

# **Probus V**

## **Befehlsreferenz**

**Grundmodul  
ADDAT30 mit Firmware PIC0162 V4.0  
Beschreibungsversion V2.22**

## Inhalt

1	Übersicht	4
1.1	Software-Schnittstelle zum Kunden	4
1.2	Kunden-Interface Ausführungen	4
1.2.1	RS-232 Schnittstelle, Standardausführung	4
1.2.2	RS-485 Schnittstelle, Standardausführung	4
1.2.3	USB Schnittstelle, Standardausführung	4
1.2.4	Ethernet, LAN, TCP/IP	4
1.2.5	IEEE-488, Standardausführung	4
1.2.6	PROFIBUS-DP, Standardausführung	5
1.2.7	Externe Lichtleiter	5
1.3	Mehrere ADDA Teile im System	5
1.3.1	Serielle LWL Verbindung	5
1.3.2	Parallele Ansteuerung	5
1.4	Konfigurationsschalter der Baugruppe ADDAT30 (DIP-Schalter)	6
2	Befehlssatz	7
3	Registerbefehle	8
3.1	Allgemeines	8
3.1.1	Befehlsform ohne Adressierung	8
3.1.2	Befehlsform mit Adressierung	9
3.1.3	Rücklesedaten ohne Adressierung	10
3.1.4	Rücklesedaten mit Adressierung	11
3.2	Registersatz	12
3.2.1	Sollwert Steuerung	12
3.2.2	Digitale Ausgänge	16
3.2.3	analoge Monitor-Register	18
3.2.4	Digitale Eingänge	19
3.2.5	Konfigurations- und Statuswerte	20
3.2.6	Kalibrationswerte	21
3.2.7	Virtuelle Register	24
3.2.8	gemeinsames Setzen und Lesen mehrerer Register	25
3.3	Service Request , Automatische Übertragung bei Änderungen	28
3.4	Checksumme	29
3.4.1	Checksumme Typ1	29
4	Sonderbefehle und Probus IV kompatible Befehle	30
4.1	Identifikationsstring ausgeben	30
4.2	Device Clear	30
4.3	Ausgangsspannung ein-/ausschalten	31
4.4	Spannungssollwert setzen	31
4.5	Stromsollwert setzen	31
4.6	Rücklesedaten für getriggerte Datenabfrage vorwählen	31
4.7	getriggerte Datenabfrage	32
4.8	AD-Wandlungsrate und Auflösung einstellen	32
4.9	Ausgangspolarität einstellen	32
4.10	X-Ausgänge ein-/ausschalten	33
4.11	Execute-on-X ein-/ausschalten	33
4.12	Execute on X	34
4.13	Set Terminator	34
4.14	SRQ-Maske setzen	34
4.15	Sonderbefehle	35
4.15.1	Sonderbefehl SRQ-Maske setzen	35
4.15.2	Sonderbefehl Datenabfrage	35

---

4.15.3	Sonderbefehl Trigger-on-Talk	35
5	Fehlercodes	36
6	Technische Daten	37

# 1 Übersicht

Die Baugruppe ADDAT30 (kurz "ADDA") ist ein AD/DA-Interface zur Ansteuerung von Netzgeräten über Lichtleiter mit serieller Datenübertragung.

Sie sitzt als SMD-Tochterplatine direkt auf der Geräte-Elektronik.

Mehrere ADDAs können über eine LWL-Kette in Reihe geschaltet werden. Durch eine Sub-Adressierung ist es möglich, z.B. über USB mehrere ADDAs in einem Gerät anzusprechen.

## 1.1 Software-Schnittstelle zum Kunden

Über die serielle Schnittstelle werden alle Befehle und Rücklesedaten als lesbare ASCII Zeichen übertragen.

Es sind verschiedene Befehlsgruppen implementiert:

- Probus IV kompatible Befehle (weitestgehend IBIG40-kompatibel)
- erweiterter Register-Befehlssatz für einheitlichen Zugriff auf alle internen Funktionen
- spezielle Befehle für die Kommunikation mit dem IEEE-488/seriell Umsetzer IBIG50

## 1.2 Kunden-Interface Ausführungen

Die Anbindung zum Kunden erfolgt über einen Interface Konverter, der die Umwandlung vom Kundeninterface auf den seriellen Datenstrom über Lichtleiter vornimmt.

Einige Ausführungsbeispiele:

### 1.2.1 RS-232 Schnittstelle, Standardausführung

An der Interface-Platte am Netzgerät sitzt ein aktiver Umsetzer von LWL auf RS-232 mit 9-poligem Sub-D Stecker.

### 1.2.2 RS-485 Schnittstelle, Standardausführung

An der Interface-Platte am Netzgerät sitzt ein aktiver Umsetzer von LWL auf RS-485 mit 9-poligem Sub-D Stecker.

### 1.2.3 USB Schnittstelle, Standardausführung

An der Interface-Platte am Netzgerät sitzt ein aktiver Umsetzer von USB auf LWL.

Dieser Umsetzer arbeitet zusammen mit einer Treibersoftware im PC als virtueller COM-Port, so dass das Netzgerät ohne USB-Kenntnisse mit vorhandener Software angesprochen werden kann.

### 1.2.4 Ethernet, LAN, TCP/IP

An der Interface-Platte am Netzgerät sitzt ein aktiver Umsetzer von Ethernet auf LWL.

Dieser Umsetzer arbeitet zusammen mit einer Treibersoftware im PC als virtueller COM-Port, so dass das Netzgerät ohne umfangreiche Netzwerk-Kenntnisse mit vorhandener Software angesprochen werden kann.

### 1.2.5 IEEE-488, Standardausführung

An der Interface-Platte am Netzgerät sitzt ein Umsetzer von LWL auf IEEE-488 (GPIB).

Der Umsetzer bietet zwei Betriebsmodi:

Mode 1 arbeitet wie der bisherige Probus IV mit Datenausgabe per Trigger-on-talk. In diesem Modus ist das Interfacesystem sowie die Befehle zur Steuerung des Netzgerätes kompatibel zum bisherigen Probus IV.

Da hier der Funktionsumfang stark eingeschränkt ist, sollte dieser Modus nicht für neue Softwareprojekte verwendet werden.

Mode 2 bietet ein einheitliches Handshake-Verfahren, mit dem sich die volle Leistungsfähigkeit des Probus V Systems nutzen lässt.

Der DIP-Schalter zur IEEE-Einstellung ist an der Rückseite des Netzgerätes zugänglich.

## 1.2.6 PROFIBUS-DP, Standardausführung

An der Interface-Platte am Netzgerät sitzt ein Umsetzer von LWL auf PROFIBUS-DP. Dieser sendet den Eingangsdatenblock der Profibus-DP Seite zyklisch an die ADDA. (Siehe unten, Beschreibung Register >H0)

## 1.2.7 Externe Lichtleiter

Die oben erwähnten Interface Konverter können auch außerhalb des Netzgerätes nahe am steuernden PC oder SPS platziert werden.

Die Verbindung zwischen Netzgerät und PC erfolgt dann direkt über Lichtleiter bis zu Entfernungen von 20m. Längere Distanzen bis zu einigen 100m sind bei Bedarf realisierbar.

Durch die Lichtleiter Verbindung ist höchste Störsicherheit erreichbar.

Als Lichtleiterverbinder kommt standardmäßig die HFBR-0500 Serie von Agilent (HP) zum Einsatz.

## 1.3 Mehrere ADDA Teile im System

Diese Anwendung trifft zur für Sondergeräte mit speziellen Funktionen zu. Beachten Sie hierzu die dem Sondergerät beiliegende Dokumentation der Funktionen.

### 1.3.1 Serielle LWL Verbindung

Es ist möglich, mehrere ADDAs über LWL seriell in Reihe zu schalten. Damit lassen sich dann auch komplexere Geräte über nur einen Anschluss von der Kundenseite aus ansteuern. (jedoch nicht möglich mit Profibus-DP Interface Konverter)

Die Unterscheidung der verschiedenen ADDAs erfolgt über eine Adresse, die im EEPROM der jeweiligen ADDA eingestellt ist.

Achtung: nicht verwechseln mit der GPIB Primäradresse!

Die letzte ADDA in der LWL-Kette muss stets auf die Adresse 0 eingestellt sein.

Für die meisten Befehle ist dann eine vorangestellte Adresse in der Form "#n" nötig.

Bestimmte Befehle sind jedoch weiterhin unadressiert, z.B. der Device-Clear-Befehl "=", der dann auf alle ADDAs in der Kette wirkt.

Näheres siehe Befehlsbeschreibung.

### 1.3.2 Parallele Ansteuerung

Für Sonderfälle können bis zu 256 ADDAs parallel betrieben werden.

