

**Serielles Kommunikations-
Protokoll
für motrona-Geräte**

**Serial Communication
Protocol
for motrona Units**



<u>Inhaltsverzeichnis:</u>	Seite	<u>Table of contents:</u>	Page
1. Allgemeine Angaben	3	1. General Information	3
2. Geräte- Adressen	3	2. Device Addresses	3
3. Serielle Zugriffs-Codes	3	3. Serial Access Codes	3
4. Anforderung von Daten	4	4. Request for Data	4
5. Schreiben von Daten	5	5. Writing to Registers	5
6. Übertragung von Steuerbefehlen	7	6. Transmission of Control Commands	7
7. Praktisches Beispiel zur seriellen Kommunikation mit motrona-Geräten	7	7. Practical example for communication with motrona motion controllers	7
7.1 Daten-String	8	7.1 Data string	8
7.2 Bestätigung abwarten	9	7.2 Wait for acknowledgement	9
7.3 Weitere Parameter senden	9	7.3 Transmit more parameters	9
7.4 Daten aktivieren	9	7.4 Activate data	9
7.5 Daten im EEPROM speichern	9	7.5 Save data to EEPROM	9
8. Historie	10	8. History	10

1. Allgemeine Angaben

Das verwendete Protokoll entspricht dem LECOM- Standard (DIN ISO 1754), der häufig in der Antriebstechnik verwendet wird. Die Einstellung der gewünschten Baudrate, des Datenformats sowie die Anschlußbelegung des Datensteckers sind in der jeweiligen Bedienungsanleitung des verwendeten Gerätes beschrieben.

2. Die Geräte - Adressen

Es sind die Geräteadressen 11 bis 99 vorgesehen, die dem Gerät entweder über eine eingebaute Tastatur oder einen DIL-Schalter oder seriell vorgegeben und im EEPROM abgespeichert werden.

Adressen, die eine "0" enthalten, dürfen nicht verwendet werden, da diese als Gruppen- oder Sammeladresse interpretiert werden. Die Generaladresse "00" adressiert sämtliche angeschlossenen Geräte gleichzeitig, während die Gruppenadresse "10" alle Geräte mit den Nummern 11 bis 19 gleichzeitig anspricht und die Gruppenadresse "20" die Geräte Nr. 21 bis 29 adressiert usw.

Wenn die serielle Adresse eines Gerätes unbekannt ist, kann diese mit der SCAN-Funktion der zugehörigen motrona-Bediener- software ermittelt werden. Werksseitig sind alle motrona-Geräte auf die Adresse "11" eingestellt.

Es ist zu beachten, daß seitens des Gerätes jegliche Antwort oder Rückmeldung unterbleibt, sobald dieses über eine Sammeladresse angesprochen wird.

3. Serielle Zugriffs-Codes (Register Codes)

Für das Ansprechen der Codestellen in den einzelnen Geräten wird entweder die „Standard-Adressierung“ oder die „Erweiterte Adressierung“ verwendet, je nachdem, ob das Gerät nur über wenige oder sehr viele Codestellen verfügt. Die Bedienungsanleitung des entsprechenden Gerätes gibt Auskunft über die Codes und die Art der Adressierung

Die Erweiterte Register-Adressierung beginnt zur klaren Unterscheidung grundsätzlich mit einem Ausrufezeichen. Die Subcodes S1 und S2 haben grundsätzlich den Wert „0“, sofern in der Bedienungsanleitung des Gerätes nicht ausdrücklich anders angegeben.

1. General Information

The units use the LECOM protocol (DIN ISO 1745) which is a common standard with drive applications. Setting of the desired Baud rate and the data format as well as the pinout of the data connector are clearly described in the general operating instructions of the unit in use.

2. The Device Addresses

The protocol supports unit addresses between 11 and 99 which can be set either by a keypad or a DIL switch or by serial setup. The unit address will be stored in the EEPROM of the unit.

Addresses containing a "0" must not be used, because they will be interpreted as a collective address for several units. The general address "00" will talk to all units connected at this time. Addresses like "10" or "20" will talk to all units from 11 to 19 respectively from 21 to 29 etc.

Where the serial address of a unit should be unknown, the SCAN function of the corresponding motrona operator software will help you to find out. Ex factory, all motrona units are set to default address No. "11"

Please note that any response of a unit will be suppressed after access by the general address "00" or a collective address like "20".

