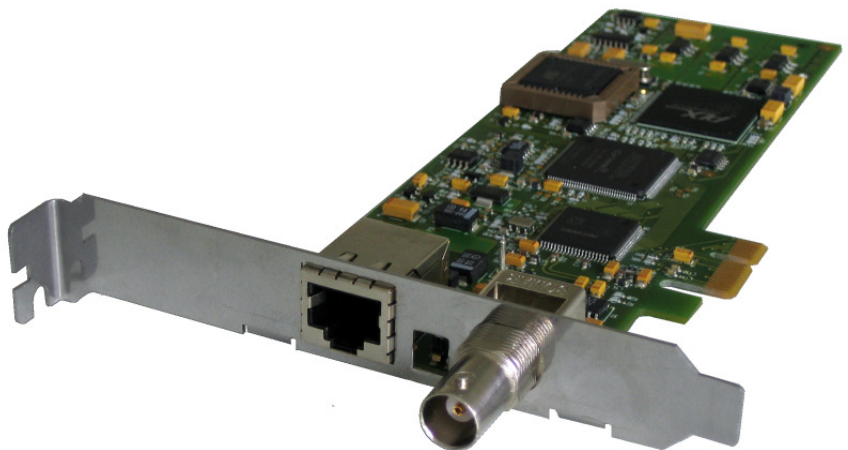


Timecode Leser Karte

PCL PCI / PCL PCIe



PCL PCI D



PCL PCIe HD

Inhaltsverzeichnis	Seite
A1 HINWEISE ZUM SICHEREN GEBRAUCH	
A2 COPYRIGHT	
A3 CE-ERKLÄRUNG	
B FUNKTIONSBESCHREIBUNG	1
B1 ÜBERSICHT	1
B2 TIMECODE FUNKTIONEN	2
Timecode Register	2
Framerate	3
Fehlerüberprüfung	3
Schwungrad	4
VITC Levelanpassung	4
DVITC Levelanpassung	4
B3 ANSCHLÜSSE UND TECHNISCHE DATEN	5
PCL PCI	6
PCL PCIe	7
B4 INSTALLATION	8
B5 FIRMWARE UPDATE	9
C PROGRAMMIERUNG	10
C1 WINDOWS-TREIBER AVPCL32	10
Struktur	10
Treiber-Dateien	10
SDK-Dateien	12
C2 DATENZUGRIFF	13
Speicher- bzw. I/O-Zugriff	13
Registersatz	13
Register	14
Kommandos	16
D OPTIONEN	28
D1 OPTION „GPI OUT“: ZWEI SIGNALAUSGÄNGE	30
Beschreibung	30
Kommandos	31
Anschluss und technische Daten	32
Treiber und Anwenderprogramme	33

A1 Hinweise zum sicheren Gebrauch

- Allgemein gilt** Benutzen Sie das Gerät nur zum bestimmungsgemäßen Gebrauch in trockenen Räumen. Behandeln Sie die **PCL PCI / PCL PCIe** mit der gleichen Sorgfalt, mit der auch andere PC-Steckkarten behandelt werden müssen. Beachten Sie die entsprechenden Hinweise in der Bedienungsanleitung unseres Gerätes.
- Transportschäden** Bei offensichtlichen Transportschäden muss das zuständige Speditionunternehmen benachrichtigt werden. Setzen Sie sich in diesem Fall sofort mit Ihrem Händler in Verbindung.
- Reparaturen** Ihr Gerät ist dank modernster elektronischer Bauteile wartungsfrei. Auf der Karte befinden sich keine Teile, die von Ihnen repariert werden können. **Überlassen Sie daher jeden Eingriff nur einem autorisierten Servicepartner.**
- EMV** Zur Einhaltung der EMV-Richtlinien müssen für alle Datenanschlüsse hochqualitative abgeschirmte Kabel verwendet werden.

A2 Copyright

Copyright © Alpermann+Velte Electronic Engineering GmbH 2009. Alle Rechte vorbehalten. Informationen in dieser Funktionsbeschreibung ersetzen alle vorhergehend publizierten Informationen. Technische Änderungen sind vorbehalten. Die Nennung von Produkten anderer Hersteller in dieser Bedienungsanleitung dient ausschließlich Informationszwecken und stellt keinen Warenzeichenmissbrauch dar.

Für weitere Informationen wenden Sie sich bitte an:

Alpermann+Velte

Electronic Engineering GmbH

Otto-Hahn-Str. 42

D-42369 Wuppertal

Fon.: ++49 - (0)202 – 244 111 0

Fax: ++49 - (0)202 – 244 111 5

E-Mail: info@alpermann-velte.com

Internet: <http://www.alpermann-velte.com>

A3 CE-Erklärung

Alpermann+Velte

Electronic Engineering GmbH
Otto-Hahn-Straße 42
D-42369 Wuppertal

erklärt hiermit, dass das Produkt

PCL PCI / PCL PCIe

den folgenden Richtlinien, Normen und Sicherheitsregeln entspricht:

89/336/EWG EMV-Richtlinie

EN 50081-1 Störaussendung

- EN 55022
- EN 55103-1

EN 50082-1 Störfestigkeit

- EN 55024
- EN 55103-2

B Funktionsbeschreibung

B1 Übersicht

Typen

PCL PCI / PCL PCIe sind Timecode-Leser-Boards für PCs mit PCI-Bus. Es ist eine einfache, schnelle, professionelle und technisch ausgereifte Möglichkeit, Timecode in Applikation zu integrieren. Das Board verfügt über ein eigenes Prozessorsystem mit einem Registersatz zum Datentransfer. Die zeitkritischen Timecode-Routinen sind so vollständig von der PC-CPU entkoppelt.

Die Reihe der AV-PCL PCI / PCL PCIe-Boards besteht grundsätzlich aus den folgenden Typen:

PCL PCI L	High-Speed LTC-Leser, universeller 3,3V / 5V PCI Bus
PCL PCI LV	High-Speed LTC-Leser und VITC-Leser für FBAS-Video, universeller 3,3V / 5V PCI Bus
PCL PCI D	High-Speed LTC-Leser und DVITC-Leser für SD-Video, universeller 3,3V / 5V PCI Bus
PCL PCI HD	High-Speed LTC-Leser, DVITC- und ATC-Leser für HD/SD-Video, universeller 3,3V / 5V PCI Bus
PCL PCIe L	High-Speed LTC-Leser, PCIe Bus
PCL PCIe LV	High-Speed LTC-Leser und VITC-Leser für FBAS-Video, PCIe Bus
PCL PCIe D	High-Speed LTC-Leser und DVITC-Leser für SD-Video, PCIe Bus
PCL PCIe HD	High-Speed LTC-Leser, DVITC- und ATC-Leser für HD/SD-Video, PCIe Bus
PCL PCIe 3G	High-Speed LTC-Leser, DVITC- und ATC-Leser für 3G/HD/SD-Video, PCIe Bus

Weitere Varianten sind nach Absprache möglich.

LTC-Leser

Leser für LTC Timecode nach der Spezifikation ANSI/SMPTE 12M-1995. Lesebereich 1 bis 2500 Frames pro Sekunde (fps), vorwärts und rückwärts. Gleichzeitige Dekodierung der Time- und User-Informationen. Eingang 100mV bis 5Vpp, symmetrisch oder asymmetrisch, 2 x Cinch-Steckverbindung. Automatische Erkennung der Framerate (abschaltbar).

VITC-Leser

Leser für VITC Timecode nach der Spezifikation ANSI/SMPTE 12M-1995. Lesebereich von Standbild bis Search-Geschwindigkeit mit Time- und User-Information. Automatische Pegelanpassung für VITC-Leser. 1 x BNC-Steckverbindung mit abschaltbarer Terminierung 75 Ω . VITC-Zeilen werden automatisch erkannt, können aber auch als Einzelzeile, 2-Zeilen oder Zeilenblock bestimmt werden. Automatische Erkennung der Framerate (abschaltbar).

DVITC-Leser

Leser für DVITC Timecode nach der Spezifikation SMPTE 266M-1994. Time- und User-Information werden gleichzeitig ausgelesen. Automatische Levelanpassung für DVITC-Leser möglich. Eingang SD Video (4:2:2 Component, SMPTE 259M-1997), 1 x BNC-Steck-

Verbindung an 75 Ω . DVITC-Zeilen werden automatisch erkannt, können aber auch als Einzelzeile, 2-Zeilen oder Zeilenblock bestimmt werden. Automatische Erkennung der Framerate (abschaltbar).

ATC-Leser

Leser für ATC Timecode (SMPTE RP 188-1999) und HANC Timecode (SMPTE RP 196-1997). Time- und User-Information werden gleichzeitig ausgelesen. Eingang SD Video (4:2:2 Component, SMPTE 259M-1997), HD Video (SMPTE 292M-1998) oder 3G Video (SMPTE 424M-2006), 1 x BNC-Steckverbindung an 75 Ω . Automatische Erkennung der Framerate (abschaltbar).

PC-Interface

32-Byte-Registersatz zur Datenübergabe auf PCI-Interface. Paralleler Betrieb mehrerer PCL PCI / PCL PCIe-Boards ist möglich. Selektierbare Interruptsteuerung. 32 Bit / 33 MHz PCI-Karte.

Treiberunterstützung

Treiber und Beispielprogramme für Windows 2000, XP, Server 2003, Vista, 7 und Server 2008, jeweils in der 32-Bit- und der 64-Bit-Version, werden mitgeliefert. DLL-Funktionen zum Lesen der Timecodewerte und zur Konfiguration stehen zur Verfügung. Unterstützt werden C/C++, Delphi und Visual Basic.

Für Linux wird ein Treiber im Source-Code mitgeliefert.

B2 Timecode Funktionen

Timecode Register

Die Datenübernahme vom PCL PCI / PCL PCIe Board erfolgt über Register (Kommando 0x43: `pclGetRegister`).

Das LTC-Register enthält die Daten des LTC-Lesers. Sobald plausible Daten (Prüfung der Zeitdaten) gelesen werden, wird überprüft, ob LTC in „Vorwärtsrichtung“, in „Rückwärtsrichtung“ oder als „stehender“ LTC vorliegt. Bei einem „stehenden“ LTC werden gleiche Zeitwerte gelesen, die Daten werden direkt in das Register übertragen. In den anderen Fällen werden die Zeitdaten um 1 Frame addiert (bei „Vorwärtsrichtung“) bzw. um 1 Frame subtrahiert (bei „Rückwärtsrichtung“) und dann in das Register übertragen. Die Richtung wird in einem Bit gekennzeichnet.

