

Produktbeschreibung

PXI / PCI 3080

CAN/ LIN/ K-LINE/ J1850 Interfaces

Nutzerhandbuch

Version 1.5



GOPEL electronic GmbH
Göschwitzer Str. 58/60
D-07745 Jena
Tel.: +49-3641-6896-597
Fax: +49-3641-6896-944
E-Mail: ats_support@goepel.com
<http://www.goepel.com>

© 2012 GÖPEL electronic GmbH. Alle Rechte vorbehalten.

Die in diesem Handbuch beschriebene Software sowie das Handbuch selbst dürfen nur in Übereinstimmung mit den Lizenzbedingungen verwendet oder kopiert werden.
Zu Sicherungszwecken darf der Käufer eine Kopie der Software anfertigen.

Der Inhalt des Handbuchs dient ausschließlich der Information, ist nicht als Verpflichtung der GÖPEL electronic GmbH anzusehen und kann ohne Vorankündigung verändert werden.
Hard- und Software unterliegen ebenso möglichen Veränderungen im Sinne des technischen Fortschritts.

Die GÖPEL electronic GmbH übernimmt keinerlei Gewähr oder Garantie für Genauigkeit und Richtigkeit der Angaben in diesem Handbuch.

Ohne vorherige schriftliche Genehmigung der GÖPEL electronic GmbH darf kein Teil dieser Dokumentation in irgendeiner Art und Weise übertragen, vervielfältigt, in Datenbanken gespeichert oder in andere Sprachen übersetzt werden (es sei denn, dies ist durch die Lizenzbedingungen ausdrücklich erlaubt).

Die GÖPEL electronic GmbH haftet weder für unmittelbare Schäden noch für Folgeschäden aus der Anwendung ihrer Produkte.

Gedruckt: 23.04.2012

Alle in diesem Handbuch verwendeten Produkt- und Firmennamen sind Markennamen oder eingetragene Markennamen ihrer jeweiligen Eigentümer.

Stand: April 2012

1	INSTALLATION DER BOARDS	1-1
1.1	HARDWAREINSTALLATION	1-1
1.2	TREIBERINSTALLATION	1-2
1.2.1	<i>Windows Device Treiber</i>	1-2
1.2.2	<i>VISA Device Treiber</i>	1-3
2	HARDWARE PXI/ PCI 3080	2-1
2.1	BESTIMMUNG	2-1
2.2	TECHNISCHE DATEN	2-3
2.2.1	<i>Allgemeines</i>	2-3
2.2.2	<i>Abmessungen</i>	2-3
2.2.3	<i>PXI 3080/ PCI 3080 Kennwerte</i>	2-3
2.3	AUFBAU	2-4
2.3.1	<i>Allgemeines</i>	2-4
2.3.2	<i>Adressierung</i>	2-5
2.3.3	<i>Kommunikationsschnittstellen</i>	2-6
2.3.4	<i>Bestückung</i>	2-9
2.3.5	<i>Belegung Frontsteckverbinder</i>	2-11
2.3.6	<i>LED Anzeige</i>	2-12
2.4	LIEFERHINWEISE	2-13
3	ANSTEUERSOFTWARE	3-1
3.1	PROGRAMMIEREN ÜBER G-API	3-2
3.2	PROGRAMMIEREN ÜBER DLL-FUNKTIONEN	3-3
3.2.1	<i>Windows Device Treiber</i>	3-4
3.2.1.1	Driver Info	3-5
3.2.1.2	DLL Version	3-6
3.2.1.3	XILINX Download	3-7
3.2.1.4	XILINX Write Data	3-8
3.2.1.5	DPRAM Write Instruction	3-9
3.2.1.6	DPRAM Read Response	3-10
3.2.1.7	Reset Port	3-11
3.2.2	<i>VISA Device Treiber</i>	3-12
3.2.2.1	Init	3-13
3.2.2.2	Done	3-13
3.2.2.3	Driver Info	3-14
3.2.2.4	XILINX Download	3-15
3.2.2.5	XILINX Write Data	3-16
3.2.2.6	Write Data	3-17
3.2.2.7	Read Data	3-18
3.2.2.8	Reset Port	3-19
3.3	PROGRAMMIEREN MIT LABVIEW	3-20
3.3.1	<i>LabVIEW über G-API</i>	3-20
3.3.2	<i>LLB unter Verwendung des Windows Device Treibers</i>	3-20
3.3.3	<i>LLB unter Verwendung des VISA Device Treibers</i>	3-20
3.4	WEITERE GÖPEL SOFTWARE	3-20

1 Installation der Boards

1.1 Hardwareinstallation



Warnung

Stellen Sie bitte unbedingt sicher, dass alle Installationsarbeiten im **ausgeschalteten** Zustand Ihres Systems erfolgen!
Die Stromversorgung sollte abgeklemmt sein.



Vergleichen Sie bitte auch das Handbuch für Ihr PXI/ PCI-System.
Ggf. sind darin weitere zu beachtende Installationshinweise enthalten.



Warnung

Elektrostatische Entladungen (ESD) können Ihr System schädigen und elektronische Bauelemente zerstören. Das kann zu irreparablen Schäden am PXI/ PCI 3080-Board oder an dem System führen, in dem das Board betrieben wird.
Folge sind unerwartete Fehlfunktionen Ihres Prüfsystems. Berühren Sie daher niemals die Boardoberfläche, Steckverbinderanschlüsse oder elektronische Bauelemente.

Das PCI™-, CompactPCI™- oder PXI™-System wird entsprechend seinen Gegebenheiten geöffnet. Wählen Sie einen freien Steckplatz in Ihrem System aus.

Beim ausgewählten Steckplatz entfernen Sie das vorhandene Slotblech. Dazu müssen die Befestigungsschrauben gelöst werden.

(Wenn es notwendig ist, Transceivermodule zu tauschen, sind die allgemeinen Regeln zur Vermeidung von elektrostatischen Aufladungen zu beachten. Die Module dürfen nie unter Spannung gezogen oder gesteckt werden! Ein lagerichtiges Stecken der Module ist unbedingt zu realisieren.)

Das Board ist vorsichtig in den vorbereiteten Steckplatz einzuführen. PXI-Boards werden mit dem an der Frontplatte befindlichen Hebel das letzte Stück eingeschoben.

Nach dem Kontaktieren des Boards wird dieses mit den Schrauben am Frontblech befestigt. Somit ist das Board ordnungsgemäß eingebaut.

Danach sind ggf. die Arbeiten am System auszuführen, die dieses wieder betriebsbereit machen.

1.2 Treiberinstallation

1.2.1 Windows Device Treiber

PXI/ PCI 3080-Boards können unter Windows® 2000/ XP sowie unter Windows® 7/ 64 Bit betrieben werden.

Durch die Plug-and-Play Fähigkeit von Windows® wird automatisch (über den Hardwareassistenten) eine Treiberinstallation für jede neu erkannte Hardwarekomponente gestartet.

Mit der auf der beiliegenden CD im Ordner *GPxi3080* enthaltenen *inf*-Datei kann der Hardwareassistent die Installation des Devicetreibers durchführen.

Bei Bedarf finden Sie die jeweils erforderliche *inf*-Datei

- ♦ *GPxi3080.inf* für Windows® 2000/ XP im Ordner *Win2000 (Version xx)*
- ♦ *GPxi3080_x64.inf* für Windows® 7/ 64 Bit im Ordner *Win7_x64 (Version xx)*

Ein Neustart des Systems ist nicht zwingend erforderlich.



Der folgende Schritt ist nur erforderlich, wenn Sie nicht mit der G-API arbeiten (siehe auch [Ansteuersoftware](#)).

Wenn Sie eigene Software für die Boards erstellen wollen, benötigen Sie ggf. zusätzliche Dateien für die anwenderspezifische Programmierung (**.LLB*, **.H*). Diese werden nicht automatisch übernommen und müssen deshalb manuell von der mitgelieferten CD in Ihr Entwicklungsverzeichnis kopiert werden.

