

Produktbeschreibung

# *USB 4009*

## *basicCON 4009*

Widerstandsdekaden  
Nutzerhandbuch Version 1.2



GÖPEL electronic GmbH  
Göschwitzer Str. 58/60  
D-07745 Jena  
Tel.: +49-3641-6896-597  
Fax: +49-3641-6896-944  
E-Mail: [ats\\_support@goepel.com](mailto: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: 13.03.2012

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

**Stand: März 2012**

<b>1</b>	<b>INSTALLATION .....</b>	<b>1-1</b>
1.1	HARDWAREINSTALLATION .....	1-1
1.2	TREIBERINSTALLATION .....	1-2
<b>2</b>	<b>HARDWARE .....</b>	<b>2-1</b>
2.1	BESTIMMUNG .....	2-1
2.2	TECHNISCHE DATEN .....	2-3
2.2.1	<i>Abmessungen</i> .....	2-3
2.2.2	<i>Kennwerte</i> .....	2-3
2.3	AUFBAU UND FUNKTION .....	2-4
2.3.1	<i>Allgemeines</i> .....	2-4
2.3.2	<i>Adressierung</i> .....	2-5
2.3.3	<i>Stromversorgung</i> .....	2-5
2.3.4	<i>Bestückung</i> .....	2-6
2.3.5	<i>Front-LEDs</i> .....	2-6
2.3.6	<i>Belegung Frontsteckverbinder</i> .....	2-7
<b>3</b>	<b>ANSTEUERSOFTWARE .....</b>	<b>3-1</b>
3.1	PROGRAMMIEREN ÜBER G-API .....	3-2
3.2	PROGRAMMIEREN ÜBER DLL-FUNKTIONEN .....	3-3
3.2.1	<i>Driver_Info</i> .....	3-4
3.2.2	<i>DLL_Info</i> .....	3-5
3.2.3	<i>Xilinx_Download</i> .....	3-6
3.2.4	<i>Xilinx_Version</i> .....	3-7
3.2.5	<i>Write_COMMAND</i> .....	3-8
3.2.6	<i>Read_COMMAND</i> .....	3-9
3.2.7	<i>Reset Decade</i> .....	3-10
3.2.8	<i>Write Decade Data</i> .....	3-11
3.2.9	<i>Read Decade Data</i> .....	3-12
3.2.10	<i>Update Decade</i> .....	3-13
3.3	PROGRAMMIEREN MIT LABVIEW .....	3-14
3.3.1	<i>LabVIEW über G-API</i> .....	3-14
3.3.2	<i>LLB unter Verwendung des Windows Device Treibers</i> .....	3-14
3.4	WEITERE GÖPEL SOFTWARE .....	3-14
3.5	STEUERBEFEHLE USB CONTROLLER .....	3-15
3.5.1	<i>USB Befehlsaufbau</i> .....	3-15
3.5.2	<i>USB Antwortaufbau</i> .....	3-15
3.5.3	<i>USB Befehle</i> .....	3-15



# 1 Installation

## 1.1 Hardwareinstallation



Wir empfehlen, die Gerätetreiber-Software vor dem Anschließen der Baugruppen an den PC/ Laptop zu installieren (siehe Abschnitt [Treiberinstallation](#)).

### USB 4009:

Das USB 4009-Board kann nur in einem der GÖPEL electronic USB-Racks USB 1004, USB 1008 oder USB 1016 betrieben werden.



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

Wählen Sie einen freien Steckplatz in Ihrem USB-Rack aus. Falls vorhanden, muss zuerst das Slotblech entfernt werden, welches den Steckplatz abdeckt. Dazu sind die beiden Schrauben zu lösen. Führen Sie das Board über die Führungsschienen vorsichtig in den vorbereiteten Steckplatz ein und drücken Sie es das letzte Stück, mit etwas Kraft, bis zum Anschlag in den Steckplatz hinein. Schrauben Sie die beiden äußeren, an der Frontplatte befindlichen Schrauben fest, damit das Board einen sicheren Sitz hat.



Fassen Sie das Board bei der Montage nur an den Rändern oder an der Frontplatte an. Berühren Sie niemals die Oberfläche, da sonst die Gefahr der Zerstörung von Bauteilen durch elektrostatische Entladung besteht.

Zur Entfernung des Boards aus dem USB-Rack (falls notwendig), sind die beiden äußeren Schrauben wieder zu lösen. Mit dem an der Frontplatte befindlichen Hebel kann das Board aus dem Steckplatz herausgelöst und anschließend herausgezogen werden.

### basicCON 4009:

Das basicCON 4009 kann direkt an den PC oder Laptop angeschlossen werden.

Verbinden Sie die auf der Rückseite befindliche USB-Buchse über das beigelegte USB-Kabel mit Ihrem Rechner.