Das VITC-Register enthält die Daten des VITC-Lesers bzw. DVITC-Lesers. Sind die gelesenen Daten plausibel (Prüfung der Zeitdaten), werden sie in das Register übertragen. Dies erfolgt mit jedem Halbbild, das erste bzw. zweite Halbbild wird durch ein Flagbit gekennzeichnet.

Das Mixed-Register kann LTC oder VITC/DVITC Daten enthalten, gesteuert durch die beiden Kommandos 0x20 (`pclMixedEnable`) und 0x21 (`pclPriority`). Kommando 0x20 ermöglicht generell die Datenübertragung in das Mixed-Register, Kommando 0x21 regelt die Priorität, wenn LTC und VITC/DVITC gleichzeitig gelesen werden und beide Daten in das Mixed-Register übertragen werden sollen.

Der ATC-Leser empfängt Datenpakete und überprüft die Art des Timecodes, um dann die dekodierten Informationen in die speziellen Register zu übertragen. Es gibt Register für ATC LTC, ATC VITC, HANC LTC, HANC VITC. Diese Timecodes können nicht in das Mixed-Register übertragen werden.

Framerate

Eine Automatik ermittelt die Framerate zu 24, 25, 30 (ohne Drop) oder 30-Drop, getrennt für jeden Timecode (LTC, VITC, ATC, HANC). Beim Einschalten ist die Framerate auf 25 gesetzt. Die Automatik kann ausgeschaltet werden, indem die Framerate explizit gesetzt wird. Kommando 0x30 (pclTcFrames) bewirkt dies für alle Timecodes gleichzeitig, Kommando 0x31 nur für LTC (als Audiosignal oder als ATC_LTC oder HANC_LTC) und Kommando 0x32 nur für alle Videotimecodes: VITC/DVITC, ATC_VITC und HANC_VITC.

Fehlerüberprüfung

Die Timecode-Leser überprüfen die gelesenen Zeitdaten auf Plausibilität und auf korrekte, lückenlos zeitliche Reihenfolge. Sind die Daten plausibel, werden sie in die Register übertragen und als neue Werte gemeldet. Fehler werden in einem Fehlerzähler gekennzeichnet. Dies ermöglicht die Überprüfung einer kontinuierlich aufwärts zählenden Timecode-Quelle auf Fehlerfreiheit.

Der Fehlerzähler für den LTC wird mit dem Kommando 0x17 (pclLtcError) ausgelesen. Fehler der Zeitdaten durch nicht plausible Daten oder wegen falscher zeitlicher Reihenfolge (± 1 Frame) werden gezählt.

Der Fehlerzähler für den VITC/DVITC wird mit dem Kommando 0x18 (pclVitcError) ausgelesen. Fehler der Zeitdaten durch nicht plausible Daten oder wegen falscher zeitlicher Reihenfolge (+ 1 Frame Überprüfung in jedem 1. Halbbild, Gleichheit der Zeitwerte im 2. Halbbild) werden gezählt. Ist die automatische Levelanpassung eingeschaltet (siehe Abschnitte unten), werden durch Aktivierung der Automatik nach jedem Videoausfall zusätzliche Fehler erzeugt.

Eine Fehlerüberprüfung in Verbindung mit der Schwungrad-Funktion (siehe nächsten Abschnitt) ist bezüglich der Überprüfung auf korrekte Reihenfolge der Zeitwerte nicht anwendbar.

Die Fehlerzähler für die ATC Timecodes werden mit den Kommandos 0x22 (pclAtcLtcError), 0x23 (pclAtcVitcError), 0x25 (pclHancLtcError) und 0x26 (pclHancVitcError), ausgelesen. Fehler der Zeitdaten durch nicht plausible Daten oder wegen falscher zeitlicher Reihenfolge (bei LTC: + 1 Frame; bei VITC: von Halbbild zu Halbbild identische Zeit oder + 1) werden gezählt.

Je nach Anwendung können die Kommandos in folgenden Varianten benutzt werden:

- Mit CMDD0 = \$00 wird der aktuelle Fehlerzähler angefragt und automatisch neu initialisiert. Nach diesem Kommando ist der Zähler auf 0, und ein erster neu gelesener Timecode führt nicht zu einem Fehler. Dieses Kommando sollte zu Beginn einer neuen Überprüfung gesendet werden.
- Mit CMDD0 = \$01 wird der aktuelle Fehlerzähler angefragt und nicht weiter beeinflusst. Der Fehlerzähler zählt bis 255 und bleibt bei weiteren Fehlern stehen. Bei einer Anwendung, in der die Fehlerzahl nicht unbedingt interessant ist, sondern nur die Verifikation auf Fehlerfreiheit, ist dieses Kommando für die laufende Überprüfung geeignet.

- Mit CMDD0 = \$02 wird der aktuelle Fehlerzähler angefragt und anschließend automatisch auf 0 gestellt. Wird eine größere Fehlerzahl erwartet und die Anzahl der Fehler soll ausgewertet werden, kann die Anwendersoftware dieses Kommando senden und einen eigenen Fehlerzähler verwalten.

Schwungrad

Um eine Störung in einer kontinuierlich aufwärts zählenden Timecode-Quelle überbrücken zu können, kann die „Schwungrad“-Funktion je für LTC und VITC/DVITC aktiviert werden (Kommando 0x19 pclFlyWheel). Für die ATC Timecodes steht diese Funktion nicht zur Verfügung. Beim Einschalten ist diese Funktion nicht aktiviert. Fällt eine vorher fehlerfreie Quelle aus, so wird mit dieser Funktion im Frame-Rhythmus die Zeit automatisch weitergezählt und in das Timecode-Register geschrieben. Ein Flagbit in dem Timecode-Register zeigt an, dass die neuen Daten über die Schwungrad-Funktion erzeugt wurden. Das Schwungrad ist keine präzise Uhr, sondern dient zur Überbrückung einiger Sekunden.

Für den LTC kann sich das Schwungrad auf die LTC-Frequenz im Bereich von 15 - 60 Frames/Sekunde synchronisieren. Die Genauigkeit beträgt maximal ± 1 Frame pro Minute.

Für den VITC/DVITC liefert das Schwungrad nur Werte entsprechend dem ersten Halbbild. Je nach Framerate wird auf das zutreffende Fernsehsystem geschlossen: 25 = 625/50 (PAL), 30 = 525/60 (NTSC). Die Genauigkeit ist besser als 1 Frame je 10 Minuten.

VITC Levelanpassung

Nach dem Einschalten ist die automatische Levelanpassung eingeschaltet. Der VITC-Leser analysiert den VITC-Pegel und stellt dann einen optimalen Wert für die Datenabtrennung ein. Daher ist ein sicheres Lesen auch über einen weiten Bereich des Videopegels möglich. Ist der VITC zwischendurch gestört, wird die Automatik erneut aktiviert. Bis der optimale Wert für die Datenabtrennung gefunden wird, können einige Sekunden vergehen. Bei einem schlechten Video (z.B. VHS-Rekorder im Jog/Search-Modus) kann die Automatik auf der einen Seite dafür sorgen, dass weiterhin VITC gelesen wird, auf der anderen Seite können aber auch zusätzliche Lesefehler erzeugt werden. Daher muss in der Praxis für die konkrete Anwendung getestet werden, ob mit oder ohne Automatik gelesen werden soll. Mit dem Kommando 0x16 (pclVitcLevelControl) kann die Automatik an/ausgeschaltet werden, zudem kann der aktuelle Wert der Datenabtrennung abgefragt und auch geändert werden. Der Wertebereich beträgt 0x00 bis 0x5A, dies entspricht ca. einem Spannungsbereich von -100mV bis +1400mV (0V = Schwarzscherle des Videosignals). Der Anfangswert ist 0x16 (= ca. 270mV).

DVITC Levelanpassung

Wie für den analogen VITC kann auch für den DVITC der optimale Wert für die Datenabtrennung automatisch ermittelt werden. Nach dem Einschalten ist die Automatik allerdings ausgeschaltet, und ein mittlerer Wert für die Datenabtrennung ist voreingestellt. Mit dem Kommando 0x16 (pclVitcLevelControl) kann die Automatik an/ausgeschaltet werden, zudem kann der aktuelle Wert der Datenabtrennung abgefragt und auch geändert werden. Der Wertebereich beträgt 0x00 bis 0x5A, dies wird umgerechnet zu einem Wertebereich im digitalen Video von 0x10 - 0xC4. Der Anfangswert ist 0x32 (= 0x74).

B3 Anschlüsse und Technische Daten

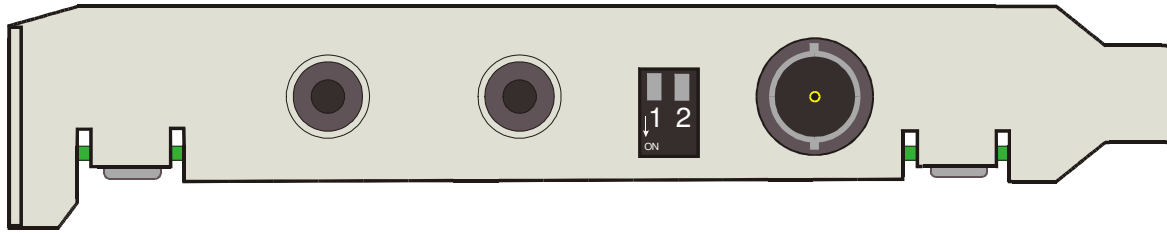
Eingänge

Eingang	Steckverbindung	Signalbeschreibung
LTC-1 LTC-2 (PCI Karten)	2 x Cinch-Buchse	LTC-Eingang, symmetrisch oder asymmetrisch, 100 mV _{pp} bis 5 V _{pp} , Lesebereich 1 bis 2500 Frames/Sek.
LTC / GPI (PCIe Karten)	RJ45 Buchse	LTC-Eingang, symmetrisch oder asymmetrisch, 100 mV _{pp} bis 5 V _{pp} , Lesebereich 1 bis 2500 Frames/Sek.
Video Input (PCI LV PCIe LV)	BNC, 75 Ω (IEC 169-8)	Composite-Video-Eingang mit abschaltbarer 75 Ω Terminierung. Fernsehsystem 625/50 (PAL) oder 525/60 (NTSC). DC Offset: $\pm 12V$, Sync Amplitude = 300mV ± 6dB (150mV-600mV). VITC Datenpegel: V_{\min} („1“) = 200mV, V_{\max} („0“) = 500mV, V_{1-0} („1“ - „0“) ≥ 200mV.
Video Input (PCI D PCIe D)	BNC, 75 Ω (IEC 169-8)	SD Video Eingang, 8/10 Bit nach SMPTE 259M-C: 270 Mb/s, 525/625 Komponenten.
Video Input (PCI HD PCIe HD)	BNC, 75 Ω (IEC 169-8)	SD Video Eingang, 8/10 Bit nach SMPTE 259M-C: 270 Mb/s, 525/625 Komponenten. HD Video Eingang, 8/10 Bit nach SMPTE 292M-1998: 1.485 Gb/s.
Video Input (PCIe 3G)	BNC, 75 Ω (IEC 169-8)	SD Video Eingang, 8/10 Bit nach SMPTE 259M-C: 270 Mb/s, 525/625 Komponenten. HD Video Eingang, 8/10 Bit nach SMPTE 292M-1998: 1.485 Gb/s. 3G Video Eingang, 8/10 Bit nach SMPTE 424M-2006: 2.970 Gb/s.

