

Produktbeschreibung

PXI 6161

MOST150 Interface

Nutzerhandbuch Version 1.2



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: 12.11.2012

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

Stand: November 2012

1	INSTALLATION DER BOARDS	1-1
1.1	HARDWARE INSTALLATION	1-1
1.2	TREIBERINSTALLATION	1-2
1.2.1	<i>Installation Windows Device Treiber.....</i>	<i>1-2</i>
1.2.2	<i>Installation VISA Device Treiber.....</i>	<i>1-3</i>
1.2.3	<i>Ethernet.....</i>	<i>1-5</i>
2	HARDWARE	2-1
2.1	BESTIMMUNG	2-1
2.2	TECHNISCHE DATEN	2-3
2.2.1	<i>Allgemeines.....</i>	<i>2-3</i>
2.2.2	<i>Abmessungen.....</i>	<i>2-3</i>
2.2.3	<i>PXI 6161 Kennwerte</i>	<i>2-4</i>
2.3	AUFBAU UND FUNKTION	2-5
2.3.1	<i>Allgemeines.....</i>	<i>2-5</i>
2.3.2	<i>Isolation</i>	<i>2-6</i>
2.3.3	<i>MOST Schnittstelle.....</i>	<i>2-6</i>
2.3.4	<i>Adressierung</i>	<i>2-7</i>
2.3.5	<i>LED Anzeige.....</i>	<i>2-7</i>
2.3.6	<i>Anschlussbelegung</i>	<i>2-8</i>
2.3.7	<i>OnBoardSchnittstellen</i>	<i>2-9</i>
2.3.8	<i>AV-Erweiterung</i>	<i>2-11</i>
2.4	PRODUKTINFORMATIONEN.....	2-12
3	ANSTEUERSOFTWARE	3-1
3.1	PROGRAMMIEREN ÜBER G-API	3-2
3.2	USERCODE PROGRAMMIERUNG	3-3
3.3	PROGRAMMIEREN ÜBER DLL-FUNKTIONEN.....	3-5
3.3.1	<i>Windows Device Treiber.....</i>	<i>3-6</i>
3.3.1.1	<i>System Info</i>	<i>3-7</i>
3.3.1.2	<i>Transceiver Info.....</i>	<i>3-8</i>
3.3.1.3	<i>Write Instruction</i>	<i>3-9</i>
3.3.1.4	<i>Read Response</i>	<i>3-10</i>
3.3.1.5	<i>Read Response Block.....</i>	<i>3-11</i>
3.3.2	<i>VISA Device Treiber.....</i>	<i>3-12</i>
3.3.2.1	<i>Init.....</i>	<i>3-13</i>
3.3.2.2	<i>Done.....</i>	<i>3-13</i>
3.3.2.3	<i>System Info</i>	<i>3-14</i>
3.3.2.4	<i>Transceiver Info.....</i>	<i>3-15</i>
3.3.2.5	<i>Write Instruction</i>	<i>3-16</i>
3.3.2.6	<i>Read Response</i>	<i>3-17</i>
3.4	PROGRAMMIEREN MIT LABVIEW.....	3-18
3.4.1	<i>LabVIEW über G-API.....</i>	<i>3-18</i>
3.4.2	<i>LLB unter Verwendung des Windows Device Treibers.....</i>	<i>3-18</i>
3.4.3	<i>LLB unter Verwendung des VISA Device Treibers.....</i>	<i>3-18</i>
3.5	WEITERE GÖPEL SOFTWARE	3-18

1 Installation der Boards

1.1 Hardware Installation



Warnung

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



Vergleichen Sie bitte auch das Handbuch für Ihr PXI-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 6161-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 ggf. 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. Zuletzt wird es mit dem an der Frontplatte befindlichen Hebel das letzte Stück eingeschoben.

Nach dem Kontaktieren wird das Boards 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 Installation Windows Device Treiber

PXI 6161-Boards können unter Windows® 2000/ XP sowie unter Windows® 7/ 32 Bit und 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 *Series61xx* enthaltenen *inf*-Datei kann der Hardwareassistent die Installation des Devicetreibers durchführen.

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

- *Pxi61xx.inf* für Windows® 2000/ XP
im Ordner *Driver PXI_PCI - W2K, WinXP (Version xx)*
- *Pxi61xx.inf* für Windows® 7/ 32 Bit
im Ordner *Driver PXI_PCI - Win7_x32 (Version xx)*
- *Pxi61xx_x64.inf* für Windows® 7/ 64 Bit
im Ordner *Driver PXI_PCI - 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 [Programmieren über G-API](#)).

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 Installation VISA Device Treiber

1. Schritt

Kopieren Sie die mitgelieferten Ordner *VISA_Driver PXI_PCI - W2K, WinXP (Version xx)* bzw. *VISA_Driver PXI_PCI - Win7_x32_x64 (Version xx)* aus dem Ordner *Series61xx* der mitgelieferten CD auf die Festplatte. (Empfehlung: vollständige Ordner auf *C: *)

2. Schritt

VISA für Windows® 2000/ XP

Durch die Plug-and-Play Fähigkeit wird über den Hardwareassistenten für jede neu erkannte Hardwarekomponente eine automatische Treiberinstallation gestartet. Folgen Sie den Anweisungen und geben Sie bei der Suche nach dem Treiber das Zielverzeichnis an, in dem sich die Datei *PXI61xx.inf* befindet (nach Empfehlung: *C:\VISA_Driver PXI_PCI - W2K, WinXP (Version xx)*).

VISA für Windows® 7

Durch die Plug-and-Play Fähigkeit wird über den Hardwareassistenten für jede neu erkannte Hardwarekomponente eine automatische Treiberinstallation gestartet. Folgen Sie den Anweisungen und geben Sie bei der Suche nach dem Treiber das Zielverzeichnis an, in dem sich die Datei *PXI61xx_VISA.inf* befindet (nach Empfehlung: *C:\VISA_Driver PXI_PCI - Win7_x32_x64 (Version xx)*).

VISA für LabView RT

Für den Einsatz von PXI 6161-Boards unter dem RT Betriebssystem muss die Datei *PXI61xx.inf* aus dem oben angegebenen Verzeichnis 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 Treiberinstallation können Sie überprüfen, ob die Boards einwandfrei vom System eingebunden worden sind:

