

USB 3116

basicCON 3116

Relaisschaltmodule
Nutzerhandbuch Version 1.0

© 2011 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: 15.03.2011

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

Stand: März 2011

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	2-3
2.2.1	<i>Abmessungen</i>	2-3
2.2.2	<i>Kennwerte</i>	2-3
2.2.3	<i>Adressierung</i>	2-3
2.2.4	<i>Stromversorgung</i>	2-3
2.3	AUFBAU	2-4
2.3.1	<i>Allgemeines</i>	2-4
2.3.2	<i>Belegung Frontsteckverbinder</i>	2-5
3	ANSTEUERSOFTWARE	3-1
3.1	PROGRAMMIEREN ÜBER DLL-FUNKTIONEN	3-1
3.1.1	<i>Driver_Info</i>	3-2
3.1.2	<i>DLL_Info</i>	3-3
3.1.3	<i>Xilinx_Download</i>	3-4
3.1.4	<i>Xilinx_Version</i>	3-5
3.1.5	<i>Write_COMMAND</i>	3-6
3.1.6	<i>Read_COMMAND</i>	3-7
3.1.7	<i>SetRelay</i>	3-8
3.1.8	<i>GetRelay</i>	3-9
3.1.9	<i>UpdateRelay</i>	3-10
3.2	PROGRAMMIEREN MIT LABVIEW	3-11
3.3	STEUERBEFEHLE USB CONTROLLER	3-12
3.3.1	<i>USB Befehlsaufbau</i>	3-12
3.3.2	<i>USB Antwortaufbau</i>	3-12
3.3.3	<i>USB Befehle</i>	3-12

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 3116:

Das USB 3116-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!

Wählen Sie einen freien Steckplatz in Ihrem USB-Rack aus. Falls vorhanden, muss zuerst das Slotblech entfernt werden, das 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 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.



Bitte vergleichen Sie das Kapitel [Adressierung](#) zur Installation mehrerer USB 3116-Boards.

basicCON 3116:

Das basicCON 3116 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 das beigelegte Netzteil oder eine eigene Spannungsquelle an die entsprechenden Anschlüsse **ext. Power Supply** auf der Rückseite des Gehäuses an (siehe Kapitel [Stromversorgung](#)).



Die Anschlüsse auf der Frontseite der USB 3116-Boards bzw. basicCON 3116-Geräte 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 gegenwärtig ausschließlich Windows® 2000/ XP-Systeme!

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.



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

Durch die Plug-und-Play Fähigkeit von Windows® 2000/ XP 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 wurden.

Die folgende Abbildung zeigt die erfolgreiche Einbindung eines USB 3116-Boards bzw. eines basicCON 3116:

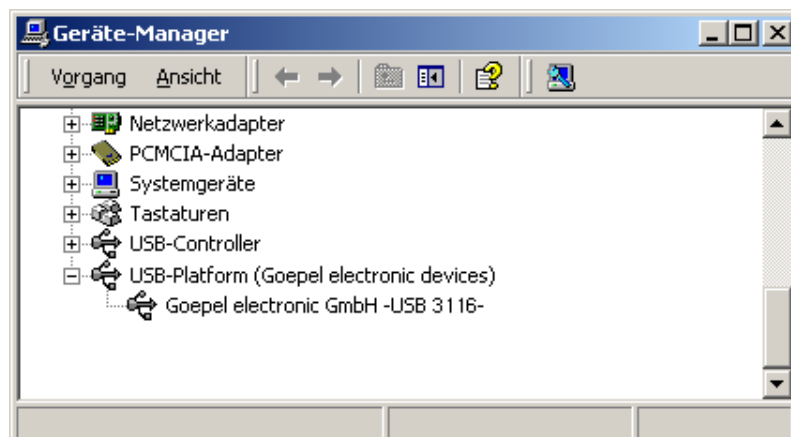


Abbildung 1-1:
Anzeige Geräte-Manager



Beachten Sie bitte, dass der Geräte-Manager ALLE USB-Controller anzeigt, die von diesem Treiber unterstützt werden.

2 Hardware

2.1 Bestimmung

USB 3116 ist ein Relaisboard mit USB 2.0-Interface der GÖPEL electronic GmbH.

Diese Boards werden in der Steuerungs- und Prüftechnik, u.a. in der Automobiltechnik, zum Schalten großer Lasten verwendet.

Das Board hat 16 voneinander unabhängige Relais mit Wechsler Charakteristik.



Abbildung 2-1:
Relaisboard USB 3116



Beachten Sie bitte, dass ein Download des Xilinx FPGAs für die Funktion des Boards unabdingbar ist (siehe [Xilinx Download](#) unter [Programmieren über DLL-Funktionen](#)).