1.2.2 VISA Device Treiber

1. Schritt

Kopieren Sie den mitgelieferten Ordner *VISA (Version xx)* aus dem Ordner *GPxi3080* der mitgelieferten CD auf die Festplatte (Empfehlung: vollständigen Ordner auf *C: *).

2. Schritt

VISA für Windows® 2000, WindowsXP :

Durch die Plug-and-Play Fähigkeit wird für jede neu erkannte Hardwarekomponente automatisch eine Treiberinstallation über den Hardwareassistenten gestartet. Folgen Sie den Anweisungen und geben Sie bei der Suche nach dem Treiber das Zielverzeichnis an, in dem sich die Datei *PXI3080_NT5.inf* befindet (nach Empfehlung: *C: \VISA (Version xx) \ Installation*).

VISA für LabViewRT :

Für den Einsatz der PXI/ PCI 3080-Boards unter dem RT Betriebssystem muss die Datei *P3080_RT.inf* aus dem Verzeichnis *C: \VISA (Version xx) \ Installation* verwendet werden.

Kopieren Sie diese Datei in das Verzeichnis *\ni-rt\system* des embedded controllers.

Nutzen Sie dafür den NI Measurement & Automation Explorer.

Unter **Netzwerkverbindung** finden Sie den angeschlossenen RT-Controller. Über die rechte Maustaste öffnet sich ein Popup-Menü. Wählen Sie daraus den Menüeintrag **Dateitransfer** und folgen Sie den weiteren Anweisungen.



Zum ggf.späteren Erstellen einer *startup.rtexe* sollte auch die Datei *cvi_lvrt.dll* in das Verzeichnis *\ni-rt\system* kopiert werden.

3. Schritt:

Nach einem Neustart des Computers ist die Installation abgeschlossen.

Nach der Hardware- und Treiberinstallation können Sie überprüfen, ob die Boards einwandfrei vom System eingebunden worden sind:

Abbildung 1-1:
Windows

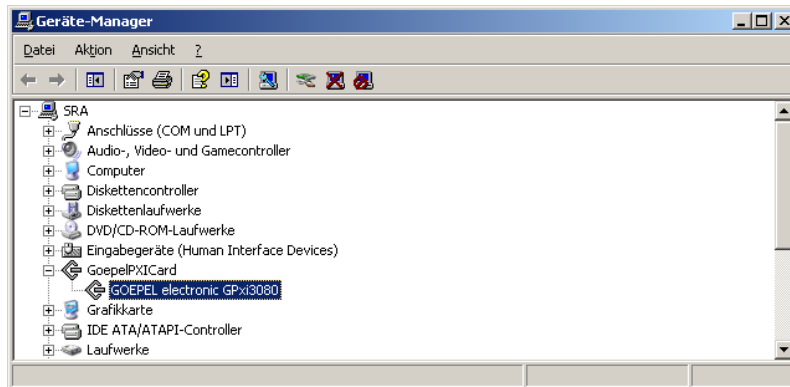


Abbildung 1-2:
VISA für Windows XP

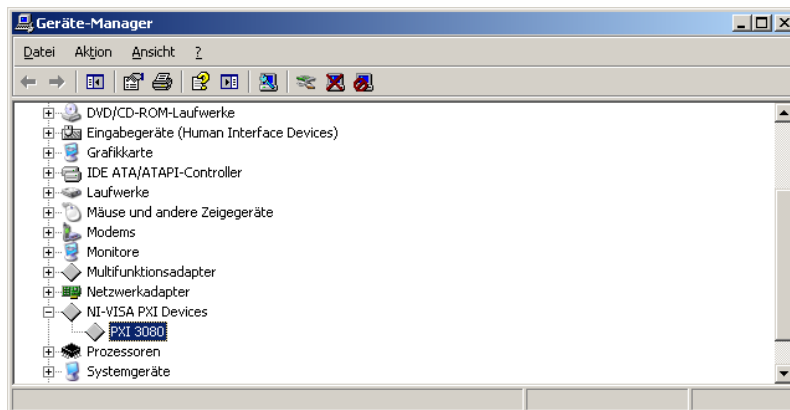
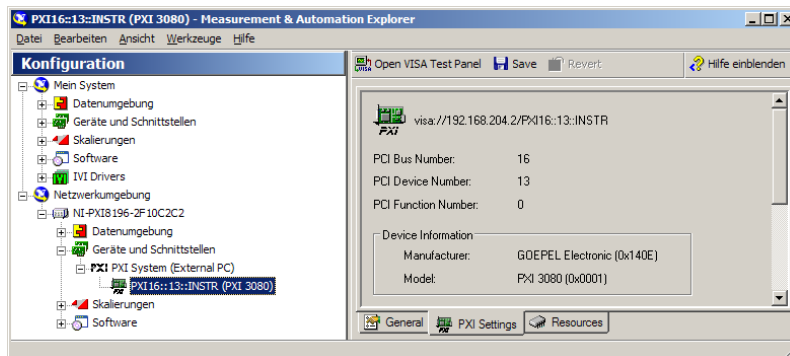


Abbildung 1-3:
VISA für LabVIEW RT



2 Hardware PXI/ PCI 3080

2.1 Bestimmung

Die Multi-Interface-Boards PXI 3080/ PCI 3080 sind Kommunikationsboards der GÖPEL electronic GmbH.

Diese Boards werden in der allgemeinen Steuerungstechnik, speziell in der Automobiltechnik, verwendet und bieten in der maximalen Ausbaustufe folgende Ressourcen:

- ◆ 2 x CAN
- ◆ 2 x K-Line oder LIN
- ◆ 1 x J1850 VPW
- ◆ 1 x J1850 PWM
(in diesem Fall ist nur EIN LIN oder K-Line Interface möglich)
- ◆ 4 x digital Input
- ◆ 4 x digital Output
- ◆ 2 x analog Input
- ◆ 1 x Weckleitung

Die Ressourcen sind galvanisch vom PXI- bzw. PCI-Interface getrennt.



Abbildung 2-1:
PXI 3080

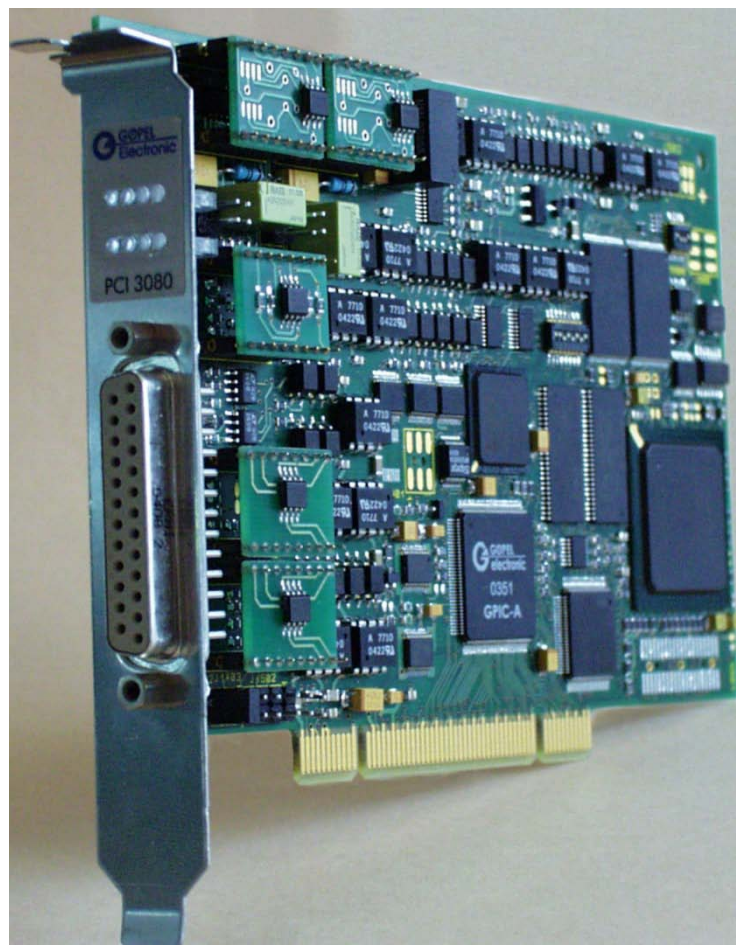


Abbildung 2-2:
PCI 3080

2.2 Technische Daten

2.2.1 Allgemeines

Das Kommunikationsboard PXI 3080 ist ein Einsteckboard, das für den PXI™-Bus (PCI eXtensions for Instrumentation) entwickelt wurde. Basis für diesen Bus ist der CompactPCI™-Bus.

