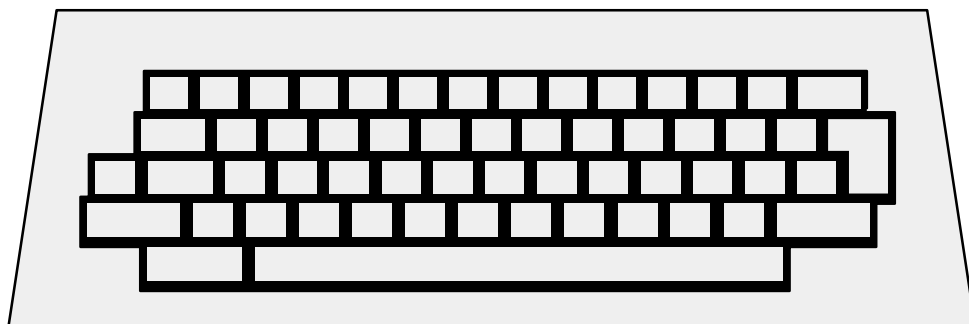
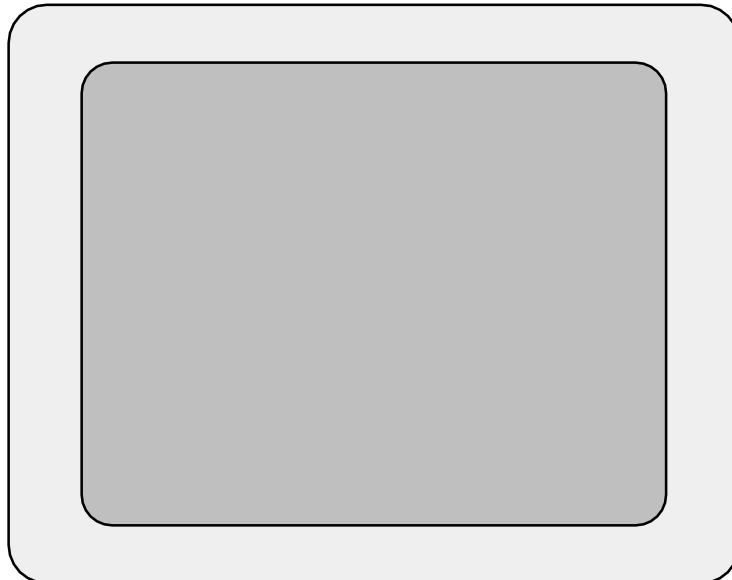


Die Beschreibung der Bedien- und Testfunktionen, Monitorfunktionen und Speicherübersichten ist in der Systembeschreibung Advant Controller 31 (Griff 7), Bestell-Nr. 1SAC 1316 99 R0101 enthalten. In dieser PDF-Dokumentation ist diese Beschreibung hier mit eingefügt.



Advant Controller 31

Dezentrale, intelligente Automatisierungstechnik

Programmierung

Programmier- und Testhilfsmittel **7.3**
Zugang, Bedien- und Testfunktionen
Monitorfunktionen
Speicherübersichten
Funktionen in der Anweisungsliste

Inhaltsverzeichnis, Griff 7.3

1	Programmier- und Testhilfsmittel	1
1.1	Programmiersoftware 907 PC 33	1
1.2	Programmierung über ARCNET	4
2	Zugang, Bedien- und Testfunktionen	5
2.1	Einführung	7
2.2	Operanden	9
2.3	Serielle Schnittstelle COM1	14
2.4	Bedien- und Testfunktionen	16
3	Monitorfunktionen	41
4	Speicherübersichten	50
5	Funktionen in der Anweisungsliste	54
5.1	Texte in der AWL	54
5.2	Syntaxdiagramme für AWL	56

Anmerkung:

Ein detailliertes Inhaltsverzeichnis befindet sich, soweit notwendig, bei den einzelnen Kapiteln.

1 Programmier- und Testhilfsmittel

1.1 Programmiersoftware 907 PC 33

Allgemeines

Die Programmier- und Testsoftware 907 PC 33 ist als

- Programmier- und Testsoftware 907 PC 331 für ABB Procontic CS31 / Advant Controller 31 (Zentraleinheiten 07 KR 31, 07 KR 91, 07 KT 92, 07 KT 93, 07 KT 94) und ABB Procontic T200 (Kommunikationsprozessor 07 KP 62), Bestell-Nummer GJP5204500R0102
- Programmier- und Testsoftware 907 PC 332 für ABB Procontic T200, Bestell-Nummer GJP5204300R0102

erhältlich. Die Software-Produkte 907 PC 331 und 907 PC 332 werden jeweils inclusive systemspezifischer Dokumentation ausgeliefert.

Die für die beiden Software-Produkte identischen Funktionen sind in der Dokumentation "Allgemeiner Teil 907 PC 33" beschrieben. Dieser Dokumentationsordner kann unter der Bestell-Nummer GJP5203900R0102 separat bestellt werden.

Die Software ist lauffähig auf IBM/AT-kompatiblen Personal-Computern. Ein weitgehend automatisch arbeitendes Installationsprogramm installiert das Softwarepaket 907 PC 33 auf diesem Gerät oder auf einem anderen IBM/AT-kompatiblen Personal-Computer.

Die Programmier- und Testsoftware 907 PC 33 gestattet eine einfache und wirtschaftliche Programmierung von SPS-Programmen in:

- Funktionsplan (FUP)
- Kontaktplan (KOP)
- Komfortabler Anweisungsliste (Komf. AWL)

Die Programmeingabe ist sowohl symbolisch als auch absolut möglich. Das SPS-Programm wird ergänzt durch symbolische Bezeichner, Langtext und Kommentar. Jederzeit aufrufbare Hilfs- und Fehlermeldungen sowie eine Syntaxprüfung erleichtern die Programmeingabe. Die Erstellung als FUP oder als KOP erfolgt in einem gemeinsamen Editor. Dadurch sind Elemente aus FUP und KOP mischbar und können miteinander verbunden werden. Die Bibliothek enthält zahlreiche Verknüpfungselemente und Funktionsbausteine, die die Realisierung von komplexen Funktionen wesentlich vereinfachen (z. B. PID-Regler).

Merkmale

Der Umfang der aufgeführten Merkmale ist abhängig von den Fähigkeiten der einzelnen Steuerungen.

Allgemeine Merkmale

- Alle Funktionen können mit der Maus gesteuert werden.
- Übersichtliche Darstellung von Projektdaten und Programmkonfiguration auf einen Blick
- Scrollfähigkeit in alle Richtungen in den Editoren
- Automatisches Erkennen von Änderungen

Menü-Führung

- Moderne, übersichtlich gegliederte Menüoberfläche in Pop-up-Menü-Technik
- Darstellung in Farbe
- Schnelle Anwahl der Menüpunkte mit der Maus oder mit der Tastatur
- Aufruf externer Programme auf DOS-Ebene direkt aus dem Menü (DOS-Shell)

Pfadangaben

- Eingabe eines Dateinamens mit dazugehörigem DOS-Pfad
- Anzeige der Projektübersicht mittels Dateiverzeichnis
Übernahme des Dateinamens inkl. Pfad, Auswahl mit Cursor.

Paßwort-Schutz

- Zugriffsberechtigung in mehreren Privilegienstufen

Datensicherung

- Datensicherung direkt aus dem Editor
- Datensicherung von ganzen Projekten auf Diskette

Modularisierung

- Handhabung von Großprojekten
- Gliederung der Projekte in logische Strukturen
- Aufteilung in Programm- und Variablenmodule
- Modulwechsel innerhalb des FUPKOP und der Komfortablen AWL möglich
- Module lassen sich aus allen Ebenen aufrufen (Gesamtprojekt-/Programmmodul-Funktionsauswahl)
- Vereinfachte Eingabe vom Modulnamen und zugehörigem Dateinamen

Teilpläne

- Durch Aufteilung der Programme bzw. Programmmodule in Teilpläne erreicht man gute Programmierübersicht.
- Einfache Verwaltung durch Teilplanname und Teilplannummer

FUPKOP-Editor

- Einheitlicher Editor für die Programmierung mit grafischen Symbolen als Funktionsplan und als Kontaktplan
- Verbindung von Kontaktplannetzwerken mit Elementen des Funktionsplanes

Komfortabler AWL-Editor

- Darstellung mit Symbolen und Langtext in verschiedene Formen
- Auswahl von Verknüpfungselementen über ein Auswahlmenü mit der Maus
- Integration der AWL-Fähigkeiten in der Komfortablen AWL
- Entfall der Übersetzung, wenn die Komfortable AWL keine VEs enthält
- Die übersetzte AWL kann angezeigt werden, wobei die Teilplanstruktur erhalten bleibt. Auch in Online möglich.

Editor-Funktionen

Zur Programmerstellung steht ein umfangreiches Befehlsspektrum in den Editoren zur Verfügung:

- Syntaxprüfung und Plausibilitätsprüfung bei der Eingabe von Variablen
- Blockbefehle
 - zur Bearbeitung von Programmteilen und Variablen
 - löschen
 - verschieben
 - kopieren
 - abspeichern
 - laden
 - drucken
 - unbenutzte Variablen löschen
- Suchbefehle
 - nach Satznummer
 - nach Wortnummer
 - nach Variable
 - nach Symbol
 - nach Befehl
 - nach Zeilennummer
 - wiederholen
 - nach Teilplan
 - nach VE
 - nach nicht belegtem Anschluß

- Suchen und Ersetzen

- Einfügen

- Löschen

Bibliothek

- Bedienoberfläche mit Mausunterstützung
- Programmierung eines Verknüpfungselements (VE) im FUPKOP
- Hierarchische Gliederungsmöglichkeit der VEs (ähnlich DOS-Verzeichnissen)
- Hilfstexte und Kurzkomentare für VEs
- Anschlußbelegungsprüfung zum frühzeitigen Erkennen von Programmierfehlern.
- Zu jedem Hersteller-VE ist eine ausführliche Funktionsbeschreibung als Hilfstext auch direkt aus dem FUP aufrufbar.

Variablen-Editor

- Komplette Liste aller eingegebenen Variablen
- Wählbare Sortierung nach absoluten oder symbolischen Variablen
- Variablen können wahlweise ein- oder mehrfach Symbolnamen zugeordnet werden
- Übernahme und Übergabe der Variablenlisten von und nach beliebigen Textverarbeitungssystemen
- Bereitstellung und Übernahme der Variablenlisten für bestimmte CAD/CAE-Systeme

Text-Editor

- Eingabe von beliebigen ASCII-Dateien, maximal 255 Zeichen pro Zeile

Kommentare

- Verbale Beschreibung von Netzwerken oder Programmsegmenten

ONLINE-Funktionen

Zahlreiche ONLINE-Funktionen unterstützen den Anwender bei der Inbetriebnahme durch Möglichkeit zu

- Statusanzeige in
 - Funktionsplan
 - Kontaktplan
 - Anweisungsliste
 - Variablenliste
- Programm-
 - Übertragung
 - Start
 - Abbruch
 - Stopp
 - Fortsetzung
 - Zustand
- Einzelzyklus ein/aus
- Einzelschritt ein/aus
- Breakpoint
 - setzen
 - anzeigen
 - Verfolgung über das gesamte Programm hinweg
 - löschen
- Triggern
 - auf Zeit
 - Variable
- Überschreiben
- Tippen
- Forcen
- Ändern von
 - Zeit- und Zählersollwerten
 - Variablen
 - Programmteilen
- Online-Programmänderung

Darüber hinaus lassen sich ausgewählte Variablen in ONLINE-Listen zusammenfassen und deren Zustand auf dem Bildschirm anzeigen.

- "Hotkeys" für schnellere Bedienung
- Umschalten in den ONLINE-Betrieb direkt aus dem FUPKOP, aus der Komfortablen AWL, Variablenliste, ONLINE-Liste.
- Direkte SPS-Kommunikation, z. B. "Programm senden" aus den Editoren
- Programm-Änderungen übersetzen und übertragen mit einem Tastendruck

- Direktes "Überschreiben" und "Forcen" aus den Editoren
- ONLINE-Liste mit direkter Übernahme von Variablen aus dem SPS-Programm zu "Forcen", "Überschreiben"
- Einfaches Setzen von Breakpoints mit Cursor auch im FUP
- ONLINE-Darstellung von Variablen in verschiedenen Zahlenformaten (dezimal, oktal, hexadezimal, binär).

Programmdokumentation

Die automatische Programmdokumentation umfaßt die Ausgabe folgender Listen auf dem Drucker:

- Funktionsplan
- Anweisungsliste
- VE-Bibliothek
- Logikplan
- Kontaktplan
- Variablenliste
- Querverweisliste
- Kommentarliste
- ONLINE-Liste
- Textseite
- Datenbereich
- Modularisierungsliste
- Gesamt-Variablenliste
- Gesamt-Querverweisliste
- Systemkonfiguration

Anpassung der Ausgabe an beliebige Drucker

Druckformat-Editor

Ein spezieller Druckformat-Editor erlaubt die Ergänzung der jeweiligen Listen durch einen individuellen Schriftkopf und Schriftfuß. In diesem Schriftkopf bzw. Schriftfuß können automatisch bestimmte Daten mit ausgegeben werden, wie z. B. Name der Projektdatei, Datum und Uhrzeit.

1.2 Programmierung über ARCNET

Es ist möglich, auf DOS-Ebene Steuerungen der AC31-Serie über ARCNET zu programmieren. Hierzu sind erforderlich:

- PC mit eingebauter ARCNET-Karte oder Koppler, der an eine parallele Druckerschnittstelle angeschlossen wird
- AC31-Zentraleinheiten mit integriertem ARCNET-Koppler
- Spezielle Treiber-Software
- 907 PC 331

Weitere Informationen siehe Beschreibung 907 PC 331 R0402.

2 Zugang, Bedien- und Testfunktionen

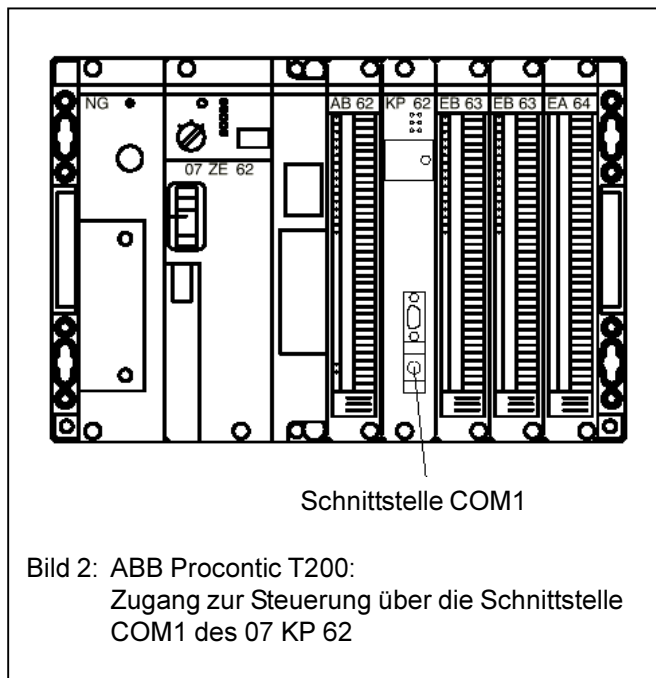
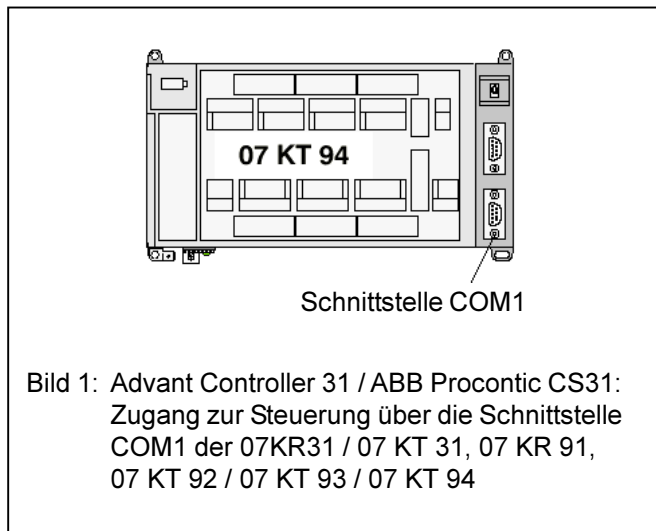
Inhaltsverzeichnis des Kapitels Zugang, Bedien- und Testfunktionen

2.1 Einführung	7	2.2 Operanden	9
2.1.1 Zugang zu den Zentraleinheiten 07 KR/KT 31 07 KR 91, 07 KT 92, 07 KT 93, 07 KT 94 und zum Kommunikationsprozessor 07 KP 62 der ABB Procontic T200	7	2.2.1 Operanden der 07 KT 94, 07 KT 93 R202/262 07 KT 92 R303/363	9
2.1.2 Schnittstellennorm	7	2.2.1.1 Verfügbare Variablen und Konstanten	9
2.1.3 Schnittstellenbetriebsart	7	2.2.1.2 Direkte Konstanten	10
2.1.4 Systemverhalten der SPS	7	2.2.1.3 Marken	10
2.1.5 Synchronisation des Datenaustausches	8	2.2.1.4 Systemkonstanten	10
2.1.6 Echo	8	2.2.1.5 Systemmerker / Diagnosemerker	11
2.1.7 Abbruch einer Zeichenausgabe	8	2.2.1.6 CS31-Statuswort	11
2.1.8 Fertigmeldung	8	2.2.2 Operanden der 07 KR 31 und 07 KT 31	11
2.1.9 Fehlermeldung	8	2.2.2.1 Verfügbare Variablen und Konstanten	11
2.1.10 Hinweise zur Implementierung	8	2.2.2.2 Direkte Konstanten	12
		2.2.2.3 Marken	12
		2.2.2.4 Systemkonstanten	12
		2.2.2.5 Diagnosemerker	12
		2.2.2.6 CS31-Statuswort	12
		2.2.3 Operanden des 07 KP 62	12
		2.2.3.1 Verfügbare Variablen und Konstanten	12
		2.2.3.2 Direkte Konstanten	13
		2.2.3.3 Marken	13
		2.2.3.4 Systemkonstanten	13
		2.2.3.5 Diagnosemerker	13
		2.3 Serielle Schnittstelle COM1	14
		2.4 Bedien- und Testfunktionen	16

2.1 Einführung

2.1.1 Zugang zu den Zentraleinheiten 07 KR/KT 31, 07 KR 91, 07 KT 92, 07 KT 93, 07 KT 94 und zum Kommunikationsprozessor 07 KP 62 der ABB Procontic T200

Der Zugang zu den AC31/CS31 Zentraleinheiten (07 KR 31, 07 KR 91, 07 KT 92 bis 07 KT 94) und zum Kommunikationsprozessor 07 KP 62 der ABB Procontic T200 erfolgt über die serielle Schnittstelle COM1.



Jede Bedien- und Testfunktion der SPS ist über ein ASCII-Klartexttelegramm aufrufbar. Bei der seriellen Schnittstelle muß dazu die Betriebsart "Aktiv-Mode" eingestellt sein.

