Series 61

PXI / PCI Controller Family

User Manual
(Translation of Original docu)
Document Version 1.7

GOEPEL electronic GmbH
Goeschwitzer Str. 58/60 • D-07745 Jena
+ 49-3641-6896-597 • ats_support@goepel.com • www.goepel.com

Printed: 08.01.2015

Issue: January 2015

1 Notes on the EC Declaration of Conformity

GOEPEL electronic GmbH
Goeschwitzer Straße 58-60
D-07745 Jena

With the EC Declaration of Conformity we declare the compliance of the GOEPEL electronic GmbH product described in this Manual with the requirements of the Directive
2006/95/EG – Low Voltage Directive and with the Directive 2004/108/EG about the Electromagnetic Compatibility.
Any modification to the product, not authorized by us, will invalidate the corresponding declaration.

The product is marked with the symbol CE

2 Board Installation

2.1 Hardware Installation

Before beginning with the hardware installation you have to ensure that your system is switched off and disconnected from the mains supply.

Please refer also to the user manual of your PXI/ PCI system for additional installation instructions that possibly have to be followed.

Electro Static Discharge (ESD) can harm your system and destroy electronic components. This can lead to irreparable damage on both the controller board and the system hosting the board as well as to unexpected malfunction of your test system.
Therefore do not touch the board surface or any connector pins and electronic components.

The PCI™, CompactPCI™ or PXI™ system is to be opened according to its conditions. A free slot is to be selected in your system. Now, the slot cover is to be taken away from the slot selected. To do this, unscrew the fixation screw(s) and remove the cover from the slot.

(If it is necessary to exchange transceiver modules,
pay attention to the general rules to avoid electro static discharging, see the warning above.
Transceiver modules must never be removed or mounted with the power switched on! Additionally, the right alignment is absolutely required.)

Insert the board carefully into the prepared slot. For PXI boards, use the lever at the front plate in order to push in the board finally.

When the board has been inserted properly, it is to be fixed by means of the screw(s) at the front plate.
Now, the board has been installed correctly.

Afterwards, carry out the operations required at the system to make it ready for operation anew.

2.2 Driver Installation

2.2.1 Windows Device Driver

PXI/ PCI 61xx boards can be operated under Windows® XP as well as under Windows® 7/ 32bit and Windows® 7/ 64bit.

Due to the plug and play capability of Windows®, for every newly recognized hardware component a driver installation is started automatically via the hardware assistant.
The hardware assistant can carry out the installation of the device driver by using the *inf* file contained on the enclosed CD.

It is not absolutely essential to restart the system.

The following step is only required in case you do not use the G-API.

If you want to create your own software for the boards, you possibly need additional files for user specific programming (**.LLB*, **.H*). These files are not automatically copied to the computer and have to be transferred individually from the supplied CD to your development directory.

2.2.2 VISA Device Driver

1st Step

Copy the
VISA_Driver PXI_PCI – W2K, WinXP (Version xx) folder
and for Windows® 7 additionally the
VISA_Driver PXI_PCI – Win7_x32_x64 (Version xx) folder
of the delivered CD/ folder *Series61xx* to your hard disk.
(Recommendation: Copy the complete folder(s) to *C:* \)

2nd Step

VISA for Windows® XP:

Due to the plug-and-play capability, for every newly recognized hardware component a driver installation is started automatically via the hardware assistant. Follow the instructions and enter as target directory the one which contains the *PXI61xx.inf* file
(according to the Recommendation above:
C: *VISA_Driver PXI_PCI* – W2K, WinXP *(Version xx)*.

VISA for Windows® 7:

Due to the plug-and-play capability, for every newly recognized hardware component a driver installation is started automatically via the hardware assistant. Follow the instructions and enter as target directory the one which contains the *PXI61xx_VISA.inf* file
(according to Recommendation above:
C: *VISA_Driver PXI_PCI – Win7_x32_x64 (Version xx)*.

VISA for LabVIEW RT:

For operating Series 61 boards under the RT operating system, use the *PXI61xx.inf* file of the
C: *VISA_Driver PXI_PCI – W2K, WinXP (Version xx)* folder.

Copy this file to the *ni-rt* *system* folder of the embedded controller (Recommendation: Copy by the NI Measurement Explorer).

If you intend to create a *startup.rtexe* later, copy also the *cvi_lvrt.dll* file to the *ni-rt* *system* folder.

3rd Step:

Reboot your computer to complete installation.

After Hardware Installation/ Driver Installation you may check whether the boards have been embedded properly by the system:

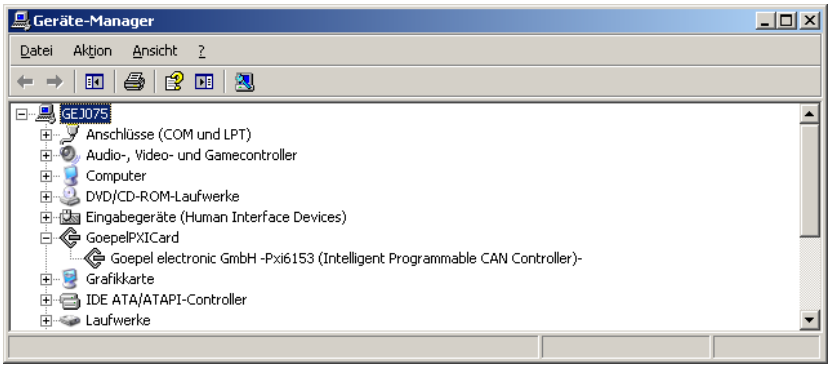

Figure 2-1: Windows

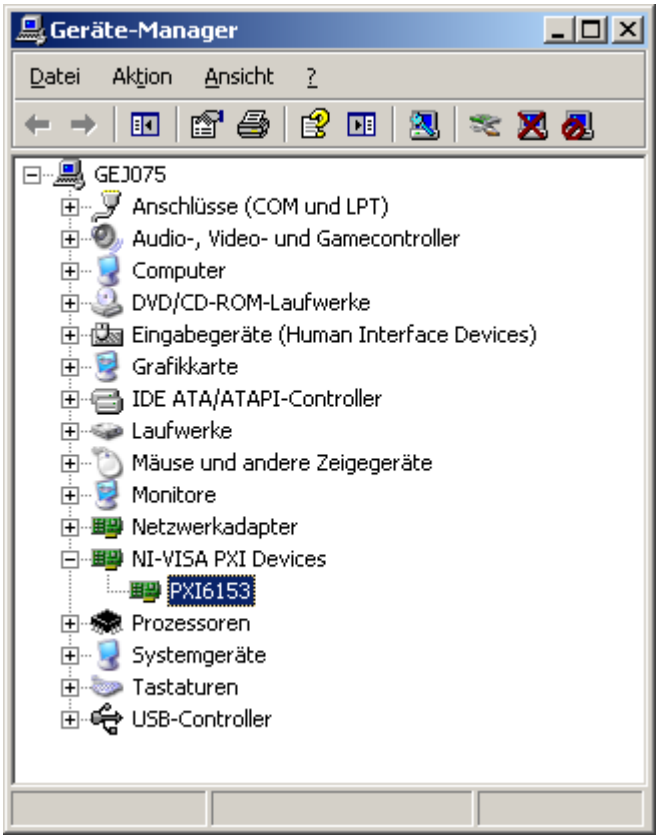

Figure 2-2: VISA for Windows XP

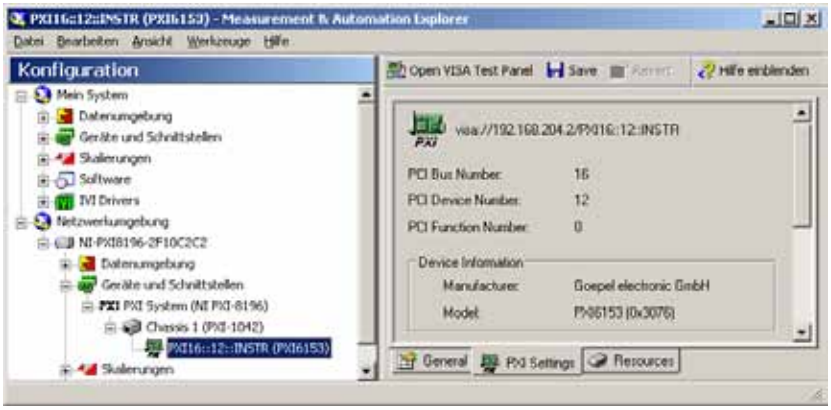

Figure 2-3: VISA for LabVIEW RT

2.2.3 Ethernet

If the Ethernet interface is used for communication with the control PC, there is no driver installation required.

