

USB 3072

basicLIN 3072

LIN Schnittstellen
Nutzerhandbuch Version 1.2

© 2010 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: 10.06.2010

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

Stand: Juni 2010

1	INSTALLATION	1-1
1.1	HARDWAREINSTALLATION	1-1
1.2	TREIBERINSTALLATION	1-2
2	HARDWARE	2-1
2.1	BESTIMMUNG	2-1
2.2	TECHNISCHE DATEN	2-4
2.2.1	<i>Abmessungen</i>	2-4
2.2.2	<i>Kennwerte</i>	2-4
2.3	AUFBAU	2-5
2.3.1	<i>Allgemeines</i>	2-5
2.3.2	<i>Adressierung</i>	2-5
2.3.3	<i>Kommunikationsschnittstellen</i>	2-6
2.3.4	<i>Bestückung</i>	2-8
2.3.5	<i>Belegung Steckverbinder</i>	2-9
2.3.6	<i>LED Anzeige</i>	2-10
2.4	LIEFERHINWEISE	2-11
3	ANSTEUERSOFTWARE	3-1
3.1	PROGRAMMIEREN ÜBER G-API	3-1
3.2	PROGRAMMIEREN ÜBER DLL-FUNKTIONEN	3-1
3.2.1	<i>Windows Device Treiber</i>	3-2
3.2.1.1	<i>Driver_Info</i>	3-4
3.2.1.2	<i>DLL_Info</i>	3-5
3.2.1.3	<i>Write_FIFO</i>	3-6
3.2.1.4	<i>Read_FIFO</i>	3-7
3.2.1.5	<i>Read_FIFO_Timeout</i>	3-8
3.2.1.6	<i>Write_COMMAND</i>	3-9
3.2.1.7	<i>Read_COMMAND</i>	3-10
3.2.1.8	<i>Xilinx_Download</i>	3-11
3.2.1.9	<i>Xilinx_Version</i>	3-12
3.3	PROGRAMMIEREN MIT LABVIEW	3-13
3.3.1	<i>LabVIEW über G-API</i>	3-13
3.3.2	<i>LLB unter Verwendung des Windows Device Treibers</i>	3-13
3.4	WEITERE GÖPEL SOFTWARE	3-13
3.5	STEUERBEFEHLE USB CONTROLLER	3-14
3.5.1	<i>USB Befehlsaufbau</i>	3-14
3.5.2	<i>USB Antwortaufbau</i>	3-14
3.5.3	<i>USB Befehle</i>	3-14

1 Installation

1.1 Hardwareinstallation

Die Hardware-Installation beschränkt sich bei USB/ basicLIN 3072 i. Allg. auf den Austausch von Transceivermodulen.



Stellen Sie bitte unbedingt sicher, dass alle Installationsarbeiten im **ausgeschalteten** Zustand Ihres Systems erfolgen!

Wenn es notwendig ist, Transceivermodule zu tauschen, wird das entsprechende Gerät gemäß seinen Gegebenheiten geöffnet.

Dabei sind die allgemeinen Regeln zur Vermeidung von elektrostatischen Aufladungen zu beachten.

Transceivermodule dürfen nie unter Spannung gezogen oder gesteckt werden!

Außerdem ist unbedingt ein lagerichtiges Stecken der Module zu realisieren (siehe [Bestückung](#)).

1.2 Treiberinstallation

Um die GÖPEL electronic USB-Treiber auf Ihrem System einzurichten, muss das GUSB Treiber Setup ausgeführt werden. 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 Devicetreiber unterstützt gegenwärtig ausschließlich Windows® 2000/ XP-Systeme!

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



Die USB-Schnittstelle nutzt, falls möglich, die high-speed Datenrate entsprechend USB2.0 Spezifikation (ansonsten full-speed).

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

Die folgende Abbildung zeigt die erfolgreiche Einbindung einer USB 3072 bzw. basicLIN 3072-Baugruppe mit drei Controllern:

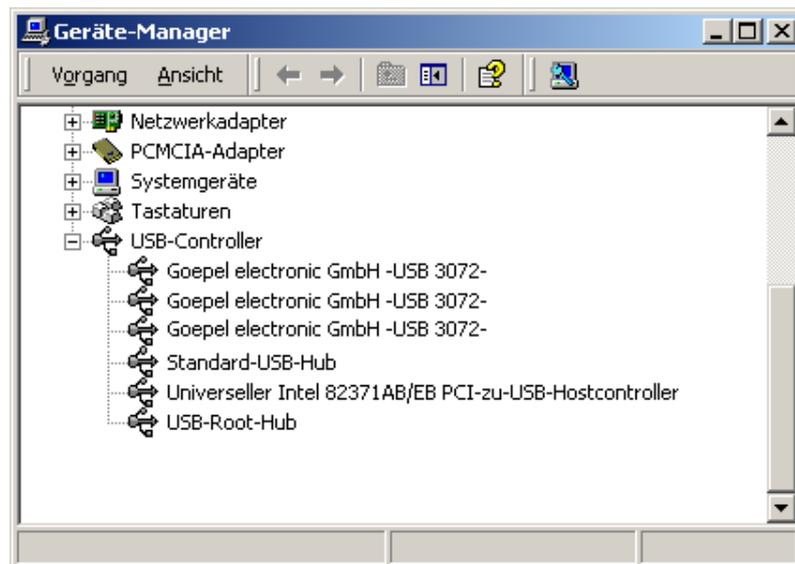


Abbildung 1-1:
Anzeige Geräte-Manager



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

2 Hardware

2.1 Bestimmung

USB 3072 LIN Boards sind Kommunikationsboards mit USB 2.0-Interface der GÖPEL electronic GmbH.