Anschließbare Geräte:

- Terminal im VT100-Mode
- Rechner mit VT100-Emulation
- Rechner mit einem Programm zur Handhabung der Klartexttelegramme der Bedien- und Testfunktionen

2.1.2 Schnittstellenorm

Schnittstellenorm: EIA-RS232

2.1.3 Schnittstellenbetriebsart

Zur Nutzung der Bedien- und Testfunktionen muß die serielle Schnittstelle COM 1 auf die Betriebsart "Aktiv-Mode" eingestellt sein.

RUN/STOP-Schalter in Stellung: STOP

In der Schalterstellung STOP stellt die SPS an COM 1 grundsätzlich die Betriebsart "Aktiv-Mode" ein.

RUN/STOP-Schalter in Stellung: RUN

In der Schalterstellung RUN ist an COM 1 die Betriebsart "Aktiv-Mode" eingestellt, wenn eine der beiden folgenden Bedingungen erfüllt ist:

- Systemkonstante KW 00,06 = 1

oder

- Systemkonstante KW 00,06 = 0 **und** Pin 6 an COM1 hat 1-Pegel (1-Pegel an Pin 6 wird eingestellt durch Verwendung des Systemkabels 07 SK 90 oder dadurch, daß Pin 6 nicht angeschlossen wird.)

2.1.4 Systemverhalten der SPS

Es gilt:

Die Bearbeitung des SPS-Programmes hat höhere Priorität als die Kommunikation über die seriellen Schnittstellen.

Die SPS bedient die serielle Schnittstelle COM1 in Empfangsrichtung interruptgesteuert. Während eines laufenden SPS-Programm-Zyklus ankommende Zeichen lösen jeweils einen Interrupt aus, der die Bearbeitung des SPS-Programmes solange unterbricht, bis das empfangene Zeichen im Empfangsbuffer abgelegt ist. Um eine permanente Unterbrechung der Programmbearbeitung zu vermeiden, steuert die SPS über die RTS-Leitung den *Datenempfang* so, daß dieser in den Pausen zwischen zwei SPS-Zyklen stattfindet.

Die SPS *bearbeitet* die über COM1 erhaltenen Aufträge ausschließlich *in den Pausen* zwischen den SPS-Programmzyklen. Die Ausgabe von Zeichen über COM1 erfolgt ebenfalls nur in den Pausen zwischen zwei Programmzyklen. Je niedriger die Auslastung der SPS ist, desto länger sind die Pausen zwischen den Programmzyklen und desto höher ist die mögliche Kommunikationsleistung an COM1.

2.1.5 Synchronisation des Datenaustausches

Die Synchronisation des Datenaustausches zwischen der Steuerung und dem angeschlossenen Gerät erfolgt über die Hardware-Handshake-Leitungen RTS und CTS.

Die SPS sperrt über die RTS-Leitung den Datenempfang unter folgenden Randbedingungen:

- Empfangsbuffer hat einen bestimmten Füllstand erreicht.
- Es läuft gerade ein SPS-Programm-Zyklus

Die Steuerung *reagiert* bei der Ausgabe von Zeichen zusätzlich noch auf XOFF/XON-Zeichen des angeschlossenen Gerätes. Die Steuerung selbst benützt diese SW-Handshakes *nicht*.

2.1.6 Echo

In den Pausen zwischen zwei SPS-Programmzyklen bearbeitet die SPS die Aufträge, die sich im Empfangsbuffer angesammelt haben. Dazu werden die Zeichen von der SPS aus dem interruptgesteuerten Empfangsbuffer ausgelesen, sofort über COM 1 geechot, auf syntaktische Richtigkeit überprüft und dann verarbeitet. Die Zeichen werden in derselben Reihenfolge geechot, wie sie über COM 1 empfangen wurden.

2.1.7 Abbruch einer Zeichenausgabe

Durch Senden von <CTRL C> kann das angeschlossene Gerät die SPS veranlassen, die gerade laufende Zeichenausgabe abbrechen. Die zugehörige Bedien- oder Testfunktion wird dadurch ebenfalls abgebrochen.

2.1.8 Fertigmeldung

Nach der vollständigen Bearbeitung einer Bedien- oder Testfunktion ist die SPS wieder bereit für einen neuen Auftrag. Diese Bereitschaft wird durch Ausgabe der ASCII-Zeichen

CR LF > (0D_H 0A_H 3E_H)

über COM 1 signalisiert.

Das bedeutet, daß das ASCII-Zeichen > (Größer-Zeichen) an den Anfang einer neuen Zeile gesetzt wird.

2.1.9 Fehlermeldung

Erhält die SPS einen unzulässigen oder syntaktisch unkorrekten Auftrag, so wird dies wie folgt über COM 1 signalisiert:

<# Fehlermeldung als Klartext>

Anschließend wird die Bereitschaft für einen neuen Auftrag durch Ausgabe der ASCII-Zeichen

CR LF > (0D_H 0A_H 3E_H)

über COM 1 signalisiert.

Das bedeutet, daß das ASCII-Zeichen > (Größer-Zeichen) an den Anfang einer neuen Zeile gesetzt wird.

2.1.10 Hinweis zur Implementierung

Sollen die Bedien- und Testfunktionen der SPS von einem an COM 1 angeschlossenen Rechner aufgerufen werden, so können diese Funktionen zunächst problemlos mit einem Terminal im VT100-Mode ausprobiert werden.

Werden die Bedien- und Testfunktionen zur Mensch-Maschine-Kommunikation (MMK) benutzt, dann werden vorwiegend die folgenden Funktionen benötigt:

- **Variable / indirekte Konstante überschreiben**
Y-Befehl
- **Status von Variable anzeigen**
Z-Befehl
ZO-Befehl
ZD-Befehl
ZZ-Befehl

Wird an COM1 ein Rechner angeschlossen, so ist der ZZ-Befehl empfehlenswert. Beim ZZ-Befehl schickt die SPS keine ESC-Sequenzen zur Cursor-Steuerung.

- **Werte von indirekten Konstanten eingeben/ändern**
K-Befehl

2.2 Operanden 07 KT 94 (Variablen und Konstanten)

Bei der Mensch-Maschine-Kommunikation spielt das Anzeigen und Ändern von Operanden eine große Rolle. Aus diesem Grund wird hier ein Überblick über alle Operanden der SPS gegeben.

2.2.1 Operanden der Zentraleinheit 07 KT 94

2.2.1.1 Verfügbare Variablen und Konstanten

Eingänge

E 00,00...E 61,15 : Digitale Eingänge, CS31-Vor-Ort-Module
E 62,00...E 63,15 : Digitale Eingänge der Zentraleinheit 07 KT 94
E 64,00...E 64,07 : Digitale Eingänge der Zentraleinheit 07 KT 94 (gebildet aus EW 06,00...EW 6,07)

E 65,00...E 99,15 : reserviert
E 100,00...E 163,15 : reserviert
E 200,00...E 263,15 : reserviert

EW 00,00...EW 05,15 : Analog-Eingänge, CS31-Vor-Ort-Module
EW 06,00...EW 06,07 : Analog-Eingänge der Zentraleinheit 07 KT 94
EW 07,00...EW 07,07 : reserviert
EW 07,08...EW 07,14 : Echtzeituhr lesen
EW 07,15 : Status für CS31-Systembus
EW 08,00...EW 15,15 : Analog-Eingänge, CS31-Vor-Ort-Module
EW 16,00...EW 34,15 : reserviert
EW 100,00...EW 107,15 : reserviert
EW 200,00...EW 207,15 : reserviert

Ausgänge

A 00,00...A 61,15 : Digitale Ausgänge, CS31-Vor-Ort-Module
A 62,00...A 63,07 : Digitale Ausgänge der Zentraleinheit 07 KT 94
A 62,00 : schneller Zähler, nach Aktivierung direkte Ausgabe des "Zähler-Nulldurchgangs"

A 65,00...A 99,15 : reserviert
A 100,00...A 163,15 : reserviert
A 200,00...A 263,15 : reserviert

AW 00,00...AW 05,15 : Analog-Ausgänge, CS31-Vor-Ort-Module
AW 06,00...AW 06,03 : Analog-Ausgänge der Zentraleinheit 07 KT 94
AW 07,00...AW 07,15 : reserviert
AW 08,00...AW 15,15 : Analogausgänge, CS31-Vor-Ort-Module
AW 16,00...AW 34,15 : reserviert
AW 100,00...AW 107,15 : reserviert
AW 200,00...AW 207,15 : reserviert

Interne Operanden

M 00,00...M 254,15 : Binär-Merker
M 255,00 : Oszillator ca. 2 Hz
M 255,01 : Oszillator ca. 1 Hz
M 255,02 : Oszillator ca. 0,5 Hz
M 255,03 : Oszillator mit Periodendauer von ca. 1 Minute
M 255,04 : Oszillator ca. 1/8 Hz
M 255,05 : Oszillator ca. 4 Hz
M 255,06 : Oszillator ca. 8 Hz

M 255,10 : Summenfehlermeldung
M 255,11 : Fehlermeldung FK1
M 255,12 : Fehlermeldung FK2
M 255,13 : Fehlermeldung FK3
M 255,14 : Fehlermeldung FK4

M 255,15 : Erkennung "Neustart"
M 256,00...M 279,15 : Systemmerker / reserviert
M 280,00...M 511,15 : Binär-Merker

S 00,00...S 255,15	:	Schritte
K 00,00...K 00,01	:	Binär-Konstanten
MW 00,00...MW 253,15	:	Wort-Merker
MW 254,00...MW 255,15	:	Fehlermeldung
MW 256,00...MW 259,15	:	Systemmerker / reserviert
MW 260,00...MW 511,15	:	Anwenderbereich
KW 01,00...KW 79,15	:	Wort-Konstanten
MD 00,00...MD 63,15	:	Doppelwort-Merker
KD 00,01...KD 23,15	:	Doppelwort-Konstanten

Zeitwerte für Zeitfunktionen

KD yy,xx	:	Zeitwerte für Zeitfunktionen wie ESV, ASV usw. werden als <i>Konstante Doppelwort</i> oder als <i>Merker Doppelwort</i> projiziert. Zulässig sind ganzzahlige Vielfache von 1 ms.
MD yy,xx	:	<i>Merker Doppelwort</i> projiziert. Zulässig sind ganzzahlige Vielfache von 1 ms.

2.2.1.2 Direkte Konstanten

Direkte Konstanten sind nur bei Funktionsbausteinen an bestimmten Eingängen zulässig. Wo dies der Fall ist, wird in der Beschreibung der Funktionsbausteine erläutert.

#	-32768...+32767
#H	0000...FFFF

2.2.1.3 Marken

Marken dienen als Sprungziele für Vorwärtssprünge und Laufzahlblöcke.

MA 0...999

2.2.1.4 Systemkonstanten

Betriebsarten einstellen

Die Konstanten KW 00,00...KW 00,15 sind als Systemkonstanten reserviert. Auch die noch nicht belegten Konstanten KW 00,13...KW 00,15 und KW 80,00...KW 89,15 dürfen *auf keinen Fall* anderweitig benutzt werden.

KW 00,00	:	SPS-Einsatzart einstellen, (Stand-alone-SPS, Master-SPS, Slave-SPS)
KW 00,01	:	Initialisierung: Binär-Merker-Bereich
KW 00,02	:	Initialisierung: Wort-Merker-Bereich
KW 00,03	:	Initialisierung: Doppelwort-Merker-Bereich
KW 00,04	:	Initialisierung: Schrittketten-Merker-Bereich
KW 00,05	:	Initialisierung: Vergangenheitswerte
KW 00,06	:	Verwendungsart der seriellen Schnittstelle COM 1
KW 00,07	:	SPS-Reaktion auf Fehler der Klasse 3
KW 00,08	:	SPS-Reaktion auf Überlast/Kurzschluß an den Transistorausgängen
KW 00,09	:	Mindestanzahl der in den CS31-Systembus-Zyklus aufgenommenen Vor-Ort-Module
KW 00,10	:	Größe des Sendebereichs der Slave-SPS
KW 00,11	:	Größe des Empfangsbereichs der Slave-SPS
KW 00,12	:	Automatischer Warmstart nach FK2-Fehler
KW 00,15	:	Oszillatoren auf M 255,00...M 255,06 deaktivieren
KW 85,00...KW 85,03	:	Konfiguration der Signalverzögerung digitaler Eingänge
KW 85,02	:	Konfiguration der Betriebsarten des schnellen Zählers
KW 86,00...KW 86,07	:	Konfiguration der analogen Eingänge
KW 88,00...KW 88,03	:	Konfiguration der analogen Ausgänge

Zykluszeit einstellen

KD 00,00	:	Diese Konstante dient zur Vorgabe der Zykluszeit für das SPS-Programm. Die Zykluszeit wird in der Einheit Millisekunden angegeben. Zulässig sind ganzzahlige Vielfache von 1 ms.
----------	---	--

2.2.1.5 Systemmerker / Diagnosemerker

M 00,00...M 254,15	: Binär-Merker
M 255,00	: Oszillator ca. 2 Hz
M 255,01	: Oszillator ca. 1 Hz
M 255,02	: Oszillator ca. 0,5 Hz
M 255,03	: Oszillator mit Periodendauer von ca. 1 Minute
M 255,04	: Oszillator ca. 1/8 Hz
M 255,05	: Oszillator ca. 4 Hz
M 255,06	: Oszillator ca. 8 Hz

M 255,10 : Summen-Fehlermeldung,	signalisiert, daß von der SPS ein Fehler erkannt wurde
M 255,11 : Fehlermeldung FK1, fataler Fehler,	Detailinformation in MW 254,00...MW 254,07
M 255,12 : Fehlermeldung FK2, schwerer Fehler,	Detailinformation in MW 254,08...MW 254,15
M 255,13 : Fehlermeldung FK3, leichter Fehler,	Detailinformation in MW 255,00...MW 255,07
M 255,14 : Fehlermeldung FK4, Warnung,	Detailinformation in MW 255,08...MW 255,15

M 255,15 : Erkennung "Neustart"

MW 254,00...MW 255,15 : Fehlermeldungen

Erst-Zyklus-Erkennung

M 255,15

Dieser Binär-Merker kann zur Erkennung des ersten Programmzyklus nach einem Programmstart benutzt werden. Er wird grundsätzlich, d. h. unabhängig von den Initialisierungsvorgaben durch die Systemkonstanten, bei jedem Start des Anwenderprogramms zu "0" initialisiert. Wird dieser Merker im Anwenderprogramm abgefragt und anschließend auf den Wert "1" gesetzt, so kann festgestellt werden, ob das Anwenderprogramm neu gestartet wurde oder nicht.

B7.6 CS31-Statuswort

EW 07,15

Bit 0 = 1	: kein CS31-Fehler der Klasse 2 vorhanden
Bit 1 = 1	: SPS ist in CS31-Zyklus aufgenommen (nur relevant beim Einsatz als Slave)
Bit 2 = 1	: Uhrzeit und Datum gültig
Bit 3 = 1	: Batterie vorhanden
Bit 4...7	: unbenutzt
Bit 8..15	: bis zum momentanen Zeitpunkt festgestellte maximale Anzahl von Geräten am CS31-Systembus (nur relevant beim Einsatz als Master)

2.2.2 Operanden der 07 KR 31 und 07 KT 31

2.2.2.1 Verfügbare Variablen und Konstanten

Eingänge

E 00,00...E 61,15	: Digitale Eingänge, CS31-Vor-Ort-Module
E 62,00...E 62,11	: Digitale Eingänge der Zentraleinheit 07 KR 31 / 07 KT 31
E 63,14	: Digitale Eingänge schnell (T _v = 0,02 ms), Signal identisch mit E 62,00
E 63,13	: schneller Zähler, "Nulldurchgang" abfragen
EW 00,00...EW 05,15	: Analog-Eingänge, CS31-Vor-Ort-Module
EW 06,15	: schneller Zähler, "Zählerstand" abfragen
EW 07,00...EW 07,07	: reserviert (für Diagnose am CS31-Systembus)
EW 07,08...EW 07,14	: Echtzeituhr lesen
EW 07,15	: Status für CS31-Systembus

Ausgänge

A 00,00...A 61,15	: Digitale Ausgänge, CS31-Vor-Ort-Module
A 62,00...A 62,07	: Digitale Relais-Ausgänge der Zentraleinheit 07 KR 31 / 07 KT 31
A 63,15	: schneller Zähler, Anfangswert übernehmen
AW 00,00...AW 05,15	: Analog-Ausgänge, CS31-Vor-Ort-Module
AW 06,15	: schneller Zähler, Anfangswert

Interne Operanden

M 00,00...M 21,15 : Binär-Merker
M 230,00...M 239,15
M 255,00...M 255,15 : Diagnosemerker
S 00,00...S 15,15 : Schritte
K 00...K 00,01 : Binär-Konstanten

MW 00,00...MW 05,15 : Wort-Merker
MW 230,00...MW 239,15
MW 254,00...MW 255,15 : Diagnose-Worte
KW 00,00...KW 07,15 : Wort-Konstanten
MD 00,00...MD 01,15 : Doppelwort-Merker
KD 00,00...KD 01,15 : Doppelwort-Konstanten

Zeitwerte für Zeitfunktionen

KD yy,xx : Zeitwerte für Zeitfunktionen wie ESV, ASV usw. werden als *Konstante Doppelwort* oder als *Merker Doppelwort* projiziert. Zulässig sind ganzzahlige Vielfache von 5 ms.
MD yy,xx : *Merker Doppelwort* projiziert. Zulässig sind ganzzahlige Vielfache von 5 ms.

2.2.2.2 Direkte Konstanten

Direkte Konstanten sind nur bei Funktionsbausteinen an bestimmten Eingängen zulässig. Wo dies der Fall ist, wird in der Beschreibung der Funktionsbausteine erläutert.

-32768...+32767
#H 0000...FFFF

2.2.2.3 Marken

Marken dienen als Sprungziele für Vorwärtssprünge und Laufzahlblöcke.
MA 0...999

2.2.2.4 Systemkonstanten

Identisch mit Kapitel 2.2.1.4, außer:
KW 00,08 : nicht benutzt

2.2.2.5 Diagnosemerker

Identisch mit Kapitel 2.2.1.5

2.2.2.6 CS31-Status

Identisch mit Kapitel 2.2.1.6, außer:
Bit 2...7 : nicht benutzt

2.2.3 Operanden des 07 KP 62

2.2.3.1 Verfügbare Variablen und Konstanten

Eingänge

Das Gerät besitzt *keine* Prozeß-Eingänge.

EW 00,04...EW 00,07 : schnelle Eingänge *von* der ABB Procontic T200

Ausgänge

Das Gerät besitzt *keine* Prozeß-Ausgänge.

AW 00,04...AW 00,03 : schnelle Ausgänge *zur* ABB Procontic T200

Interne Operanden

MW 00,00...MW 05,15 : Ausgabe-Merker *zur* ABB Procontic T200
MW 06,00...MW 11,15 : Eingabe-Merker *von* der ABB Procontic T200
MW 12,00...MW 253,15 : freie Wortmerker

Zykluszeit einstellen

KD 00,00 : Diese Konstante dient zur Vorgabe der Zykluszeit für das SPS-Programm. Die Zykluszeit wird in der Einheit Millisekunden angegeben. Zulässig sind ganzzahlige Vielfache von 5 ms.