Schließen Sie, wenn nötig, siehe Kapitel [Stromversorgung](#), das beigelegte Netzteil oder eine eigene Spannungsquelle an die entsprechenden Anschlüsse ext. Power Supply auf der Rückseite des Gehäuses an.



Bitte vergleichen Sie das Kapitel [Adressierung](#) zur Installation mehrerer USB 4009-Boards.



Die Anschlüsse auf der Frontseite des USB 4009-Boards werden im Kapitel [Belegung Frontsteckverbinder](#) näher erläutert.

## 1.2 Treiberinstallation

Um die GÖPEL electronic USB-Treiber auf Ihrem System einzurichten, empfehlen wir das GUSB Treiber Setup.

Starten Sie dazu das auf der mitgelieferten CD enthaltene Setup Programm *GUSB-Setup-\*.exe* (der Stern steht für die Versionsnummer) und folgen Sie den Anweisungen.



Der zur Verfügung stehende Gerätetreiber unterstützt sowohl Windows® 2000/ XP- als auch Windows® 7-Systeme!



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

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

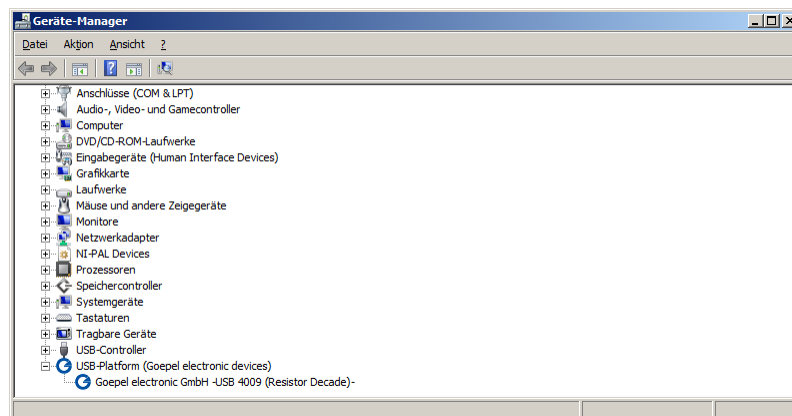
Durch die Plug-and-Play Fähigkeit von Windows® wird das Gerät automatisch vom Betriebssystem erkannt.

Am Ende des Installationsprozesses werden Sie von Windows® aufgefordert, Ihren Rechner neu zu starten.

Für einen sicheren und zuverlässigen Betrieb wird ein Neustart des Systems unbedingt empfohlen.

Nach der Treiberinstallation/ Hardwareinstallation können Sie überprüfen, ob die Boards einwandfrei vom System eingebunden worden sind.

Die folgende Abbildung zeigt die erfolgreiche Einbindung eines USB 4009-Boards:



**Abbildung 1-1:**  
**Anzeige Geräte-Manager**



Beachten Sie bitte, dass der Geräte-Manager ALLE USB-Controller anzeigt.

## 2 Hardware

### 2.1 Bestimmung

USB 4009 ist ein Widerstandsdekaden-Board der GÖPEL electronic GmbH. Diese Boards werden in der allgemeinen Steuerungs- und Prüftechnik verwendet, u.a. in der Automobiltechnik.

Die Boards haben folgende Eigenschaften:

- ♦ Zwei voneinander unabhängige Widerstandsdekaden
- ♦ Beide Widerstandsdekaden sind als ein Potentiometer mit Mittelabgriff kombinierbar
- ♦ Ausgabe der Widerstandswerte zur Eliminierung von Leitungs- und Übergangswiderständen in 4-Leiter-Technik möglich
- ♦ Galvanische Trennung des Anwendungsteils vom USB-Interface der Boards



Für den Betrieb des USB 4009-Boards ist ein GÖPEL electronic USB-Rack USB 1004, USB 1008 oder USB 1016 erforderlich, das bis zu 16 GÖPEL electronic USB-Boards aufnehmen kann. Die Stromversorgung erfolgt über das Netzteil des USB-Racks.



Abbildung 2-1:  
USB 4009



Beachten Sie bitte, dass ein Download des Xilinx FPGAs für die Funktion eines USB 4009/ basicCON 4009 unabdingbar ist (siehe [Xilinx Download](#) unter [Programmieren über DLL-Funktionen](#))!

basicCON 4009 ist ein GÖPEL electronic GmbH stand-alone Gerät auf der Grundlage eines USB 4009 Boards zum Anschluss an einen PC oder Laptop, das für den eigenständigen Einsatz außerhalb komplexer Testsysteme entwickelt wurde.

Das basicCON 4009 kann über die USB-Schnittstelle oder über die Buchsen für ext. Power Supply versorgt werden (siehe [Stromversorgung](#)).