The device can be directly addressed via the IP Address.
This IP Address can be changed by the HardwareExplorer.
The newly set IP Address becomes effective after a restart.

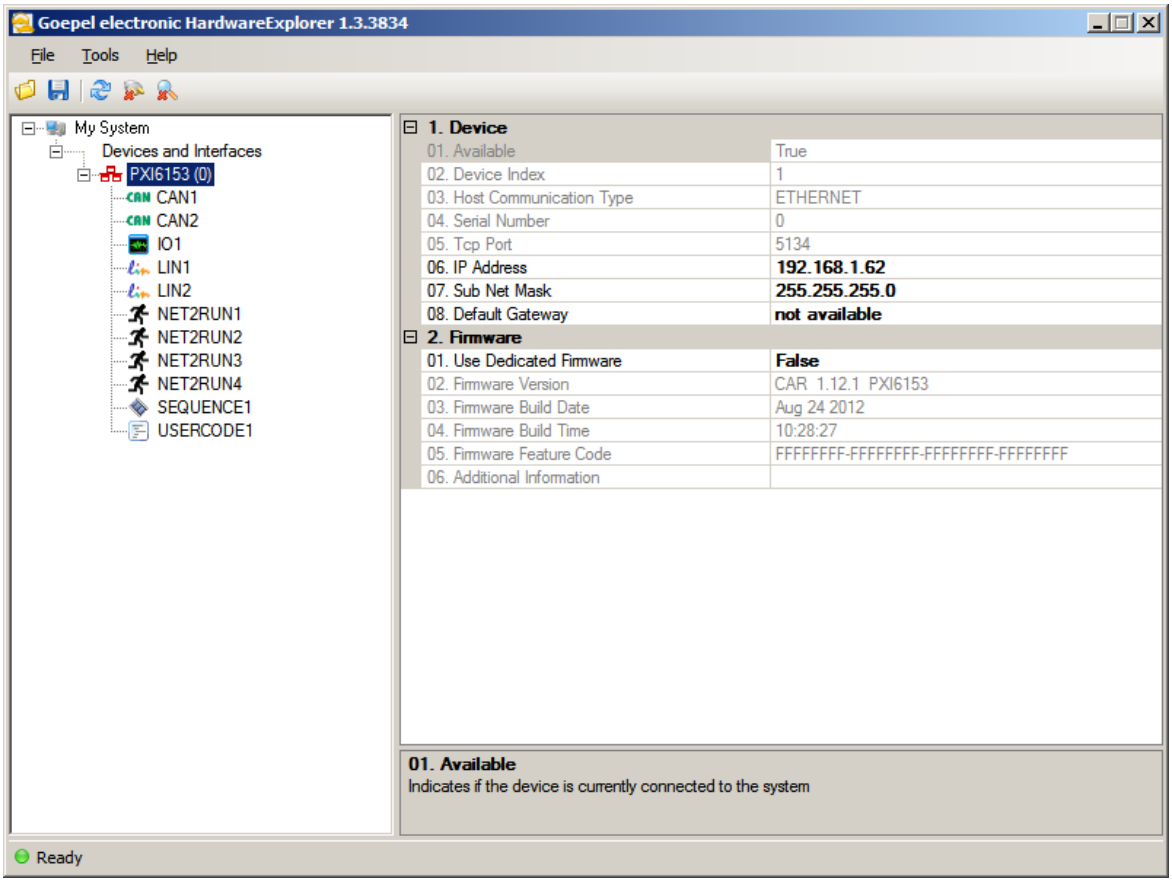

Figure 2-4: IP Address in the GOEPEL electronic HardwareExplorer

3 Hardware

3.1 Definition

PXI/ PCI 61xx (Series 61) controllers from GOEPEL electronic GmbH are programmable intelligent multibus controllers providing various communication interfaces for vehicle network testing and general control applications.

The Series 61 controller can generally provide the following features:

- Basic versions of the communication interfaces:
  PXI/ PCI 6153 – 2 CAN interfaces
  PXI/ PCI 6173 – 2 LIN or KLine interfaces
  PXI/ PCI 6181 – 1 CAN and 1 LIN or KLine interface
  PXI/ PCI 6191 – 2 FlexRay nodes with 2 channels each
  (see OnBoard Interfaces and FlexRay Extension Board)
- optionally 2 further (in case of PXI/ PCI 6191 4 further) CAN/ LIN/ KLine interfaces onboard
  (see OnBoard Interfaces)
- optionally 2 FlexRay nodes with 2 channels each on FlexRay Extension boards (not for PXI/ PCI 6191)
  (see FlexRay Extension Board)
- optionally instead of the second FlexRay Extension board 1 CAN Extension Board with 2 CAN interfaces
  (see CAN Extension Board)
- 4 digital inputs, 4 digital outputs with TTL level onboard
- optionally 4 additional digital inputs/ outputs with extended voltage level
  (see IO Extension Board)
- optionally 4 or 6 additional analog inputs/ outputs
  (see IO Extension Board)
- 600MHz Power PC with 512MB RAM, 256MB Flash
- Galvanic isolation between communication interfaces/ IO channels and user interface
- High flexibility through pluggable transceiver modules and Extension Boards
- Control of the devices via PXI/ PCI or Ethernet
  (see Addressing and Ethernet)
- The 1 Gbit Ethernet interface at the Front panel is also useable as volume data and debug interface
- Visualization of the operating states by four LEDs arranged at the front panel (see Status LEDs)

The figure below shows the PXI 6153 controller board in its maximum hardware configuration with optional plug-in modules (but without CAN Extension Board):

Figure 3-1: PXI 6153

The PCI 6153 board is shown in its basic version (2x CAN):

Figure 3-2: PCI 6153

3.2 Technical Data

3.2.1 General

The PXI 61xx controller board is a slot-in board developed for the PXI™ bus. **P**CI e**X**tensions for **I**nstrumentation (PXI) is a modular instrumentation platform originally introduced in 1997 by National Instruments and now promoted by the PXI Systems Alliance (PXISA).

PXI™ is based on the CompactPCI™ bus, and it offers all of the benefits of the PCI architecture including performance, industry adoption and COTS technology. PXI adds a rugged CompactPCI mechanical form-factor, an industry consortium that defines hardware, electrical, software, power and cooling requirements, leaving nothing to chance. Then PXI™ adds integrated timing and synchronization that is used to route synchronization clocks, and triggers internally. PXI™ is a future-proof technology, and is designed to be simply and quickly reprogrammed as test, measurement, and automation requirements change.

The PXI 61xx controller board operates as PXI slave, therefore the board may be plugged into any desired slot of a PXI chassis except slot 1. The PCI Plug & Play auto detection mechanism is supported by the PXI 61xx controller board. No jumper configuration is needed for PXI integration.

CompactPCI and PXI products are interchangeable, i.e. they can be used in either CompactPCI or PXI chassis, but installation in the alternate chassis type may eliminate certain clocking and triggering features. So for example you could mount a CompactPCI network interface controller in a PXI rack to provide additional network interface functions to a test stand. Conversely, a PXI module installed in a CompactPCI chassis would not utilize the additional clocking and triggering features of the PXI module.

The PXI 61xx controller board is already prepared for use in a PXI Express hybrid slot, which delivers support for both the PCI and the PCI Express bus by taking advantage of available pins on the high-density backplanes.

The PCI 61xx controller board was developed as a slot-in board for the PCI bus system Rev. 3.0. It supports a 32bit wide data transmission at 33MHz, in 3.3V systems as well as in 5V systems.

The timing and synchronization mechanisms known from PXI™ systems are also provided from PCI 61xx controller boards via the TriggerConnector.

