

Produktbeschreibung

PXI / PCI 3052

CAN Interfaces
Nutzerhandbuch
Version 1.3



GÖPEL 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: 02.03.2012

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

Stand: März 2012

1	INSTALLATION DER BOARDS	1-1
1.1	HARDWARE INSTALLATION	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 3052	2-1
2.1	BESTIMMUNG	2-1
2.2	TECHNISCHE DATEN	2-4
2.2.1	<i>Allgemeines</i>	2-4
2.2.2	<i>Abmessungen</i>	2-4
2.2.3	<i>PXI 3052/ PCI 3052 Kennwerte</i>	2-4
2.3	AUFBAU	2-5
2.3.1	<i>Allgemeines</i>	2-5
2.3.2	<i>Adressierung</i>	2-6
2.3.3	<i>Triggerverhalten</i>	2-6
2.3.4	<i>Kommunikationsschnittstellen</i>	2-7
2.3.5	<i>Bestückung</i>	2-8
2.3.6	<i>Belegung Frontsteckverbinder</i>	2-10
2.3.7	<i>LED Anzeige</i>	2-11
2.4	LIEFERHINWEISE	2-12
3	ANSTEUERSOFTWARE	3-1
3.1	PROGRAMMIEREN ÜBER G-API	3-1
3.2	PROGRAMMIEREN ÜBER DLL-FUNKTIONEN	3-1
3.2.1	<i>Windows Device Treiber</i>	3-2
3.2.1.1	<i>DriverInfo</i>	3-3
3.2.1.2	<i>DLL Version</i>	3-4
3.2.1.3	<i>XILINX Download</i>	3-5
3.2.1.4	<i>XILINX Write Data</i>	3-6
3.2.1.5	<i>DPRAM Write Instruction</i>	3-7
3.2.1.6	<i>DPRAM Read Response</i>	3-8
3.2.1.7	<i>DPRAM Read Monitor</i>	3-9
3.2.1.8	<i>Reset Port</i>	3-10
3.2.2	<i>VISA Device Treiber</i>	3-11
3.2.2.1	<i>Init</i>	3-12
3.2.2.2	<i>Done</i>	3-12
3.2.2.3	<i>Driver Info</i>	3-13
3.2.2.4	<i>XILINX Download</i>	3-14
3.2.2.5	<i>XILINX Write Data</i>	3-15
3.2.2.6	<i>Write Data</i>	3-16
3.2.2.7	<i>Read Data</i>	3-17
3.2.2.8	<i>Read Monitor</i>	3-18
3.2.2.9	<i>Reset Port</i>	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 Hardware Installation



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.



Elektrostatische Entladungen (ESD) können Ihr System schädigen und elektronische Bauelemente zerstören. Das kann zu irreparablen Schäden am PXI/ PCI 3052-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 3052-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 *GPxi3052* enthaltenen *inf*-Datei kann der Hardwareassistent die Installation des Devicetreibers durchführen.

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

- ♦ *GPxi3052.inf* für Windows® 2000/ XP im Ordner *Win2000 (Version xx)*
- ♦ *GPxi3052_x64.inf* für Windows® 7/ 64 Bit im Ordner *Win7_x64 (Version xx)*

Ein Neustart des Systems ist nicht zwingend erforderlich.

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.



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

1.2.2 VISA Device Treiber

1. Schritt

Kopieren Sie den Ordner *VISA (Version xx)* aus dem Ordner *GPxi3052* der mitgelieferten CD auf die Festplatte.

(Empfehlung: vollständigen Ordner auf *C:*)

2. Schritt

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 *PXI3052_NT5.inf* befindet (nach Empfehlung: *C:\VISA (Version xx)\Installation*).

LabViewRT :

Für den Einsatz der PXI/ PCI 3052-Boards unter dem RT Betriebssystem muss die Datei *P3052_RT.inf* im Verzeichnis *C:\VISA (Version xx)\Installation* verwendet werden.

Kopieren Sie diese Datei in das Verzeichnis *\ni-rt\system* des embedded controllers (Empfehlung: über den NI Measurement Explorer).



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 Treiberinstallation können Sie überprüfen, ob die Boards einwandfrei vom System eingebunden worden sind:

Abbildung 1-1:
Windows

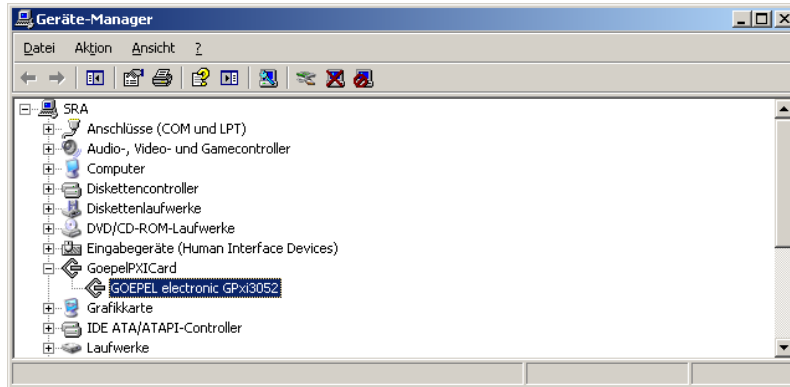


Abbildung 1-2:
Visa

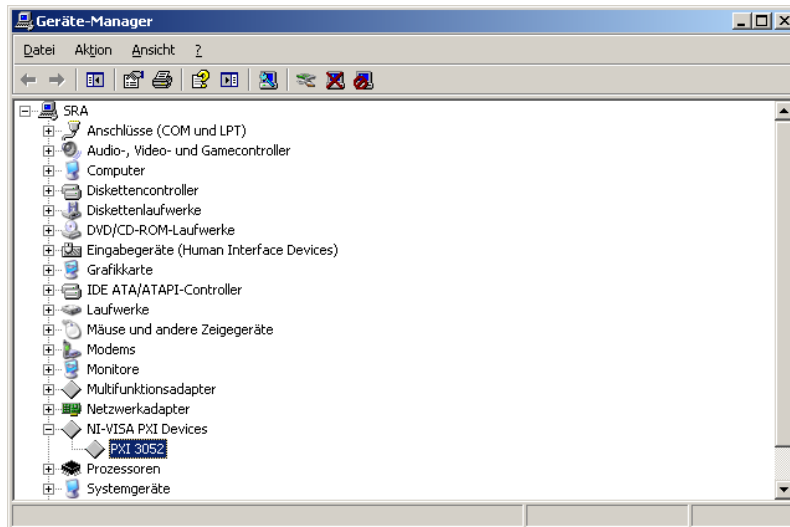
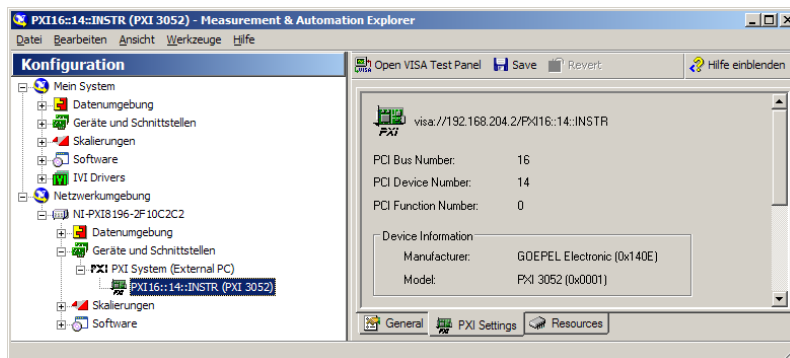


Abbildung 1-3:
Visa RT



2 Hardware PXI/ PCI 3052

2.1 Bestimmung

Die CAN Interface Boards PXI 3052/ PCI 3052 sind Kommunikationsboards der GÖPEL electronic GmbH.

Diese Boards werden in der allgemeinen Steuerungstechnik verwendet, speziell in der Automobiltechnik. Sie haben folgende Eigenschaften:

- ♦ 2 bis 4 CAN Schnittstellen Version 2.0b je nach Ausbaustufe
- ♦ erweiterte Triggerfunktionen mit je einem Triggerinput und Triggeroutput zum Frontsteckverbinder
- ♦ Abschaltmöglichkeit der CAN Sendepfade ohne Verlust der Empfangsdaten, wenn kein CAN-Acknowledge empfangen wird (Firmware-Befehl 0x1C CAN Hardware-Sendepfad steuern)
- ♦ galvanische Trennung der CAN Schnittstellen vom PXI- bzw. PCI-Interface der Boards
- ♦ jede CAN Schnittstelle verfügt über einen 32 Bit µController (TriCore TC1765, 40MHz) mit 2 Mbyte RAM
- ♦ Visualisierung der Controllerzustände auf der Frontplatte mit je zwei LEDs (siehe [LED Anzeige](#))
- ♦ Hohe Flexibilität durch steckbare Transceivermodule



In diesem Nutzerhandbuch ist unter Controller IMMER der einer CAN Schnittstelle zugeordnete Microcontroller zu verstehen (unabhängig von der Bezeichnung „CAN Controller“ für das gesamte Board auf der Frontplatte).