Abbildung 1-1:
Windows

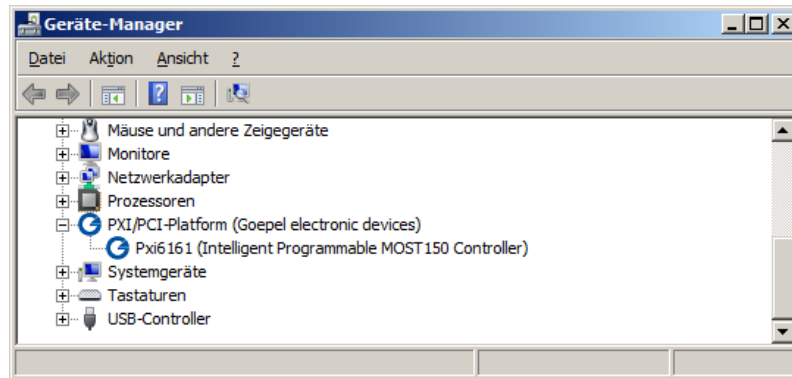


Abbildung 1-2:
VISA für Windows XP

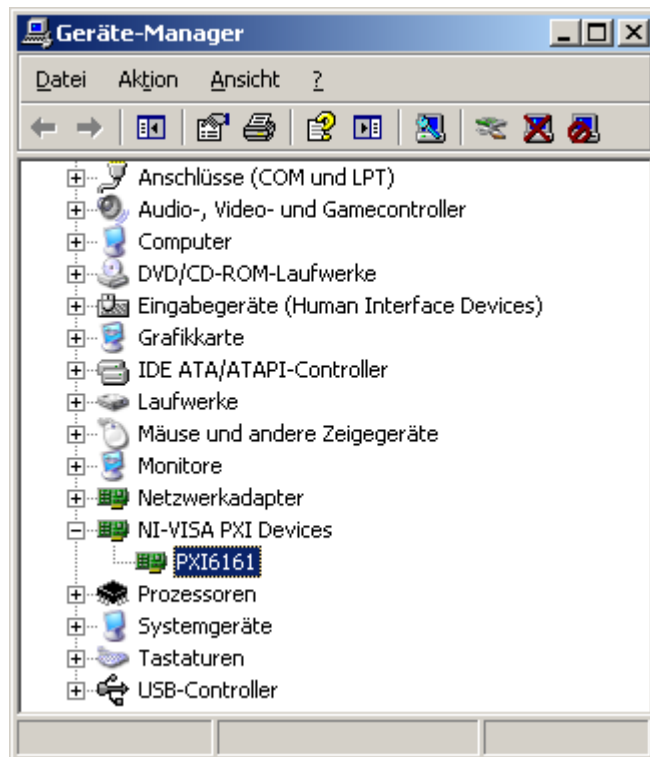
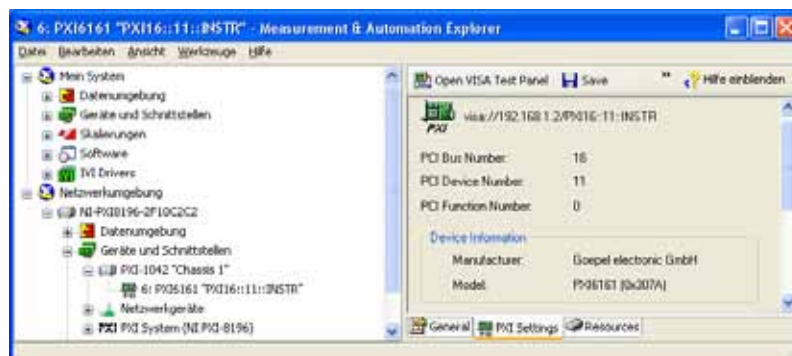


Abbildung 1-3:
VISA für LabVIEW RT



1.2.3 Ethernet

Bei Verwendung der Ethernet-Schnittstelle zur Kommunikation mit dem Steuerrechner ist keine Treiberinstallation erforderlich.

Das Gerät kann direkt über die IP-Adresse angesprochen werden (siehe auch [Adressierung](#)). Diese IP-Adresse kann mittels des HardwareExplorers geändert werden, wobei die eingegebene IP-Adresse nach erfolgtem Neustart wirksam wird:

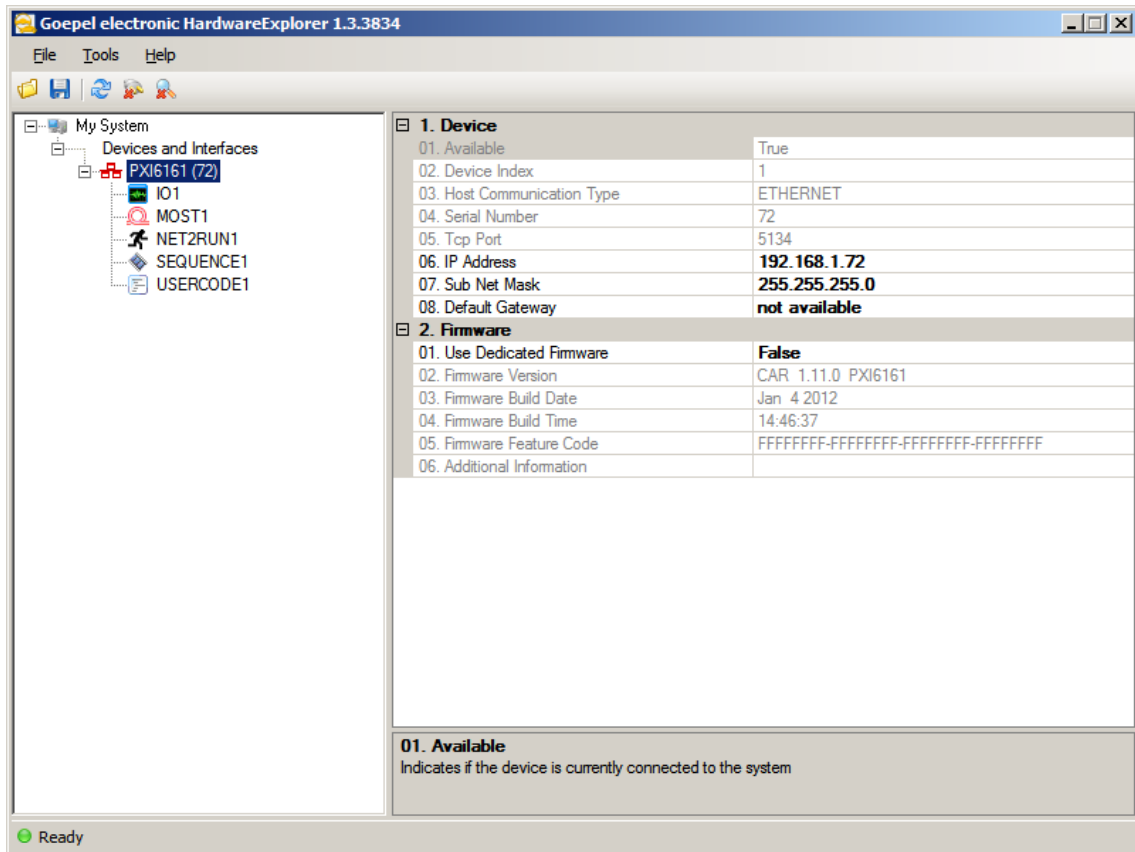


Abbildung 1-4: IP Adresse im GÖPEL electronic HardwareExplorer

2 Hardware

2.1 Bestimmung

PXI 6161 der GÖPEL electronic GmbH ist ein programmierbarer, intelligenter MOST150 Controller für die Automobil- und Unterhaltungselektronik, der optional durch zusätzliche Kommunikationsschnittstellen aufrüstbar ist.

Folgende Eigenschaften kennzeichnen PXI 6161 Controllerboards:

- 1 optische MOST150 Schnittstelle mit Datenraten bis 150MBit/s
- unterstützt 3 Betriebsmodi: Master, Slave, Bypass
- Spy Funktion: Möglichkeit des Monitors der MOST Daten als passiver Busteilnehmer (Karte im Bypass Mode) oder als aktiver Busteilnehmer (Karte im Master Mode bzw. Slave Mode)
- unabhängiger Timer onboard mit Zeitstempelauflösung $\geq 8\text{ns}$
- Möglichkeit der Umschaltung der Masterframeraten zwischen 44,1kHz und 48kHz
- Optional 1 DVI-Ausgang, je 1 S/PDIF Ein- bzw. Ausgang
- Ringbruchdiagnose-Funktion über Frontsteckverbinder
- 4 digitale Eingänge und 4 digitale Ausgänge am Frontsteckverbinder (z.B. für erweiterte Triggerfunktionen)
- optional 2 CAN/ LIN/ K-Line Schnittstellen
- 600MHz Power-PC mit 512MB RAM, 256MB Flash
- Kommunikations- und E/A-Schnittstellen galvanisch vom Anwenderinterface getrennt (betrifft **nicht** die Ressourcen des Extensionboards!)
- Hohe Flexibilität durch steckbare Transceivermodule und mögliche Varianten des Extensionboards
- Steuerung des Gerätes über USB 2.0 oder Ethernet (siehe [Adressierung](#))
- 1 GBit Ethernetschnittstelle auf der Frontplatte auch als Volumendaten- und Debug-Schnittstelle nutzbar
- Betriebszustandsanzeige mittels 4 LEDs auf der Frontplatte (siehe [LED Anzeige](#))

Die folgende Abbildung zeigt ein Board PXI 6161 in der Basisvariante (MOST Schnittstelle mit gelber Schutzabdeckung):



*Abbildung 2-1:
PXI 6161*

2.2 Technische Daten

2.2.1 Allgemeines

Das Controllerboard PXI 6161 ist als Einsteckkarte für das PXI™-Bus-System entwickelt worden.