Zeitwerte für Zeitfunktionen

KD yy,xx : Zeitwerte für Zeitfunktionen wie ESV, ASV usw. werden als *Konstante Doppelwort* oder als
MD yy,xx : *Merker Doppelwort* projiziert. Zulässig sind ganzzahlige Vielfache von 5 ms.

2.2.3.2 Direkte Konstanten

Direkte Konstanten sind nur bei Funktionsbausteinen an bestimmten Eingängen zulässig. Wo dies der Fall ist, wird in der Beschreibung der Funktionsbausteine erläutert.

-32768...+32767

#H 0000...FFFF

2.2.3.3 Marken

Marken dienen als Sprungziele für Vorwärtssprünge und Laufzahlblöcke.

MA 0...999

2.2.3.4 Systemkonstanten

Betriebsarten einstellen

Die Konstanten KW 00,00...KW 00,15 sind als Systemkonstanten reserviert. Auch die noch nicht belegten Konstanten KW 00,12...KW 00,15 dürfen auf keinen Fall anderweitig benutzt werden.

Beim Gerät 07 KP 62 gibt es nur die Systemkonstanten KW 00,01...KW 00,07.

Zykluszeit einstellen

KD 00,00 : Diese Konstante dient zur Vorgabe der Zykluszeit für das SPS-Programm. Die Zykluszeit wird in der Einheit Millisekunden angegeben. Zulässig sind ganzzahlige Vielfache von 5 ms.

2.2.3.5 Diagnosemerker

M 255,10 : Summen-Fehlermeldung,	signalisiert, daß von der SPS ein Fehler erkannt wurde
M 255,11 : Fehlermeldung FK1, fataler Fehler,	Detailinformation in MW 254,00...MW 254,07
M 255,12 : Fehlermeldung FK2, schwerer Fehler,	Detailinformation in MW 254,08...MW 254,15
M 255,13 : Fehlermeldung FK3, leichter Fehler,	Detailinformation in MW 255,00...MW 255,07
M 255,14 : Fehlermeldung FK4, Warnung,	Detailinformation in MW 255,08...MW 255,15

Erst-Zyklus-Erkennung

M 255,15

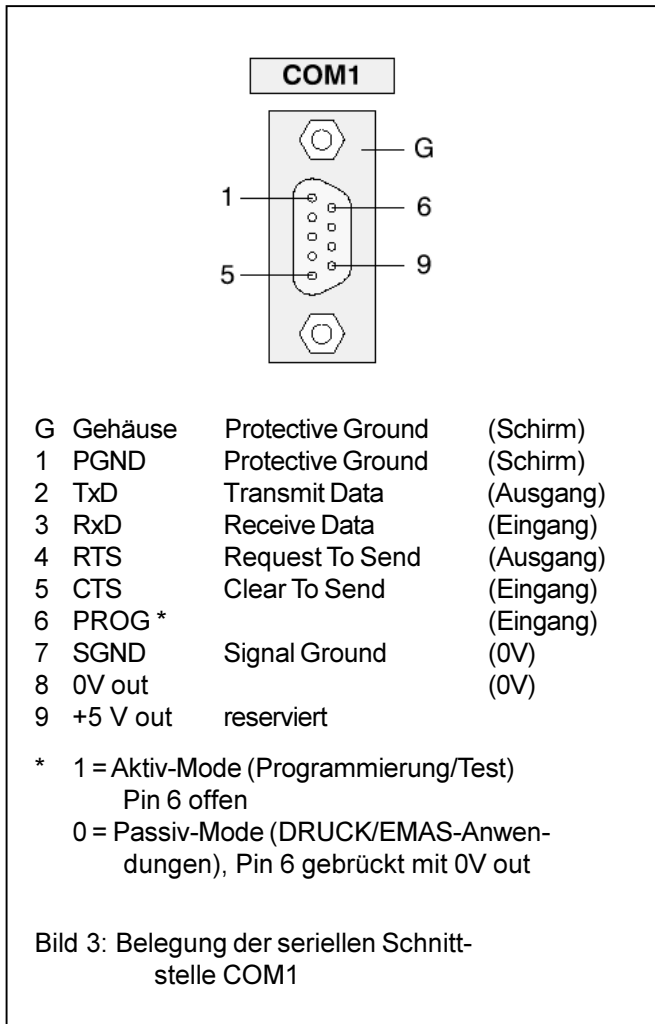
Dieser Binär-Merker kann zur Erkennung des ersten Programmzyklus nach einem Programmstart benutzt werden. Er wird grundsätzlich, d. h. unabhängig von den Initialisierungsvorgaben durch die Systemkonstanten, bei jedem Start des Anwenderprogramms zu "0" initialisiert. Wird dieser Merker im Anwenderprogramm abgefragt und anschließend auf den Wert "1" gesetzt, so kann festgestellt werden, ob das Anwenderprogramm neu gestartet wurde oder nicht.

2.3 Serielle Schnittstelle COM1

Schnittstellennorm: EIA RS-232

Belegung der seriellen Schnittstelle COM1

Die serielle Schnittstelle COM1 hat folgende Anschlußbelegung:



Betriebsarten der seriellen Schnittstelle COM1

Abhängig von der jeweiligen Anwendung

- Programmierung und Test oder
 - Mensch-Maschine-Kommunikation MMK
- muß die Betriebsart der Schnittstelle eingestellt werden:

Aktiv-Mode: Der Aktiv-Mode dient zur Programmierung und zum Test der Zentraleinheit, d. h. er bietet den Zugang zu allen Programmier- und Testfunktionen der Zentraleinheit.

Passiv-Mode: Der Passiv-Mode dient dazu, eine mit den Bausteinen DRUCK und EMAS projektierte Kommunikation zwischen dem Anwenderprogramm und einem an die serielle Schnittstelle angeschlossenen Gerät durchzuführen.

Bedingungen für die Einstellung der Betriebsarten der Schnittstelle COM1

RUN/STOP-Schalter	Systemkonstante KW00,06	Verbindungskabel/Gerät	damit eingestellter Mode
STOP	x	x	Aktiv
RUN	1	x	Aktiv
RUN	2	x	Passiv
RUN	0, <0, >2	07 SK 90	Aktiv
RUN	0, <0, >2	07 SK 91	Passiv

x: ohne Einfluß

Zeitweiliges Verlassen des Passiv-Modus

Während einer laufenden Kommunikation zwischen den Bausteinen DRUCK bzw. EMAS und einem an COM1 angeschlossenen Gerät kann es erforderlich sein, z. B. eine Programmänderung durchzuführen. Dazu muß COM1 vom Passiv-Mode in den Aktiv-Mode umgeschaltet werden.

Umschalten: Passiv-Mode —> Aktiv-Mode

Für die Umschaltung gibt es die drei folgenden Möglichkeiten:

- RUN/STOP-Schalter in Stellung "STOP" bringen
- Kabel 07 SK 91 ersetzen durch Kabel 07 SK 90 (wenn KW 00,06 auf <0 oder >2 eingestellt ist)
- Folgendes Spezialkommando an die SPS senden:

Die zuletzt genannte Möglichkeit bietet den Vorteil, die Umschaltung auch ferngesteuert, z. B. über Telefonleitung und geeignete Wähl-Modems, durchzuführen. Das ASCII-Zeichen hat den Dezimalcode 127 und den Hexadezimalcode 7F_H. Beim PC wird dieses Zeichen durch gleichzeitiges Drücken der Steuerungstaste <CTRL> und der Löschtaste <— erzeugt.

Hinweise:

Bei deutschen Tastaturen ist die Steuerungstaste nicht mit <CTRL> sondern mit <Strg> beschriftet.

Erfolgte die Umschaltung in den Aktiv-Mode mit dem Spezialkommando , so gilt:

Bei *laufendem* SPS-Programm darf die Systemkonstante KW 00,06 **nicht** zur SPS gesendet werden, da dies die Rückschaltung in den Passiv-Mode zur Folge hat.

Das Spezialkommando weist dem im Operandenspeicher angesiedelten Abbild der Systemkonstanten KW 00,06 den Wert "1" zu. Die SPS wertet den Wert dieses Abbildes aus und stellt die Verwendungsart von COM1 entsprechend ein.

Zurückschalten: Aktiv-Mode → Passiv-Mode

Für die Zurückschaltung gibt es die drei folgenden Möglichkeiten:

- RUN/STOP-Schalter wieder in Stellung "RUN" bringen
- Kabel 07 SK 90 wieder ersetzen durch Kabel 07 SK 91
- Spezialkommando wieder wie folgt aufheben:
 - Falls sich das SPS-Programm im Zustand "abgebrochen" befindet:

das SPS-Programm starten.

- Falls sich das SPS-Programm im Zustand "läuft" befindet:

den Originalwert der Systemkonstanten KW 00,06 erneut zur Steuerung senden (907 PC 33-Menüpunkt "Konstanten senden")

oder

die Systemkonstante KW 00,06 mit dem Originalwert überschreiben (907 PC 33-Menüpunkt "Überschreiben")

Schnittstellenparameter

Aktiv-Mode: Die Einstellung der Schnittstellenparameter kann nicht verändert werden.

Datenbits:	8
Stoppbits:	1
Parity Bit:	keines
Baudrate:	9600
Synchronisation:	RTS/CTS

Passiv-Mode: Default-Einstellung

Synchronisation:	RTS/CTS
Schnittstellenkennung COM1:	1
Baudrate:	9600
Stoppbits:	1
Datenbits:	8
Parity Bit:	keines
Echo:	aus
Send Break Character:	0
Freigabe Endezeichen für Senderichtung:	nein 1)
Endezeichen Senden:	<CR> 1)
Endezeichen Empfangen:	<CR> 2)

- 1) Das Default-Endezeichen für die Senderichtung (CR) wird nicht verschickt. Dieses Default-Endezeichen (CR) darf aber trotzdem im Telegrammtext des zugeordneten Druckbausteines nicht vorkommen.
- 2) In Empfangsrichtung ist grundsätzlich ein Endezeichen notwendig. Dieses Default-Endezeichen (CR) darf im Telegrammtext und in den Nutzdaten des zugeordneten EMAS-Bausteines nicht vorkommen.

Für den Passiv-Mode von COM1 können die Schnittstellenparameter mit dem Funktionsbaustein SINIT umgestellt werden. Sind die geänderten Werte nicht plausibel, dann arbeitet die Schnittstelle COM1 mit den Default-Werten.

Bei jeder Umschaltung der Betriebsart wird die Schnittstelle neu initialisiert.

Im Aktiv-Mode stellen sich die Aktiv-Mode-Parameter ein, im Passiv-Mode stellen sich die durch den Baustein SINIT vorgegebenen Parameter bzw. die Default-Werte ein.

2.4 Bedien- und Testfunktionen

Bedien-Kommandos

Die Bedienkommandos lassen sich gliedern in:

- Kommandos zur Erstellung und Änderung von Anwenderprogrammen
- Kommandos zum Testen der Anwenderprogramme
- Kommandos zum Konfigurieren der SPS

Anmerkungen:

- Bei den Eingaben sind keine "Blanks" notwendig, evtl. eingegebene "Blanks" werden ignoriert.
- Bei der Beschreibung der Kommandos werden zur besseren Übersichtlichkeit die Anwendereingaben

- für Schlüsselwörter in

GROSSBUCHSTABEN

- und für sonstige Eingaben (Adressen u. ä.) in

kleinbuchstaben

dargestellt.

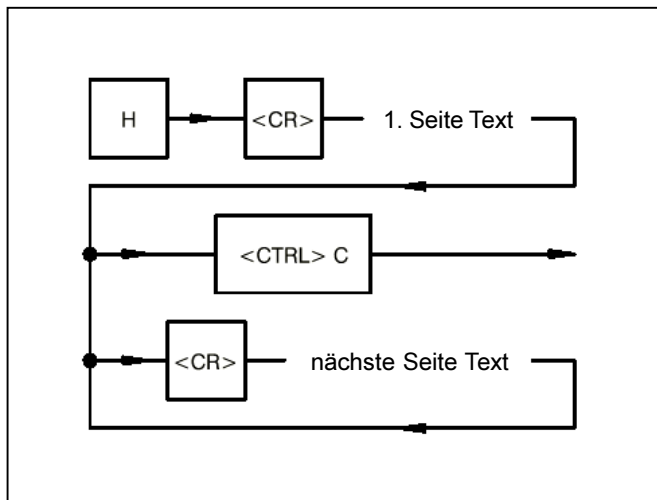
- Von der SPS-Software erzeugte Ausgaben auf dem Bildschirm werden in

kleinbuchstaben kursiv

dargestellt.

Alle verfügbaren Kommandos werden durch den HELP-Befehl auf dem Bildschirm angezeigt.

Help-Befehl



Funktion:

Auf dem Bildschirm werden alle verfügbaren Bedien- und Testfunktionen angezeigt. Mit <CR> wird im HELP-Text weitergeblättert.

Kommandos zur Erstellung des Anwenderprogramms (Übersicht)

Befehl	Funktion	Seite
AEND	Programmänderung an einem laufenden SPS-Programm vorbereiten	18
AEND	Verwerfen einer noch nicht freigegebenen Programmänderung	18
ALT	Verwerfen einer freigegebenen Änderung an einem laufenden SPS-Programm und Reaktivierung des alten Programm-Zustandes	18
AL	Auslastung anzeigen	18
CROSS *)	CROSS-Referenzliste anzeigen	19
D	Programm anzeigen	20
DEEP	SPS-Programm im Flash-EPROM löschen	20
F *)	String im Anwenderprogramm suchen	20
FREI	Freigabe einer Programmänderung an einem laufenden SPS-Programm	20
FRD	Datensätze vom Flash-EPROM lesen	21
FWR	Datensätze ins Flash-EPROM schreiben	21
FDEL	Datensegment im Flash-EPROM löschen ...	21
FCINIT *)	SmartMedia Card initialisieren	22
FCRD *)	Datensatz aus der SmartMedia Card lesen .	22
FCWR *)	Datensatz zur SmartMedia Card schreiben .	22
FCDEL *)	Datensegment der SmartMedia Card löschen	22
SIZE16	Anwenderprogrammspeicher verdoppeln	23
IDA	Programm-Identifikation anzeigen	23
IDR	Programm-Identifikation löschen	23
IDS	Programm-Identifikation eingeben	23
K	Werte von indirekten Konstanten eingeben/ändern	23
NOP	Programm-Teil löschen, d. h. mit NOPs überschreiben	24
O	Programm optimieren	25
P	Anzeigen des freien Programmspeichers	25
PA *)	Anwender-Programm aufbereiten	25
S	Anwenderprogramm eingeben/ändern (Substitute)	25
SO *)	Anwenderprogramm eingeben/ändern ohne Echo	26
SP	SPS-Programm im Flash-EPROM und in der SmartMedia Card sichern	26

*) **nicht** bei Serie 30, 40, 50

) **nur bei Serie 30, 40, 50

Kommandos zum Testen des Anwenderprogramms (Übersicht)

Befehl	Funktion	Seite
V	Anwenderprogramm verschieben	26
A	Anwenderprogramm abbrechen	27
BA *)	Breakpoints anzeigen	27
BR *)	Breakpoints rücksetzen	27
BS *)	Breakpoints setzen	27
<CTRL>W *)	Umschalten Bedienfunktionen <—> Monitor	28
EA *)	E/A-Test	28
EAA *)	E/A-Test ausschalten	28
ES *)	Einzelschritt einschalten	28
ESA *)	Einzelschritt ausschalten	28
EZ *)	Einzelzyklus einschalten	28
EZA *)	Einzelzyklus ausschalten	29
FEHLER	Anzeigen des Inhalts des Fehlerregisters	29
FORC	Force-Wert eingeben	30
FORCA	Force-Wert anzeigen	30
FORC R	Forcen löschen	30
G	Anwenderprogramm starten	30
KALT	Kaltstart durchführen	31

Befehl	Funktion	Seite
WARM	Warmstart durchführen	31
L *)	Anwenderprogramm fortsetzen	31
PS	Programmstatus anzeigen	31
ST	SPS-Status anzeigen	32
TRACE *)	TRACE-Betrieb	32
TRACE *)	TRACE-Speicher anzeigen	32
TRACE E *)	TRACE-Betrieb einschalten	32
TRACE A *)	TRACE-Betrieb ausschalten	32
W *)	Anwenderprogramm stoppen	33
Y	Wert einer Variablen mit einem vorzugebenden Wert überschreiben	33
Z	Status von Variablen anzeigen	33
ZZ	Nur die Werte der Variablen anzeigen	34
ZD	Status von Variablen anzeigen und laufend aktualisieren	34

Kommandos zum Konfigurieren

KONFS	Betriebsarten anzeigen/ändern	35
MAIL	Konfiguration CS31-Vor-Ort-Module	35
PASS **)	Paßwort	39
UHR	Anzeige von Uhrzeit und Datum	39
UHRS	Stellen von Uhrzeit und Datum	40

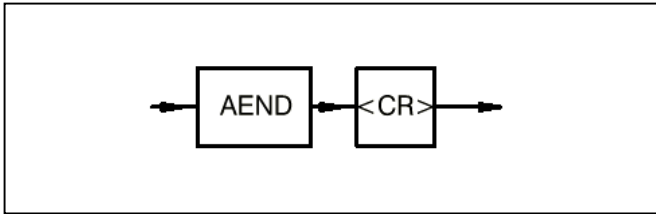
*) **nicht** bei Serie 30, 40, 50

) **nur bei Serie 30, 40, 50

2.4.1 Kommandos zur Erstellung des Anwenderprogramms

Programmänderung an einem laufenden SPS-Programm vorbereiten

Befehl:



Funktion:

Der Befehl kündigt der SPS an, daß Änderungen am laufenden SPS-Programm durchgeführt werden sollen. Nach Eingabe dieses Befehls ist die SPS bereit, die Programm- und Konstantenänderungen entgegenzunehmen.

Durch Eingabe des Befehls AEND werden alle momentan aktiven Testfunktionen inaktiv. Force-Werte von E/A-Signalen bleiben aber wirksam.

Folgende Befehle zur Programmbearbeitung und Bedienung der SPS sind nach Eingabe des Befehls AEND zulässig:

AL, CROSS, D, F, IDA, IDR, IDS, K, N, NOP, O, P, PA, S, SO, V, CTRL W, FEHLER, LED.

Verwerfen einer noch nicht freigegebenen Programmänderung

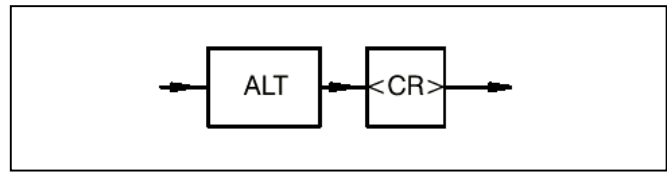
Eine erneute Eingabe des AEND-Befehls verwirft alle bisher durchgeführten Programmänderungen, und die SPS ist bereit, erneut Programmänderungen entgegenzunehmen.

Folgende Befehle werden beim **laufenden** Programm wirksam **und** verwerfen zusätzlich den AEND-Befehl und damit alle nach Eingabe des AEND-Befehls durchgeführten Programmänderungen:

A, BA, BR, BS, EA, EAA, ES, ESA, EZ, EZA, FORC, FORCA, FORC R, G, L, PS, ST, TRACE, TRACE E, W, Y. Um erneut Programmänderungen durchführen zu können, muß wiederum der Befehl AEND eingegeben werden.