Abbildung 2-1:
PXI 3052 mit CAN 1 und 2

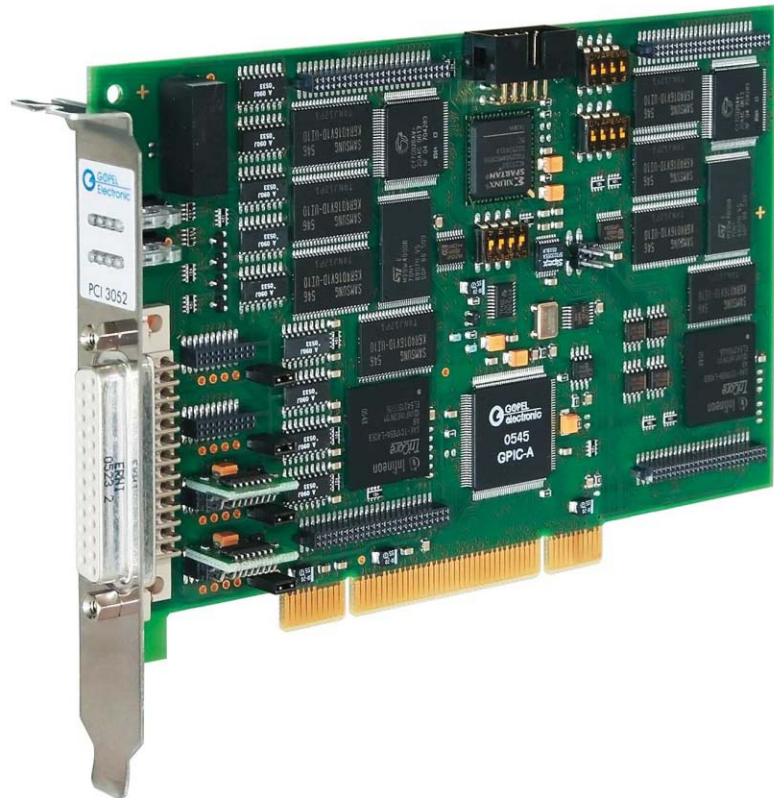


Abbildung 2-2:
PCI 3052 mit CAN 1 und 2



Beachten Sie bitte, dass ein Download des Xilinx FPGAs für die Funktion der Boards unabdingbar ist (siehe [XILINX Download](#) für verschiedene Treiber unter [Programmieren über DLL-Funktionen](#))!

2.2 Technische Daten

2.2.1 Allgemeines

Das Kommunikationsboard PXI 3052 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 3052 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 3052 CAN Interface Board: 160 mm x 100 mm (L x B)
- ◆ PCI 3052 CAN Interface Board: 168 mm x 106 mm (L x B)

2.2.3 PXI 3052/ PCI 3052 Kennwerte

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bemerkung
U _{BAT}	Batteriespannung		12	27/ 50	V	abh. v. Transceivertyp
	Übertragungsrate			1	MBaud	
R _{bus}	Abschlusswiderstand 1		120		Ohm	Jumper J1..J4 gesteckt
R _{bus}	2 x Abschlusswiderstände 2		5,1		kOhm	auf Transceiver-Modul
	Externer Triggereingang	3,3		50	V	
	Externer Triggenerausgang			U _{BAT}	V	Open Collector Ausgang über npn-Transistor: max. 50mA/ 200mW



Um am externen Triggenerausgang (open Collector) einen Spannungshub zu erzeugen, muss ein externer Pullup-Widerstand über eine Spannungsquelle an diesen Ausgang geschaltet werden (z. B. 10kΩ über U_{Bat}).

2.3 Aufbau

2.3.1 Allgemeines

Beide Boards verfügen in der Basisversion über zwei CAN Schnittstellen der Version 2.0b und können über Aufsatzboards und weitere Transceiver auf insgesamt vier CAN Schnittstellen erweitert werden.

Abbildung 2-3 zeigt den schematischen Aufbau der Boards in einem Blockschaltbild.

Bei den PXI/ PCI 3052-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 3052 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 (Trig. connector in Abbildung 2-5).

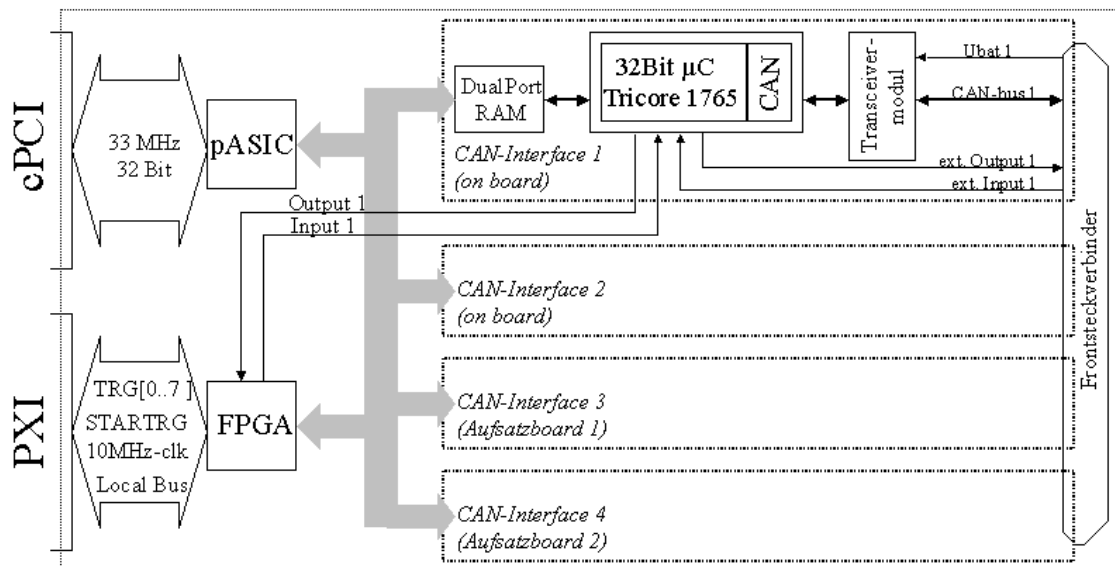


Abbildung 2-3: Blockdiagramm eines Kommunikationsboards PXI/ PCI 3052

2.3.2 Adressierung

PXI 3052: 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 3052 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 [Programmieren über DLL-Funktionen](#)).

