Technical Specification

PXI / PCI 3080

CAN/ LIN/ K-LINE/ J1850 Interfaces

User Manual

Version 1.5

GOPEL electronic

Get the total Coverage!

GOEPEL electronic GmbH
Goeschwitzer Str. 58/60
D-07745 Jena
Phone: +49-3641-6896-597
Fax: +49-3641-6896-944
E-Mail: ats_support@goepel.com
http://www.goepel.com

Printed: 23.04.2012

Issue: April 2012

1 Board Installation

1.1 Hardware Installation

Please make absolutely certain that all of the installation procedures described below are carried out with your system 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 PXI/ PCI 3080 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 screws if necessary and remove the cover from the slot.

(If it is necessary to exchange transceiver modules, pay attention to the general rules to avoid electrostatic charging, 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 screws 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.

1.2 Driver Installation

1.2.1 Windows Device Driver

PXI/ PCI 3080 controller boards can be operated under Windows® 2000/ XP as well as under Windows®7/ 64 bit.

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 in the *GPxi3080* folder of the supplied CD.

If necessary, you can find the required *inf* files as follows:

- *GPxi3080.inf* for Windows® 2000/ XP in the *Win2000 (Version xx)* folder
- *GPxi3080_x64.inf* for Windows®7/ 64 Bit in the *Win7_x64 (Version xx)* folder

It is not absolutely essential to restart the system.

The following step is only required in case you do not use the G-API (see also Control Software).

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.

1.2.2 VISA Device Driver

First step

Copy the *VISA (Version xx)* directory from the *GPxi3080* folder of the delivered CD to your hard disk
(recommendation: Complete directory to *C:* \).

Second step

VISA for Windows2000, WindowsXP :

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. Enter as target directory the one which contains the *PXI3080_NT5.inf* file (according to recommendation: *C:* *VISA (Version xx)* *Installation*).

VISA for LabViewRT :

For operating PXI/ PCI 3080 boards under the RT operating system, use the *P3080_RT.inf* file from the *C:* *VISA (Version xx)* *Installation* directory.

Copy this file to the *ni-rt* *system* folder of the embedded controller. Use the NI Measurement & Automation Explorer for that.

The connected RT controller can be found under Remote Systems. Open a pop-up menu by pressing the right mouse button. In this menu, select the File Transfer entry and follow the instructions.

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

Third step:

Reboot your computer to complete installation.

After hardware and driver installation, you can check whether the boards are properly imbedded by the system:

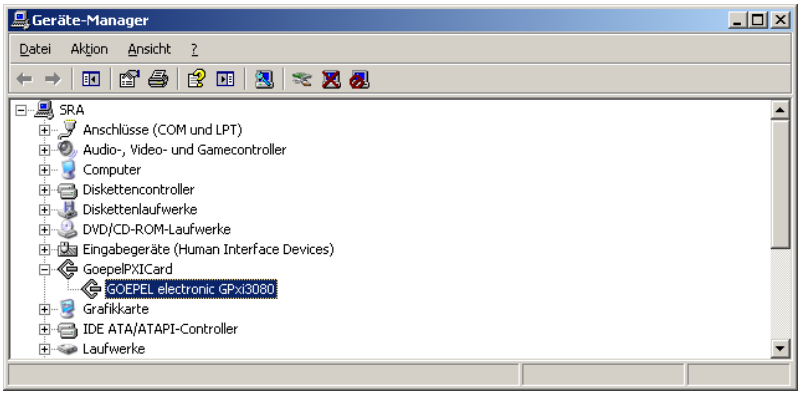

Figure 1-1: Windows

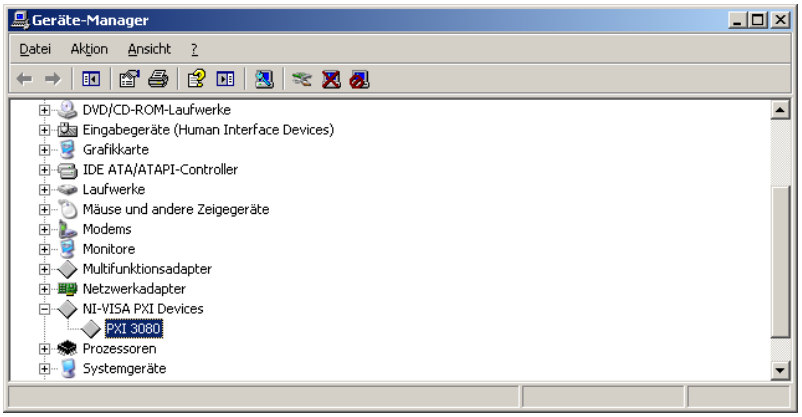

Figure 1-2: VISA for Windows XP

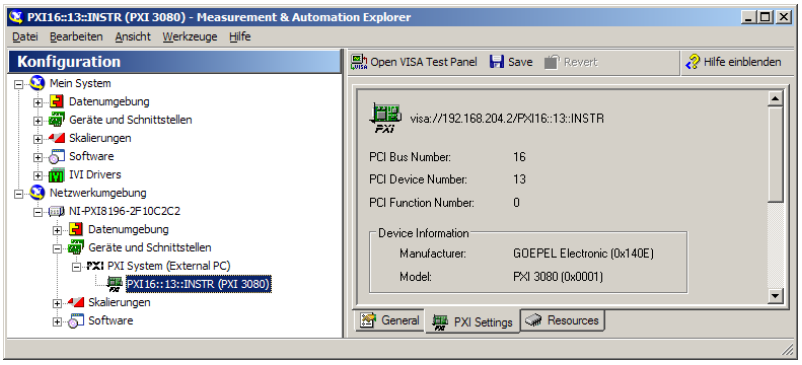

Figure 1-3: ViISA for LabVIEW RT

2 PXI / PCI 3080 Hardware

2.1 Definition

The PXI 3080/ PCI 3080 multi-interface-boards are communication boards of GOEPEL electronic GmbH.