Verwerfen einer freigegebenen Änderung an einem laufenden SPS-Programm und Reaktivierung des alten Programm-Zustandes.

Befehl:



Funktion:

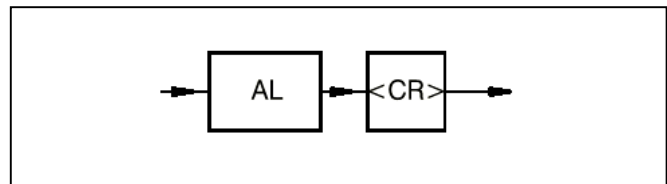
Die an einem laufenden SPS-Programm durchgeführten **und** freigegebenen Änderungen werden wieder verworfen. Zusätzlich stellt die SPS wieder den alten Programmzustand her. Der alte Programmzustand ist der Zustand des Programms, der **vor** der Programmänderung, also vor Eingabe des Befehls AEND in der SPS vorhanden war.

Nach Eingabe des Befehls ALT erfolgt die Reaktivierung des alten Programmzustands ohne weiteres Zutun des Anwenders innerhalb von ca. 1 ms.

Der Befehl kann benutzt werden, wenn der Anwender erkennt, daß die durchgeführten Programmänderungen nicht den von ihm gewünschten Erfolg haben.

Auslastung anzeigen

Befehl:



Funktion:

Die momentane Auslastung der SPS wird in Prozent angezeigt. Die Auslastungs-Anzeige sagt aus, wie stark die SPS durch die Bearbeitung des Anwenderprogrammes ausgelastet ist.

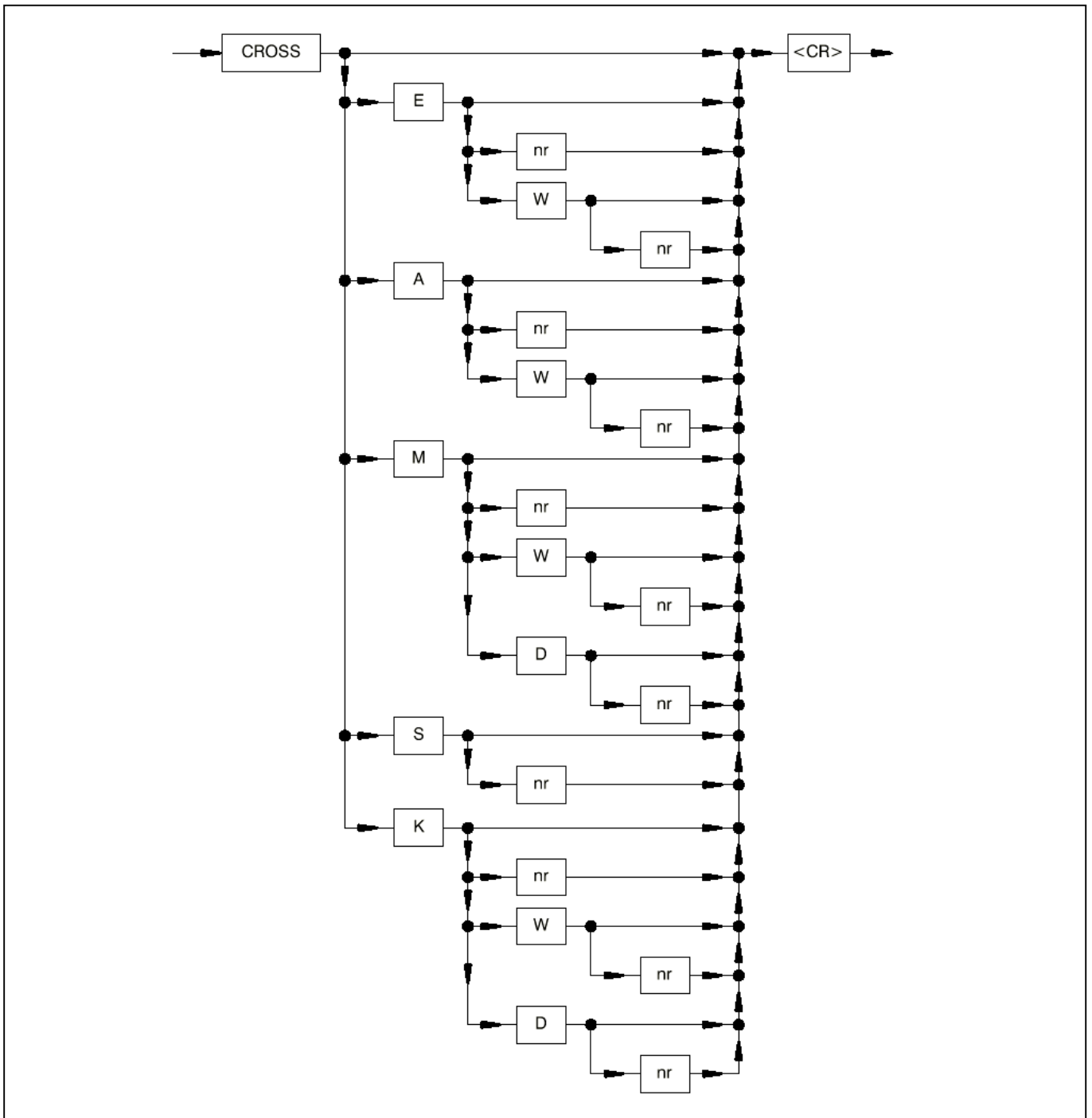
Die Prozessorkapazität, die der Differenz zwischen 100 % und der Auslastungsanzeige entspricht, steht für die Bedienung der seriellen Schnittstellen zur Verfügung, d. h. zur Kommunikation mit den an den seriellen Schnittstellen angeschlossenen Geräten. Die Auslastung sollte beim längsten Programmpfad nicht größer als 95 % sein, damit über die seriellen Schnittstellen noch eine Kommunikation möglich ist. Es ist zu beachten, daß die Auslastung der SPS durch die momentan aktuellen Programmverzweigungen (bedingte Sprünge und Laufzahlblöcke) mit bestimmt wird.

Anmerkung:

Die Auslastungsanzeige zeigt die durch das Anwenderprogramm verursachte Auslastung nur dann korrekt an, wenn zum Zeitpunkt der Anzeige *keine* Kommunikation über die seriellen Schnittstellen erfolgt.

Cross-Referenzliste anzeigen

Befehl:



Dabei ist:

E: Kürzel für Eingang
A: Kürzel für Ausgang
S: Kürzel für Schritt
M: Kürzel für Merker
K: Kürzel für Konstante
W: Kürzel für Wortgröße
D: Kürzel für Doppelwortgröße
nr: Nummer des Operanden

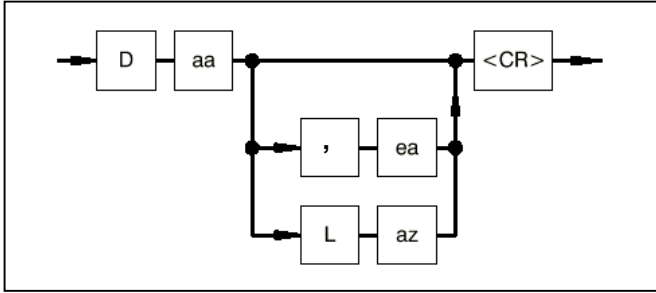
Funktion:

Die Crossreferenzliste ist die Zuordnung von Operanden zu den Programmspeicheradressen, an denen sie vorkommen. Die Crossreferenzliste kann ausgegeben werden für

- alle im Programm vorkommenden Operanden, Eingabe: CROSS <CR>
- einen bestimmten Operanden-Typ, Eingabe z. B.: CROSS E <CR>
- einen einzelnen Operanden, Eingabe z. B.: CROSS KD 00,12 <CR>

Programm anzeigen

Befehl:



aa: Anfangsadresse, ab der das Programm angezeigt werden soll

ea: Endadresse des anzuzeigenden Programmteils

L: Länge (Schlüsselwort)

az: Anzahl der anzuzeigenden Programmspeicherworte

Funktion:

Der spezifizierte Programmteil wird angezeigt.

Beispiel:

- D 0,20 <CR>

Das Anwenderprogramm wird von Adresse 0 bis zur Adresse 20 auf dem Bildschirm angezeigt.

- D 10 L 20 <CR>

Es werden 20 Programmspeicherworte von der Adresse 10 an angezeigt.

Anzeigeformat bei Sätzen:

anfangsadresse operator operand
:
:
:

Anzeigeformat bei Bausteinaufrufen:

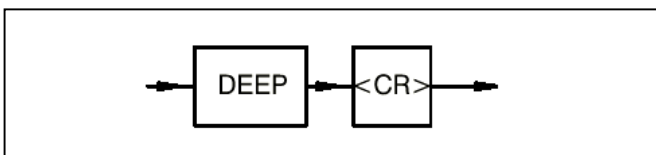
adresse n !ba nummer
adresse n+1 typ
adresse n+2 inhalt von adr n+2

Beispiel:

```
000000 !E 00,00
000002 &E 00,01
000004 =A 00,00
000006 !BA001
000007 AWT
000008 A 00,00
000009 KW 00,00
000010 KW 00,01
000011 AW 00,00
```

SPS-Programm im Flash-EPROM löschen

Befehl:

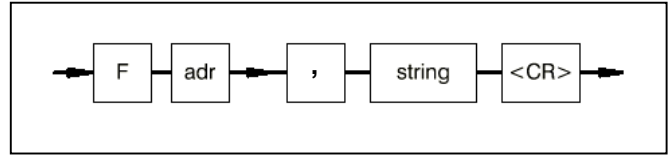


Funktion:

Ein im Flash-EPROM abgelegtes SPS-Programm wird gelöscht (ungültig gemacht).

String im Anwenderprogramm suchen (Find)

Befehl:



adr: Anfangsadresse, ab der gesucht werden soll. Wird keine Anfangsadresse eingegeben, wird ab Adresse 0 gesucht.

string: maximal 8 Befehle, d. h. 16 Worte des Zwischencodes.

Funktion:

Der Anwenderprogrammspeicher wird nach dem vom Anwender eingegebenen String ab der eingegebenen Anfangsadresse bis zum Ende des Anwenderprogrammspeichers durchsucht. Wird der String gefunden, wird die Adresse angezeigt. Kommt der String im Programm mehrfach vor, so wird durch Eingabe eines Semikolons (;) immer die nächste Programmadresse angezeigt, die mit dem String übereinstimmt.

Beispiel:

F, E 0,0 & E 0,1 <CR>

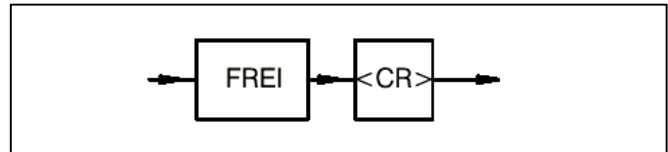
Der eingegebene String wird ab der Programmspeicher-Anfangsadresse 0 gesucht.

F 100, !BA1 <CR>

Der Bausteinaufruf 1 wird ab der Programmspeicher-Anfangsadresse 100 gesucht.

Freigabe einer Programmänderung an einem laufenden SPS-Programm

Befehl:



Funktion:

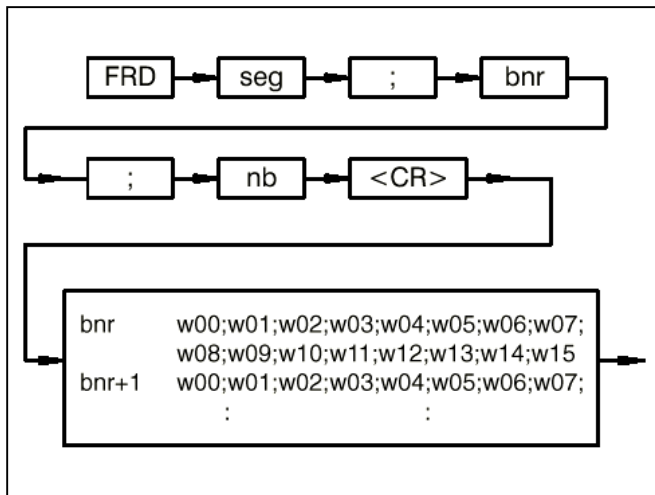
Die nach Eingabe des Befehls AEND durchgeführten Änderungen an einem laufenden SPS-Programm werden zur Ausführung freigegeben.

Vor Eingabe des Befehls FREI werden die durchgeführten Änderungen noch nicht von der SPS bearbeitet.

Nach Eingabe des Befehls FREI werden die durchgeführten Änderungen von der SPS bearbeitet. Mit dem Befehl ALT kann der alte Programmzustand wieder reaktiviert werden. Durch eine erneute Programmänderung kann die Funktionalität des SPS-Programms weiter geändert werden.

Datensätze vom FLASH-EPROM lesen

Befehl:



seg: Nummer des Datensegments im FLASH-EPROM
gültige Werte: 0...3

bnr: Nummer des Blocks im Datensegment,
gültige Werte: 0...480

nb: Anzahl der Blöcke,
gültige Werte: 1...481

:: Die einzelnen Werte des Befehls müssen durch Semikolon getrennt sein.

bnr: Nummer des Blocks im Datensegment

w00: 1. Wortwert des Blocks

: : :

w15: 16. Wortwert des Blocks

:: die einzelnen Werte der Antwort sind durch Semikolon getrennt

Funktion:

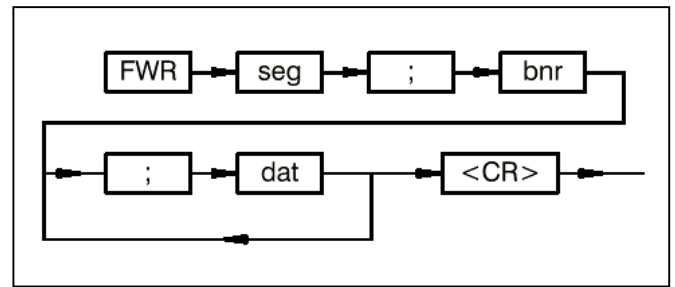
Der Anwender hat die Möglichkeit, Datensätze aus dem FLASH-EPROM zu lesen. Die Daten sind blockweise (16 Worte) im FLASH-EPROM abgelegt. Die Daten jedes Blocks sind mit einer Prüfsumme gesichert. Wird beim Auslesen eines Blocks ein Prüfsummenfehler festgestellt, wird statt der Nummer des Blocks (bnr) "ERROR" ausgegeben. Der Prüfsummenfehler wird gleichzeitig als FK3-Fehler (Fehlerkennung: 131 (83_H), Detail-Info: Offset, Segment) in das entsprechende Datenfeld eingetragen.

Beim Spannungszuschalten wird ein Prüfsummentest des gesamten FLASH-EPROMs durchgeführt. Wird ein Prüfsummenfehler festgestellt, wird der FK3-Fehler mit der Fehlerkennung 131 auf dem Bildschirm gemeldet und in die entsprechenden Fehlermerker eingetragen.

Anmerkung: Beim Starten eines SPS-Programms wird der FK3-Fehlermerker (Binär-Merker M 255, 13) prinzipiell gelöscht. Die Detailinformationen (Fehlerkennung, Detail-Info) bleiben im Wortmerker-Datenfeld (MW 255,00... MW 255,07) erhalten.

Datensätze ins FLASH-EPROM schreiben

Befehl:



seg: Nummer des Datensegments im FLASH-EPROM
gültige Werte: 0...3

bnr: Nummer des Blocks im Datensegment
gültige Werte: 0...480

:: Die einzelnen Werte des Befehls müssen durch Semikolon getrennt sein.

dat: neuer Wert

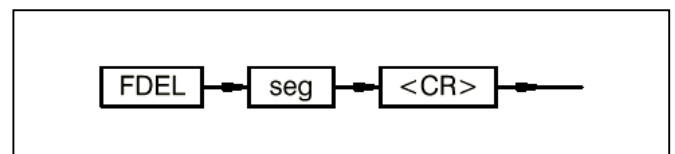
:: Die einzelnen Werte sind durch Semikolon getrennt.

Funktion:

Der Anwender hat die Möglichkeit, Datensätze ins FLASH-EPROM zu schreiben. Die Daten werden in dezimaler Schreibweise (-32768...+32767) eingegeben. Die Daten werden immer blockweise im FLASH-EPROM abgelegt und mit einer Prüfsumme gesichert. Jeder Block kann 16 Worte aufnehmen. Werden weniger als 16 Wortwerte eingegeben, werden die restlichen Worte mit dem Wert Null gefüllt. Nach der Eingabe von 8 Wortwerten werden ein <CR><LF> und 2 Blanks auf dem Bildschirm ausgegeben.

Datensegment im Flash-EPROM löschen

Befehl:



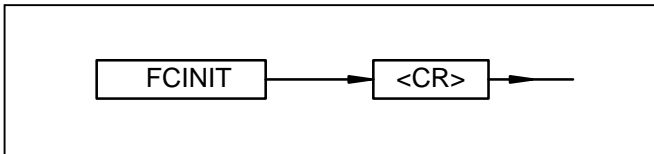
seg: Nummer des Datensegments im FLASH-EPROM
gültige Werte: 0...3

Funktion:

Der Anwender hat die Möglichkeit, ein Datensegment im Flash-EPROM zu löschen. Beim Löschvorgang gehen alle Daten in diesem Datensegment verloren.

SmartMedia Card initialisieren

Befehl:



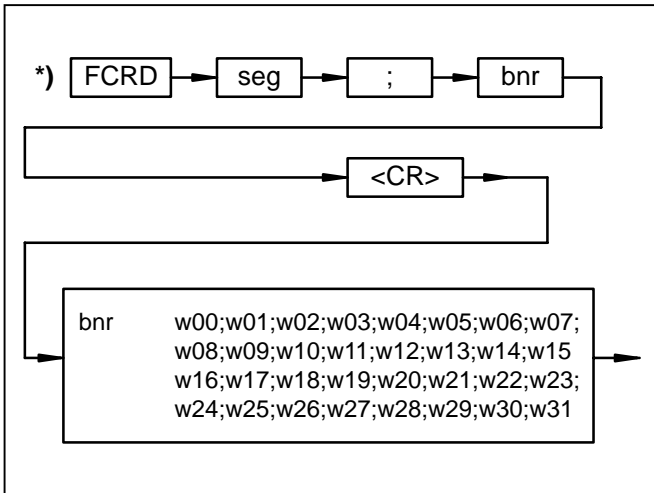
Funktion:

Die SmartMedia Card wird als Anwenderdatenkarte initialisiert. Es können nur Daten auf **initialisierte** Karten geschrieben werden.

Beim Initialisierungsvorgang gehen alle vorherigen Daten auf der Karte verloren. Eine als Anwenderdatenspeicher initialisierte SmartMedia Card kann nicht mehr als Anwendungsprogrammkarte verwendet werden.