PCI 3052: PCI-Racks besitzen keine geographische Slotadressierung. Um dennoch mehrere PCI 3052 auf den Steckplätzen eindeutig identifizieren zu können, verfügt das Board über ein separates Adressjumper-Feld (Address jumper in Abbildung 2-5). Damit ist es möglich, bis zu 16 Adressvarianten zu wählen. Der mit diesem Jumperfeld gesetzte Binärwert (0..15) kann mit der Software ausgelesen werden.

2.3.3 Triggerverhalten

Jede CAN Schnittstelle besitzt 2 x 2 zusätzliche Input- bzw. Outputleitungen.

Je ein Input und Output pro CAN Schnittstelle befindet sich auf dem Frontsteckverbinder. Der jeweils zweite In- und Output kann über die entsprechende Treiberkonfiguration mit den PXI-Signalen Startrigger und Trigger[0..7] bzw. den beiden zusätzlichen I/O-Leitungen des PCI-Boards verschaltet werden.

Die Firmware der Controller kann so konfiguriert werden, dass die Schnittstelle entweder zu den externen Triggersignalen oder zu den PXI-Triggersignalen aktiviert ist.

Die folgenden Funktionen sind möglich:

- ◆ ENABLE-Funktion:
ein von außen anliegendes Inputsignal aktiviert/ deaktiviert die Möglichkeit des Sendens von Botschaften einer CAN Schnittstelle
- ◆ TRIGGER_IN-Funktion:
ein von außen anliegendes Triggersignal löst das Senden von zuvor definierten vorbereiteten CAN Botschaften aus
- ◆ TRIGGER_OUT-Funktion:
es wird ein Ausgangssignal beim Senden bzw. Empfang einer bestimmten Botschaft erzeugt.

2.3.4 Kommunikations-schnittstellen

Maximal 4 x CAN Schnittstellen Version 2.0b:

Für die uneingeschränkte Funktion einer CAN Schnittstelle in 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 3052-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 Transceiver verwendet, ist i. Allg. ein 120 Ohm Widerstand für jede CAN-Schnittstelle auf dem Board bestückt. Diese Widerstände können durch Ziehen der Jumper J1..J4 deaktiviert und durch bedrahtete Widerstände mit dem gewünschten Wert an den Positionen RP11, 12 bis RP41, 42 ersetzt werden (Abbildung 2-4 und Abbildung 2-5).

Bei Verwendung von Lowspeed Transceivern befinden sich zwei Abschlusswiderstände von 5,1 kOhm für RTH und RTL pro CAN Schnittstelle auf dem Transceiver-Modul. In diesem Fall darf KEIN bedrahteter Widerstand bestückt werden, und der Jumper muss geöffnet sein.

Für die folgenden Transceivertypen ist der Anschluss der Batteriespannung an die Pins 15, 18, 21 bzw. 24 des Steckverbinders XS1 (V_Bat1..V_Bat4, siehe [Belegung Frontsteckverbinder](#)) für die jeweilige CAN-Schnittstelle notwendig:

- ◆ TJA1041A
- ◆ TJA1054
- ◆ PCA82C252
- ◆ B10011S

2.3.5 Bestückung

Abbildung 2-4 und Abbildung 2-5 zeigen schematisch die Bestückungsseite der Boards. In diesen Abbildungen ist die Lage der optionalen Aufsatzboards sowie der Transceivermodule für jede Schnittstelle zu erkennen. Die Positionen der Jumper J1..J4 für die Aktivierung/ Deaktivierung der Bus-Abschlusswiderstände bzw. der optional zu bestückenden bedrahteten Widerstände sind ebenfalls ersichtlich. Ein gesteckter Jumper bedeutet, dass der Bus-Abschlusswiderstand von 120 Ohm aktiv ist.