Funktionsbeschreibung PCL PCI / PCL PCIe

Seite 6

PCL PCI



LTC-1 LTC-2 SW1 SW2 Video Input

Schalter

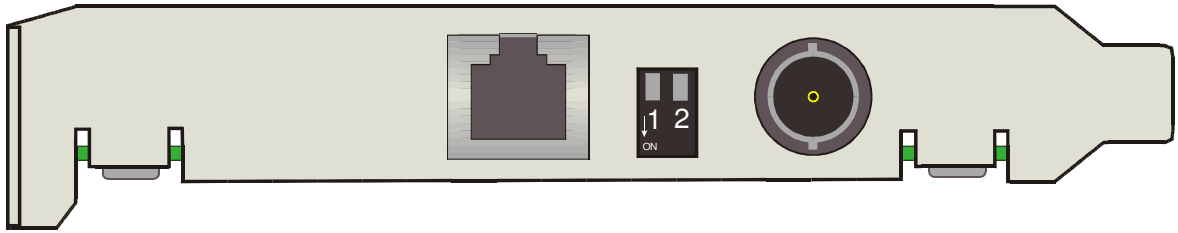
SW1	LTC Input
ON	Asymmetrischer LTC Input an LTC-2
OFF	Symmetrischer LTC Input an LTC-1 und LTC-2

SW2	PCI LV: Video Terminierung	PCL PCI D: Equalization	PCL PCI HD:
ON	75 Ohm Terminierung	Equalization automatisch für bis zu 100m Anschlusskabel (Bel- den 8281, PSF 1/3 oder gleich- wertig), 270 Mb/s.	n/a
OFF	ohne Terminierung	Equalization ausgeschaltet, für kurzes Anschlusskabel (<10m).	n/a

Technische Daten

Abmessungen über alles (Länge x Höhe x Breite)	141 x 120 x 22 mm
Gewicht	ca. 110 g
Betriebsspannung	3,3 VDC und 5 VDC
Stromverbrauch	2 W
Zulässige Umgebungstemperatur	5 – 40 °C
Relative Luftfeuchte	35 – 85 %

PCL PCIe



LTC / GPI SW1 SW2 Video Input

LTC / GPI

Pin	Signal
3	LTC Input LTC-1
6	LTC Input LTC-2
4	LTC GND
5	

Pin	Signal
1	GPI 1 GND
2	GPI 1 I/O
7	GPI 2 GND
8	GPI 2 I/O

Schalter

SW1	LTC Input
ON	Asymmetrischer LTC an LTC-2
OFF	Symmetrischer LTC an LTC-1 und LTC-2

SW2	PCL PCIe LV: Video Terminierung	PCL PCIe 3G PCL PCIe HD PCL PCIe D:
ON	75 Ohm Terminierung	n/a
OFF	ohne Terminierung	n/a

Technische Daten

Abmessungen über alles (Länge x Höhe x Breite)	PCL PCIe 3G: 150 x 120 x 22 mm (Standard Profile Slotblech) 150 x 80 x 22 mm (Low Profile Slotblech) PCL PCIe L / LV / D / HD: 191 x 120 x 22 mm (Standard Profile Slotblech) 191 x 80 x 22 mm (Low Profile Slotblech)
Gewicht	ca. 80 g
Betriebsspannung	3,3 VDC und 12 VDC
Stromverbrauch	3 W
Zulässige Umgebungstemperatur	5 – 40 °C
Relative Luftfeuchte	35 – 85 %

B4 Installation

Zur Installation in einem Windows-PC gehen Sie bitte wie folgt vor:

- Melden Sie sich als Administrator an und starten Sie von der mitgelieferten CD-ROM das Skript „\Driver\PCI\Install Driver.bat“. Damit wird ein Device Driver für die PCL PCI / PCL PCIe Boards installiert.
- Stellen Sie die DIP-Schalter auf die Eingangssignale des PCL PCI / PCL PCIe Boards nach dem vorherigen Abschnitt ein. Sie können die Einstellung später im eingebauten Zustand der Karte noch ändern.
- Fahren Sie den PC herunter und schalten Sie ihn aus.
- Bauen Sie das PCL PCI / PCL PCIe Board in einen freien PCI- bzw. PCIe-Steckplatz ein. Achten Sie auf guten Sitz im PCI-Slot und schrauben Sie sie fest.
- Schalten Sie den PC an und fahren Sie ihn hoch. Melden Sie sich als Administrator an.
- Windows erkennt die neue Steckkarte automatisch. Wenn Sie nach einem Treiber gefragt werden, legen Sie die mitgelieferte CD-ROM ein und geben Sie sie als Quelle für den Treiber an. Sie können auch die jeweils aktuelle Version des Treibers von <http://www.alpermann-velte.com> herunterladen (im Menü unter Service), ihn auspacken und als Quelle für die Treiberdateien angeben.
- Die Treiberinstallation ist damit abgeschlossen. Sie können die Funktion des PCL PCI / PCL PCIe Boards mit den Programmen PclTest.exe und IntTest.exe testen.

B5 Firmware Update

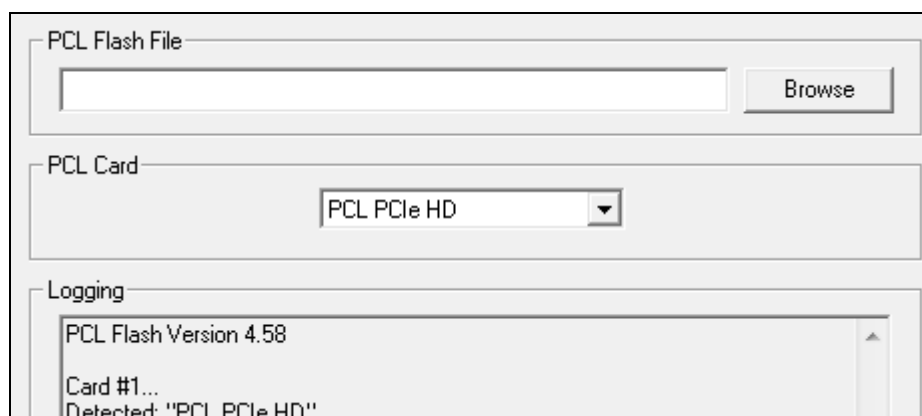
Die PCL PCIe-Karten und die PCL PCI HD verfügen über einen Flash-Speicher und können einen Firmware Update erhalten. Das ist in diesem Kapitel beschrieben. Bei den Karten PCL PCI LV, PCL PCI L, PCL PCI V und PCL PCI TS erfordert das einen Chip-Tausch. In diesem Fall setzen Sie sich bitte mit Alpermann+Velte in Verbindung.

Für einen Firmware Update wird ein PC mit Windows Betriebssystem und das PC Programm **PCLFlash** benötigt. Die neueste Programmversion steht zum Download bereit unter: <http://www.alpermann-velte.com/service/software/software.html>.

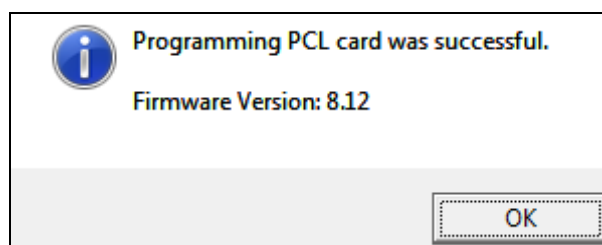
Speichern Sie die neue Firmware als **.tcf** Datei auf der lokalen Festplatte Ihres Computers.

Im Detail sind dann folgende Schritte durchzuführen:

1. Den PC mit gesteckter PCL-Karte starten, als Administrator anmelden den aktuellen Treiber installieren (wenn nicht schon geschehen).
2. Das Programm **PCLFlash** auf dem Computer starten. Das Programm durchsucht den PC nach PCL-Karten und listet alle gefundenen Geräte auf. Aus der Drop-Down-Liste die PCL-Karte wählen, in die Sie das Update flashen möchten.



3. Mit „Browse“ die **.tcf**-Datei öffnen. Ein Klick auf „Flash!“ startet das Update. Das Programm überprüft, ob die Firmware für diesen Typ des Gerätes bestimmt ist. Besteht keine Übereinstimmung, erscheint die Fehlermeldung: „Incompatible Flash Update File“. Update startet automatisch, wenn kein Fehler vorliegt. Nach erfolgreichem Update „OK“ drücken.



4. Das Update ist fertig. Sie können nur weitere Karten flashen oder **PCLFlash** schließen.

C Programmierung

C1 Windows-Treiber AVPCL32

Struktur

Der Treiber für PCL PCI / PCL PCIe bestehen aus zwei Teilen:

- Die Datei AvPcl32.dll stellt die Schnittstelle zum Anwenderprogramm dar. Sie kennt Funktionen wie `pclOpen()`, `pclClose()`, `pclGetTc()`, usw.
- Avpcl6.sys ist der Gerätetreiber. Er wickelt den Zugriff auf den PCI-Bus ab.

Treiber-Dateien

Die folgenden Verzeichnisse und Dateien befinden sich in der Datei „avpci.zip“:

Hauptverzeichnis

Im Hauptverzeichnis befinden Testprogramme, mit denen die Funktion des PCL PCI / PCL PCIe Boards und des korrekte Installation des Treibers überprüft werden kann.

- **PclTest.exe** ist eine einfache WIN32-Konsolen-Anwendung, die den PCL PCI / PCL PCIe Timecode anzeigt. Der Quelltext befindet sich im SDK im Verzeichnis „Samples“.
- **InfTest.exe** ist eine einfache WIN32-Konsolen-Anwendung, die den PCL PCI / PCL PCIe Timecode anzeigt. Im Unterschied zu PclTest benutzt sie Interrupts. Der Quelltext befindet sich im SDK im Verzeichnis „Samples“.