Abbildung 2-2:  
basicCON 4009

An der Geräterückseite des basicCON 4009 befinden sich die folgenden Anschlüsse:



Abbildung 2-3:  
basicCON 4110 - Rückseite

- ◆ USB-B-Buchse für das USB 2.0 Interface mit USB-Standardbelegung
- ◆ DC-Buchse für das mitgelieferte Steckernetzteil
- ◆ Bananenbuchsen zur Stromversorgung



Bitte nutzen Sie für die externe Stromversorgung entweder die beiden Bananenbuchsen ODER die DC-Buchse.



Zur [Stromversorgung](#) vergleichen Sie bitte die Hinweise im entsprechenden Kapitel.



## 2.2 Technische Daten

### 2.2.1 Abmessungen

(Breite x Höhe x Tiefe):

- USB 4009: 4 TE x 130 mm x 185 mm
- basicCON 4009: 120 mm x 50 mm x 180 mm



Die Angaben für USB 4009 beziehen sich auf das Board in einem GÖPEL electronic USB-Rack.

### 2.2.2 Kennwerte

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bemerkungen
$U_B$	Betriebsspannung an den Widerstandsdekaden			48	V	
$P_B$	Betriebsleistung an den Widerstandsdekaden			300	mW	
$R_{Var}$	Einstellbarer Widerstandsbereich	1		$10^6$	$\Omega$	
$R_{Step}$	Auflösung		1		$\Omega$	
G	Genauigkeit Enthaltener Offset	1		150	% m $\Omega$	> 20 $\Omega$ < 20 $\Omega$



**warnung**

Bitte verwenden Sie zum Anschluss der Baugruppe an die USB-Schnittstelle des PCs/ Laptops das im Lieferumfang enthaltene USB-Kabel.

Andere Kabel sind u. U. nicht geeignet!

## 2.3 Aufbau und Funktion

### 2.3.1 Allgemeines

USB 4009 (auch im basicCON 4110 verwendet) ist ein Widerstandsgeber-Board mit USB 2.0 Schnittstelle. Ein ASIC mit integriertem 8Bit Microcontroller dient als Interface zum USB-Host im Rechner.

Das Board hat keine Jumper zur Hardwareerkennung und wird automatisch in das jeweilige System eingebunden.

USB 4009 bzw. basicCON 4009 verfügen über zwei Widerstandsdekaden mit den Werten von  $1\Omega$  bis  $1M\Omega$ . Die jeweiligen Werte werden über langzeit- und temperaturstabile Signalrelais verschaltetet.



Zur Einstellung der benötigten Widerstandswerte informieren Sie sich bitte im Abschnitt [Programmieren über DLL-Funktionen](#).

Das Stand-Alone-Gerät basicCON 4009 beruht auf dem Widerstandsgeber-Board USB 4009.

Abbildung 2-4 zeigt den prinzipiellen Aufbau einer Widerstandsdekade:

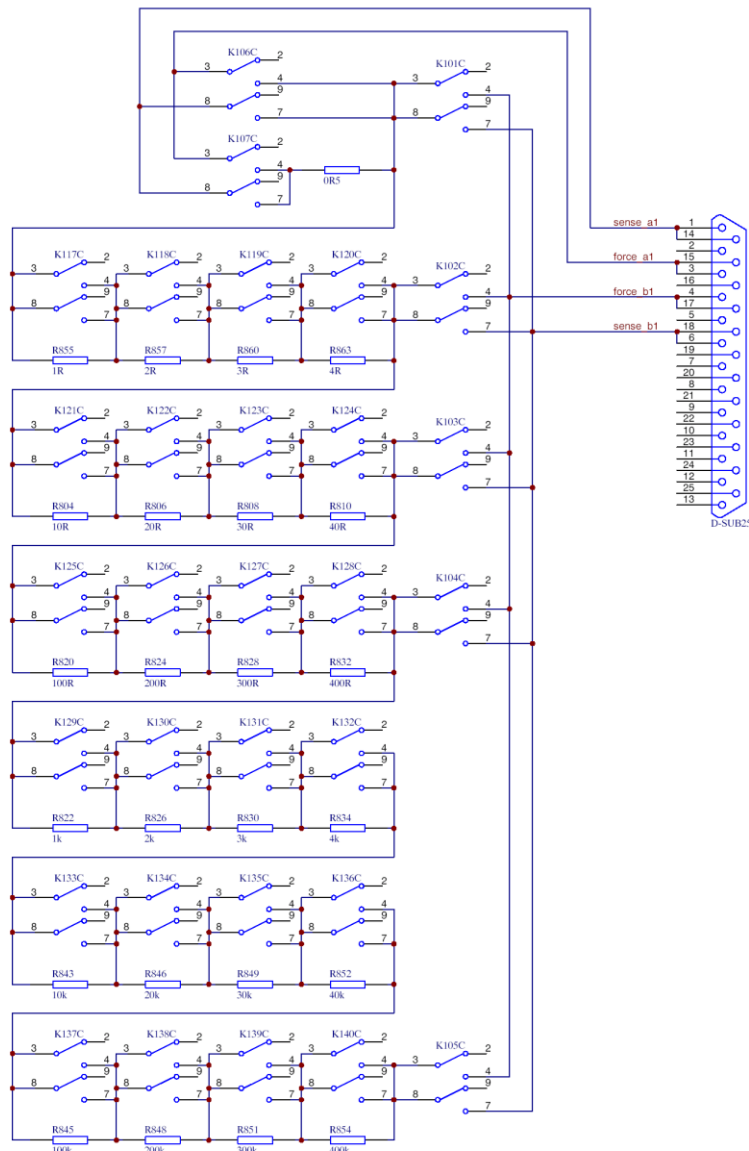


Abbildung 2-4: Prinzipielle Darstellung einer Widerstandsdekade

### 2.3.2 Adressierung

Bei Betrieb mehrerer USB 4009-Boards im GÖPEL electronic USB-Rack erfolgt die Adressierung ausschließlich über deren Seriennummern (siehe [Ansteuersoftware](#)): Das Gerät mit der KLEINSTEN Seriennummer ist immer das Gerät Nummer 1.



Zur Erhöhung der Übersichtlichkeit empfehlen wir, mehrere USB 4009-Boards in aufsteigender Reihenfolge ihrer Seriennummern im USB-Rack anzuordnen.

### 2.3.3 Stromversorgung

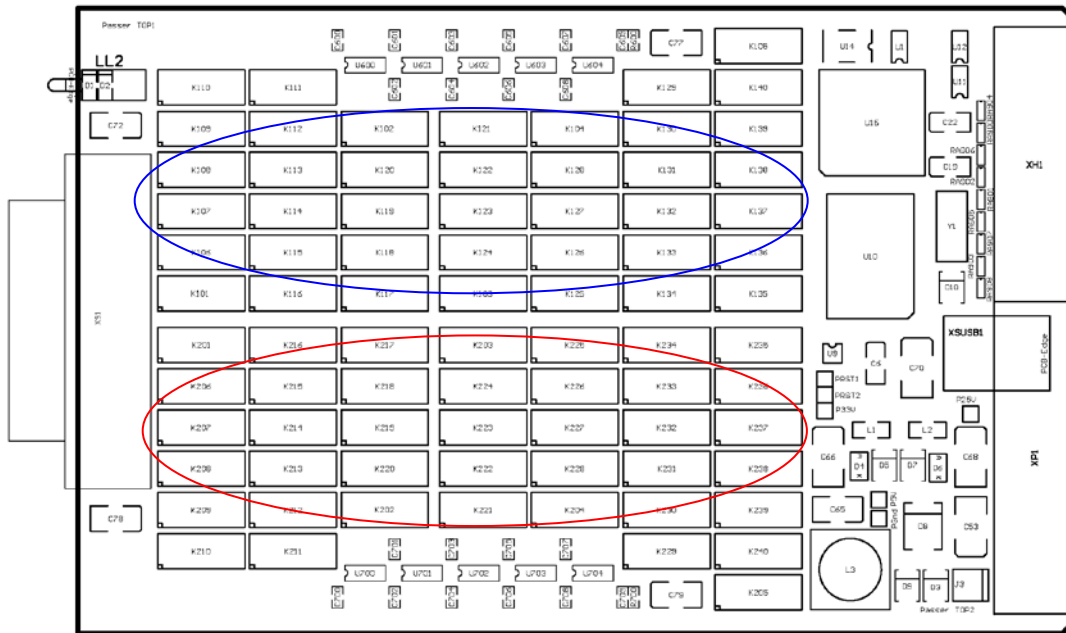
Das Board USB 4009 wird über das USB-Rack versorgt, in dem es installiert wurde.

Die Stromversorgung eines basicCON 4009 kann über die USB-Schnittstelle erfolgen, sofern diese für den benötigten Strom von ca. 500mA konfiguriert ist.

Bei Betrieb an passiven USB-Hubs oder Laptops mit geringerer USB-Stromversorgung muss das basicCON 4110 extern über die beiden Buchsen für ext. Power Supply (rot = plus/ blau = minus) mit 7-25VDC (max. 300mA bei 12V) oder über die Buchse für das beigelegte Steckernetzteil mit dem Hohlstecker (2,1 x 5,5mm/ Polarität + innen) versorgt werden, siehe auch Abbildung 2-3.