3. Serial Access Codes (Register Codes)

For serial access to the registers within one unit the protocol uses either "Standard Addressing" or "Extended Addressing", depending on the total number of registers to be accessed.

Please see the operating instructions of the corresponding unit to find out the code assignments and the mode of register addressing

For reasons of clear differentiation, Extended register codes are always leaded by an exclamation mark.

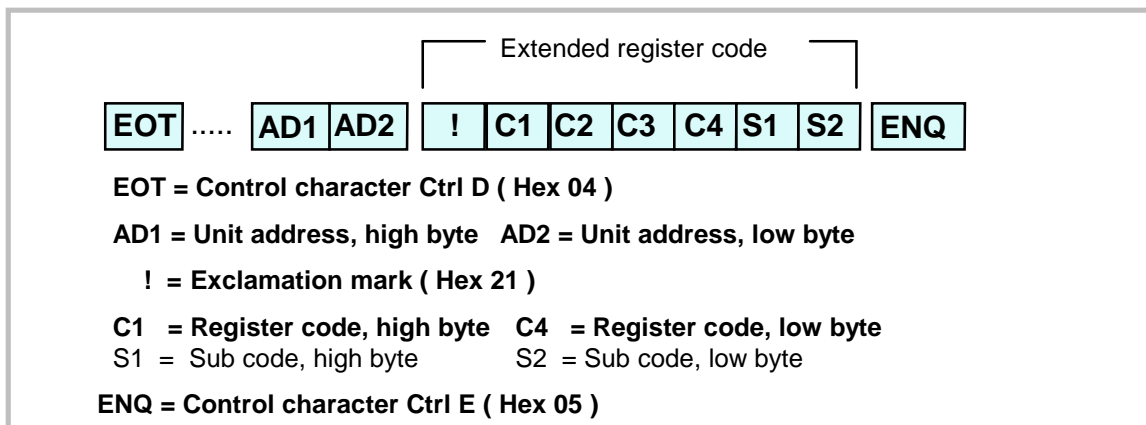
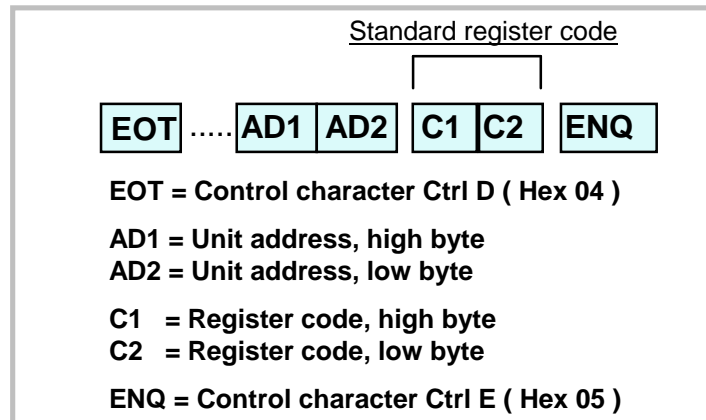
Sub codes S1 and S2 must always be "0", except other values are explicitly specified in the manual of the unit.

4. Anforderung von Daten

Zum Auslesen von Daten aus dem Arbeitsspeicher (RAM) des Gerätes gilt das folgende Anforderungs-Protokoll:

4. Request for Data

To read out data from the operational registers (RAM), the subsequent request string must be used:



Die zu verwendenden Register-Codes C1 und C2 sind aus der Parameter- Liste des jeweiligen Gerätes zu entnehmen.

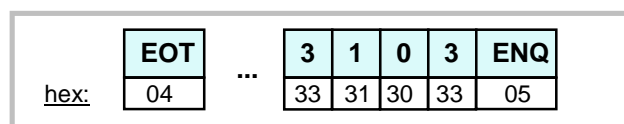
For the valid register codes C1 and C2, please refer to the parameter list of the unit in use.