PXI™ (PCI eXtensions for Instrumentation) ist eine modulare Geräte-Plattform, die 1997 durch National Instruments eingeführt wurde und derzeit durch die PXI Systems Alliance (PXISA) weitergeführt wird.

PXI™, basierend auf dem *CompactPCI™*-Bus, bietet alle Vorteile der PCI-Architektur wie Leistungsfähigkeit, Industrie-Einsatz und COTS Technologie.

PXI bietet einen stabilen mechanischen Formfaktor (*CompactPCI*) und wird durch ein Industrie-Konsortium, das Hardware-, Elektronik-, Software-, Stromversorgungs- und Kühlanforderungen festlegt, standardisiert.

Außerdem verfügt PXI™ über integrierte Timing- und Synchronisationsmechanismen, die es ermöglichen, Takt- und Triggersignale zwischen den Busteilnehmern zu verteilen.

PXI™ ist eine zukunftssträchtige Technologie und wurde entwickelt, um einfach und schnell auf Änderungen bei Test-, Mess- und Automatisierungsanforderungen reagieren zu können.

Ein PXI 6161-Controllerboard fungiert als PXI-Slave, deshalb kann es in beliebigen PXI-Slots betrieben werden (ausgenommen Slot1).

Auch der PCI Plug & Play Autoerkennung-Mechanismus wird von PXI 6161-Controllerboards unterstützt.

Für die PXI-Integration ist keine Jumper-Konfiguration erforderlich.

CompactPCI- und PXI-Produkte sind untereinander austauschbar.

D.h., sie können sowohl in einem CompactPCI- als auch in einem PXI-Rack betrieben werden. Allerdings können durch den Betrieb im jeweils anderen Rack bestimmte Takt- und Triggermöglichkeiten verlorengehen.

So können Sie z.B. einen CompactPCI Netzwerkinterface-Controller in ein PXI-Rack einbauen, um für einen Prüfstand zusätzliche Netzwerkinterface-Funktionen zur Verfügung zu stellen.

Andererseits würde ein in einem CompactPCI-Rack eingebautes PXI-Modul nicht seine zusätzlichen Takt- und Triggermöglichkeiten nutzen können.

Das Controllerboard PXI 6161 ist bereits für die Nutzung in einem PXI Express Hybridslot vorbereitet, der sowohl PCI als auch PCI Express unterstützt.

2.2.2 Abmessungen

Ein Controllerboard PXI 6161 ist ein 3U Standard-PXI-Modul und belegt eine Slotbreite.

Boardabmessungen ohne Frontplatte und Griff:

- PXI 6161: 160 mm x 100 mm (L x B)

2.2.3 PXI 6161 Ein PXI 6161-Board hat folgende Kennwerte: Kennwerte

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bemerkung
Optische MOST150 Schnittstelle						
N	Anzahl		1			
C	Übertragungsrate			150	MBit/s	
Ringbruchdiagnose-Schnittstelle						Bezugspotenzial GNDiso
N	Anzahl		1			
U _O	Ausgangsspannung	1		UBAT	V	
UBAT _{internRBD}	interne Batteriespannung		12		V	abschaltbar
UBAT _{externRBD}	externe Batteriespannung		12	27	V	
Digitale Eingänge						Bezugspotenzial GNDiso
N	Anzahl		4			
U _{IH}	High-level Eingangsspannung	3,5		5,5	V	
U _{IL}	Low-level Eingangsspannung			1,5	V	
I _L	Eingangskriechstrom			35	μA	
Digitale Ausgänge						Bezugspotenzial GNDiso
N	Anzahl		4			
U _{OH}	High-level Ausgangsspannung	4,8		5	V	
U _{OL}	Low-level Ausgangsspannung			0,5	V	
I _{OUT}	Ausgangsstrom			8	mA	
Zusätzliche Schnittstellen						
	CAN/ LIN/ KLine Schnittstellen			2		Siehe OnBoard Schnittstellen
	Erweiterung AV Ressourcen			1		Siehe AV-Erweiterung

2.3 Aufbau und Funktion

2.3.1 Allgemeines

Ein leistungsfähiger 600MHz AMCC 460EX PowerPC bildet den Kern eines PXI 6161 MOST150 Boards. Als 32-Bit RISC CPU basiert der Power-PC auf der Book-E Enhanced PowerPC-Architektur, die dank Superscalar-Technologie das gleichzeitige Laden zweier Integer-Befehle ermöglicht und die Abarbeitungsreihenfolge der einzelnen Befehle in der Befehls-Pipeline optimiert. Mit seiner hoch optimierten, doppelt präzisen Gleitkomma-Einheit bietet dieser Prozessor die Rechenleistung, die für die komplexen Anwendungen erforderlich ist. Außerdem befinden sich ein schneller, mit 400MHz getakteter DDR2 RAM von 512MB und ein 256MB Flash-Speicher, von dem über 80% für Anwenderprogramme zur Verfügung stehen, auf dem Board.

Das PXI 6161 Board ist als hoch flexible MOST150 Controller-Plattform entwickelt worden. Es bietet eine optische MOST150 Schnittstelle zur Übertragung von asynchronen, isochronen, Control- und MOST Ethernet Datenpaketen.

Insgesamt werden drei verschiedenen Betriebsmodi unterstützt:

- Master (aktiver Busteilnehmer)
- Slave (aktiver Busteilnehmer)
- Bypass (passiver Busteilnehmer)

Dabei ist das Monitoren empfangener oder der gesendeter MOST150 Daten in jedem Betriebsmode möglich.

Die folgende schematische Darstellung zeigt die Verteilung des MOST Signals:

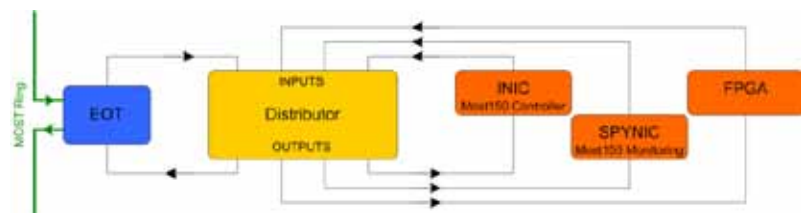


Abbildung 2-2:
Verteilung MOST Signal

Der Einschluss in den MOST Ring erfolgt über den Elektrisch-Optischen Transceiver (EOT).