Es ist möglich, das Board in einem CompactPCI™- oder einem PXI™-System zu betreiben. Dieses Board kann in jeden beliebigen Steckplatz (ausgenommen Steckplatz 1) eines solchen Systems gesteckt werden. Es ist auch bei gleichzeitigem Gebrauch mehrerer Boards dieses Typs in einem Rack eindeutig identifizierbar.

Das Kommunikationsboard PCI 3080 ist ein PC-Einsteckboard für den PCI Local Bus Rev. 2.2 und kann in jedem beliebigen PCI-Steckplatz (32Bit, 33MHz, 3,3V) betrieben werden.

Beide Boards haben keine Jumper zur Hardwareerkennung und werden automatisch in das jeweilige System eingebunden.

2.2.2 Abmessungen

Die Abmessungen beider Boards entsprechen Standard-Abmessungen des zugehörigen Bussystems:

- PXI 3080 Multi Interface Board: 160 mm x 100 mm (L x B)
- PCI 3080 Multi Interface Board: 168 mm x 106 mm (L x B)

2.2.3 PXI 3080/ PCI 3080 Kennwerte

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bemerkung
V _{BAT}	Batteriespannung		12	27/ 50	V	Abh. v. Transceivertyp
	Übertragungsrate			1	MBaud	CAN
	Übertragungsrate			22	kBaud	LIN
R _{bus}	Abschlusswiderstand 1		120		Ohm	CAN Jumper gesteckt
R _{bus}	Abschlusswiderstände 2		10		kOhm	CAN Jumper gesteckt
R _{Pullup}	Pullup-Widerstand		680		Ohm	K-Line Jumper gesteckt
V _{in}	Eingangsspannung	3,3		50	V	Digital Input
V _{out}	Ausgangsspannung			V _{BAT}	V	Digital Output, OC
V _{in}	Eingangsspannung			26	V	Analog Input
V _{iso}	galvanische Trennung	1000			V	PXI/ PCI In-/Output



Die analogen Eingangs-Kanäle (analog inputs) sind mit dem Schaltkreis LTC 1400 (AD-Wandler) von Linear Technology realisiert. Dieses Bauelement weist eine Auflösung von 12 Bit sowie einen Eingangsspannungsbereich von 0..4,095V auf.

Durch den Eingangs-Spannungsteiler (122K/22K) ergibt sich für die gemessene Spannung:

$$V_{\text{mess}} = \text{AD-Wandlerwert} * 1\text{mV} * (122\text{K}/22\text{K}).$$

2.3 Aufbau

2.3.1 Allgemeines

Bei den PXI/ PCI 3080-Boards dient ein ASIC als Interface zum PCI- oder cPCI-Bus. Dieser beinhaltet alle notwendigen Funktionsblöcke, die für eine Kommunikation mit dem Rechner-Bus notwendig sind.

Das Board PCI 3080 besitzt kein PXI-Interface. Um dennoch Triggersignale mit anderen PCI-Boards von GÖPEL electronic auszutauschen, befindet sich ein zusätzlicher Steckverbinder mit zwei jeweils als Input oder Output konfigurierbaren Leitungen auf diesem Board (J902 in Abbildung 2-7).

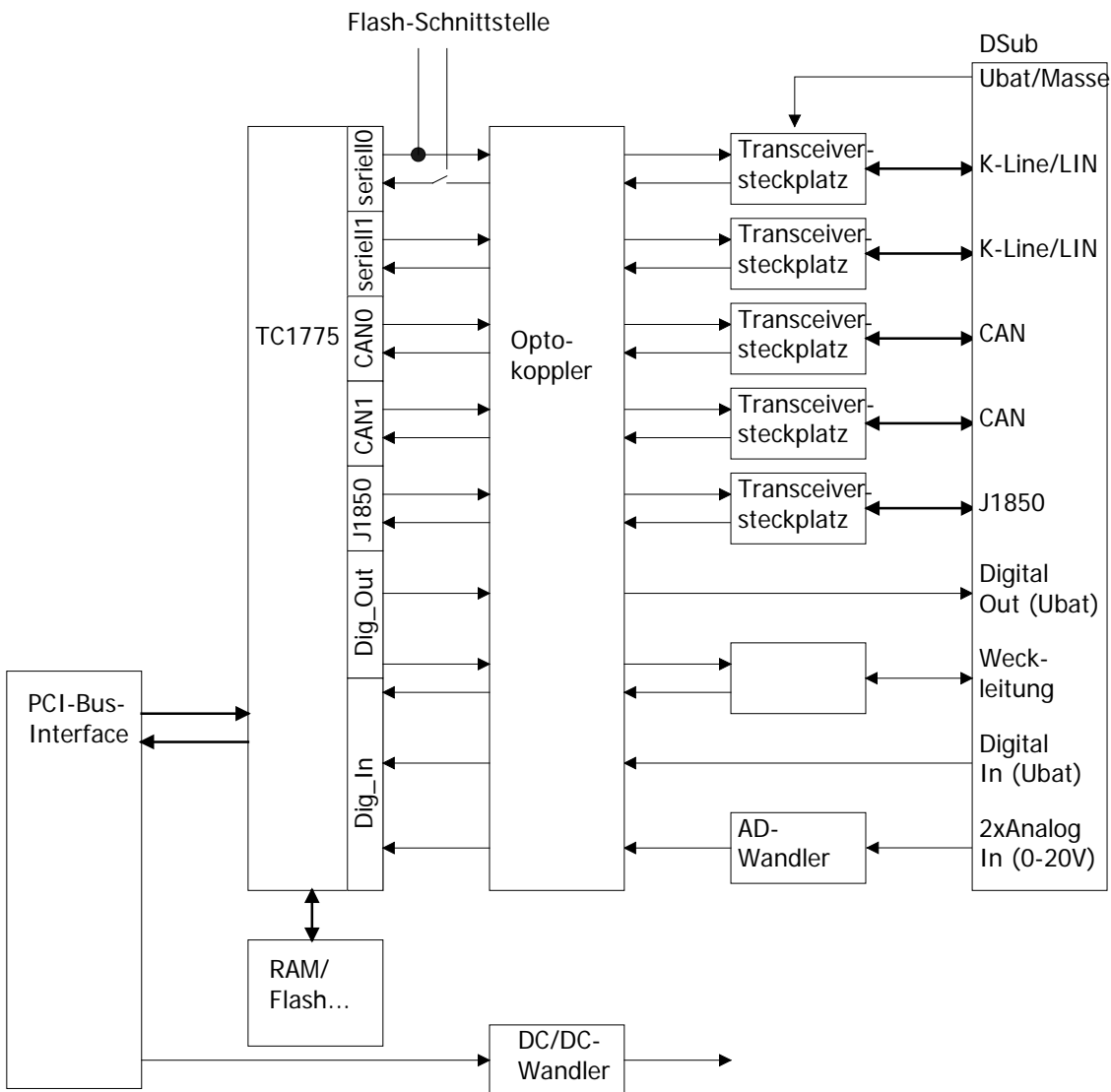


Abbildung 2-3: Blockdiagramm eines Kommunikationsboards PXI/ PCI 3080

2.3.2 Adressierung

PXI 3080: PXI-Racks besitzen eine eigene geographische Slotadressierung der Backplane. Die Nummerierung beginnt mit 1 und ist auf der Gehäusefrontseite sichtbar. Steckplatz 1 ist immer mit einem embedded Controller oder einer MXI-Karte zu bestücken.

Ein PXI 3080 Board kann die geographische Slotadresse auslesen. Hierzu muss der XILINX mit dem zugehörigen FPGA File geladen sein (siehe Funktionen [XilinxDownload](#) für unterschiedliche Treiber im Abschnitt [Ansteuersoftware](#)).