Diese Boards werden in der allgemeinen Steuerungstechnik verwendet, u.a. in der Automobiltechnik.



Abbildung 2-1:
USB 3072



Beachten Sie bitte, dass ein Download des Xilinx FPGAs für die Funktion eines USB 3072 Boards unabdingbar ist (siehe [Xilinx Download](#) unter [Windows Device Treiber](#)).



Zum Betrieb von USB 3072 Boards ist das GÖPEL electronic USB-Rack erforderlich, das bis zu 16 GÖPEL electronic USB-Boards aufnehmen kann.

Die Stromversorgung erfolgt in diesem Fall über das eingebaute Netzteil.

basicLIN 3072 ist ein GÖPEL electronic GmbH Stand-alone-Gerät auf der Grundlage eines USB 3072 Kommunikationsboards zum Anschluss an einen PC oder Laptop, das für den eigenständigen Einsatz außerhalb komplexer Testsysteme entwickelt wurde.

Die externe Spannungszufuhr erlaubt die Nutzung dieses Gerätes zur Datenaufnahme und Signalkontrolle u.a. in Kraftfahrzeugen.



Abbildung 2-2:
basicLIN 3072

Die Stromversorgung mit 8-25 VDC (und ca. 500 mA bei 12V) erfolgt über die beiden Buchsen für ext. Power Supply (rot = plus/ blau = minus) an der dem Steckverbinder für die LIN-Schnittstellen gegenüber liegenden Geräteseite.

Diese Buchsen werden zur Versorgung der internen Logik genutzt. Der Stromversorgungsanschluss an der blauen Buchse ist mit den GND-Anschlüssen der USB-Schnittstelle und der LIN-Schnittstellen verbunden.

Eigenschaften von USB/ basicLIN 3072:

- ◆ 3 LIN Schnittstellen Version 2.0
- ◆ Optional kann jede Schnittstelle auch als K-Line Interface ausgeführt werden
- ◆ Erweiterte Triggerfunktionen mit je einem Triggerinput und Triggeroutput pro Schnittstelle zum Frontsteckverbinder bzw. zur Backplane
- ◆ Jede LIN Schnittstelle verfügt über einen 32 Bit µController (TriCore TC1765, 40MHz)
- ◆ Visualisierung der Controllerzustände auf der Frontplatte mit je zwei LEDs (siehe [LED Anzeige](#))
- ◆ Hohe Flexibilität durch steckbare Transceivermodule



In diesem Nutzerhandbuch ist unter Controller IMMER einer der jeder LIN/ K-Line-Schnittstelle zugeordneten Microcontroller zu verstehen (unabhängig von der Bezeichnung „LIN Controller“ auf der Frontplatte eines USB 3072 Boards).

Sollten die Ressourcen eines basicLIN3072 für Ihre Anwendungen nicht ausreichen, steht das GÖPEL electronic USB-Rack zur Verfügung, das bis zu 16 GÖPEL electronic USB-Boards aufnehmen kann.

In diesem Fall erfolgt die Stromversorgung über ein eingebautes Netzteil mit 230V oder 115V Kaltgeräte-Anschluss an der Rückseite des Racks.

2.2 Technische Daten

- 2.2.1 Abmessungen (Breite x Höhe x Tiefe):
- ◆ USB 3072: 4 TE x 130 mm x 185 mm
 - ◆ basicLIN 3072: 145 mm x 70 mm x 220 mm



Die Angaben für USB 3072 beziehen sich auf ein installiertes Board.

- 2.2.2 Kennwerte Eine USB/ basicLIN 3072-Baugruppe hat folgende Kennwerte:

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bemerkung
Ext. Power Supply	Versorgungsspannung interne Logik	8		25	V	
V _{BAT}	Batteriespannung		12	27	V	für LIN
	Übertragungsrate			22	kBaud	
	Externer Triggereingang	3,3		50	V	
	Externer Triggerausgang		5		V	



Die zusätzlichen Triggereingänge können auch für die Einspeisung eines externen Taktsignals genutzt werden (siehe auch Firmware-Befehl `0x51 LIN Monitor - Zeitstempel konfigurieren` in der Dokumentation GÖPEL Firmware).

2.3 Aufbau

2.3.1 Allgemeines

USB/ basicLIN 3072-Baugruppen verfügen über drei LIN Schnittstellen (optional auch K-Line möglich).

Jede LIN Schnittstelle wird durch einen eigenen Microcontroller bedient.

Die USB-Informationen werden auf den Baugruppen an die Controller verteilt.

