

Anwendung

Das SET-GLT Protokoll wird zum Datenaustausch zwischen einer oder mehreren MC 2001 Regeleinheiten und einer externen Leittechnik-Steuereinheit benutzt.

Die Datenübertragung erfolgt im Master-Slave Betrieb, wobei die Leittechnik-Steuereinheit den Master und die Regeleinheit(en) den(die) Slave(s) darstellen. Mehrere Slaves werden durch ihre Leittechnik-Adresse voneinander unterschieden. Aus Kompatibilitätsgründen muss auch bei Verwendung von nur einem einzigen Slave eine Adresse benutzt werden.

Die Daten werden nach Aufforderung durch den Master vom Slave als einzelne Werte gesendet, wobei keine selektive Auswahl stattfindet. Um effizient zu sein, werden immer nur geänderte Werte übertragen. Der Master kann einzelne Werte neu definieren, sofern sie für Schreiben vorgesehen sind.

Slave-Adressierung (<adr>)

Die intern benutzte Leittechnik-Adresse (000 bis 200) wird zur Übertragung auf dem Bus um 48 (30h) erhöht und dann als 1 Byte (ASCII „0“ bis {F8}) übertragen. Zu beachten ist jedoch, dass an einem Bus nur maximal 16 Slaves zulässig sind.

Datenumfang

Die Protokolldefinition erlaubt jeweils 100 Adressen für analoge, digitale und ganzzahlige (integer) Daten, somit können insgesamt 300 Werte ausgetauscht werden. Die Zuordnung der Datenadressen zu den Werten ist im Steuerungsprogramm definiert und in einer Adresstabelle dokumentiert.

Darstellung der Daten (<typ> und <wert>)

Analog- und Integerwerte werden in hexadezimaalem Format als jeweils 4 ASCII-codierte Bytes, Digitalwerte als 1 Byte übertragen. Für die Skalierung der Werte gilt folgende Tabelle:

Datentyp	<typ>	Bytes	Wertebereich	Skalierung	Beispiel
Analog	A	4	-32768 ... +32767	1 Nachkommastelle, 2er Komplement	10,0 = 0064h 28,8 = 0120h - 0,1 = FFFFh - 3276,8 = 8000h
Digital	D	1	0, 1	0...Falsch, 1...Wahr	
Integer	I	4	0 ... 65535	ganzzahlig, entsprechend der Verwendung im Programm	100 = 0064h 288 = 0120h 65535 = FFFFh

Darstellung der Datenadresse (<nr>)

Die jeweilige Datenadresse liegt im Zahlenbereich 1 bis 100. Zur Übertragung wird ihre dezimale Adresse um 48 (30h) erhöht und dann als einstelliger Zeichencode verwendet. Damit entsteht folgende Abbildung:

1 ... 100 → 49 ... 148 entsprechend 31h ... 94h („1“ ... „ö“)

Prüfsumme (<chk>)

Sofern bei der Datenübertragung eine Prüfsumme verlangt ist, wird sie folgendermaßen gebildet:

Alle zur Prüfsummenbildung benutzten Bytes werden modulo 256, also als 1 hexadezimales Byte, addiert (ein Übertrag entfällt). Das entstehende Summenbyte wird in die Einzelziffern aufgespaltet, jede Hexadezimalziffer wird um 48 (30h) erhöht und die Prüfsumme dann zweistellig übertragen, wobei folgende Abbildung gilt:

0 ... Fh → 30h ... 3Fh entsprechend den ASCII-Zeichen „0“ bis „?“

Beispiel: Die Bytefolge „02h 31h 44h 63h 31h 03h“ ergibt als Bytesumme 10Eh, dies wird abgeschnitten zu 0Eh, der Übertrag entfällt und die Prüfsumme wird als 30h 3Eh oder „0>“ übertragen.

Übertragungssteuerung

Jede Übertragung wird vom Master gestartet und mit einem Steuerkommando eingeleitet, bei Bedarf mit Daten versehen und kommandoabhängig vom Slave beantwortet und am Ende vom Master und/oder Slave quittiert. Steuerkommandos sowie Antwort- oder Quittungscodes sind ASCII-Zeichen aus dem Bereich der ersten 32 Bytecodes, die an keiner anderen Stelle auftreten. Folgende Zeichen werden benutzt:

Name	Code	Typ*	Verwendung
NUL	00h	A	Slave: keine Daten verfügbar
STX	02h	K/A	Master/Slave: Daten folgen
ETX	03h	(K/A)	Master/Slave: Daten-Ende, Prüfsumme folgt; (kein eigenständiges Kommando)
ENQ	05h	K	Master: Datenanforderung
ACK	06h	Q	Master/Slave: Übertragung i.O.
BEL	07h	Q	Master/Slave: Übertragungsfehler

*) A...Antwort, K...Kommando, Q...Quittung

Nachstehend sind die benutzten Kommandofunktionen einzeln aufgelistet. Die beigefügten Beispiele (grau unterlegt) benutzen die Slave-Adresse "1". Zu jedem Kommando bzw. jeder Antwort sind Beispiele als ASCII-Strings enthalten. Nicht darstellbare ASCII-Zeichen sind dabei als zweistelliger Binärwert in geschweiften Klammern dargestellt (z.B. {02}).

Funktion: Abfrage von Daten durch den Master:

1. Master sendet: **<ENQ> <adr>** {05}1
2. Slave antwortet: **<STX> <adr> <typ> <nr> <wert> <ETX> <chk>**
z.B. für Inhalt von Puffer A27 als Wert 277 : {02}1AK0115{03}89
oder: **<NUL>** wenn keine Nachrichten im Puffer liegen {00}
3. Master bestätigt: **<ACK>** wenn Übertragungsprüfung erfolgreich, {06}
sonst erfolgt keine Antwort

Nach Bestätigung durch den Master (<ACK>) löscht der angesprochene Slave die Nachricht aus seinem Datenpuffer und sendet bei der nächsten Abfrage die nächste Nachricht usw., bis alle Nachrichten gesendet wurden. Danach antwortet der Slave auf jede Anfrage sofort mit <NUL>, bis wieder aktualisierte Nachrichten in seinem Puffer liegen.