PCI 3080: PCI-Racks besitzen keine geographische Slotadressierung. Um dennoch mehrere PCI 3080 auf den Steckplätzen eindeutig identifizieren zu können, verfügt das Board über einen separaten DIP-Schalter (S900 in [Abbildung 2-7](#)). Damit ist es möglich, bis zu 16 Adressvarianten zu wählen.

Der mit diesem DIP-Schalter gesetzte Binärwert (0..15) kann über die mitgelieferte Software ausgelesen werden.

2.3.3 Kommunikations-schnittstellen

2 x CAN-Interface Version 2.0b:

Für die uneingeschränkte Funktion eines CAN-Interfaces an einem Netzwerk ist der verwendete Transceiver entscheidend. Häufig funktionieren CAN-Netzwerke nur, wenn alle Teilnehmer kompatible Transceiver im Netz haben.

Damit die Nutzer eines PXI/ PCI 3080-Boards keinen Einschränkungen unterliegen, sind die Transceiver als steckbare Module ausgeführt. Dabei stehen verschiedene Varianten (Highspeed, Lowspeed, Single-Wire u.a.) zur Auswahl, die einfach auszutauschen sind.

Neben dem Transceiver ist der Busabschlusswiderstand für die einwandfreie Funktion des CAN-Netzwerkes wichtig.

Werden Highspeed CAN-Transceiver verwendet, ist i. Allg. ein 120 Ohm Widerstand für jede CAN-Schnittstelle aktiv. Diese Widerstände können durch Ziehen der Jumper J1401 bzw. J1501 deaktiviert werden.

Bei Verwendung von Lowspeed CAN-Transceivern sind i. Allg. zwei Abschlusswiderstände von 10 kOhm für RTH und RTL für jede CAN-Schnittstelle aktiv (durch Bestücken der Jumper J1402/ J1403 bzw. J1502/ J1503). In diesem Fall müssen die Jumper J1401 bzw. J1501 geöffnet sein.

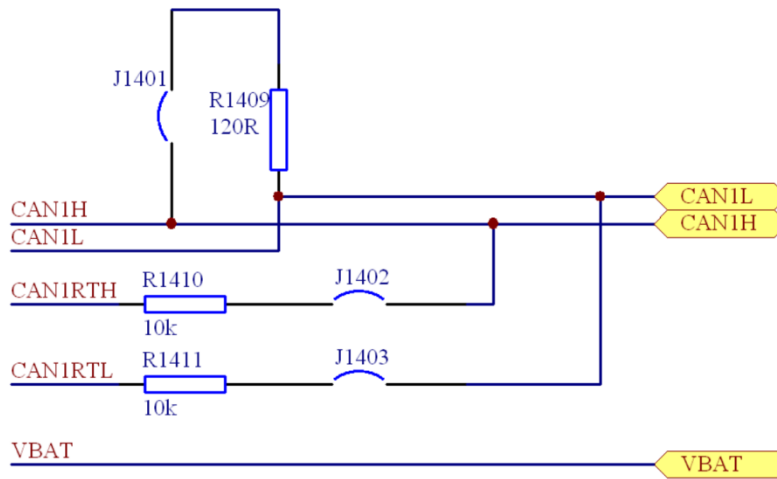


Abbildung 2-4:
CAN Schnittstelle

**2 x K-Line Interface (ISO 9141) oder
2 x LIN-Interface Version 2.0:**

K-Line:

Die Transceiver sind als steckbare Module ausgeführt. I. Allg. wird der L9637 der Fa. ST für diese Transceiver verwendet.

Über die Anschlüsse V_{BAT} wird die Versorgungsspannung der Transceivermodule angeschlossen. Zum Überbrücken der Verpoldiode für V_{BAT} für LIN müssen die Jumper J1602 bzw. J1703 bestückt sein.

Falls der Pullup-Widerstand gegen V_{BAT} aktiviert werden soll, müssen die Jumper J1601 bzw. J1701 bestückt werden.

LIN:

Die Transceiver sind als steckbare Module ausgeführt. I. Allg. wird der TJA1020 von Philips für diese Transceiver verwendet.

In der Standardausführung der Transceivermodule kann per Software über die Relais Rel1 für LIN1 bzw. Rel2 für LIN2 zwischen Master- und Slave-Konfiguration umgeschaltet werden. Die Pullup-Widerstände für LIN befinden sich auf dem Transceivermodul, sodass die Jumper J1601 bzw. J1701 nicht bestückt werden dürfen.

Über die Anschlüsse V_{BAT} wird die Versorgungsspannung der Transceivermodule angeschlossen. Gemäß der LIN-Spezifikation soll diese Versorgung über eine Verpoldiode erfolgen, sodass die Jumper J1602 bzw. J1703 nicht bestückt werden dürfen.

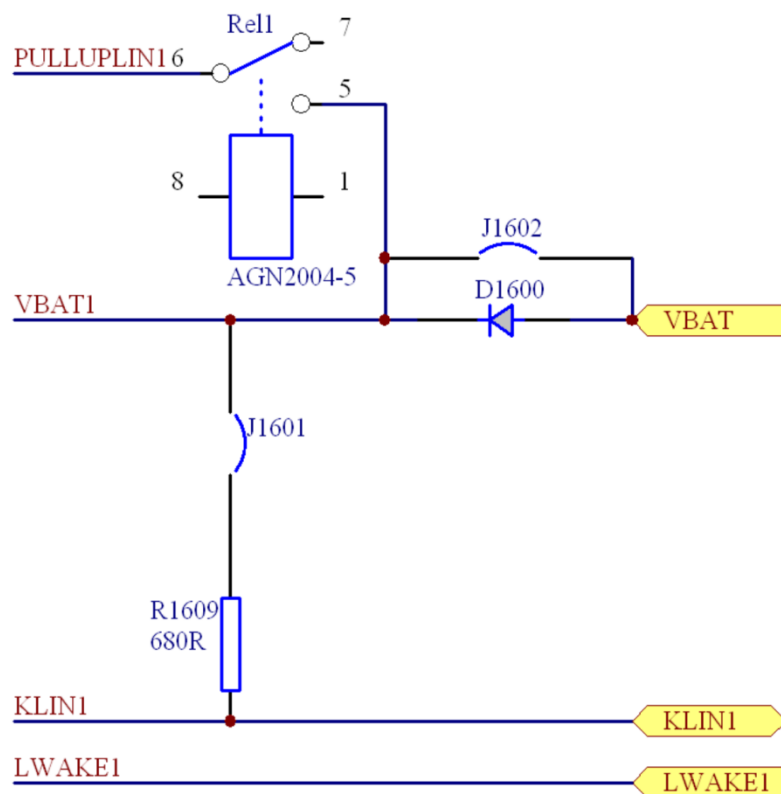


Abbildung 2-5:
LIN/ K-Line Schnittstelle

J1850 Interfaces:

Die Transceiver sind als steckbare Module ausgeführt.

I. Allg. wird der AU5780 von Philips für J1850 VPW Transceiver verwendet.

Die Ausgangsschaltung für den J1850 PWM Transceiver ist mit diskreten Bauelementen realisiert.

Falls ein J1850 VPW Interface realisiert werden soll, muss der Transceiver auf dem Steckplatz für J1850 bestückt werden.

Um ein J1850 PWM Interface zu realisieren, wird der Transceiver auf dem Steckplatz für K-Line/ LIN 2 bestückt (siehe Abbildung 2-6 und Abbildung 2-7).



J1701 darf bei einem J1850 PWM Interface nicht bestückt sein!

2.3.4 Bestückung

Abbildung 2-6, Abbildung 2-7 zeigen schematisch die Bestückungsseite der Boards. In den Abbildungen ist die Lage der Transceivermodule, Steckverbinder, DIP-Schalter und Jumper zu erkennen.

