

<h1 style="margin: 0;">DL603</h1> <h2 style="margin: 0;">4-fach Ratemeter</h2>
--

1. FUNKTION	2
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
2. BETRIEB	4
2.1. FRONT.....	4
2.1.1. Inputs.....	4
2.1.2. Outputs.....	4
2.2. RÜCKSEITE.....	4
2.3. PROGRAMMIERUNG.....	5
2.3.1. Allgemein.....	5
2.3.2. Befehle.....	5
2.3.3. CMD-Register (nur Schreiben).....	6
2.3.4. Status-Register (nur Lesen).....	6
3. FERTIGUNG	8
3.1. DL636.....	8
3.1.1. Module auf Steckplatz 0..2.....	8
3.1.2. Modul auf Steckplatz 3.....	8
3.2. DL601.....	8

1. FUNKTION

1.1. Datenblatt

1.1.1. Anwendung

4 fach digitaler Gategenerator mit Pulszähler (Ratemeter).
Ermittlung von Detektorzählraten in weiten Bereichen

1.1.2. Daten

Parameter	
Clock	10MHz (<80Mhz)
Zeitähler	32 Bit
Zeitbereich	100ns..429s
Zeitgenauigkeit (Quartz)	ca. 10e-8
Pulsrate	<80MHz
Pulsdauer	>5ns
Pulszähler	32 Bit

1.1.3. Besonderheiten

Microcontroller für eigenständige Betriebsweise und einfaches Interface über RS232.
Funktion durch CPLD programmierbar.
RS232-Bus fähig.
CAN-Bus (MP46).

1.1.4. Aufbau

1/12 NIM.
MP35 oder MP46 Controller.
A351 Controller.
DL601 Interface.
4 * DL636 CPLD.

1.1.5. Stromversorgung

Spannung	Strom	Leistung
+5V	2A	10W
-5.2V	1A	5W
+12V		
-12V		
Gesamt		15W

1.2. Blockdiagramm

1.3. Beschreibung

Jeweils ein Ratemeter wird in einem **DL636** CPLD Board realisiert.

Mit dem **Trigger** wird ein 32 Bit **Zeitzähler** zunächst mit dem vorprogrammiertem Wert aus einem **Gate-Register** vorgeladen und anschließend mit der zentralen System-Clock (10MHz) bis auf 0 heruntergezählt. Während dieser Zeit wird ein **Gate-Impuls** erzeugt.

Ein weiterer interner 32 Bit **Pulszähler** zählt während dieser Zeit eventuell anliegende **Pulse** und kann anschließend ausgelesen werden.

Durch Anschluß des **Inverse Gate**-Signals an den **Trigger** eines anderen Moduls im gleichen Gerät kann dort ein synchroner Start generiert werden. Somit kann auch ein komplexerer Messablauf mit den 4 identischen Submodulen aufgebaut werden.

Unmittelbar vor Ende der Gate-Zeit wird noch ein **Carry-Impuls** erzeugt, der das um 50ns überlappende Starten weiterer Ratemeter erlaubt.

Der **Trigger** ist während der Gate-Zeit gesperrt. Der Gategenerator ist somit nicht retriggerbar.

Verschiedene zusätzliche Betriebsarten bestimmen die Funktionsweise **nach** Ablauf der Gate-Zeit:

Mit der Betriebsart **Resume** wird der Puls-Zähler bei einem erneuten Starten nicht auf 0 gesetzt, sondern zählt weiter. Damit können Zählraten über mehrere Gate-Zeiten aufsummiert werden.

In der Betriebsart **Single** wird der Trigger jeweils für folgende Ereignisse zunächst gesperrt. Über den Status kann dann der Ablauf der Gatezeit festgestellt, die Daten ausgelesen und anschließend der Trigger wieder eingeschaltet werden. Somit gehen hier keine Daten verloren. Ansonsten führt jeder folgende Trigger automatisch wieder zum erneuten Start des Gategenerators mit u.U. Rücksetzen des Puls-Zählers.