3.2.2 Dimensions

The PXI/ PCI 61xx controller board is a 3U standard module and occupies one slot width.

Board dimensions without bracket and handle:

- PXI 61xx: 160 mm x 100 mm (L x W)
- PCI 61xx: 168 mm x 107 mm (L x W)

3.2.3 Series61 Technical Data

The PXI/ PCI 61xx controller board has the following technical specification:

Symbol	Indication	Min.	Typ.	Max.	Unit	Remarks
	CAN/ LIN/ KLine Interfaces (onboard)	2		4		See OnBoard Interfaces
	FlexRay Extension Board			2		See FlexRay Extension Board
	CAN Extennsion Board			1		See CAN Extension Board
	IO Resources Extension			1		See IO Extension Board
	Digital inputs (onboard)					
N	Number			4		
U_{IH}	High-level input voltage	3.5		5.5	V	
U_{IL}	Low-level input voltage			1.5	V	
I_L	Input leakage current			35	µA	
	Digital outputs (onboard)					
N	Number			4		
U_{OH}	High-level output voltage	4.8		5	V	
U_{OL}	Low-level output voltage			0.5	V	
I_{OUT}	Output current			8	mA	

If required, the CAN Extension Board with two CAN interfaces including Transceivers is plugged in at the same position as usually the second FlexRay Extension board.

3.3 Construction

3.3.1 General

The core of the PXI/ PCI 61xx controller board builds a strong 600MHz AMCC 460EX PowerPC. This dual-issue, superscalar 32bit RISC CPU is based on the Book-E enhanced PowerPC architecture. With features including out-of-order execution, dynamic branch prediction and a highly pipelined double precise floating-point unit, this processor provides the calculation power required for processing complex residual bus simulation on multiple bus interfaces. Furthermore the controller comes equipped with a 512MB fast 400MHz DDR2 RAM and 256MB Flash memory, of which over 80% is available for user programs.

The PXI/ PCI 61xx controller board has been designed as highly flexible multibus controller platform. Providing up to four universal serial bus nodes, where each can be configured to operate as either CAN or LIN (KLine) interface. Each node has a transceiver socked assigned to it. The transceiver module that is plugged into that socked defines the interface that operates the node.

For example: If a CAN transceiver is plugged into transceiver socked TXR2 then this node connects to interface CAN 2 (ID 2). If a LIN transceiver is used instead then this node connects to interface LIN 2 (ID 6). The software addresses the interfaces by their interface ID (ID 2 and ID 6 in the example).

In addition to the four universal serial bus nodes explained before, the PXI/ PCI 61xx controller board provides three extension sockets.
Two of them are prepared to take on a dual channel FlexRay Extension board. Whereas the third socked is prepared to carry a multi IO Exptension board. This allows to plug-in up to two FlexRay Extension boards into the PXI/ PCI 61xx controller board. Each of them with an independent FlexRay controller and two FlexRay transceivers, providing full dual channel functionality. The FlexRay nodes are statically mapped to the interfaces 13 and 14. Please refer to the FlexRay Extension Board section for detailed information.

Optionally a PXI/ PCI 61xx board can be equipped with a CAN Extension Board with two CAN interfaces instead of the second FlexRay Extension board (see CAN Extension Board). These two CAN nodes are statically mapped to the interfaces 15 und 16.

With this, up to six CAN interfaces are possible for a PXI/ PCI 61xx board.

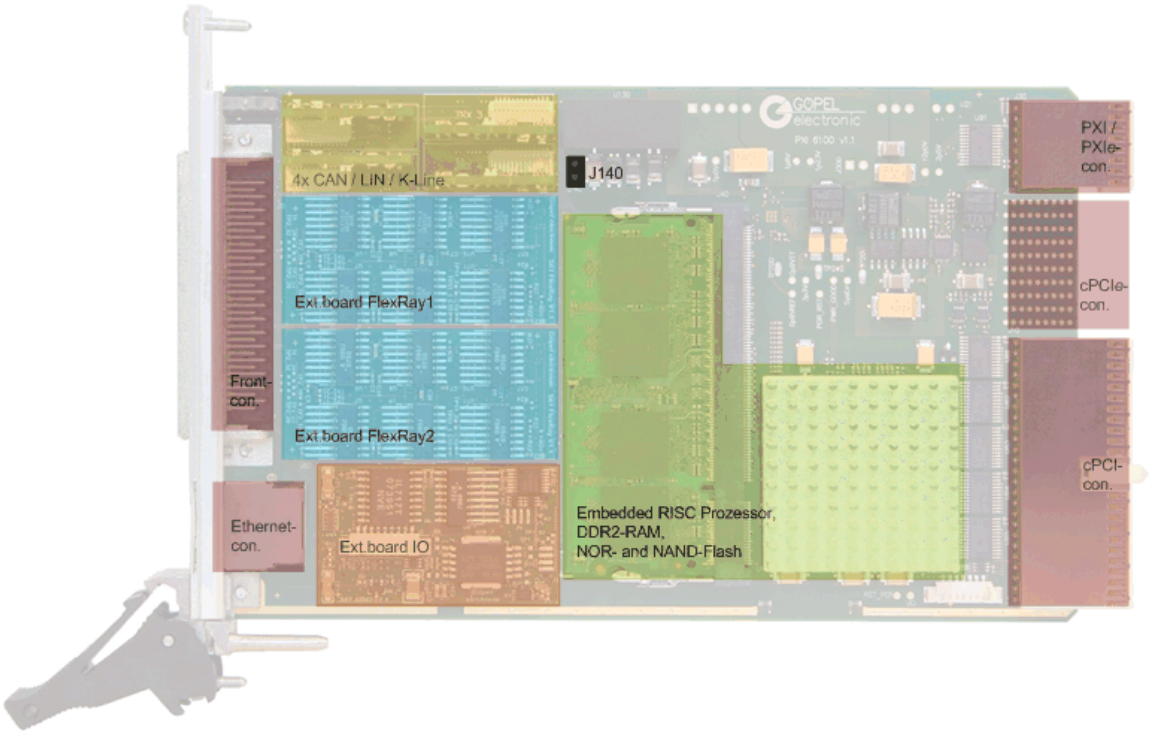

Figure 3-3: PXI 61xx board (schematically)

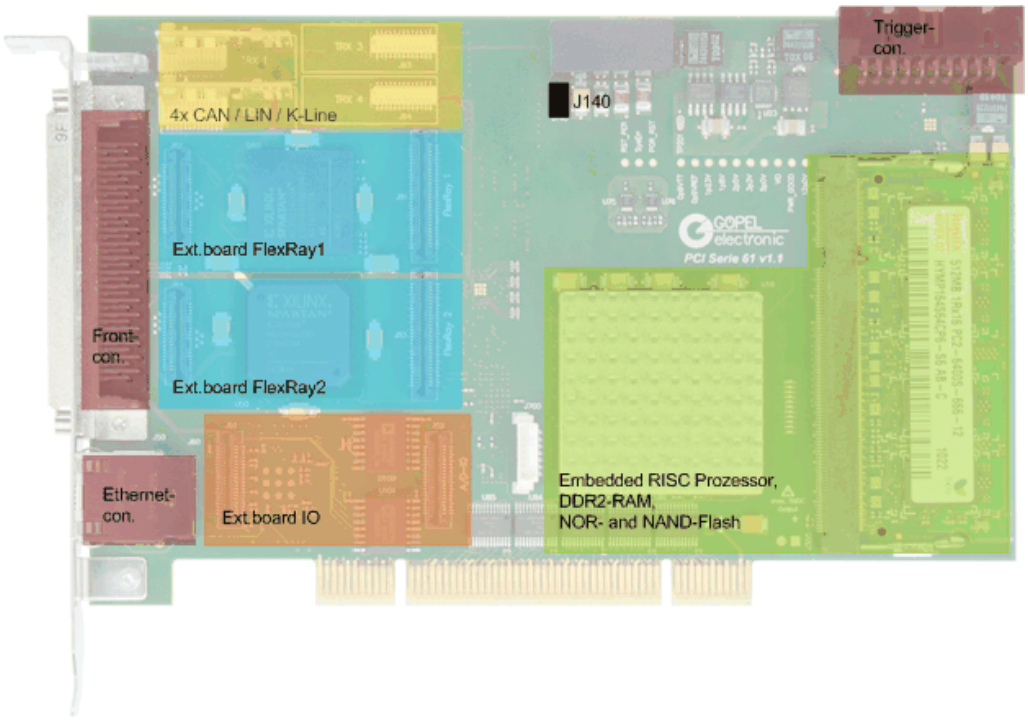

Figure 3-4: PCI 61xx board (schematically)

The PXI/ PCI 61xx controller base board provides 4 digital inputs and 4 digital outputs. By inserting a multi IO module into the third socket, the IOs can be extended by up to 4 digital inputs and 4 digital outputs plus 6 analog inputs and 6 analog outputs.
Please refer to the IO Extension Board section for detailed information.