2.3.4 Bestückung Abbildung 2-5 zeigt schematisch die Bestückungsseite des Boards.

Widerstandsdekade 1



Widerstandsdekade 2

Abbildung 2-5: Schematischer Bestückungsplan USB4009

2.3.5 Front-LEDs

LED \ Status	EIN	AUS
1	Konfiguration/ Download des FPGAs läuft	Konfiguration/ Download des FPGAs läuft nicht
2	FPGA ist konfiguriert	FPGA ist nicht konfiguriert
3	nicht verwendet	
4	nicht verwendet	

### 2.3.6 Belegung Front- steckverbinder

Typ am Gerät: DSub 25-polig Stecker

Typ am Kabel: DSub 25-polig Buchse

Die Werte der beiden Widerstandsdekaden stehen über diesen Steckverbinder an der Frontseite zur Verfügung.

Die Belegung ist in der folgenden Tabelle dargestellt:

Ifd. Nr.	Anschluss XS1	Signalname	Bemerkung
1	1	Sense A1	Senseleitung Dekade 1
	14	Sense A1	
2	3	Force A1	Forceleitung Dekade 1
	15	Force A1	
3	4	Force B1	Forceleitung Dekade 1
	17	Force B1	
4	6	Sense B1	Senseleitung Dekade 1
	18	Sense B1	
5	8	Sense A2	Senseleitung Dekade 2
	21	Sense A2	
6	10	Force A2	Forceleitung Dekade 2
	22	Force A2	
7	11	Force B2	Forceleitung Dekade 2
	24	Force B2	
8	13	Sense B2	Senseleitung Dekade 2
	25	Sense B2	

#### Hinweise zur Steckverbinderbelegung

Das Layout eines USB 4009-Boards ermöglicht es, zur Eliminierung von Leitungs- und Übergangswiderständen die 4-Leiter-Technik anzuwenden.

Entsprechend sind die Anschlüsse des Steckverbinders für die Force- und Senseleitungen zu nutzen.

Damit kann eine genauere Einstellung der Widerstandswerte erreicht werden.

Selbstverständlich ist auch herkömmliche 2-Leiter-Technik möglich. Dann sind nur die Steckverbinder-Anschlüsse für die Forceleitungen zu belegen.



## 3 Ansteuersoftware

Zur Einbindung der USB 4009-/ basicCON 4009-Hardware in Ihre eigenen Applikationen existieren mehrere Möglichkeiten:

- ♦ [Programmieren über G-API](#)
- ♦ [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 damit 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.

Deshalb ist keine Anforderung von der Applikation erforderlich.

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 Board 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 integrierten Selbsttests zu überprüfen.

Die folgende Abbildung zeigt den GÖPEL electronic Hardware Explorer:

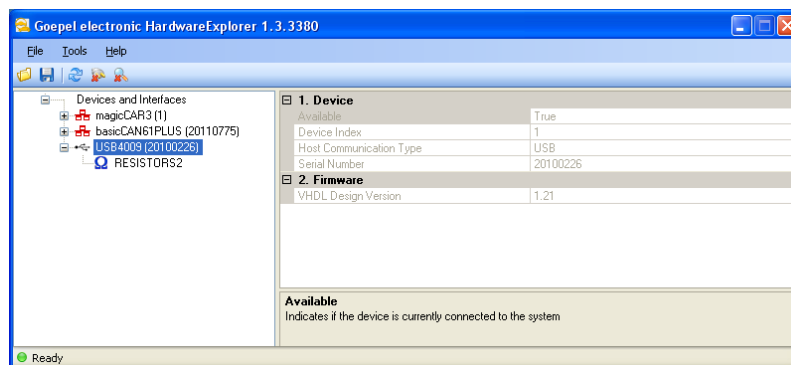


Abbildung 3-1  
Hardware Explorer



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

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



## 3.2 Programmieren über DLL-Funktionen



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



Der in der folgenden Funktionsbeschreibung verwendete Begriff GUSB\_Platform ist der Name eines USB Treibers der GÖPEL electronic GmbH.

Informationen zu den Strukturen, Funktionen und Error-Codes enthält das C-Header File *GUSB\_Platform.h* der mitgelieferten CD.

### Windows Device Treiber

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

- ◆ [Driver Info](#)
- ◆ [DLL Info](#)
- ◆ [Xilinx Download](#)
- ◆ [Xilinx Version](#)
- ◆ [Write COMMAND](#)
- ◆ [Read COMMAND](#)
- ◆ [Reset Decade](#)
- ◆ [Write Decade Data](#)
- ◆ [Read Decade Data](#)
- ◆ [Update Decade](#)

### 3.2.1 Driver\_Info

Die Funktion `GUSB_Platform_Driver_Info` dient zur Status-Abfrage des Hardware-Treibers und zur internen Initialisierung der erforderlichen Handles.