These boards can be used in general control technology, especially for applications in automotive technology.

A PXI 3080/ PCI 3080 board in the maximum construction stage offers the following resources:

- 2 x CAN
- 2 x LIN or K-Line
- 1 x J1850 VPW
- 1 x J1850 PWM
  (in this case only ONE LIN or K-Line interface is possible)
- 4 x Digital Input
- 4 x Digital Output
- 2 x Analog Input
- 1 x Wake line

The resources are galvanically separated from the PXI/ PCI interface.

Figure 2-1: PXI 3080

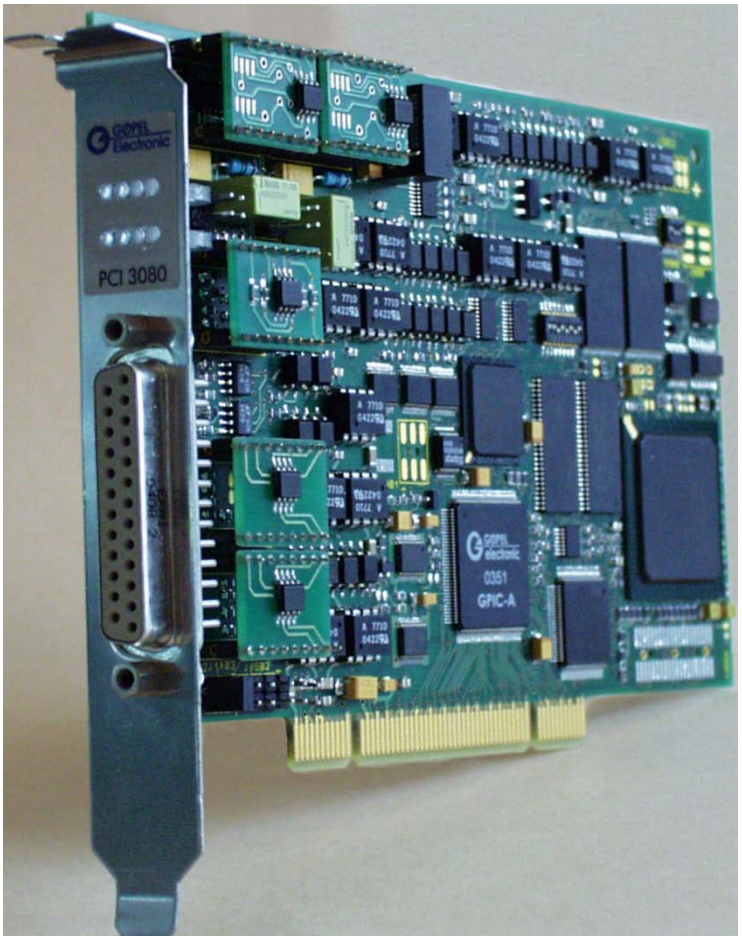

Abbildung 2-2: PCI 3080

2.2 Technical Data

2.2.1 General

The PXI 3080 communication board is a plug-in board developed for the PXI™ bus (**P**CI e**X**tensions for **I**nstrumentation). Basis of this bus is the CompactPCI™ bus.

The board can be plugged into any desired slot of a CompactPCI™ or PXI™ system (except for slot 1). It can be definitely identified also in the case that several boards of this type are used in the same rack.

The PCI 3080 communication board is a PC plug-in board for the PCI Local Bus Rev. 2.2.
It can be operated at any PCI slot (32 bits, 33 MHz, 3.3 V)

Both boards do not have jumpers for hardware detection and are automatically integrated into the respective system.

2.2.2 Dimensions

The dimensions of both boards correspond to standard dimensions of the accompanying bus system:

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

2.2.3 PXI 3080/ PCI 3080 Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Remarks
V_{BAT}	Battery voltage		12	27/ 50	V	Acc. to transceiver's type
	Transmission rate			1	MBaud	CAN
	Transmission rate			22	kBaud	LIN
R_{bus}	Terminating resistor 1		120		Ohms	CAN jumper plugged in
R_{bus}	Terminating resistors 2		10		kOhms	CAN jumper plugged in
R_{Pullup}	Pull-up resistor		680		Ohms	K-Line jumper plugged in
V_{in}	Input voltage	3.3		50	V	Digital input
V_{out}	Output voltage			V_{BAT}	V	Digital output, OC
V_{in}	Input voltage			26	V	Analog input
V_{iso}	Galvanic separation	1000			V	PXI/ PCI Input/ Output

The Analog input channels are designed with the LTC 1400 (analog-to-digital transducer) of Linear Technology.
This component has a Resolution of 12 Bits and an Input voltage range of 0..4.095V.

Caused by the input voltage divider (122K/22K) the following results for the measured voltage:
V_{meas} = AD transducer value * 1mV * (122K/22K).

2.3 Construction

2.3.1 General

An ASIC is used as the interface to the PCI or cPCI bus on the PXI/ PCI 3080 boards. It includes all the function blocks required for the communication with the computer bus.