The front side of the PXI/ PCI 61xx controller board features a 68 pin male SCSI connector, providing access to all bus interfaces and auxiliary IO. Above the SCSI connector there are 4 status LEDs placed, which are indicating the operation states on the controller board (see Status LEDs).

An Ethernet jack providing access to the 1Gbit Ethernet interface is placed underneath the SCSI connector.
It is used for the control of the PXI/ PCI 61xx controller instead of PXI/ PCI, or it serves as debug interface for user programs as well as volume data interface (e.g. for streaming monitoring data).

The table below shows the mapping of communication interfaces:

Interface ID	Interface	Transceiver position/ Extension Board
1	CAN 1	TRX 1
2	CAN 2	TRX 2
3	CAN 3	TRX 3
4	CAN 4	TRX 4
5	LIN 1	TRX 1
6	LIN 2	TRX 2
7	LIN 3	TRX 3
8	LIN 4	TRX 4
9	KLine 1	TRX 1
10	KLine 2	TRX 2
11	KLine 3	TRX 3
12	KLine 4	TRX 4
13	FlexRay 1	FlexRay 1
14	FlexRay 2	FlexRay 2
15	CAN 5	FlexRay 2
16	CAN 6	FlexRay 2

Please refer also to sections OnBoard Interfaces, FlexRay Extension Board, CAN Extension Board and Figure 3-3/ Figure 3-4.

3.3.2 Addressing

PXI 61xx boards provide a 1Gbit Ethernet interface and a PXI interface.

Both interfaces can be used for the communication of the unit with the host PC.

In case of using the Ethernet interface, the boards can be controlled via the default IP Address 192.168.1.62, Port 5134, which can be changed if required (see also Driver Installation/ Ethernet).

In principle, there are two ways for this:

- HardwareExplorer: Select the device,
  under Device set the required IP Address;
  the new IP Address is effective after restart
- G API Command G_Common_Ethernet_IpAddress_Set;
  the new IP Address is effective after restart

PXI 61xx: PXI racks do have an own geographical slot addressing of the backplane.
Numbering starts with 1 and can be seen at the cover's front side.
Mount always an embedded controller or an MXI card at slot 1.

A PXI 61xx board can read out this geographical slot address.

3.3.3 Isolation

Electric surges can harm expensive test equipment and lead to unreliable test results. Electric isolation protects against electric surges and can help to suppress dangerous electrical transients. It also eliminates ground loops, responsible for data errors due to ground potential differences.

The PXI/ PCI 61xx controller board provides electric isolation between the PXI system and all the IO signals available at the front connector. This includes the CAN, LIN (KLine) and FlexRay communication interfaces as well as digital and analog IOs. The system requires a ground reference between the GNDiso potential on the front connector and the ground potential of the device under test (ECU, etc.).

Jumper J140 provides a means of introducing a ground reference between the GNDiso potential on the front connector and the ground potential of the PXI system.

Caution: Introducing a ground reference to the PXI system may could result in high current flow over the test lead and the PXI/ PCI 61xx controller board.

This could lead to malfunction, wrong test results as well as damage the controller board or other test equipment.

Before closing jumper J140, you must ensure that the device under test and all other devices connected to the front connector of the controller board are only supplied via an **isolated power supply**!

3.3.4 Status LEDs

The LEDs arranged at the front panel indicate the current operation state of a PXI/ PCI 61xx board:

Figure 3-5: Status LEDs

The table below describes the meaning of the LED states:

LED 1	LED 2	LED 3	LED 4	Remarks
Permanently ON				Controller does not run (error!)
Alternately blinking				Bootloader software runs
	blinking			Firmware runs
ON (shortly)				State during execution of a Firmware command
			ON	Ethernet connection established

3.3.5 Connector Pinout

Used connector: SCSI 68 poles male
Connector for connection cable: SCSI 68 poles female

The following table shows the pinout of the frontal connector:

Pin	Signal				Pin	Signal	
1	**CAN1_H/**	**LIN1**	**K-Line1**		35	R_{low}-CAN1_H	*) $UBat_{extern\_iso1}$
2	**CAN1_L**		**L-Line1**		36	R_{low}-CAN1_L	*) GND_{iso1}
3	GNDiso				37	$UBAT_{extern1}$	*) do not connect!
4	**CAN2_H**	**LIN2**	**K-Line2**		38	R_{low}-CAN2_H	*) $UBat_{extern\_iso2}$
5	**CAN2_L**		**L-Line2**		39	R_{low}-CAN2_L	*) GND_{iso2}
6	GNDiso				40	$UBAT_{extern2}$	*) do not connect!
7	**CAN3_H/**	**LIN3**	**K-Line3**		41	R_{low}-CAN3_H	*) $UBat_{extern\_iso3}$
8	**CAN3_L/**		**L-Line3**		42	R_{low}-CAN3_L	*) GND_{iso3}
9	GNDiso				43	$UBAT_{extern3}$	*) do not connect!
10	**CAN4_H/**	**LIN4**	**K-Line4**		44	R_{low}-CAN4_H	*) $UBat_{extern\_iso4}$
11	**CAN4_L**		**L-Line4**		45	R_{low}-CAN4_L	*) GND_{iso4}
12	GNDiso				46	$UBAT_{extern4}$	*) do not connect!
13	**FlexRay1A_BP**				**47**	**FlexRay1B_BP**	
14	**FlexRay1A_BM**				**48**	**FlexRay1B_BM**	
15	GNDiso				49	GNDiso	
16	**FlexRay2A_BP**		**CAN5_H**		**50**	**FlexRay2B_BP**	**CAN6_H**
17	**FlexRay2A_BM**		**CAN5_L**		**51**	**FlexRay2B_BM**	**CAN6_L**
18	GNDiso				52	GNDiso	
19	DIGITAL_OUT1				53	DIGITAL_IN1	
20	DIGITAL_OUT2				54	DIGITAL_IN2	
21	DIGITAL_OUT3				55	DIGITAL_IN3	
22	DIGITAL_OUT4				56	DIGITAL_IN4	
23	**IO_EXP1**				**57**	**IO_EXP11**	
24	**IO_EXP2**				**58**	**IO_EXP12**	
25	**IO_EXP3**				**59**	**IO_EXP13**	
26	**IO_EXP4**				**60**	**IO_EXP14**	
27	GNDiso				61	$UEXT_{IO}$	
28	**IO_EXP5**				**62**	**IO_EXP15**	
29	**IO_EXP6**				**63**	**IO_EXP16**	
30	**IO_EXP7**				**64**	**IO_EXP17**	
31	**IO_EXP8**				**65**	**IO_EXP18**	
32	**IO_EXP9**				**66**	**IO_EXP19**	
33	**IO_EXP10**				**67**	**IO_EXP20**	
34	GNDiso				68	GNDiso	

The pinout of the **Communication interfaces** is in accordance with the plugged-in transceiver resp. the additional FlexRay or CAN board (see OnBoard Interfaces, FlexRay Extension Board and CAN Extension Board, while the **pins 23..33** and **57..67** have different functionality according the type of the used IO Extension Board.

*) For isolated LIN Transceivers

3.3.6 Trigger-Connector PCI 61xx

According to the trigger and synchronization mechanisms of the PXI technology, the PCI 61xx controller board has a separate TriggerConnector. The functionality of the TRG0..7 and CLK10MHz signal pins is the same as that of the PXI 61xx board.