Datensätze aus der SmartMedia Card lesen

Befehl:



seg: Nummer des Datensegments in der SmartMedia Card
gültige Werte: 1...250

bnr: Nummer des Blocks im Datensegment,
gültige Werte: 0...127

:: Die einzelnen Werte des Befehls müssen durch Semikolon getrennt sein.

bnr: Nummer des Blocks im Datensegment

w00: 1. Wortwert des Blocks

: : :
w31: 32. Wortwert des Blocks

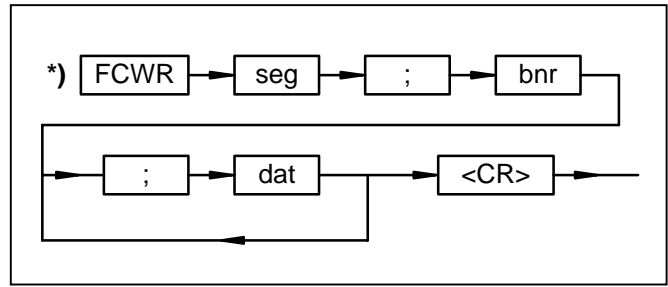
:: die einzelnen Werte der Antwort sind durch Semikolon getrennt

Funktion:

Der Anwender hat die Möglichkeit, Datensätze aus der SmartMedia Card zu lesen. Die Daten sind blockweise (32 Worte) in der SmartMedia Card abgelegt. Die Daten jedes Blocks sind mit einer Prüfsumme gesichert.

Datensätze in die SmartMedia Card schreiben

Befehl:



seg: Nummer des Datensegments in der SmartMedia Card
gültige Werte: 1...250

bnr: Nummer des Blocks im Datensegment
gültige Werte: 0...127

:: Die einzelnen Werte des Befehls müssen durch Semikolon getrennt sein.

dat: neuer Wert

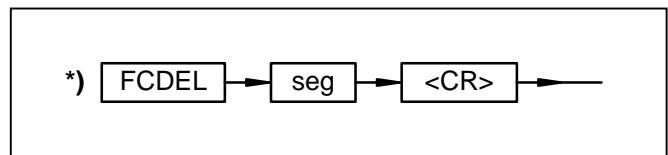
:: Die einzelnen Werte sind durch Semikolon getrennt.

Funktion:

Der Anwender hat die Möglichkeit, Datensätze in die SmartMedia Card zu schreiben. Die Daten werden in dezimaler Schreibweise (-32768...+32767) eingegeben. Die Daten werden immer blockweise in der SmartMedia Card abgelegt und mit einer Prüfsumme gesichert. Jeder Block kann 32 Worte aufnehmen. Werden weniger als 32 Wortwerte eingegeben, werden die restlichen Worte mit dem Wert Null gefüllt. Nach der Eingabe von 8 Wortwerten werden ein <CR><LF> und 2 Blanks auf dem Bildschirm ausgegeben. Ein Block kann nur einmal beschrieben werden. Vor erneutem Beschreiben muß das Segment gelöscht werden.

Datensegment in der SmartMedia Card löschen

Befehl:



seg: Nummer des Datensegments in der SmartMedia Card
gültige Werte: 1...250

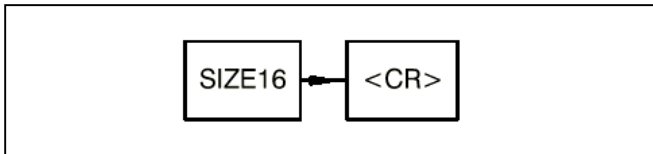
Funktion:

Der Anwender hat die Möglichkeit, ein Datensegment in der SmartMedia Card zu löschen. Beim Löschvorgang gehen alle Daten in diesem Datensegment verloren.

*) Ein Block kann nur einmal beschrieben werden. Vor erneutem Beschreiben muß das Segment gelöscht werden.

Anwenderprogrammspeicher verdoppeln

Befehl:



SIZE16 gibt es nur für die Zentraleinheiten 07 KR 91, 07 KT 92 R202/R262 und 07 KT 93 R101/R171. Bei den anderen Zentraleinheiten ist SIZE16 wegen des größeren Speichers nicht mehr notwendig.

Funktion:

Der Anwender-Programmspeicher wird verdoppelt (auf 15296 Anweisungen). Nach Eingabe dieses Befehls entfällt die Fähigkeit der Programmänderung an einem laufenden SPS-Programm.

Der Befehl kann nur unter folgenden Voraussetzungen eingegeben werden:

- kein Fehler der Fehlerklasse 2 vorhanden **und**
- SPS-Programm im Zustand "ABGEBROCHEN" **und**
- ungültiges Anwenderprogramm (DEEP-Befehl) im FLASH-EPROM

Nach der Eingabe dieses Befehls muß der SP-Befehl (Anwenderprogramm im FLASH-EPROM sichern) ausgeführt werden. Dadurch wird die Programmverdopplung spannungsausfallsicher gespeichert.

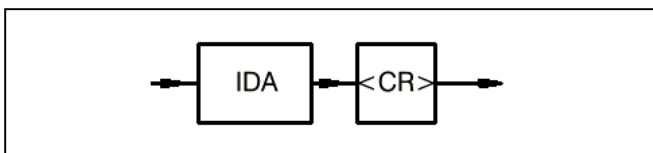
Wird der SP-Befehl nicht ausgeführt, wird die Verdopplung des Programmspeichers bei Spannung AUS/EIN, WARM-Befehl oder KALT-Befehl wieder unwirksam.

Die Verdopplung des Programmspeichers wird wie folgt rückgängig gemacht:

- DEEP-Befehl ausführen **und**
- Spannung AUS/EIN, WARM-Befehl oder KALT-Befehl

Programm-Identifikation anzeigen

Befehl:

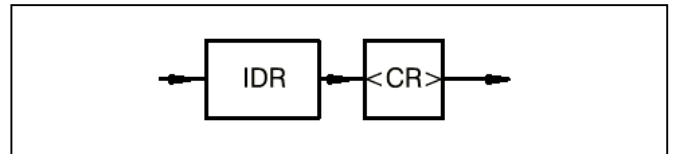


Funktion:

Die vom Anwender für das Anwenderprogramm eingegebene Identifikation wird angezeigt. Wurde für das Programm keine Identifikation vergeben, so wird auch nichts angezeigt (siehe auch Befehl: IDS).

Programm-Identifikation löschen

Befehl:

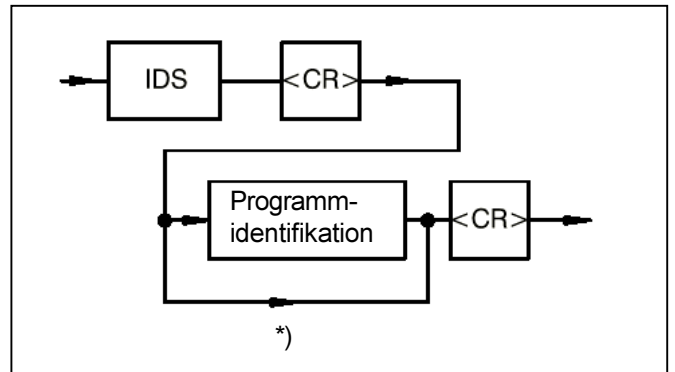


Funktion:

Die vom Anwender für das Anwenderprogramm eingegebene Identifikation wird gelöscht.

Programm-Identifikation eingeben

Befehl:



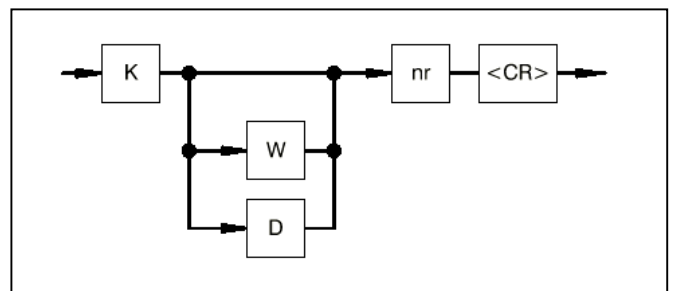
Programm-Identifikation: Diese Zeichen werden dem Anwenderprogramm als Kennung zugeordnet.

Funktion:

Die vom Anwender für das Anwenderprogramm eingegebene Identifikation wird im Programmspeicher abgelegt. Die Identifikation kann maximal 16 Zeichen umfassen. Sie dient z. B. dazu, den Projekt-Namen und das Erstellungsdatum des Programms in der SPS abzulegen.

Werte von indirekten Konstanten eingeben/ändern

Befehl:

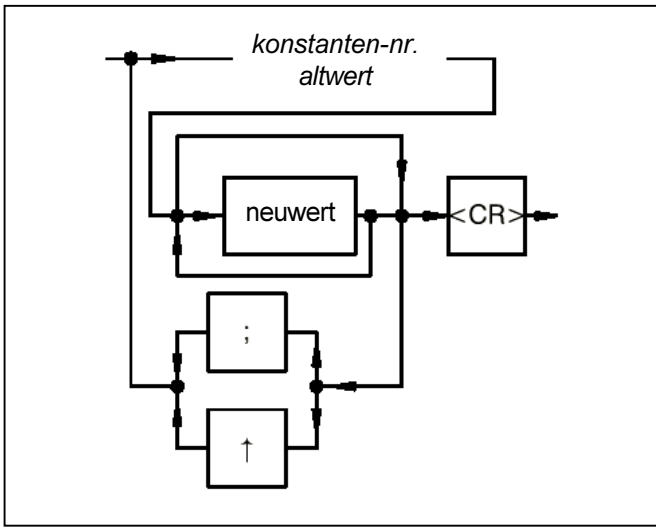


W: Kürzel für Wortkonstanten

D: Kürzel für Doppelwortkonstanten

nr: eingegebene Nummer der Konstanten

*) : Bei diesem Pfad wird keine Programm-Identifikation eingetragen. Eine schon vorhandene Programm-Identifikation wird gelöscht.



konstantennr. altwert:

Angezeigte Nummer und Wert der Konstanten.

neuwert:

Der Wert der angezeigten Konstanten kann durch einen neuen Wert vom Anwender überschrieben werden. Bei den Wort- und Doppelwort-Konstanten kann anstatt eines Dezimalwerts auch ein Hexadezimal-Wert eingegeben werden. Dazu wird dem Zahlenwert ein H vorangestellt.

Vorsicht: Die Werte H8000 und H8000 0000 sind bei der Zweierkomplement-Arithmetik verboten (nur sinnvoll z. B. bei Masken).

:: Die Eingabe eines Semikolons bewirkt das Anzeigen von Nummer und Wert der Konstanten mit der nächst höheren Nummer. Wird das Semikolon ohne Eingabe eines neuen Wertes eingegeben, so wird der alte Wert der angezeigten Konstanten beibehalten.

↑: Die Eingabe des Zeichens "↑" bewirkt das Anzeigen von Nummer und Wert der Konstanten mit der nächst niedrigeren Nummer. Wird das Zeichen "↑" ohne Eingabe eines neuen Wertes eingegeben, so wird der alte Wert der angezeigten Konstanten beibehalten. (Auf der PC-Tastatur muß das Zeichen " ^ " genommen werden.)

<CR>: Durch Eingabe eines <CR> wird der Befehl beendet.

Funktion:

Den indirekten Konstanten werden die gewünschten Zahlenwerte zugewiesen.

Diese Wertzuweisung kann auch bei laufendem Anwenderprogramm erfolgen. Damit können z. B. an der laufenden Anlage Zeitwerte von Zeitgliedern geändert werden.

Zykluszeit:

Mit der Doppelwortkonstanten KD 00,00 wird die Zykluszeit eingestellt. Die eingestellte Zykluszeit muß ein ganzzahliges Vielfaches der Grundzeit von 1 ms sein, d. h. 1 ms, 7 ms, 23 ms usw.

Beispiel:

K 0,0 <CR>

Ausgabe der Nummer und des Wertes der binären Konstanten K 00,00. Dieser Wert kann bei Bedarf überschrieben werden. Durch Eingabe eines Semikolons wird Nummer und Wert der nächsten binären Konstanten (K 00,01) ausgegeben.

KW 0,4 <CR>

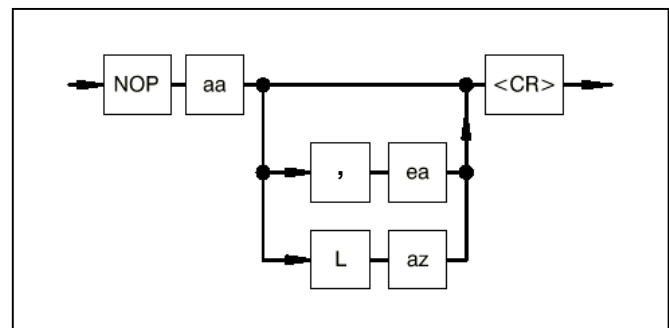
Ausgabe der Nummer und des Wertes der Wortkonstanten KW 00,04.

KD 0,0 <CR>

Ausgabe der Nummer und des Wertes der Doppelwortkonstanten KD 00,00. Mit dieser Konstanten wird die Zykluszeit vorgegeben.

Programm-Teil löschen, d. h. mit NOPs überschreiben

Befehl:



aa: Anfangsadresse des zu löschenden Programm-Teils

ea: Endadresse des zu löschenden Programm-Teils

L: Länge (Schlüsselwort)

az: Anzahl der zu löschenden Programmspeicherwörter

Funktion:

Der spezifizierte Programmteil wird gelöscht. Vor dem Löschen erfolgt noch eine Sicherheitsabfrage. Der Anwender muß dabei noch einmal das Löschen mit "J" bestätigen oder mit "N" verwerfen.

Beispiel:

NOP 0,20 <CR>

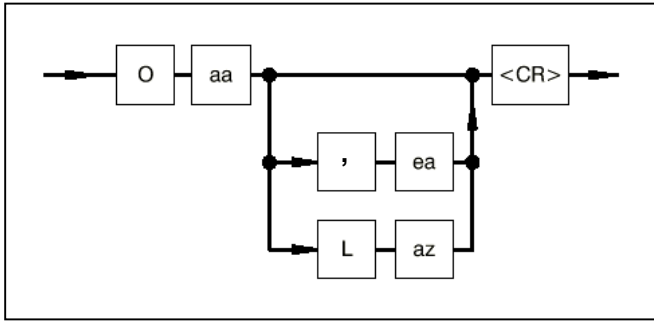
Das Anwenderprogramm wird von Adresse 0 bis zur Adresse 20 gelöscht.

NOP 10 L 20 <CR>

Es werden 20 Programmspeicherwörter ab der Adresse 10 gelöscht.

Programm optimieren

Befehl:



aa: Anfangsadresse des Bereichs, ab der der Programmspeicher optimiert werden soll.

ea: Endadresse des Bereichs

L: Länge (Schlüsselwort)

az: Anzahl der Programmspeicherworte

Funktion:

Im vorgegebenen Programmabschnitt werden alle NOPs entfernt und das Programm entsprechend zusammengesoben.

Beispiel:

O 0 <CR>

Der ganze Programmspeicher wird optimiert.

O 0,10 <CR>

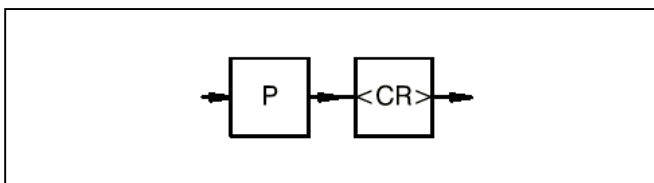
Der Programmspeicher wird ab der Adresse 0 bis zur Adresse 10 optimiert.

O 10 L 10 <CR>

Die NOPs innerhalb der nächsten 10 Programmspeicherworte ab Adresse 10 werden entfernt und das Programm entsprechend zusammengesoben.

Anzeigen des freien Programmspeichers

Befehl:

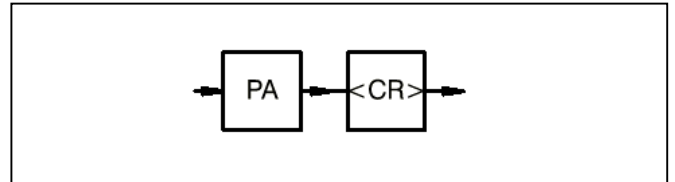


Funktion:

Der Programmspeicher wird vom Ende her auf NOPs untersucht. Wird im Zwischencode ein Wort gefunden, das nicht einem NOP entspricht, wird die Anzahl der gefundenen NOPs, d. h. der freien Programmspeicherworte, angezeigt.

Anwenderprogramm aufbereiten

Befehl:

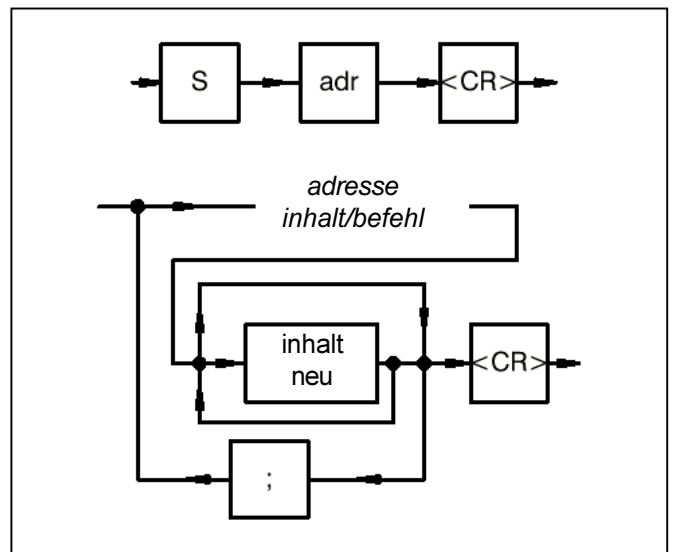


Funktion:

Die im Anwenderprogramm projektierten E/A-Signale werden in der E/A-Konfigurationsliste der SPS freigegeben. Ferner wird für das Anwenderprogramm ein Syntaxcheck durchgeführt. Bei Sätzen mit Vergleichsoperatoren, bei denen Klammerausdrücke benutzt werden, wird die KLAMMER ZU, die vor der binären Zuweisung steht, vom Übersetzer als binäre KLAMMER ZU im Zwischencode abgelegt. Diese binäre KLAMMER ZU wird durch die Programmaufbereitung zu einer Wortklammer korrigiert. PA berechnet für die Sprungbausteine und Laufzahlblöcke die Zieladressen und die zu überspringenden Vergangenheitswerte. Der PA-Befehl wird automatisch bei jedem Programmstart (G-Befehl) aufgerufen.

Anwenderprogramm eingeben/ändern (Substitute)

Befehl:



adr: Programmspeicher-Adresse, ab der das Programm in Anweisungsliste eingegeben oder geändert werden soll

adresse: Die Programmspeicher-Adresse, deren Inhalt verändert werden soll, wird von der SPS angezeigt.

inhalt: Gilt nur für Bausteinaufrufe. Es wird der rückübersetzte Inhalt der Programmspeicher-Adresse angezeigt.

befehl: Gilt für Sätze und den Bausteinkopf (Nummer und Typ). Es wird der rückübersetzte Befehl bzw. Bausteinkopf angezeigt und zwar immer als ganzer Befehl, d. h. Operand und Operator oder Bausteinaufruf und Bausteintyp. Wird eine Adresse eingegeben, die nicht auf einen Befehlsanfang oder auf einen Bausteinaufruf zeigt, so wird diese von der SPS auf den Befehlsanfang korrigiert.

inhalt neu: neuer Inhalt des Anwenderprogramms

:: Durch Eingabe eines Semikolons wird die nachfolgende Programmspeicher-Adresse und deren Inhalt angezeigt und kann bei Bedarf geändert werden. Wird vor dem Semikolon kein neuer 'inhalt' eingegeben, bleibt der alte Inhalt der angezeigten Programmspeicheradresse unverändert.