The PCI 3080 communication board does not have a PXI interface. To exchange trigger signals with other GOEPEL electronic PCI boards despite of that, an additional plug connector is on this board with two lines configurable as input or output (J902 in Figure 2-7).

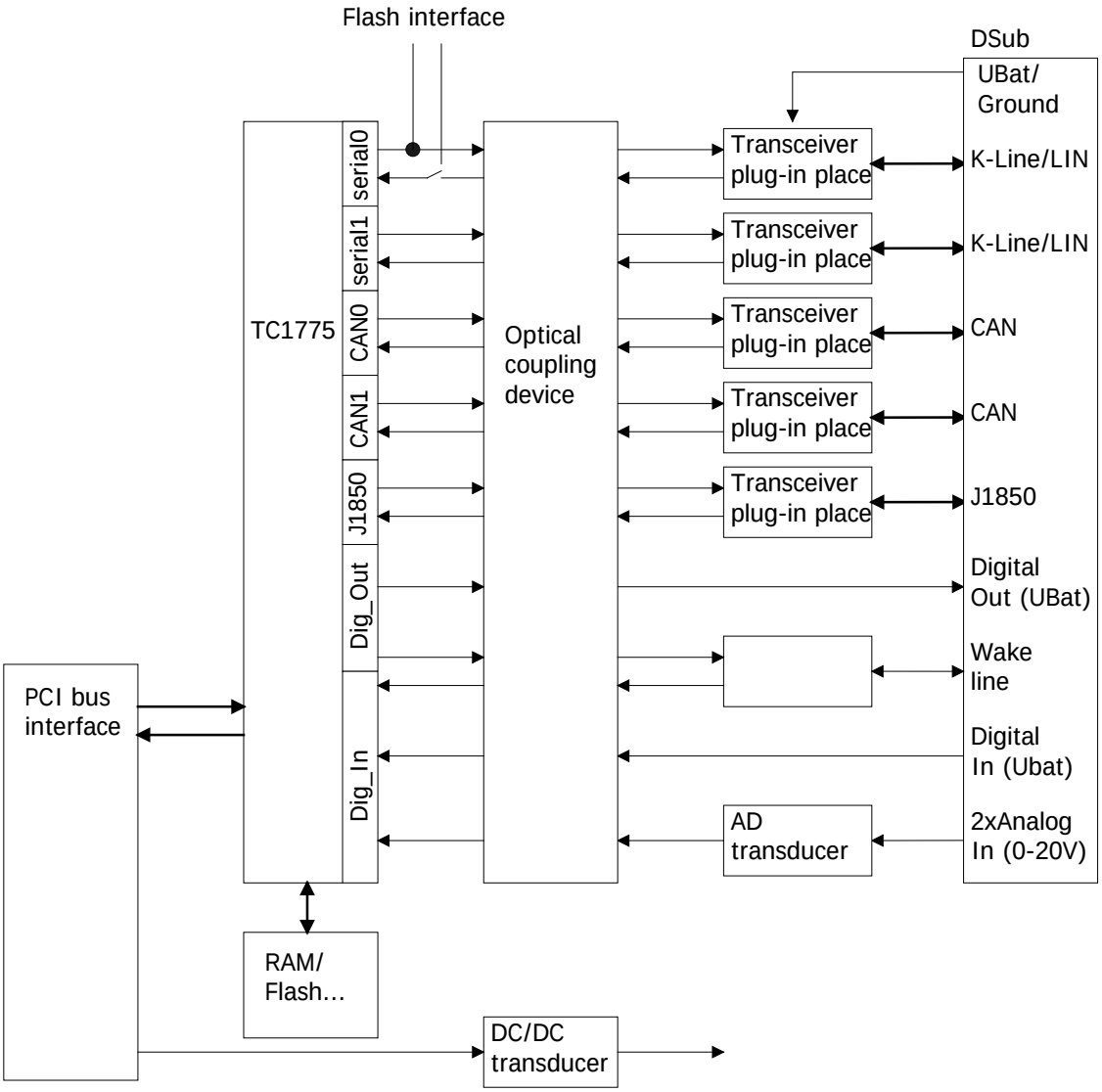

Figure 2-3: Block diagram of a PXI/ PCI 3080 Communication Board

2.3.2 Addressing

PXI 3080: 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.

The PXI 3080 board can read out this geographical slot address. For that the belonging FPGA file has to be loaded to the XILINX (see the XilinxDownload functions for different drivers in the Control Software section).

PCI 3080: PCI racks do not have an own geographical slot addressing. There is a separate DIP switch (S900 in Figure 2-7) for clear identification of the board in a system with several PCI 3080 boards. You can select up to 16 addressing variants by this. The corresponding binary value (0 to 15) set with the S900 switch can be read out by the delivered software.

2.3.3 Communication Interfaces

2 x CAN Interface Version 2.0b:

The type of the mounted transceiver is decisive for proper operation of a CAN interface in a network. Often CAN networks do only operate properly in the case that all members use a compatible type of transceiver.

To offer maximal flexibility to the users of the PXI/ PCI 3080 boards, the transceivers are designed as plug-in modules. There are several types (high speed, low speed, single-wire etc.) that can be easily exchanged.

Not only the type of the mounted transceiver, but also the terminating resistor of the bus is very important for proper operation of a CAN network.
For the use of highspeed CAN transceivers, usually one 120 Ohm resistor is active on each CAN interface.These resistors can be deactivated by removing the J1401 or J1501 jumpers.