For the Trigger signals, 5V at most are permissible.

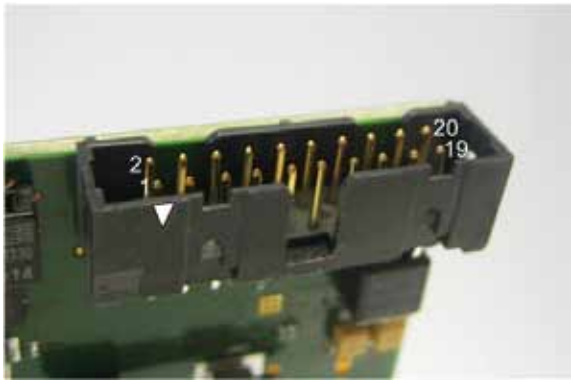

Figure 3-6: Trigger Connector PCI 61xx

Pinout of the TriggerConnector for PCI 61xx:

Pin	Signal	Pin	Signal
1	Gnd		
3	Gnd	4	TRG 7
5	Gnd	6	TRG 0
7	Gnd	8	TRG 1
9	Gnd	10	TRG 2
11	Gnd	12	TRG 3
13	Gnd	14	TRG 4
15	Gnd	16	TRG 5
17	Gnd	18	TRG 6
19	Gnd	20	CLK10MHz

3.3.7 OnBoard Interfaces

The PXI/ PCI 61xx controller board has up to four communication interfaces, designed as CAN, LIN or KLine interface (also mixed up possible as multibus variant). Two of the communication interfaces have a preferred basic assignment, see Definition. The two others may remain empty, but can optionally designed as further CAN, LIN or KLine interfaces.

If required, the assignment can be changed by changing the belonging transceiver (even the preferred assignment!).

Warning

Please proceed extremely carefully when changing transceivers, and pay attention to their position and orientation.

The position of the transceivers and their assignment to the interface number is shown in the following figure:

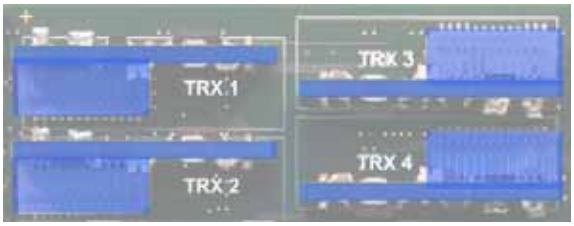

Figure 3-7
Transceiver placement

Each Transceiver type is coded and can be identified clearly.
For the available types of transceivers see Product Information.

All four interfaces are supplied by an internal voltage of 12V ($UBAT_{intern}$). In case of using other voltages, this internal voltage can be switched off individually by software.

G-API commands
G_Can_Node_InternalVBat_Disable
G_Lin_Node_InternalVBat_Disable or
G_KLine_Node_InternalVBat_Disable

Then, an external voltage ($UBAT_{extern}$) must be supplied via the predefined pins of the frontal connector.

In case the internal power supply should be later used again, proceed the

G_Can_Node_InternalVBat_Enable
G_Lin_Node_InternalVBat_Enable or
G_Kline_Node_InternalVBat_Enable G-API commands.

Symbol	Identification	min.	typ.	max.	Unit	Remarks
CAN V2.0B Interfaces onboard						
C	Transmission rate			1	Mbit/s	
$UBAT_{intern}$	internal battery voltage		12		V	detachable
$UBAT_{extern}$	external Battery voltage			27	V	
RCAN	Termination high-speed transceiver		120		Ω	detachable
RCAN	Termination low-speed transceiver		10		kΩ	
LIN V2.1 Interfaces onboard						
C	Transmission rate			19.2	kbit/s	
$UBAT_{intern}$	internal battery voltage		12		V	detachable
$UBAT_{extern}$	external battery voltage		12	27	V	
RLIN	Pullup Resistor	1	30		kΩ	switchable Master/Slave
KLine Interfaces onboard						
	Transmission rate			9.6	kbit/s	
$UBAT_{intern}$	internal battery voltage		12		V	detachable
$UBAT_{extern}$	external battery voltage		12	27	V	

Notes on R_{CAN} for the high speed transceiver:
The 120Ω bus terminating resistor can be deactivated by software (G-API command G_CAN_Node_BusTermination_Disable, activation anew by G_CAN_Node_BusTermination_Enable).

Notes on R_{CAN} for den low speed transceiver:
The internal 10kΩ bus terminating resistor can be reduced if required by adding external resistors.
Connect the external resistors, if applicable, between the pins with the signals R_{low}-CANx_H and CANx_H/ R_{low}-CANx_L and CANx_L (see Connector Pinout).

Notes on RLIN: The 1kΩ pullup resistor corresponds to the LIN Master bus termination and can be activated by software
(G-API command G_Lin_PullUpResistor_Enable à Master, Deactivating by G_Lin_PullUpResistor_Disable à Slave).
If it is not active, the internal termination resistor of the LIN transceiver becomes effective (typically 30kΩ for TJA1020).

3.3.8 FlexRay Extension Board

The PXI/ PCI 61xx controller board has two extension sockets on the top side prepared to take on a FlexRay plug-in Extension board (see Figure 3-3, Figure 3-4). Each board has an independent FlexRay controller and two FlexRay transceivers, providing full dual channel functionality.

For a PXI/ PCI 61xx Controller board with CAN Extension Board, only one FlexRay Extension board can be plugged in.

Each FlexRay plug-in module provides the following features:

- FlexRay controller (Freescale MFR4310)
- FlexRay 2.1 protocol compliant
- supports FlexRay transmission rates of 10, 8, 5 and 2.5Mbit/s
- 2 FlexRay transceivers (NXP TJA 1080)
- wake-up detection
- switchable termination resistors
- full galvanic isolation
- isolated power supply of transceivers

The following table shows the main characteristics of the FlexRay module:

Symbol	Identification	min.	typ.	max.	Unit	Remarks
FlexRay Interface						
C	Transmission rate	2.5		10	Mbit/s	per channel
R_{FR}	Termination resistor		100		Ω	detachable

Notes on R_{FR}:
The 100Ω bus terminating resistor can be deactivated by software (G-API command G_FlexRay_Node_BusTermination_Disable, activation anew by G_FlexRay_Node_BusTermination_Enable).

When configured with two FlexRay modules, both FlexRay modules can be used jointly to startup a FlexRay cluster. In this case one node will be the leading cold starter and the other one the following cold starter.

In cases where the ECU under test is a cold start node itself a single module could start up the cluster. This way the second module could be used to operate a second FlexRay cluster independently.

3.3.9 CAN Extension Board

In the case more CAN interfaces are required, a CAN Extension Board with two CAN interfaces can be plugged in the position of FlexRay Node B (generally for CAN5 and CAN6).
The TJA1041A highspeed transceivers for these interfaces can not be substituted by other transceiver types. Additionally, no external supply by $UBAT_{ext}$ is possible; supply is effected with the fixed $UBAT_{int}$ (12V) voltage.
The terminating resistor for both transceivers can be switched off.

CAN V2.0B Interfaces generally Node 5..6 (optionally)					
Transmission rate			1	Mbit/s	
Internal battery voltage		12		V	
Termination for high-speed transceiver		120		Ω	detachable

3.3.10 IO Extension Board

Additional analog and digital inputs and outputs as well as various other interfaces become available by plugging in this extension board.

GOEPEL electronic GmbH offers two different types:
Type1 and Type2.