Der Betrieb erfolgt entweder an einem gemeinsamen RS-485 Bus oder über LWL-Verteiler in Sondergeräten.

Im Gegensatz zur LWL-Reihenschaltung (1.3.1) werden bei paralleler Adressierung Befehle mit fremder Adresse von der jeweiligen ADDA ignoriert und nicht weitergeleitet.

(Siehe auch Kalibrierregister >CPAR.)

## 1.4 Konfigurationsschalter der Baugruppe ADDAT30 (DIP-Schalter)

<b>Baudrate</b>	
<p>9600 Bd</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>	<p>38400 Bd</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>
<p>625 kBd</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>	<p>Spez. Baudrate (Register &gt;CBAUD)</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>
<b>Kalibrierdaten-Schutz</b>	
<p>Kalibrierdaten geschützt</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>	<p>Kalibrierdaten änderbar</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>
<b>Adress-Modus</b>	
<p>ADDA ist im Standard-Modus, IBIG40 kompatibel</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>	<p>ADDA ist im adressierbaren Modus (Adresse im EEPROM)</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>
<b>Betriebsart</b>	
<p>ADDA ist bei SEL-D aktiv (Modus für Digitalprogrammierung)</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>	<p>ADDA ist bei SEL-A aktiv (Modus für LWL-Analogprogrammierung)</p> <p>0 OFF 1 ON</p> <p>1 2 3 4 5 6</p>

## 2 Befehlssatz

- Jeder Befehl muss mit mindestens einem Terminatorzeichen abgeschlossen werden. Zulässig sind: CR, LF oder 0x00 oder jede beliebige Kombination davon.
- Nach jedem Befehl sendet die ADDA genau einen Antwortstring zurück.
- "leere" Befehlsstrings, d.h. Strings die nur aus Terminator-Zeichen bestehen, werden verworfen und liefern auch keinen Antwortstring
- Alle Lesedaten und Handshake-Strings von der ADDA werden mit dem eingestellten Terminator abgeschlossen (siehe Register ">KT" bzw. ">CKT" sowie Y-Befehl)
- **In den weiteren Beschreibungen der Befehle wird der Terminator zur besseren Übersicht weggelassen.**
- Groß- und Kleinschreibung wird an keiner Stelle im gesamten Befehlssatz unterschieden
- Receive- Timeout:  
Wenn länger als 5000ms kein Zeichen an die ADDA übermittelt wurde, werden die bisher empfangenen Zeichen eines evtl. angefangenen Befehlsstrings verworfen.  
Durch das relativ lange Timeout ist es auch möglich einzelne Befehle per Hand mit einem Terminalprogramm (z.B. Hyperterminal) zu übermitteln.
- Befehlslänge: jeder einzelne Befehl darf eine maximale Länge von 50 Zeichen nicht überschreiten.
- Empfangspuffer:  
Das Interface hat einen 255 Zeichen großen FIFO-Empfangspuffer.
- **Checksumme:** für kritische Anwendungen kann die Übertragung mit einer Checksumme abgesichert werden. Siehe dazu Abschnitt 3.4 und die Beschreibung des Registers >CCS im Abschnitt 3.2.6.

## 3 Registerbefehle

In der ADDA sind alle wesentlichen Einstellungen, Zustände, Sollwerte, Monitore etc. in Registern abgelegt.

Auf diese Register kann auf einheitliche Art und Weise zugegriffen werden.

Es gibt Register im RAM, die beim Einschalten des Gerätes auf definierte Werte gesetzt werden und im Betrieb beliebig programmiert werden können.

Zum Konfigurieren und Kalibrieren liegen einige Register im EEPROM, die alle mit der Bezeichnung ">C..." beginnen. Diese Register können jederzeit ausgelesen, aber nur bei eingeschaltetem Kalibrierschalter verändert werden.

### 3.1 Allgemeines

#### 3.1.1 Befehlsform ohne Adressierung

Die Befehle kommen in dieser Form bei Geräten mit nur einer ADDA zur Anwendung.

**Die ADDA ist mit DIP- Schalter auf Standardmodus gestellt.**

**">name x"**

(Groß- und Kleinschreibung werden an keiner Stelle im gesamten Befehlssatz unterschieden)

**name:** Bezeichnung des Registers. z.B. "S0" für einen Sollwert

Zwischen **name** und dem Argument **x** muss mindestens ein Leerzeichen stehen. Das Leerzeichen kann nur entfallen, wenn bei der Registerabfrage x ein '?' ist.

**x:** Argument. Normalerweise eine Zahl, die dann in das Register geschrieben wird.

Wenn x = "?", wird das entsprechende Register ausgegeben.

*Beispiel:*

**">S1 33.5e-2"**

Setzt den I-Sollwert auf 335mA

Die ADDA antwortet im Normalfall mit

**"E0"**

E0 steht für "kein Fehler"

(siehe auch Fehlercodes unter Punkt **5. Fehlercodes**)



### 3.1.2 Befehlsform mit Adressierung

Die ADDA ist mit dem DIP- Schalter auf adressierbaren Modus eingestellt.

Die Befehle kommen in dieser Form bei Netzgeräten mit mehreren ADDAs zur Anwendung.

**"#a>name x"**

**#a** : Adresse des Interfaces, 0..127

**name**: Bezeichnung des Registers. z.B. "S0" für einen Sollwert 0

Zwischen der Adresse **a** und der Registerbezeichnung **>name** sind Leerzeichen zulässig.

Zwischen **name** und dem Argument **x** muss mindestens ein Leerzeichen stehen. Das Leerzeichen kann nur entfallen, wenn bei der Registerabfrage x ein '?' ist.

**x**: Argument. Normalerweise eine Zahl, die dann in das Register geschrieben wird.

Wenn x = "?", wird das entsprechende Register ausgegeben.

*Beispiel:*

**"#2 >S0R 1.25e2"**

Setzt im Interface mit Adresse 2 die Rampensteilheit auf 125V/sec.

Das Interface antwortet zur Kontrolle mit

**"#2 E0"**

#2 E0 steht für "Interface 2, kein Fehler"

### 3.1.3 Rücklesedaten ohne Adressierung

Das Lesen eines Registers wird eingeleitet durch einen Befehl wie:

**">name?"**

Alle Daten werden in folgendem Format ausgegeben:

**">name : x"**

**name**: Bezeichnung des Registers. z.B. "S0" für einen Sollwert

**x**: Lesedaten, z.B. Integer- oder Float-Zahl, ASCII String etc. (siehe Beschreibung der einzelnen Register)

*Beispiel 1:*

**">S1A?"**

Fragt aktuell gültigen I-Sollwert ab.

Als Antwort kommt der Registerinhalt:

**"S1A : +2 . 33400e+03"**

*Beispiel 2:*

**">M0?"**

Fragt die aktuelle Ausgangsspannung ab.

Als Antwort kommt der Registerinhalt:

**"M0 : +2 . 33400e+03"**

### 3.1.4 Rücklesedaten mit Adressierung

Das Lesen eines Registers wird eingeleitet durch einen Befehl wie:

**"#a>name?"**

Alle Daten werden in folgendem Format ausgegeben:

**"#a>name : x"**

**#a**: Adresse des Interfaces, 0..127

**name**: Bezeichnung des Registers. z.B. "S0" für einen Sollwert 0

Zwischen der Adresse **a** und der Registerbezeichnung **>name** sind Leerzeichen zulässig.

**x**: Lesedaten, z.B. Integer- oder Float-Zahl, ASCII String etc. (siehe Beschreibung der einzelnen Register)

*Beispiel 1:*

**"#2>S1?"**

Fragt von ADDA mit Adresse 2 den aktuell gesetzten I-Sollwert ab.

Als Antwort kommt dann der Registerinhalt:

**"#2 S1 : 3.35000e-01"**

*Beispiel 2:*

**"#2 >M0 ?"**

Fragt die aktuelle Ausgangsspannung von ADDA mit Adresse 2 ab.

