Mit dem Parameter *nHFilt* wird der verwendete Filter an die Bildgröße angepasst:

Automatische Filterwahl macht die Filtereinstellung abhängig von der gesetzten Bildgröße (siehe **Set_Image**). Daneben kann die Filter manuell auf eine der normierten Bildformate CIF (= 1/2 Vollbild), QCIF (1/4 Vollbild) und ICON (1/8 Vollbild) angepasst werden.
Default: Luma-TP ist ausgeschaltet.

④ Ein-/Ausschalten des Testbilds

void Set_ColorBars(WORD nDevNo, WORD nColorBars);

nColorBars: 0 = Testbild ausblenden
1 = Testbild einschalten

Diese Funktion blendet ein Testbild ein. Das Testbild besteht aus senkrechten, farbigen Balken. Das Testbild ist unabhängig von einem Eingangssignal. Um das Bild vollständig sehen zu können, sollte die Bildgröße etwa CIF-Format betragen.

④ Kontrolle des Wertebereichs

```
void LumaControl(WORD nDevNo,  
                WORD nRange,  
                WORD nCore);
```

nRange: 0 = Luma Spannweite 16 - 253
 1 = Luma Spannweite 0 - 255
nCore: 0 = 0 alle Helligkeitswerte werden übermittelt
 2 = 16 alle Helligkeitswerte <= 16 werden als 0 interpretiert
 3 = 32 alle Helligkeitswerte <= 32 werden als 0 interpretiert

Mit dieser Funktion kann das Ausgabeformat der Helligkeits- und Farbwerte an die Applikation angepasst werden.

nRange bestimmt den Wertebereich der Helligkeit (mögliche Grauwerte).

- *nRange* = 0 entspricht dem normalen Wertebereich, der in CCIR 601 spezifiziert ist. Der Helligkeitsbereich ist damit auf die Werte 16 bis 253 begrenzt, wobei der Wert $Y = 16$ schwarz entspricht. Der Farbwertebereich ist 2...253 mit $Cr/Cb = 128$ als Null (vorzeichenbehaftet).
- *nRange* = 1 ermöglicht die Nutzung des vollen Wertebereichs, das ist für Y der Bereich 0...255 mit 0 = schwarz, der Chroma-Bereich ist wie bei *nRange* = 0 definiert.

⑤ Daten über den GPIO-Port lesen und ausgeben, Jumper auslesen

An der Stiftleiste *Option-Port* sind 8 I/O-Leitungen verfügbar, die über die folgenden drei Funktionen gesteuert werden können:

void Set_GPIO_Direction (WORD nDevNo, WORD nDirection);

void Set_GPIO_Data (WORD nDevNo, WORD nData);

WORD Get_GPIO_Data (WORD nDevNo);

nDirection: Richtungskontrolle für GPIO-Port (0...4096), bitcodiert

nData: Daten, die über den GPIO-Port ausgegeben werden sollen

Rückgabewert Get_GPIO_Data(): Daten, die vom GPIO-Port gelesen werden

Der pciGrabber-4x4 besitzt 8 Erweiterungspins, welche getrennt zum Einlesen oder zum Ausgeben von beliebigen Digitalsignalen genutzt werden können. Mit den GPIO-Funktionen steuern Sie den Erweiterungspins. Sie können definieren, welche der Pins als Eingang und welche als Ausgang fungieren, Ausgangspins auf High- oder Low-Pegel setzen und den Status von Eingangspins lesen.

Weiterhin können Sie mit dieser Funktion den Zustand der Jumper auf der Framegrabberkarte abfragen.

Die Jumper werden über die Bits 9 bis 12 angesprochen.

Das heißt Bit 0 bis 7 sind die I/Os des Option Ports und Bit 8 bis 11 die Jumper:

11 ³ ₂₀₄₈	10 ⁴ ₁₀₂₄	9 ² ₅₁₂	8 ¹ ₂₅₆	7 ³ ₁₂₈	6 ⁴ ₆₄	5 ² ₃₂	4 ¹ ₁₆	3 ² ₈	2 ⁴ ₄	1 ² ₂	0 ¹ ₁
JP4	JP3	JP2	JP1	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
0	0	0	0	0	0	0	0	0	0	0	0

GPIO_Data()

Ein gesteckter Jumper liefert den Wert 0 zurück.