In the case of lowspeed CAN transceivers, usually two resistors with a resistance value of 10 kOhm for RTH and RTL are active for each CAN interface (by inserting the J1402/ J1403 or J1502/ J1503 jumpers). Then the J1401 or J1501 jumpers must NOT be plugged-in.

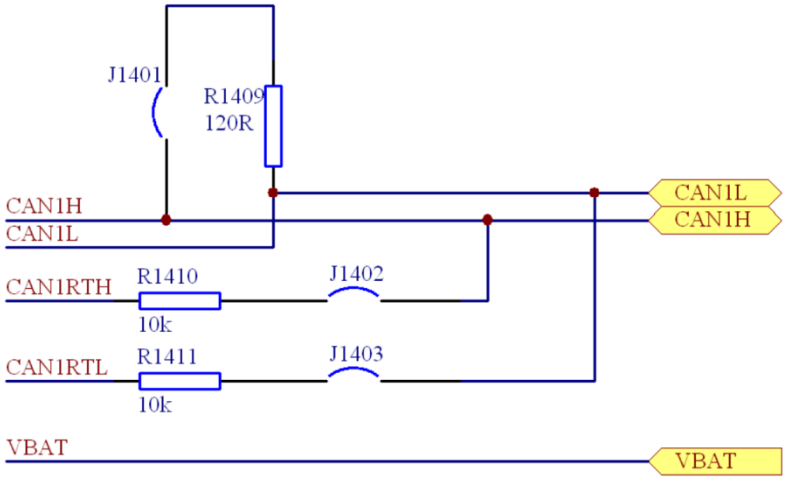

Figure 2-4: CAN Interface

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

K-Line:

The transceivers are designed as plug-in modules. Generally, the L9637 of ST is used for this type of transceicer.

Via the V_{Bat} contacts the power supply of the transceiver modules is connected. To bridge the reverse-connect protection diode for V_{Bat} for LIN, the J1602 or J1703 jumpers must be plugged-in.

In the case the pull-up resistor to V_{Bat} is to be activated, the J1601 or J1701 jumpers must be plugged-in.

LIN:

The transceivers are designed as plug-in modules. Generally, the TJA1020 of Philips is used for this type of transceicer.

For the standard design of the transceiver modules, it is possible to change over between Master and Slave configuration per software using the Rel1 relay for LIN1 and Rel2 for LIN2. The pull-up resistors for LIN are located on the transceiver module. Therefore the J1601 or J1701 jumpers must NOT be plugged-in.

Via the V_{Bat} contacts the power supply of the transceiver modules is connected. According to the LIN specification, this power supply is to be carried out via a reverse-connect protection diode. Therefore the J1602 or J1703 must NOT be plugged-in.

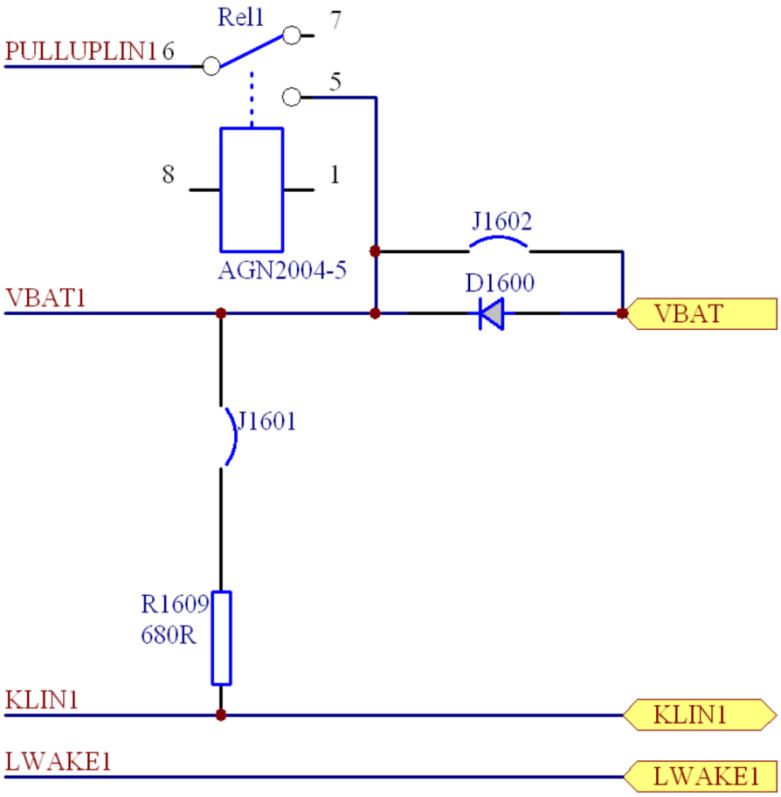

Figure 2-5: LIN/ K-Line Interface

J1850 Interfaces:

The transceivers are designed as plug-in modules.

Generally, the AU5780 of Philips is used for J1850 VPW transceivers.

The output circuitry of a J1850 PWM transceiver is realized by discrete components.

The transceiver for a J1850 VPW interface has to be inserted at the position for J1850 transceivers.
On the other hand, the transceiver for a J1850 PWM interface must be inserted at the position for a K-Line/ LIN 2 transceiver (see Figure 2-6 and Figure 2-7).

J1701 must NOT be mounted in the case of a J1850 PWM interface!

2.3.4 Assembly

Figure 2-6 and Figure 2-7 show schematically the component side of the boards. You can see the positions of the transceiver modules, plug connectors, DIP switches and jumpers.