Als Antwort kommt der Registerinhalt:

**"#2 M0 : 3.35000e-01"**

## 3.2 Registersatz

### 3.2.1 Sollwert Steuerung

#### Sollwert 0 (U-Sollwert)

Name	Read/ Write	Funktion	zulässige Werte für x	Wert nach RESET oder DCL
>S0	R/W	gesetzter Sollwert	Float-Zahl oder ?	0
>S0A	R/W	tatsächlich gültiger Sollwert (je nach Rampenfunktion)	Float-Zahl oder ?	0
>S0R	R/W	Rampensteilheit in V/sec	Float-Zahl oder ?	im Register ">CS0R"
>S0B	R/W	Steuerung Sollwert-Verhalten: <b>0: Standard, &gt;S0A wird sofort auf den Wert von &gt;S0 gesetzt.</b> 1: >S0A folgt dem Wert in >S0 mit der eingestellten Rampensteilheit sowohl aufwärts als auch abwärts. 2: >S0A folgt dem Wert in >S0 mit der eingestellten Rampensteilheit nur aufwärts. Abwärts werden die neuen Werte sofort übernommen. 3: >S0A folgt dem Wert in >S0 mit einer speziellen Rampenfunktion nur aufwärts. Abwärts werden die neuen Werte sofort übernommen. Im Bereich 0..1 mit 11.11 pro Sekunde Darüber mit >S0R 4: wie 2, jedoch wird bei >DON = 0 sowohl >S0 als auch >S0A auf null gesetzt	Byte-Zahl (0..255) oder ?	im Register ">CS0B"
>S0S	R (Read only)	Status der Rampenfunktion: 0: >S0A ist gleich >S0 1: >S0A ist nicht gleich >S0	?	
>S0H	R/W	High Resolution Mode Sollwert wird um +/- 2 Bit PWM moduliert, Auflösung 22Bit	0: Normalbetrieb 1: High Res. Mode	im Register ">CS0H"

Ist eine der Rampenfunktionen 1..4 gewählt, wird S0A auf 0 gesetzt, wenn die „Gerät-Ein Rückmeldung“ nicht anliegt. Damit wird sichergestellt, dass die Rampe nach einem Ein-Befehl oder sonstiger Freigabe stets bei null beginnt.

Der Befehl U setzt ebenfalls das Register ">S0".

### Sollwert 1 (I-Sollwert)

Name	Read/Write	Funktion	zulässige Werte für x	Wert nach RESET oder DCL
>S1	R/W	gesetzter Sollwert	Float-Zahl oder ?	0
>S1A	R/W	tatsächlich gültiger Sollwert (je nach Rampenfunktion)	Float-Zahl oder ?	0
>S1R	R/W	Rampensteilheit in A/sec	Float-Zahl oder ?	im Register ">CS1R"
>S1B	R/W	Steuerung Sollwert-Verhalten: <b>0: Standard, &gt;S1A wird sofort auf den Wert von &gt;S1 gesetzt</b> 1: >S1A folgt dem Wert in >S1 mit der eingestellten Rampensteilheit sowohl aufwärts als auch abwärts 2: >S1A folgt dem Wert in >S1 mit der eingestellten Rampensteilheit nur aufwärts. Abwärts werden die neuen Werte sofort übernommen. 3: >S1A folgt dem Wert in >S1 mit einer speziellen Rampenfunktion nur aufwärts. Abwärts werden die neuen Werte sofort übernommen. Im Bereich 0..1 mit 11.11 E-3 pro Sekunde Darüber mit >S1R 4: wie 2, jedoch wird bei >DON = 0 sowohl >S1 als auch >S1A auf null gesetzt	Byte-Zahl (0..255) oder ?	im Register ">CS1B"
>S1S	R (Read only)	Status der Rampenfunktion: 0: >S1A ist gleich >S1 1: >S1A ist nicht gleich >S1	?	
>S1H	R/W	High Resolution Mode Sollwert wird um +/- 2 Bit PWM moduliert, Auflösung 22Bit	0: Normalbetrieb 1: High Res. Mode	im Register ">CS1H"

Der Befehl I setzt ebenfalls das Register ">S1".

Ist eine der Rampenfunktionen 1..4 gewählt, wird S1A auf 0 gesetzt, wenn die „Gerät-Ein Rückmeldung“ nicht anliegt. Damit wird sichergestellt, dass die Rampe nach einem Ein-Befehl oder sonstiger Freigabe stets bei null beginnt.

*Beispiel 1: (nur eine ADDA im Gerät)*

**">s0 27.334e3"**

Setzt den U-Sollwert auf 27334.

Die ADDA antwortet mit

**"E0"**

**Beispiel 2: (Rampenfunktion, nur eine ADDA im Gerät)**

Bei einem Hochspannungsgerät soll die Ausgangsspannung dem Sollwert langsam nach oben folgen. Bei Sollwertverringering soll dagegen die Ausgangsspannung sofort verringert werden.

Entsprechende Rampenfunktion auswählen:

**">S0B 2"**

Antwortstring:

**"E0"**

Die Rampensteilheit soll 250V pro Sekunde betragen:

**">S0R 250"**

Antwortstring:

**"E0"**

Jetzt den zu erreichenden Sollwert programmieren:

**">S0 10000"** oder alternativ **U 10000**

Antwortstring:

**"E0"**

Die Ausgangsspannung des Netzgerätes wird jetzt mit 250V/sec. bis 10000V hochlaufen.

...

Der Sollwert soll jetzt wieder verringert werden:

**">S0 5000"** oder alternativ **U 5000**

Antwortstring:

**"E0"**

Die Ausgangsspannung wird jetzt je nach Belastung mit maximal möglicher Geschwindigkeit auf 5000V abfallen.

Das Gerät wird jetzt mit F0 abgeschaltet.

**"F0"**

Antwortstring:

**"E0"**

Der programmierte Sollwert von 5000 bleibt weiterhin im Register S0 vorhanden und kann testweise ausgelesen werden:

**">S0?"**

Antwortstring:

**"S0 : 5 . 00000E03"**

Dagegen ist der aktuell gültige Sollwert auf null gesetzt, da das Gerät über das Signal ON-STAT meldet, dass es abgeschaltet ist.

Testweises Abfragen des ON-Status:

**">DON?"**

Antwortstring:

**"DON : 0"**

Testweises Auslesen des aktuell gültigen Sollwerts:

**">S0A?"**

Antwortstring:

**"S0A:0.00000E00"**

Jetzt wird das Gerät mit F1 wieder eingeschaltet:

**"F1"**

Antwortstring:

**"E0"**

Der aktuell gültige Sollwert und damit die Ausgangsspannung läuft jetzt ab Null mit 250V pro Sekunde bis auf 5000V hoch.

Der gerade anstehende Sollwert kann jederzeit zwischendurch ausgelesen werden:

**">S0A?"**

Antwortstring:

**"S0A:1.23300E03"**

Der Sollwert ist also im Augenblick gerade bei 1233V angelangt.

### 3.2.2 Digitale Ausgänge

Die 5 digitalen Ausgänge der ADDA können entweder dauerhaft eingeschaltet, dauerhaft ausgeschaltet oder gepulst werden. (z.B. für Reset-Funktionen)

Name	Read/ Write	Funktion	zulässige Werte für x	Wert nach RESET oder DCL
>B0	R/W	Setzen, Pulsen, Löschen oder Abfragen von Ausgangsbit 0 <b>(Ausgang X0)</b> [1 = positiv – 0 = negativ]	0 oder 1 oder ?	0
>B0A	R (Read only)	Abfragen des tatsächlichen Ausgangswertes von Ausgang X0 (kann sich bei Pulsfunktion von >B0 unterscheiden)	?	0
>B1	R/W	Setzen, Pulsen, Löschen oder Abfragen von Ausgangsbit 1 <b>(Ausgang X1)</b> [1 = positiv – 0 = negativ]	0 oder 1 oder ?	0
>B1A	R (Read only)	Abfragen des tatsächlichen Ausgangswertes von Ausgang X1 (kann sich bei Pulsfunktion von >B1 unterscheiden)	?	0
>B2	R/W	Setzen, Pulsen, Löschen oder Abfragen von Ausgangsbit 2 <b>(Ausgang X2)</b> [1 = positiv – 0 = negativ]	0 oder 1 oder ?	0
>B2A	R (Read only)	Abfragen des tatsächlichen Ausgangswertes von Ausgang X2 (kann sich bei Pulsfunktion von >B2 unterscheiden)	?	0
>BX	R/W	Setzen, Pulsen, Löschen oder Abfragen von <b>Ausgang X-CMD</b> [0 = positiv – 1 = negativ]	0 oder 1 oder ?	0
>BXA	R (Read only)	Abfragen des tatsächlichen Ausgangswertes von Ausgang <b>X-CMD</b> (kann sich bei Pulsfunktion von >BX unterscheiden)	?	0
>BON	R/W	Setzen, Pulsen, Löschen oder Abfragen von Ausgang <b>ON-CMD</b> [0 = positiv – 1 = negativ]	0 oder 1 oder ?	0
>BONA	R (Read only)	Abfragen des tatsächlichen Ausgangswertes von Ausgang <b>ON-CMD</b> (kann sich bei Pulsfunktion von >BON unterscheiden)	?	0



Das Verhalten der digitalen Ausgänge kann jeweils mit Kalibrierwerten beeinflusst werden:  
z.B. mit >CB0P kann die Polarität des Ausgangs X0 und mit >CB0T die Pulszeit in 10ms Schritten  
eingestellt werden. Siehe Beschreibung der Kalibrierregister.

Bei eingestellter Pulszeit wird der Ausgangspuls bei einem 0->1 Übergang im Register >B0 ausgelöst  
(positiv flankengetriggert).