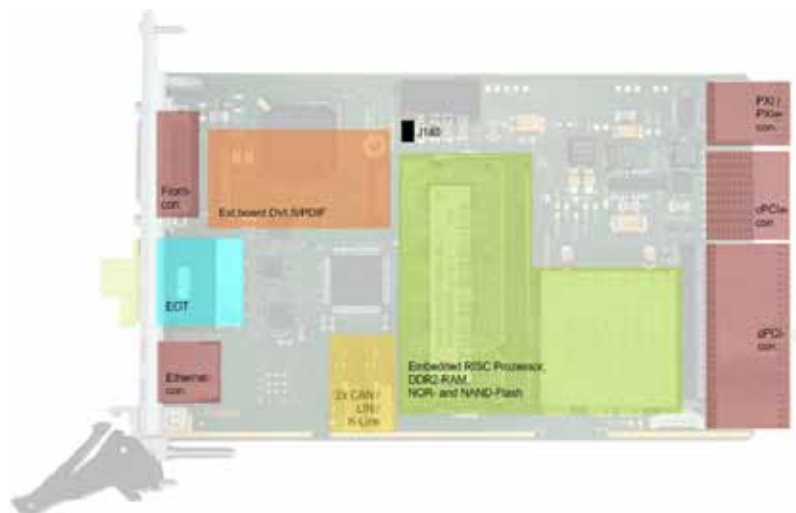


Abbildung 2-3:
PXI 6161 schematisch

Unterhalb des elektrisch-optischen Transceivers (EOT) befindet sich in Abbildung 2-3 eine Buchse mit Zugang zur 1 GBit Ethernet-Schnittstelle. Sie dient zur Gerätesteuerung, als Debug-Interface oder zum Übertragen großer Datenmengen (z.B. Monitoraten).

Darüber hinaus bietet das Board PXI 6161 optional zwei weitere serielle Busknoten, von denen prinzipiell jeder als CAN-, LIN oder KLine-Schnittstelle konfiguriert werden kann. Jeder Knoten hat einen ihm zugeordneten Transceiver-Steckplatz, wobei der jeweils gesteckte Transceiver den Typ der Schnittstelle des zugeordneten Knotens bestimmt (siehe auch [OnBoard Schnittstellen](#)).

Diese sind, wie auch alle übrigen elektrischen Schnittstellen (Ringbruchdiagnose, Audio-, Videoausgänge) und digitalen Ein-/Ausgänge, auf den 50-poligen Frontsteckverbinder geführt.

Über dem Frontsteckverbinder finden Sie vier Status-LEDs, die den Betriebszustand des Controllerboards anzeigen (siehe [LED Anzeige](#)).

2.3.2 Isolation

Überspannungen können kostspielige Testausrüstungen schädigen oder zu unsicheren Prüfergebnissen führen. Die Potenzialtrennung schützt vor Überspannungen und kann gefährliche Stromstöße unterdrücken. Sie verhindert außerdem Erdschleifen, die für Datenfehler aufgrund von Erdungspotenzial-Differenzen verantwortlich sind.

Beim PXI 6161-Controllerboard sind das PXI-System und alle Eingangs- und Ausgangssignale des Frontsteckverbinders elektrisch voneinander getrennt. Das umfasst sowohl die CAN-, LIN- und MOST Kommunikationsschnittstellen als auch die digitalen und analogen Ein- und Ausgänge.

Das Testsystem benötigt u.U. eine Verbindung zwischen dem Potenzial GNDiso am Frontsteckverbinder und dem Groundpotential des Testobjektes (ESG, usw.).

Mit Jumper J140 (siehe Abbildung 2-3) kann eine Verbindung zwischen dem Potenzial GNDiso am Frontsteckverbinder und dem Groundpotential des PXI-Systems hergestellt werden.



Das Herstellen einer Ground-Verbindung zum PXI-System kann dazu führen, dass extrem hohe Ströme über die Prüfanschlüsse und das PXI 6161-Controllerboard fließen.

Das kann Fehlfunktionen, falsche Prüfergebnisse oder Schäden am Controllerboard oder anderem Testequipment zur Folge haben.

Deshalb muss vor dem Schließen von Jumper J140 sichergestellt werden, dass das Prüfobjekt und alle anderen am Frontsteckverbinder angeschlossenen Geräte über ein isoliertes Netzgerät gespeist werden!

2.3.3 MOST Schnittstelle

Für die uneingeschränkte Funktion einer MOST Schnittstelle in einem Netzwerk ist es notwendig, dass alle Teilnehmer mit demselben Systemtakt kommunizieren. Dieser hängt u.a. von der Masterframerate des Systems ab.

PXI 6161-Boards unterstützen folgende Masterframeraten:

- 48 kHz und
- 44,1 kHz

(G-API Befehl G_Most_Node_SetProperties, Parameter ClockMode)

2.3.4 Adressierung

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 6161 Board kann diese geographische Slotadresse auslesen. Wird die Ethernet Schnittstelle genutzt, kann die Baugruppe über die Default IP Adresse 192.168.1.62 adressiert werden, die bei Bedarf auch geändert werden kann (siehe auch [Treiberinstallation/ Ethernet](#)).

Prinzipiell gibt es dafür zwei Wege:

- HardwareExplorer: Auswahl der Baugruppe, unter Device Eingabe der erforderlichen IP Adresse; die neue IP Adresse wird nach Neustart wirksam
- G API Befehl G_Common_Ethernet_IpAddress_Set; die neue IP Adresse wird nach Neustart wirksam

2.3.5 LED Anzeige

Die auf der Frontplatte angeordneten Leuchtdioden geben Auskunft über den momentanen Betriebszustand eines PXI 6161-Boards:

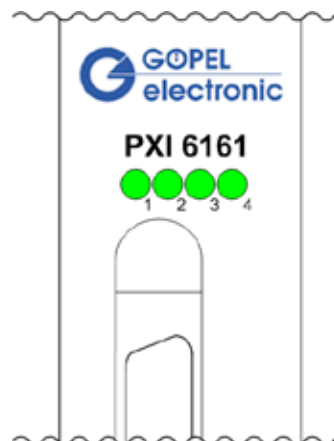


Abbildung 2-4:
LED Anzeige

Wesentliche Anzeigezustände der LEDs werden in der folgenden Tabelle erläutert:

Zustand				Bemerkung
LED 1	LED 2	LED 3	LED 4	
blinken abwechselnd				Bootloadersoftware läuft; wahrscheinliche Fehlerursache: Softwarereset nicht erfolgt
	blinkt			Firmware läuft
leuchtet (kurz)				Anzeige, während Firmwarebefehl abgearbeitet wird
			leuchtet	Ethernetverbindung hergestellt



Diese Anzeige erfolgt mit niedriger Priorität und kann durch andere laufende Programme beeinflusst werden.

2.3.6 Anschluss-
belegung

Verwendeter Steckverbinder:
Benötigtes Anschlusskabel:

Samtec VRDPC-50-01-M-RA
Samtec VPSTP-16-1000-01

Die folgende Tabelle enthält die Pinbelegung des
Frontsteckverbinders:

<i>Pin</i>	<i>Signal</i>			<i>Pin</i>	<i>Signal</i>		
1	GND			26	GND		
2	CAN1_H	LIN1	K-Line1	27	CAN2_H	LIN2	K-Line2
3	CAN1_L		L-Line1	28	CAN2_L		L-Line2
4	GND			29	GND		
5	UBAT _{externCAN1}			30	UBAT _{externCAN2}		
6	GNDiso			31	GNDiso		
7	GND			32	GND		
8	DIGITAL_OUT1			33	DIGITAL_IN1		
9	DIGITAL_OUT2			34	DIGITAL_IN2		
10	GND			35	GND		
11	DIGITAL_OUT3			36	DIGITAL_IN3		
12	DIGITAL_OUT4			37	DIGITAL_IN4		
13	GND			38	GND		
14	RingBruchDiagnose			39	GNDiso		
15	UBAT _{externRBD}			40	IO_EXP7		
16	GND			41	GND		
17	IO_EXP1			42	IO_EXP8		
18	IO_EXP2			43	IO_EXP9		
19	GND			44	GND		
20	IO_EXP3			45	IO_EXP10		
21	IO_EXP4			46	IO_EXP11		
22	GND			47	GND		
23	IO_EXP5			48	IO_EXP12		
24	IO_EXP6			49	IO_EXP13		
25	GND			50	GND		

Die Belegung der **Kommunikationsschnittstellen** richtet sich nach dem jeweils gesteckten Transceiver (siehe [OnBoard Schnittstellen](#)), während die Belegung der **Pins 17..24** und **40..49** mit dem verwendeten AV-Erweiterungsboard variiert, siehe [AV-Erweiterung](#).