Zum Betrieb von USB 3116 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 in diesem Fall über das im Rack eingebaute Netzteil.

basicCON 3116 ist ein GÖPEL electronic GmbH stand-alone Gerät auf der Grundlage eines USB 3116 Relaisboards zum Anschluss an einen PC oder Laptop, das für den eigenständigen Einsatz außerhalb komplexer Testsysteme entwickelt wurde.

Die externe Spannungszufuhr von 7-25 VDC erlaubt die Nutzung dieses Gerätes zum Schalten elektrischer Signale bei beliebigen Anwendungen.



Abbildung 2-2:
basicCON 3116

An der Geräterückseite befinden sich die folgenden Anschlüsse:



Abbildung 2-3:
Geräterückseite

- ◆ 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):

Abmessungen

- USB 3116: 4 TE x 130 mm x 185 mm
- basicCON 3116: 120 mm x 50 mm x 180 mm



Die Angaben für USB 3116 beziehen sich auf ein Board im GÖPEL electronic USB-Rack.

2.2.2 Kennwerte Ein Relaisboard USB 3116 hat folgende Kennwerte:

Symbol	Kennwert	Min.	Typ.	Max.	Einheit	Bedingung
I_S	Schaltstrom DC (Nom.)			5	A	Pro Relaiskontakt
U_S	Schaltspannung DC			230	V	Pro Relaiskontakt
P_S	Schaltleistung			240	W	Pro Relaiskontakt
R_{contact}	Kontaktwiderstand		6	30	m Ω	Pro Relaiskontakt
N	Schaltspiele (elektrisch)	10^5				
t_{on}	Anzugszeit		5	10	ms	
t_{off}	Abfallzeit		5	10	ms	
U_{EXT}	Externe Betriebsspannung	7	12	25	V	



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!

2.2.3 Adressierung

Die Adressierung von USB 3116-Boards im GÖPEL electronic USB-Rack oder basicCON 3116 Geräten bei Betrieb am gleichen Rechner erfolgt ausschließlich über die Seriennummern (siehe [Ansteuersoftware](#)): Das Board/ Gerät mit der KLEINSTEN Seriennummer hat immer die DeviceNumber 1.



Zur Erhöhung der Übersichtlichkeit empfehlen wir, im Falle mehrerer USB 3116-Boards diese in aufsteigender Reihenfolge ihrer Seriennummern im USB-Rack anzuordnen.

2.2.4 Stromversorgung

Das Board USB 3116 wird über das USB-Rack versorgt, in dem es installiert worden ist.

basicCON 3116 wird extern über die beiden Buchsen für ext. Power Supply (rot = plus/ blau = minus) mit 7-25VDC versorgt (max. 400mA bei 12V).

Stattdessen kann auch die Buchse für das beigelegte Steckernetzteil mit dem Hohlstecker (2,1 x 5,5mm/ Polarität + innen) genutzt werden, siehe Abbildung 2-3.

2.3 Aufbau

2.3.1 Allgemeines

USB 3116 ist zum potenzialfreien Schalten großer Lasten einsetzbar. Das Board verfügt über 16 Wechsler Relais.

Die Anschlüsse der jeweiligen Relais sind auf den Steckverbinder XS1 geführt.

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

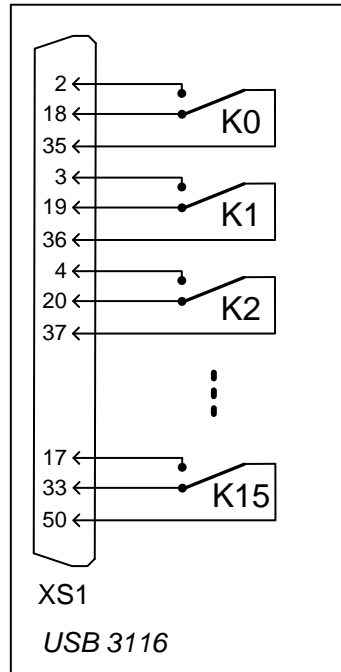


Abbildung 2-4:
Struktur USB 3116

Abbildung 2-4 zeigt eine schematische Darstellung von USB 3116.

Das stand-alone Gerät basicCON 3116 beruht auf dem Relais-Board USB 3116.