Hinweis:

Die GPIO-Ports des *Option-Ports* sind mit dem ersten Video-Decoder der Grabberkarte verbunden.

Um auf diese Funktionen zugreifen zu können, müssen Sie für `nDevNo` die Geräteadresse des (jeweils) ersten Videodecoders einsetzen. Dies ist der Decoder, der bei `Read_Order-Code()` die Bezeichnung `VD-012[-xx] Decoder 1` zurückliefert.

Der Aufruf dieser Funktionen bei anderen Decodern ist wirkungslos.

Mit Hilfe der Funktion **Set_GPIO_Direction** kann jeder Portpin getrennt als Ein-oder Ausgabepin konfiguriert werden. Hierzu werden die unteren 8 Bits des Parameters *nDirection* ausgewertet. Ist ein Bit auf 1 gesetzt, wird der entsprechende Portpin auf Ausgabe konfiguriert. Eine 0 bewirkt eine Umschaltung des Portpins auf Einlesen.

Achtung!

Achten Sie darauf, dass ein Portpin erst dann auf Ausgabe geschaltet wird, wenn kein externes Signal an diesem Pin anliegt. Ansonsten ist es möglich, dass der Pin zerstört wird.

Beim Starten des Rechners sind alle Pins auf Eingabe konfiguriert. Beachten Sie, dass die Pins dann hochohmig sind und damit der logische Pegel nicht definiert ist. Um das Verhalten von angesteuerten Komponenten während des Starten des Rechners bis zur Konfigurierung des GPIO-Ports zu bestimmen, müssen daher hardwaremäßig entsprechende Maßnahmen vorgesehen werden (z.B. Pull-Up-Widerstände).

Setzen Sie die den Jumpers zugeordneten Bits nicht auf „Ausgang“!

Mit **Set_GPIO_Data** können Sie denjenigen Portpins, die auf „Ausgang“ geschaltet sind, den Pegel „High“ = 1 oder „Low“ = 0 zuordnen. Bei `nData` handelt es sich wieder um einen 12-bit-Wert, wobei jedes Bit einem Pin zugeordnet ist. Es wirken sich nur die Bits aus, für die die Datenflußrichtung auf „Ausgang“ gestellt ist.

Die Funktion **Get_GPIO_Data** liest den Zustand der als „Eingang“ geschalteten Portpins ein und liefert diesen als 12-bit-Wert zurück. Für die auf „Ausgang“ geschalteten Pins wird der Wert zurückgeliefert, auf den der Pin gesetzt wurde.

⑤ ☆ Daten über die I²C-Schnittstelle übertragen

Mit diesen Funktionen können Sie Bausteine, die an der I²C-Schnittstelle angeschlossen sind, lesen und beschreiben.

Hinweis:

Die I²C-Schnittstelle des *Option-Ports* ist mit dem ersten Video-Decoder der Grabberkarte verbunden.

Um auf diese Funktionen zugreifen zu können, müssen Sie für `nDevNo` die Geräteadresse des (jeweils) ersten Videodecoders einsetzen. Dies ist der Decoder, der bei `Read_Order-Code()` die Bezeichnung `VD-012[-xx] Decoder 1` zurückliefert.

Der Aufruf dieser Funktionen bei anderen Decodern ist wirkungslos.

Achtung!

Das auf der Grabberkarte befindliche I²C-EEPROM ist vor versehentlichem Überschreiben geschützt. Daher sind keine Zugriffe auf den Geräteadressenbereich `0xA0` bis `0xA3` möglich.

Zum Zugriff auf den internen EEPROM-Speicherbereich benutzen Sie bitte die dafür vorgesehenen gesonderten Funktionen.

void I2C_Set_BR_Mode (WORD nDevNo, BYTE bMode)

bMode Baudrate
pciGrabber-4x4: 0 = 99,2 kHz, 1 = 396,8 kHz

Diese Funktion bestimmt die Baudrate, mit der die Übertragung auf dem I²C-Bus stattfindet. Es sind eine hohe und eine niedrige Übertragungsrate wählbar.