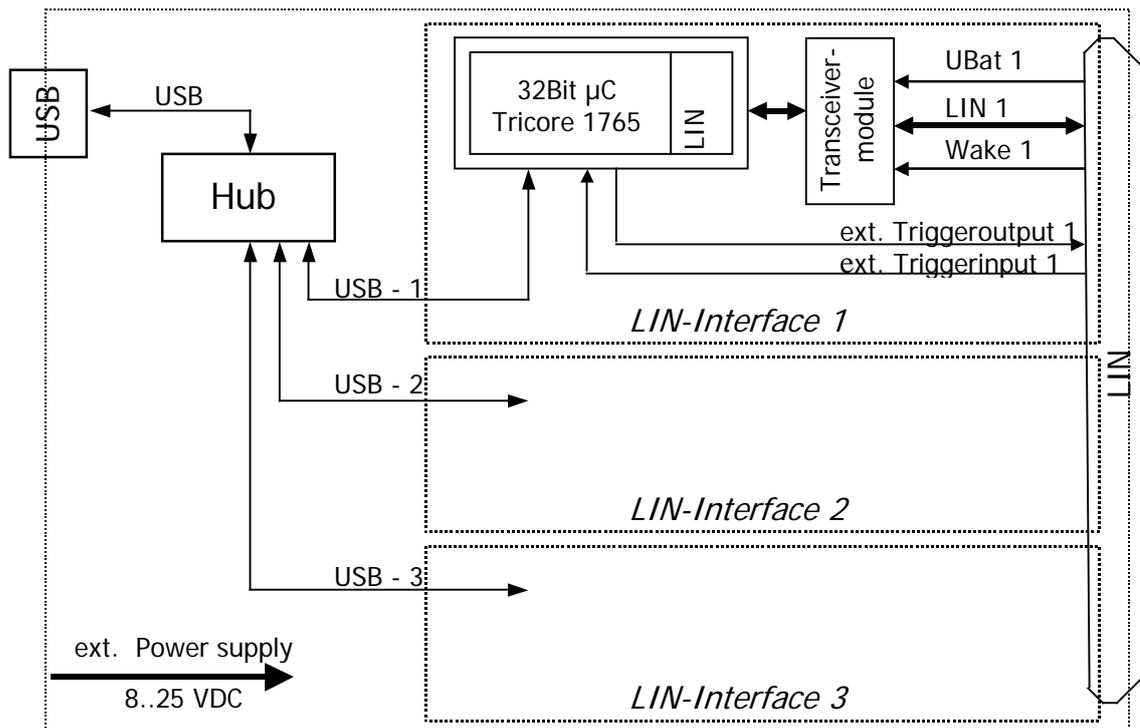


Abbildung 2-3: Blockdiagramm USB 3072



warnung

Bitte verwenden Sie zum Anschluss von USB/ basicLIN 3072-Baugruppen an die USB-Schnittstelle des PCs die im Lieferumfang enthaltenen USB-Kabel.

Andere Kabel sind u. U. nicht geeignet!

2.3.2 Adressierung

Die Adressierung einzelner USB/ basicLIN 3072-Baugruppen bei gleichzeitigem Betrieb mehrerer Baugruppen am gleichen Rechner erfolgt ausschließlich über die Seriennummern der LIN-Controller (siehe [Ansteuersoftware](#)): Der LIN-Controller mit der KLEINSTEN Seriennummer ist immer das Gerät Nummer 1.



Tip

Zur Erhöhung der Übersichtlichkeit empfehlen wir, USB/ basicLIN 3072-Baugruppen in aufsteigender Reihenfolge der Seriennummern ihrer LIN-Controller im USB-Rack anzuordnen/ am Rechner anzuschließen.

2.3.3 Kommunikations-schnittstellen

**max. 3 x LIN-Interface Version 2.0 oder
max. 3 x K-Line Interface (ISO 9141):**

Die folgende Abbildung zeigt die Ausgangsbeschaltung eines USB 3072-Boards zwischen den Transceivermodulen und dem Frontsteckverbinder:

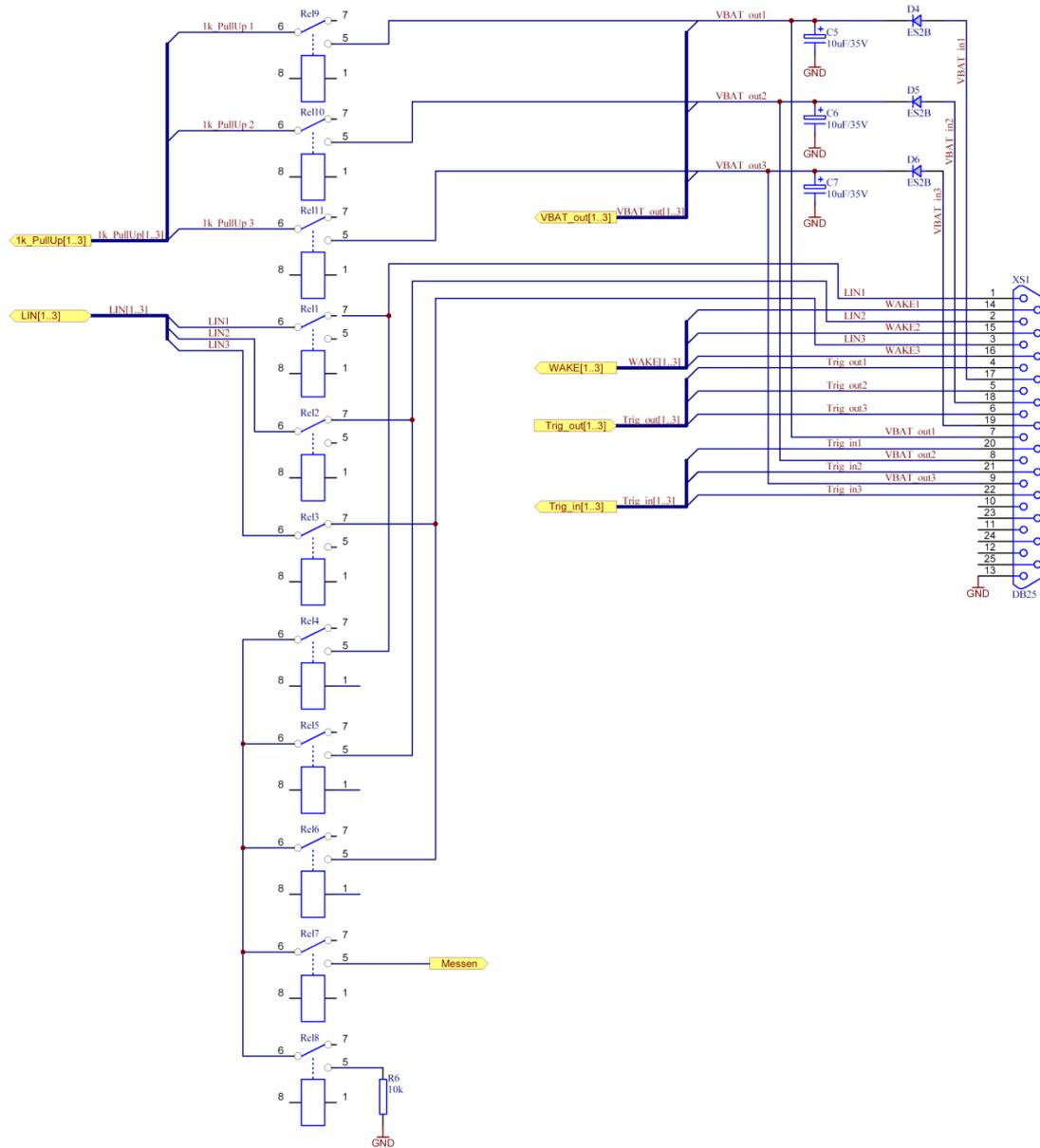


Abbildung 2-4: Stromlaufplanausschnitt Ausgang USB 3072

Bei eventuellem Wechsel der Transceiver Module muss darauf geachtet werden, dass die eckigen Pins (Pin1) von Fassung und Transceiver-Modul übereinander liegen.

LIN:

Die Transceiver sind als steckbare Module ausgeführt. I. Allg. wird der TJA1020 für diese Transceiver verwendet.

In der Standardausführung der Transceivermodule kann per Software über die Relais Rel9 für LIN1, Rel10 für LIN2 bzw. Rel11 für LIN3 zwischen Master- und Slave-Konfiguration umgeschaltet werden.

Die Pullup-Widerstände für LIN befinden sich auf dem Transceivermodul.

Über die Anschlüsse V_{BAT} wird die Versorgungsspannung der Transceivermodule angeschlossen. Gemäß der LIN-Spezifikation soll diese Versorgung über eine Verpoldiode und einen Stützkondensator erfolgen, sodass die Spannung V_{BAT} über die Anschlüsse $VBAT\ in1..VBAT\ in3$ eingespeist wird. Die hier eingespeiste Spannung darf nur im erlaubten Bereich des Transceivers liegen (z.B. ist bei einem TJA1020 die obere Grenze für V_{BAT} 27 V).

Die Anschlüsse $VBAT\ out1..VBAT\ out3$ sind Monitoringausgänge, um die tatsächlich am Transceiver anliegende Spannung zu messen und für eventuelle Vergleiche mit dem LIN-Signalepegel zu nutzen.

Für spezielle Steuerungs- und Messaufgaben verfügen die Kommunikationsboards über je einen externen Hardwaretriggerausgang und -eingang pro Schnittstelle. Deren genaue Funktion kann dem Abschnitt LIN Befehle der Dokumentation GÖPEL Firmware entnommen werden.

Die USB 3072-Boards bieten zudem die Möglichkeit, den LIN-Kommunikationsbus über die Relais 1..3 vom Testobjekt (device oder unit under test) zu trennen.

Die Relais 4..6 ermöglichen das Zusammenschalten aller drei LIN-Interfaces zu einem gemeinsamen Bus auf den Boards (siehe Abbildung 2-4).