Funktion:

Eingeben oder Ändern des SPS-Programms in Anweisungsliste. Es wird ein Programmspeicherwort angewählt und auf dem Bildschirm als Anweisung bzw. Operand ausgegeben. Danach kann der angezeigte Inhalt überschrieben werden.

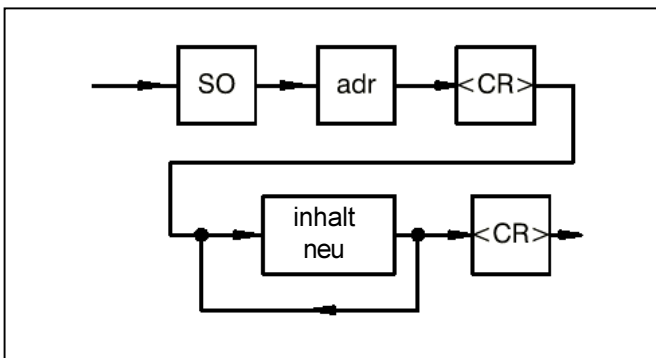
Hinweis:

Zum Eingeben/Ändern der Anweisungsliste mit diesem Befehl finden Sie am Ende dieses Anhangs noch folgende Angaben:

- Syntaktischer Aufbau der Anweisungsliste
- Wie werden Texte für die Funktionsbausteine DRUCK/EMAS eingegeben und angezeigt.

Anwenderprogramm eingeben/ändern ohne Echo

Befehl:



adr: Programmspeicher-Adresse, ab der das Programm eingegeben oder geändert werden soll

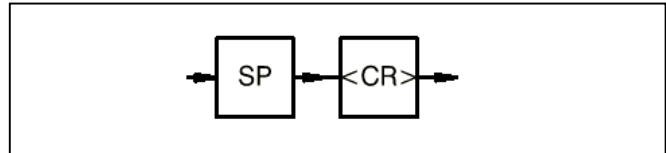
inhalt neu: neuer Inhalt des Anwenderprogramms

Funktion:

Es wird die Programmspeicher-Adresse, ab der das Programm eingegeben werden soll, vorgegeben. Das Programm kann dann fortlaufend eingegeben werden. Die SPS schickt dabei kein Echo des eingegebenen Programms zurück. Im Fehlerfall wird aber von der SPS eine Fehlermeldung (z. B. Falsche Eingabe) zurückgeschickt.

SPS-Programm im Flash-EPROM und in der SmartMedia Card sichern

Befehl:

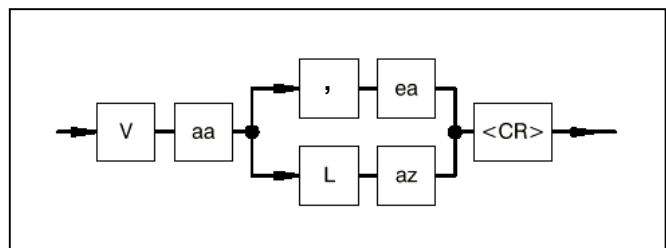


Funktion:

Das SPS-Programm wird vom RAM ins Flash-EPROM und, wenn vorhanden, gleichzeitig in die SmartMedia Card übertragen. Während des Programmiervorgangs wird auf dem Bildschirm in Abständen von ca. 1 Sekunde das Zeichen '<*>' ausgegeben.

Anwenderprogramm verschieben

Befehl:



aa: Anfangsadresse des Programmteils, der verschoben werden soll

ea: Endadresse

L: Länge (Schlüsselwort)

az: Anzahl der Programmspeicherwörter, um die verschoben werden soll

Funktion:

Das Programm wird von Adresse aa nach Adresse ea oder von Adresse aa um die angegebene Anzahl von Programmspeicherworten verschoben. Die dabei entstehende Lücke wird mit NOPs gefüllt. In diese Lücke können neue Programmteile eingefügt werden. Ein Verschieben ist nur möglich, wenn am Ende des Anwenderprogramms der notwendige Platz noch zur Verfügung steht; dies wird aber automatisch geprüft.

Beispiel:

V 0,10 <CR>

Das Programm wird von Adresse 0 nach Adresse 10 verschoben. Von Adresse 0 bis zu der Adresse 9 werden NOPs eingefügt.

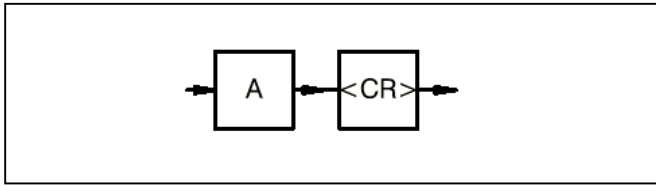
V 10 L 20 <CR>

Das Programm wird von Adresse 10 um 20 Programmspeicherwörter nach Adresse 30 verschoben und 20 NOPs werden eingefügt.

2.4.1 Kommandos zum Testen des Anwenderprogramms

Anwenderprogramm abbrechen

Befehl:



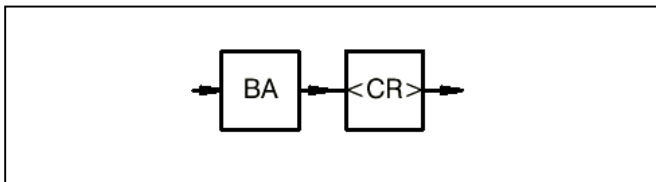
Funktion:

Die Bearbeitung des Anwenderprogramms wird abgebrochen. Dabei werden alle Ausgänge (Binär und Wort) zu Null gesetzt. Das Anwenderprogramm kann durch Eingabe von "G" wieder gestartet werden.

Gestartete Zeitwerke laufen unabhängig vom Programmzustand im Betriebssystem weiter. Sie werden nur durch einen Kaltstart oder Spannung AUS/EIN abgebrochen.

Breakpoints anzeigen

Befehl:

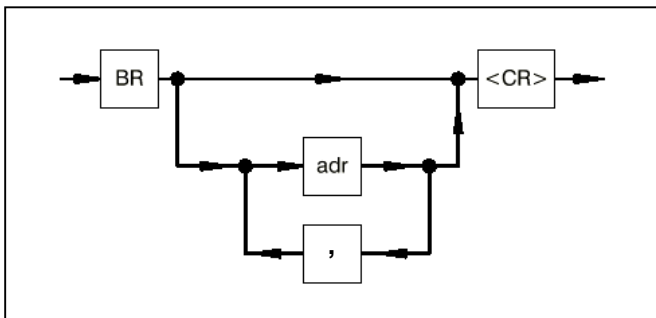


Funktion:

Es werden alle Breakpoints des Programms angezeigt. Bei einem Befehl wird dabei nicht die Breakpoint-Adresse, sondern die Adresse des Befehlsanfangs und deren Inhalt angezeigt.

Breakpoints rücksetzen

Befehl:



adr: Adresse des zu löschenden Breakpoints

,: Werden nur bestimmte Breakpoints gelöscht, müssen die einzelnen Adressen bei der Eingabe durch ein Komma getrennt werden.

Funktion:

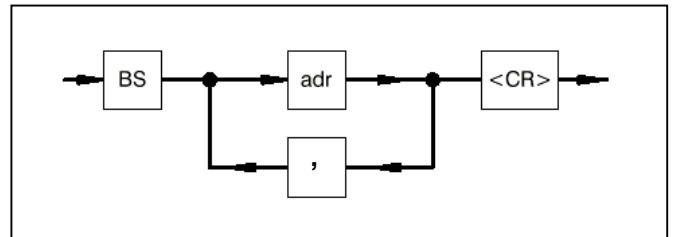
Die Breakpoints können einzeln gelöscht werden. Mit dem Befehl

BR <CR>

werden sämtliche Breakpoints des Programms gelöscht.

Breakpoints setzen

Befehl:



adr: Adresse des Breakpoints.

,: Werden mehrere Breakpoints gesetzt, müssen bei der Eingabe die Adressen durch ein Komma getrennt werden.

Breakpoints können gesetzt werden:

- auf die Adresse des Operanden nach einem Zuweisungszeichen
- auf die Adresse einer KLAMMER ZU
- auf die Adresse des letzten Parameters eines Bausteins
- auf die Adresse vom Programmende

Funktion:

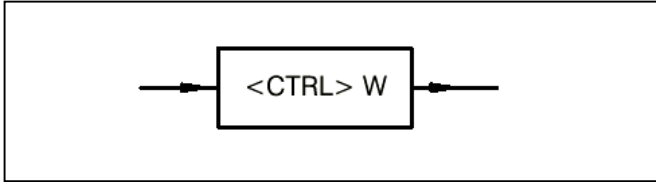
Nach dem Programm-Start hält das Programm beim ersten Breakpoint an. Breakpoints können auch bei laufendem Programm eingegeben werden. Es können maximal 15 Breakpoints vorgegeben werden.

Weiterschalten zum nächsten Haltepunkt: Durch Eingabe eines Semikolons läuft das Programm nach Ablauf der Zykluszeit bis zum nächsten Haltepunkt und zeigt die Programmadresse und den dort stehenden Befehl an. Wird der nächste Haltepunkt – bedingt durch eine lange Zykluszeit – nach einer bestimmten Zeit nicht erreicht, so kann der Anzeigevorgang, falls gewünscht, durch Eingabe von <CTRL>C abgebrochen werden.

Wird ein Breakpoint an eine Programmstelle gesetzt, die z. B. wegen eines Sprungs nicht bearbeitet wird, so läuft das Programm nach wie vor rund, aber mit der vierfachen Zykluszeit, was sich nachteilig auf die Funktionalität auswirken kann.

Umschalten Bedienfunktionen <—> Monitor

Befehl:

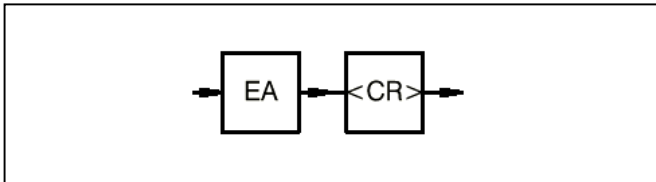


Funktion:

Durch gleichzeitiges Drücken der Taste <CTRL> und der Taste W gelangt man in das Monitorprogramm der SPS. Hier werden dem Anwender einige Grundfunktionen auf der Monitor-Ebene zur Verfügung gestellt. Befindet man sich im Monitor, gelangt man durch erneute Eingabe von <CTRL> und W wieder in das Bedienprogramm der SPS.

E/A-Test

Befehl:



Funktion:

Mit dieser Betriebsart kann der Anwender die Verdrahtung seiner E/A-Signale von der SPS bis zum Prozeß auf ihre Richtigkeit überprüfen.

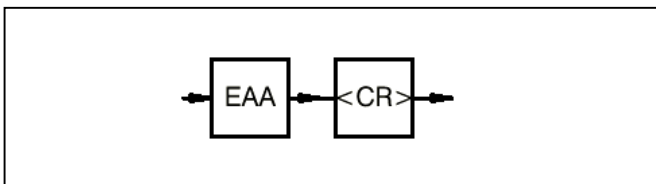
Nach dem Starten des Anwender-Programms wird dieses *nicht* bearbeitet. Es werden nur die im Programm projizierten E/A-Signale bedient. D. h. die Eingabe-Signale werden eingelesen, und die Ausgabe-Signale werden ausgegeben.

Durch Betätigung von Endschaltern usw. kann überprüft werden, ob die Signale in der SPS unter der vereinbarten Bezeichnung ankommen. Durch gezieltes Setzen von Ausgängen kann überprüft werden, ob die Signale an der richtigen Stelle im Prozeß ankommen. Mit den Befehlen Z bzw. ZD können in der SPS die gewünschten E/A-Variablen angezeigt werden.

Die Eingabe des Kommandos "EA" ist auch bei laufendem Programm möglich. Dabei wird die Betriebsart erst beim Beginn des nächsten Programmzyklus wirksam.

E/A-Test ausschalten

Befehl:

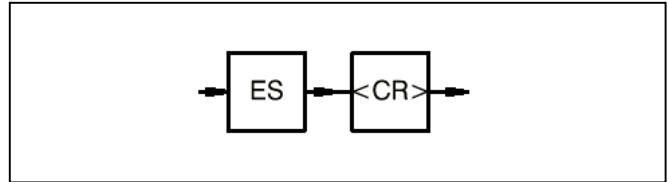


Funktion:

Die Betriebsart E/A-Test wird mit diesem Befehl ausgeschaltet, d. h. das Anwenderprogramm läuft ab sofort normal weiter. Es ist sinnvoll, vor dem Ausschalten des E/A-Tests das Programm abzubrechen.

Einzelschritt einschalten

Befehl:



Funktion:

Nach dem Starten des Programmes wird nur ein Satz oder ein Baustein abgearbeitet und das Programm hält nach jeder Zuweisung, KLAMMER ZU und am Ende jedes Bausteins an.

Mit dem Befehl Z können dabei Variablenwerte angezeigt werden.

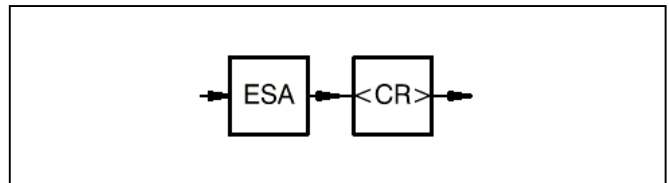
Die Eingabe des Kommandos "ES" ist auch bei laufendem Programm möglich. Dabei wird die Betriebsart erst beim Beginn des nächsten Programmzyklus wirksam.

Weiterschalten um einen Schritt:

Durch Eingabe eines Semikolons läuft das Programm nach Ablauf der Zykluszeit bis zum nächsten Haltepunkt und zeigt die Programmadresse und den dort stehenden Befehl an. Wird der nächste Haltepunkt – bedingt durch eine lange Zykluszeit – nach einer bestimmten Zeit nicht erreicht, so kann der Anzeigevorgang, falls gewünscht, durch Eingabe von <CTRL>C abgebrochen werden.

Einzelschritt ausschalten

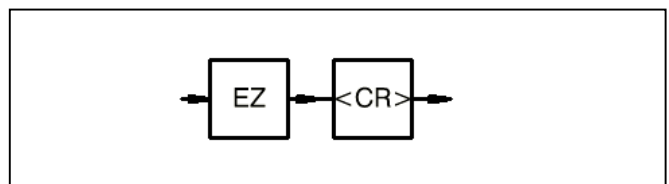
Befehl:



Der Einzelschrittbetrieb wird ausgeschaltet, d. h. das Anwenderprogramm läuft ab dem momentanen Haltepunkt normal weiter.

Einzelschritt einschalten

Befehl:



Funktion:

Beim Starten des Programms hält das Programm am Programmende an. Die Eingabe des Kommandos "EZ" ist auch bei laufendem Programm möglich.

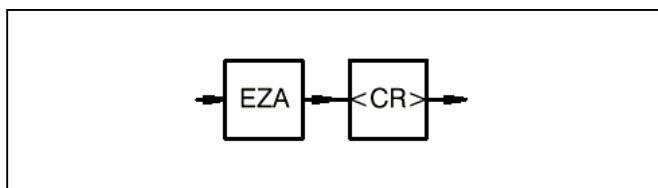
Dabei wird die Betriebsart erst beim Beginn des nächsten Programmzyklus wirksam.

Weiterschalten um einen Programmzyklus:

Durch Eingabe eines Semikolons wird das Programm nach Ablauf der Zykluszeit einmal durchlaufen und zeigt die Programmadresse und den dort stehenden Befehl (!PE) an. Wird der nächste Haltepunkt – bedingt durch eine lange Zykluszeit – nach einer bestimmten Zeit nicht erreicht, so kann der Anzeigevorgang, falls gewünscht, durch Eingabe von <CTRL>C abgebrochen werden.

Einzelzyklus ausschalten:

Befehl:

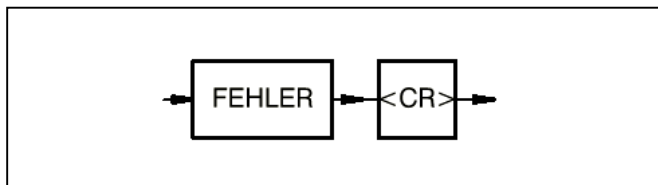


Funktion:

Der Einzelzyklusbetrieb wird ausgeschaltet, d. h. das Programm wird wieder normal abgearbeitet.

Anzeigen des Inhalts des Fehlerregisters

Befehl:



Funktion:

Die in der SPS gespeicherte Fehlerinformation wird ausgegeben.

Forcen von E/A-Signalen

Bei der SPS kann der Benutzer Eingangssignale und Ausgangssignale "forcen". Dabei werden für E/A-Signale vom Benutzer Werte vorgegeben. Die SPS arbeitet dann anstatt mit den wirklichen Eingangssignalen mit den Force-Werten. An die Ausgabegeräte werden von der SPS nicht die im SPS-Programm errechneten Ausgangssignale gegeben, sondern wiederum die Force-Werte. Die Force-Werte gelten solange, bis das Forcen für einzelne E/A-Signale oder für alle E/A-Signale zurückgenommen wird. Sowohl die von Eingabegeräten gelieferten Werte als auch die im SPS-Programm auf Ausgänge zugewiesenen Werte sind während des Forcens also ohne Einfluß. Forcen ist anwendbar sowohl auf binäre E/A-Signale als auch auf Wort-E/A-Signale.

Maximale Anzahl der zu forcenden E/A-Signale:

- Binäre Eingänge: 64
- Wort-Eingänge: 16
- Binäre Ausgänge: 64
- Wort-Ausgänge: 16

Das Forcen erfolgt auf folgende Art und Weise:

Forcen von Eingängen

Die SPS erstellt jeweils am Anfang des Programmzyklus ein Abbild der im SPS-Programm projektierten Eingangssignale. Falls Eingänge zu forcen sind, werden nach dem Einlesen deren wirkliche Werte durch die vom Benutzer vorgegebenen Force-Werte ersetzt. Die SPS arbeitet während des Programmzyklus ausschließlich mit dem modifizierten Eingabe-Abbild, wodurch Signaländerungen am Eingabegerät während des Programmzyklus keine Rolle spielen.

Forcen von Ausgängen

Am Ende des Programmzyklus überträgt die SPS das Ausgabe-Abbild der im SPS-Programm projektierten Ausgangssignale an die Ausgabe-Geräte. Falls Ausgänge zu forcen sind, werden deren wirklichen Werte vor deren Ausgabe im Ausgabe-Abbild durch die Force-Werte ersetzt.