Beispiel 1 (Standard-Adressierung):

Auslesen der Integralgeschwindigkeit "Int-Time" (Code 03) bei einem BY 150 Gleichlaufregler mit der Geräteadresse "31" :

Example 1 (Standard addressing):

Read the integration speed register "Int-Time" (register code 03) from a BY 150 synchro controller with device address "31"

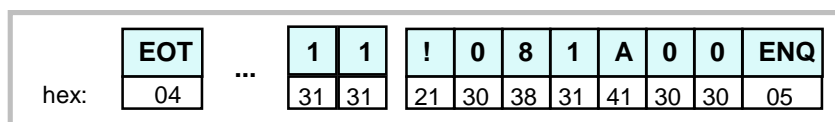


Beispiel 2 (Erweiterte Adressierung):

Auslesen der aktuellen Bahngeschwindigkeit (Code ! 081A) bei einem MC700-Regler mit Wickler-Firmware WR701 und der Geräte-Adresse "11":

Example 2 (Extended addressing):

Read the actual line speed (Code ! 081A) from a MC700 motion controller with Winding Firmware WR701 and the unit address "11"

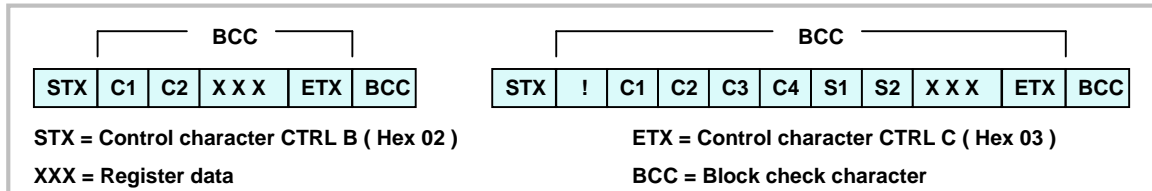


Bitte beachten Sie, dass neben den Ziffern 0–9 auch die Zeichen A-F zur Adressierung verwendet werden können, die mit den Hexadezimal-Codes 4.1 bis 4.6 ausgedrückt werden.

Bei richtiger Geräteadresse und gültigem Register-Code antwortet das Gerät je nach Adressierungs-Art mit einem der folgenden beiden Strings:

Please note that figures 0-9 and characters A-F may be used for register addressing, where A-F are expressed by hexadecimal codes 4.1 to 4.6.

With a correct unit address and a valid register code, the unit responds with one of the subsequent strings (depending on mode of addressing):



Die Stellenzahl der gesendeten Datenwerte "XXX" hängt vom jeweiligen Wert des angewählten Registers ab und kann bei negativen Werten auch ein Minuszeichen enthalten. Vornullen werden grundsätzlich unterdrückt und erscheinen nicht im Antwort-Telegramm.

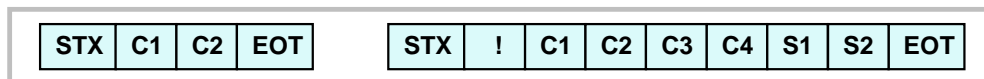
Der Block-Check-Character "BCC" ist das Exklusiv-Oder aller gesendeten Zeichen von „C1“ bzw. „!“ bis „ETX“ (jeweils einschließlich).

Wenn in der Anforderung ein ungültiger bzw. unbekannter Datencode (C1-C4 oder S1-S2) gesendet wurde, antwortet das Gerät nur :

The total number of data characters "XXX" depends on the actual numeric value of the selected data register and may also be leaded by a minus sign with negative values. Leading zeros are always suppressed and will not appear in the telegram.

The block check character "BCC" is generated by an Exclusive-OR over all characters between "C1" resp. "!" and "ETX" (both included)

If the request string contains an invalid or unknown register code (C1 – C4 or S1-S2), the response of the unit will just be :



Wenn sonstige Fehler im Anfrage-Protokoll festgestellt werden, antwortet das Gerät nur mit "NAK" (Hex 15)