Unterverzeichnis „Driver“

Im diesem Verzeichnis und seinen Unterverzeichnissen befinden sich alle Dateien, die für die Installation von PCL PCI / PCL PCIe auf einem PC mit Windows-Betriebssystem benötigt werden.

- **Avpcl6.inf** ist die Beschreibung des Gerätetreibers für die Installation.
- **Avpcl6.cat** enthält eine digitale Signatur für die Datei avpcl6.inf auf 32-Bit-Windows-Betriebssystemen.
- **Avpcl6.ntamd64.cat** enthält eine digitale Signatur für die Datei avpcl6.inf auf 64-Bit-Windows-Betriebssystemen.
- **Install Driver.bat** ist ein Skript für die Installation des Gerätetreibers. Es prüft, ob es auf einem 32-Bit- oder 64-Bit- Windows-Betriebssysteme läuft und ruft abhängig vom Ergebnis eines der Skripts im amd64- oder i386-Unterverzeichnis auf.
- **Pcl-Pci.inf** ist die Beschreibung des PCL PCI / PCL PCIe-Treibers für die Installation.
- **Pcl-pci.cat** enthält eine digitale Signatur für die Datei pcl-pci.inf auf 32-Bit-Windows-Betriebssystemen.

- **Pcl-pci.ntamd64.cat** enthält eine digitale Signatur für die Datei pcl-pci.inf auf 64-Bit-Windows-Betriebssystemen.
- **Update Driver.bat** ist ein Skript für die Aktualisierung des Treibers. Es prüft, ob es auf einem 32-Bit- oder 64-Bit- Windows-Betriebssysteme läuft und ruft abhängig vom Ergebnis eines der Skripts im amd64- oder i386-Unterverzeichnis auf.

Unterverzeichnis „Driver \ i386“

Dieses Unterverzeichnis enthält die Bestandteile des Treibers für 32-Bit-Windows-Betriebssysteme.

- **AvPcl32.dll** ist die Schnittstelle vom Anwenderprogramm zum Low-Level-Treiber avpcl6.sys. Diese Datei wird während der Installation in das Verzeichnis \Windows\system32 kopiert.
- **Avpcl6.sys** ist der Gerätetreiber. Diese Datei wird während der Installation in das Verzeichnis \Windows\system32\driver kopiert.
- **Difxapi.dll** wird von wdreg_gui.exe und wdreg.exe benötigt.
- **Install Driver.bat** ist ein Skript für die Installation des Gerätetreibers. Es ruft wdreg_gui.exe mit passenden Parametern auf.
- **Update Driver.bat** ist ein Skript für die Aktualisierung des Treibers. Es ruft wdreg_gui.exe mit passenden Parametern auf.
- **Wdreg_gui.exe** ist ein Programm, mit dem Gerätetreibers installiert werden können.
- **Wdreg.exe** ist die Kommandozeilen-Version von wdreg_gui.exe.

Unterverzeichnis „Driver \ amd64“

Dieses Unterverzeichnis enthält die Bestandteile des Treibers für 64-Bit-Windows-Betriebssysteme.

- **AvPcl32.dll** ist die Schnittstelle vom Anwenderprogramm zum Low-Level-Treiber avpcl6.sys für 64-Bit-Anwenderprogramme. Diese Datei wird während der Installation in das Verzeichnis \Windows\system32 kopiert.
- **AvPcl64.dll** ist die Schnittstelle vom Anwenderprogramm zum Low-Level-Treiber avpcl6.sys für 32-Bit-Anwenderprogramme unter Nutzung des WoW64 Subsystems (Windows on Windows 64 Bit). Diese Datei wird während der Installation in das Verzeichnis \Windows\sysWow64 kopiert und dabei in avpcl32.dll umbenannt.
- **Avpcl6.sys** ist der Gerätetreiber. Diese Datei wird während der Installation in das Verzeichnis \Windows\system32\driver kopiert.
- **Install Driver.bat** ist ein Skript für die Installation des Gerätetreibers. Es ruft wdreg_gui.exe mit passenden Parametern auf.
- **Difxapi.dll** wird von wdreg_gui.exe und wdreg.exe benötigt.
- **Update Driver.bat** ist ein Skript für die Aktualisierung des Treibers. Es ruft wdreg_gui.exe mit passenden Parametern auf.
- **Wdreg_gui.exe** ist ein Programm, mit dem Gerätetreibers installiert werden können.
- **Wdreg.exe** ist die Kommandozeilen-Version von wdreg_gui.exe.

SDK-Dateien

Das SDK befindet sich getrennt von den Treiberdateien in der Datei „avpcisdsk.zip“. Es enthält die benötigten Dateien, um PCL PCI / PCL PCIe-Anwenderprogramme zu schreiben.

Hauptverzeichnis

- **History.txt** enthält einen Überblick über die Entstehungsgeschichte des Treibers.

Unterverzeichnis „Lib“

Dieses Unterverzeichnis enthält gemeinsam genutzte Bibliotheken.

- **AvPcl.h** enthält alle Deklarationen der Bibliothek AvPcl32.dll. Die Datei ist ausführlich kommentiert und dient so auch als Referenz für den Anwendungsprogrammierer.
- **AvPcl32.def** ist die Moduldefinitionsdatei zu AvPcl32.dll.

Unterverzeichnis „Lib \ i386“

Dieses Unterverzeichnis enthält Bibliotheken für 32-Bit-Anwenderprogramme.

AvPcl32.lib ist die Importbibliothek zu AvPcl32.dll in der 32-Bit-Version.

Unterverzeichnis „Lib \ amd64“

Dieses Unterverzeichnis enthält Bibliotheken für 64-Bit-Anwenderprogramme.

AvPcl32.lib ist die Importbibliothek zu AvPcl32.dll in der 64-Bit-Version.

Unterverzeichnis „Samples“

Dieses Verzeichnis enthält Quelltexte von einfachen PCL PCI / PCL PCIe Testprogrammen. Sie sind als WIN32-Konsolenanwendung geschrieben und zeigen die grundsätzliche Benutzung von PCL PCI / PCL PCIe.

- **AvPcl.h** ist ein Verweis auf die Datei AvPcl.h im Verzeichnis Lib.
- **PclTest.cpp** ist der Quelltext des Testprogramms PclTest.exe. Es zeigt, wie die PCL PCI / PCL PCIe Boards im Polling-Modus angesprochen werden.
- **IntTest.cpp** ist der Quelltext des Testprogramms PclTest.exe. Es zeigt, wie die PCL PCI / PCL PCIe Boards im Interrupt-Modus angesprochen werden.

Unterverzeichnis „Samples \ msdev_5“

Dieses Verzeichnis enthält Arbeitsbereich- und Projektdateien für Microsoft Visual C++ 5.0.

- **Samples.dsw** enthält den Arbeitsbereich für beide Beispielprogramme.
- **PclTest \ PclTest.dsp** ist die Projektdatei für das PclTest Beispielprogramm.
- **IntTest \ IntTest.dsp** ist die Projektdatei für das IntTest Beispielprogramm.

Unterverzeichnis „Samples \ msdev_2008“

Dieses Verzeichnis enthält Solution- und Projektdateien für Microsoft Visual Studio 2008.

- **Samples.sln** ist die Solution-Datei für beide Beispielprogramme. Sie ist für 32-Bit- und 64-Bit-Plattformen vorbereitet.
- **PclTest \ PclTest.vcproj** ist die Projektdatei für das PclTest Beispielprogramm. Sie ist für 32-Bit- und 64-Bit-Plattformen vorbereitet.
- **IntTest \ IntTest.vcproj** ist die Projektdatei für das IntTest Beispielprogramm. Sie ist für 32-Bit- und 64-Bit-Plattformen vorbereitet.

Unterverzeichnis „Pascal“

Dieses Verzeichnis enthält die benötigten Dateien, um mit Turbo/Borland-Pascal bzw. Delphi auf PCL PCI / PCL PCIe zuzugreifen.

- **AvPcl.pas** enthält die Deklarationen für den Zugriff auf AvPcl32.dll.
- **PclTest.pas** ist ein einfaches Testprogramm als WIN32 Konsolenanwendung.

Unterverzeichnis „Basic“

Dieses Verzeichnis enthält die Informationen, um PCL PCI / PCL PCIe mit Visual Basic zu benutzen.

- **AvPcl32.vb** enthält alle Deklarationen für 32-Bit Visual-Basic-Programme.

Unterverzeichnis „Linux“

Dieses Verzeichnis enthält den Linux-Treiber im Quelltext. Bitte lesen Sie die Datei „README“ für weitere Informationen.

C2 Datenzugriff

Dieser Abschnitt beschreibt den Zugriff auf PCL PCI / PCL PCIe auf Hardware-Ebene. Soweit eines der unterstützten Betriebssysteme zum Einsatz kommt, kann auf diese Informationen verzichtet werden, da der Zugriff mit Hilfe des mitgelieferten Treibers abgewickelt wird, so wie im Abschnitt C1 beschrieben.

Speicher- bzw. I/O-Zugriff

PCL PCI / PCL PCIe blendet in den Speicher- und I/O-Adressraum drei Blöcke ein:

Block	Speicher oder I/O	Größe	Beschreibung
BAR0	Speicher	PCI: 0x80, PCIe: 0x100	Lokale Konfigurationsregister
BAR1	I/O	PCI: 0x80, PCIe: 0x100	Lokale Konfigurationsregister
BAR2	Speicher	0x100	PCL PCI / PCL PCIe-Registersatz

Die lokalen Konfigurationsregister dienen zur Konfiguration des verwendeten PCI-Interface-Chips. Sie sollten nicht verändert werden.

Der PCL PCI-Registersatz ist nur im Speicher-Adressraum vorhanden.

Registersatz

Von jedem der insgesamt 64 Langwörter ist nur das unterste Byte belegt. Dadurch ergibt sich, dass nur Byte-Zugriffe auf Adressen möglich sind, die durch vier teilbar sind. Von diesen 64 möglichen Bytes sind derzeit 21 belegt.