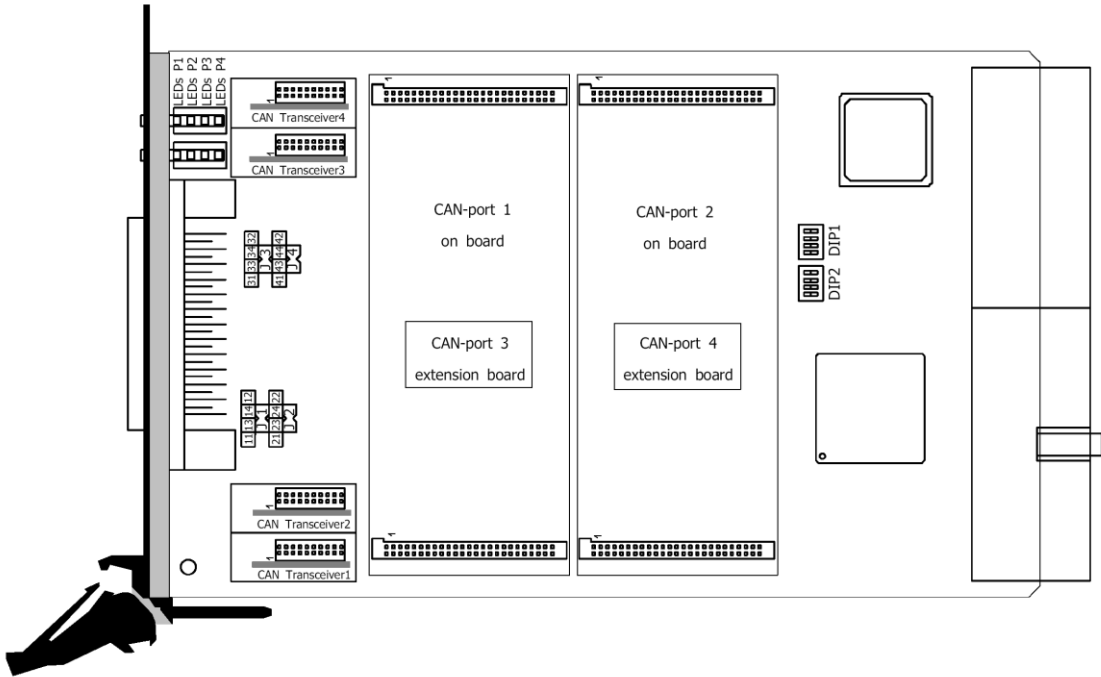


Abbildung 2-4: Schematischer Bestückungsplan Kommunikationsboard PXI 3052

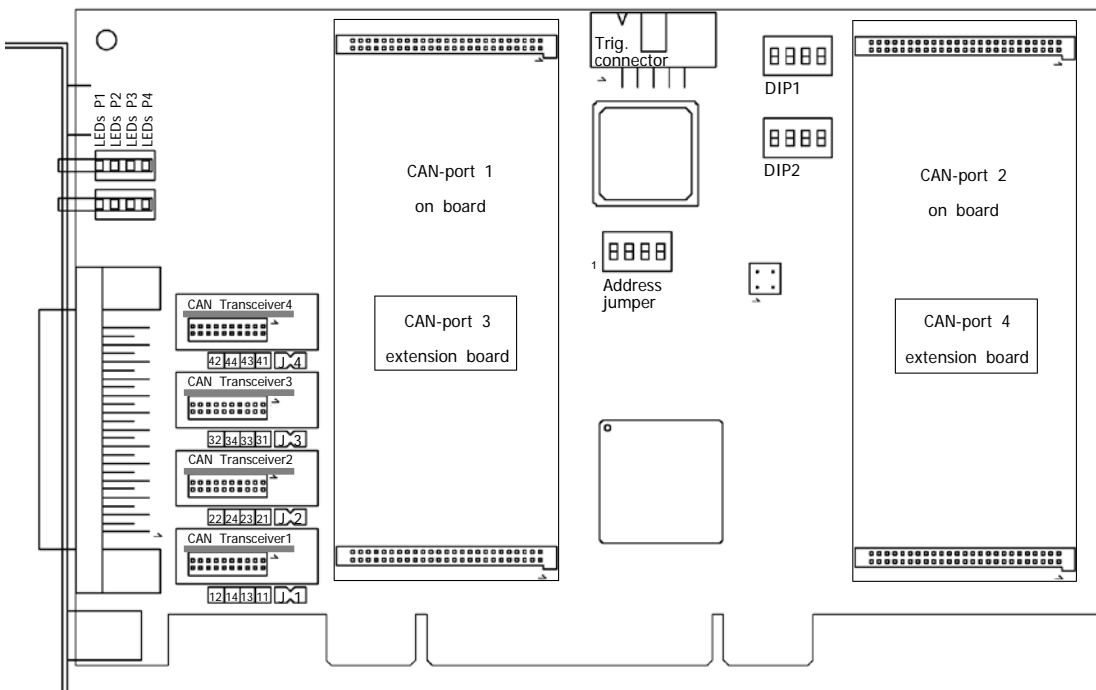


Abbildung 2-5: Schematischer Bestückungsplan Kommunikationsboard PCI 3052

Die Konfigurationselemente aus Abbildung 2-4 und Abbildung 2-5 werden in der folgenden Tabelle erläutert:

CAN Transc.1	Transceivermodul für CAN1
CAN Transc.2	Transceivermodul für CAN2
CAN Transc.3	Transceivermodul für CAN3
CAN Transc.4	Transceivermodul für CAN4
J1	Jumper zum Aktivieren des 120Ω Busabschlusswiderstandes (onboard) für CAN1
J2	Jumper zum Aktivieren des 120Ω Busabschlusswiderstandes (onboard) für CAN2
J3	Jumper zum Aktivieren des 120Ω Busabschlusswiderstandes (onboard) für CAN3
J4	Jumper zum Aktivieren des 120Ω Busabschlusswiderstandes (onboard) für CAN4
RP 11, 12	Position für optionalen bedrahteten Abschlusswiderstand – CAN1
RP 21, 22	Position für optionalen bedrahteten Abschlusswiderstand – CAN2
RP 31, 32	Position für optionalen bedrahteten Abschlusswiderstand – CAN3
RP 41, 42	Position für optionalen bedrahteten Abschlusswiderstand – CAN4
Address jumper	Das Jumperfeld auf der PCI 3052 dient der eindeutigen Adressierung des Boards in einem System mit mehreren PCI 3052 (analog „geografische Adressierung“ der PXI-Spezifikation). Dazu kann ein entsprechender Binärwert (0..15) über das Jumperfeld eingestellt werden, der über die mitgelieferte Software ausgelesen wird.
Trig. connector	Steckverbinder auf der PCI 3052 zum Austausch von Triggersignalen mit anderen GÖPEL electronic PCI-Boards