1.4. Asynchroner Trigger

In der Regel ist das externe Trigger-Signal nicht mit der internen Systemclock synchron. Durch die Synchronisation ergibt sich damit ein Zeitjitter für die Dauer des Gates von 1 Periode der Systemclock (hier 100ns). Eine eingestellte Gatezeit von z.B. "10" führt somit zu einer Gatedauer von 900..1000 ns.

ACHTUNG: Für asynchrone Trigger ist die Gatezeit =1 NICHT erlaubt!

Dieses Problem ist bei einem synchronen Trigger, wie z.B. bei Verkettung im gleichen Modul nicht gegeben und somit alle Zeiten erlaubt.

2. BETRIEB

2.1. Front

Jedes der vier Ratemeter ist identisch aufgebaut und besitzt folgende Ein- und Ausgänge.

2.1.1. Inputs

NIM: aktiv = -16mA@50 Ohm, entsprechend -0.8V

Trigger: NIM, Start bei Flanke aktiv.

Puls: NIM, Zählung bei Flanke aktiv.

2.1.2. Outputs

NIM: aktiv = -16mA, entsprechend **-0.8V@50 Ohm**

Gate: NIM, Zeitablauf bei Zustand aktiv.

_Gate: NIM, Zeitablauf bei Zustand nicht aktiv.

Carry: Zeit-Zähler=1 bei Zustand aktiv.

2.2. Rückseite

Auf der Rückseite des Moduls befinden sich folgende Anschlüsse:

NIM-Power: zur Aufnahme im NIM-Überrahmen.

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.

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. Allgemein

Mit dem Befehl '?' kann jederzeit über RS232 eine Übersicht der verfügbaren Kommandos abgefrufen werden.

```
-----
4* GateGenerator & Counter: DL603_5 vw171201
#1
Physik.Inst., Uni HD: vWalter, Rausch
-----
?          Help (n=1..4, 0=all)
C n,v/c n  Cmd (3bit) n Set/Get
G n,v/g n  Gate (32bit) n Set/Get
P n/p n    Pulsecounter (32bit) n Clear/Get
R n,v      Reset n (1=Cmd,2=Gate,4=Time,8=Puls,16=Run)
S/s        reSet/Status
t n        Trigger n
-----
```

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).

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.

Falls die Kanalnummer n=0 eingegeben wird, gilt das Kommando für alle Ratemetermodule von 1..4.

2.3.2. Befehle

- ? Liefert ein Kurz-Liste der möglichen Befehle
- Cn,v Das Ratemeter n wird mit dem Kommando (Cmdregister) v programmiert. Siehe im folgenden unter "Cmd" die Bedeutung der Bits!
- cn Der Status (Statusregister) des Ratemeter n wird ausgelesen. Siehe im folgenden unter "Status" die Bedeutung der Bits!
- Gn,v Das Gateregister des Ratemeters n wird mit dem Wert v geladen.
z.B. "G1,10000" : Für das Ratemeter 1 wird die Gatezeit 1ms eingestellt.
ACHTUNG: der Wert 0 bedeutet max. Gatezeit (=4294967295).
- gn Das Gateregister des Ratemeters n wird ausgelesen.
z.B. "g1" : Der Wert des Ratemeters 1 wird zurückgelesen (i.B.10000).
- Pn Der Puls-Counters des Ratemeters n wird auf 0 zurückgesetzt. Alle anderen Register bzw. Funktionen sind davon unberührt.
- pn Der Wert des Puls-Counters des Ratemeters n wird gelesen.
ACHTUNG: Dieser Wert sollte nicht während der Laufzeit des Gates ausgelesen werden! Erst nach Ablauf des Gates ist dieser Wert sicher verfügbar!
z.B.: "p4" gibt den Wert des Ratemeters 4 zurück.
- Rn,v Setzt selektiv durch v bestimmte Register bzw. Funktionen des Ratemeters n auf 0 bzw. auf STOP. Durch Kombination (Addition) können auch mehrere

oder alle Funktionen ausgewählt werden (z.B. v=31: löscht alles):
 v=1: Das Cmd-Register wird auf 0 gesetzt.
 v=2: Das Gate-Register wird auf 0 gesetzt.
 v=4: Das Timecounter-Register wird auf 0 gesetzt.
 v=8: Das Pulscounter-Register wird auf 0 gesetzt.
 v=16: Das RUN-FF wird auf STOP gesetzt.