The Type1 IO Extension board has additional resources as follows:

Symbol	Indication	min.	typ.	max.	Unit	Remarks
Digital inputs						
N	Number of inputs			4		
U_{IH}	High-level input voltage	3.5		5.5	V	
U_{IL}	Low-level input voltage			1.5	V	
I_L	Input leakage current			35	µA	
Digital outputs						
N	Number of outputs			4		
U_{OH}	High-level output voltage	4.8		5	V	
U_{OL}	Low-level output voltage			0.5	V	
I_{OUT}	Output current			8	mA	
Analog inputs						
N	Number of inputs			6		
U_{IN}	Input voltage	0		10	V	
R	Resolution			10	bit	
Analog outputs						
N	Number of outputs			6		
U_{OUT}	Output voltage	0		10	V	
	Resolution			10	bit	

The following table shows the pinout of the frontal connector when the Type1 IO Extension board is plugged in:

Pin	Signal		Signal	Pin
23	DIGITAL_OUT5		DIGITAL_IN5	57
24	DIGITAL_OUT6		DIGITAL_IN6	58
25	DIGITAL_OUT7		DIGITAL_IN7	59
26	DIGITAL_OUT8		DIGITAL_IN8	60
27	GNDiso		$UEXT_{IO}$	61
28	ANALOG_OUT1		ANALOG_IN1	62
29	ANALOG_OUT2		ANALOG_IN2	63
30	ANALOG_OUT3		ANALOG_IN3	64
31	ANALOG_OUT4		ANALOG_IN4	65
32	ANALOG_OUT5		ANALOG_IN5	66
33	ANALOG_OUT6		ANALOG_IN6	67
34	GNDiso		GNDiso	68

The Type2 IO Extension board has additional resources as follows:

Symbol	Indication	min.	typ.	max.	Unit	Remarks
Digital inputs						
N	Number of inputs			4		
U_{IH}	High-level input voltage	10		25	V	
U_{IL}	Low-level input voltage			3.9	V	
I_L	Input resistance		49		kΩ	
Digital outputs						
N	Number of outputs			4		
U_{OH}	High-level output voltage			25	V	Supply via the $UEXT_{IO}$ pin necessary
U_{OL}	Low-level output voltage		open		V	Integrated recovery diode
I_{OUT}	Output current			200	mA	
Analog inputs						
N	Number of inputs			4		
U_{IN}	Input voltage	0		25	V	
	Resolution			10	bit	
R_L	Input resistance		125		kΩ	
Analog outputs						
N	Number of outputs			4		
U_{OUT}	Output voltage	0		25	V	Supply via the $UEXT_{IO}$ pin necessary
I_{OUT}	Output current per channel			10	mA	
	Resolution			10	bit	
External supply voltage input U_{EXT_IO}						
$UEXT_{IO}$	External supply voltage	7		26	V	

The following table shows the pinout of the frontal connector when the Type2 IO Extension board is plugged in:

Pin	Signal		Signal	Pin
23	DIGITAL_OUT5		DIGITAL_IN5	57
24	DIGITAL_OUT6		DIGITAL_IN6	58
25	DIGITAL_OUT7		DIGITAL_IN7	59
26	DIGITAL_OUT8		DIGITAL_IN8	60
27	GNDiso		$UEXT_{IO}$	61
28	ANALOG_OUT1		ANALOG_IN1	62
29	ANALOG_OUT2		ANALOG_IN2	63
30	ANALOG_OUT3		ANALOG_IN3	64
31	ANALOG_OUT4		ANALOG_IN4	65
32	Do not connect!		Do not connect!	66
33	Do not connect!		Do not connect!	67
34	GNDiso		GNDiso	68

3.4 Product Information

The Series 61 intelligent programmable multibus controller family is a highly customizable controller platform. Currently the series consist of four base variants for CAN, LIN/ KLine, FlexRay and Multibus systems that can be combined with a whole number of options.

Please refer to the list below for all the options available.

PXI/ PCI 6153 CAN controller for Windows XP/ Windows 7	
PXI/ PCI 6153	CAN controller with 2 CAN nodes and 2 CAN transceiver modules as well as 4 digital inputs and 4 digital outputs (all onboard)
PXI/ PCI 6173 LIN/ KLine controller for Windows XP/ Windows 7	
PXI/ PCI 6173	LIN controller with 2 LIN nodes and 2 LIN transceiver modules OR KLine controller with 2 KLine nodes and 2 KLine transceiver modules as well as 4 digital inputs and 4 digital outputs (all onboard)
PXI/ PCI 6181 Multi bus controller for Windows XP/ Windows 7	
PXI/ PCI 6181	Multi bus controller with 1 CAN node with 1 CAN transceiver module and 1 LIN node and 1 LIN transceiver module OR 1 KLine node and 1 KLine transceiver as well as 4 digital inputs and 4 digital outputs (all onboard)
PXI/ PCI 6191 FlexRay controller for Windows XP/ Windows 7	
PXI/ PCI 6191	FlexRay controller with 2 FlexRay nodes and 2 dual channel FlexRay modules (on FlexRay Extension board) as well as 4 digital inputs and 4 digital outputs (onboard)

Options for PXI/ PCI 61xx controller boards	
CAN Node	Further CAN node for PXI/ PCI 61xx onboard to upgrade on 3 or 4 communication nodes, incl. transceiver module Note: The total quantity of installable CAN/ LIN/ KLine nodes at the same time amounts 4 per PXI/ PCI 61xx board without CAN Extension board
CAN Extension board	Additional CAN nodes for PXI/ PCI 61xx generally to upgrade on 5 and 6 CAN nodes, incl. transceiver modules Note: The total quantity of installable additional CAN Extension boards amounts 1 per PXI/ PCI 61xx board
LIN node	Further LIN node for PXI/ PCI 61xx onboard to upgrade on 3 or 4 communication nodes, incl. transceiver module Note: The total quantity of installable CAN/ LIN/ KLine nodes at the same time amounts 4 per PXI/ PCI 61xx board without CAN Extension board
KLine node	Further KLine node for PXI/ PCI 61xx onboard to upgrade on 3 or 4 communication nodes, incl. transceiver module Note: The total quantity of installable CAN/ LIN/ KLine nodes at the same time amounts 4 per PXI/ PCI 61xx board without CAN Extension board
FlexRay node	Additional FlexRay node for PXI/ PCI 61xx, (but not for PXI/ PCI 6191), incl. dual channel FlexRay module, FlexRay controller MFR 4310 with 2 transceivers type TJA 1080 (at FlexRay Extension board) Note: The total quantity of installable FlexRay nodes at the same time amounts 2 per PXI/ PCI 61xx board (1 if the CAN Extension Board is installed); This option is usable independent from and additional to options CAN/ LIN/ KLine nodes and IO Extension board

For the continuation of this table please see next page.

Options for PXI / PCI 61xx controller boards	
IO Extension board Type 1	General Input/ Output module for PXI/ PCI 61xx, incl. 6 analog inputs, 6 analog outputs, 4 digital inputs and 4 digital outputs Note: The total quantity of installable IO Extension boards at the same time amounts 1 per PXI/ PCI 61xx board; this option is usable independent from and additional to options CAN/ LIN/ KLine and FlexRay
IO Extension board Type 2	General Input/ Output module for PXI/ PCI 61xx, incl. 4 analog inputs, 4 analog outputs, 4 digital inputs, 4 digital outputs Note: The total quantity of installable IO Extension boards at the same time amounts 1 per PXI/ PCI 61xx board; this option is usable independent from and additional to options CAN/ LIN/ KLine and FlexRay
CAN TJA1054	CAN low speed transceiver type TJA1054
CAN TJA1041A	CAN high speed transceiver type TJA1041A
CAN NCV7356D1G	CAN single wire transceiver type NCV7356D1G
LIN TJA1020	LIN transceiver type TJA1020
LIN TJA1020 Iso	LIN transceiver type TJA1020 isolated channel selective
LIN TLE7259G	LIN transceiver type TLE7259G
K-Line L9637D	KLine transceiver type L9637D
K-Line L9637D Iso	K-Line transceiver type L9637D isolated channel selective
RS232 TRSF3221E	RS232 transceiver type TRSF3221E

For the continuation of this table please see next page.