2.3.6 Belegung Frontsteckverbinder

Typ: DSub 25-polig Buchse

Die CAN Schnittstellen stehen über diesen Steckverbinder an der Frontseite der Kommunikationsboards zur Verfügung.

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

lfd. Nr.	Anschluss XS1	Signalname	Bemerkung
1	14	CAN1_High	CAN-Bus-Leitung High
2	2	CAN1_Low	CAN-Bus-Leitung Low
3	15	V_Bat1	Bezugspotenzial Plus Transceiver
4	1	GND	Massepotenzial Transceiver
5	17	CAN2_High	CAN-Bus-Leitung High
6	5	CAN2_Low	CAN-Bus-Leitung Low
7	18	V_Bat2	Bezugspotenzial Plus Transceiver
8	4	GND	Massepotenzial Transceiver
9	20	CAN3_High	CAN-Bus-Leitung High
10	8	CAN3_Low	CAN-Bus-Leitung Low
11	21	V_Bat3	Bezugspotenzial Plus Transceiver
12	7	GND	Massepotenzial Transceiver
13	23	CAN4_High	CAN-Bus-Leitung High
14	11	CAN4_Low	CAN-Bus-Leitung Low
15	24	V_Bat4	Bezugspotenzial Plus Transceiver
16	10	GND	Massepotenzial Transceiver
17	3	INPUT1	Input CAN-Controller 1
18	16	OUTPUT1	Output CAN-Controller 1
19	6	INPUT2	Input CAN-Controller 2
20	19	OUTPUT2	Output CAN-Controller 2
21	9	INPUT3	Input CAN-Controller 3
22	22	OUTPUT3	Output CAN-Controller 3
23	12	INPUT4	Input CAN-Controller 4
24	25	OUTPUT4	Output CAN-Controller 4
25	13	GND	Massepotenzial

2.3.7 LED Anzeige

Die auf der Frontplatte angeordneten Leuchtdioden geben Auskunft über den momentanen Betriebszustand des jeder CAN Schnittstelle (auch als „CAN Port“ bezeichnet) zugeordneten Controllers. Einer Schnittstelle sind je eine grüne und rote LED zugeteilt, die wie folgt angeordnet sind:

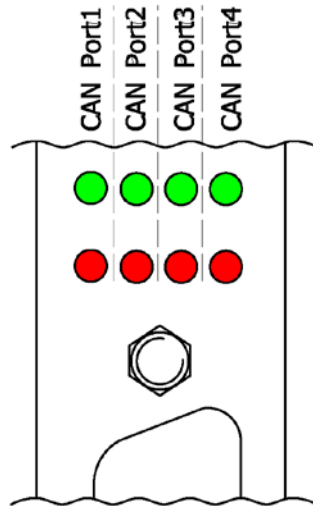


Abbildung 2-6:
LED Anzeige

Die Anzeigezustände dieser LEDs werden in einer Tabelle erläutert:

LED grün	LED rot	Bemerkung
leuchten dauerhaft		Controller läuft nicht (Xilinx Download ?)
blinken abwechselnd		Bootloadersoftware läuft (Softwarereset ?)
leuchten nicht		Firmware läuft
leuchtet (kurz)	leuchtet nicht	Während Firmwarebefehl abgearbeitet wird

2.4 Lieferhinweise

PXI/ PCI 3052-Boards werden in folgenden Varianten geliefert:

- ♦ 2x CAN Schnittstelle
- ♦ 3x CAN Schnittstelle
- ♦ 4x CAN Schnittstelle

Zur jeweiligen CAN Schnittstelle muss auch der Typ des zugehörigen Transceivers festgelegt werden.

Für jede CAN Schnittstelle sind außerdem die erforderlichen Funktionalitäten anzugeben.

3 Ansteuersoftware

Zur Einbindung der PXI 3052/ PCI 3052-Hardware in eigene Applikationen existieren drei Möglichkeiten:

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

3.1 Programmieren über G-API

Das bevorzugte User Interface für diese GÖPEL Hardware ist die G-API (GÖPEL-API).

Sie finden alle benötigten Informationen im Ordner *G-API* der mitgelieferten 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 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 *GPxi3052* und *PXI3052* in der folgenden Funktionsbeschreibung stehen für *PXI 3052/ PCI 3052*.

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



In diesem Nutzerhandbuch ist unter *Controller IMMER* der einer CAN-Schnittstelle zugeordnete Microcontroller zu verstehen (unabhängig von der Bezeichnung „CAN Controller“ für das gesamte Board auf der Frontplatte).