BYTE I2C_ReadByte (WORD nDevNo, BYTE bChipAddress, BYTE bSubAddress, BYTE *bByteRead)

bChipAddress: Geräteadresse des I²C-Geräts auf dem Bus
bSubAddress: Speicher-Adresse innerhalb des I²C-Geräts
*bByteRead: Zeiger auf eine Byte-Variable, in die das Ergebnis geschrieben wird

Returnwert: SUCCESS, NOACK, INVALID_ADDRESS

Mit *I2C_ReadByte* wird ein Byte aus einer Speicherstelle eines I²C-Geräts gelesen werden. Das Ergebnis wird in einer Variablen vom Typ Byte abgelegt, die zuvor definiert werden muss.

Als Returnwert gibt die Funktion einen Fehlercode zurück. `NOACK` zeigt an, dass sich unter der angegebenen Geräteadresse kein I²C-Gerät gemeldet hat. `INVALID_ADDRESS` bedeutet, dass versucht wurde, auf den geschützten Bereich des auf dem Grabber befindlichen EEPROMs zuzugreifen.

BYTE I2C_WriteByte (WORD nDevNo, BYTE bChipAddress, BYTE bSubAddress, BYTE bData)

bChipAddress: Geräteadresse des I²C-Geräts auf dem Bus

bSubAddress: Speicher-Adresse innerhalb des I²C-Geräts

bData: Byte, das in die spezifizierte Adresse geschrieben wird

Returnwert: `SUCCESS`, `NOACK`, `INVALID_ADDRESS`,
`WRITE_FAILED`

Schreibt ein Byte in eine Speicherstelle des spezifizierten I²C-Geräts. Als Returnwert gibt die Funktion einen Fehlercode zurück (s.o.).

Hinweis:

Manche Geräte – wie z.B. EEPROMs – benötigen u.U. intern längere Verarbeitungszeiten für einen Schreibbefehl als der I²C-Buszyklus dauert. Folgen mehrere Schreibbefehle schnell aufeinander, so liefern diese Geräte eine Fehlermeldung (`NOACK`) zurück, da sie evtl. noch nicht wieder schreibbereit sind.

Um dies zu vermeiden, können Sie nach dem Schreiben auf einen solchen Baustein einen Lesezugriff durchführen und prüfen, ob der zurückgelesene Wert mit dem Eingeschriebenen übereinstimmt. Erst wenn dies der Fall ist, wurde der interne Schreibvorgang abgeschlossen, und Sie können erneut schreibend auf den Baustein zugreifen.

Beachten Sie diesbezüglich auch die Dokumentation des I²C-Geräts.

⑤ ☆ internes EEPROM verwenden

Der pciGrabber-4x4 besitzt pro Decoder einen internen, nichtflüchtigen Speicher, der vom Anwender benutzt werden kann, um beliebige Parameter zu speichern.

Insgesamt stehen in diesem Speicher 252 Bytes zur Verfügung.

BYTE I2C_ReadEEProm (WORD nDevNo, BYTE bSubAddress, BYTE *bByteRead)

bSubAddress: zu lesende Speicheradresse (0x00 ... 0xFB)

*bByteRead : Zeiger auf eine Byte-Variable, in die das Ergebnis geschrieben wird

Returnwert: Fehlercode = SUCCESS, NOACK

Liest aus dem EEPROM einen Byte-Wert aus. Spezifiziert wird bei Aufruf die Speicheradresse, die gelesen werden soll. Das Ergebnis wird in einer Variablen vom Typ Byte abgelegt, die zuvor definiert werden muss.

Die Funktion gibt als Returnwert einen Fehlercode zurück (siehe *I2C_ReadByte*).

BYTE I2C_WriteEEProm (WORD nDevNo, BYTE bSubAddress, BYTE bData)

bSubAddress: Speicheradresse, auf die geschrieben wird (0x00...0xFB)

bData: Datenbyte, das geschrieben wird

Returnwert: Errorcode = SUCCESS, NOACK, WRITE_FAILED

Schreibt ein Byte in das interne EEPROM. Spezifiziert wird die gewünschte Speicheradresse und das dorthin zu schreibende Byte.

Die Funktion liefert einen Fehlercode zurück (siehe *I2C_WriteByte*).

Hinweis:

Die Lebensdauer der internen EEPROM-Speichers ist mit 1 Million Schreibzugriffen spezifiziert. Die Anzahl der Lesezugriffe ist nicht begrenzt.