Verhalten nach Spannungsausfall, RESET oder Warmstart

Nach einem Spannungsausfall hat die SPS den Force-Auftrag "vergessen". Die vor dem Spannungsausfall eingegebene Liste der zu forcenden E/A-Signale ist aber in der SPS noch vorhanden und kann mit dem Befehl FORC A <CR> auch angezeigt werden. Durch Eingabe eines einzigen zu forcenden Signals wird die gesamte Force-Liste wieder reaktiviert und das Forcen wieder wirksam.

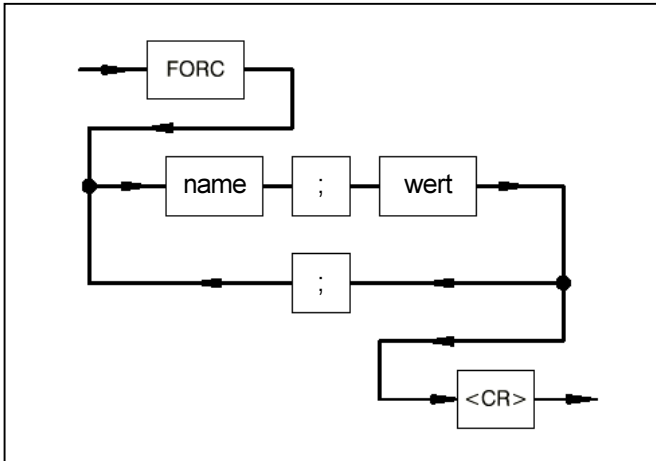
Für das Forcen von E/A-Signalen stehen folgende Befehle zur Verfügung:

- FORC: Force-Wert eingeben
- FORC A: Force-Wert anzeigen
- FORC R: Forcen löschen

Force-Wert eingeben

Der Name des zu forcenden E/A-Signals und der Force-Wert werden mit dem Befehl FORC eingeben.

Befehl:



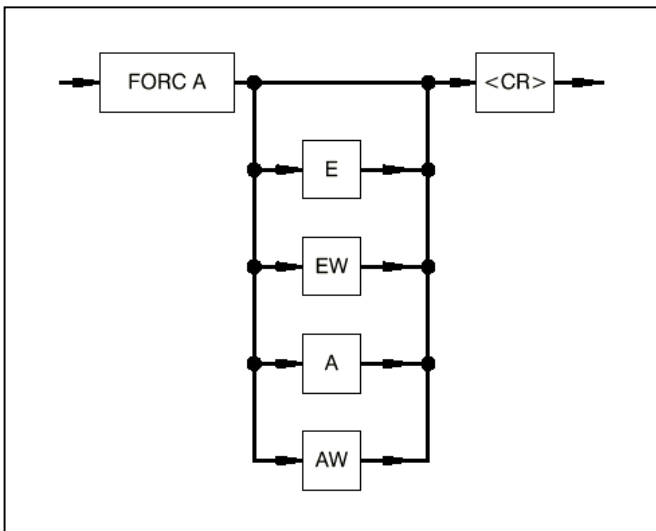
name: Name des zu forcenden Ein- oder Ausgangs-Signals

wert: Force-Wert für den Ein- oder Ausgang

:: Zwischen dem Namen und dem Force-Wert steht als Trennzeichen ein Semikolon. Sind mehrere Ein-/Ausgänge zu forcen, müssen diese ebenfalls durch ein Semikolon getrennt werden.

Force-Wert anzeigen

Befehl:

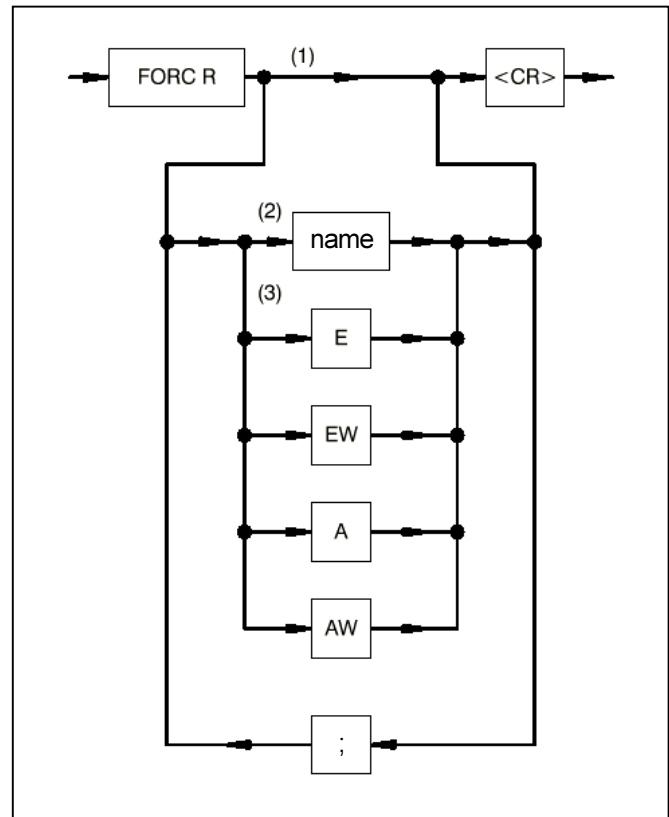


Funktion:

- Anzeigen aller zu forcenden Ein- und Ausgänge
- Anzeigen aller zu forcenden Ein- und Ausgänge einer bestimmten Gruppe von Ein-/Ausgängen

Forcen löschen

Befehl:



name: Name der Ein-/Ausgänge, für die das Forcen zu beenden ist

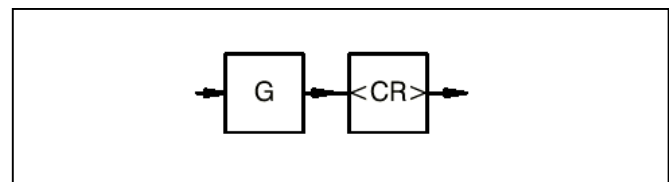
:: Wird das Forcen nur für bestimmte Ein-/Ausgänge beendet, müssen die einzelnen Namen bei der Eingabe durch ein Semikolon getrennt werden.

Funktion:

- (1) Beenden des Forcens für alle E/A-Signale
- (2) Beenden des Forcens für einzelne E/A-Signale
- (3) Beenden des Forcens für eine bestimmte Gruppe von E/A-Signalen

Anwenderprogramm starten

Befehl:



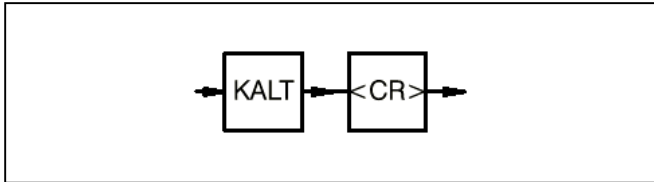
Funktion:

Das Anwenderprogramm wird gestartet, und die Operanden werden initialisiert.

Die Initialisierung der Operandenbereiche wird gemäß der entsprechenden Systemkonstanten durchgeführt.

Kaltstart durchführen

Befehl:



Funktion:

Der Kaltstartbefehl ist nur zulässig, wenn das SPS-Programm "abgebrochen" ist.

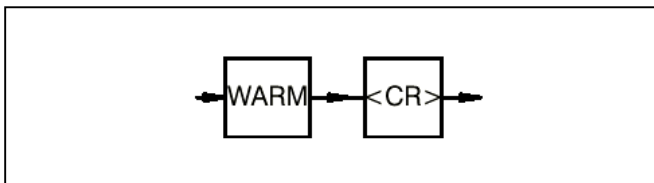
- Alle RAM-Speicher werden getestet und gelöscht.
- Falls im Flash-EPROM kein Anwenderprogramm vorhanden ist, werden für alle Systemkonstanten die Defaultwerte eingestellt (wie im Auslieferungszustand).
- Falls im Flash-EPROM ein Anwenderprogramm vorhanden ist, wird dieses einschließlich der Systemkonstanten ins RAM geladen.
- Die durch die Systemkonstanten vorgegebenen Betriebsarten werden eingestellt.
- Der CS31-Systembus wird neu initialisiert (nur bei Einsatz als Master am CS31-Systembus).

Auslösung eines Kaltstarts

- Befehl KALT <CR> im Terminalbetrieb oder
- Spannung AUS/EIN, wenn keine Batterie vorhanden ist oder
- Menüpunkt "Kaltstart" im Programmiersystem

Warmstart durchführen

Befehl:



Funktion:

Der Warmstartbefehl ist nur zulässig, wenn das SPS-Programm "abgebrochen" ist.

Warmstart

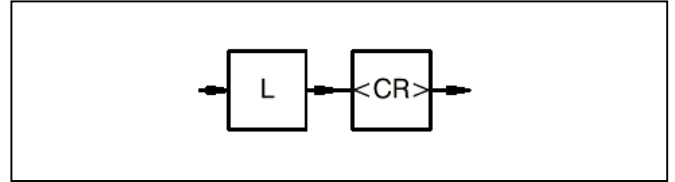
- Alle RAM-Speicher mit Ausnahme des Programmspeichers und des Operandenspeichers (Merker) werden getestet und gelöscht.
- Falls im Flash-EPROM ein Anwenderprogramm vorhanden ist, wird dieses einschließlich der Systemkonstanten ins RAM geladen.
- Die durch die Systemkonstanten vorgegebenen Betriebsarten werden eingestellt.
- Der CS31-Systembus wird neu initialisiert (nur bei Einsatz als Master am CS31-Systembus).

Auslösung eines Warmstarts

- Befehl WARM <CR> im Terminalbetrieb oder
- Spannung AUS/EIN, wenn Batterie vorhanden ist oder
- Menüpunkt "SPS-Art freigeben" im Programmiersystem

Anwenderprogramm fortsetzen

Befehl:



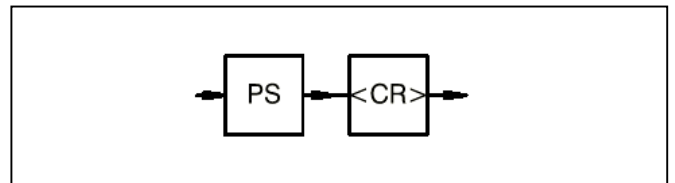
Funktion:

Nach einem vorangegangenen Stoppen ("W") wird das Anwenderprogramm wieder fortgesetzt. Die Merker und internen Zustände haben beim Fortsetzen den gleichen Wert wie beim Programm-Stopp.

Gestartete Zeitwerke laufen unabhängig vom Programmzustand im Betriebssystem weiter. Sie werden nur durch einen Kaltstart oder Spannung AUS/EIN abgebrochen.

Programmstatus anzeigen

Befehl:

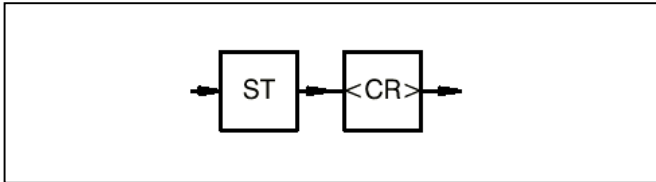


Funktion:

Der Programm-Status (Programm auf Breakpoint, Programm abgebrochen, Programm gestoppt, Programm läuft) des Anwenderprogrammes wird angezeigt.

SPS-Status anzeigen

Befehl:



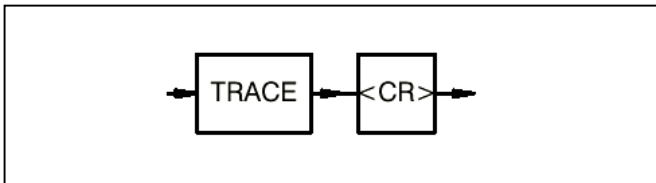
Funktion:

Der gesamte SPS-Status wird wie folgt angezeigt:

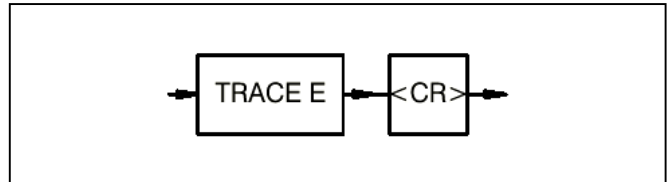
- Programmidentifikation
- Zykluszeit
- Programmstatus
- Aktive Testfunktionen
- TRACE-Register
- Fehlermeldungen
- Auslastung

TRACE-Betrieb

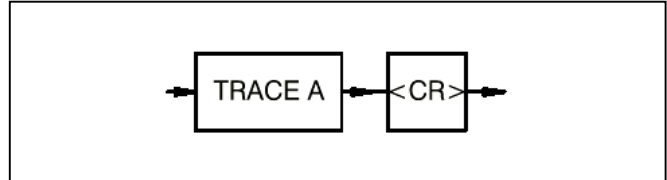
Befehl: TRACE-Speicher anzeigen



Befehl: TRACE-Betrieb einschalten



Befehl: TRACE-Betrieb ausschalten

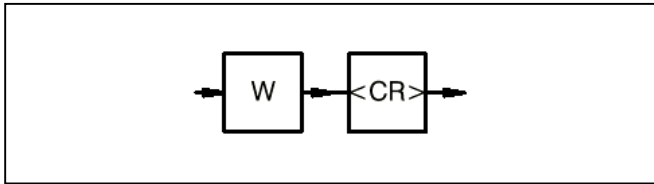


Funktion:

Im TRACE-Betrieb merkt sich die SPS die Adresse des zuletzt ausgeführten Bausteins bzw. der zuletzt ausgeführten Anweisung. Nach einem Systemabsturz hat man somit die Information, wie weit das Anwenderprogramm bearbeitet wurde. Der TRACE-Speicher-Inhalt bleibt bei einem RESET erhalten.

Anwenderprogramm stoppen

Befehl:



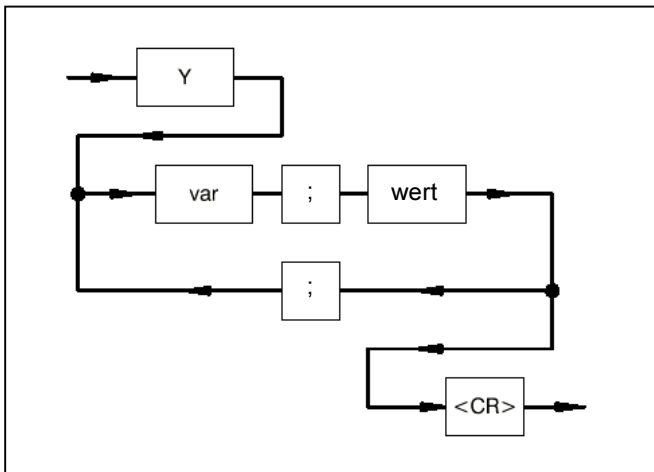
Funktion:

Das Anwenderprogramm wird gestoppt.

Die Werte der Ausgänge und der Merker bleiben erhalten. Gestartete Zeitwerke laufen unabhängig vom Programmzustand im Betriebssystem weiter. Sie werden nur durch einen Kaltstart oder Spannung AUS/EIN abgebrochen.

Wert einer Variablen mit einem vorzugebenden Wert überschreiben

Befehl:



var: Name der Variablen oder indirekten Konstanten

wert: neuer Wert, der der Variablen zugewiesen werden soll

:: Zwischen dem Namen und dem Wert der Variablen muß ein Semikolon stehen. Sollen mehrere Variablen überschrieben werden, müssen diese ebenfalls durch ein Semikolon getrennt werden.

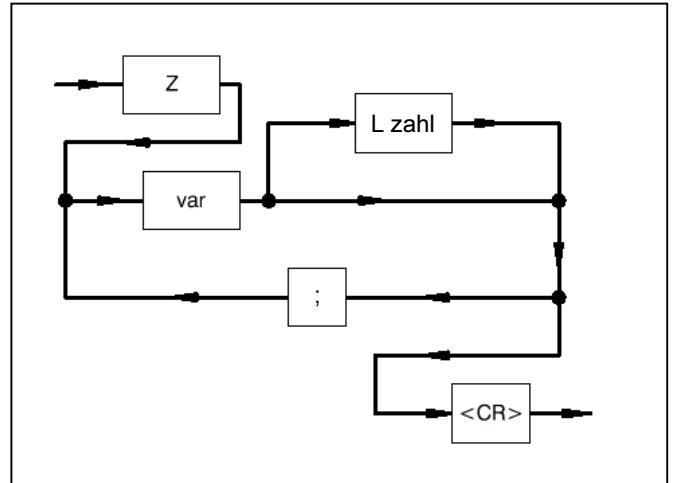
Anmerkung:

Handelt es sich bei der Variablen um eine Schrittvariable, so kann diese nur gesetzt, nicht aber zurückgesetzt werden. Beim Setzen der Schrittvariablen werden automatisch alle anderen Schritte der Kette zurückgesetzt.

Wird eine indirekte Konstante mit diesem Befehl geändert, so erfolgt diese Änderung nur im Operandenspeicher und nicht im Programmspeicher. D. h. nach dem nächsten Programmstart wird dieser Wert wieder durch denjenigen aus dem Programmspeicher überschrieben.

Status von Variablen anzeigen

Befehl:

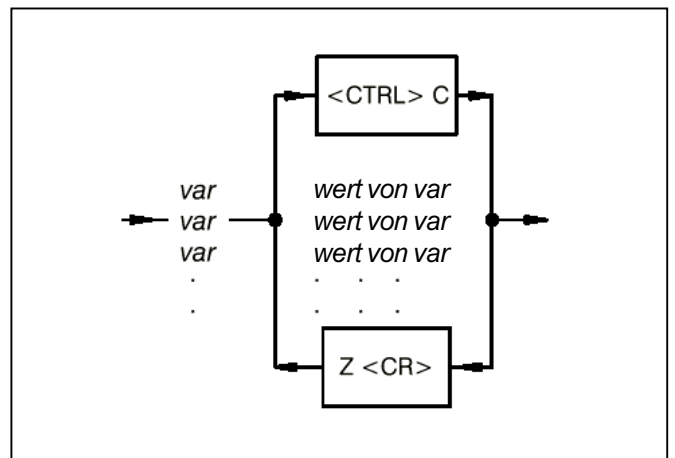


var: Variable (Merker, Schritt, Eingang, Ausgang, indirekte Konstante), die angezeigt werden soll

:: Die einzelnen Variablen müssen durch Semikolon getrennt sein.

L zahl: Anzahl der fortlaufend nummerierten Variablen ab der Variablen var, die angezeigt werden sollen.

Bsp.: M 0,0 L 3
es wird angezeigt:
M 0,0 M 0,1 M 0,2



Z: Die Werte der Variablen (max. 22) werden jeweils bei Eingabe des Zeichens Z <CR> aktualisiert.

Funktion:

Die vom Anwender vorgegebenen Variablenamen werden auf dem Bildschirm ausgegeben. Der Wert dieser Variablen wird nach jeder Eingabe der Zeichens Z <CR> aktualisiert. Die angezeigten Variablenwerte stammen immer aus dem selben Programmzyklus und stellen eine Momentaufnahme am Zyklusende dar.

Die Anzahl der anzuzeigenden Variablen wird bei diesem Kommando auf 22 beschränkt, da nicht mehr Bildschirmzeilen zur Verfügung stehen.