Funktionsbeschreibung PCL PCI / PCL PCIe

Der PCL PCI / PCL PCIe-Registersatz ist wie folgt definiert:

PCL PCI / PCL PCIe		Register	Beschreibung
Offset	r/w		
0x00	r	DATA0	Daten von PCL PCI / PCL PCIe
0x04	r	DATA1	"
0x08	r	DATA2	"
0x0C	r	DATA3	"
0x10	r	DATA4	"
0x14	r	DATA5	"
0x18	r	DATA6	"
0x1C	r	DATA7	"
0x20	r	DATA8	"
0x24	r	DATA9	"
0x28	r	DATAA	"
0x2C	r	INTFLGL	Interrupt-Flags L
0x30	r	CMDR	Kommando-Antwort
0x34	r	ACK	Bestätigung von PCL
0x38	w	CMD	Kommando an PCL
0x3C	w	CMDD0	Daten an PCL PCI / PCL PCIe
0x40	w	CMDD1	"
0x44	w	CFG	PCL Konfiguration
0x48	r	INTFLGH	Interrupt-Flags H
0x4C	w	INTACK	Interrupt-Bestätigung
0x7C	r	VERSION	PCL PCI / PCL PCIe Chip-Version

Jedes der Register erlaubt nur den Datentransfer in einer Richtung, also entweder lesend oder schreibend. Ein Nur-Lese-Register darf also nicht beschrieben werden, umgekehrt kann und darf von einem Nur-Schreib-Register nicht gelesen werden.

Register

Die Register im Einzelnen:

DATA0 bis DATAA

Über diese Register liefern Kommandos Daten zurück.

INTFLGL

Interrupt Flags: In diesem Register kann abgelesen werden, für welche der möglichen Interrupt Quellen von PCL PCI / PCL PCIe Interrupt Anforderungen vorliegen (siehe auch INTFLGH). Die Bits sind wie folgt kodiert:

Bit	Hex	Interrupt Anforderung
0	0x01	Mixed-Register
2	0x04	LTC-Register
3	0x08	VITC-Register
6	0x40	Ungerade Halbbilder (1. Halbbild)
7	0x80	Gerade Halbbilder (2. Halbbild)

Ein gesetztes Bit 2 bedeutet beispielsweise, dass ein neuer LTC gelesen wurde. Es ist unerheblich, ob die Interrupt Anforderung maskiert wurde oder nicht (siehe Kommando pclIntMask), d.h. ob sie einen Hardware Interrupt ausgelöst hat oder nicht. Diese Register kann also auch im Polling-Betrieb genutzt werden.

Interrupt Anforderungen müssen zurückgesetzt werden. Das geschieht für die Mixed-, LTC- und VITC-Bits über das Lesen des entsprechenden Registers. Die restlichen Bits können über das Kommando pclIntAck zurückgesetzt werden.

CMDR

Command Response: In diesem Register schreibt PCL PCI / PCL PCIe nach der Abarbeitung eines Kommandos die Kommandonummer. Damit kann überprüft werden, ob das zuletzt abgesetzte Kommando korrekt ausgeführt wurde. Über die Kommandonummer können auch die Daten in DATA0 bis DATAA interpretiert werden.

Bei einem unbekanntem Kommando wird 0xFF in dieses Register geschrieben.

INTFLGH (nur PCL PCI / PCL PCIe HD und 3G)

Interrupt Flags: In diesem Register kann abgelesen werden, für welche der möglichen Interrupt Quellen von PCL PCI / PCL PCIe Interrupt Anforderungen vorliegen (siehe auch Beschreibung zu INTFLGL). Die Bits sind wie folgt kodiert:

Bit	Hex	Interrupt Anforderung
0	0x01	ATC LTC
1	0x02	ATC VITC
3	0x08	HANC LTC
4	0x10	HANC VITC

ACK

Acknowledge: Nach erfolgreicher Ausführung eines Kommandos wird dieses Register inkrementiert.

CMD

Command: Um ein Kommando an PCL PCI / PCL PCIe zu senden, wird die Kommandonummer in dieses Register geschrieben.

CMDD0 und CMDD1

Command Data: Über diese Register können Kommandodaten an PCL PCI / PCL PCIe übergeben werden.

CFG

Dieses Register dient Diagnosezwecken. Es sollte nicht beschrieben werden.

INTACK

Interrupt Acknowledge: Durch einen Schreibzugriff auf dieses Register wird ein Hardware Interrupt von PCL PCI / PCL PCIe zurückgesetzt. Zusätzlich muss die Interrupt Anforderung über ein geeignetes Kommando (z.B. pclIntAck) beantwortet werden.

VERSION

Die Chip-Version von PCL PCI / PCL PCIe kann in diesem Register abgefragt werden. Im High-Nibble ist der Chip-Typ kodiert (derzeit 0xA), im Low-Nibble die Versionsnummer (derzeit 0x2). Der Chip-Typ gibt Auskunft über die Ausführung von PCL PCI / PCL PCIe, d.h. ob

Funktionsbeschreibung PCL PCI / PCL PCIe

sie mit LTC- und/oder VITC-Leser bestückt ist. Die Versionsnummer wird in zukünftigen Chip-Versionen von PCL PCI / PCL PCIe inkrementiert.

Kommandos

Fast alle Zugriffe auf PCL PCI / PCL PCIe erfolgen über Kommandos. Das sind Befehle an PCL PCI / PCL PCIe, wie beispielsweise das Setzen einer Voreinstellung oder das Lesen von Timecode. Ein Kommando wird wie folgt an PCL PCI / PCL PCIe geschickt:

- Wenn für das Kommando definiert, schreibe Daten für das Kommando in CMDD0 und CMDD1.
- Lese ACK.
- Schreibe Kommandonummer in CMD.
- Warte, bis sich ACK ändert. Wenn das nicht innerhalb max. 40ms geschieht, Timeout-Fehler zurückmelden.
- Prüfen, ob CMDR gleich der Kommandonummer ist. Wenn nicht, Kommandofehler zurückmelden.
- Wenn für das Kommando definiert, lese Ergebnis des Kommandos aus DATA0 bis DATAA.

Die folgenden Kommandos sind möglich. Die Fabrikwerte, so wie sie nach dem Einschalten gesetzt sind, sind in Klammern angegeben. Wenn nicht explizit unterschiedlich aufgeführt gilt alles, was für VITC beschrieben wird, auch für DVITC. ATC und HANC sind nur bei der PCL PCI / PCL PCIe HD und 3G möglich.

Kommando	Nr.	Parameter	Ergebnis
pclReset	0x01	CMDD0: Betriebsmodus (0x00) <ul style="list-style-type: none">• 0x00: Normaler Modus• 0x80: Hardware reset• 0xF0: Reserviert (Testmodus)	
		PCL PCI / PCL PCIe wird zurückgesetzt. Alle Einstellungen werden auf die Fabrikwerte zurückgesetzt, die nach dem Einschalten gesetzt waren.	

Kommando	Nr.	Parameter	Ergebnis
pclIntMask	0x03	CMDD0: Interrupt-Maske L (0x00) <ul style="list-style-type: none"> • 0x00: Alle Interrupts aus • 0x01: Mixed-Register • 0x04: LTC-Register • 0x08: VITC-Register • 0x40: Ungerade Halbbilder • 0x80: Gerade Halbbilder • 0xFF: Alle Interrupts an CMDD1: Interrupt-Maske H (0x00) (nur PCL PCI / PCL PCIe HD und 3G) <ul style="list-style-type: none"> • 0x00: Alle Interrupts aus • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC • 0xFF: Alle Interrupts an 	DATA0: Interrupt-Maske L <ul style="list-style-type: none"> • 0x01: Mixed-Register • 0x04: LTC-Register • 0x08: VITC-Register • 0x40: Ungerade Halbbilder • 0x80: Gerade Halbbilder DATA1: Interrupt-Maske H (nur PCL PCI / PCL PCIe HD und 3G) <ul style="list-style-type: none"> • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC
		Interrupts werden ein- oder ausgeschaltet. In CMDD0 und CMDD1 (PCL PCI / PCL PCIe HD und 3G) wird eine Bitmaske übergeben, in der jedes gesetzte Bit eine Interrupt Quelle einschaltet. Gelöschte Bits schalten den zugehörigen Interrupt aus. In DATA0 und DATA1 (PCL PCI / PCL PCIe HD und 3G) wird die aktuelle Interrupt-Maske zurückgeliefert.	
pclSetIntMask	0x04	CMDD0: Interrupt-Maske L <ul style="list-style-type: none"> • 0x01: Mixed-Register • 0x04: LTC-Register • 0x08: VITC-Register • 0x40: Ungerade Halbbilder • 0x80: Gerade Halbbilder • 0xFF: Alle Interrupts an CMDD1: Interrupt-Maske H (0x00) (nur PCL PCI / PCL PCIe HD und 3G) <ul style="list-style-type: none"> • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC • 0xFF: Alle Interrupts an 	DATA0: Interrupt-Maske L <ul style="list-style-type: none"> • 0x01: Mixed-Register • 0x04: LTC-Register • 0x08: VITC-Register • 0x40: Ungerade Halbbilder • 0x80: Gerade Halbbilder DATA1: Interrupt-Maske H (nur PCL PCI / PCL PCIe HD und 3G) <ul style="list-style-type: none"> • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC
		Für jedes gesetzt Bit wird der zugehörige Interrupt eingeschaltet. Als Ergebnis wird in DATA0 und DATA1 (PCL PCI / PCL PCIe HD und 3G) die aktuelle Interrupt-Maske zurückgeliefert.	