↔ ⑤ Direktes Lesen und Beschreiben von Registern

**WORD Read_Local_DWord (WORD nDevNo,
WORD nRegister_Number,
DWORD *IContent);**

nRegister_Number: Nummer des Registers
IContent: Inhalt des Registers

**WORD Write_Local_DWord (WORD nDevNo,
WORD nRegister_Number,
DWORD IContent);**

nRegister_Number: Nummer des Registers
IContent: Inhalt des Registers

Fast alle Funktionen des Video-Decoders lassen sich über die Routinen des Treibers steuern. Wir empfehlen ausdrücklich die Verwendung der Standard-Funktionen.

Für den Fall, dass der Anwender direkt Register eines Decoders beeinflussen möchte, stehen diese beiden Funktionen zur Verfügung, mit denen sich alle Register des Videoprozessors lesen und beschreiben lassen. Sollten Sie weitere Informationen über die Register benötigen, wenden Sie sich bitte an den PHYTEC-Support.

Achtung!

Manche Register des pciGrabber-4x4 müssen in der vorliegenden Gerätekonfiguration mit bestimmten Werten belegt werden, um die Funktion der Karte sicherzustellen.

Prüfen Sie daher vor Änderungen genau die Bedeutung der veränderten Register / Bits und die Auswirkungen.

PHYTEC kann keine Haftung für Schäden übernehmen, die durch Manipulation der Register möglicherweise entstehen.

7 Störungssuche

- Die Farbdarstellung im Windows-Demo-Programm ist stark reduziert.
 - Prüfen Sie die Einstellung Ihrer Grafikkarte. Um die volle Farbtiefe des pciGrabber-4x4 darstellen zu können, muss die Grafikkarte auf mindestens 16 Mio. Farben konfiguriert sein.

- Es ist nur ein blaues Bild zu sehen.
 - An dem gewählten Eingang ist keine Videoquelle angeschlossen. In der Regel wird dann ein blaues Bild geliefert. Prüfen Sie, ob der richtige Eingang eingestellt ist.

- Das digitalisierte Bild hat Schlieren / Streifen
 - Es kann sich um Moiré-Störungen bedingt durch den Farbträger handeln. Prüfen Sie, ob die Farbfalle eingeschaltet ist.
 - Die Signalleitung von der Kamera ist defekt (Abschirmung prüfen).
 - Eine Masseschleife durch eine zusätzliche Masseverbindung zwischen PC und Kamera ist Ursache von Brummstörungen. Prüfen Sie, ob z.B. durch Erdung oder Spannungsversorgung einen Brummschleife aufgebaut wurde.

- Am S-Videoeingang wird kein Bild digitalisiert.
 - Ist der *S-Video*-Eingang korrekt verbunden?
 - Wurde der pciGrabber-4x4 richtig auf *S-Video*-Betrieb konfiguriert?
 - Wurde die richtige Eingangsbuchse gewählt?
 - Wurde nach Umschaltung auf *S-Video* mit Set_Channel ein anderer Kanal gewählt?

- Bei *S-Video*-Betrieb ist das Bild nur schwarz/weiß
 - Wird das Chroma-Signal korrekt zugeführt?
 - Ist der Grabber auf *S-Video*-Betrieb umgestellt?

- Beim *S-Video*-Betrieb ist das Bild schwarz/weiß mit starken Farbstörungen.
 - Ist das Chroma-Signal korrekt angeschlossen?
 - Handelt es sich um eine *S-Video*-Quelle?
 - Ist keine Bildquelle mehr angeschlossen?

- Das Bild wird unvollständig / sehr langsam angezeigt
 - Erhöhen Sie die Wartezeit zwischen den Statusabfragen. Der PCI Express-Bus wird durch zu häufige Statusabfragen blockiert.
 - Wurde die zeitliche Auflösungsreduzierung benutzt?

- Das Bild ist kontrastarm / zu hell / zu dunkel
 - Prüfen Sie die Einstellungen von Helligkeit, Kontrast usw.
 - Schalten Sie ggfs. die AGC ein

- Das Bild springt oder zeigt Halbbildversatz
 - Es wurde keine genügend große Wartezeit zwischen dem Umschalten der Eingangskanäle eingehalten.

- Das Bild erscheint ohne Farbe / im falschen Format
 - Ist das richtige Farbsystem eingestellt?
 - Wurde der Grabber korrekt initialisiert?
 - Wurde eine ausreichende Wartezeit bei der Kanalumschaltung eingehalten?