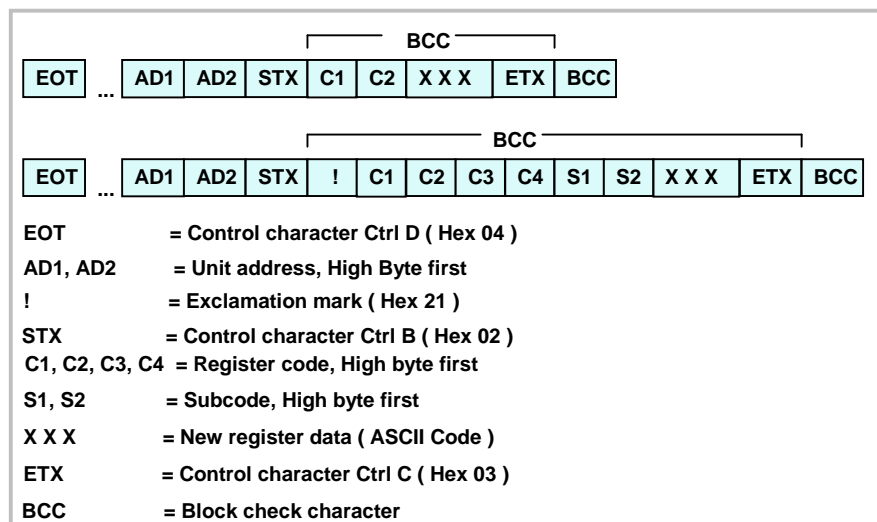
In case of other errors in the request string, the unit just responds by "NAK" (Hex 15)

5. Das Schreiben von Daten

Zur Veränderung von Parametern und Registerdaten durch einen Rechner wird, je nach Adressierungsart, eines der nachfolgenden Sendeprotokolle verwendet:

5. Writing to Registers

For modification of operational or other register data by a computer, one of the subsequent strings must be used, depending on addressing mode:



Die Länge des Datenstrings "XXX" ist hierbei beliebig und kann wahlweise auch Vornullen und ein negatives Vorzeichen enthalten. Der Block-Check-Character "BCC" ist wiederum das Exklusiv- Oder aller gesendeten Zeichen zwischen "C1" bzw. „!“ und "ETX". Bei korrekter Übertragung des Protokolls antwortet das Gerät mit "ACK", in allen anderen Fällen mit "NAK".

Die neu übertragenen Daten werden bei motrona-Geräten zuerst in einem Pufferspeicher hinterlegt, greifen also noch nicht in die Funktion des Gerätes ein ! Erst nach Übertragung des Befehls "Activate Data" werden die neuen Werte aktiv. Dadurch ist es möglich, bei laufender Maschine beliebig viele Parameter zu verändern, und dann mit einem einzigen Befehl alle neuen Vorgaben gemeinsam zu aktivieren.

Hinweise zur Speicher-Organisation :

Beim Auslesen von Parametern sendet das Gerät immer nur die aktuellen Daten aus dem Arbeitsspeicher (RAM). Parameterwerte, die zwar gesendet, aber noch nicht aktiviert wurden, können folglich nicht zurückgelesen werden

Nach Netzabschaltung wird automatisch der Parameter-Satz aus dem EEPROM in das RAM geladen. Seriell übertragene Daten sind verloren, wenn sie nicht vor Netzabschaltung aktiviert und anschließend im EEPROM gespeichert wurden.

The data string "XXX" can have any number of characters and may also contain leading zeros or a negative sign.

The block check character "BCC" represents again the Exclusive-OR function of all characters between "C1" resp. "!" and "ETX".

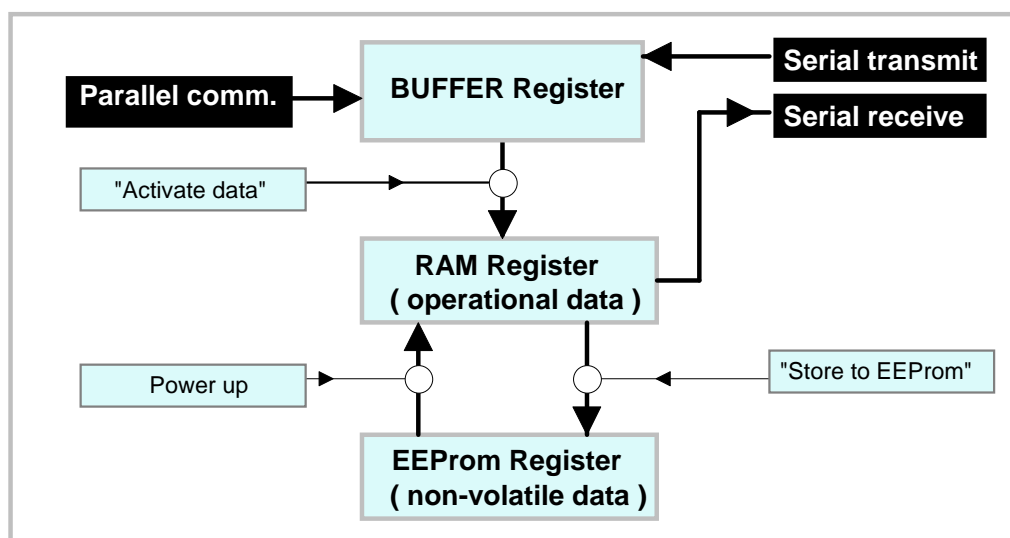
Upon a correct transmission of above protocol, the unit will respond by "ACK" and in all other cases by "NAK" .