2.3.2 Belegung Front- steckverbinder

Typ am Gerät: DSub-Steckverbinder Steckerleiste 50-polig

Typ am Kabel: DSub-Steckverbinder Buchsenleiste 50-polig

Die Belegung des Frontsteckverbinders ist in der folgenden Tabelle dargestellt:

Nr.	Anschluss XS1	Signalname	Bemerkung
1	2	Ch0_NO	Channel0 - Kontakt „Normally Open“
2	35	Ch0_COM	Channel0 - Kontakt „Common“
3	18	Ch0_NC	Channel0 - Kontakt „Normally Closed“
4	3	Ch1_NO	Channel1 - Kontakt „Normally Open“
5	36	Ch1_COM	Channel1 - Kontakt „Common“
6	19	Ch1_NC	Channel1 - Kontakt „Normally Closed“
7	4	Ch2_NO	Channel2 - Kontakt „Normally Open“
8	37	Ch2_COM	Channel2 - Kontakt „Common“
9	20	Ch2_NC	Channel2 - Kontakt „Normally Closed“
10	5	Ch3_NO	Channel3 - Kontakt „Normally Open“
11	38	Ch3_COM	Channel3 - Kontakt „Common“
12	21	Ch3_NC	Channel3 - Kontakt „Normally Closed“
13	6	Ch4_NO	Channel4 - Kontakt „Normally Open“
14	39	Ch4_COM	Channel4 - Kontakt „Common“
15	22	Ch4_NC	Channel4 - Kontakt „Normally Closed“
16	7	Ch5_NO	Channel5 - Kontakt „Normally Open“
17	40	Ch5_COM	Channel5 - Kontakt „Common“
18	23	Ch5_NC	Channel5 - Kontakt „Normally Closed“
19	8	Ch6_NO	Channel6 - Kontakt „Normally Open“
20	41	Ch6_COM	Channel6 - Kontakt „Common“
21	24	Ch6_NC	Channel6 - Kontakt „Normally Closed“
22	9	Ch7_NO	Channel7 - Kontakt „Normally Open“
23	42	Ch7_COM	Channel7 - Kontakt „Common“
24	25	Ch7_NC	Channel7 - Kontakt „Normally Closed“
25	10	Ch8_NO	Channel8 - Kontakt „Normally Open“
26	43	Ch8_COM	Channel8 - Kontakt „Common“
27	26	Ch8_NC	Channel8 - Kontakt „Normally Closed“
28	11	Ch9_NO	Channel9 - Kontakt „Normally Open“
29	44	Ch9_COM	Channel9 - Kontakt „Common“
30	27	Ch9_NC	Channel9 - Kontakt „Normally Closed“
31	12	Ch10_NO	Channel10 - Kontakt „Normally Open“
32	45	Ch10_COM	Channel10 - Kontakt „Common“
33	28	Ch10_NC	Channel10 - Kontakt „Normally Closed“
34	13	Ch11_NO	Channel11 - Kontakt „Normally Open“
35	46	Ch11_COM	Channel11 - Kontakt „Common“
36	29	Ch11_NC	Channel11 - Kontakt „Normally Closed“
37	14	Ch12_NO	Channel12 - Kontakt „Normally Open“
38	47	Ch12_COM	Channel12 - Kontakt „Common“
39	30	Ch12_NC	Channel12 - Kontakt „Normally Closed“
40	15	Ch13_NO	Channel13 - Kontakt „Normally Open“
41	48	Ch13_COM	Channel13 - Kontakt „Common“
42	31	Ch13_NC	Channel13 - Kontakt „Normally Closed“
43	16	Ch14_NO	Channel14 - Kontakt „Normally Open“
44	49	Ch14_COM	Channel14 - Kontakt „Common“
45	32	Ch14_NC	Channel14 - Kontakt „Normally Closed“
46	17	Ch15_NO	Channel15 - Kontakt „Normally Open“
47	50	Ch15_COM	Channel15 - Kontakt „Common“
48	33	Ch15_NC	Channel15 - Kontakt „Normally Closed“

3 Ansteuersoftware

3.1 Programmieren über DLL-Funktionen

Mit den nachfolgend beschriebenen Funktionsaufrufen können USB 3116-Boards oder basicCON 3116-Geräte direkt aus diversen Hochsprachen angesprochen werden (VisualC++, CVI).



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` auf 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](#)
- ◆ [SetRelay](#)
- ◆ [GetRelay](#)
- ◆ [UpdateRelay](#)

3.1.1 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 (Speicherbereich)

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 eines Zeigers `pDriverInfo` übergeben werden. Mit Hilfe des Parameters `LengthInByte` prüft die Funktion intern den korrekt initialisierten Anwenderspeicher.

Die Funktion füllt die 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 sind für die weitere Nutzung der USB-Hardware zwingend erforderlich.

3.1.2 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.

3.1.3 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 3116 = 20)

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`).

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 `*.cdf`).



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

3.1.4 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 3116 = 20)

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.