- Der pciGrabber-4x4 verliert nach einem Standby unter Windows seine Funktionsfähigkeit.
 - Der pciGrabber-4x4 unterstützt den Standbybetrieb unter Windows nur, wenn die 3,3 Volt Spannungsversorgung auf dem PCI Express-Bus nicht abgeschaltet werden. Ansonsten muss ein Neustart durchgeführt werden.

- Am unteren Bildrand fehlt ein Stück / es wird kein Bild digitalisiert
 - Die vertikale Bildgröße wurde zu groß gewählt.
 - Es wurde nicht beachtet, dass ab einer Zeilenzahl von mehr als 288 (PAL) bzw. 262 (NTSC) mit Vollbildern gearbeitet werden muss.

- Die Farben werden nicht korrekt dargestellt.
 - Die Werte *hpos* und *hsize* müssen gerade sein, damit die Farbrückgewinnung korrekt arbeitet.
 - In eigenen Programmen: Sie haben Cr/Cb vertauscht.
 - Haben Sie das richtige Farbformat gewählt?
 - Die Farbsättigungsregister wurden verstellt. Beachten Sie, dass für eine korrekte Farbtondarstellung die U- und V-Farbsättigung im *Prozentwert* identisch sein müssen. Die *Registerwerte* sind dabei jedoch unterschiedlich!

- Beim kontinuierlichen Grabben springt das Bild um eine Zeile nach oben / unten.
 - Es wird nicht die Parität des Halbbilds beachtet. Achten Sie darauf, stets das gleiche Halbbild anzufordern. Speicherbereich von Even- und Odd-Bild dürfen nicht gleich sein.

Hinweis:

Der Versatz beträgt genaugenommen eine halbe Zeile, deswegen man nicht durch Versatz um eine Zeile das Fehlerbild kompensieren kann.

- Die Videoquelle liefert kein sauberes Signal
- Die Kanalumschaltung zwischen zwei Kameras erfolgt zu schnell

- Bei der Darstellung von Vollbildern sind diagonale Linien / Kreise auffällig gerastert oder haben gezackte Linien.
 - Gerades und ungerades Halbbild werden vertauscht dargestellt (nur möglich, wenn Sie einzelne Halbbilder benutzen).

- Nach der Digitalisierung eines Bildes ist keine weitere Digitalisierung mehr möglich.
 - Sie müssen vor der Anforderung eines neuen Digitalisierungsvorgangs den vorherigen durch **Stop_Grabber()** abschließen.

- Im Vollbild-Modus entsteht bei kontinuierlicher Darstellung ein Nachzieheffekt, obwohl doch der Grabber zwei zueinander gehörende Halbbilder in Echtzeit digitalisiert.
 - Der Effekt ist in der endlichen Darstellungsgeschwindigkeit des Rechners begründet. Die Aktualisierung des Bildschirms wird nicht genügend schnell vorgenommen, obwohl die Bilder in Echtzeit im Arbeitsspeicher vorliegen.

- Das dargestellte Bild ist durch horizontale Streifen unterbrochen, in denen man möglicherweise den Inhalt vorhergehender Bilder sieht. Bei bewegten Objekten ergeben sich horizontale Nachzieheffekte.
 - Der Grabber kann die Bilddaten nicht in Echtzeit über den PCIExpress-/PCI-Bus übertragen, da andere Karten/Decoder den Bus zu stark beanspruchen oder die Bus-Einstellungen des BIOS nicht korrekt sind. Überprüfen Sie die Einstellungen der anderen Karten und die Einstellungen des BIOS.

Index

A

Adressbereich	19
AGC	
(Treiber).....	117
analoge Fernsehtechnik	58
Anschlußbelegung	21
Anschlüsse	15
<i>Arithmetics</i>	66
Arithmetische Operationen an	
Standbildern	66
Auflösung	124
Ausschnitt	125
AUTO-Funktion	
Treiber	112