Options for PXI / PCI 61xx controller boards	
DIAG KW2000 TP1.6	Keyword 2000 based on TP1.6 CAN Diagnostic software on board for PXI/ PCI 61xx
DIAG KW2000 TP2.0	Keyword 2000 based on TP2.0 CAN Diagnostic software on board for PXI/ PCI 61xx
DIAG KW2000 ISO-TP	Keyword 2000 based on CAN-ISO-TP CAN Diagnostic software on board for PXI/ PCI 61xx
DIAG UDS ISO-TP	UDS based on CAN-ISO-TP CAN Diagnostic software on board for PXI/ PCI 61xx
DIAG GMLan	GMLan CAN Diagnostic software on board for PXI/ PCI 61xx
DIAG J1939	J1939 CAN Diagnostic software on board for PXI/ PCI 61xx
CAL CCP2.1	CAN calibration protocol CCP2.1 for PXI/ PCI 61xx
LIN adv-lib	Advanced library for test of the LIN-protocol specific. 2.0/ 2.1 for PXI/ PCI 61xx
Net2Run	Software tool to generate signal based Rest bus Simulation(s) in heterogeneous car networks. This software solution is based on the AUTOSAR approach. Direct signal access (reading and manipulation) is provided via G-API functions. Further Net2Run features a gateway routing editor with PDU and signal mapping functionality. Net2Run supports the automatic import of bord net data in the **.dbc*, **.ldf* and *Fibex* formats.
Net2Run Runtime	Runtime module for executing the rest bus simulation files (**.rbs* files) created by Net2Run. This option is necessary for each PXI/ PCI 61xx board.
Net2Run IDE	Software programming environment (Windows host) to build G-API based on-board UserCode programs for PXI/ PCI 61xx boards; includes: Net2Run IDE, QNX Neutrino CLT, G-API on-board libraries, single developer license
UserCode Runtime	UserCode runtime module for the execution of G-API based on-board UserCode programs on PXI/ PCI 61xx boards; This option is necessary for each PXI/ PCI 61xx board.

4 Software

There are several ways for the embedding of PXI/ PCI 61xx boards in your own applications:

- G-API Programming
- UserCode Programming
- Programming via DLL Functions
  (Windows Device Driver and VISA Device Driver)
- Programming with LabVIEW

4.1 G-API Programming

The G-API (GOEPEL-API) is the C-based user interface for GOEPEL electronic hardware under Windows®.
It provides a wide hardware independent command set for CAN, LIN/ KLine, FlexRay, MOST, LVDS, ADIO and Diagnostic services. No matter whether a PXI/ PCI, USB and Ethernet device is used, the commands remain the same.

The hardware abstraction introduced with the G-API gives the test application parallel access to the hardware, allowing one application to access multiple hardware interfaces. As well as multiple applications can access the same hardware interface in parallel.

Another feature introduced by the G-API is the asynchronous hardware access. This means no execution blocking for pending firmware commands. The command acknowledgement is provided via a callback mechanism.

With the HardwareExplorer (see also Ethernet) GOEPEL electronic provides an efficient hardware configuration and management tool, offering users an easy way to manage their hardware configurations and identifying specific hardware interfaces by logical names. Using logical interface names in the application saves from rebuilding the application when porting it to another interface or controller board, as the interface can be easily reassigned in the HardwareExplorer.
Furthermore the HardwareExplorer provides a simple means of testing the interaction between hardware and software by executing the integrated self-tests.

Please consult the G-API documentation for further information.
This documentation and the installation software are located in the *G-API* folder of the supplied “Product Information” CD.

4.2 UserCode Programming

PXI/ PCI 61xx boards can execute user programs direct on their PowerPC processor. This requires the UserCode run-time module being enabled.

The UserCode run-time module is an option for PXI/ PCI 61xx boards (plus other GOEPEL devices) and requires one license per unit.

Executing programs directly on the PowerPC improves the real-time performance remarkable and frees up PCI bandwidth on the host system.

Therefore GOEPEL electronic has ported and enhanced by additional on-board functionality their C-programming user interface called G-API from Windows® to the QNX Neutrino real-time operating system.
The QNX Neutrino real-time operating system is based on a micro kernel architecture, providing clear separation between the kernel and each individual application.
This allows user applications to run in a separate virtual memory space, which ensures safe test execution and improves reliability.

The UserCode run-time module uses a superset of the G-API commands for Windows® ensuring an easy migration of existing program source code. Additional functions will provide access to event notifications, timer tasks, the FLASH file system and other RT OS resources as well as standard C libraries.

The PowerPC processor uses big-endian byte order which must be taken care of when writing or porting code for the UserCode run-time module. For smooth migration from little to big-endian, a library of conversion macros is provided with the Net2Run IDE development system.

With the Net2Run IDE development system, GOEPEL electronic provides a complete tool chain for creating UserCode programs and for their direct execution on PXI/ PCI 61xx boards.

The Net2Run IDE development system is based on Eclipse IDE and contains the QNX Neutrino Command Line Tools (CLT), including PowerPC-Compiler, Linker and Debugger.

UserCode programs can be downloaded and debugged direct from Net2Run IDE via an Ethernet connection.

The figure below shows the Net2RunIDE development system:

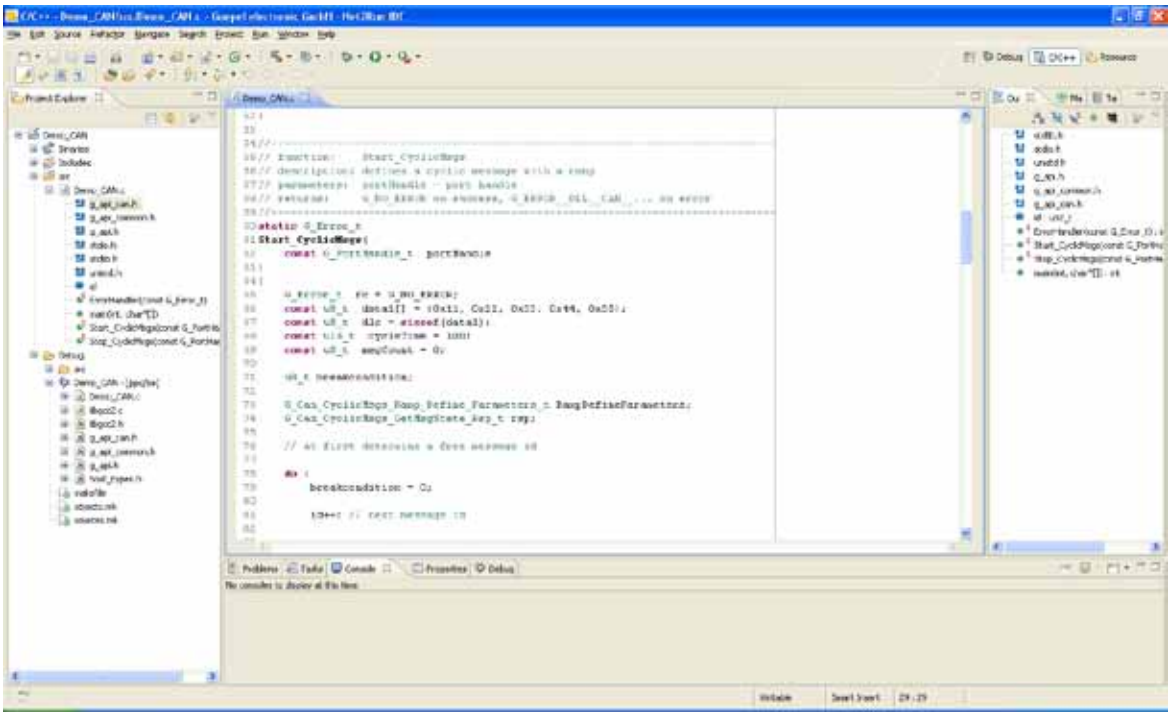

Figure 4-1: Net2Run IDE Window

Please consult the G-API documentation for further information. This documentation and the installation software are located in the *G-API* folder of the supplied “Product Information” CD.

4.3 Programming via DLL Functions

For the used structures, data types and error codes refer to the headers – you find the corresponding files on the supplied CD.

4.3.1 Windows Device Driver

The DLL functions for programming using the Windows® device driver are described in the following chapters:

- System Info
- Transceiver Info
- Write Instruction
- Read Response
- Read Response Block

4.3.1.1 System Info

The Pxi61xx__SystemInfo function is used for the status query of the hardware driver and for query of the board properties.

