

<b>DL601</b>  <b>NIM-Basismodul</b>
---

<b>1. FUNKTION .....</b>	<b>2</b>
1.1. DATENBLATT.....	2
1.1.1. Anwendung.....	2
1.1.2. Daten.....	2
1.1.3. Besonderheiten.....	2
1.1.4. Aufbau.....	2
1.1.5. Stromversorgung.....	2
1.2. BLOCKDIAGRAMM.....	3
1.3. BESCHREIBUNG.....	3
<b>2. BETRIEB .....</b>	<b>4</b>
2.1. FRONT.....	4
2.2. RÜCKSEITE.....	4
2.3. PROGRAMMIERUNG.....	5
2.3.1. Überblick.....	5
2.3.2. Allgemeine Befehle.....	5
2.3.3. Spezielle Befehle zur Ablaufsteuerung .....	6
2.3.4. Spezielle Befehle für FIFO-Datenspeichers DL643 TDC .....	7
<b>3. FERTIGUNG.....</b>	<b>8</b>

## 1. FUNKTION

### 1.1. Datenblatt

#### 1.1.1. Anwendung

Allgemeines Basismodul in NIM für DL6xx Anwenderkarten mit RS232 oder USB Interface.

#### 1.1.2. Daten

Parameter	
Anzahl von DL6xx Modulen	4

#### 1.1.3. Besonderheiten

Microcontroller für eigenständige Betriebsweise.

USB-Bus.

RS232-Bus fähig.

CAN-Bus (MP46).

#### 1.1.4. Aufbau

1/12 NIM.

MP35 oder MP46 Controller.

#### 1.1.5. Stromversorgung

Spannung	Strom	Leistung
+5V		
+6V		
-6V		
+12V		
-12V		
Gesamt		

## **1.2. Blockdiagramm**

## **1.3. Beschreibung**

Der Microcontroller stellt über einen CPLD ein Interface zu beliebigen Anwenderkarten aus der DL6xx-Familie dar.

Als Interface nach außen können verschiedene serielle Busse verwendet werden. Standard ist der Universal Serial Bus (USB 1.0), es kann aber auch noch eine normale serielle Schnittstelle oder der CAN-Bus implementiert werden.

Durch allgemeine Befehle können alle Karten (Submodule) beschrieben und ausgelesen werden.

Zusätzlich sind, je nach Anwendung, aber auch spezielle Verarbeitungen und Befehle im Microcontroller implementierbar. Damit kann gegebenenfalls durch Vorverarbeitung und autonome Steuerung die serielle Schnittstelle entlastet werden.

## 2. BETRIEB

### 2.1. Front

Die Front bestimmt sich durch die individuellen Anwenderkarten aus der DL6xx-Familie.

### 2.2. Rückseite

Auf der Rückseite des Moduls befinden sich folgende Anschlüsse:

**NIM-Power:** zur Aufnahme im NIM-Überrahmen.

**USB-Buchse:** für den seriellen USB-Bus

*Alternativ:*

**DB9-Buchse RS232:** Alle 4 Ratemeter werden über die eingebaute serielle Schnittstelle programmiert bzw. ausgelesen.  
Die Schnittstelle wird standard mit 9600 Baud, 8 Datenbits, 2 Stopbits, NoParity betrieben.

*oder*

**DB9-Stecker CAN:** Bei Bestückung mit dem entsprechenden Controller-Modul (MP46) kann hier auch über den CAN Feldbus das Modul angesprochen werden.

**LEMO-Buchse 1:** derzeit noch nicht verwendet!

**LEMO-Buchse 2:** derzeit noch nicht verwendet!

**LEMO-Buchse 3:** derzeit noch nicht verwendet!

## 2.3. Programmierung

### 2.3.1. Überblick

Bei RS232 ist das Protokoll 9600Baud, 8 Bit, NoParity, 1 StopBit.

Mit USB kann ein entsprechender Treiber von FTDI eingesetzt werden.

Mit dem Befehl '?' kann jederzeit über die serielle Schnittstelle eine Übersicht der verfügbaren Kommandos abgefrufen werden.

```

-----
Basismodul DL604 vw020804
#1
Physik.Inst., Uni HD: vWalter, Back, Rausch
-----
?          Help
I n/i      Interrupt 1=Enable/0=Disable/Get
V n/v      Verbose 1=Enable/0=Disable/Get

S/s        Reset/Get Status (I3,I2,I1,I0,S3,S2,S1,S0)
M n/m      Module (0..3) Set/Get
A n/a      Address (0..15) Set/Get
D n/d      Write data (0..65535)/Read data

F n/f      FIFO read/read number of bytes in FIFO

R n/r n    Run List/Edit List @ line n
L/l        List New/Show
!          List Commands
-----

```

Alle Kommandos sind zeilenorientiert und werden zunächst über einen Eingabepuffer von max. 64 Zeichen gesammelt. Jedes Zeichen wird dabei grundsätzlich bei der Eingabe zurückgesendet (Echo).

Mehrere Befehle können immer auch mit Komma getrennt in eine Zeile zusammengefasst werden.