Diese Funktion MUSS einmalig vor dem Aufruf aller anderen Funktionen des `GUSB_Platform` Treibers ausgeführt werden.

#### Format:

```
int GUSB_Platform_Driver_Info(GUSB_Platform_DriverInfo *pDriverInfo,  
                             unsigned int LengthInByte);
```

#### Parameter:

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

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

`LengthInByte`

Länge der Datenstruktur, auf die `pDriverInfo` zeigt, in Byte

#### Beschreibung:

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

Dazu muss der Funktion die Adresse einer Struktur `pDriverInfo` übergeben werden. Mit Hilfe des Parameters `LengthInByte` prüft die Funktion intern den korrekt initialisierten Anwenderspeicher.

Die Funktion füllt die Struktur mit Angaben zur Treiberversion, der Anzahl aller sich im System befindenden `USB Controller` (die von diesem Treiber unterstützt werden) und Informationen darüber, wie z.B. die Seriennummer(n).



Die Bereitstellung der Hardwareinformationen und die Initialisierung der zugehörigen Handles sind für die weitere Nutzung der USB-Hardware zwingend erforderlich.

**3.2.2 DLL\_Info** Die Funktion `GUSB_Platform_DLL_Info` dient zur Abfrage von Informationen über die DLL.

**Format:**

```
int GUSB_Platform_DLL_Info(GUSB_Platform_DLLInfo *DLLInformation);
```

**Parameter**

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

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

**Beschreibung:**

Die Funktion `GUSB_Platform_DLL_Info` gibt die Struktur `DLLInfo` zurück.

Der erste Integerwert enthält die Versionsnummer der `GUSB_Platform.dll`.

**Beispiele:**

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

### 3.2.3 Xilinx\_ Download

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

**Format:**

```
int GUSB_Platform_Xilinx_Download(unsigned int DeviceName,
                                  unsigned int DeviceNumber,
                                  char *pFileName,
                                  unsigned char *pFirmwareErrorCode);
```

**Parameter:**

**DeviceName**

Typ des adressierten Gerätes (Nummer, die in *GUSB\_Platform\_def.h* deklariert ist, für USB 4009/ basicCON 4009 = 12)

**DeviceNumber**

Nummer des adressiertenGerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer DeviceNumber 1).

**pFileName**

Pfad des zu ladenden FPGA-Files

**pFirmwareErrorCode**

Fehlercode, der während der Abarbeitung dieser DLL-Funktion auftritt (Bei Fehlercode 0 ist kein Fehler aufgetreten.)

(error codes -> card firmware siehe *GUSB\_Platform\_def.h*)

**Beschreibung:**

Die Funktion `GUSB_Platform_Xilinx_Download` dient zum Laden eines FPGA-Files in den XILINX (Extension *\*.cfd*).



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

### 3.2.4 Xilinx\_ Version

Die Funktion `GUSB_Platform_Xilinx_Version` ermöglicht das Auslesen der geladenen XILINX-Firmwareversion.

#### Format:

```
int GUSB_Platform_Xilinx_Version(unsigned int DeviceName,  
                                unsigned int DeviceNumber,  
                                unsigned int *Version);
```

#### Parameter:

##### DeviceName

Typ des adressierten Gerätes (Nummer, die in `GUSB_Platform_def.h` deklariert ist, für USB 4009/ basicCON 4009 = 12)

##### DeviceNumber

Nummer des adressierten Gerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer DeviceNumber 1).

##### Version

XILINX Softwareversion

#### Beschreibung:

Mit der Funktion `GUSB_Platform_Xilinx_Version` kann die Versionsnummer der im FPGA geladenen Software ausgelesen werden.

#### Beispiele:

Die Versionsnummer 2.34 wird als Wert 234 zurückgegeben, Version 2.60 als Wert 260.

### 3.2.5 Write\_COMMAND

Die Funktion `GUSB_Platform_Write_COMMAND` dient zum Senden eines Configurations-Befehls zum USB Controller.

#### Format:

```
int GUSB_Platform_Write_COMMAND(
                                unsigned int DeviceName,
                                unsigned int DeviceNumber,
                                t_USB_COMMAND_Interface_Buffer *pWrite,
                                unsigned int DataLength);
```

#### Parameter:

##### DeviceName

Typ des adressierten Gerätes (Nummer, die in `GUSB_Platform_def.h` deklariert ist, für USB 4009/ basicCON 4009 = 12)

##### DeviceNumber

Nummer des adressierten Gerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer DeviceNumber 1).

Zeiger, z.B. `pWrite`, auf den Bereich für Schreibdaten

##### DataLength

Byteanzahl der in `pWrite` zu schreibenden Daten  
(z. Zt. max. 64 Byte pro Befehl,  
siehe auch [Steuerbefehle USB Controller](#))

#### Beschreibung:

Die Funktion `GUSB_Platform_Write_COMMAND` sendet einen Befehl zum USB Controller.