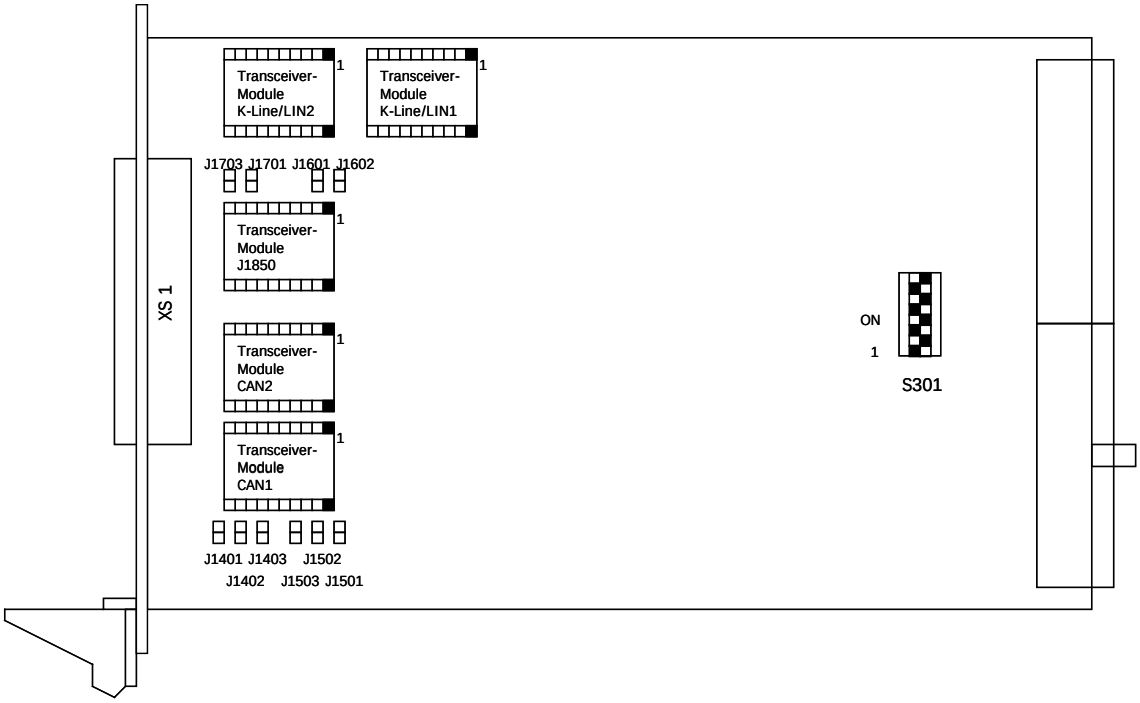

Figure 2-6: Component side of a PXI 3080 communication board (schematically)

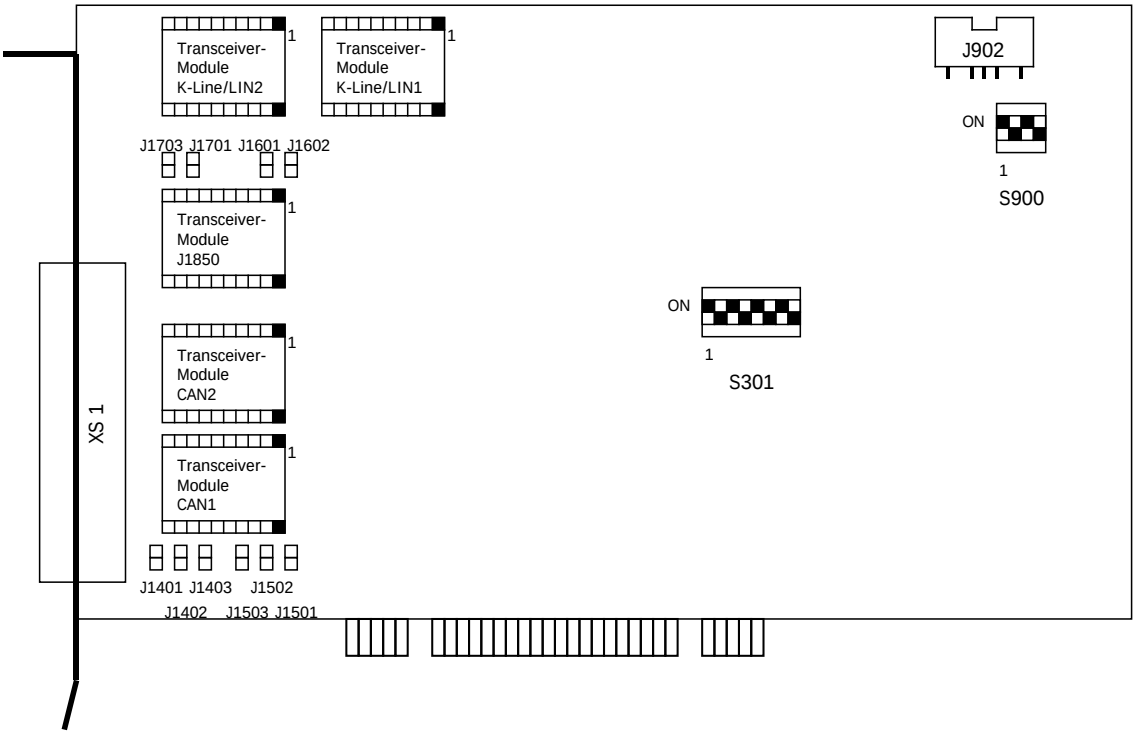

Figure 2-7: Component side of a PCI 3080 communication board (schematically)

The transceiver for a J1850 VPW interface has to be inserted at the position for J1850 transceivers.
On the other hand, the transceiver for a J1850 PWM interface must be inserted at the position for a K-Line/ LIN 2 transceiver.