With motrona units, all data from a serial transmission will first be latched in a buffer register and will not affect the current operation, until the command "Activate Data" has been transmitted. This procedure enables the user to prepare a complete new set of parameters in the background of the unit, with no effect to the production in progress, and to activate all at the same time by one single command.

Remarks to the Register Organisation

Any readout of parameters will read the current operational data from the RAM. Parameters, that have been written to the buffer memory only, but have not been activated yet, are not readable by serial request.

Upon power up, the unit automatically transfers the EEPROM data to the RAM register. Previous serial register modifications will be lost, unless these data have been activated first and then stored to the EEPROM.



6. Übertragung von Steuerbefehlen

Alle Steuerbefehle benutzen dasselbe Übertragungsformat wie in Abschnitt 4 gezeigt. Der Datenstring "XXXXX" besteht hier allerdings nur aus der Ziffer "1" zum Einschalten und der Ziffer "0" zum Ausschalten von Steuerkommandos. Einige der Befehle setzen sich unmittelbar nach Aktivierung automatisch auf "0" zurück (z.B. "Activate data" oder "Store to EEPROM". Andere Befehle bleiben statisch erhalten, bis sie über ein erneutes seriell Kommando wieder zurückgesetzt werden (z.B. Reset, Start/Stop, Trimm usw.)

Bezüglich der seriell aktivierbaren Befehle finden Sie alle Angaben in der Bedienungsanleitung des entsprechenden Gerätes.

Die serielle Aktivierung eines Steuerbefehls hat dieselbe Auswirkung, als wenn der entsprechende Hardware- Eingang auf "High" gelegt wird.

Anmerkung:

Hardware- Steuerbefehle und Software- Steuerbefehle arbeiten in einer logischen "ODER"-Verknüpfung. Es müssen daher sowohl der Hardware- Eingang als auch der zugehörige serielle Befehl gleichzeitig auf "0" gesetzt sein, damit der entsprechende Befehl deaktiviert ist.

Wenn z.B. seriell der Reset-Befehl auf "1" gesetzt wurde, ist das Gerät im Reset-Zustand, unabhängig davon ob der entsprechende Hardware- Eingang High oder Low ist. Erst wenn seriell das Reset-Kommando auf "0" zurückgesetzt wurde, ist der Hardware- Reset wieder voll funktionsfähig !

Bei Netzzuschaltung werden alle seriellen Befehle automatisch auf "0" gesetzt.

7. Praktisches Beispiel zur seriellen Kommunikation mit motrona-Geräten

Das nachfolgende Beispiel zeigt, wie der Parameter „Factor1“ bei einem BY125-Gleichlaufregler verändert wird.

BY125 benutzt die Standard-Adressierung für die seriellen Codestellen. Bei Geräten mit erweiterter Adressierung gilt das untenstehende Beispiel ebenso, bis auf die erweiterte Darstellung der Codestellen mit den Subcodes.

6. Transmission of Control Commands

All control commands are using the same transmission protocol like shown in section 4. However, the data string "XXXXX" uses only one character which is "1" to switch the function on and which is "0" to switch the function off again. Some of the serial commands will automatically reset to zero upon execution of the corresponding commands (i.e. "Activate data" or "Restore to EEPROM"). Others will need to be set and also to be reset by serial command (i.e. Reset, Start/Stop, Trim etc.)

For applicable command codes, please see the operating instructions of the unit you use.

All serial transmission of a control command will have the same result as when you set the appropriate hardware input to "High".

Important remark:

There is a logical "OR" condition between all hardware control inputs and their corresponding serial command flags. For this reason, it is essential to have the hardware input and the serial command set to OFF at the same time in order to switch a command OFF !