*Beispiel:*

**">B0 1"**

Schaltet den freien Digitalausgang X0 ein.

Die ADDA antwortet mit

**"E0"**

### 3.2.3 analoge Monitor-Register

Die Monitorregister werden im Hintergrund laufend mit den aktuellen Messwerten des AD-Wandlers versorgt.

Name	Read/ Write	Funktion	zulässige Werte für x
>M0	R (Read only)	Monitor 0 (U-Monitor)	?
>M0R	R (Read only)	Monitor 0 (U-Monitor), unkalibrierter Roh-Messwert des AD-Wandlers	?
>M0I	R/W	Monitor 0 (U-Monitor), Auflösung und Integrationszeit des AD-Wandlers	0..7 oder ?
>M1	R (Read only)	Monitor 1 (I-Monitor)	?
>M1R	R (Read only)	Monitor 1 (I-Monitor), unkalibrierter Roh-Messwert des AD-Wandlers	?
>M1I	R/W	Monitor 1 (I-Monitor), Auflösung und Integrationszeit des AD-Wandlers	0..7 oder ?

Argument x	Auflösung	Integrationszeit
0	14Bit + VZ	256µs
1	15Bit + VZ	1ms
2	15Bit + VZ	4ms
3	17Bit + VZ	20ms
4	17Bit + VZ	40ms
5	typ. 18Bit + VZ	80ms
6	typ. 19Bit + VZ	200ms
7	typ. 20Bit + VZ	800ms

Anm.: Der Definitionsbereich des Arguments stellt weitgehende Kompatibilität zu früheren Probus IV Versionen her.

Der Probus IV kompatible Befehl Sn schreibt sein Argument n in beide Register >M0I und >M1I.

Auflösung ist definiert als nutzbare Anzahl Bits (effective Number of Bits, enob). Die Bruchteile des Messwertes mit geringerer Wertigkeit rauschen.

*Beispiel: (mit Adressierung)*

**"#1 m0?"**

Fragt von ADDA mit Adresse 1 die aktuelle Ausgangsspannung ab.

Als Antwort wird der Registerinhalt ausgegeben:

**"#1 M0: +2.33400e+03"**

### 3.2.4 Digitale Eingänge

Auf diese Register kann nur lesend (mit ?) zugegriffen werden.

Name	Read/ Write	Funktion	zulässige Werte für x
>DVR	R (Read only)	Abfragen von Eingangsbit <b>V-REG</b> (ist auf 1 wenn Gerät in U-Regelung)	?
>DIR	R (Read only)	Abfragen von Eingangsbit <b>I-REG</b> (ist auf 1 wenn Gerät in I-Regelung)	?
>D3R	R (Read only)	Abfragen von Eingangsbit <b>3-REG</b> (3. Regelkreis für Sonderanwendungen)	?
>DX	R (Read only)	Abfragen von Eingangsbit <b>X-STAT</b> (für Sonderanwendungen) [0 = positiv – 1 = negativ]	?
>DON	R (Read only)	Abfragen von Eingangsbit <b>ON-STAT</b> (ist auf 1, wenn Ausgang eingeschaltet ist)	?
>DSD	R (Read only)	Abfragen von Eingangsbit <b>SEL-D</b> (1=Gerät wird digital gesteuert)	?
>DSA	R (Read only)	Abfragen von Eingangsbit <b>SEL-A</b> (1=Gerät wird analog gesteuert)	?
>DCAL	R (Read only)	Abfragen von Eingangsbit <b>CAL</b> (ist auf 1, wenn ADDA im CAL Modus)	?

*Beispiel: (mit Adressierung)*

**"#1 >DVR ?"**

Fragt in der ADDA mit Adresse 1 das Statusbit V-REG ab.

Die ADDA antwortet z.B. mit

**"#1 DVR:1"**

### 3.2.5 Konfigurations- und Statuswerte

Name	Read/Write	Funktion	zulässige Werte für x	Wert nach RESET oder DCL
>KT	R/W	Terminator für die Antwortstrings (siehe Befehl Y)	0..3 oder ? 0: CR+LF 1: LF+CR <b>2: LF</b> 3: CR	im Kalibrier-Register ">CKT"
>KS	R (Read only)	Statusbyte zusammengefasster Status (kompatibel zu Probus IV), Ausgabe als 0/1 Folge, MSB zuerst Bit 7: I-REG Bit 6: V-REG Bit 5: ON-Status Bit 4: 3-Reg Bit 3: X-Stat (Polarität) Bit 2: Cal-Mode Bit 1: unbenutzt Bit 0: SEL-D	?	entsprechend dem tatsächlichen Gerätezustand
>KQS	R (Read only)	SRQ-Statusbyte Ausgabe als Dezimalzahl Bit 1: Gerät ist in I-Regelung Bit 2: Gerät ist in U-Regelung	?	0
>KQM	R/W	SRQ-Mask (Service-Request, siehe Befehl M)	0...255 bzw ? Freigabe bestimmter Bits für SRQ <b>0: kein SRQ (default nach RESET)</b> Bit 1: SRQ bei Wechsel auf I-Regelung Bit 2: SRQ bei Wechsel auf U-Regelung	0
>KX	R (Read only)	steuert Execute-on-X	0: sofortige Befehlsausführung 1: Befehlsausführung erst bei X-Befehl	0
>KN	R/W	Vorwahl der Rücklesedaten für Trigger-on-Talk (wird mit N-Befehl gesetzt)	0..6 oder ? <b>0: Register &gt;M0 (U-Monitor)</b> 1: Register >M1 (I-Monitor) 2: Register >KS (Statusbyte als 0/1 Folge) 3: Register >CS0T (Sollwert-Typenspannung) 4: Register >CS1T (Sollwert-Typenstrom) 5: Register >CFV (Firmware Versionsnummer) 6: Register >CFN (Seriennummer)	im Kalibrier-Register ">CKN"
>KE	R (Read only)	Error-Code des letzten Befehls	?	0

### 3.2.6 Kalibrationswerte

Alle Kalibrationswerte sind in einem nichtflüchtigen Speicher (EEPROM) abgelegt.

Um versehentliche Änderungen auszuschließen, ist das EEPROM mit einem Schreibschutz versehen, der über DIP-Schalter gesteuert wird (siehe 1.4).

Auf alle Kalibrationswerte kann beliebig schreibend oder lesend zugegriffen werden. Beim Schreiben auf eines der C-Register wird der Wert sofort im EEPROM abgelegt. Ein separater Speicherbefehl wie bei Probus IV ist nicht nötig.

**ACHTUNG:** Die Lebensdauer des EEPROMs ist auf 100000 Schreibzyklen begrenzt.

Wird bei eingeschaltetem Schreibschutz in ein Kalibrierregister geschrieben, bleibt der alte Wert unverändert und es wird ein Fehlercode zurückgeliefert.

Kalibrationswerte für analoge Sollwerte				
Name	Read/Write	Funktion		zulässige Werte für x
>CS0T	R/W	Analogausgang 0 (V-Set)	Typenwert	Float-Zahl oder ?
>CS0GP	R/W		Gain positiv	Float-Zahl oder ?
>CS0GN	R/W		Gain negativ	Float-Zahl oder ?
>CS0OP	R/W		Offset positiv (in Bits)	Integer-Zahl oder ?
>CS0ON	R/W		Offset negativ (in Bits)	Integer-Zahl oder ?
>CS0R	R/W		default Rampengeschwindigkeit	Float-Zahl oder ?
>CS0B	R/W		default Sollwert-Verhalten	0..4 oder ?
>CS0H	R/W		default High Res. Mode	0 oder 1 oder ?
>CS1T	R/W	Analogausgang 1 (I-Set)	Typenwert	Float-Zahl oder ?
>CS1GP	R/W		Gain positiv	Float-Zahl oder ?
>CS1GN	R/W		Gain negativ	Float-Zahl oder ?
>CS1OP	R/W		Offset positiv (in Bits)	Integer-Zahl oder ?
>CS1ON	R/W		Offset negativ (in Bits)	Integer-Zahl oder ?
>CS1R	R/W		default Rampengeschwindigkeit	Float-Zahl oder ?
>CS1B	R/W		default Sollwert-Verhalten	0..4 oder ?
>CS1H	R/W		default High Res. Mode	0 oder 1 oder ?