The configuration elements of Figure 2-6 and Figure 2-7 are explained in the following table:

XS1401	Transceiver module for CAN1
J1401	Jumper to activate the **120Ω** terminating resistor for CAN1
J1402	Jumper to activate the RTH **10kΩ** terminating resistor for CAN1
J1403	Jumper to activate the RTL **10kΩ** terminating resistor for CAN1
XS1501	Transceiver module for CAN2
J1501	Jumper to activate the **120Ω** terminating resistor for CAN2
J1502	Jumper to activate the RTH **10kΩ** terminating resistor for CAN2
J1503	Jumper to activate the RTL **10kΩ** terminating resistor for CAN2
XS1601	Transceiver module for LIN1/ K-Line1
J1601	Jumper to activate the **680Ω** pull-up resistor to V_{BAT} for K-Line1
J1602	Jumper to bridge the reverse-connect protection diode for V_{Bat} for LIN1
XS1701	Transceiver module for LIN2/ K-Line2/ J1850 PWM
J1701	Jumper to activate the **680Ω** pull-up resistor to V_{BAT} for K-Line2
J1703	Jumper to bridge the reverse-connect protection diode for V_{Bat} for LIN2
XS1801	Transceiver module for J1850 VPW
S301	DIP switches of the PXI/ PXI 3080 boards to configurate the micro controller. **Do NOT change the settings!**
S900	This DIP switch on a PCI 3080 board is for clear identification of the board (analogously to “geographical addressing” of the PXI specification) in a system with several PCI 3080 boards. You can select up to 16 addressing variants this way (0..15). The corresponding binary value set with the S900 switch can be read out by the delivered software.
J902	Plug connector to exchange trigger signals with other GOEPEL electronic PCI boards

2.3.5 Frontal Plug Connector Pinout

Type: DSub 25 poles socket

The interfaces are provided via this plug connector at the frontal edge of the PXI/ PCI 3080 communication boards.

The pinout of both boards is identical according to the following table:

No.	XS1 pin	Signals name	Remarks
1	1	CAN1_High	
2	14	CAN1_Low	
3	2	CAN2_High	
4	15	CAN2_Low	
5	3	VBAT	Reference potential plus transceiver
6	16	GND	Ground potential transceiver
7	4	K-Line1/ LIN1	depending on transceiver
8	17	L-Line1/ WAKE1	depending on transceiver
9	5	K-Line2/ LIN2/ J1850 PWM+	depending on transceiver
10	18	L-Line2/ WAKE2/ J1850 PWM-	depending on transceiver
11	6	J1850 VPW	
12	19	-	Please do not assign
13	7	VBAT	Reference potential plus transceiver
14	20	GND	Ground potential transceiver
15	8	Analog Input1	
16	21	Analog Input2	
17	9	Digital Input1	
18	22	Digital Output1	
19	10	Digital Input2	
20	23	Digital Output2	
21	11	Digital Input 3	
22	24	Digital Output3	
23	12	Digital Input 4	
24	25	Digital Output4	
25	13	Wake line	

For K-Line the connections for the L-Line are wired to PIN 17/ 18 if necessary (depending on the output circuitry).

For LIN the connections for the Wake-Line are wired to PIN 17/ 18 if necessary (depending on the selection of the transceiver).

The pins 3 and 7 as well as 16 and 20 are bridged on the board!

2.3.6 LED Display

The LEDs indicate the following states:

- Red LED D100: /HDRST hardware reset indication output of the micro controller
- Green LED D700: Voltage 5V status
- Green LED D701: Voltage 3.3V status
- Green LED D702: Voltage 2.5V status

- Yellow LED D801: Micro controller CAN 1 status
- Yellow LED D802: Micro controller CAN 2 status
- Yellow LED D803: Micro controller K-Line/ LIN 1 status
- Yellow LED D804: Micro controller K-Line/ LIN 2 status

The LEDs are arranged as follows at the front panel:

D702 D701 D700 D100

D801 D802 D803 D804

2.4 Delivery Notes

PXI/ PCI 3080 boards are delivered in the following basic variants:

- 1x CAN interface and 1x LIN interface or
- 1x CAN interface and 1x K-Line interface

These basic variants can be extended by the following options:

- 1x Additional CAN interface
- 1x Additional LIN interface or K-Line interface
- 1x Additional J1850 VPW interface
- 1x Additional J1850 PWM interface

If you select the 1x Additional J1850 PWM interface option, the 1x Additional LIN interface or K-Line interface option is NOT possible.

In addition to selecting an interface, the type of the corresponding CAN/ LIN/ K-Line/ J1850 transceiver as well as the required Functionalities for each CAN/ LIN/ K-Line/ J1850 interface must be selected.

3 Control Software

There are three ways to integrate PXI 3080/ PCI 3080 hardware in your own applications:

- Programming via G-API
- Programming via DLL Functions
- Programming with LabVIEW

3.1 Programming via G-API

The G-API (GOEPEL-API) is a programming environment based on “C” for GOEPEL electronic hardware under Windows®. So the G-API is the preferred programming environment for this hardware.
It provides a wide, hardware independent command set for CAN, LIN, K-Line, 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 Hardware Explorer 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 Hardware Explorer.
Furthermore, the Hardware Explorer provides a simple means of testing the interaction between hardware and software by executing the integrated self-tests.