Die Ausführung wird jeweils mit <CR> gestartet. In der folgenden Einzelbeschreibung ist dieses CR nicht mehr angegeben!

Alle zurückgesendeten Daten sind ebenfalls mit CR abgeschlossen (Ausnahme F/f).

### 2.3.2. Allgemeine Befehle

- ? Liefert ein Kurz-Liste der möglichen Befehle (siehe oben).
- I n Damit wird beim Controller das Interrupt-Monitoring ein- (n=1) oder aus- (n=0) geschaltet. D.h. falls in irgendeinem Modul ein Interrupt ausgelöst wird, wird dies auch über die serielle Schnittstelle mitgeteilt, es erscheint die Meldung „INTERRUPT: i“! i ist eine Zahl, die binär angibt, von welchem Submodul der Interrupt kommt (z.B. i = 5: Modul 0 und Modul 3).

- i Liefert die aktuelle Einstellung von I zurück (0/1).
- V n Bei n=1 (default) wird bei der Ausgabe u.U. zusätzliche Information zur besseren Lesbarkeit ausgegeben (Betrieb mit Terminal!). Bei n=0 werden lediglich die relevanten Daten ausgegeben (Betrieb mit Computer).
- v Liefert die aktuelle Einstellung von V zurück (0/1).
- S Generiert auf allen Submodulen einen Hardware-Reset.
- s Liefert über die serielle Schnittstelle Informationen über die Status-Leitungen von den einzelnen Submodulen. Dies ist wieder binär kodiert (z.B. i = 6: Modul 1 und Modul 3).
- M n Setzt die Adressierung auf das Submodul n=0..3. Alle nachfolgenden Schreib- und Lesebefehle beziehen sich dann auf dieses Modul.
- m Liefert die aktuelle Einstellung von M zurück (0..3).
- A n Setzt die Adressierung für die Subadresse (im aktuellen Submodul) auf entsprechend n=0..15. Alle nachfolgenden Schreib- und Lesebefehle beziehen sich dann auf diese Subadresse.
- a Liefert die aktuelle Einstellung von A zurück (0..15).
- D n Hier wird ein Schreibvorgang auf das aktuelle Modul und die aktuelle Subadresse ausgeführt. n ist ein 16 Bit Wort (0.. 65535).
- d Hier wird ein Lesevorgang vom aktuellen Modul und der aktuellen Subadresse ausgeführt. Über die serielle Schnittstelle wird das gelesene 16 Bit Wort (0.. 65535) gesendet.

### 2.3.3. Spezielle Befehle zur Ablaufsteuerung

Für eine autonome Ablaufsteuerung können bestimmte Befehle in einer Liste aufgeführt werden, die wie ein Programm (Skript) ausgeführt werden kann. Diese Liste kann direkt am Terminal eingegeben oder über den PC geladen werden. Sie kann dann auch fest im Controller abgespeichert werden.

- R n Die Abarbeitung der Liste wird an der Zeile n gestartet. Damit können auch gleichzeitig mehrere Programme abgespeichert und individuell gestartet werden!
- r n Hier kann eine Zeile n mit einer neuen Anweisung ersetzt werden.
- L Die Liste wird komplett gelöscht.
- l Die komplette Liste wird ausgegeben.
- ! Zeigt eine Übersicht über die verwendbaren Anweisungen in einer Liste, wie folgt:

-----  
List Commands:

```

S          Reset all Submodules
s n       Send Status (I3,I2,I1,I0,S3,S2,S1,S0) & add n
A n       Set Address with n (0..15)
a n       Add n to Address
M n       Set Moduleaddress with n (0..3)
m n       Add n to Moduleaddress
D n       Write Data n (0..65535)
d n       Read Data, send & add n

```

```

& n      AND n
# n      OR n
< n      if < 0 (>32767) jump to line n
> n      if > 0 jump to line n
= n      if = 0 jump to line n
: n      Jump to line n
.        STOP
N n      Set Counter with n
n n      Add n to Counter
" n      Wait n seconds
*        Send result

```

-----

### 2.3.4. Spezielle Befehle für FIFO-Datenspeichers DL643 TDC

Der Steckplatz des TDC-Moduls muss durch den Befehl „M n“ eingestellt werden!

- f Hier wird lediglich die Anzahl der Hits im FIFO-Speicher ohne Auslesen angezeigt. Das Format ist in allen Fällen in ASCII-Darstellung.
- F Hier werden die TDC-Hits aus dem FIFO ausgelesen. Das Format ist je nach Einstellung „Verbose“ unterschiedlich:

**V1: Die Ausgabe erfolgt in ASCII (lesbar):**

Modul=m FIFO=n <CR>  
 1. HitPattern & Zeitangabe <CR>  
 ...  
 n. HitPattern & Zeitangabe <CR>

z.B.  
 M=0 FIFO= 2  
 3 1002  
 2 2223

**V0: Die Ausgabe erfolgt in binär {Bytes} (ohne <CR>):**

{n LOW}{n HIGH}  
 {1. HitPattern}{Zeit LOW}{Zeit HIGH}  
 ...  
 {n. HitPattern}{Zeit LOW}{Zeit HIGH}

### 3. FERTIGUNG