Funktionsbeschreibung PCL PCI / PCL PCIe

Kommando	Nr.	Parameter	Ergebnis
pclResetIntMask	0x05	CMDD0: Interrupt-Maske L • 0x01: Mixed-Register • 0x04: LTC-Register • 0x08: VITC-Register • 0x40: Ungerade Halbbilder • 0x80: Gerade Halbbilder • 0xFF: Alle Interrupts aus CMDD1: Interrupt-Maske H (0x00) (nur PCL PCI / PCL PCIe HD und 3G) • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC • 0xFF: Alle Interrupts aus	DATA0: Interrupt-Maske L • 0x01: Mixed-Register • 0x04: LTC-Register • 0x08: VITC-Register • 0x40: Ungerade Halbbilder • 0x80: Gerade Halbbilder DATA1: Interrupt-Maske H (nur PCL PCI / PCL PCIe HD und 3G) • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC
		Für jedes gesetzte Bit wird der zugehörige Interrupt ausgeschaltet. Als Ergebnis wird in DATA0 und DATA1 (PCL PCI / PCL PCIe HD und 3G) die aktuelle Interrupt-Maske zurückgeliefert.	
pclReaderEnable	0x10	CMDD0: Leser-Maske (0x06) • 0x00: LTC/VITC Leser aus • 0x02: LTC-Leser an • 0x04: VITC-Leser an CMDD1: Leser-Maske (0x1F) (nur PCL PCI / PCL PCIe HD und 3G) • 0x00: ATC/HANC Leser aus • 0x01: ATC LTC an • 0x02: ATC VITC an • 0x08: HANC LTC an • 0x10: HANC VITC an	
		Für jedes gesetzte Bit wird der zugehörige Leser ein-, für jedes gelöschte Bit ausgeschaltet. Ausgeschaltet bedeutet, dass der Leser keinerlei Werte mehr liefert, also auch über sein eigenes Leseregister nicht mehr abgefragt werden kann.	
pclVitcSetupLines	0x14	CMDD0: Erste Zeile • Bereich: 6 bis 26 CMDD1: Letzte Zeile • Bereich: 6 bis 26 CMDD0 muss kleiner oder gleich CMDD1 sein.	DATA0: Fehlermeldung • 0x00: Kein Fehler • 0x01: Bereichsüberschreitung erste Zeile • 0x02: Bereichsüberschreitung letzte Zeile • 0x03: Erste Zeile > letzte Zeile
		Der Zeilenbereich des VITC-Lesers wird gesetzt. Für PAL sind die Zeilen 6 bis 22 spezifiziert, für NTSC die Zeilen 10 bis 20. Das Ergebnis in DATA0 liefert eine eventuelle Fehlermeldung. Nach dem Einschalten ist der maximale Zeilenbereich gewählt.	

Kommando	Nr.	Parameter	Ergebnis
pclVtcSetupMode	0x15	CMDD0: VITC-Lesermodus (0x01) <ul style="list-style-type: none"> • 0x00: Zwei Zeilen • 0x01: Block 	
		Der Modus des VITC-Lesers wird gesetzt. Es kann aus zwei Zeilen oder aus einem Block von Zeilen gelesen werden. Ist erste Zeile = letzte Zeile (siehe Kommando 0x14), wird nur aus einer Zeile gelesen.	
pclVtcLevelControl	0x16	CMDD0: Modus der VITC/DVITC Levelanpassung <ul style="list-style-type: none"> • 0x00: Automatik einschalten • 0x01: Automatik mit Setzwert in CMDD1 • 0x02: Automatik und Initialisierung wie nach dem Einschalten • 0x03: Automatik ausschalten, der aktuelle Wert bleibt erhalten • 0x04: Automatik ausschalten mit Setzwert in CMDD1 CMDD1: Setzwert der Datenabtrennung (falls CMDD0 = 0x01 oder 0x04), Bereich 0x00 bis 0x5A.	DATA0: Aktueller Wert der Datenabtrennung <ul style="list-style-type: none"> • Bereich: 0x00 bis 0x5A
		VITC/DVITC Levelanpassung (siehe entsprechendes Kapitel für Details): die Automatik kann an/ausgeschaltet werden, zudem kann der Wert für die Datenabtrennung explizit gesetzt werden. Als Ergebnis wird der aktuelle Wert der Datenabtrennung zurückgeliefert. Als Fabrikwert ist die Automatik für den VITC eingeschaltet, für den DVITC ausgeschaltet.	
pclLtcError	0x17	CMDD0: Art der Anfrage <ul style="list-style-type: none"> • 0x00: Anfrage mit anschließender Initialisierung • 0x01: Nur anfragen • 0x02: Anfrage mit anschließendem Rücksetzen des Fehlerzählers 	DATA0: Aktueller Fehlerzähler
		Anfrage des LTC Fehlerzählers. Er erfasst unplausible Zeit und Zeitsprünge.	

Funktionsbeschreibung PCL PCI / PCL PCIe

Kommando	Nr.	Parameter	Ergebnis
pclVtcError	0x18	CMDD0: Art der Anfrage <ul style="list-style-type: none"> • 0x00: Anfrage mit anschließender Initialisierung • 0x01: Nur anfragen • 0x02: Anfrage mit anschließendem Rücksetzen des Fehlerzählers 	DATA0: Aktueller Fehlerzähler
		Anfrage des VITC Fehlerzählers. Er erfasst unplausible Zeit und Zeitsprünge.	
pclFlyWheel	0x19	CMDD0: Schwungradfunktion ein- oder ausschalten (0x00) <ul style="list-style-type: none"> • 0x00: Schwungrad aus • 0x02: LTC-Schwungrad an • 0x04: VITC-Schwungrad an • 0x06: VITC- und LTC-Schwungrad an 	
		Mit der Schwungradfunktion wird ein Timecode-Ausfall überbrückt, siehe entsprechendes Kapitel. Die Funktion ist für LTC und VITC verfügbar, nicht für ATC oder HANC Timecodes.	
pclMixedEnable	0x20	CMDD0: Leser-Maske (0x06) <ul style="list-style-type: none"> • 0x00: Alle Leser ausschalten • 0x02: LTC-Leser an • 0x04: VITC-Leser an 	
		Für jedes gesetzte Bit wird der zugehörige Leser über eine Prioritätssteuerung ins Mixed-Register übernommen, für jedes gelöschte Bit wird er ignoriert, kann aber weiter über sein eigenes Leseregister abgefragt werden. Die Funktion ist für LTC und VITC verfügbar, nicht für ATC oder HANC Timecodes.	
pclPriority	0x21	CMDD0: Priorität (0x01) <ul style="list-style-type: none"> • 0x01: VITC vor LTC • 0x02: LTC vor VITC 	
		Die Priorität bestimmt, welche Leserwerte ins Mixed-Register übernommen werden, wenn auf beiden zugeschalteten Lesern Werte gelesen werden.	
pclAtcLtcError	0x22	CMDD0: Art der Anfrage <ul style="list-style-type: none"> • 0x00: Anfrage mit anschließender Initialisierung • 0x01: Nur anfragen • 0x02: Anfrage mit anschließendem Rücksetzen des Fehlerzählers 	DATA0: Aktueller Fehlerzähler
		Anfrage des ATC LTC Fehlerzählers. Er erfasst unplausible Zeit und Zeitsprünge.	

Kommando	Nr.	Parameter	Ergebnis
pclAtcVtcError	0x18	CMDD0: Art der Anfrage <ul style="list-style-type: none"> • 0x00: Anfrage mit anschließender Initialisierung • 0x01: Nur anfragen • 0x02: Anfrage mit anschließendem Rücksetzen des Fehlerzählers 	DATA0: Aktueller Fehlerzähler
		Anfrage des ATC VITC Fehlerzählers. Er erfasst unplausible Zeit und Zeitsprünge.	
		•	
pclHancLtcError	0x22	CMDD0: Art der Anfrage <ul style="list-style-type: none"> • 0x00: Anfrage mit anschließender Initialisierung • 0x01: Nur anfragen • 0x02: Anfrage mit anschließendem Rücksetzen des Fehlerzählers 	DATA0: Aktueller Fehlerzähler
		Anfrage des HANC LTC Fehlerzählers. Er erfasst unplausible Zeit und Zeitsprünge.	
pclHancVtcError	0x18	CMDD0: Art der Anfrage <ul style="list-style-type: none"> • 0x00: Anfrage mit anschließender Initialisierung • 0x01: Nur anfragen • 0x02: Anfrage mit anschließendem Rücksetzen des Fehlerzählers 	DATA0: Aktueller Fehlerzähler
		Anfrage des HANC VITC Fehlerzählers. Er erfasst unplausible Zeit und Zeitsprünge.	
pclGpi1H:M (Option GPI OUT)	0x28	CMDD0: Stunden CMDD1: Minuten	
		Übergabe der Vergleichszeit für GPI_1. Siehe auch Kommando 0x2A.	
pclGpi1S:F (Option GPI OUT)	0x29	CMDD0: Sekunden CMDD1: Frames	
		Übergabe der Vergleichszeit für GPI_1. Siehe auch Kommando 0x2A.	

Funktionsbeschreibung PCL PCI / PCL PCIe

Kommando	Nr.	Parameter	Ergebnis
pclGpi1Init (Option GPI OUT)	0x2A	<p>CMDD0: Modus</p> <ul style="list-style-type: none"> • Bits 0..2: Signalausgang 0 = statisch „H“ 1 = statisch „L“ 2 = „L“-Puls wiederholt 3 = „H“-Puls wiederholt 4 = „L“-Puls einmalig 5 = „H“-Puls einmalig 6 = ↓ Flanke einmalig 7 = ↑ Flanke einmalig • Bit 3: Zeitvergleich ohne (=0) bzw. mit (=1) Frames • Bits 4..5: Zeitvergleich 0 = „=“ (LTC nur vorw.) 1 = „>=“ (LTC nur vorw.) 2 = „=“ 3 = „>=“ • Bits 6..7: Pulsdauer 0 = ca. 10 ms 1 = ca. 40 ms 2 = ca. 500 ms 3 = ca. 1000 ms <p>CMDD1: Initialisierung</p> <ul style="list-style-type: none"> • = 0x00: GPI_1 wird initialisiert, d.h. der Ausgang wird in den Grundzustand versetzt und das nächste Ereignis kann die gesetzte Funktion auslösen. • <> 0x00: zusätzlich zur Initialisierung wird die durch Kommandos 0x28 und 0x29 übergebene Zeit als neue Vergleichszeit akzeptiert. 	
		Für GPI_1 wird die Funktion bestimmt und eine Initialisierung durchgeführt.	
pclGpi1Status (Option GPI OUT)	0x2B		<p>DATA0: seit letzter Initialisierung wurde kein (= 0) bzw. ein (<> 0) Ereignis entdeckt.</p> <p>DATA1: <> 0, wenn GPI_1 Signalpuls noch aktiv ist (= Timer für Pulsdauer).</p>
		Statusanfrage für GPI_1.	
pclGpi2H:M (Option GPI OUT)	0x2C	<p>CMDD0: Stunden</p> <p>CMDD1: Minuten</p>	
		Übergabe der Vergleichszeit für GPI_2. Siehe auch Kommando 0x2E.	