3.1.5 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 3116 = 20)

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

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.

3.1.6 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 3116 = 20)

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 Lesebuffer

(Nach erfolgreicher Funktionsausführung befinden sich die Daten im Lesebuffer, bestehend aus Antwortkopf und Antwortbytes)

DataLength

Vor Funktionsaufruf: Anzugebende Größe des Lesebuffers in Bytes

Nach Funktionsausführung: Anzahl der tatsächlich gelesenen Bytes

Siehe auch [Steuerbefehle USB Controller](#)

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

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!

3.1.7 SetRelay

Die Funktion `GUSB_Platform_3116_SetRelay` konfiguriert die Sollkonfiguration der Relais des mit `DeviceNumber` indizierten USB 3116 Boards.

Format:

```
int GUSB_Platform_3116_SetRelay
(unsigned int DeviceNumber,
 unsigned int RelVal_1_16,
 unsigned int HandlingControl);
```

Parameter:

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`).

RelVal_1_16

Relaisbits 1..16 (32 Bit Wert, Bit-orientiert;
nur die 16 niederwertigen Bits werden berücksichtigt)

`HandlingControl` (siehe auch unter `Beschreibung` weiter unten)

32 Bit Wert, 0 = Init, 1 = Normal

Beschreibung:

Die Funktion führt bei Übergabe des Wertes 0 für `HandlingControl` eine Initialisierung des Board-internen Relaishandlings aus, das nach jedem Powerup des Boards vor dem ersten Schalten erfolgen muss. Der Parameter `RelVal_1_16` wird hierbei NICHT beachtet.

Mit `HandlingControl = 1` konfiguriert die Funktion die Sollstellung ALLER 16 Relais mittels eines 32 Bit-Wertes, bei dem nur die 16 niederwertigen Bits in Betracht gezogen werden. Jedes Relaisbit steht für die Sollstellung eines Relais. Zum Setzen des jeweiligen Relais muss das entsprechende Bit mit 1 und zum Rücksetzen des Relais mit 0 übergeben werden.



Das physikalische Schalten der Relais auf dem USB 3116 Board gemäß dieser Sollkonfiguration erfolgt mit dem Befehl [UpdateRelay](#).

3.1.8 GetRelay

Die Funktion `GUSB_Platform_3116_GetRelay` gibt die im Board programmierte Sollkonfiguration der Relais des mit `DeviceNumber` indizierten USB 3116 Boards zurück.

Format:

```
int GUSB_Platform_3116_GetRelay
(unsigned int DeviceNumber,
 unsigned int *RelVal);
```

Parameter:

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`).

RelVal

Zeiger auf eine Variable

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

Beschreibung:

Der Bitwert ergibt den aktuellen Inhalt des Sollkonfigurationsregisters.



Beachten Sie bitte, dass mit dieser Funktion NICHT die tatsächlichen physikalischen Zustände der Relais ermittelt werden.

3.1.9 Update-Relay

Die Funktion `GUSB_Platform_3116_UpdateRelay` schaltet die Relais 1..16 des mit `DeviceNumber` indizierten USB 3116 Boards wie in der Sollkonfiguration angegeben.

Format:

```
int GUSB_Platform_3116_UpdateRelay  
    (unsigned int DeviceNumber);
```

Parameter:

`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`).

Beschreibung:

Nach Ausführung dieser Funktion entspricht der tatsächliche Schaltzustand der Relais den Sollkonfigurationen.

Zum Setzen des jeweiligen Relais muss das entsprechende Bit der Sollkonfiguration auf 1 und zum Rücksetzen des Relais auf 0 stehen.

3.2 Programmieren mit LabVIEW

LLB unter Verwendung des Windows Device Treibers

In den Dateien *GUSB_Platform.llb* und *GUSB_Platform 3116.llb* der mitgelieferten CD befinden sich VIs, mit deren Hilfe USB 3116-Boards oder basicCON 3116-Geräte direkt unter LabVIEW angesprochen werden können.

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

3.3 Steuerbefehle USB Controller

Der USB Controller ist für die Anbindung der USB 3116 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.

3.3.1 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

3.3.2 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

3.3.3 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

A

Adressierung2-3

BbasicCON 3116
Anschlüsse2-2

DDeviceNumber2-3
DLL-Funktionen3-1

G

GÖPEL USB-Racks1-1

IInstallation
Hardware1-1
Treiber1-2

R

Ressourcen2-1

SSteckverbinder
Front2-5
Stromversorgung2-3

UUSB Antwortaufbau3-12
USB Befehle3-12
USB Befehlsaufbau3-12
USB Controller3-12
USB Kabel2-3

W

Windows Treiber3-1