If, i.e. by serial communication, the Reset flag has been set to "1", the unit will be in it's Reset state, independant of what the logical level at the hardware reset input is. Only with the serial reset flag at "0" and the reset input at "Low", the unit will be in it's normal operative state !

Upon power up, all serial command flags will automatically be reset to "0" .

7. Paractical example for communication with motrona motion controllers

The following example describes how to transmit a new "Factor1" setting to a BY125 synchro controller.

BY125 uses standard addressing for register access. Where you use a unit with extended addressing, the subsequent example is fully valid,too, except of the extended version of address codes and sub codes.

Im Beispiel benutzen wir ein BY125-Gerät mit der **Geräteadresse „11“**. Dem Parameter „Factor1“ ist laut Gerätebeschreibung die serielle **Codestelle „00“** zugeordnet. Wir wollen Factor1 auf den **Wert „0.9873“** setzen.

*The example uses a BY125 controller with serial **unit address** “11”. According to the manual, register “Factor1” uses the serial **access code** “00”. The example shows how to set the register to a **value of “0.9873”***

7.1 Zuerst senden wir den Daten-String
bestehend aus insgesamt 13 ASCII-Zeichen:

7.1 First, we transmit the data string
consisting of totally 13 ASCII characters:

No.	Expression	ASCII	Hex	Binary code		Comment
				Hi-----	-----Lo	
01	EOT	EOT	04	0000	0100	Control character Initialisation
02	AD1	1	31	0011	0001	Address, high byte
03	AD2	1	31	0011	0001	Address, low byte
04	STX	STX	02	0000	0010	Control character
05	C1	0	30	0011	0000	Register code high byte
06	C2	0	30	0011	0000	Register code low byte
07	X (Data)	0	30	0011	0000	Factor, highest digit
08	X (Data)	9	39	0011	1001	
09	X (Data)	8	38	0011	1000	
10	X (Data)	7	37	0011	0111	
11	X (Data)	3	33	0011	0011	Factor, lowest digit
12	ETX	ETX	03	0000	0011	Control character
13	BCC	6	36	0011	0110	Block check character

Die Zeichen mit grauem Hintergrund werden mittels **Exklusiv-Oder** zur Bildung des Block-Check-Characters benutzt. Betrachten Sie bitte hierzu im Feld „**Binary Code**“ die **8 Spalten** (nur den grau unterlegten Bereich). In der höchstwertigen Spalte ganz links finden wir ausschliesslich Nullen. Das Exklusiv-Oder und damit das entsprechende Bit des BCC ist daher ebenfalls 0.

*Characters with gray background are used to form the block check character by an **exclusive- OR** function. Consider each of the **8 columns in the binary code field** (only gray rows). In the high bit column we find all zeros, therefore the exclusive OR is 0 and the high bit of the block check character must be zero at this position.*

In der kleinstwertigen Spalte ganz rechts finden wir die Bitfolge 0-0-0-1-0-1-1-1 (von oben nach unten). Das Exklusiv-Oder und damit das entsprechende Bit des BCC ist daher ebenfalls 0. In der Spalte links neben dem kleinsten Bit ergibt die EX-OR-Funktion den Wert 1

In the low bit column we find 0-0-0-1-0-1-1-1 (from up to down) and the exclusive OR is also „0“. Therefore the low bit of the block check character must be „0“ again at this position. In the column left of the lowest bit, the X-OR function results in a „1“ etc.

Allgemein gilt: Bei einer **geradzahligen** Anzahl von „1“ in einer Spalte ist das BCC-Bit an dieser Stelle „0“. Bei einer **ungeradzahligen** Anzahl von „1“ in einer Spalte ist das BCC-Bit an dieser Stelle „1“

In general we can say: when in one column we find an **even** number of „1“, the block check bit must be „0“ at this position. When we find an **odd** number of „1“, the block check bit must be „1“ at this position

Wenn wir nun die resultierenden Bits des BCC in einer Reihe lesen, ergibt sich damit die Binärzahl

0-0-1-1-0-1-1-0

So we find with our example, that the 8 Bits of the block check character, read as a **row**, are

0-0-1-1-0-1-1-0

as dem Hexadezimal-Code „36“ oder dem ASCII-Zeichen „6“ entspricht.

which equals to the Hexadecimal code “36” or to the ASCII character of „6“.