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:

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

3.2.1.1 DriverInfo Die Funktion `GPxi3052_GetDriverInfo` dient zur Status-Abfrage des Hardware-Treibers.

Format:

```
int GPxi3052_GetDriverInfo(GPxi3052_StructDriverInfo *pDriverInfo);
```

Parameter:

Zeiger, z.B. `pDriverInfo`, auf eine Datenstruktur
Zur Struktur siehe das File `GPxi3052.h` auf der mitgelieferten CD

Beschreibung:

Die Funktion `GPxi3052_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 `GPxi3052_DLL_Version` dient zur Abfrage der Versionsnummer der DLL.

Format:

```
int GPxi3052_DLL_Version(unsigned long *pVersion);
```

Parameter

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

Beschreibung:

Die Funktion `GPxi3052_DLL_Version` gibt die Versionsnummer der *GPxi3052w.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 `GPxi3052_XilinxDownload` dient zum Laden eines FPGA-Files in den XILINX.

Format:

```
int GPxi3052_XilinxDownload(unsigned long card, char *pFileName);
```

Parameter:

`card`

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

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

Beschreibung:

Die Funktion `GPxi3052_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 die Controller ein Power-On-Reset durchlaufen.

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

3.2.1.4 XILINX Write Data

Die Funktion `GPxi3052_XilinxWriteData` ermöglicht das Konfigurieren und Ausführen von Funktionen, die der XILINX bereitstellt.

Format:

```
int GPxi3052_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 `GPxi3052_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 3052-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 `GPxi3052_DpramWriteInstruction` dient zum Senden eines Befehls zum ausgewählten Controller.

Format:

```
int GPxi3052_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 `GPxi3052_DpramWriteInstruction` sendet einen Befehl zum ausgewählten Controller.

Im Header der Struktur, auf die `data` zeigt, befinden sich die Informationen zum durch die Funktion anzusprechenden PXI/ PCI 3052-Board und zum Controller. Deshalb sind diese Parameter nicht gesondert anzugeben.

3.2.1.6 DPRAM Read Response

Die Funktion `GPxi3052_DpramReadResponse` dient zum Lesen einer Antwort vom ausgewählten Controller.

Format:

```
int GPxi3052_DpramReadResponse(unsigned long card, unsigned long port,  
                               unsigned char *data, unsigned long *length);
```

Parameter:

`card`

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

`port`

Nummer des Controllers (1..4)

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 `GPxi3052_DpramReadResponse` liest die älteste vom Controller (1..4) in den Response-Bereich des DPRAM geschriebene Antwort zurück.

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

Aufrufe von `GPxi3052_DpramReadResponse` liefern dann solange Werte, bis diese Liste keine Einträge mehr enthält.

3.2.1.7 DPRAM Read Monitor

Die Funktion `GPxi3052_DpramReadMonitor` dient zum Lesen der Monitor-
daten des ausgewählten Controllers.

Format:

```
int GPxi3052_DpramReadMonitor(unsigned long card, unsigned long port,  
                             unsigned char *data, unsigned long *length);
```

Parameter:

`card`

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

`port`

Nummer des Controllers (1..4)

Zeiger, z.B. `data`, auf den Bereich für Lesedaten (max. 20kByte)

`length`

Parameterwert vor Funktionsaufruf:

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

Parameterwert nach Funktionsaufruf:

Anzahl der tatsächlich gelesenen Monitoreinträge

Beschreibung:

Die Funktion `GPxi3052_DpramReadMonitor` liest die im Monitorbereich
des DPRAM befindlichen Daten.

Dabei handelt es sich ausschließlich um die Daten, die im
Monitormode **Pufferempfang** vom ausgewählten Controller
bereitgestellt werden. Das heißt, dass der normale Response-Bereich
vom Daten-Bereich des Monitors (**Pufferempfang**) getrennt ist.

Pro Monitoreintrag werden 20 Byte benötigt. Die zurückgegebene
Länge `length` ist bereits durch diese 20 Byte geteilt, entspricht somit
der Anzahl der rückgelesenen Monitoreinträge.

3.2.1.8 Reset Port Die Funktion `GPxi3052_ResetPort` dient zur Auslösung eines Software-Resets für den ausgewählten Controller.

Format

```
int GPxi3052_ResetPort(unsigned long card, unsigned long port);
```

Parameter:

`card`

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

`port`

Nummer des Controllers (1..4)

Beschreibung:

Die Funktion `GPxi3052_ResetPort` löst ein Software-Reset für den ausgewählten 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](#)
- ◆ [Read Monitor](#)
- ◆ [Reset Port](#)

3.2.2.1 Init Die Funktion `PXI3052_Init` dient zur Eröffnung von VISA Sessions für alle im System befindlichen `PXI/ PCI 3052`-Boards und deren Initialisierung.

Format:

```
ViStatus PXI3052_Init(ViUInt32 *CardCount);
```

Parameter:

`CardCount`

Anzahl der vom VISA Treiber erkannten `PXI/ PCI 3052`-Boards im System.

Beschreibung:

Die Funktion `PXI3051_Init` sucht alle im System befindlichen `PXI/ PCI 3052`-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 `PXI3052_Done` schließt alle VISA Sessions für im System befindliche `PXI/ PCI 3052`-Boards.