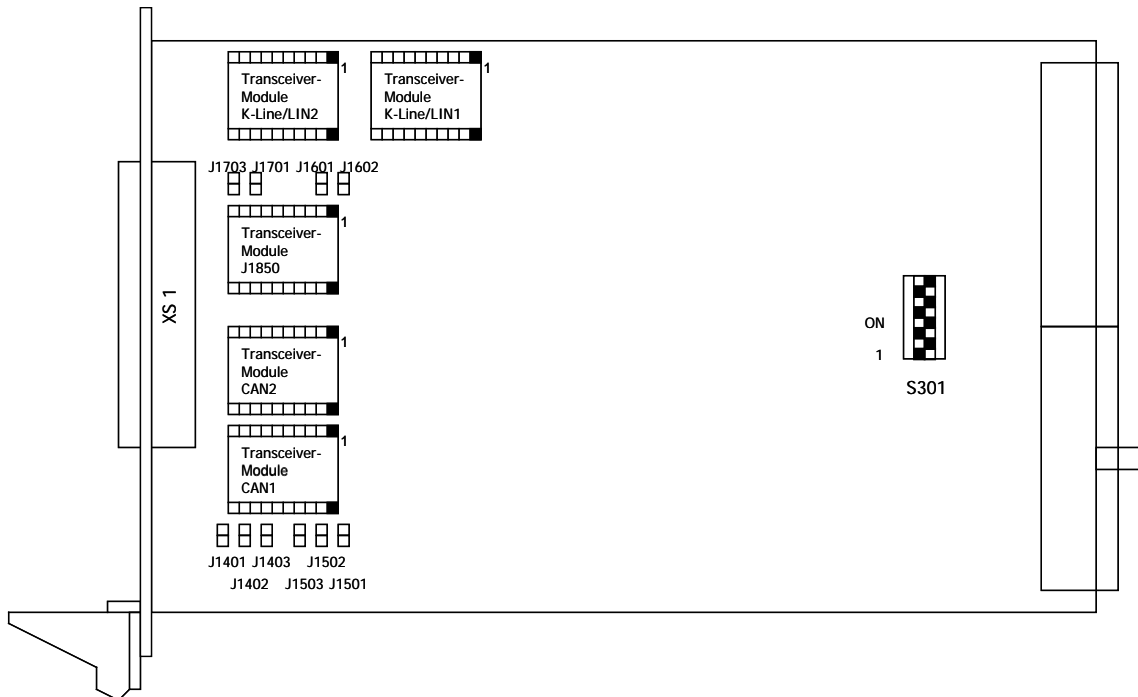


Abbildung 2-6: Schematischer Bestückungsplan Kommunikationsboard PXI 3080

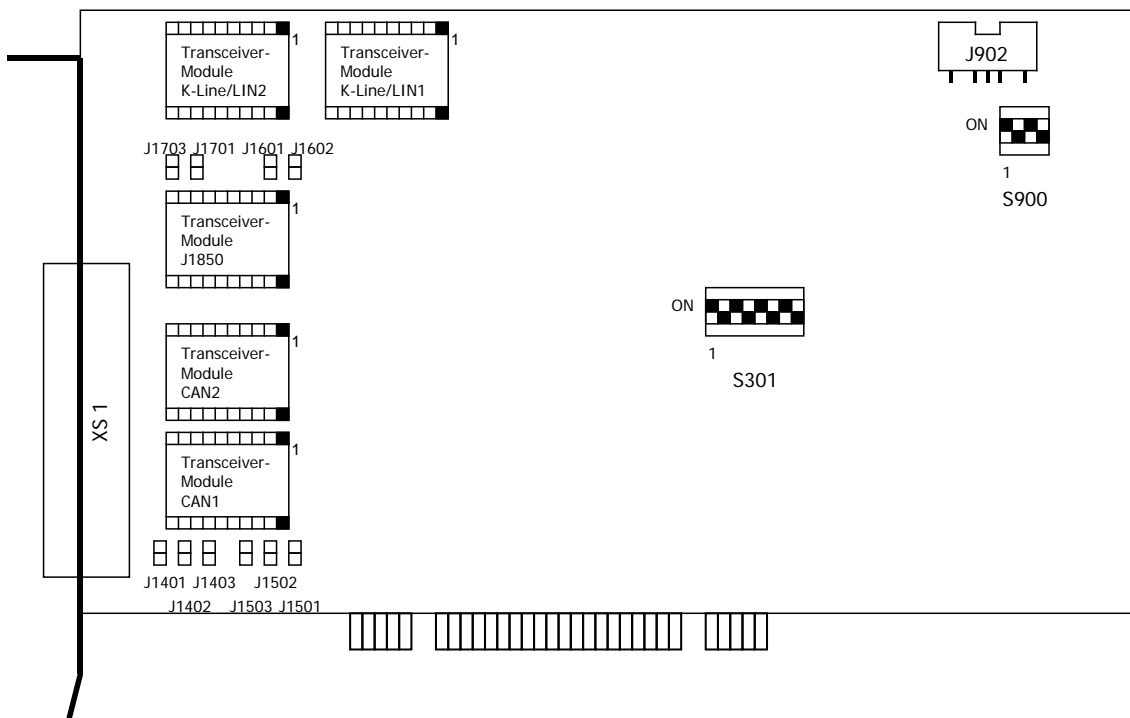


Abbildung 2-7: Schematischer Bestückungsplan Kommunikationsboard PCI 3080



Falls ein J1850 VPW Interface realisiert werden soll, muss der Transceiver auf dem Steckplatz für J1850 bestückt werden. Um ein J1850 PWM Interface zu realisieren, wird der Transceiver auf dem Steckplatz für einen K-Line/ LIN 2 Transceiver bestückt.

Die Konfigurationselemente aus Abbildung 2-6 und Abbildung 2-7 werden in der folgenden Tabelle erläutert:

XS1401	Transceivermodul für CAN1
J1401	Jumper zum Aktivieren des 120Ω Abschlusswiderstandes für CAN1
J1402	Jumper zum Aktivieren des 10kΩ Abschlusswiderstandes RTH für CAN1
J1403	Jumper zum Aktivieren des 10kΩ Abschlusswiderstandes RTL für CAN1
XS1501	Transceivermodul für CAN2
J1501	Jumper zum Aktivieren des 120Ω Abschlusswiderstandes für CAN2
J1502	Jumper zum Aktivieren des 10kΩ Abschlusswiderstandes RTH für CAN2
J1503	Jumper zum Aktivieren des 10kΩ Abschlusswiderstandes RTL für CAN2
XS1601	Transceivermodul für LIN1/ K-Line1
J1601	Jumper zum Aktivieren des 680Ω Pullupwiderstandes gegen V_{BAT} für K-Line1
J1602	Jumper zum Überbrücken der Verpoldiode für V_{BAT} für LIN1
XS1701	Transceivermodul für LIN2/ K-Line2/ J1850 PWM
J1701	Jumper zum Aktivieren des 680Ω Pullupwiderstandes gegen V_{BAT} für K-Line2
J1703	Jumper zum Überbrücken der Verpoldiode für V_{BAT} für LIN2
XS1801	Transceivermodul für J1850 VPW
S301	DIP-Schalter der PXI/ PCI 3080-Boards zur Konfiguration des Microcontrollers. Die Einstellung darf nicht verändert werden!
S900	Dieser DIP-Schalter auf der PCI 3080 dient zur eindeutigen Adressierung des Boards in einem System mit mehreren PCI 3080 Boards (analog „geografische Adressierung“ der PXI-Spezifikation). Dazu kann ein entsprechender Binärwert (0..15) über den Schalter eingestellt werden, der über die mitgelieferte Software ausgelesen wird.
J902	Steckverbinder zum Austausch von Triggersignalen mit anderen PCI-Boards von GÖPEL electronic

2.3.5 Belegung Front- steckverbinder

Typ: DSub 25-polig Buchse

Die Schnittstellen stehen über diesen Steckverbinder an der Frontseite der Kommunikationsboards PXI/ PCI 3080 zur Verfügung.