Format:

```
S32 Pxi61xx__SystemInfo(t_System_Info *pSystemInfo,
                        U32  LengthInByte)
```

Parameters:

Pointer, for example pSystemInfo,
to a data structure
(For the structure, see the *Pxi61xx.h* file on the supplied CD)

LengthInByte
Size of the buffer pSystemInfo is pointing to, in bytes

Description:

The Pxi61xx__SystemInfo function returns information regarding the status of the hardware driver.

For this, the address of a pSystemInfo pointer has to be transferred to the function.
Within the function, the structure pSystemInfo is pointing to is filled with the corresponding pieces of information.

4.3.1.2 Transceiver Info

The Pxi61xx__TransceiverInfo returns information regarding the plugged-in transceivers as well as their number.

Format:

```
S32 Pxi61xx__TransceiverInfo(t_Transceiver_Properties *pTransceiverProperties,
                                                      U32  LengthInByte)
```

Parameters:

Pointer, for example pTransceiverProperties,
to a data structure
(For the structure, see the *Pxi61xx.h* file on the supplied CD)

LengthInByte
Size of the buffer pTransceiverProperties is pointing to, in bytes

Description:

The Pxi61xx__TransceiverInfo function returns information regarding the transceiver properties.
For this, the address of a pTransceiverProperties pointer has to be transferred to the function.
Within the function, the structure pTransceiverProperties is pointing to is filled with the corresponding pieces of information.

4.3.1.3 Write Instruction

The Pxi61xx__WriteInstruction function is for sending a command to the PXI/ PCI 61xx controller.

Format:

```
S32 Pxi61xx__WriteInstruction(U8 *pData,
                              U16  DataLength)
```

Parameters:

Pointer, for example pData,
to the writing data area,
consisting of Command header and Command bytes
(At present max. 1024 bytes per command)

DataLength
Size of the writing area pData is pointing to, in bytes

Description:

The Pxi61xx__WriteInstruction function sends a command to the PXI/ PCI 61xx controller.

In the header of the structure pData is pointing to, there is the information regarding the PXI/ PCI 61xx board to be activated by this function.
Therefore this parameter is not to be given separately.

4.3.1.4 Read Response

The Pxi61xx__ReadResponse function is for reading a response from the PXI/ PCI 61xx controller.

Format:

```
S32 Pxi61xx__ReadResponse(U8  Device,
                          U8 *pData,
                         U32 *DataLength)
```

Parameters:

Device

Index of the PXI/ PCI 61xx board, beginning left with 1

Pointer, for example pData,
to the reading data area,

consisting of Response header and Response bytes
(At present max. 1024 bytes per response)

DataLength

Parameter value before function call:
Size of the buffer pData is pointing to, in bytes

Parameter value after function call:
Number of bytes actually read

Description:

The Pxi61xx__ReadResponse function reads back the oldest response written by the PXI/ PCI 61xx controller in the response area.

If several responses have been provided by the controller, but not read, they are not lost but stored in the form of a list.

On calling up, the Pxi61xx__ReadResponse function continues to supply data until this list contains no more entries.

4.3.1.5 Read Response Block

The Pxi61xx__ReadResponseBlock function is for reading all available responses from the PXI/ PCI 61xx controller.

Format:

```
S32 Pxi61xx__ReadResponseBlock(U8  Device,
                               U8 *pData,
                              U32 *DataLength,
                              U32 *BlockCounter)
```

Parameters:

Device

Index of the PXI/ PCI 61xx board, beginning left with 1

Pointer, for example pData,
to the reading data area,
consisting of Response header and Response bytes
(At present max. 1024 bytes per response)

DataLength

Parameter value before function call:
Size of the buffer pData is pointing to, in bytes

Parameter value after function call:
Number of bytes actually read

Pointer, for example BlockCounter

Number of the contained individual responses

Description:

The Pxi61xx__ReadResponseBlock function reads back all responses written by the PXI/ PCI 61xx controller in the response area.

4.3.2 VISA Device Driver

The DLL functions for programming using the VISA device driver are described in the following sections:

- Init
- Done
- System Info
- Transceiver Info
- Write Instruction
- Read Response

4.3.2.1 Init

The PXI61xx_Init function is for opening VISA sessions for the system's PXI/ PCI 61xx boards including initialization.

Format:

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

Parameter:

CardCount

Number of the system's PXI/ PCI 61xx boards recognized by the VISA driver.

Description:

The PXI61xx_Init function searches for all PXI/ PCI 61xx boards of the system and opens the required sessions.
Additionally, board internal initializations are carried out.

Therefore this function must be executed as the first step.

4.3.2.2 Done

The PXI61xx_Done function closes all VISA sessions of the system's PXI/ PCI 61xx boards.

Format:

```
ViStatus PXI61xx_Done(void)
```

Parameter:

none

Description:

The PXI61xx_Done function closes all VISA sessions of the system's PXI/ PCI 61xx boards.

No further access to the boards is possible, then.

4.3.2.3 System Info

The PXI61xx_SystemInfo function provides general driver and board information.

Format:

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

Parameters:

Pointer, for example DriverData,
to a data structure

(For the structure, see the *PXI61xx_API.h* file on the supplied CD)

LengthInByte

Size of the buffer DriverData is pointing to, in bytes

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

Description:

The PXI61xx_SystemInfo function provides information regarding the driver and the system's PXI/ PCI 61xx boards.

The DeviceName indicates the resource names registered by VISA. This information correlates with the display of NI MAX.

4.3.2.4 Transceiver Info

The PXI61xx_TransceiverInfo function provides information regarding the plugged-in transceivers as well as their number.

Format:

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

Parameters:

Pointer, for example TransceiverProperties,
to a data structure

(For the structure, see the *PXI61xx_API.h* file on the supplied CD)

LengthInByte

Size of the buffer TransceiverProperties is pointing to, in bytes

Description:

The PXI61xx_TransceiverInfo function provides information regarding the transceiver properties.

For this, the address of a TransceiverProperties pointer has to be transferred to the function.
Within the function, the structure TransceiverProperties is pointing to is filled with the corresponding pieces of information.

4.3.2.5 Write Instruction

The PXI61xx_WriteInstruction function is for writing data to the PXI/ PCI 61xx controller.

Format:

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

Parameters:

Pointer, for example pData,
to the writing data area,
consisting of Command Header and Command Bytes
(currently max. 1024 bytes per command)

DataLength

Size if the storage area pData is pointing to, in bytes

Description:

The PXI61xx_WriteInstruction function allows writing of data to the PXI/ PCI 61xx controller.

In the header of the structure pData is pointing to, there is the information regarding the PXI/ PCI 61xx board to be activated by this function.
Therefore this parameter is not to be given separately.

4.3.2.6 Read Response

The PXI61xx_ReadResponse function is for reading data from the PXI/ PCI 61xx-Controller.

Format:

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

Parameters:

Device

Index of the PXI/ PCI 61xx board, beginning left with 1

Pointer, for example pData,
to the reading data area,
consisting of Response Header and Response Bytes
(currently max. 1024 bytes per response)

DataLength

Value of the parameter before function call:
Size of the buffer pointed by pData, in bytes

Value of the parameter after function call:
Number of bytes actually read

Description:

The PXI61xx_ReadResponse function allows reading of data provided by the PXI/ PCI 61xx controller (see also the ReadResponse function in the Windows Device Driver section).

4.4 Programming with LabVIEW

4.4.1 LabVIEW via the G-API

The supplied CD contains VIs for activating PXI/ PCI 61xx boards under LabVIEW.

These LabVIEW VIs use the functions of the GOEPEL G-API.

4.4.2 LLB using the Windows Device Driver

The supplied CD contains VIs for activating PXI/ PCI 61xx boards under LabVIEW.

The functions described in the Windows Device Driver section are used for this.

4.4.3 LLB using the VISA Device Driver

The supplied CD contains VIs for activating PXI/ PCI 61xx boards under LabVIEW.

The functions described in the VISA Device Driver section are used for this.

4.5 Further GOEPEL Software

PROGRESS, Program Generator and myCAR of GOEPEL electronic GmbH are comfortable software programs for testing with GOEPEL hardware.

Please refer to the corresponding User Manual to get more information regarding these programs.

T

U

V

W