- S Setzt ALLE Ratemeter in einen Anfangszustand mit allen Registern=0.
Sinnvoll nach dem Einschalten bzw. bei Fehlfunktion!
- s Liefert einen Überblick über die momentane Funktion ALLER 4 Ratemeter.
Zurückgelesen wird ein 8 Bit Wort mit folgender Bedeutung der einzelnen Bits:
 D0: Gate Ratemeter 1
 D1: Gate Ratemeter 2
 D2: Gate Ratemeter 3
 D3: Gate Ratemeter 4
 D4: /Enable Ratemeter 1
 D5: /Enable Ratemeter 2
 D6: /Enable Ratemeter 3
 D7: /Enable Ratemeter 4
ACHTUNG: Bits 4..7 sind aus technischen Gründen **invertiert!**
- tn Löst einen Trigger an Ratemeter n aus und startet gegebenenfalls den
Messzyklus (gleichbedeutend mit externem Trigger!).
ACHTUNG: Externer Trigger darf nicht aktiv sein!

2.3.3. CMD-Register (nur Schreiben)

Mit diesem Register kann in jedem Ratemeter unabhängig die Funktion eingestellt werden. Zur Zeit sind 4 Bits definiert:

- D0: **Enable** Das Setzen dieses Bits erlaubt das Auslösen eines erneuten Messzyklus durch einen Trigger.
ACHTUNG: Dieses Bit wird auch durch die Hardware zurückgesetzt, wenn das Single-Bit gesetzt ist und muß dann jeweils immer wieder vor einem neuen Trigger gesetzt werden.
- D1: **Single** Mit Setzen dieses Bits wird nur ein einziger Trigger ermöglicht. Nach Ablauf der Gate-Zeit wird das Enable-Bit gelöscht und der Benutzer kann anschließend ungefährdet die Pulszahl auslesen bzw. eine neue Gate-Zeit setzen.
- D2: **Resume** Normalerweise wird der Puls-Zähler mit jedem Trigger auf 0 zurückgesetzt! Falls dieses Bit gesetzt ist, erfolgt dies nicht und der Pulszähler zählt mit jedem Puls weiter. Damit kann über mehrere Gate-Zeiten aufsummiert werden.
- D3: **CCount** Mit Setzen dieses Bits zählt der Puls-Zähler die Anzahl der ausgelösten Gates (Trigger)! Damit der Zähler weiterzählt muß hier auch Resume eingeschaltet sein!

2.3.4. Status-Register (nur Lesen)

Äquivalent zum Cmd-Register kann hier die Funktion oder Status jedes Ratemeters ausgelesen werden:

- D0: **Enable** Hier wird entsprechend (falls nicht bereits per Software gelöscht) abgeprüft, ob im Single-Betrieb die Gate-Zeit abgelaufen ist und gültige Daten im Puls-Zähler stehen.

- D1: **Single** Lesen der Betriebsart "Single".
D2: **Resume** Lesen der Betriebsart "Resume".
D3: **CCount** Lesen der Betriebsart "Carry-Count".

3. FERTIGUNG

3.1. DL636

3.1.1. Module auf Steckplatz 0..2

- Quarz nicht bestücken.
- Jumper J1.1-2 setzen.
- Jumper J2.2-3 setzen.
- Jumper J3.2-3 setzen.
- Jumper J11, J24, J25 setzen.

3.1.2. Modul auf Steckplatz 3

- Quarz bestücken (10 MHz).
- Jumper J1.1-2 setzen.
- Jumper J2.2-3 setzen.
- Jumper J3.2-3 setzen.
- Jumper J11, J24, J25 setzen.
- Jumper J21, J18 setzen.

3.2. DL601

- Alle LDO-Regler bestücken.
- Jumper J101 setzen