Die Belegung ist bei beiden Boards identisch und in der folgenden Tabelle dargestellt:

lfd. Nr.	Anschluss XS1	Signalname	Bemerkung
1	1	CAN1_High	
2	14	CAN1_Low	
3	2	CAN2_High	
4	15	CAN2_Low	
5	3	VBAT	Bezugspotenzial Plus Transceiver
6	16	GND	Massepotenzial Transceiver
7	4	K-Line1/ LIN1	abhängig von Transceiver-Bestückung
8	17	L-Line1/ WAKE1	abhängig von Transceiver-Bestückung
9	5	K-Line2/ LIN2/ J1850 PWM+	abhängig von Transceiver-Bestückung
10	18	L-Line2/ WAKE2/ J1850 PWM-	abhängig von Transceiver-Bestückung
11	6	J1850 VPW	
12	19	-	Bitte nicht belegen!
13	7	VBAT	Bezugspotenzial Plus Transceiver
14	20	GND	Massepotenzial Transceiver
15	8	Analog Input1	
16	21	Analog Input2	
17	9	Digital Input1	
18	22	Digital Output1	
19	10	Digital Input2	
20	23	Digital Output2	
21	11	Digital Input 3	
22	24	Digital Output3	
23	12	Digital Input 4	
24	25	Digital Output4	
25	13	Weckleitung	



Bei K-Line sind ggf. auf PIN 17/ 18 die Anschlüsse für die L-Line verdrahtet (in Abhängigkeit von der Ausgangsbeschaltung).



Bei LIN sind ggf. auf PIN 17/ 18 die WAKE-Anschlüsse verdrahtet (in Abhängigkeit von der Transceiver-Wahl).



Pin 3 und 7 sowie Pin 16 und 20 sind auf dem Board gebrückt!

2.3.6 LED Anzeige

Die LEDs zeigen folgende Zustände an:

Rote LED D100: /HDRST Hardware Reset Indication-Ausgang des Microcontrollers

Grüne LED D700: Zustandsanzeige Spannung 5V

Grüne LED D701: Zustandsanzeige Spannung 3,3V

Grüne LED D702: Zustandsanzeige Spannung 2,5V

Gelbe LED D801: Microcontroller-Zustandsanzeige CAN 1

Gelbe LED D802: Microcontroller-Zustandsanzeige CAN 2

Gelbe LED D803: Microcontroller-Zustandsanzeige K-Line/LIN 1

Gelbe LED D804: Microcontroller-Zustandsanzeige K-Line/LIN 2

Die LEDs sind auf der Frontplatte folgendermaßen angeordnet:

D702 D701 D700 D100

D801 D802 D803 D804

2.4 Lieferhinweise

PXI/ PCI 3080-Boards werden in folgenden Basisvarianten geliefert:

- ♦ 1x CAN Schnittstelle und 1x LIN Schnittstelle oder
- ♦ 1x CAN Schnittstelle und 1x K-Line Schnittstelle

Diese Basisvarianten können durch folgende Optionen erweitert werden:

- ♦ 1x Zusätzliche CAN Schnittstelle
- ♦ 1x Zusätzliche LIN Schnittstelle oder K-Line Schnittstelle
- ♦ 1x Zusätzliche J1850 VPW Schnittstelle
- ♦ 1x Zusätzliche J1850 PWM Schnittstelle



Wenn Sie die Option 1x Zusätzliche J1850 PWM Schnittstelle wählen, ist die Option 1x Zusätzliche LIN Schnittstelle oder K-Line Schnittstelle NICHT möglich.

Außer der Auswahl der Schnittstelle selbst muss auch der Typ des zugehörigen CAN/ LIN/ K-Line/ J1850 Transceivers für jede Schnittstelle festgelegt werden.

Für jede CAN/ LIN/ K-Line/ J1850-Schnittstelle sind außerdem die erforderlichen Funktionalitäten anzugeben.

3 Ansteuersoftware

Zur Einbindung der PXI 3080/ PCI 3080-Hardware in eigene Applikationen existieren drei Möglichkeiten:

- ♦ [Programmieren über G-API](#)
- ♦ [Programmieren über DLL-Funktionen](#)
- ♦ [Programmieren mit LabVIEW](#)

3.1 Programmieren über G-API

Die G_API (GÖPEL-API) ist eine auf „C“ basierende Programmier-Umgebung für GÖPEL electronic-Hardware unter Windows® und die bevorzugte Programmier-Umgebung für diese Hardware. Sie stellt einen umfangreichen, Hardware-unabhängigen Befehlssatz für CAN, LIN, K-Line, FlexRay, MOST, LVDS, analoge und digitale Ein-/Ausgänge sowie Diagnosedienste zur Verfügung. Egal ob ein PXI-/ PCI-, USB- oder Ethernet-Gerät genutzt wird – die Befehle sind dieselben.

Die mit der G-API einher gehende Hardware-Abstraktion erlaubt der Testapplikation Parallelzugriff auf die Hardware. Das ermöglicht einer Applikation den Zugriff auf mehrere Hardware-Schnittstellen. Andererseits können auch mehrere Applikationen parallel auf die gleiche Hardware-Schnittstelle zugreifen.

Ein weiteres Feature der G-API ist der asynchrone Hardware-Zugriff. Das bedeutet: Keine Ausführungs-Einschränkungen für wartende Firmwarebefehle. Die Befehls-Quittierung wird über einen Callback-Mechanismus geliefert.

Mit dem Hardware Explorer stellt die GÖPEL electronic GmbH ein Hardware Konfigurations- und Management-Tool zur Verfügung, das den Anwendern die bequeme Möglichkeit bietet, ihre Hardware-Konfigurationen zu verwalten und auf die einzelnen Hardware-Schnittstellen über logische Namen zuzugreifen. Durch die Verwendung logischer Namen ist ein erneutes Compilieren der Applikation beim Wechsel auf eine andere Schnittstelle oder ein anderes Controllerboard nicht mehr erforderlich: Die Schnittstellen können im Hardware Explorer einfach neu zugeordnet werden. Außerdem bietet der Hardware Explorer eine einfache Möglichkeit, das Zusammenwirken von Hard- und Software durch die Ausführung integrierter Selbsttests zu überprüfen.

Die folgende Abbildung zeigt den GÖPEL electronic Hardware Explorer:

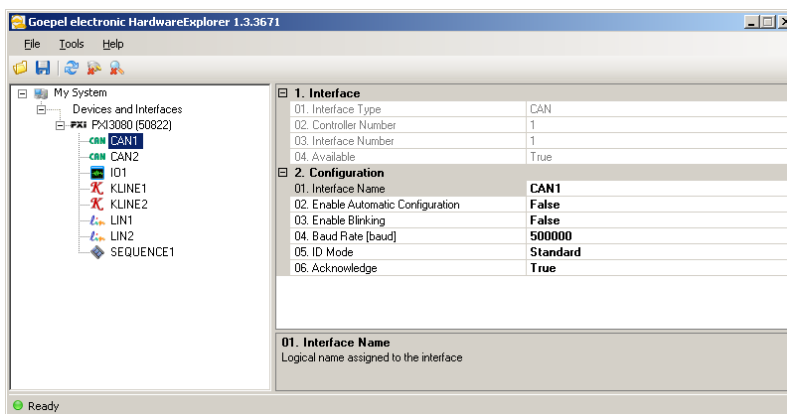


Abbildung 3-1
Hardware Explorer



Bitte vergleichen Sie die G-API Dokumentation für weitere Informationen.

Diese Dokumentation und die Installationssoftware finden Sie im Ordner *G-API* der mitgelieferten PXI Produkt CD.

3.2 Programmieren über DLL-Funktionen



Die Programmierung über DLL-Funktionen ist weiterhin für bestehende Projekte möglich, bei denen noch nicht mit der GÖPEL electronic Programmierumgebung G-API gearbeitet werden kann.

Die Dokumentation GÖPEL Firmware senden wir Ihnen auf Anforderung gern zu. Bitte setzen Sie sich bei Bedarf mit unserem Vertrieb in Verbindung.



Die Begriffe GPxi3080 und PXI3080 in der folgenden Funktionsbeschreibung stehen für PXI 3080/ PCI 3080.

Informationen zu den Strukturen, Datentypen und Error-Codes enthalten die Header – die entsprechenden Dateien finden Sie auf der mitgelieferten CD.