The figure below shows the GOEPEL electronic Hardware Explorer:

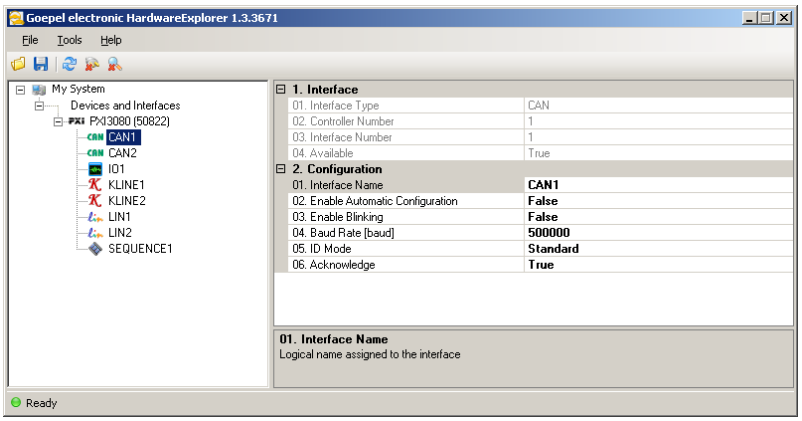

Figure 3-1 Hardware Explorer

Please consult the G-API documentation for further information. This documentation and the installation software are located in the *G-API* folder on the PXI product CD.

3.2 Programming via DLL Functions

Programming via DLL Functions is possible also in future for existing projects which can not be processed with the GOEPEL electronic programming interface G-API.

We would be pleased to send the GOEPEL Firmware documentation to you on your request. Please get in touch with our sales department in case you need that.

The GPxi3080 and PXI3080 expressions used in the following function description stand for PXI 3080/ PCI 3080.

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

3.2.1 Windows Device Driver

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

- Driver Info
- DLL Version
- XILINX Download
- XILINX Write Data
- DPRAM Write Instruction
- DPRAM Read Response
- Reset Port

3.2.1.1 Driver Info

The GPxi3080_GetDriverInfo function is for the status query of the hardware driver.

Format:

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

Parameter:

Pointer, for example pDriverInfo , to a data structure
For the structure, see the *GPxi3080.h* file on the supplied CD

Description:

The GPxi3080_GetDriverInfo function returns information regarding the status of the hardware driver.

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

3.2.1.2 DLL Version

The GPxi3080_DLL_Version function is for the version number query of the DLL.

Format:

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

Parameter

Pointer, for example pVersion, to the Version number

Description:

The GPxi3080_DLL_Version function returns the version number of the *GPxi3080w.dll* as an integer value.

Example:

Version number **1.23** is returned as **123**,
and version number **1.60** as **160**.

3.2.1.3 XILINX Download

The GPxi3080_XilinxDownload function is to load an FPGA file to the XILINX.

Format:

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

Parameters:

Device
Index of the PXI/ PCI 3080 board, beginning left with 1

Pointer, for example pFileName,
to the Path of the FPGA file to be loaded

Description:

The GPxi3080_XilinxDownload function allows it to load an FPGA file (**.cfd* extension) to the XILINX. This file serves, among other possibilities, to read the geographical slot address in the PXI Rack.

The loaded data is volatile. Therefore the function has to be executed again after switching off power.

After XilinxDownload, a delay of about 500 ms is required (as the controller executes a power-on reset).

Then, carry out the 0x10 Software Reset firmware command to come in the normal operation mode from the bootloader mode.

3.2.1.4 XILINX Write Data

The GPxi3080_XilinxWriteData function allows the configuration and execution of functions provided by the XILINX.

Format:

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

Parameters:

Pointer, for example data , to the Write data area
(currently max. 128 bytes per command)

length
Size of the storage area data is pointing to, in bytes

Description:

Before using the functionality of the XILINX, the belonging FPGA file must have been loaded by GPxi3080_XilinxDownload
(see XILINX Download).

The data format consists of four bytes including the command.
If necessary parameter bytes can follow.

Data format:
- 1st byte: 0x48 (StartByte)
- 2nd byte: Device (index of the PXI/ PCI 3080 board, beginning left with 1)
- 3rd byte: 0x00 (Reserved byte)
- 4th byte: XILINX command

Currently supported XILINX command:

0x10 PowerOnReset for the complete board

3.2.1.5 DPRAM Write Instruction

The GPxi3080_DpramWriteInstruction is for sending a command to the PXI/ PCI 3080 controller.

Format:

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

Parameters:

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

length
Size of the storage area data is pointing to, in bytes

Description:

The GPxi3080_DpramWriteInstruction function sends a command to the PXI/ PCI 3080 controller.

In the header of the structure data is pointing to, there is the information regarding the PXI/ PCI 3080 board to be activated by this function.
Therefore this parameter has not to be given separately.

3.2.1.6 DPRAM Read Response

The GPxi3080_DpramReadResponse function is for reading a response from the PXI/ PCI 3080 controller.

Format:

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

Parameters:

card

Index of the PXI/ PCI 3080 board, beginning left with 1

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

length

Value of the parameter before function call:

Size of the buffer pointed by data in bytes

Value of the parameter after function call:

Number of bytes actually read

Description:

The GPxi3080_DpramReadResponse function reads back the oldest response written by the PXI/ PCI 3080 controller in the Response area of the DPRAM.