Kalibrationswerte für analoge Monitore				
Name	Read/Write	Funktion		zulässige Werte für x
>CM0T	R/W	ADC 0 (V-Monitor)	Typenwert	Float-Zahl oder ?
>CM0GP	R/W		Gain positiv	Float-Zahl oder ?
>CM0GN	R/W		Gain negativ	Float-Zahl oder ?
>CM0O	R/W		Offset (in Bits)	Long-Zahl oder ?
>CM0I	R/W		default Integrationszeit	0..7 oder ?
>CM1T	R/W	ADC 1 (I-Monitor)	Typenwert	Float-Zahl oder ?
>CM1GP	R/W		Gain positiv	Float-Zahl oder ?
>CM1GN	R/W		Gain negativ	Float-Zahl oder ?
>CM1O	R/W		Offset (in Bits)	Long-Zahl oder ?
>CM1I	R/W		default Integrationszeit	0..7 oder ?

Kalibrationswerte für digitale Ausgänge				
Name	Read/Write	Funktion		zulässige Werte für x
>CB0P	R/W	Ausgang X0	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CB0T	R/W		Pulszeit in 10ms (0=kein Pulsen)	Byte-Zahl oder ?
>CB1P	R/W	Ausgang X1	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CB1T	R/W		Pulszeit in 10ms (0=kein Pulsen)	Byte-Zahl oder ?
>CB2P	R/W	Ausgang X2	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CB2T	R/W		Pulszeit in 10ms (0=kein Pulsen)	Byte-Zahl oder ?
>CBXP	R/W	Ausgang X-CMD	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CBXT	R/W		Pulszeit in 10ms (0=kein Pulsen)	Byte-Zahl oder ?
>CBONP	R/W	Ausgang ON-CMD	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CBONT	R/W		Pulszeit in 10ms (0=kein Pulsen)	Byte-Zahl oder ?

Kalibrationswerte für digitale Eingänge				
Name	Read/Write	Funktion		zulässige Werte für x
>CDVRP	R/W	Eingang V-REG	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CDIRP	R/W	Eingang I-REG	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CD3RP	R/W	Eingang 3-REG	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CDXP	R/W	Eingang X-STAT	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?
>CDONP	R/W	Eingang ON-STAT	Polarität (0=normal, 1=invertiert)	0 oder 1 oder ?

allgemeine Kalibrationswerte				
Name	Read/Write	Funktion		zulässige Werte für x
>CFN	R/W	Fabriknummer-String		String mit max. 50 Zeichen oder ?
>CFNNUM	R/W	numerische Fabriknummer (u.A. für Profibus)		0..2147483647
>CFV	R (Read only)	Firmware-Version		?
>CADR	R/W	Adresse für adressierbaren Modus		Byte-Zahl oder ?
>CKT	R/W	default Terminator		0 ... 3 oder ?
>CBAUD	R/W	spezielle Baudrate		? 0: 4800Bd 1: 9600Bd 2: 19200Bd 3: 38400Bd 4: 115kBd <b>5: 230kBd (default ab Werk)</b> 6: 500kBd 7: 625kBd
>CASM	R/W	Auto-Send Master Mode (Verwendung bei potentialfreier Analogprogrammierung über Lichtleiter, ADDAT mit >CASM=1 sendet >HAXX Befehl an ferne ADDAT) bei Profibus: >CASM 3		<b>0: Es ist keine ferne ADDAT angeschlossen</b> 1: diese ADDAT steuert ferne ADDAT 2: diese ADDAT wird von ferner ADDAT gesteuert 3: ADDAT erwartet mindestens alle 500ms bzw. 5 sek. neuen >H0 String

>CONBR	R/W	Umschaltung der ON-STAT Rückmeldung im Register >DON sowie im Statusbyte >KS: 0: >DON zeigt den tatsächlichen Gerätezustand (nur bei Geräten der Serien HCP und MCP implementiert) 1: >DON wird stets auf den Wert von >BONA gesetzt (Probus IV kompatibel)	0 oder 1 oder ?
>CKN	R/W	default N-Wert für Ausgabe-Vorwahl	Byte-Zahl oder ?
>CCS	R/W	Typ der Checksumme für Befehls- und Antwortstrings siehe auch Abschnitt 3.4	0: keine Checksumme 1: 2 Byte Hex Checksumme
>CPAR	R/W	Parallelbetrieb (z.B. RS-485) Register wirkt sich nur im adressierbaren Modus aus (DIP-Schalter)	<b>0: Normalbetrieb, fremde Adressen werden weitergeleitet</b> 1: Parallelbetrieb: fremde Adressen werden ignoriert

*Beispiel 1: (Typenspannung setzen, ohne Adressierung)*

**">CS0T 12500"**

Setzt die Typenspannung auf 12500.

Wenn Schreibschutz aufgehoben ist, wird der Wert sofort ins EEPROM geschrieben und nach einigen ms der Befehl quittiert:

**"E0"**

Wenn Schreibschutz eingeschaltet ist, wird der Wert verworfen und der Antwortstring E8 ausgegeben

**"E8"**

EEPROM Write Protect Error

*Beispiel 2: (Typenspannung lesen, ohne Adressierung)*

**">CS0T?"**

Startet die Abfrage der Typenspannung

Die ADDA antwortet mit

**"CS0T: +1.25000e+04"**

**Achtung:** Durch die interne binär-Arithmetik kann es zu Rundungsfehlern kommen. Der Rücklesewert kann daher an der letzten Stelle abweichen. (meist +/- 1 Digit)

### 3.2.7 Virtuelle Register

Diese Register dienen zum automatischen Anpassen eines universellen Steuerprogramms an die jeweilige Firmware-Version der ADDA.

Name	Read/ Write	Funktion	zulässige Werte für x
>CLIST	W	Rücksetzen der C-Konstanten Abfrage und Ausgabe der Spaltenüberschriften	kein Argument
	R	bei jedem Lesevorgang wird die jeweils nächste Kalibrierkonstante mit vorangestelltem > ausgegeben (kann in File gespeichert werden zum Dokumentieren und Kopieren von Kalibrierkonstanten)	?
>RLIST	W	Rücksetzen der Register Abfrage und Ausgabe der Spaltenüberschriften	kein Argument
	R	bei jedem Lesevorgang wird das jeweils nächste Register ausgegeben im Format: Name, Hilfetext, DatenTyp, R/WTyp, Inhalt z.B.: "CS0T";"U Soll Typenwert";2;0;1.00000E01	?

#### Datentypen bei RLIST: (Spalte DataType)

- 0: Floatingpoint Zahl
- 1: 32Bit Integer Zahl mit Vorzeichen (long)
- 2: 16Bit Integer Zahl mit Vorzeichen (int)
- 3: 8Bit Zahl ohne Vorzeichen (unsigned char)
- 4: 8Bit Wert, Ausgabe als 0/1 Folge mit MSB zuerst (unsigned char)
- 5: 8Bit Zahl mit Vorzeichen (signed char)
- 6: 1Bit Zahl
- 7: String
- 8: String
- 9: Hexadezimaler Eingabeblock
- 10: Hexadezimaler Ausgabeblock

#### R/W Eigenschaften bei RLIST: (Spalte Rd/Wr/Cal)

- 0: Read Only
- 1: Write Only
- 2: Read/Write
- 3: Read immer / Write nur im Kalibriermodus



### 3.2.8 gemeinsames Setzen und Lesen mehrerer Register

Speziell für die Kommunikation mit zyklisch arbeitenden binär organisierten Bus-Systemen (z.B. Profibus) sind die H-Register vorgesehen.

Mit einem einzigen Befehlsstring, der vom Bus-Gateway zyklisch übermittelt wird, werden beide Sollwertausgänge und alle digitalen Ausgänge gleichzeitig gesetzt.

Die ADDA antwortet mit einem längeren Antwortstring, in dem beide Monitorwerte sowie die digitalen Zustände zusammengefasst sind.