2.3.7 OnBoard Schnittstellen

Ein PXI 6161 Controllerboard bietet zwei weitere Kommunikationsschnittstellen. Diese beiden Schnittstellen können entweder frei bleiben oder optional als CAN-, LIN- bzw. K-Line Schnittstellen ausgeführt sein. Bei Bedarf kann die Zuordnung durch Wechsel des entsprechenden Transceivers geändert werden.



Gehen Sie bitte beim Transceiver-Wechsel äußerst vorsichtig vor, und achten Sie beim Stecken der Transceiver auf deren Position und Orientierung.

Position und Orientierung der Transceiver sind in der folgenden Abbildung dargestellt:

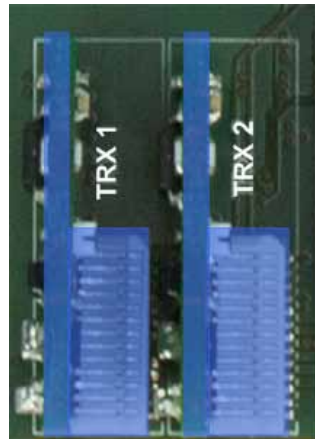


Abbildung 2-5:
Transceiveranordnung

Jeder Modultyp ist codiert und eindeutig identifizierbar. Zur Zeit sind folgende Transceivermodule verfügbar.

- TJA1041A – CAN (high-speed)
- PCA82C251 – CAN (high-speed)
- TJA1054 – CAN (low-speed)
- B10011S – CAN (truck and trailer)
- TJA 1020 – LIN
- L9637D – K-Leitung

Diese beiden Schnittstellen werden i. Allg. mit einer internen 12V-Spannung ($UBAT_{intern}$) versorgt. Bei Verwendung anderer Spannungspegel kann die interne Spannung per Software einzeln abgeschaltet werden.

(G-API Befehle

G_Can_Node_InternalVBat_Disable bzw.

G_Lin_Node_InternalVBat_Disable bzw.

G_KLine_Node_InternalVBat_Disable)

In diesem Fall muss eine externe Spannung ($UBAT_{extern}$) über die vordefinierten Pins am Frontsteckverbinder eingespeist werden.



Beide Kommunikationsschnittstellen inklusive der dazugehörigen $UBAT_{extern}$ Eingänge verwenden als Bezugspotenzial die Pins GNDiso.

Soll später wieder die interne Spannungsversorgung genutzt werden, sind die G-API Befehle G_Can_Node_InternalVBat_Enable bzw. G_Lin_Node_InternalVBat_Enable bzw. G_Kline_Node_InternalVBat_Enable auszuführen.

Die folgende Tabelle zeigt die Zuordnung der Kommunikationsschnittstellen:

ID	Schnittstelle	Knoten
1	CAN 1	TRX1
2	CAN 2	TRX2
3	Reserved	
4	Reserved	
5	LIN 1	TRX1
6	LIN 2	TRX2
7	Reserved	
8	Reserved	
9	K-LINE 1	TRX1
10	K-LINE 2	TRX2

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bemerkung
CAN V2.0B Schnittstellen						Bezugspot. GNDiso
C	Übertragungsrate			1	MBit/s	
UBAT _{intern}	interne Batteriespannung		12		V	abschaltbar
UBAT _{extern}	externe Batteriespannung			27	V	
RCAN	Abschlusswid. high-speed Transceiver		120		Ω	abschaltbar
RCAN	Abschlusswid. low-speed Transceiver		10		kΩ	
LIN V2.1 Schnittstellen						Bezugspot. GNDiso
C	Übertragungsrate			19,2	kBit/s	
UBAT _{internCAN}	interne Batteriespannung		12		V	abschaltbar
UBAT _{externCAN}	externe Batteriespannung		12	27	V	
RLIN	Pullupwiderstand	1	30		kΩ	umschaltbar Master/Slave
K-Line Schnittstellen						Bezugspot. GNDiso
C	Übertragungsrate			9.6	kBit/s	
UBAT _{intern}	interne Batteriespannung		12		V	abschaltbar
UBAT _{extern}	externe Batteriespannung		12	27	V	



Anmerkung zu R_{CAN} für den high-speed Transceiver: Der 120Ω Busabschlusswiderstand kann per Software deaktiviert werden (G-API Befehl G_CAN_Node_BusTermination_Disable, erneute Aktivierung mit G_CAN_Node_BusTermination_Enable).



Anmerkung zu R_{LIN}: Der 1kΩ Pullupwiderstand entspricht dem LIN Master Bus Abschluss und kann per Software aktiviert werden (G-API Befehl G_Lin_PullUpResistor_Enable à Master, Deaktivierung mit G_Lin_PullUpResistor_Disable à Slave). Im deaktivierten Zustand wird der interne Abschlusswiderstand des LIN Transceivers wirksam (typisch 30kΩ beim TJA1020).

2.3.8 AV- Erweiterung

Durch Aufstecken eines AV-Erweiterungsboards stehen zusätzliche Schnittstellen für Audio- und Videosignale zur Verfügung.

Die GÖPEL electronic GmbH bietet derzeit einen Typ an, das AV-Erweiterungsboard Typ1 hat zusätzliche Ressourcen mit den Kennwerten:

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bemerkung
DVI-D Ausgang		DVI 1.0 compliant				Bezugspot. GND
N _{LINK}	Anzahl der Links		1			Single Link
C _{DVI}	Übertragungsrate	25		48	MBit/s	
R	Auflösung	640x420		1600x1200	Pixel	
S/PDIF Audio Ausgang		IEC60958-3 compliant				Bezugspot. GND
C _{SPDIF OUT}	Übertragungsrate	32		48	k Frames/s	
S/PDIF Audio Eingang		IEC60958-3 compliant				Bezugspot. GND
C _{SPDIF IN}	Übertragungsrate	32		48	k Frames/s	

Die folgende Tabelle zeigt die Pinbelegung des Frontsteckverbinders für die AV-Erweiterung Typ1:

Pin	Signal	Signal	Pin
15	UBAT _{externRBD}	DVI.+5Vout	40
16	GND	GND	41
17	S/PDIF in	DVI.SDA	42
18	S/PDIF out	DVI.SCL	43
19	GND	GND	44
20	DVI.TX0_p	DVI.TX2_p	45
21	DVI.TX0_n	DVI.TX2_n	46
22	GND	GND	47
23	DVI.TX1_p	DVI.TXC_p	48
24	DVI.TX1_n	DVI.TXC_n	49
25	GND	GND	50



Alle Schnittstellen (Pins) des AV-Erweiterungsboards verwenden als Bezugspotenzial die Pins GND. Das GND Potenzial ist so mit dem Groundpotenzial des PXI-Chassis verbunden.

2.4 Produktinformationen

Der intelligente, programmierbare MOST150 Controller PXI 6161 bildet eine hoch flexible, anpassungsfähige Plattform, die mit verschiedenen Hard- und Softwareressourcen ergänzt werden kann.