Die allgemeine Struktur ist im Abschnitt [Steuerbefehle USB Controller](#) beschrieben.

### 3.2.6 Read\_COMMAND

Die Funktion `GUSB_Platform_Read_COMMAND` dient zum Lesen einer Antwort vom USB Controller.

#### Format:

```
int GUSB_Platform_Read_COMMAND(
                                unsigned int DeviceName,
                                unsigned int DeviceNumber,
                                t_USB_COMMAND_Interface_Buffer *pRead,
                                unsigned int *DataLength);
```

#### Parameter:

##### DeviceName

Typ des adressierten Gerätes (Nummer, die in `GUSB_Platform_def.h` deklariert ist, für USB 4009/ basicCON 4009 = 12)

##### DeviceNumber

Nummer des adressierten Gerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer DeviceNumber 1).

Zeiger, z.B. `pRead`, auf den Lesepuffer

(Nach erfolgreicher Funktionsausführung befinden sich die Daten im Lesepuffer, bestehend aus Antwortkopf und Antwortbytes, z. Zt. min. 64 Byte pro Antwort, siehe auch [Steuerbefehle USB Controller](#)).

##### DataLength

Vor Funktionsaufruf: Anzugebende Größe des Lesepuffers in Bytes

Nach Funktionsausführung: Anzahl der tatsächlich gelesenen Bytes

#### Beschreibung:

Die Funktion `GUSB_Platform_Read_COMMAND` liest die älteste vom USB Controller geschriebene Antwort.

Werden mehrere Antworten vom USB Controller bereitgestellt, werden maximal zwei dieser Antworten in den Puffer des USB Controllers geschrieben.

Weitere ggf. bereitgestellte Antworten gehen verloren!

### 3.2.7 Reset Decade

Die Funktion `GUSB_Platform_4009_ResetDecade` ermöglicht das Rücksetzen der Widerstandsdekaden.

**Format:**

```
int GUSB_Platform_4009_ResetDecade(unsigned int DeviceNumber,  
                                   unsigned int ResetMode);
```

**Parameter:**

**DeviceNumber**

Nummer des adressiertenGerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer DeviceNumber 1).

**ResetMode**

setzt die angegebene Dekade zurück  
(Dekade 1, Dekade 2, Dekade 1+2)

**Beschreibung:**

Mit der Funktion `GUSB_Platform_4009_ResetDecade` kann ein dekadenselektiver RESET-Vorgang ausgelöst werden.

Dabei werden entsprechend dem gewählten `ResetMode` alle internen Register zurückgesetzt, und die zugehörigen Relais werden geöffnet ( $R \Rightarrow \infty$ ).



### 3.2.8 Write Decade Data

Die Funktion `GUSB_Platform_4009_WriteDecadeData` konfiguriert den Widerstandswert für die angegebene Dekade.

#### Format:

```
int GUSB_Platform_4009_WriteDecadeData(unsigned int DeviceNumber,  
                                       unsigned int Decade,  
                                       double DecadeValue);
```

#### Parameter:

##### DeviceNumber

Nummer des adressiertenGerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer `DeviceNumber 1`).

##### Decade

Nummer der Dekade (1..2)

##### DecadeValue

Zu setzender Widerstandswert in Ohm

#### Beschreibung:

Die Funktion `GUSB_Platform_4009_WriteDecadeData` schreibt den Widerstandswert in den karteninternen Registerbereich, jedoch erfolgt noch keine Datenausgabe am zugehörigen DSUB-Pin.

### 3.2.9 Read Decade Data

Die Funktion `GUSB_Platform_4009_ReadDecadeData` liest den aktuellen Widerstandswert für die angegebene Dekade.

**Format:**

```
int GUSB_Platform_4009_ReadDecadeData(unsigned int DeviceNumber,
                                       unsigned int Decade,
                                       t_GUSB_Platform_4009_ReadDecadeData *DecadeValue),
                                       double *DecadeValueSingle);
```

**Parameter:**

**DeviceNumber**

Nummer des adressiertenGerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer DeviceNumber 1).

**Decade**

Nummer der Dekade (1..2)

**DecadeValue**

Anzeige der einzelnen Widerstandswerte, aus denen sich der Gesamtwiderstandswert der zugehörigen Dekade zusammensetzt

**DecadeValueSingle**

Anzeige des Gesamtwiderstands

**Beschreibung:**

Die Funktion `GUSB_Platform_4009_ReadDecadeData` liest den Widerstandswert aus dem karteninternen Registerbereich.