Name	Read/ Write	Funktion	zulässige Werte für x
>H0	W (Write only)	alle Sollwerte und Digitalausgänge setzen Antwortstring ist Inhalt von Register H1	String aus 32 Hex-Ziffern, entspricht 16 Bytes
>H1	R (Read only)	alle Monitore und Zustände in einem zusammengefassten String ausgeben.	?
>HA	W (Write only)	Kurzform von >H0 für direkte Kommunikation mit zweiter, ferner ADDAT bei analoger Lichtleiter-Programmierung	String aus 20 Hex-Ziffern, entspricht 10 Bytes

*Beispiel:*

**">H0 000F4240000003E80000000000000001"**

Das Interface antwortet z.B. mit

**"000F420100050000000010102030001"**

Datenformat Befehlsstring >H0		
Byte		
0	LSB	Sollwert 0 (Spannungssollwert), unsigned integer, 0...65535 entspricht 0...Typenwert in >CS0T, >S0 wird gesetzt
1	MSB	
2	LSB	Sollwert 1 (Stromsollwert), unsigned integer 0...65535 entspricht 0...Typenstrom in >CS1T, >S1 wird gesetzt
3	MSB	
4	Bit 0	Vorzeichen für Sollwert 0 (0=pos., 1=neg.)
	Bit 1	Vorzeichen für Sollwert 1 (0=pos., 1=neg.)
	Bit 2	unbenutzt
	Bit 3	unbenutzt
	Bit 4	unbenutzt
	Bit 5	unbenutzt
	Bit 6	unbenutzt
	Bit 7	0: bei >CASM 3 Timeout 5sek. / 1: bei >CASM 3 Timeout 500ms
5	Bit 0..3	Sollwert Verhalten von Sollwert 0, Wert wird in Register >S0B übertragen
	Bit 4..7	Sollwert Verhalten von Sollwert 1, Wert wird in Register >S1B übertragen
6		unbenutzt
7	Bit 0	Gerät EIN, Wert wird in Register >BON übertragen
	Bit 1	X-CMD (Umpolbefehl bei Geräten mit Motorumpoler), wird in Register >BX übertr.
	Bit 2	Ausgang X0, Wert wird in Register >B0 übertragen
	Bit 3	Ausgang X1, Wert wird in Register >B1 übertragen
	Bit 4	Ausgang X2, Wert wird in Register >B2 übertragen
	Bit 5	unbenutzt
	Bit 6	unbenutzt
	Bit 7	unbenutzt
8	LSB	Sollwert 0 Rampensteilheit, unsigned integer, 0...65535 entspricht 0... 1xTypenwert pro Sekunde, >S0R wird gesetzt
9	MSB	
10	LSB	Sollwert 1 Rampensteilheit, unsigned integer, 0...65535 entspricht 0... 1xTypenwert pro Sekunde, >S1R wird gesetzt
11	MSB	
12		unbenutzt
13		unbenutzt
14		unbenutzt
15		SYNC Byte, Wert wird in Ausgangsdatenblock übertragen

<b>Datenformat Ausgabestring &gt;H1</b>		
<b>Byte</b>		
0	LSB	Monitor 0 (Spannungsmonitor), unsigned integer, 0...65535 entspricht 0... Typenwert in >CM0T
1	MSB	
2	LSB	Monitor 1 (Strommonitor), unsigned integer, 0...65535 entspricht 0... Typenwert in >CM1T
3	MSB	
4	Bit 0	Vorzeichen von Monitor 0 ((0=pos., 1=neg.)
	Bit 1	Vorzeichen von Monitor 1 ((0=pos., 1=neg.)
	Bit 2	unbenutzt
	Bit 3	unbenutzt
	Bit 4	unbenutzt
	Bit 5	unbenutzt
	Bit 6	unbenutzt
	Bit 7	unbenutzt
5	Bit 0	Eingang SEL-D (1=Programmierung auf Digital gestellt)
	Bit 1	unbenutzt
	Bit 2	CAL-Mode (1=Kalibrierschalter steht auf Kalibrierfreigabe)
	Bit 3	Eingang X-STAT (Stellung des Umpolschalters bei Motorumpolung)
	Bit 4	Eingang 3-REG (3. Regelkreis für Sonderanwendungen)
	Bit 5	Eingang ON-STAT (1=Ausgangsspannung ist freigegeben)
	Bit 6	Eingang V-REG (1=Gerät in Spannungsregelung)
	Bit 7	Eingang I-REG (1=Gerät in Stromregelung)
6		unbenutzt
7		unbenutzt
8		unbenutzt
9		unbenutzt
10	LSB	Seriennummer des Gerätes, long integer
11		
12		
13	MSB	
14		Errorcode des letzten Befehls
15		SYNC Byte, Wert wird von Eingangsdatenblock übertragen

### 3.3 Service Request , Automatische Übertragung bei Änderungen

Diese Funktion dient ausschließlich der internen Steuerung des Interface Konverters für IEEE-488 (Baugruppe IBIG50).

Es wird nicht empfohlen, diese Funktion ohne den IEEE-488 Interface Konverter zu verwenden, da bei häufigen SRQs der serielle Empfangspuffer des PCs überlaufen kann.

Das Konfigurationsregister >KQM (SRQ-Mask) steuert die **sofortige** und **asynchrone** Übertragung eines Service-Request-Strings wenn sich bestimmte digitale Zustände ändern.

Das Register >KQM kann auch mit dem Probus IV- kompatiblen Befehl M bzw. dem Sonderbefehl ~M geändert werden.

Alle Sendedaten, die auf diese Art automatisch erzeugt wurden, sind mit einem vorangestellten "~Q" gekennzeichnet.

~Q2 Seit der letzten ~Qn Übertragung hat das Gerät in den Zustand I-konstant gewechselt

~Q4 Seit der letzten ~Qn Übertragung hat das Gerät in den Zustand U-konstant gewechselt

~Q6 Seit der letzten ~Qn Übertragung hat das Gerät in den Zustand I-konstant oder U-konstant gewechselt

Um eine Überlastung der Kommunikation zu vermeiden, ist der Abstand zwischen zwei ~Qn Strings mindestens 100ms.

Wenn mehrere ADDAs zusammen mit einer IBIG50 betrieben werden, sind in jeder ADDA mit adressierten Befehlen die >KQM Register zu setzen.

Der unadressierte M-Befehl wirkt in diesem Fall nur auf den Interface Konverter als gemeinsame SRQ-Maske.

*Beispiel:*

**"M2 "**

Gibt die automatische Übertragung der Meldung "**~Q2 "** frei, wenn der Regelzustand in I-konstant geht.

Das Interface antwortet zunächst mit

**"E0 "**

Bei einer Änderung Zustandsbits I-Regelung nach log. 1 wird die entsprechende Meldung sofort asynchron zur normalen Kommunikation übertragen.

**"~Q2 "**

Um die automatische Übertragung wieder zu stoppen, ist das Register >KQM auf Null zu setzen:

**"M0 " oder ">KQM 0 "**

### 3.4 Checksumme

Für kritische Anwendungen kann die Übertragung über eine Checksumme abgesichert werden. Siehe dazu auch Beschreibung des Registers >CCS im Abschnitt 3.2.6.

Sonderfälle:

- Im Kalibriermodus wird stets ohne Checksumme gearbeitet.
- Der Befehl \*IDN sowie die Sonderbefehle (~T1, ~T2, ~M) werden ohne Checksumme akzeptiert.
- Im adressierbaren Modus wird die Checksumme nur für Befehle überprüft, die an die lokale Adresse gerichtet sind.

#### 3.4.1 Checksumme Typ1

Das Register >CCS ist auf 1 gesetzt.

##### Berechnung der Checksumme:

An den Befehlsstring ohne Terminatorzeichen wird zunächst ein Leerzeichen angehängt zur Trennung von der nachfolgenden Prüfsumme.

Es werden dann alle ASCII-Codes der Zeichen des reinen Befehlsstrings incl. des Leerzeichens addiert. Die Berechnung erfolgt als vorzeichenlose Addition mit 16Bit Breite.

Die so ermittelte Prüfsumme wird als 4-stellige Hexadezimalzahl an den Befehlsstring angehängt.

Erst danach wird der Befehlsstring mit einem oder mehreren Terminatorzeichen abgeschlossen (CR, LF oder 0x00).

Alle Antwortstrings werden in gleicher Weise mit einer Checksumme vervollständigt.

##### Beispiel 1:

Die Ausgangsspannung soll auf 15.3V gesetzt werden:

Ursprünglicher Befehlsstring "U 15.3"

mit angehängtem Leerzeichen "U 15.3 "

Addition der ASCII Codes:  $85+32+49+53+46+51+32 = 348$ , in Hex-Darstellung 0x015C

Diese hexadezimale Zahl wird nun an den String angehängt

Befehlsstring mit Checksumme "U 15.3 015C"

Die folgenden Terminatorzeichen sind nicht Teil der Checksumme.

Das Gerät antwortet mit "E0 0095"

##### Beispiel 2:

Die Prüfsumme ist eingeschaltet (Register >CCS ist auf 1) und soll abgeschaltet werden. Dazu muss bei freigeschalteter Kalibriersperre das Register >CCS auf 0 gesetzt werden:

Ursprünglicher Befehlsstring ">CCS 0"

mit angehängtem Leerzeichen ">CCS 0 "

Addition der ASCII Codes  $62+67+67+83+32+48+32 = 391$ , in Hex-Darstellung 0x0187

Diese hexadezimale Zahl wird nun an den String angehängt

Befehlsstring mit Checksumme ">CCS 0 0187"

Die folgenden Terminatorzeichen sind nicht Teil der Checksumme.

Das Gerät antwortet mit "E0 0095"

## 4 Sonderbefehle und Probus IV kompatible Befehle

### 4.1 Identifikationsstring ausgeben

Syntax	Antwort-String	Funktion
*IDN?	#n FuG . . . . ADDA ist im adressierbaren Modus	gibt Fabriknummer aus und setzt in der 1. ADDA in der LWL-Kette Monitor-Auswahl auf N6
	FuG . . . . ADDA ist im Standard Mode	gibt Fabriknummer aus und setzt Monitor-Auswahl auf N6

Als Antwort wird der im Register ">CFN" stehende Seriennummer-String ausgegeben. Wenn mehrere ADDAs in Reihe geschaltet sind, gibt nur die erste ADDA den String aus.

