

SERIELLES KOMMUNIKATIONSPROTOKOLL

Anzeigen Serien WAY

Weiterführende Informationen entnehmen Sie bitte dem entsprechenden Datenblatt und Handbuch unter <https://www.waycon.de/produkte/anzeigen-displays/>

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 den jeweiligen Handbüchern des verwendeten Gerätes beschrieben.

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 Bediener-Software ermittelt werden. Werksseitig sind alle 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.

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.

ANFORDERUNG VON DATEN

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

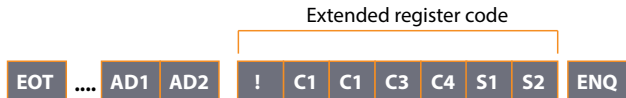


EOT = Control character Ctrl D (Hex 04)

AD1 = Unit address, high byte **AD2** = Unit address, low byte

C1 = Register code, high byte **C2** = Register code, low byte

ENQ = Control character Ctrl E (Hex 05)



EOT = Control character Ctrl D (Hex 04)

AD1 = Unit address, high byte **AD2** = Unit address, low byte

! = Exclamation mark (Hex 21)

C1 = Register code, high byte

C2 = Register code, low byte

S1 = Sub code, high byte

S2 = Sub code, low byte

ENQ = Control character Ctrl E (Hex 05)

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

Beispiel 1 (Standard-Adressierung):

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



Beispiel 2 (Erweiterte Adressierung):

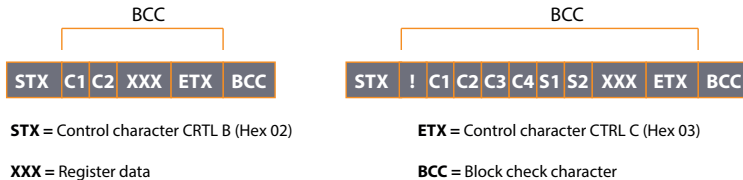
Auslesen der aktuellen Bahngeschwindigkeit (Code ! 081A) bei einem MC700-Regler mit Wickler-Firmware WR701 und der Geräte-Adresse „11“:



ANFORDERUNG VON DATEN

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:



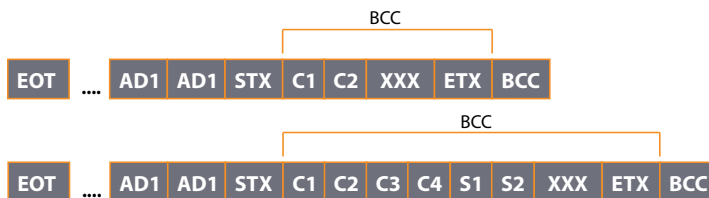
Die Stellenzahl der gesendeten Datenwerte „XXX“ hängt vom jeweiligen Wert des angewählten Registers ab und kann bei negativen Werten auch ein Minuszeichen enthalten. Vornullen werden grundsätzlich unterdrückt und erscheinen nicht im Antwort-Telegramm. Der Block-Check-Character „BCC“ ist das Exklusiv-Oder 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:



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

DAS SCHREIBEN VON DATEN

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



EOT = Control character Ctrl D (Hex 04)
AD1, AD2 = Unit address, High Byte first
! = Exclamation mark (Hex 21)
STX = Control character Ctrl B (Hex 02)
C1, C2, C3, C4 = Register code, High byte first
S1, S2 = Subcode, High byte first
XXX = New register data (ASCII Code)
ETX = Control character Ctrl (Hex 03)
BCC = Block check character

DAS SCHREIBEN VON DATEN

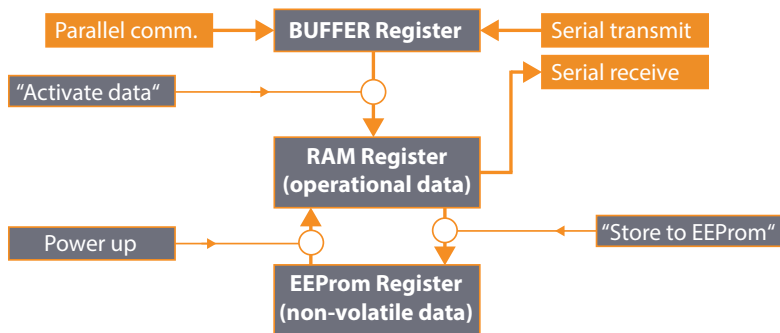
Die Länge des Datenstrings „XXX“ ist hierbei beliebig und kann wahlweise auch Vornullen und ein negatives Vorzeichen enthalten. Der Block-Check-Character „BCC“ ist wiederum das Exklusiv- Oder aller gesendeten Zeichen zwischen „C1“ bzw. „!“ und „ETX“. Bei korrekter Übertragung des Protokolls antwortet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

Die neu übertragenen Daten werden bei den 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.



ÜBERTRAGUNG VON STEUERBEFEHLEN

Alle Steuerbefehle benutzen dasselbe Übertragungsformat wie in Abschnitt „Anforderung von Daten“ gezeigt. Der Datenstring „XXXX“ 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.

ÜBERTRAGUNG VON STEUERBEFEHLEN

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.

PRAKTISCHES BEISPIEL ZUR SERIELLEN KOMMUNIKATION

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.

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.

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

No.	Expression	ASC II	Hex	Binary code Hi----- ----Lo	Comment
01	EOT	EOT	0 4	0000 0100	Control character Initialisation
02	AD1	1	3 1	0011 0001	Address, high byte
03	AD2	1	3 1	0011 0001	Address, low byte
04	STX	STX	0 2	0000 0010	Control character
05	C1	0	3 0	0011 0000	Register code high byte
06	C2	0	3 0	0011 0000	Register code low byte
07	X (Data)	0	3 0	0011 0000	Factor, highest digit
08	X (Data)	9	3 9	0011 1001	
09	X (Data)	8	3 8	0011 1000	
10	X (Data)	7	3 7	0011 0111	
11	X (Data)	3	3 3	0011 0011	Factor, lowest digit
12	ETX	ETX	0 3	0000 0011	Control character
13	BCC	6	3 6	0011 0110	Block check character

PRAKTISCHES BEISPIEL ZUR SERIELLEN KOMMUNIKATION

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.

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.

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

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

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.

3. Weitere Parameter senden

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

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.

No.	Expression	ASCII	Hex	Binary code Hi-----Lo	Comment
01	EOT	EOT	0 4	0000 0100	Control character Initialisation
02	AD1	1	3 1	0011 0001	Address, high byte
03	AD2	1	3 1	0011 0001	Address, low byte
04	STX	STX	0 2	0000 0010	Control character
05	C1	6	3 6	0011 0110	Register code high byte
06	C2	7	3 7	0011 0111	Register code low byte
07	X (Data)	1	3 1	0011 0001	Activate command „ON“
08	ETX	ETX	0 3	0000 0011	Control character
09	BCC	3	3 3	0011 0011	Block check character



PRAKTISCHES BEISPIEL ZUR SERIELLEN KOMMUNIKATION

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.

Hinweis:

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