Format:

```
ViStatus PXI3052_Done(void);
```

Parameter:

keine

Beschreibung:

Die Funktion `PXI3052_Done` schließt alle VISA Sessions für im System befindliche `PXI/ PCI 3052`-Boards.

Damit ist kein weiterer Boardzugriff möglich.

3.2.2.3 Driver Info Die Funktion `PXI3052_DriverInfo` liefert allgemeine Treiber- und Boardinformationen.

Format:

```
ViStatus PXI3052_DriverInfo(PXI3052_StructDriverInfo *DriverData,  
                           ViChar *DeviceName);
```

Parameter:

Zeiger, z.B. `DriverData`, auf eine Datenstruktur
Zur Struktur siehe File `PXI3052_API.h` der mitgelieferten CD)

DeviceName

Array[K_DEV_MAX][K_RES_NAME_LENGTH]
(siehe `PXI3052_API.h`)

Beschreibung:

Die Funktion `PXI3052_DriverInfo` stellt verschiedene Informationen zum Treiber und zu den im System befindlichen PXI/ PCI 3052-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 `PXI3052_XilinxDownload` dient zum Laden eines FPGA Files in den XILINX.

Format:

```
ViStatus PXI3052_XilinxDownload(ViUInt32 Card, ViChar *FileName);
```

Parameter:

Card

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

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

Beschreibung:

Die Funktion `GPxi3052_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 alle Controller ein Power-On-Reset durchlaufen.

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

3.2.2.5 XILINX Write Data

Die Funktion `PXI3052_XilinxWriteData` ermöglicht das Konfigurieren und Ausführen von Funktionen, die der XILINX bereitstellt.

Format:

```
ViStatus PXI3052_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 `PXI3052_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 3052-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 `PXI3052_WriteData` dient zum Schreiben von Daten zum ausgewählten Controller.

Format:

```
ViStatus PXI3052_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 Byte pro Befehl)

`Length_In_Bytes`

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

Beschreibung:

Die Funktion `PXI3052_WriteData` ermöglicht das Schreiben von Daten zum ausgewählten Controller.

Im Header der Struktur, auf die `WriteData` zeigt, befinden sich die Informationen zum durch die Funktion anzusprechenden PXI/ PCI 3052-Board und zum Controller. Deshalb sind sie nicht gesondert als Parameter anzugeben.

3.2.2.7 Read Data Die Funktion `PXI3052_ReadData` dient zum Lesen von Daten vom ausgewählten Controller.

Format:

```
ViStatus PXI3052_ReadData(ViUInt32 Card, ViUInt32 Port,  
                          ViUInt8 *ReadData, ViUInt32 *Length);
```

Parameter:

Card

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

Port

Nummer des Controllers (1..4)

Zeiger, z.B. `ReadData`, 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 `ReadData` zeigt, in Bytes

Parameterwert nach Funktionsaufruf:

Anzahl der tatsächlich gelesenen Bytes

Beschreibung:

Die Funktion `PXI3052_ReadData` ermöglicht das Lesen von Daten, die vom ausgewählten Controller bereitgestellt worden sind (siehe auch Funktion `GPxi3052_DpramReadResponse` im Abschnitt [Windows Device Treiber](#)).

3.2.2.8 *Read Monitor*

Die Funktion `PXI3052_ReadMonitor` dient zum Lesen von Monitordaten des ausgewählten Controllers.

Format:

```
ViStatus PXI3052_ReadMonitor(ViUInt32 Card, ViUInt32 Port,  
                             ViUInt8 *MonitorData, ViUInt32 *Length);
```

Parameter:

Card

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

Port

Nummer des Controllers (1..4)

Zeiger, z.B. `MonitorData`, auf den Bereich für Lesedaten (max. 20kByte)

Length

Parameterwert vor Funktionsaufruf:

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

Parameterwert nach Funktionsaufruf:

Anzahl der tatsächlich gelesenen Monitoreinträge

Beschreibung:

Die Funktion `PXI3052_ReadMonitor` ermöglicht das Lesen von Daten aus dem Monitorbereich des ausgewählten Controllers (siehe auch Funktion `GPxi3052_DpramReadResponse` im Abschnitt [Windows Device Treiber](#)).

Dabei handelt es sich um einen separaten Lesebereich des Boards.

3.2.2.9 Reset Port Die Funktion `Pxi3052_ResetPort` dient zur Auslösung eines Software-Resets für den ausgewählten Controller.

Format

```
ViStatus Pxi3052_ResetPort(ViUInt32 Card ViUInt32 Port);
```

Parameter:

Card

Index des PXI/ PCI 3052-Boards, links beginnend mit 1

Port

Nummer des Controllers (1..4)

Beschreibung:

Die Funktion `Pxi3052_ResetPort` löst ein Software-Reset für den ausgewählten 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 3052-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 3052-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 3052-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-API3-1

L

LED Anzeige..... 2-11

PPXI/ PCI 3052
Hardware Installation1-1

R

Ressourcen2-1

SSteckverbinder
Front.....2-10

V

VISA Treiber3-11

W

Windows Treiber1-2, 3-2