Es folgt die Liste der verfügbaren Optionen:

Variante	Beschreibung
Optionen für PXI 6161 Controllerboards	
CAN Node	Zusätzlicher CAN Knoten zum Upgrade auf 1 oder 2 Kommunikations-Knoten, inkl. Transceiver-Modul Die Anzahl gleichzeitig installierbarer CAN/ LIN/ K-Line Knoten pro Board ist 2
LIN Node	Zusätzlicher LIN Knoten zum Upgrade auf 1 oder 2 Kommunikations-Knoten, inkl. Transceiver-Modul Die Anzahl gleichzeitig installierbarer CAN/ LIN/ K-Line Knoten pro Board ist 2
K-Line Node	Zusätzlicher K-Line Knoten zum Upgrade auf 1 oder 2 Kommunikations-Knoten, inkl. Transceiver-Modul Die Anzahl gleichzeitig installierbarer CAN/ LIN/ K-Line Knoten pro Board ist 2
AV-Modul Typ 1	AV-Erweiterungmodul Typ 1 inkl. je 1 elektrischer S/PDIF Ein- und Ausgang und 1 DVI Ausgang Die Anzahl gleichzeitig installierbarer AV-Erweiterungmodule pro Board ist 1 Diese Option ist unabhängig von und zusätzlich zu den Optionen CAN/ LIN/ K-Line nutzbar
CAN TJA1041A	CAN high speed Transceiver Modul Typ TJA 1041A
CAN PCA82C251	CAN high speed Transceiver Modul Typ PCA 82C251
CAN TJA1054	CAN low speed Transceiver Modul Typ TJA 1054
CAN B10011S	CAN Transceiver Modul Typ B10011S (truck and trailer)
LIN TJA1020	LIN Transceiver Modul Typ TJA 1020
K-Line L9637D	K-Line Transceiver Modul Typ L9637D
DIAG KW2000 TP1.6	Keyword 2000 auf TP1.6 on-board CAN Diagnosesoftware
DIAG KW2000 TP2.0	Keyword 2000 auf TP2.0 on-board CAN Diagnosesoftware
DIAG KW2000 ISO-TP	Keyword 2000 auf CAN-ISO-TP on-board CAN Diagnosesoftware
DIAG UDS ISO-TP	UDS auf CAN-ISO-TP on board CAN Diagnosesoftware
DIAG GMLan	GMLan on-board CAN Diagnosesoftware
DIAG J1939	J1939 on-board CAN Diagnosesoftware
CAL CCP2.1	CAN Calibration Protokoll CCP2.1
LIN adv-lib	Advanced library für Test des LIN-Protokolls specific. 2.0/ 2.1
CAN DBC	Automatisches Einlesen der CAN Datenbasis (dbc-file) (LabVIEW-basierend)
LIN LDF	Automatisches Einlesen der LIN Datenbasis (ldf-file) (LabVIEW-basierend)
Net2Run IDE	Software Programmierumgebung (Windows host) zur Erstellung G-API basierender Onboard UserCode Programme auf PXI 6161 Boards, enthält: Net2Run IDE, QNX Neutrino CLT, G-API onboard-Bibliotheken, Einzel-Entwicklerlizenz
UserCode Runtime	UserCode Runtime-Modul für die Ausführung G-API basierender Onboard UserCode Programme auf PXI 6161 Boards (diese Option ist für jedes PXI 6161 Board erforderlich)

3 Ansteuersoftware

Zur Einbindung von PXI 6161 Boards in Ihre eigenen Applikationen stehen verschiedene Möglichkeiten zur Verfügung:

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

3.1 Programmieren über G-API

Die G_API (GÖPEL-API) ist das C-User-Interface für GÖPEL electronic-Hardware unter Windows® und das bevorzugte User Interface 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 der integrierter Selbsttests zu überprüfen.

Die folgende Abbildung zeigt den GÖPEL electronic Hardware Explorer:

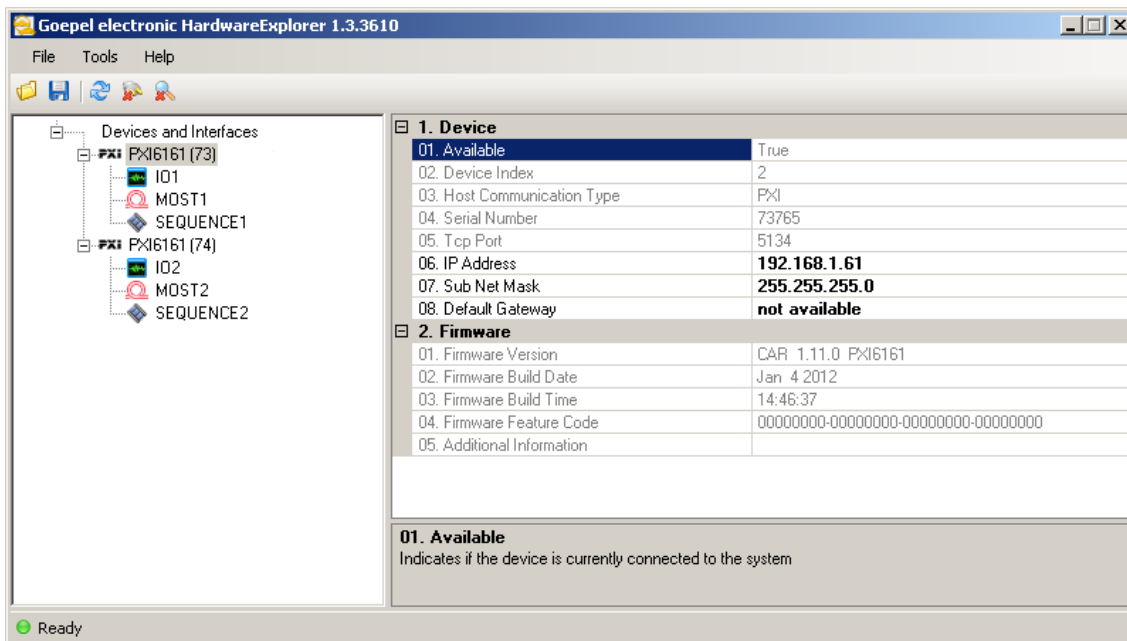


Abbildung 3-1: Hardware Explorer



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

Diese Dokumentation und die Installationssoftware finden Sie im Ordner G-API der mitgelieferten CD „Produktinformationen“.

3.2 UserCode Programmierung

Das PXI 6161 Controllerboard kann Benutzerprogramme direkt auf seinem PowerPC-Prozessor ausführen. Dies erfordert eine Freischaltung des UserCode Run-Time Moduls.

Das UserCode Run-Time Modul wird optional für PXI 6161 Boards angeboten und erfordert eine Lizenz pro Controllerboard.

Die Ausführung von Programmen direkt auf dem PowerPC verbessert das Echtzeit-Verhalten und entlastet den PCI-Bus des Host-Computers. Zu diesem Zweck hat GÖPEL electronic die bestehende C-API für Windows[®] auf das QNX Neutrino Echtzeit-Betriebssystem portiert und um zusätzliche On-Board Funktionalitäten erweitert. Das QNX Neutrino Echtzeit-Betriebssystem basiert auf einer Microkernel-Architektur, die sich durch eine saubere Trennung von Kernel und Applikation auszeichnet. Hierdurch ist es möglich, Benutzerprogramme in einem eigenen virtuellen Speicher auszuführen, was eine sichere Programmausführung garantiert und die Stabilität verbessert.

Für eine reibungslose Portierung von bestehenden Programm-Quellcodes nutzt die Onboard G-API eine Obermenge der bekannten Windows[®] G-API-Befehle. Darüber hinaus bieten zusätzliche Funktionen Zugang zu Eventhandling, Timer Tasks, sowie den FLASH Dateisystemen und weiteren Betriebssystem-Ressourcen sowie zu den Standard C-Bibliotheken.