K-Line:

Die Transceiver sind als steckbare Module ausgeführt. I. Allg. wird der L9637 der Fa. ST für diese Transceiver verwendet.

Über die Anschlüsse $VBAT\ in1..VBAT\ in3$ wird die Versorgungsspannung der Transceivermodule angeschlossen. Die hier eingespeiste Spannung darf nur im erlaubten Bereich des Transceivers liegen (z.B. ist bei einem L9637 die obere Grenze für V_{BAT} 36 V).

Zum Überbrücken der Verpolschutzdiode für V_{BAT} für LIN kann die Spannung V_{BAT} auch über die Anschlüsse $VBAT\ out1..VBAT\ out3$ eingespeist werden.

2.3.4 Bestückung

Abbildung 2-5 zeigt schematisch die Bestückungsseite eines USB 3072-Boards.

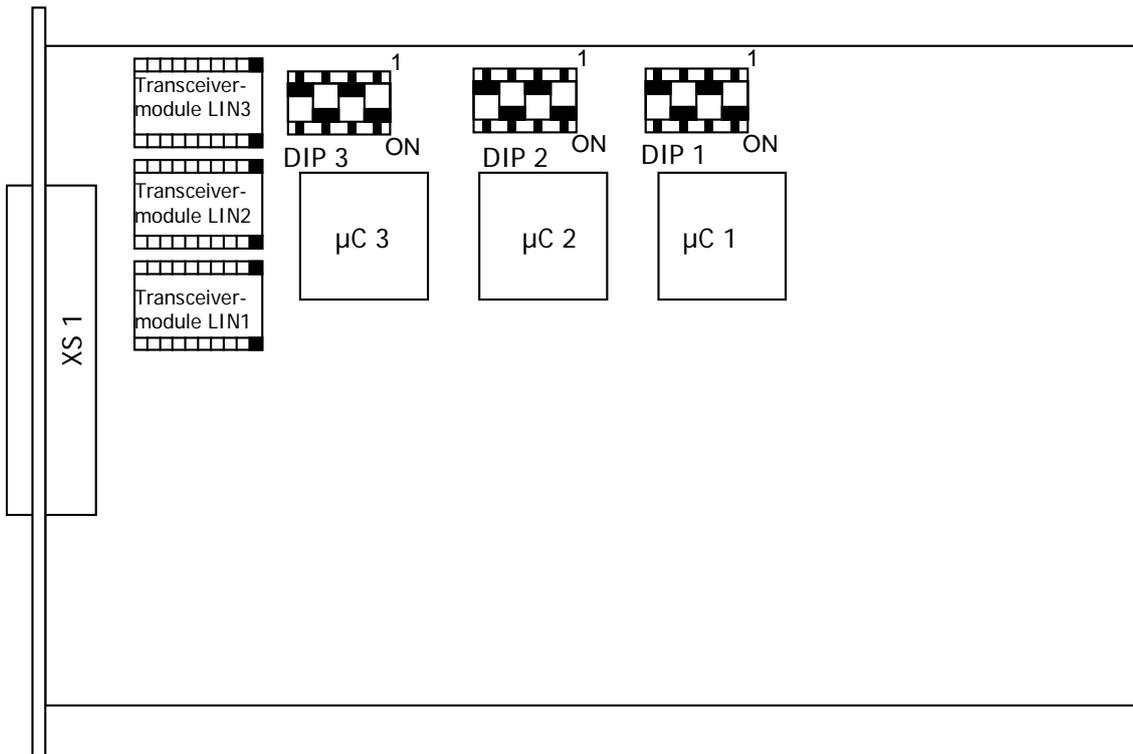


Abbildung 2-5: Schematischer Bestückungsplan eines USB 3072-Boards

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

LIN1	Transceivermodul für LIN1/ KLine1
LIN2	Transceivermodul für LIN2/ KLine2
LIN3	Transceivermodul für LIN3/ K-Line3
DIP1..3	DIP-Schalter der USB 3072-Boards zur Konfiguration der Microcontroller Die Einstellung darf nicht verändert werden!

2.3.5 Belegung Steckverbinder

Typ: DSub 25-polig Buchse

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

Anschluss XS1	Signalname	Bemerkung
1	LIN1/ K-Line1	LIN-(K-Line)-Bus-Leitung1
2	LIN2/ K-Line2	LIN-(K-Line)-Bus-Leitung2
3	LIN3/ K-Line3	LIN-(K-Line)-Bus-Leitung3
4	Trig out1	Triggerausgang 1
5	Trig out2	Triggerausgang 2
6	Trig out3	Triggerausgang 3
7	VBAT out1	Messausgang für Versorgungsspannung Transceiver 1 bzw. Versorgungsspannungseingang für Transceiver 1 ohne Verpolschutzdiode
8	VBAT out2	Messausgang für Versorgungsspannung Transceiver 2 bzw. Versorgungsspannungseingang für Transceiver 2 ohne Verpolschutzdiode
9	VBAT out3	Messausgang für Versorgungsspannung Transceiver 3 bzw. Versorgungsspannungseingang für Transceiver 3 ohne Verpolschutzdiode
10	-	nicht belegt
11	-	nicht belegt
12	-	nicht belegt
13	Gnd	Massepotenzial
14	Wake1	Transceiver Wake-Eingang 1
15	Wake2	Transceiver Wake-Eingang 2
16	Wake3	Transceiver Wake-Eingang 3
17	VBAT in1	Versorgungsspannungseingang für Transceiver 1 über Verpolschutzdiode
18	VBAT in2	Versorgungsspannungseingang für Transceiver 2 über Verpolschutzdiode
19	VBAT in3	Versorgungsspannungseingang für Transceiver 3 über Verpolschutzdiode
20	Trig in1	Triggereingang 1
21	Trig in2	Triggereingang 2
22	Trig in3	Triggereingang 3
23	-	nicht belegt
24	-	nicht belegt
25	-	nicht belegt