B

Basisadresse.....	19
Bewegung	80
Bild	
Auflösung	124
Ausschnitt	125
blau (Störung).....	156
Digitalisierungsmodus.....	134
falsches Format (Störung) ...	157
Farbformat	130
Farbsättigung (Treiber).....	146
Farbstörungen	157
Größe	124
Halbbildversatz (Störung) ...	157
horizontale Streifen (Störung)	
.....	159
kein (Störung).....	156
kleine Formate (Treiber)	148
kontrastarm (Störung).....	157
Parameter (Treiber)	125
Raster/Zacken (Störung).....	158
schwarz/weiß (Störung).....	156
springen (Störung).....	158

unvollständig / fehlt (Störung)	
.....	158
unvollständig / langsam	
(Störung).....	157
Bildauflösung	13
Bilddaten	
Übertragung in Speicher	83
Bilder	
auslassen	119
Bilder abspeichern	
(Demoprogramm)	71
Bildgröße	123
Bildhelligkeit	145
Bildhelligkeitssignal	81
Bildrauschen	
verringern (Demoprogramm)	68
Bildspeicher	133
Auslesen unter Windows	140
BNC-Stecker	38
<i>BtYUV</i>	131

C

Chroma-Signal	81
CIF	148
<i>Color Bars</i>	66
<i>Color Meter</i>	65
<i>Color Mode</i>	50
Composite-Eingänge.....	23, 36
Compositequelle	
anschießen	38
Composite-Quellen	57
<i>Cross-Color-Effekt</i>	115

D

Datenformat	130
Datenformate	13
Delphi	
Programmieren unter	97
Demoprogramm	

automatisch starten	63
Bedienung	44
Channel selection.....	56
Grundeinstellung.....	45
Image Resolution	57
Image Selection	57
Image Settings	56
Installation	33
Luma-Low-Pass.....	61
Normierung.....	67
Digitalisierung	
Grundlagen	88
nur eine (Störung).....	159
Zeitpunkt.....	136
Digitalisierungsvorgang	
starten.....	135
DLL.....	90
Anwendung.....	95
DMA-Kanäle	83
E	
EEPROM	154
Entwicklungsumgebungen.....	90
Erweiterungsport.....	26
programmieren.....	150
even (Halbbild)	78
F	
Fadenkreuz	
einblenden.....	64
Farbauszüge	81
Farbbalken	66
Farbdarstellung	
reduziert (Störung).....	156
Farbdifferenzsignal	81
Farbfalle	115
Farbsättigung	62, 146
Farbstörungen	157
Farbton	62
Korrektur (NTSC).....	147
Farbtöter.....	118
Farbwerte	
messen.....	65
<i>Field Aligned</i>	116
Flimmereffekt.....	78
<i>Frame-Rate</i>	50
Funktionen	
GetVersionNumber (Win)	
.....	103
Data_Present().....	138
FreeLibrary(...).....	95
Get_Brightness()	145
Get_CaptureCounter().....	144
Get_Contrast().....	146
Get_Error().....	102
Get_GPIO_Data()	150
Get_Hue()	147
Get_Sat_U()	146
Get_Sat_V()	146
Get_Signal_Status()	143
Get_Video_Status().....	109
GetPictureBufferAddress()	
(Win).....	140
GetProcAddress(...).....	95
I2C_ReadByte.....	152
I2C_ReadEEProm()	154
I2C_Set_BR_Mode()	152
I2C_WriteByte()	153
I2C_WriteEEProm()	154
Initialize().....	105
Klassifizierung	99
LoadLibrary(...).....	95
LumaControl()	149
Max_Device_Number()	104
Read_GrabberInfo().....	106
Read_Local_DWord().....	155
Read_OrderCode().....	108
Reset_CaptureCounter	144
Set_AGC()	117
Set_Brightness().....	145
Set_BW()	115
Set_ChannelEx().....	113
Set_CKill()	118
Set_Color_System().....	109
Set_ColorBars()	148