Kommando	Nr.	Parameter	Ergebnis
pclGpi2S:F (Option GPI OUT)	0x2D	CMDD0: Sekunden CMDD1: Frames	
		Übergabe der Vergleichszeit für GPI_2. Siehe auch Kommando 0x2E.	
pclGpi2Init (Option GPI OUT)	0x2E	CMDD0: Modus (identisch zu GPI_1, siehe Beschreibung zu Kommando 0x2A). CMDD1: Initialisierung (identisch zu GPI_1, siehe Beschreibung zu Kommando 0x2A).	
		Für GPI_2 wird die Funktion bestimmt und eine Initialisierung durchgeführt.	
pclGpi2Status (Option GPI OUT)	0x2F		DATA0: seit letzter Initialisierung wurde kein (= 0) bzw. ein (<> 0) Ereignis entdeckt. DATA1: <> 0, wenn GPI_2 Signalpuls noch aktiv ist (= Timer für Pulsdauer).
		Statusanfrage für GPI_2.	
pclTcFrames	0x30	CMDD0: Framerate (0x00) • 0x00: Automatik • 0x01: 24 fps • 0x02: 25 fps • 0x03: 30 fps ohne Drop • 0x04: 30 fps mit Drop	
		Die Framerate, mit der Timecode gelesen wird, kann automatisch aus den Leserwerten ermittelt oder manuell gesetzt werden. Wird die Framerate mit diesem Kommando gewählt, gilt diese Framerate für alle Timecodes. Mit den folgenden beiden Kommandos können den beiden Timecodearten (LTC bzw. Video-Timecode) unterschiedliche Frameraten zugewiesen werden.	
pclLtcFrames	0x31	CMDD0: Framerate (0x00) • 0x00: Automatik • 0x01: 24 fps • 0x02: 25 fps • 0x03: 30 fps ohne Drop • 0x04: 30 fps mit Drop	
		Die LTC Framerate kann automatisch aus den Leserwerten ermittelt oder manuell gesetzt werden. Bei PCL PCI / PCL PCIe HD und 3G wirkt dieses Kommando gleichzeitig auch auf ATC LTC und HANC LTC. Das Kommando ‚pclGetRegister‘ liefert die aktuelle Framerate als DATA9.	

Funktionsbeschreibung PCL PCI / PCL PCIe

Kommando	Nr.	Parameter	Ergebnis
pclVtcFrames	0x32	CMDD0: Framerate (0x00) <ul style="list-style-type: none"> • 0x00: Automatik • 0x01: 24 fps • 0x02: 25 fps • 0x03: 30 fps ohne Drop • 0x04: 30 fps mit Drop 	
		Die Framerate der Video-Timecodes (VITC/DVITC) kann automatisch aus den Leserwerten ermittelt oder manuell gesetzt werden. Bei PCL PCI / PCL PCIe HD und 3G wirkt dieses Kommando gleichzeitig auch auf ATC VITC, und HANC VITC. Das Kommando ‚pclGetRegister‘ liefert die aktuelle Framerate als DATA9.	
pclSendldata	0x38	CMDD0: IDATA Adresse	DATA0: IDATA Daten
pclSetldata	0x39	CMDD0: IDATA Adresse CMDD1: IDATA Daten	
pclSendpdata	0x3A	CMDD0: PDATA Adresse	DATA0: PDATA Daten
pclSetpdata	0x3B	CMDD0: PDATA Adresse CMDD1: PDATA Daten	
pclSendxdata	0x3C	CMDD0: XDATA Adresse H CMDD1: XDATA Adresse L	DATA0: XDATA Daten
Die ldata- Pdata- und Xdata-Kommandos sind für Diagnosezwecke vorgesehen.			
pclGetTc	0x41		DATA0: Frames DATA1: Sekunden DATA2: Minuten DATA3: Stunden
		Lese die Zeitinformation des Timecodes aus dem Mixed-Register. Die Daten werden als gepackte BCD-Zahlen übergeben.	
pclGetUser	0x42		DATA0: U1/2 („Frames“) DATA1: U3/4 („Sekunden“) DATA2: U5/6 („Minuten“) DATA3: U7/8 („Stunden“)
		Lese die Userbits des Timecodes aus dem Mixed-Register	

Kommando	Nr.	Parameter	Ergebnis
pciGetRegister	0x43	CMDD0: Registernummer • 0x00: Mixed-Register • 0x02: LTC-Register • 0x03: VITC-Register • 0x11: ATC LTC • 0x12: ATC VITC • 0x14: HANC LTC • 0x15: HANC VITC	DATA0: Frames DATA1: Sekunden DATA2: Minuten DATA3: Stunden DATA4: Flagbits LTC • 0x01: LTC Bit 10 (30 Drop) • 0x02: LTC Bit 11 (CF) • 0x04: LTC Bit 27 • 0x08: LTC Bit 43 • 0x10: LTC Bit 58 • 0x20: LTC Bit 59 • 0x40: 0, wenn Werte als LTC gelesen. 1, wenn Werte durch Schwungrad erzeugt. • 0x80: 0, wenn LTC vorwärts. 1, wenn LTC rückwärts oder DATA4: Flagbits VITC • 0x01: VITC Bit 14 (30 Drop) • 0x02: VITC Bit 15 (CF) • 0x04: VITC Bit 35 (30 Field) • 0x08: VITC Bit 55 • 0x10: VITC Bit 74 • 0x20: VITC Bit 75 (25 Field) • 0x40: 0, wenn Werte als VITC gelesen. 1, wenn Werte durch Schwungrad erzeugt. DATA5: U1/2 („Frames“) DATA6: U3/4 („Sekunden“) DATA7: U5/6 („Minuten“) DATA8: U7/8 („Stunden“) DATA9: Framerate • 0x24: 24 fps • 0x25: 25 fps • 0x30: 30 fps ohne Drop • 0xB0: 30 fps mit Drop DATAA: Neue-Daten-Flags • 0x00: Keine neuen Daten • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC • 0x40: LTC • 0x80: VITC/DVITC
Lese die Zeit, die Userbits, die Flagbits und Statusinformationen aus den verschiedenen Timecode-Registern.			

Funktionsbeschreibung PCL PCI / PCL PCIe

Kommando	Nr.	Parameter	Ergebnis
pclGetRegStatus	0x44	CMDD0: Registernummer <ul style="list-style-type: none"> • 0x11: ATC LTC • 0x12: ATC VITC • 0x15: HANC VITC 	DATA0: DID 1 (ATC) DATA1: DID 2 (ATC) bzw. Line Select (HANC VITC)
		Lese erweiterten Registerstatus.	
pclSetDBB1	0x45	CMDD0: ATC Register CMDD1: DBB1	
		Setze DBB1 eines ATC Registers	
pclGetVersion	0x46		DATA0: Produkt-Identifikation <ul style="list-style-type: none"> • 0x00: Normale Firmware • sonst: Sonderversion DATA1: Versionsnummer <ul style="list-style-type: none"> • Bit 7..6: 0x40 = PCL PCI / PCL PCIe • Bit 5..4: Hauptversion • Bit 3..0: Nebenversion
		Lese Versionsnummer der Firmware und Produkttyp.	
pclGetVerSpecial	0x47		DATA0: Produkt-Identifikation DATA1: Produkt-Identifikation DATA2: Produkt-Identifikation
		Lese Identifikation von Sonder-Firmwareversionen.	
pclIntAck	0x48	CMDD0: Interrupt-Maske L <ul style="list-style-type: none"> • 0x01: Mixed-Register • 0x04: LTC-Register • 0x08: VITC-Register • 0x40: Ungerade Halbbilder • 0x80: Gerade Halbbilder CMDD1: Interrupt-Maske H (nur PCL PCI / PCL PCIe HD und 3G) <ul style="list-style-type: none"> • 0x01: ATC LTC • 0x02: ATC VITC • 0x08: HANC LTC • 0x10: HANC VITC 	
		Setze für alle gesetzten Bits die Interrupt-Anforderung zurück.	
pclSetDID	0x49	CMDD0: ANC Register <ul style="list-style-type: none"> • 0x00: ATC • 0x01: HANC LTC • 0x02: HANC VITC CMDD1: DID	
		Setze DID eines ANC Registers	
pclSetSDID	0x4A	CMDD0: ANC Register <ul style="list-style-type: none"> • 0x00: ATC • 0x01: HANC LTC • 0x02: HANC VITC CMDD1: SDID	
		Setze SDID eines ANC Registers	

Kommando	Nr.	Parameter	Ergebnis
pclGetVideoStandard	0x4B		<ul style="list-style-type: none"> • DATA0: Video System 0x00: Unbek. SD Video 0x01: 625i25 0x02: 525i29.97 0x03: 1080i25 0x04: 1080i29.97 oder 30 0x05: 1080i24 0x06: 720p50 0x07: 720p59.94 oder 60 0x08: 1080p25 0x09: 1080p29.97 oder 30 0x0A: 1080p/24 0x0B: 720p24 0x0C: 720p25 0x0D: 720p29.97 oder 30 0x0E: 1035i29.97 oder 30 0x0F: Unbek. HD Video 0x10: Unbek. 3G Video 0x11: 1080p50 0x12: 1080p59.94 oder 60 Die folgenden Videosysteme können nur mit Unterstützung durch ein Payload Identifier Paket nach SMPTE 352M erkannt werden: 0x13: 720p23.98 0x14: 720p24 0x15: 720p29.97 0x16: 720p30 0x17: 720p59.94 0x18: 720p60 0x19: 1080i25 0x1A: 1080i29.97 0x1B: 1080i30 0x1C: 1080psF23.98 0x1D: 1080psF24 0x1E: 1080psF25 0x1F: 1080psF29.97 0x20: 1080psF30 0x21: 1080p23.98 0x22: 1080p24 0x23: 1080p29.97 0x24: 1080p30 0x25: 1080p59.94 0x26: 1080p60 • DATA1: Format 0x00: Zeilensprung 0x01: Vollbild
Lese Informationen über angeschlossenes Videosignal			

Funktionsbeschreibung PCL PCI / PCL PCIe

Kommando	Nr.	Parameter	Ergebnis
pclGetVideoRaster	0x4C		<ul style="list-style-type: none"> • DATA0: Words / active line (low byte) • DATA1: Words / active line (high byte) • DATA2: Total words / line (low byte) • DATA3: Total words / line (high byte) • DATA4: Active lines / field (low byte) • DATA5: Active lines / field (high byte) • DATA6: Total lines / frame (low byte) • DATA7: Total lines / frame (high byte)
			Lese Informationen über angeschlossenes Videosignal
pclSetAncDID	0x4D	CMDD0: User ANC DID CMDD1: User ANC SDID	
		Setze DID und SDID des benutzerdefinierten ANC-Lesers	
pclGetAncData	0x4E		<ul style="list-style-type: none"> • DATA0: data address pointer (low byte) • DATA1: data address pointer (high byte) • DATA2: descriptor address pointer (low byte) • DATA3: descriptor address pointer (high byte)
			Lese Adresszeiger des benutzerdefinierten ANC-Lesers
pclGetGpi	0x4F		<ul style="list-style-type: none"> • DATA0: GPI IN Status 0x01: GPI1 aktiv 0x02: GPI2 aktiv • DATA1: GPI OUT Status 0x01: GPI1 aktiv 0x02: GPI2 aktiv
			Lese Status der GPI Ein- und Ausgänge