Bei LIN sind ggf. auf PIN 14/ 15/ 16 die WAKE-Anschlüsse verdrahtet (in Abhängigkeit von der Transceiver-Wahl).

USB-Schnittstelle

Die USB-B-Buchse (USB-Standardbelegung) für das USB 2.0 Interface befindet sich an der dem Steckverbinder für die LIN Schnittstellen gegenüber liegenden Seite eines USB 3072-Boards.

2.3.6 LED Anzeige

Die auf der Frontplatte des USB 3072-Boards angeordneten Leuchtdioden geben Auskunft über den momentanen Betriebszustand des jeder LIN Schnittstelle (auch als „LIN Port“ bezeichnet) zugeordneten Controllers. Einer Schnittstelle sind je eine grüne und rote LED zugeordnet, die wie folgt angeordnet sind:

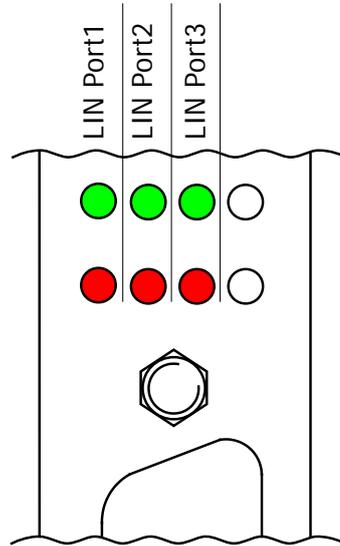


Abbildung 2-6:
LED Anzeige

Anzeigezustände dieser LEDs:

LED grün	LED rot	Bemerkung
leuchten dauerhaft		Controller läuft nicht Wahrscheinliche Fehlerursache: Xilinx download nicht erfolgt
blinken abwechselnd		Bootloadersoftware läuft Wahrscheinliche Fehlerursache: Softwarereset nicht erfolgt
leuchten nicht		Firmware läuft
leuchtet (kurz)	leuchtet nicht	Firmware läuft und arbeitet Befehl ab

2.4 Lieferhinweise

USB / basicLIN 3072-Baugruppen werden wie folgt geliefert:

- ♦ 3x LIN Schnittstelle,
jeweils per Software als Master oder Slave konfigurierbar.



Jede Schnittstelle kann optional auch als KLine-Interface ausgeführt werden.

Außer der Auswahl der Schnittstelle selbst muss auch der Typ des zugehörigen LIN/ K-Line Transceivers für jede Schnittstelle festgelegt werden.

Für jede LIN/ K-Line-Schnittstelle sind außerdem die erforderlichen Funktionalitäten anzugeben).



Zum Betrieb von USB 3072 Boards ist das GÖPEL electronic USB-Rack erforderlich, das bis zu 16 GÖPEL electronic USB-Boards aufnehmen kann.

Die Stromversorgung erfolgt in diesem Fall über das eingebaute Netzteil.

3 Ansteuersoftware

Zur Einbindung der USB/ basicLIN3072-Hardware in eigene Applikationen existieren drei Möglichkeiten:

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

3.1 Programmieren über G-API

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

Sie finden alle benötigten Informationen im Ordner *G-API* der mitgelieferten CD.

3.2 Programmieren über DLL-Funktionen



Die Programmierung über DLL-Funktionen ist weiterhin für bestehende Projekte möglich, bei denen noch nicht mit der GÖPEL G-API gearbeitet werden kann.

Die Dokumentation *GÖPEL Firmware* senden wir Ihnen auf Anforderung gern zu. Bitte setzen Sie sich bei Bedarf mit unserem Vertrieb in Verbindung.



Der Begriff *GUSB_Platform* in der folgenden Funktionsbeschreibung steht jeweils für eine USB/ basicLIN 3072-Baugruppe.

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

3.2.1 Windows Device Treiber

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

- ◆ [Driver_Info](#)
- ◆ [DLL_Info](#)
- ◆ [Write_FIFO](#)
- ◆ [Read_FIFO](#)
- ◆ [Read_FIFO_Timeout](#)
- ◆ [Write_COMMAND](#)
- ◆ [Read_COMMAND](#)
- ◆ [Xilinx_Download](#)
- ◆ [Xilinx_Version](#)

Zuordnung der USB Controller zu einer USB 3072- bzw. basicLIN 3072-Baugruppe

Eine USB/ basicLIN 3072-Baugruppe erscheint mit drei USB Geräten im Gerätemanager von Windows, da jeder Controller (und somit jeder LIN Knoten) einen eigenen USB Controller besitzt (siehe Abbildung 1-1 im Abschnitt [Treiberinstallation](#)).