Bei der Onboard G-API Programmierung ist zu beachten, dass der PowerPC-Prozessor Big-Endian Byte-Reihenfolge verwendet. Zur einfacheren Portierung sind Macros für die Endian-Konvertierung im Lieferumfang des Net2Run IDE Entwicklungs-Systems enthalten.

Mit dem Net2Run IDE Entwicklungs-System bietet GÖPEL electronic eine komplette Kette von Entwicklungswerkzeugen für die Erstellung von UserCode-Programmen und zu deren direkter Ausführung auf dem PXI 6161 Controller-Board.

Das Net2Run IDE Entwicklungs-System basiert auf Eclipse IDE und enthält die QNX Neutrino Command Line Tools (CLT), inklusive PowerPC-Compiler, Linker und Debugger.

UserCode-Programme lassen sich über eine Ethernet-Verbindung direkt aus Net2Run IDE downloaden und debuggen.

Die folgende Abbildung zeigt die Entwicklungsumgebung Net2Run IDE:

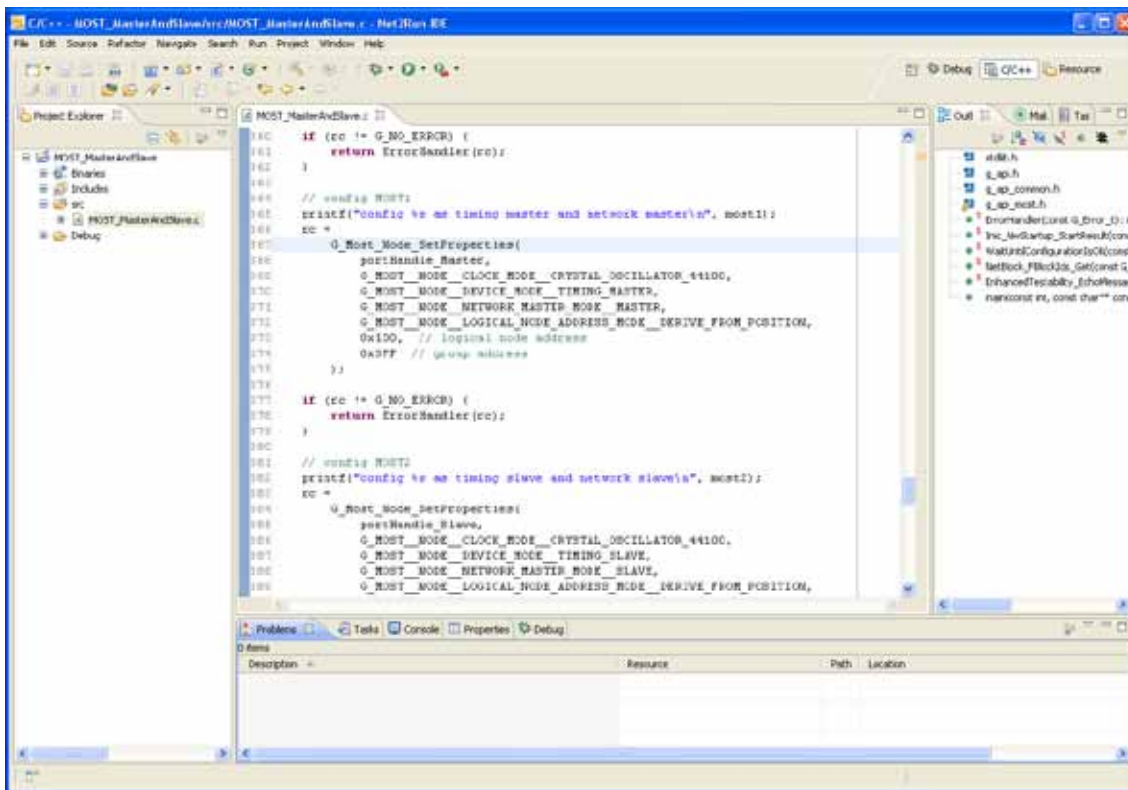


Abbildung 3-2: Net2Run IDE Entwicklungsumgebung



Weitere Informationen finden Sie in der G-API Dokumentation. Diese finden Sie zusammen mit der Installationssoftware im Ordner *G-API* der mitgelieferten CD „Produktinformationen“.

3.3 Programmieren über DLL-Funktionen



Die Programmierung über DLL Funktionen ist auch zukünftig z.B. für existierende Projekte möglich, bei denen nicht mit dem GOPEL electronic Programmierinterface 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.

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

3.3.1 Windows Device Treiber

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

- [System Info](#)
- [Transceiver Info](#)
- [Write Instruction](#)
- [Read Response](#)
- [Read Response Block](#)

3.3.1.1 System Info Die Funktion `Pxi61xx_SystemInfo` dient zur Status-Abfrage des Hardware-Treibers und der Karteneigenschaften.

Format:

```
S32 Pxi61xx_SystemInfo(t_System_Info *pSystemInfo,  
                    U32 LengthInByte);
```

Parameter:

Zeiger, z.B. `pSystemInfo`, auf eine Datenstruktur
(Zur Struktur siehe das File `Pxi6100_UserInterface.h`
auf der mitgelieferten CD)

`LengthInByte`

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

Beschreibung:

Die Funktion `Pxi61xx_SystemInfo` gibt Informationen über den Status des Hardware-Treibers zurück.

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

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

3.3.1.2 *Transceiver Info*

Die Funktion `Pxi61xx_TransceiverInfo` liefert Informationen über die gesteckten Transceiver sowie deren Anzahl zurück.

Format:

```
S32 Pxi61xx_TransceiverInfo(t_Transceiver_Properties *pTransceiverProperties,  
                           U32 LengthInByte);
```

Parameter:

Zeiger, z.B. `pTransceiverProperties`, auf eine Datenstruktur
(Zur Struktur siehe das File `Pxi6100_UserInterface.h`
auf der mitgelieferten CD)

LengthInByte

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

Beschreibung:

Die Funktion `Pxi61xx_TransceiverInfo` gibt Informationen über die Transceivereigenschaften zurück.

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

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

3.3.1.3 Write Instruction

Die Funktion `Pxi61xx_WriteInstruction` dient zum Senden eines Befehls zum PXI 6161-Controller.

Format:

```
S32 Pxi61xx_WriteInstruction(U8 *pData,  
                           U16 DataLength);
```

Parameter:

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

`DataLength`

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

Beschreibung:

Die Funktion `Pxi61xx_WriteInstruction` sendet einen Befehl zum PXI 6161-Controller.

Im Header der Struktur, auf die `pData` zeigt, befindet sich die Information zum anzusprechenden PXI 6161-Board. Deshalb ist dieser Parameter nicht gesondert anzugeben.

3.3.1.4 *Read Response*

Die Funktion `Pxi61xx_ReadResponse` dient zum Lesen einer Antwort vom PXI 6161-Controller.

Format:

```
S32 Pxi61xx_ReadResponse(U8 Device,  
                        U8 *pData,  
                        U32 *DataLength);
```

Parameter:

Device

Index des PXI 6161-Boards, links beginnend mit 1

(Aber: Zählung einschließlich aller anderen PXI/ PCI 61xx-Boards im selben System)

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

DataLength

Parameterwert vor Funktionsaufruf:

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

Parameterwert nach Funktionsaufruf:

Tatsächlich gelesene Byteanzahl

Beschreibung:

Die Funktion `Pxi61xx_ReadResponse` liest die älteste vom PXI 6161-Controller in den Response-Bereich 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 `Pxi61xx_ReadResponse` liefern dann solange Werte, bis diese Liste keine Einträge mehr enthält.

3.3.1.5 *Read Response Block*

Die Funktion `Pxi61xx_ReadResponseBlock` dient zum Lesen aller verfügbaren Antworten vom PXI 6161-Controller.