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

On calling up, the GPxi3080_DpramReadResponse function continues to supply data until this list contains no more entries.

3.2.1.7 Reset Port

The GPxi3080_ResetPort function is for releasing a software reset for the PXI/ PCI 3080 controller.

Format

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

Parameter:

card

Index of the PXI/ PCI 3080 board, beginning left with 1

Description:

The GPxi3080_ResetPort function releases a software reset for the PXI/ PCI 3080 controller.

This releasing procedure is executed via a separate interrupt channel, NOT via the command interpreter of the software (0x10 Software Reset firmware command).

3.2.2 VISA Device Driver

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

- Init
- Done
- Driver Info
- XILINX Download
- XILINX Write Data
- Write Data
- Read Data
- Reset Port

3.2.2.1 Init

The PXI3080_Init function is for opening VISA sessions for the system's PXI/ PCI 3080 boards including initialization.

Format:

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

Parameter:

CardCount

Number of the system's PXI/ PCI 3080 boards recognized by the VISA driver.

Description:

The PXI3080_Init function searches for all PXI/ PCI 3080 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.

3.2.2.2 Done

The PXI3080_Done function closes all VISA sessions of the system's PXI/ PCI 3080 boards.

Format:

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

Parameters:

none

Description:

The PXI3080_Done function closes all VISA sessions of the system's PXI/ PCI 3080 boards.

No further access to the boards is possible then.

3.2.2.3 Driver Info

The PXI3080_DriverInfo function provides general information regarding driver and board.

Format:

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

Parameters:

Pointer, for example DriverData, to a data structure

For the structure see the *PXI3080_API.h* file of the supplied CD

DeviceName
`Array[K_DEV_MAX][K_RES_NAME_LENGTH]`
(see *PXI3080_API.h*)

Description:

The PXI3080_DriverInfo function provides several pieces of information regarding the driver and the system's PXI/ PCI 3080 boards.

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

3.2.2.4 XILINX Download

The PXI3080_XilinxDownload function is to load an FPGA file to the XILINX.

Format:

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

Parameters:

Card

Index of the PXI/ PCI 3080 boards, beginning left with 1

Pointer, for example FileName,

to the Path of the FPGA file to be loaded

Description:

The PXI3080_XilinxDownload function allows it to load an FPGA file (**.cfd* extension) to the XILINX. This file serves, among other possibilities, to read the geographical slot address in the PXI rack.

The loaded data is volatile. Therefore the function has to be executed again after switching off power.

After XilinxDownload, a delay of about 500 ms is required (as the controller executes a power-on reset).

Then, carry out the 0x10 Software Reset firmware command to come in the normal operation mode from the bootloader mode.

3.2.2.5 XILINX Write Data

The PXI3080_XilinxWriteData function allows the configuration and execution of functions provided by the XILINX.

Format:

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

Parameter:

Pointer, for example Data, to theWrite data area
(currently max. 128 bytes per command)

Description:

Before using the functionality of the XILINX, the corresponding FPGA file must have been loaded by PXI3080_XilinxDownload (see XILINX Download).

The data format consists of four bytes including the command.
If necessary parameter bytes can follow.

Data format:
- 1st byte: 0x48 (StartByte)
- 2nd byte: card (index of the PXI/ PCI 3080 board, beginning left with 1)
- 3rd byte: 0x00 (Reserved byte)
- 4th byte: XILINX command

Currently supported XILINX command:

0x10 PowerOnReset for the complete board

3.2.2.6 Write Data

The PXI3080_WriteData function is for writing data to the PXI/ PCI 3080 controller.

Format:

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

Parameters:

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

Length_In_Bytes
Size of the storage area WriteData is pointing to, in bytes

Description:

The PXI3080_WriteData function allows writing of data to the PXI/ PCI 3080 boards.

In the header of the structure WriteData is pointing to, there is the information regarding the board to be activated.

Therefore this parameter is not to be given separately.

3.2.2.7 Read Data

The PXI3080_ReadData function is for reading data from the PXI/ PCI 3080 Controller.

Format:

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

Parameters:

Card

Index of the PXI/ PCI 3080 board, lbeginning eft with 1

Pointer, for example ReadData, to the Read data area,

consisting of Response Header and Response Bytes
(currently max. 1024 bytes per response)

Length

Value of the parameter before function call:

Size of the buffer pointed by ReadData in bytes

Value of the parameter after function call:

Number of bytes actually read

Description:

The PXI3080_ReadData function allows reading of data provided by a PXI/ PCI 3080 board
(see also GPxi3080_DpramReadResponse function
in the Windows Device Driver section).

3.2.2.8 Reset Port

The PXI3080_ResetPort function is for releasing a software reset for the PXI/ PCI 3080 controller.

Format

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

Parameter:

Card

Index of the PXI/ PCI 3080 board, beginning left with 1

Description:

The GPxi3080_ResetPort function releases a software reset for the PXI/ PCI 3080 controller.

This releasing procedure is executed via a separate interrupt channel, NOT via the command interpreter of the software (0x10 Software Reset firmware command).

3.3 Programming with LabVIEW

3.3.1 LabVIEW via the G-API

The supplied CD contains VIs for activating PXI/ PCI 3080 boards under LabVIEW.

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

3.3.2 LLB using the Windows Device Driver

The supplied CD contains VIs for activating PXI/ PCI 3080 boards under LabVIEW.

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

3.3.3 LLB using the VISA Device Driver

The supplied CD contains VIs for activating PXI/ PCI 3080 boards under LabVIEW.

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

3.4 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.