Um diese USB Geräte der/den USB/ basicLIN 3072-Baugruppe(n) und deren Controllern zuordnen zu können, sind zunächst die Seriennummern über den Befehl [Driver Info](#) zu ermitteln.

Die Zuordnung (u.a. die DeviceNumber) wird durch den Rest aus der ganzzahligen Division der Seriennummern durch die Zahl 4 bestimmt. D.h., die jeweilige Seriennummer ist durch 4 zu teilen, aber ohne Kommastellen (Modulo, mathematisches Formelzeichen `mod`).

Es gilt folgende Regel:

Seriennummer mod 4	Controller
0	1
1	2
2	3

Beispiel: zwei USB/ basicLIN 3072-Baugruppen mit jeweils drei Controllern:

Seriennummer	Controller	Baugruppe	DeviceNumber
20070060	1	1	1
20070061	2	1	2
20070062	3	1	3
20070064	1	2	4
20070065	2	2	5
20070066	3	2	6

3.2.1.1 *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`

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

Beschreibung:

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

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

Die Funktion füllt die Struktur, auf die `pDriverInfo` zeigt, 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 ist für die weitere Nutzung der USB-Hardware zwingend erforderlich.

3.2.1.2 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`.

Beispiel:

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

3.2.1.3 Write_FIFO Die Funktion `GUSB_Platform_Write_FIFO` dient zum Senden eines Befehls zum Controller.

Format:

```
int GUSB_Platform_Write_FIFO(unsigned int DeviceName,  
                             unsigned int DeviceNumber,  
                             t_USB_FIFO_Interface_Buffer *pWrite,  
                             unsigned int DataLength)
```

Parameter:

DeviceName

Typ des adressierten Gerätes (Nummer, die in `GUSB_Platform_def.h` deklariert ist, für USB/ basicLIN 3072 = 5)

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).



Beachten Sie bitte die Hinweise unter [Zuordnung der USB Controller ...](#) im Abschnitt [Windows Device Treiber](#): Die Anzahl der „Geräte“ und damit der DeviceNumbers entspricht der Anzahl der vorhandenen LIN Schnittstellen.

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

DataLength

Größe des Speicherbereiches, auf den `pWrite` zeigt, in Bytes
Die Daten bestehen aus Befehlskopf und Befehlsbytes
(z. Zt. max. 1024 Byte pro Befehl)

Beschreibung:

Die Funktion `GUSB_Platform_Write_FIFO` sendet einen Befehl zum Controller.

Die allgemeine Befehlsstruktur ist im Abschnitt [Allgemeines zur Firmware](#) der Dokumentation [GÖPEL Firmware](#) beschrieben.

3.2.1.4 Read_FIFO Die Funktion `GUSB_Platform_Read_FIFO` dient zum Lesen einer Antwort vom Controller.

Format:

```
int GUSB_Platform_Read_FIFO(unsigned int DeviceName,
                           unsigned int DeviceNumber,
                           t_USB_FIFO_Interface_Buffer *pRead,
                           unsigned int *DataLength)
```

Parameter:

DeviceName

Typ des adressierten Gerätes (Nummer, die in `GUSB_Platform_def.h` deklariert ist, für USB/ basicLIN 3072 = 5)

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).



Beachten Sie bitte die Hinweise unter Zuordnung der USB Controller ... im Abschnitt [Windows Device Treiber](#): Die Anzahl der „Geräte“ und damit der DeviceNumbers entspricht der Anzahl der vorhandenen LIN Schnittstellen.

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

Nach erfolgreicher Funktionsausführung befinden sich die Daten im Lesepuffer, bestehend aus Antwortkopf und Antwortbytes (z. Zt. max. 1024 Byte pro Antwort)

DataLength

Vor Funktionsaufruf: Anzugebende Größe des Lesepuffers in Bytes
Nach Funktionsausführung: Anzahl der tatsächlich gelesenen Bytes

Beschreibung:

Die Funktion `GUSB_Platform_Read_FIFO` liest die älteste vom Controller geschriebene Antwort. Ist während einer Timeout-Zeit von 100 ms (nicht einstellbar) keine Antwort empfangen worden, liefert die Funktion jedoch KEINEN Fehler zurück: In diesem Fall ist der Wert für die Anzahl der tatsächlich gelesenen Bytes = 0 !!!

3.2.1.5 *Read_FIFO_Timeout*

Die Funktion `GUSB_Platform_Read_FIFO_Timeout` dient zum Lesen einer Antwort vom Controller, wobei ein Timeout vorzugeben ist.