Set_Composite()	110	K	
Set_Contrast()	145	Kammeffekt	60
Set_GPIO_Data()	150	Kompatibilität zum	
Set_GPIO_Direction.....	150	pciGrabber-4	101
Set_Hue().....	147	Kontrast.....	62, 145
Set_Image() (Win).....	122	L	
Set_Interlace()	116	Lieferumfang	10
Set_LDec()	148	Live Image	63
Set_S_VideoEx().....	111	Live-Bild.....	44
Set_Saturation().....	146	M	
Start_Grabber().....	135	Moiré-Störungen.....	156
Stop_Grabber()	137	N	
TemporalDect()	119	Nachzieheffekt.....	159
Write_Local_DWord()	155	Non-Interlace	
		(Treiber).....	116
G		O	
Gerätetreiber		odd	
Win 98/ME	30	(Halbbild).....	78
Win NT 4.0.....	31	Open Image on Start	63
<i>Gr4CDLL.DLL</i>	96	P	
Grabberkarte		Parameterspeicher.....	154
Einbau.....	28	PCI-Steckplatz	28
H		Pfostenstecker	26
Halbbild	79, 116	Portpin.....	151
Halbbild-Digitalisierung	127	Ports	14
Halbbild-Digitalisierung mit		Q	
Zoom	128	QCIF	148
Halbbilder	58	R	
Halbbildversatz.....	157	R,G,B	81
Helligkeit	62, 145	Register	155
Histogramm	64	RGB	130
I		<i>RGB15</i>	131
I/O-Port		<i>RGB16</i>	130
testen (Demoprogramm)..	69, 70	<i>RGB24</i>	130
I ² C-Schnittstelle	14, 27	<i>RGB3</i>	130
programmieren	152		
ICON	148		
Interlace			
(Treiber).....	116		
Interrupt	19		

S

s/w-Signal (Treiber)	115
Schlieren / Streifen (Störung) .	156
Schnappschüsse	63
Sicherheitshinweise	16
Signalqualität	143
Single Image	63
Snapshot	63
Softwaretreiber.....	73
Standby	157
S-Video	24
S-Video - Quellen	57
Synchronisation	13

T

Testbild	148
Treiber	
AGC Ein/Ausschalten.....	117
Anzahl der digitalisierten Bilder verfolgen	144
Anzahl der Grabber bestimmen	104
Auslassen von Fields, Frames	119
Bildaufnahme starten	135
Bildaufnahme stoppen	137
Bilddaten auslesen (Win).....	140
Bildgröße und Skalierung einstellen	122
Bildhelligkeit einstellen	145
Composite Eing. einstellen ..	110
Digitalisierungsfortschritt erkennen.....	138
Ein-/Ausschalten der Farbfalle	115
Eingangskanals einstellen...	113
Erkennung des Videoformats	109
Farbsättigung einstellen	146
Farbsättigung lesen	146
Farbsystem einstellen.....	109

Farbton korrigieren (NTSC) 147	
Farbton-Register lesen	147
Farbtöter Ein-/Ausschalten ..	118
Fehlermeldungen auswerten	102
GPIO-Port programmieren ..	150
Grabber und Treiber	
initialisieren	105
Grabbername als Klartextstring lesen	108
Helligkeitseinstellung zurücklesen	145
I ² C-Schnittstelle verwenden.	152
Informationen über Grabber	
abfragen.....	106
installieren.....	30
Interlaced-Modus	
ein/ausschalten	116
internes EEPROM verwenden	154
Kontrast einstellen	145
Kontrastwert auslesen	146
Lesen / Schreiben von Registern	155
Luma Werte kontrollieren....	149
Luma-Tiefpass ein/ausschalten	148
S-Video-Modus einstellen ...	111
Testbild ein/ausschalten	148
Versionsnummer der DLL bestimmen (Win)	103
Videosignals erkennen	143
Windows'98 / NT	94
Treiber-Bibliothek.....	73
TV-Bild	78
Twain-Treiber	32
<i>Type Casting Settings</i>	68

U

Umrechnung	
RGB / YUV.....	81

V

Vermessungsanwendungen.....	124
-----------------------------	-----

Versorgung	12	X	
Videoquellen		X6 (Optionport)	26
anschließen	36	Y	
Videoquellen anschließen.....	34	Y (Helligkeitsinformation) ...	133
Videosignal.....	78	Y,Cr,Cb.....	81
Vollbild	59, 134	Y8 (<i>Grauwert-Format</i>).....	133
Vollbilddigitalisierung.....	60	YCrCb	131
Vollbild-Digitalisierung	128	YUY2	131
Grundlagen	88	Z	
Vollbildmodus	134	Zubehör.....	11

Dokument: pciGrabber-4x4
Dokumentnummer: L-720d_1, November 2009

Wie würden Sie dieses Handbuch verbessern?

Haben Sie in diesem Handbuch Fehler entdeckt? Seite

Eingesandt von:

Kundennummer: _____

Name: _____

Firma: _____

Adresse: _____

Einsenden an:

PHYTEC Technologie Holding AG
Postfach 100403
D-55135 Mainz, Germany
Fax : +49 (6131) 9221-26