3.2.1 Windows Device Treiber

Die für die Programmierung unter Verwendung des Windows Device Treibers nutzbaren DLL-Funktionen sind in den folgenden Abschnitten beschrieben:

- ◆ [Driver Info](#)
- ◆ [DLL Version](#)
- ◆ [XILINX Download](#)
- ◆ [XILINX Write Data](#)
- ◆ [DPRAM Write Instruction](#)
- ◆ [DPRAM Read Response](#)
- ◆ [Reset Port](#)

3.2.1.1 Driver Info Die Funktion `GPxi3080_GetDriverInfo` dient zur Status-Abfrage des Hardware-Treibers.

Format:

```
int GPxi3080_GetDriverInfo(GPxi3080_StructDriverInfo *pDriverInfo);
```

Parameter:

Zeiger, z.B. `pDriverInfo`, auf eine Datenstruktur

Zur Struktur siehe das File `GPxi3080.h` auf der mitgelieferten CD

Beschreibung:

Die Funktion `GPxi3080_GetDriverInfo` gibt Informationen über den Status des Hardware-Treibers zurück.

Dazu muss der Funktion die Adresse eines Zeigers `pDriverInfo` übergeben werden.

Innerhalb der Funktion wird die Struktur, auf die `pDriverInfo` zeigt, mit verschiedenen Informationen gefüllt.

3.2.1.2 DLL Version Die Funktion `GPxi3080_DLL_Version` dient zur Abfrage der Versionsnummer der DLL.

Format:

```
int GPxi3080_DLL_Version(unsigned long *pVersion);
```

Parameter

Zeiger, z.B. `pVersion`, auf die Versionsnummer

Beschreibung:

Die Funktion `GPxi3080_DLL_Version` gibt die Versionsnummer der *GPxi3080w.dll* als Integer-Wert zurück.

Beispiel:

Die Versionsnummer **1.23** wird als Wert **123** zurückgegeben, Version **1.60** als Wert **160**.

3.2.1.3 XILINX Download

Die Funktion `GPxi3080_XilinxDownload` dient zum Laden eines FPGA-Files in den XILINX.

Format:

```
int GPxi3080_XilinxDownload(unsigned long Device, char *pFileName);
```

Parameter:

Device

Index des PXI/ PCI 3080-Boards, links beginnend mit 1

Zeiger, z.B. `pFileName`, auf den Pfad des zu ladenden FPGA-Files

Beschreibung:

Die Funktion `GPxi3080_XilinxDownload` dient zum Laden eines FPGA-Files (Extension `*.cdf`), das unter anderem das Auslesen der geografischen Slotadresse im PXI-Rack ermöglicht.

Die geladenen Daten sind flüchtig. Deshalb muss die Funktion nach Power Off erneut ausgeführt werden.



Nach `XilinxDownload` ist eine Wartezeit von ca. 500 ms erforderlich, da der Controller ein Power-On-Reset durchläuft.

Anschließend ist der Firmware-Befehl `0x10 Software Reset` auszuführen, um vom Bootloader-Modus in den Normal-Modus zu gelangen.

3.2.1.4 XILINX Write Data

Die Funktion `GPxi3080_XilinxWriteData` ermöglicht das Konfigurieren und Ausführen von Funktionen, die der XILINX bereitstellt.

Format:

```
int GPxi3080_XilinxWriteData(unsigned char *data, unsigned long *length);
```

Parameter:

Zeiger, z.B. `data`, auf den Bereich für Schreibdaten
(z. Zt. max. 128 Byte pro Befehl)

length

Größe des Speicherbereiches, auf den `data` zeigt, in Bytes

Beschreibung:

Bevor die Funktionalität des XILINX genutzt werden kann, muss das zugehörige FPGA-File mit `GPxi3080_XilinxDownload` geladen worden sein (siehe [XILINX Download](#)).

Das Datenformat besteht aus vier Bytes einschließlich Befehl.
Falls erforderlich, können Parameter-Bytes folgen.

Datenformat:

1. Byte: `0x48` (StartByte)
2. Byte: Device (Index des PXI/ PCI 3080-Boards,
links beginnend mit 1)
3. Byte: `0x00` (Reserviertes Byte)
4. Byte: XILINX Befehl

z. Zt. unterstützter XILINX Befehl:

`0x10` PowerOnReset für das komplette Board

3.2.1.5 DPRAM Write Instruction

Die Funktion `GPxi3080_DpramWriteInstruction` dient zum Senden eines Befehls zum PXI/ PCI 3080-Controller.

Format:

```
int GPxi3080_DpramWriteInstruction(unsigned char *data, unsigned long length);
```

Parameter:

Zeiger, z.B. `data`, auf den Bereich für Schreibdaten, bestehend aus Befehlskopf und Befehlsbytes (z. Zt. max. 1024 Byte pro Befehl)

`length`

Größe des Speicherbereiches, auf den `data` zeigt, in Bytes

Beschreibung:

Die Funktion `GPxi3080_DpramWriteInstruction` sendet einen Befehl zum PXI/ PCI 3080-Controller.

Im Header der Struktur, auf die `data` zeigt, befindet sich die Information zum anzusprechenden PXI/ PCI 3080-Board. Deshalb ist dieser Parameter nicht separat anzugeben.

3.2.1.6 DPRAM Read Response

Die Funktion `GPxi3080_DpramReadResponse` dient zum Lesen einer Antwort vom PXI/ PCI 3080-Controller.

Format:

```
int GPxi3080_DpramReadResponse(unsigned long card,  
                               unsigned char *data,  
                               unsigned long *length);
```

Parameter:

`card`

Index des PXI/ PCI 3080-Boards, links beginnend mit 1

Zeiger, z.B. `data`, auf den Bereich für Lesedaten, bestehend aus `Antwortkopf` und `Antwortbytes` (z. Zt. max. 1024 Byte pro Antwort)

`length`

Parameterwert vor Funktionsaufruf:

Größe des Puffers, auf den `data` zeigt, in Bytes

Parameterwert nach Funktionsaufruf:

Tatsächlich gelesene Byteanzahl

Beschreibung:

Die Funktion `GPxi3080_DpramReadResponse` liest die älteste vom PXI/ PCI 3080-Controller im `Response`-Bereich des DPRAM geschriebene Antwort zurück.

Werden mehrere Antworten vom Controller bereitgestellt, ohne sie zu lesen, gehen diese nicht verloren, sondern werden in einer Art Liste abgelegt.

Aufrufe von `GPxi3080_DpramReadResponse` liefern dann solange Werte, bis diese Liste keine Einträge mehr enthält.

3.2.1.7 Reset Port Die Funktion GPxi3080_ResetPort dient zur Auslösung eines Software-Resets für den PXI/ PCI 3080-Controller.

Format

```
int GPxi3080_ResetPort(unsigned long card);
```

Parameter:

card

Index des PXI/ PCI 3080-Boards, links beginnend mit 1

Beschreibung:

Die Funktion GPxi3080_ResetPort löst ein Software-Reset für den PXI/ PCI 3080-Controller aus.

Dieser Auslösemechanismus erfolgt über einen separaten Interruptkanal und nicht über den Befehlsinterpreter der Software (Firmware-Befehl 0x10 Software Reset).

3.2.2 VISA Device Treiber

Die für die Programmierung unter Verwendung des VISA Device Treibers nutzbaren DLL-Funktionen sind in den folgenden Abschnitten beschrieben:

- ◆ [Init](#)
- ◆ [Done](#)
- ◆ [Driver Info](#)
- ◆ [XILINX Download](#)
- ◆ [XILINX Write Data](#)
- ◆ [Write Data](#)
- ◆ [Read Data](#)
- ◆ [Reset Port](#)

3.2.2.1 Init Die Funktion `PXI3080_Init` dient zur Eröffnung von VISA Sessions für alle im System befindlichen PXI/ PCI 3080-Boards und deren Initialisierung.

Format:

```
ViStatus PXI3080_Init(ViUInt32 *CardCount);
```

Parameter:

CardCount

Anzahl der vom VISA Treiber erkannten PXI/ PCI 3080-Boards im System.