Kommando	Nr.	Parameter	Ergebnis
pclGetPayload-Identifizier	0x50		<ul style="list-style-type: none"> • DATA0: S352M Byte 1 • DATA1: S352M Byte 2 • DATA2: S352M Byte 3 • DATA3: S352M Byte 4 • DATA4: Flags <ul style="list-style-type: none"> 0x01: Ein gültiges S352M-Paket ist im Videosignal. 0x02: Das gelesene S352M-Paket widerspricht dem erkannten Videostandard. 0x04: Benutze S352M-Pakete, um den Videosystem zu erkennen
		Lese Payload Identifizier (SMPTE 352M)	
pclSetUsePayload-Identifizier	0x51	CMDD0: <ul style="list-style-type: none"> • 0x00: Ignoriere S352M Pakete • 0x04: Erkenne das Video System aus S253M Paketen 	
		Setze Benutzung des Payload Identifizier (SMPTE 352M)	

D Optionen

D1 Option „GPI OUT“: Zwei Signalausgänge

Beschreibung

PCL PCI kann durch eine Zusatzbestückung und durch eine spezielle Firmware zwei Signalausgänge erhalten, bezeichnet als GPI_1 und GPI_2 (GPI = General Purpose Interface). Die beiden Ausgänge sind unabhängig voneinander und können - gesteuert über Kommandos - folgende Funktionen erfüllen:

Ausgangssignal:

- statisch „H“ („H“ = siehe technische Daten);
- statisch „L“ („L“ = siehe technische Daten);
- „L“-Puls wiederholt, d.h. bei jedem Ereignis;
- „H“-Puls wiederholt, d.h. bei jedem Ereignis;
- „L“-Puls einmalig beim nächsten Ereignis, danach neue Initialisierung nötig;
- „H“-Puls einmalig beim nächsten Ereignis, danach neue Initialisierung nötig;
- fallende Flanke (Wechsel von „H“ nach „L“) einmalig beim nächsten Ereignis, danach neue Initialisierung nötig;
- steigende Flanke (Wechsel von „L“ nach „H“) einmalig beim nächsten Ereignis, danach neue Initialisierung nötig.

Die Pulsdauer ist einstellbar in vier Schritten:

- ca. 10 ms;
- ca. 40 ms;
- ca. 500 ms;
- ca. 1000 ms.

Ein Ereignis wird durch eine Timecode Vergleichszeit und einer Bedingung definiert. Die Vergleichszeit wird über Kommandos unabhängig für jeden Signalausgang als Std:Min:Sek:Frames der PCL PCI übermittelt. Ist das Ausgangssignal nicht statisch gewählt, prüft PCL PCI anhand dem in dem Mixed Register enthaltenen Zeitwert, ob die gewählte Bedingung erfüllt ist. Folgende Bedingungen sind über Kommandos zu wählen:

- gelesener Zeitwert = Vergleichszeit - wenn LTC gelesen wird, wird nur ein LTC in „Vorwärtsrichtung“ akzeptiert;
- gelesener Zeitwert > oder = Vergleichszeit - wenn LTC gelesen wird, wird nur ein LTC in „Vorwärtsrichtung“ akzeptiert;
- gelesener Zeitwert = Vergleichszeit;
- gelesener Zeitwert > oder = Vergleichszeit.

Für jede der vier Bedingungen kann gewählt werden, ob die Vergleichszeit mit oder ohne Frames berücksichtigt wird.

Durch das Kommando 0x2A bzw. 0x2B wird automatisch eine Initialisierung durchgeführt, d.h. die Signalausgänge werden in den Grundzustand geschaltet und sind bereit, auf ein neues Ereignis zu reagieren. Nach der Initialisierung und auch nach jedem Ereignis muss mindestens ein Frame gelesen werden, dass die gewählte Bedingung nicht erfüllt, bevor auf ein Ereignis (erneut) reagiert wird. Um auf eine Liste von Vergleichszeiten reagieren zu können, sollte der Signalausgang auf „Pulse bei jedem Ereignis“ eingestellt sein. Nach einem Ereignis muss die nächste Vergleichszeit möglichst innerhalb eines Frames gesetzt werden, damit die kleinstmögliche Wiederholrate von zwei Frames ermöglicht wird. Da die beiden Signalausgänge unabhängig voneinander sind, ist es natürlich möglich, GPI_2 in einem Abstand von einem Frame nach GPI_1 reagieren zu lassen.

Zu den Bedingungen Frames ja/nein und „=“ oder „>=“ ist folgendes zu beachten:

Werden die Frames berücksichtigt, wird das Ausgangssignal framegenau gesteuert. Allerdings muss sicher gestellt werden, dass dieses Frame auch existiert, d.h. dass das Frame in dem gültigen Zählbereich liegt (je nach Framerate unterschiedlich), dass keine Drop-Outs vorliegen (durch lückenbehafteten Timecode oder durch einen 30-Drop Modus). Um ein Drop-Out Problem zu umgehen, kann die „>=“ Bedingung eingestellt werden. Bei dieser Bedingung muss sichergestellt werden, dass die Anwendung korrekt gestartet wird, d.h. es muss nach der Initialisierung ein Timecode gelesen werden, der kleiner als die Vergleichszeit ist. Werden die Frames nicht berücksichtigt, tritt das Ereignis immer mit dem Sekundenwechsel auf. Ein Drop-Out Problem ist hier nicht zu erwarten.

Nach dem Einschalten ist PCL PCI in folgendem Zustand:

- statisch „H“;
- Pulsdauer ca. 10 ms;
- gelesener Zeitwert = Vergleichszeit - wenn LTC gelesen wird, wird nur ein LTC in „Vorwärtsrichtung“ akzeptiert;
- Frames werden nicht berücksichtigt.

Kommandos

Für die oben beschriebenen Funktionen wurde die Tabelle der Kommandos erweitert:

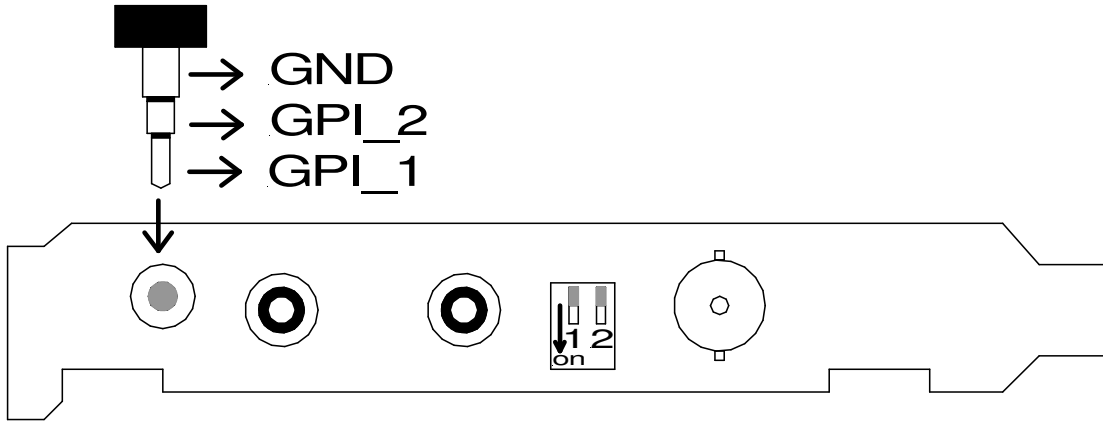
Kommandonummer	Kommando
0x28	Stunden:Minuten Vergleichszeit GPI_1
0x29	Sekunden:Frames Vergleichszeit GPI_1
0x2A	Funktion GPI_1 + Initialisierung
0x2B	GPI_1 Statusanfrage
0x2C	Stunden:Minuten Vergleichszeit GPI_2
0x2D	Sekunden:Frames Vergleichszeit GPI_2
0x2E	Funktion GPI_2 + Initialisierung
0x2F	GPI_2 Statusanfrage

Eine detaillierte Beschreibung ist in der Tabelle der Kommandos (Kapitel C) gegeben.

Anschluss und technische Daten

PCL PCI

Die Signale (GPI_1 = Signal 1, GPI_2 = Signal 2) sind am Slotblech über eine zusätzlich angebrachte Klinken-Buchse verfügbar. Der dazu passende Stereo-Klinkenstecker wird mitgeliefert.



PCL PCIe

Die Anschlüsse sind im Kapitel B3 beschrieben.

Technische Daten

Allgemeines	Open Collector Ausgang eines NPN Transistors. Max. Verlustleistung: 200 mW.
Zustand „H“	Open Collector Ausgang „offen“ (floating). Für einen logischen High-Pegel ist ein externer Pull-Up Widerstand notwendig. Typische Werte für eine Pull-Up Widerstand: 0.3 kΩ - 6 kΩ @3V 0.5 kΩ - 10 kΩ @5V 1.2 kΩ - 24 kΩ @12V 1.5 kΩ - 30 kΩ @15V 2.4 kΩ - 48 kΩ @24V Maximale externe Spannung: 30 V DC.
Zustand „L“	Ausgang wird nach GND geschaltet. Maximaler Kollektorstrom: 100 mA DC. Restspannung: @100 mA: typ. 200 mV (≤ 600 mV) @10 mA: typ. 90 mV (≤ 250 mV)

Verfügbarkeit

PCL Karte	GPI_1	GPI_2	Option / Standard
PCL PCI L	Ein/Ausgang	Ein/Ausgang	Option
PCL PCI LV	Ein/Ausgang	Ein/Ausgang	Option
PCL PCI D	Ein/Ausgang	Ein/Ausgang	Option
PCL PCI D (v2)	Ein/Ausgang	Nur Eingang	Option
PCL PCI HD	Ein/Ausgang	Nur Eingang	Option
PCL PCIe L	Ein/Ausgang	Nur Eingang	Standard
PCL PCIe LV	Ein/Ausgang	Nur Eingang	Standard
PCL PCIe D	Ein/Ausgang	Nur Eingang	Standard
PCL PCIe HD	Ein/Ausgang	Nur Eingang	Standard
PCL PCIe 3G	Ein/Ausgang	Ein/Ausgang	Standard

Treiber und Anwenderprogramme

Es stehen Treiber und auch ein Anwenderprogramm zur Verfügung. Bitte kontaktieren Sie Alpermann+Velte für nähere Informationen.