Format:

```
S32 Pxi61xx_ReadResponseBlock(U8 Device,
                              U8 *pData,
                              U32 *DataLength,
                              U32 *BlockCounter);
```

Parameter:

Device

Index des PXI 6161-Boards, links beginnend mit 1
(Aber: Zählung einschließlich aller anderen PXI/ PCI 61xx-Boards im selben System)

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

DataLength

Parameterwert vor Funktionsaufruf:
Größe des Puffers, auf den `pData` zeigt, in Bytes
Parameterwert nach Funktionsaufruf:
Tatsächlich gelesene Byteanzahl

Zeiger, z.B. `BlockCounter`

Anzahl der enthaltenen Einzelantworten

Beschreibung:

Die Funktion `Pxi61xx_ReadResponseBlock` liest alle vom PXI 6161-Controller in den Response-Bereich geschriebenen Antworten zurück.

3.3.2 VISA Device Treiber

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

- [Init](#)
- [Done](#)
- [System Info](#)
- [Transceiver Info](#)
- [Write Instruction](#)
- [Read Response](#)

3.3.2.1 Init Die Funktion `PXI61xx_Init` dient zur Eröffnung von VISA Sessions für alle im System befindlichen PXI/ PCI 61xx-Boards und deren Initialisierung.

Format:

```
ViStatus PXI61xx_Init(ViUInt32 *CardCount);
```

Parameter:

CardCount

Anzahl der vom VISA Treiber erkannten PXI/ PCI 61xx-Boards im System.

Beschreibung:

Die Funktion `PXI61xx_Init` sucht alle im System befindlichen PXI/ PCI 61xx-Boards und eröffnet die erforderlichen Sessions. Außerdem werden Board-interne Initialisierungen durchgeführt. Deshalb muss diese Funktion als erster Schritt ausgeführt werden.

3.3.2.2 Done Die Funktion `PXI61xx_Done` schließt alle VISA Sessions für im System befindliche PXI/ PCI 61xx-Boards.

Format:

```
ViStatus PXI61xx_Done(void);
```

Parameter:

keine

Beschreibung:

Die Funktion `PXI61xx_Done` schließt alle VISA Sessions für im System befindliche PXI/ PCI 61xx-Boards.

Damit ist kein weiterer Boardzugriff möglich.

3.3.2.3 System Info Die Funktion `PXI61xx_SystemInfo` liefert allgemeine Treiber- und Boardinformationen.

Format:

```
ViStatus PXI61xx_SystemInfo(t_System_Info *DriverData,  
                           ViUInt32 LengthInByte,  
                           ViChar *DeviceName);
```

Parameter:

Zeiger, z.B. `DriverData`, auf eine Datenstruktur
(Zur Struktur siehe das File `PXI61xx_API.h` auf der mitgelieferten CD)

`LengthInByte`
Größe des Puffers, auf den `DriverData` zeigt, in Bytes

`DeviceName`
`Array[K_DEV_MAX][K_RES_NAME_LENGTH]`
(siehe `PXI61xx_API.h`)

Beschreibung:

Die Funktion `PXI61xx_SystemInfo` stellt verschiedene Informationen zum Treiber und zu den im System befindlichen PXI/ PCI 61xx-Boards zur Verfügung.

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

3.3.2.4 Transceiver Info Die Funktion `PXI61xx_TransceiverInfo` liefert Informationen über die gesteckten Transceiver sowie deren Anzahl zurück.

Format:

```
ViStatus PXI61xx_TransceiverInfo(t_Transceiver_Properties *TransceiverProperties,  
                                ViUInt32 LengthInByte);
```

Parameter:

Zeiger, z.B. `TransceiverProperties`, auf eine Datenstruktur
(Zur Struktur siehe das File `PXI61xx_API.h` auf der mitgelieferten CD)

`LengthInByte`

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

Beschreibung:

Die Funktion `PXI61xx_TransceiverInfo` gibt Informationen über die Transceivereigenschaften zurück.

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

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

3.3.2.5 *Write Instruction*

Die Funktion `PXI61xx_WriteInstruction` dient zum Schreiben von Daten zum PXI 6161-Controller.

Format:

```
ViStatus PXI61xx_WriteInstruction(ViUInt8 *pData,  
                                 ViUInt16 DataLength);
```

Parameter:

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

`DataLength`

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

Beschreibung:

Die Funktion `PXI61xx_WriteInstruction` ermöglicht das Schreiben von Daten zum PXI 6161-Controller.

Im Header der Struktur, auf die `pData` zeigt, befindet sich die Information zum anzusprechenden PXI 6161-Board. Deshalb ist dieser Parameter nicht gesondert anzugeben.

3.3.2.6 *Read Response*

Die Funktion `PXI61xx_ReadResponse` dient zum Lesen von Daten vom PXI 6161-Controller.

Format:

```
ViStatus PXI61xx_ReadResponse(ViUInt8 Device,  
                              ViUInt8 *pData,  
                              ViUInt32 *DataLength);
```

Parameter:

Device

Index des PXI 6161-Boards, links beginnend mit 1
(Aber: Zählung einschließlich aller anderen PXI/ PCI 61xx-Boards
im selben System)

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

DataLength

Parameterwert vor Funktionsaufruf:
Größe des Puffers, auf den `pData` zeigt, in Bytes
Parameterwert nach Funktionsaufruf:
Anzahl der tatsächlich gelesenen Bytes

Beschreibung:

Die Funktion `PXI61xx_ReadResponse` ermöglicht das Lesen von Daten, die vom PXI 6161-Controller bereitgestellt wurden (siehe auch die Funktion `ReadResponse` im Abschnitt [Windows Device Treiber](#)).

3.4 Programmieren mit LabVIEW

3.4.1 LabVIEW über G-API

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe PXI 6161-Boards unter LabVIEW angesprochen werden können.

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

3.4.2 LLB unter Verwendung des Windows Device Treibers

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe PXI 6161-Boards unter LabVIEW angesprochen werden können.

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

3.4.3 LLB unter Verwendung des VISA Device Treibers

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe PXI 6161-Boards unter LabVIEW angesprochen werden können.

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

3.5 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.

A

Adressierung
Ethernet2-7

C

Controller
Alle Antworten lesen 3-11
Antwort lesen 3-10
Befehl senden.....3-9
Daten lesen 3-17
Daten schreiben.....3-16

E

Erweiterung
IO2-11
Ethernet 1-5, 2-7

G

G-API3-2
G-API Befehle..... 2-6, 2-7, 2-9,
..... 2-10
GÖPEL Firmware3-5

H

Hardware Explorer.....3-2
Hardwaretreiber
Statusabfrage3-7

I

IO Erweiterung.....2-11
Isolation2-6

K

Kommunikation
Masterframerate2-6
Kommunikationsschnittstellen
..... 2-10

L

LabVIEW
G-API 3-18
VISA 3-18
Windows 3-18

LabVIEW RT.....1-3

M

myCAR 3-18

O

OnBoard G-API3-3
OnBoard Schnittstellen.....2-9

P

Pinbelegung 2-8, 2-11
Programm-Generator 3-18
PROGRESS3-18
PXI 6161
Abmessungen2-3
Aufbau2-5
Eigenschaften2-1
Ethernet1-5
Hardware Installation1-1
Kennwerte2-4
Optionen 2-12
Struktur.....2-3
Treiber Installation1-2
VISA Treiber 1-3

S

Status-LEDs2-7
Steckverbinder
Front2-8

T

Transceiver
Information 3-8, 3-15

V

VISA session
Eröffnen 3-13
Schließen.....3-13
VISA Treiber3-12

W

Windows Treiber 1-2, 3-6