Beschreibung:

Die Funktion `PXI3080_Init` sucht alle im System befindlichen PXI/ PCI 3080-Boards und eröffnet die erforderlichen Sessions. Außerdem werden Board-interne Initialisierungen durchgeführt. Deshalb muss diese Funktion als erster Schritt ausgeführt werden.

3.2.2.2 Done Die Funktion `PXI3080_Done` schließt alle VISA Sessions für im System befindliche PXI/ PCI 3080-Boards.

Format:

```
ViStatus PXI3080_Done(void);
```

Parameter:

keine

Beschreibung:

Die Funktion `PXI3080_Done` schließt alle VISA Sessions für im System befindliche PXI/ PCI 3080-Boards.

Damit ist kein weiterer Boardzugriff möglich.

3.2.2.3 Driver Info Die Funktion `PXI3080_DriverInfo` liefert allgemeine Treiber- und Boardinformationen.

Format:

```
ViStatus PXI3080_DriverInfo(PXI3080_StructDriverInfo *DriverData,  
                           ViChar *DeviceName);
```

Parameter:

Zeiger, z.B. `DriverData`, auf eine Datenstruktur
Zur Struktur siehe das File `PXI3080_API.h` der mitgelieferten CD

`DeviceName`

`Array[K_DEV_MAX][K_RES_NAME_LENGTH]`
(siehe `PXI3080_API.h`)

Beschreibung:

Die Funktion `PXI3080_DriverInfo` stellt verschiedene Informationen zum Treiber und zu den im System befindlichen PXI/ PCI 3080-Boards zur Verfügung.

Der `DeviceName` gibt die von VISA erfassten Ressourcennamen an. Diese Informationen korrelieren mit der Anzeige im NI MAX.

3.2.2.4 XILINX Download

Die Funktion `PXI3080_XilinxDownload` dient zum Laden eines FPGA Files in den XILINX.

Format:

```
ViStatus PXI3080_XilinxDownload(ViUInt32 Card, ViChar *FileName);
```

Parameter:

Card

Index des PXI/ PCI 3080-Boards, links beginnend mit 1

Zeiger, z.B. `FileName`, auf den Pfad des zu ladenden FPGA Files

Beschreibung:

Die Funktion `PXI3080_XilinxDownload` dient zum Laden eines FPGA-Files (Extension `*.cdf`), das unter anderem das Auslesen der geografischen Slotadresse im PXI-Rack ermöglicht.

Die geladenen Daten sind flüchtig. Deshalb muss die Funktion nach Power Off erneut ausgeführt werden.



Nach `XilinxDownload` ist eine Wartezeit von ca. 500 ms erforderlich, da der Controller ein Power-On-Reset durchläuft.

Anschließend ist der Firmware-Befehl `0x10 Software Reset` auszuführen, um vom Bootloader-Modus in den Normal-Modus zu gelangen.

3.2.2.5 XILINX Write Data

Die Funktion `PXI3080_XilinxWriteData` ermöglicht das Konfigurieren und Ausführen von Funktionen, die der XILINX bereitstellt.

Format:

```
ViStatus PXI3080_XilinxWriteData(ViUInt8 *Data);
```

Parameter:

Zeiger, z.B. `Data`, auf den Bereich für Schreibdaten
(z. Zt. max. 128 Byte pro Befehl)

Beschreibung:

Bevor die Funktionalität des XILINX genutzt werden kann, muss das zugehörige FPGA-File mit `PXI3080_XilinxDownload` geladen worden sein (siehe [XILINX Download](#)).

Das Datenformat besteht aus vier Bytes einschließlich Befehl.
Falls erforderlich, können Parameter-Bytes folgen.

Datenformat:

1. Byte: `0x48` (StartByte)
2. Byte: `card` (Index des PXI/ PCI 3080-Boards, links beginnend mit 1)
3. Byte: `0x00` (Reserviertes Byte)
4. Byte: XILINX Befehl

z. Zt. unterstützter XILINX Befehl:

`0x10` PowerOnReset für das komplette Board

3.2.2.6 Write Data Die Funktion `PXI3080_WriteData` dient zum Schreiben von Daten für den `PXI/ PCI 3080`-Controller.

Format:

```
ViStatus PXI3080_WriteData(ViUInt8 *WriteData, ViUInt32 Length_In_Bytes);
```

Parameter:

Zeiger, z.B. `WriteData`, auf den Bereich für Schreibdaten, bestehend aus `Befehlskopf` und `Befehlsbytes` (z. Zt. max. 1024 Bytes pro Befehl)

`Length_In_Bytes`

Größe des Speicherbereiches, auf den `WriteData` zeigt, in Bytes

Beschreibung:

Die Funktion `PXI3080_WriteData` ermöglicht das Schreiben von Daten zu den `PXI/ PCI 3080`-Boards.

Im Header der Struktur, auf die `WriteData` zeigt, befindet sich die Information zum anzusprechenden Board.

Deshalb ist dieser Parameter nicht gesondert anzugeben.

3.2.2.7 Read Data

Die Funktion `PXI3080_ReadData` dient zum Lesen von Daten des PXI/ PCI 3080-Controllers.

Format:

```
ViStatus PXI3080_ReadData(ViUInt32 Card, ViUInt8 *ReadData, ViUInt32 *Length);
```

Parameter:

Card

Index des PXI/ PCI 3080-Boards, links beginnend mit 1

Zeiger, z.B. `ReadData`, auf den Bereich für Lesedaten, bestehend aus `Antwortkopf` und `Antwortbytes` (z. Zt. max. 1024 Bytes pro Antwort)

Length

Parameterwert vor Funktionsaufruf:

Größe des Puffers, auf den `ReadData` zeigt, in Bytes;

Parameterwert nach Funktionsaufruf:

Anzahl der tatsächlich gelesenen Bytes

Beschreibung:

Die Funktion `PXI3080_ReadData` ermöglicht das Lesen von Daten, die von einem PXI/ PCI 3080-Board bereitgestellt wurden (siehe auch Funktion `GPxi3080_DpramReadResponse` im Abschnitt [Windows Device Treiber](#)).

3.2.2.8 Reset Port Die Funktion `PXI3080_ResetPort` dient zur Auslösung eines Software-Resets für den PXI/ PCI 3080-Controller.

Format

```
ViStatus PXI3080_ResetPort(ViUInt32 Card);
```

Parameter:

Card

Index des PXI/ PCI 3080-Boards, links beginnend mit 1

Beschreibung:

Die Funktion `GPxi3080_ResetPort` löst ein Software-Reset für den PXI/ PCI 3080-Controller aus.

Dieser Auslösemechanismus erfolgt über einen separaten Interruptkanal und nicht über den Befehlsinterpreter der Software (Firmware-Befehl `0x10 Software Reset`).

3.3 Programmieren mit LabVIEW

3.3.1 LabVIEW über G-API

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe PXI/ PCI 3080-Boards unter LabVIEW angesprochen werden können.

Dabei nutzen die LabVIEW VIs die Funktionen der GÖPEL G-API.

3.3.2 LLB unter Verwendung des Windows Device Treibers

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe PXI/ PCI 3080-Boards unter LabVIEW angesprochen werden können.

Dabei werden die Funktionen genutzt, die im Abschnitt [Windows Device Treiber](#) beschrieben worden sind.

3.3.3 LLB unter Verwendung des VISA Device Treibers

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe PXI/ PCI 3080-Boards unter LabVIEW angesprochen werden können.

Dabei werden die Funktionen genutzt, die im Abschnitt [VISA Device Treiber](#) beschrieben worden sind.

3.4 Weitere GÖPEL Software

PROGRESS, Programm-Generator und myCAR der GÖPEL electronic GmbH sind komfortable Programme zur Prüfung mit GÖPEL-Hardware.

Weitere Informationen zur Nutzung dieser Programme finden Sie in den entsprechenden Softwarebeschreibungen.

G

G-API3-2

H

Hardware Explorer3-2

PPXI/ PCI 3080
Hardware Installation1-1

R

Ressourcen2-1

SSteckverbinder
Front2-11

V

VISA Treiber3-12

W

Windows Treiber1-2, 3-4