Format:

```
int GUSB_Platform_Read_FIFO_Timeout(unsigned int DeviceName,  
                                   unsigned int DeviceNumber,  
                                   t_USB_FIFO_Interface_Buffer *pRead,  
                                   unsigned int *DataLength,  
                                   unsigned int Timeout)
```

Parameter:

DeviceName

Typ des adressierten Gerätes (Nummer, die in `GUSB_Platform_def.h` deklariert ist, für USB/ basicLIN 3072 = 5)

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).



Beachten Sie bitte die Hinweise unter [Zuordnung der USB Controller ...](#) im Abschnitt [Windows Device Treiber](#): Die Anzahl der „Geräte“ und damit der DeviceNumbers entspricht der Anzahl der vorhandenen LIN Schnittstellen.

Zeiger, z.B. pRead auf den Lesepuffer

Nach erfolgreicher Funktionsausführung befinden sich die Daten im Lesepuffer, bestehend aus Antwortkopf und Antwortbytes (z. Zt. max. 1024 Byte pro Antwort)

DataLength

Vor Funktionsaufruf: Anzugebende Größe des Lesepuffers in Bytes
Nach Funktionsausführung: Anzahl der tatsächlich gelesenen Bytes

Timeout

Angabe in Millisekunden (Standardwert: 500)

Beschreibung:

Die Funktion `GUSB_Platform_Read_FIFO_Timeout` liest die älteste vom Controller geschriebene Antwort.
Ist während der einstellbaren Timeout-Zeit keine Antwort empfangen worden, liefert die Funktion jedoch KEINEN Fehler zurück: In diesem Fall ist der Wert für die Anzahl der tatsächlich gelesenen Bytes = 0 !!!

3.2.1.6 Write_ COMMAND

Die Funktion `GUSB_Platform_Write_COMMAND` dient zum Senden eines Configuration-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/ basicLIN 3072 = 5)

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).



Beachten Sie bitte die Hinweise unter [Zuordnung der USB Controller ...](#) im Abschnitt [Windows Device Treiber](#): Die Anzahl der „Geräte“ und damit der DeviceNumbers entspricht der Anzahl der vorhandenen LIN Schnittstellen.

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

DataLength

Größe des Speicherbereiches, auf den `pWrite` zeigt, in Bytes
Siehe auch [Steuerbefehle USB Controller](#)
(z. Zt. max. 64 Byte pro Befehl)

Beschreibung:

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

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

3.2.1.7 *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/ basicLIN 3072 = 5)

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).



Beachten Sie bitte die Hinweise unter [Zuordnung der USB Controller ...](#) im Abschnitt [Windows Device Treiber](#): Die Anzahl der „Geräte“ und damit der DeviceNumbers entspricht der Anzahl der vorhandenen LIN Schnittstellen.

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

Nach erfolgreicher Funktionsausführung befinden sich die Daten im Lesepuffer, bestehend aus Antwortkopf und Antwortbytes

Siehe auch [Steuerbefehle USB Controller](#)

(z. Zt. min. 64 Byte pro Antwort)

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 zurück.

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!

3.2.1.8 Xilinx_ Download

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

Diese Funktion kann nur auf dem USB Controller des ERSTEN Controllers einer USB/ basicLIN 3072- Baugruppe ausgeführt werden.

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/ basicLIN 3072 = 5)

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).



Beachten Sie bitte die Hinweise unter [Zuordnung der USB Controller ...](#) im Abschnitt [Windows Device Treiber](#): Die Anzahl der „Geräte“ und damit der DeviceNumbers entspricht der Anzahl der vorhandenen LIN Schnittstellen.

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.

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



Nach `Xilinx_Download` ist eine Wartezeit von ca. 500 ms erforderlich, da die Controller ein Power-On-Reset durchlaufen. Anschließend ist der Firmware-Befehl `0x10 Software Reset` auszuführen, um vom Bootloader-Modus in den Normal-Modus zu gelangen.

3.2.1.9 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/ basicLIN 3072 = 5).

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.

Beispiel:

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

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/ basicLIN 3072-Baugruppen 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/ basicLIN 3072-Baugruppen unter LabVIEW angesprochen werden können.

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

3.4 Weitere GÖPEL Software

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

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

3.5 Steuerbefehle USB Controller

Die USB Controller sind für die Anbindung der USB/ basicLIN 3072-Baugruppe an den PC über USB 2.0 zuständig.

An diese 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 USB Befehle
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 USB Befehle
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

C		S	
Controller		Steckverbinder	
Antwort	3-7, 3-8	Front	2-9
Befehl	3-6		
F		T	
Firmware	3-1	Transceiver	
		K-Line	2-7
		LIN	2-7
G		U	
G-API	3-1	USB 3072/ basicLIN 3072	
		Aufbau	2-5
		Ressourcen	2-3
		USB Antwortaufbau	3-14
		USB Befehle	3-14
		USB Befehlsaufbau	3-14
		USB Controller	
		Antwort	3-10
		Befehl	3-9
		Steuerbefehle	3-14
I		W	
Installation		Windows Treiber	3-2
Hardware	1-1		
Treiber	1-2		
L			
LabVIEW			
G-API	3-13		
Windows	3-13		
LED Anzeige	2-10		