Im Zusammenspiel mit IBIG50 im Probus IV kompatiblen Mode1 wird der Antwortstring von der IBIG50 unterdrückt.

Deshalb setzt \*IDN? auch N6, damit bei der nachfolgenden Talker-Adressierung mit erneuter Datenanfrage über "~T1" die Gerätenummer ausgegeben wird.

### 4.2 Device Clear

Syntax	Antwort-String	Funktion
=	= ADDA ist im adressierbaren Modus und nicht auf Adresse 0 eingestellt (Antwort-String dient als Befehlsstring für nachfolgende ADDAs in der LWL-Kette.)	Setzt Interface zurück
	E0 ADDA ist im Standard Mode oder im adressierbaren Mode mit Adresse 0 eingestellt	

Folgende Register und Funktionen werden im Interface zurückgesetzt:

- alle Sollwertregister auf 0
- Sollwert-Rampenfunktionen auf den kalibrierten Default-Wert im EEPROM
- Sollwert-Rampensteilheiten auf den kalibrierten Default-Wert im EEPROM
- alle digitalen Ausgänge in den Ruhezustand
- Probus IV kompatible Monitorauswahl (N-Befehl) auf den kalibrierten Default-Wert im EEPROM
- Service Request Maske auf 0
- Service Request Register auf 0
- ADC Integrationszeiten auf die kalibrierten Default-Werte im EEPROM
- Terminator für Antwortstrings auf den kalibrierten Default-Wert im EEPROM

### 4.3 Ausgangsspannung ein-/ausschalten

Syntax	Antwort-String	Funktion	Bemerkung
<b>Fx</b> oder <b>fx</b> z.B. <b>F1</b>	<b>E0</b>	schreibt x in Register ">BON"	ADDA ist im Standard Mode
<b>#nFx</b> oder <b>#nfx</b> z.B. <b>#1F0</b>	<b>#nE0</b>	schreibt x in Register ">BON" in ADDA n	ADDA ist im adressierbaren Modus

Mit F1 bzw. F0 wird die Ausgangsspannung des Gerätes elektronisch freigegeben bzw. gesperrt. Dieser Befehl wird bei Sondergeräten ggf. auch für andere Funktionen verwendet.

### 4.4 Spannungssollwert setzen

Syntax	Antwort-String	Funktion	Bemerkung
<b>Ux</b> oder <b>ux</b> z.B. <b>U3.47E2</b>	<b>E0</b>	Setzt Spannungssollwert im Register ">S0"	ADDA ist im Standard Mode
<b>#nUx</b> oder <b>#nux</b> z.B. <b>#2U5.23</b>	<b>#nE0</b>	Setzt Spannungssollwert im Register >S0 in ADDA n	ADDA ist im adressierbaren Modus

### 4.5 Stromsollwert setzen

Syntax	Antwort-String	Funktion	Bemerkung
<b>Ix</b> z.B. <b>I7.78</b>	<b>E0</b>	Setzt Stromsollwert im Register >S1	ADDA ist im Standard Mode
<b>#nIx</b> z.B. <b>#3i5</b>	<b>#nE0</b>	Setzt Stromsollwert im Register >S1 in ADDA n	ADDA ist im adressierbaren Modus

### 4.6 Rücklesedaten für getriggerte Datenabfrage vorwählen

Syntax	Antwort-String	Funktion	Bemerkung
<b>Nx</b> oder <b>nx</b> z.B. <b>N1</b>	<b>E0</b>	schreibt x in Register ">KN"	ADDA ist im Standard Mode
<b>#nNx</b> oder <b>#nnx</b> z.B. <b>#1n6</b>	<b>#nE0</b>	schreibt x in Register ">KN" in ADDA n	ADDA ist im adressierbaren Modus

Damit wird vorgewählt, welche Rücklesedaten nach GPIB Trigger-on-Talk (über ~T1) bzw. Datenabfrage (mit ?) gesendet werden sollen (Probus IV kompatibel).

Siehe Beschreibung von Register ">KN".

## 4.7 getriggerte Datenabfrage

Syntax	Antwort-String	Funktion	
?	z.B.: 1.2345E01VN	gibt den mit dem Befehl "N" bzw. dem Register ">KN" vorgewählten Ausgabestring zurück	ADDA ist im Standard Mode
#n? z.B. #3?	#3 1.2345E01VN	gibt den mit dem Befehl "#n N" bzw. dem Register "#n >KN" vorgewählten Ausgabestring zurück	ADDA ist im adressierbaren Modus

Implementierung für Probus IV Kompatibilität im RS-232 Betrieb.

Funktion ist auch möglich zusammen mit IBIG50 im Mode2.

## 4.8 AD-Wandlungsrate und Auflösung einstellen

Die Wandlungsrate des AD-Wandlers kann für jeden der beiden Kanäle getrennt gewählt werden. Die erzielbare Auflösung ist abhängig von der eingestellten Wandlungsrate.

Syntax	Antwort-String	Funktion	Bemerkung
Sx oder sx z.B. S1	E0	beeinflusst gleichzeitig Register ">M0I" und ">M1I"	ADDA ist im Standard Mode
#nSx oder #nsx z.B. #1S2	#nE0	beeinflusst gleichzeitig Register ">M0I" und ">M1I" in ADDA n	ADDA ist im adressierbaren Modus

Die Bedeutung des Arguments x ist bei den analogen Monitor-Registern in 3.2.3 beschrieben.

Mit dem Befehl S werden beide Analog-Kanäle gleichzeitig beeinflusst. Siehe Beschreibung zu den analogen Monitor-Registern um für die Kanäle unterschiedlichen Auflösungen einzustellen.

Anm.: Der Befehl S stellt weitgehende Kompatibilität zu früheren Probus IV Versionen her.

## 4.9 Ausgangspolarität einstellen

Syntax	Antwort-String	Funktion	Bemerkung
Px oder px z.B. F1	E0	schreibt x in Register ">BX"	ADDA ist im Standard Mode
#nPx oder #npx z.B. #1P0	#nE0	schreibt x in Register ">BX" in ADDA n	ADDA ist im adressierbaren Modus

P0 stellt positive Polarität ein,  
 P1 stellt negative Polarität ein.

Dieser Befehl ist nur bei Geräten mit fernsteuerbarer Umpolung aktiv.

Bei Sondergeräten kann dieser Befehl auch für andere Funktionen verwendet werden.



## 4.10 X-Ausgänge ein-/ausschalten

Syntax	Antwort-String	Funktion	Bemerkung
<b>Rx</b> oder <b>rx</b> z.B. R1	E0	beeinflusst Register ">B0", ">B1" und ">B2"	ADDA ist im Standard Mode
<b>#nRx</b> oder <b>#nrx</b> z.B. #1R2	#nE0	beeinflusst Register ">B0", ">B1" und ">B2" in ADDA n	ADDA ist im adressierbaren Modus

Die Erweiterungs-Ausgänge X0, X1 und X2 der ADDA-Baugruppe dienen der Steuerung von Funktionen bei Sondergeräten. Diese Ausgänge werden durch Setzen der Register >B0, >B1 und >B2 direkt gesteuert (siehe 3.2.2)

Diese drei Register können auch mit dem Probus IV kompatiblen Befehl R gesetzt werden.

R0..R7: ">B0", ">B1" und ">B2" werden gleichzeitig auf den binären Wert von x gesetzt (B0=LSB)

R8:                               setzt nur ">B0" auf 0  
 R9:                               setzt nur ">B0" auf 1  
 R10:                              setzt nur ">B1" auf 0  
 R11:                              setzt nur ">B1" auf 1  
 R12:                              setzt nur ">B2" auf 0  
 R13:                              setzt nur ">B2" auf 1

## 4.11 Execute-on-X ein-/ausschalten

Damit kann die Ausführung von bestimmten Befehlen verzögert werden. Die Argumente der Befehle werden zwischengespeichert und erst bei Empfang des "X" Kommandos übernommen.

Funktion ist implementiert, um Kompatibilität mit Probus IV herzustellen.

Aufgrund der sehr viel schnelleren Befehlsverarbeitungszeit des Probus V Systems im Vergleich zu Probus IV kann und sollte bei neuen Softwareprojekten auf Execute-on-X verzichtet werden.