Der Wert wird zunächst nicht als Summenwert dargestellt, sondern auf die einzelnen intern genutzten Widerstandsbereiche aufgeschlüsselt (siehe Struktur in `GUSB_Platform.h` auf der mitgelieferten CD).

Unter `DecadeValueSingle` wird der resultierende Gesamtwiderstand angezeigt.

### 3.2.10 Update Decade

Die Funktion `GUSB_Platform_4009_UpdateDecade` aktualisiert den ausgegebenen Widerstandswert.

#### Format:

```
int GUSB_Platform_4009_UpdateDecade(unsigned int DeviceNumber,  
                                     unsigned int UpdateMode);
```

#### Parameter:

##### DeviceNumber

Nummer des adressiertenGerätes. Wenn mehrere Geräte gleichen Typs angeschlossen sind, erfolgt die Nummerierung in aufsteigender Reihenfolge der Seriennummern (das Gerät mit der NIEDRIGSTEN Seriennummer hat immer DeviceNumber 1).

##### UpdateMode

gibt die zu aktualisierende Dekade an (Dekade 1, Dekade 2, Dekade 1+2)

#### Beschreibung:

Die Funktion `GUSB_Platform_4009_UpdateDecade` gibt den in den internen Registerbereichen gesetzten Widerstandswert am DSUB-Pin der entsprechenden Dekade aus.

## 3.3 Programmieren mit LabVIEW

### 3.3.1 LabVIEW über G-API

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe USB 4009-Boards oder basicCON 4009-Geräte unter LabVIEW angesprochen werden können.

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

### 3.3.2 LLB unter Verwendung des Windows Device Treibers

Auf der mitgelieferten CD befindet sich eine VI-Sammlung, mit deren Hilfe USB 4009-Boards oder basicCON 4009-Geräte unter LabVIEW angesprochen werden können.

Dabei werden die Funktionen genutzt, die im Abschnitt [Programmieren über DLL-Funktionen](#) beschrieben worden sind.

## 3.4 Weitere GÖPEL Software

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

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

## 3.5 Steuerbefehle USB Controller

Der USB Controller ist für die Anbindung der USB 4009 oder basicCON 4009 Baugruppe an den PC über USB 2.0 zuständig.

An diesen USB Controller können Nachrichten (i. Allg. USB Befehle) gesendet werden, die für Konfigurationszwecke benötigt werden.

### 3.5.1 USB Befehlsaufbau

Ein USB Befehl besteht aus vier Bytes Header und den Daten (nicht alle USB Befehle benötigen Daten!).

Der Header eines USB Befehls ist folgendermaßen aufgebaut:

Bytenummer	Bedeutung	Inhalt
0	StartByte	0x23 (ASCII-Zeichen "#")
1	Command	(0x..) Verwendete Codes entsprechend <a href="#">USB Befehle</a>
2	reserviert	0x00
3	reserviert	0x00

### 3.5.2 USB Antwortaufbau

Genau wie der USB Befehl, ist auch die USB Antwort in vier Bytes Header und die Daten unterteilt (nicht alle USB Befehle senden Daten zurück!).

Der Header einer USB Antwort ist folgendermaßen aufgebaut:

Bytenummer	Bedeutung	Inhalt
0	StartByte	0x24
1	Command	(0x..) Verwendete Codes entsprechend <a href="#">USB Befehle</a>
2	Length	Vom Befehl abhängige Länge
3	ErrorCode	Gibt den Fehlercode des Befehls zurück

### 3.5.3 USB Befehle

Gegenwärtig steht nur der USB Befehl READ\_SW\_VERSION zur Verfügung.

Command	Bezeichnung	Bedeutung
0x04	READ_SW_VERSION	Liefert die Version des USB Controllers  Antwort: Byte 4: low Byte generic Softwareversion Byte 5: high Byte generic Softwareversion Byte 6: low Byte Softwareversion des funktionellen Teiles Byte 7: high Byte Softwareversion des funktionellen Teiles



---

**4**

- 4009
  - Adressierung .....2-5
  - Bestimmung .....2-1
  - Ressourcen.....2-1
  - Stand-alone Ausführung .2-2
  - Stromversorgung .....2-5

---

**B**

- basicCON 4009
  - Anschlüsse .....2-2

---

**D**

- DLL Funktionen .....3-3

---

**G**

- G-API .....3-2
- GOPEL Firmware.....3-3
- GÖPEL USB-Racks .....1-1

---

**H**

- Hardware Explorer.....3-2

---

**I**

- Installation
  - Hardware .....1-1
  - Treiber .....1-2

---

**S**

- Steckverbinder
  - Front.....2-7

---

**U**

- USB Antwortaufbau .....3-15
- USB Befehle .....3-15
- USB Befehlsaufbau .....3-15
- USB Controller .....3-15
- USB Kabel .....2-3

---

**W**

- Windows Treiber .....3-3