7.2 Bestätigung abwarten

Bei korrekter Übertragung wird das Gerät mit dem Bestätigungs-Zeichen „**ACK**“ antworten (Hex „06“, binär 0000 0110). Falls das Gerät stattdessen mit „**NAK**“ antwortet (Hex „15“), wurde die Sendung wegen eines Fehlers nicht akzeptiert (z.B. falscher BCC oder nicht korrekte Folge der Zeichen)

Falls das Gerät überhaupt nicht antwortet, war entweder der gesendete String unvollständig, oder serielle Grundeinstellungen wie Baudrate, Datenformat usw. sind nicht korrekt.

7.2 Wait for acknowledgement

*After correct transmission, the unit will acknowledge by responding „**ACK**“ (Hex „06“ or binary 0000 0110).*

*Where the unit instead should respond „**NAK**“ (Hex „15“), the transmission was rejected due to a fault like wrong BCC or incorrect sequence of characters etc.*

Where the unit does not respond at all, this indicates an uncomplete transmission string or wrong basic serial settings like Baud rate, Data format etc.

7.3 Weitere Parameter senden

Wir können nun bei Bedarf eine beliebige Anzahl weiterer Parameter übertragen, ohne damit den Maschinenablauf zu beeinflussen.

7.3 Transmit more parameters

We are free now to transmit any number of more parameters if necessary, without any effect to the function of the machine.

7.4 Daten aktivieren

Wenn alle Parameter übertragen sind, können wir die neuen Werte aktivieren, womit diese sofort im Maschinenablauf wirksam werden. Bei BY125 heisst dies, dass wir auf die Codestelle „67“ (C1 = 6 und C2 = 7) den Wert „1“ schreiben müssen.

7.4 Activate data

After successful transmission of all parameters we must activate the new settings, to make them effective to the operation of the machine. With BY125 units, we need to write a „1“ into the register with code „67“ (C1 = 6 and C2 = 7

No.	Expression	ASCII	Hex	Binary code Hi-----Lo	Comment
01	EOT	EOT	04	0 0 0 0 0 1 0 0	Control character Initialisation
02	AD1	1	31	0 0 1 1 0 0 0 1	Address, high byte
03	AD2	1	31	0 0 1 1 0 0 0 1	Address, low byte
04	STX	STX	02	0 0 0 0 0 0 1 0	Control character
05	C1	6	36	0 0 1 1 0 1 1 0	Register code high byte
06	C2	7	37	0 0 1 1 0 1 1 1	Register code low byte
07	X (Data)	1	31	0 0 1 1 0 0 0 1	Activate command „ON“
08	ETX	ETX	03	0 0 0 0 0 0 1 1	Control character
09	BCC	3	33	0 0 1 1 0 0 1 1	Block check character

7.5 Daten im EEPROM speichern (nur bei Bedarf)

Ohne Übertragung eines „Store“-Befehls wird das Gerät die zuletzt gesendeten und aktivierten Daten bis zur nächsten Unterbrechung der Stromversorgung nutzen.

Nach erneuter Zuschaltung der Stromversorgung werden jedoch wieder die Daten aus dem EEPROM hochgeladen.

Der serielle Code für den Speicherbefehl ist „68“ (C1 = 6, C2 = 8), und wiederum muss diese Codestelle mit dem Wert „1“ beschrieben werden.

7.5 Save data to EEPROM (option only)

Without sending this command, the controller will use all data which have been transmitted and activated, until the unit will be powered down.

At next power up, the unit however will restart with data loaded from it's EEPROM.

The serial register code for the „Store“ command is „68“ (C1 = 6, C2 = 8), and again the data string XXX must just be one digit with the value of „1“.

Hinweis:

Die Lebensdauer bei EEPROMs ist auf ca. 100 000 Speicherzyklen beschränkt, danach ist eine sichere Speicherung nicht mehr gewährleistet!.

Hint:

The life time of EEPROM memory chips is limited to a total number of about 100 000 storage cycles. After this, stored data may get lost.

8. Historie**8. History of Modifications**

Version	Name:	Date:	Modifications:
Serpro01a	hk	--	Original Version
Serpro02a	hk	07.04.2004	Extended register code addressing