Syntax	Antwort-String	Funktion	Bemerkung
<b>Gx</b> oder <b>gx</b> z.B. G1	E0	schreibt x in Register ">KX"	ADDA ist im Standard Mode
<b>#nGx</b> oder <b>#ngx</b> z.B. #1G1	#nE0	schreibt x in Register ">KX" in ADDA n	ADDA ist im adressierbaren Modus

G0: Alle Befehle werden sofort ausgeführt,

G1: Bestimmte Befehle (siehe Befehl X) werden zwischengespeichert und erst beim "X"-Befehl ausgeführt.

## 4.12 Execute on X

Syntax	Antwort-String	Funktion
<b>x</b> oder <b>x</b>	<b>x</b> ADDA ist im adressierbaren Modus und nicht auf Adresse 0 eingestellt (Antwort-String dient als Befehlsstring für nachfolgende ADDAs in der LWL-Kette.)	Probus IV -kompatible Funktion Execute-on-X (Die zwischengespeicherten Argumente der vorangegangenen Befehle U, I, F, N, P werden aktiviert.)
	<b>E0</b> ADDA ist im Standard Mode oder im adressierbaren Mode mit Adresse 0 eingestellt	

## 4.13 Set Terminator

Syntax	Antwort-String	Funktion
<b>Yn</b> oder <b>yn</b> (Befehl muss stets ohne vorangestellte Adresse gesendet werden)	<b>Yn</b> ADDA ist im adressierbaren Modus und nicht auf Adresse 0 eingestellt (Antwort-String dient als Befehlsstring für nachfolgende ADDAs in der LWL-Kette.)	Y0: CR+LF Y1: LF+CR Y2: LF Y3: CR
	<b>E0</b> ADDA ist im Standard Mode oder im adressierbaren Mode mit Adresse 0 eingestellt	default-Wert im Register >CKT (normalerweise 2)

Stellt die Terminatorzeichen für Antwortstrings ein, die vom Interface gesendet werden. Der Befehl schreibt das Argument n in das Register >KT.

Im adressierbaren Modus muss dieser Befehl stets ohne vorangestellte Adresse gesendet werden. Es werden dann die >KT Register in allen ADDAs gesetzt.

Siehe auch Register ">KT" bzw. ">CKT "

## 4.14 SRQ-Maske setzen

Syntax	Antwort-String	Funktion	Bemerkung
<b>Mx</b> oder <b>mx</b> z. B. <b>m6</b>	<b>E0</b>	Setzt SRQ-Maske im Register > KQM	ADDA ist im Standard Mode
<b>#nMx</b> oder <b>#nmX</b> z. B. <b>#1m6</b>	<b>#nE0</b>	Setzt SRQ-Maske im Register > KQM in ADDA n	ADDA ist im adressierbaren Modus

Beschreibung des Arguments siehe Register >KQM in 3.2.5

Die adressierbare Form muss verwendet werden, wenn IBIG50 zusammen mit mehreren ADDAs betrieben wird.

Ein unadressierter M-Befehl des Kunden setzt bei mehreren ADDAs nur die SRQ-Maske in der IBIG50.

## 4.15 Sonderbefehle

Diese Befehle dienen der internen Kommunikation mit bestimmten Schnittstellenumsetzern und sind nicht für allgemeine Programmierung des Netzgerätes bestimmt.

Bei eingestellter Absicherung der Übertragung mit Checksumme werden diese Befehle ohne Checksumme akzeptiert.

Sie werden hier nur der Vollständigkeit halber angegeben.

### 4.15.1 Sonderbefehl SRQ-Maske setzen

Syntax	Antwort-String	Funktion	Bemerkung
~Mn z.B. ~M4	E0	keine Funktion	ADDA ist im Standard Mode
		Setzt SRQ-Maske	ADDA ist im adressierbaren Modus

Unterschiedliche Funktion, je nachdem, ob ADDA adressierbar ist oder nicht:

Wenn ADDA im nicht adressierbaren Mode ist, setzt der Befehl die SRQ Maske im Register ">KQM" (Kompatibilität zu Probus IV).

Wenn ADDA im adressierbaren Mode ist, wird lediglich der Antwortstring E0 ausgegeben, die SRQ-Maske aber unverändert gelassen.

Der ursprüngliche M-Befehl vom Kunden setzt dann nur die SRQ-Maske in IBIG50, die ihrerseits den M-Befehl auswertet.

### 4.15.2 Sonderbefehl Datenabfrage

Syntax	Antwort-String	Funktion
~T2	Sendedaten oder E1	Triggert ADDA, einen evtl. noch anstehenden Sendestring auszugeben.

Damit teilt IBIG50 im Mode2 der ADDA eine erfolgte Adressierung als Talker mit.

Da in ADDA normalerweise keine wartenden Sendedaten verfügbar sind, gibt ADDA den Fehlercode E1 aus (keine Daten verfügbar).

### 4.15.3 Sonderbefehl Trigger-on-Talk

Syntax	Antwort-String	Funktion	Bemerkung
~T1	z.B.: 1.2345E01VN	holt Messwert und gibt dann Ausgabestring zurück (je nach N-Befehl bzw. >KN)	ADDA ist im Standard Mode
	#n E11	Befehl nicht zulässig	ADDA ist im adressierbaren Modus

Damit teilt IBIG50 im Probus IV kompatiblen Mode1 der ADDA eine erfolgte Adressierung als Talker mit. Die ADDA gibt dann den mit dem N-Befehl vorgewählten Ausgabewert aus.

Der Befehl liefert eine Fehlermeldung, wenn ADDA im adressierbaren Mode ist.

## 5 Fehlercodes

Die Fehlercodes werden in der beschriebenen Form nur ausgegeben, wenn die ADDA im nicht adressierbaren Modus ist.

Läuft die ADDA im adressierbaren Modus, steht vor dem Fehlercode die Adresse der ADDA, die die Meldung gesendet hat. (z.B. "#1 E0").

Code	Bezeichnung	mögliche Ursache
E0	kein Fehler	normale Antwort auf Befehle
E1	keine Daten verfügbar	Kunde hat IBIG50 über GPIB als Talker adressiert und versucht Daten zu lesen, ohne dass vorher mit einem Abfragebefehl Daten bereitgestellt wurden. (IBIG50 hat ~T2 an ADDA gesendet)
E2	unbekannter Register-Typ	Nach dem '>' steht kein gültiger Registertyp.
E4	ungültiges Argument	Das Argument des Befehls wird nicht akzeptiert. Evtl. Wertebereich überschritten?
E5	Bereich überschritten	z.B. Sollwert größer als Typenspannung programmiert
E6	Register darf nur gelesen werden	Bestimmte Register können nur ausgelesen aber nicht beschrieben werden. z.B. Monitor-Register
E7	Receive Overflow	Der Befehlsstring war länger als 50 Zeichen.
E8	EEPROM ist schreibgeschützt	Es wurde versucht, Daten in ein Kalibrier-Register zu schreiben, wobei der entsprechende DIP-Schalter aber auf Schreibschutz stand.
E9	Adressfehler	Es wurde entweder ein nicht adressierter Befehl im adressierbaren Modus an die ADDA gesendet oder umgekehrt.
E10	unbekannter SCPI Befehl	Der SCPI Befehl ist nicht implementiert.
E11	Trigger on Talk Fehler	Es wurde versucht, über ~T1 Trigger on Talk zu starten, wobei die ADDA aber im adressierbaren Modus war.
E12	~Tn Befehl hatte ungültiges Argument	Es sind nur ~T1 und ~T2 zulässig.
E13	ungültiger N-Wert	Fehler wird erst bei Datenabfrage mit ? oder ~T1 gemeldet, wenn im Register K8 ein ungültiger N-Wert stand.
E14	Register darf nur beschrieben werden	Bestimmte Register können nur beschrieben aber nicht gelesen werden. z.B. >H0-Register
E15	String zu lang	z.B. Fabriknummer-String zu lang
E16	Checksumme falsch	Prüfsumme über den Befehlsstring ist falsch, siehe 3.4

## 6 Technische Daten

Befehlsverarbeitungszeit:	<1ms, typ. 300-600us (ab Abschlusszeichen des Befehlsstrings bis zum 1. Zeichen des Antwortstrings)
Mögliche Baudraten	4800Bd, 9600Bd, 19200Bd, 38400Bd, 115kBd 230kBd, 500kBd, 625kBd
Auflösung der Sollwerte:	Normalbetrieb: 16Bit + Vorzeichen Im hochauflösenden Modus: 22Bit + Vorzeichen (mit PWM Bit-Interpolation)
TK der Sollwerte	<10ppm/K, typ. 2-3ppm/K
Zeitliche Unsicherheit der Rampenfunktion	<1ms
Nutzbare Auflösung der Monitore:	effective number of bits, ENOB: 14 - 20Bit + Vorzeichen, je nach Messzeit und Daten des Netzgerätes
TK der Monitore	<10ppm/K, typ. 2-3ppm/K