Rechneranschluß anstatt Terminal

Wird zur Auswertung der Statuswerte anstelle des Terminals ein Rechner angeschlossen, so können bei Bedarf anstatt Z auch folgende Kommandos benutzt werden (gleiches Syntax-Diagramm wie bei Kommando Z):

ZO: Anzahl der möglichen Variablen maximal 120, ansonsten wie Kommando Z.

Bildschirmsteuerung: Bei den Kommandos Z, ZO und ZD werden von der SPS zur Bildschirmsteuerung folgende Steuerzeichen benutzt:

Wagenrücklauf: <CR>

Zeilenvorschub: <LF>

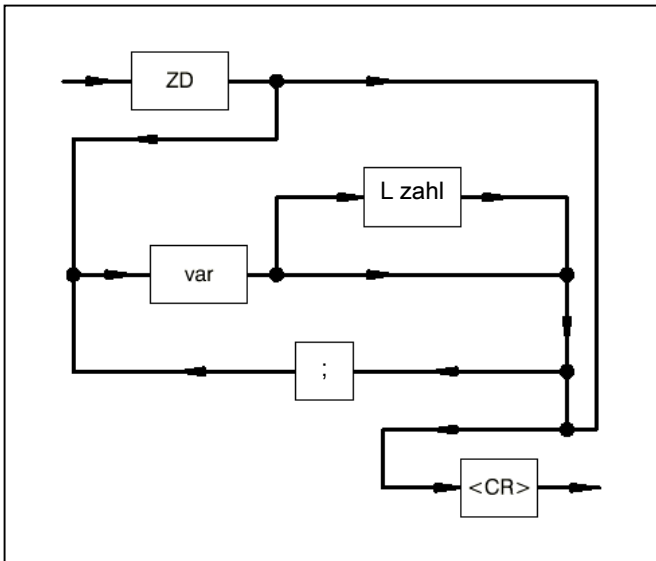
Bidschirm löschen: <ESC>[2J

Cursor positionieren: <ESC>[<zeile>;<spalte>H

ZZ: Anzahl der möglichen Variablen maximal 120. Die SPS schickt keine ESC-Sequenzen zur Bildschirmsteuerung, sondern nur die Variablenwerte jeweils gefolgt von einem <CR>. Die Variablenwerte haben die gleiche Reihenfolge wie die vorgegebene Variablenliste, sonst wie Kommando Z.

Status von Variablen anzeigen und laufend aktualisieren

Befehl:

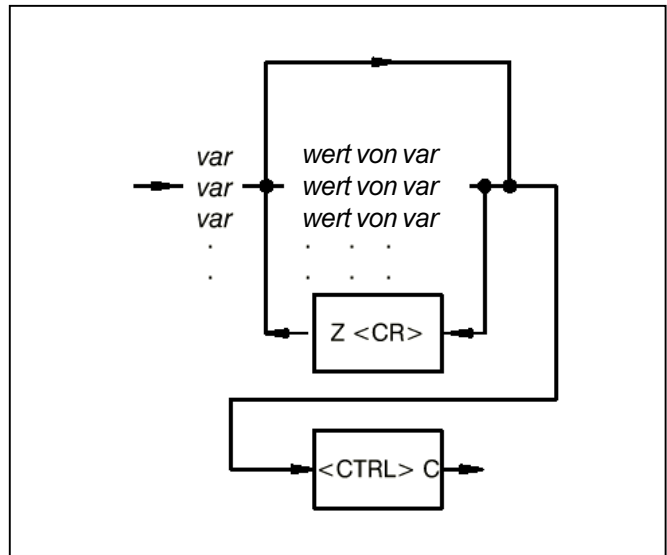


var: Variable (Merker, Eingang, Ausgang, indirekte Konstante), die angezeigt werden soll

;; Die einzelnen Variablen müssen durch Semikolon getrennt sein.

L zahl: Anzahl der fortlaufend nummerierten Variablen ab der Variablen var, die angezeigt werden sollen.

Bsp.: M 0,0 L 3
es wird angezeigt:
M 0,0 M 0,1 M 0,2



Funktion:

Die vom Anwender vorgegebenen Variablenamen werden auf dem Bildschirm ausgegeben. Die zugehörigen Variablenwerte werden automatisch aktualisiert. Die angezeigten Variablenwerte stammen immer aus dem selben Programmzyklus und stellen eine Momentaufnahme am Zyklusende dar.

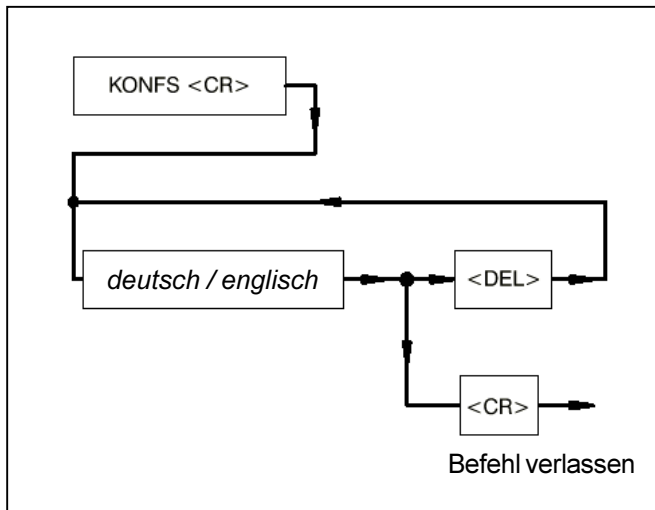
Die maximale Anzahl ist 22. Durch ein <CTRL>C wird der Befehl beendet.

Bei anschließender Eingabe der Zeichen Z <CR> oder ZD <CR> wird die Statusanzeige für die vorher eingegebenen Variablen reaktiviert.

2.4.3 Kommandos zum Konfigurieren

Betriebsarten anzeigen/ändern

Befehl:



Funktion:

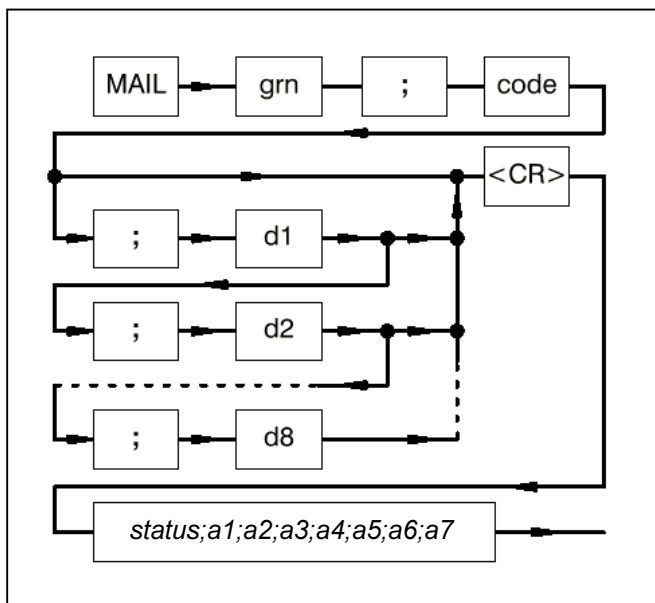
Nach Eingabe des Befehls KONFS <CR> wird auf dem Bildschirm die eingestellte Sprache angezeigt. Durch Drücken der Taste <DELETE> (bei PCs <CTRL> und Löschtaste) wird die Sprache umgeschaltet. Durch Eingabe eines <CR> wird der Befehl beendet.

Anmerkung:

Bei Personal-Computern fehlt oft die DELETE-Taste. Der Tastencode (7F_H) der DELETE-Taste kann dort durch Drücken zweier Tasten erzeugt werden. In der Regel handelt es sich dabei um die Tasten <CTRL> und die Löschtaste.

Konfiguration/Abfrage der Konfiguration von CS31-Vor-Ort-Modulen (07 KR 31, 07 KR 91, 07 KT 92, 07 KT 93, 07 KT 94)

Befehl:



grn: Gruppennummer, mit der das Vorortmodul vom SPS-Programm aus angesprochen wird

code: Auftragskennung

d1: 1. Datenbyte des Auftrags

:

d8: 8. Datenbyte des Auftrags

:: Die einzelnen Werte des Auftrages müssen durch Semikolon getrennt sein.

status: Status der Antwort:

51 (OK-Antwort)

170 (Nicht-OK-Antwort)

a1: 1. Datenbyte der Antwort

:

a7: 7. Datenbyte der Antwort

:: Die einzelnen Werte der Antwort sind durch Semikolon getrennt.

Funktion:

Der Anwender hat die Möglichkeit, CS31-Vor-Ort-Module zu konfigurieren und die eingestellte Konfiguration abzufragen. Die Aufträge werden intern über eine Sende-Mail-Box (Auftrag) und eine Empfangs-Mail-Box (Antwort) abgewickelt.

Liste der Aufträge:

Die OK-Antworten werden bei den jeweiligen Aufträgen beschrieben. Die Nicht-OK-Antworten der einzelnen Aufträge sehen immer wie folgt aus:

- **Nicht-OK-Antwort**

status: 170

a1: 1 = unbekannte Auftragskennung

2 = ungültiger Parameter,
z. B. Gruppennummer

3 = Vorortmodul antwortet nicht

10 = Mail Box ist innerhalb von 3 sec. nicht frei

11 = Auftrag wurde durch Betätigung des RUN/STOP-Schalters abgebrochen

12 = Auftrag wird innerhalb von 6 sec. nicht abgeholt

13 = keine Antwort innerhalb von 6 sec.

a2...a7: 0

- **Aktualisierung der maximalen Anzahl erkannter Vor-Ort-Module**

Im Eingangswort EW 07,15 steht u. a. die maximale Anzahl der in der Vergangenheit erkannten Vor-Ort-Module. Die momentane wirkliche Anzahl der vorhandenen Vor-Ort-Module kann geringer sein.

Mit diesem Kommando wird dieser Wert aktualisiert. Die vorhandenen Module werden gezählt und der Wert abgelegt.

Der Anwender kann diesen Wert im SPS-Programm (EW 07,15, Bit 8...15) abfragen.

- **Auftrag**
grn: 255 (Master-SPS mit Bus)
code: 132
d1...d8: nicht benutzt
- **OK-Antwort**
status: 51
a1...a7: 0
- **Abfrage, ob bei einem Eingang die Drahtbruchüberwachung ein- oder ausgeschaltet ist**
 - **Auftrag**
grn: Gruppennummer 0...63
code: 32
d1: Kanalnummer 0...15
d2...d8: nicht benutzt
 - **OK-Antwort**
status: 51
a1: 47 = Drahtbruchüberwachung EIN
32 = Drahtbruchüberwachung AUS
a2...a7: 0
- **Abfrage, ob bei einem Ausgang die Drahtbruchüberwachung ein- oder ausgeschaltet ist**
 - **Auftrag**
grn: Gruppennummer 0...63
code: 33
d1: Kanalnummer 0...15
d2...d8: nicht benutzt
 - **OK-Antwort**
status: 51
a1: 47 = Drahtbruchüberwachung EIN
32 = Drahtbruchüberwachung AUS
a2...a7: 0
- **Drahtbruchüberwachung eines Einganges ein- bzw. ausschalten**
 - **Auftrag**
grn: Gruppennummer 0...63
code: 224 = Drahtbruchüberwachung EIN
160 = Drahtbruchüberwachung AUS
d1: Kanalnummer 0...15
d2...d8: nicht benutzt
 - **OK-Antwort**
status: 51
a1...a7: 0

- **Drahtbruchüberwachung eines Ausganges ein- bzw. ausschalten**
 - **Auftrag**
grn: Gruppennummer 0...63
code: 225 = Drahtbruchüberwachung EIN
161 = Drahtbruchüberwachung AUS
d1: Kanalnummer 0...15
d2...d8: nicht benutzt
 - **OK-Antwort**
status: 51
a1...a7: 0
- **Abfrage, ob ein Kanal als Eingang oder als Eingang/Ausgang konfiguriert ist**
 - **Auftrag**
grn: Gruppennummer 0...63
code: 34
d1: Kanalnummer 0...15
d2...d8: nicht benutzt
 - **OK-Antwort**
status: 51
a1: 34 = Eingang
35 = Eingang/Ausgang
a2...a7: 0
- **Konfiguration eines Kanals als Eingang oder Eingang/Ausgang**
 - **Auftrag**
grn: Gruppennummer 0...63
code: 162 = Eingang
163 = Eingang/Ausgang
d1: Kanalnummer 0...15
d2...d8: nicht benutzt
 - **OK-Antwort**
status: 51
a1...a7: 0
- **Abfragen der Eingangsverzögerung eines Kanals**
 - **Auftrag**
grn: Gruppennummer 0...63
code: 38
d1: Kanalnummer 0...15
d2...d8: nicht benutzt
 - **OK-Antwort**
status: 51
a1: Eingangsverzögerung:
2 = 2 ms
4 = 4 ms
.
.
.
30 = 30 ms
32 = 32 ms
a2...a7: 0

- **Einstellen der Eingangsverzögerung eines Kanals**

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 166
 - d1: Kanalnummer 0...15
 - d2: Eingangsverzögerung
2 = 2 ms
4 = 4 ms
.
.
.
30 = 30 ms
32 = 32 ms
 - d3...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1...a7: 0

- **Fehler auf Vorortmodul quittieren**

Mit diesem Kommando werden die auf dem gewählten Vor-Ort-Modul registrierten Fehlermeldungen zurückgesetzt. Ein Rücksetzen ist nur möglich, wenn die Fehlerursache nicht mehr wirksam ist.

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 232
 - d1: kleinste Kanalnummer auf dem Modul:
0 = kleinste Kanalnummer auf dem Modul ist 0 (≤ 7)
8 = kleinste Kanalnummer auf dem Modul ist 8 (> 7)
 - d2: Modultyp:
0 = Digitale Eingabe
1 = Analoge Eingabe
2 = Digitale Ausgabe
3 = Analoge Ausgabe
4 = Digitale Ein-/Ausgabe
5 = Analoge Ein-/Ausgabe

- Hinweis:
Bit: gerade Zahl (0, 2, 4)
Wort: ungerade Zahl (1, 3, 5)

- d3...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1...a7: 0

- **Fehler auf Vorortmodul quittieren und Konfigurationswerte auf Standardeinstellung (Default) zurücksetzen**

Zusätzlich zum Auftrag "Fehler auf Vorortmodul quittieren" werden alle konfigurierbaren Einstellungen auf die Standardeinstellung zurückgesetzt.

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 233
 - d1: erste Kanalnummer auf dem Modul:
0 = erste Kanalnummer auf dem Modul ist 0 (≤ 7)
8 = erste Kanalnummer auf dem Modul ist 8 (> 7)

- d2: Modultyp:
0 = Digitale Eingabe
1 = Analoge Eingabe
2 = Digitale Ausgabe
3 = Analoge Ausgabe
4 = Digitale Ein-/Ausgabe
5 = Analoge Ein-/Ausgabe

- Hinweis:

- Bit: gerade Zahl (0, 2, 4)

- Wort: ungerade Zahl (1, 3, 5)

- d3...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1...a7: 0

- **Abfragen der Konfiguration eines analogen Einganges**

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 42
 - d1: Kanalnummer 0...15
 - d2...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1: 51 = Eingang ± 10 V
50 = Eingang 0...20 mA
49 = Eingang 4...20 mA
 - a2...a7: 0

- **Abfragen der Konfiguration eines analogen Ausganges**

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 43
 - d1: Kanalnummer 0...15
 - d2...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1: 50 = Ausgang 0...20 mA
49 = Ausgang 4...20 mA
51 = Ausgang ± 10 V
 - a2...a7: 0

- **Konfiguration eines analogen Einganges**

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 170
 - d1: Kanalnummer 0...15
 - d2: 51 = Eingang 10 V
50 = Eingang 0...20 mA
49 = Eingang 4...20 mA
 - d3...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1...a7: 0

- **Konfiguration eines analogen Ausganges**

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 171
 - d1: Kanalnummer 0...15
 - d2: 50 = Ausgang 0...20 mA
49 = Ausgang 4...20 mA
51 = Ausgang $\pm 10V$
 - d3...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1...a7: 0

- **Abfragen der Bus-Konfiguration**

Die Busanschaltung der Master-SPS verfügt über eine Liste, in der bestimmte Daten der Vor-Ort-Module abgelegt sind. Die Vor-Ort-Module sind in dieser Liste in der Reihenfolge numeriert, in welcher sie am CS31-Bus aufgefunden werden. Bei diesem Kommando wird die interne Nummer der Module angegeben. Als Antwort erhält man die unter dieser Nummer gespeicherte Gruppennummer sowie Statusinformationen zu dem entsprechenden Modul.

- **Auftrag**

- grn: 0 (wird nicht ausgewertet)
 - code: 80
 - d1: Nummer aus der Modulliste (1...31)
 - d2...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1: Status des Vorortmoduls:
Bit 0...3: Anzahl der Prozeßdaten-Bytes (Binärmodul) bzw. Worte (Wortmodul), die das Modul dem Master sendet
Bit 4...7: Anzahl der Prozeßdaten-Bytes (Binärmodul) bzw. Worte (Wortmodul), die der Master dem Modul sendet
 - a2: Gruppennummer (0...63)
 - a3: Bit 0: 0 = kleinste Kanalnummer ≤ 7
1 = kleinste Kanalnummer > 7
Bit 1: 0 = Binärmodul
1 = Wortmodul
 - a4...a7: 0

- **1...6 Bytes lesen (07 KR 91, 07 KT 92, 07 KT 93) und 07 KT 94**

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 49 = 1 Byte lesen
50 = 2 Bytes lesen
51 = 3 Bytes lesen
52 = 4 Bytes lesen
53 = 5 Bytes lesen
54 = 6 Bytes lesen
 - d1: erste Kanalnummer auf dem Modul:
0 = erste Kanalnummer auf dem Modul ist 0 (≤ 7)
8 = erste Kanalnummer auf dem Modul ist 8 (> 7)

- d2: Modultyp:
0 = Digitale Eingabe
1 = Analoge Eingabe
2 = Digitale Ausgabe
3 = Analoge Ausgabe
4 = Digitale Ein-/Ausgabe
5 = Analoge Ein-/Ausgabe

Hinweis:

Bit: gerade Zahl (0, 2, 4)
Wort: ungerade Zahl (1, 3, 5)

- d3: Byte-Anfangsadresse (Low-Byte)
 - d4: Byte-Anfangsadresse (High-Byte)
 - d5...d8: nicht benutzt

- **OK-Antwort**

- status: 51
 - a1: Wert des 1. Bytes
 - a2: Wert des 2. Bytes oder 0
 - a3: Wert des 3. Bytes oder 0
 - a4: Wert des 4. Bytes oder 0
 - a5: Wert des 5. Bytes oder 0
 - a6: Wert des 6. Bytes oder 0
 - a7: 0

- **1 Bit von einem Byte lesen**

- **Auftrag**

- grn: Gruppennummer 0...63
 - code: 63
 - d1: erste Kanalnummer auf dem Modul:
0 = erste Kanalnummer auf dem Modul ist 0 (≤ 7)
8 = erste Kanalnummer auf dem Modul ist 8 (> 7)

- d2: Modultyp:
0 = Digitale Eingabe
1 = Analoge Eingabe
2 = Digitale Ausgabe
3 = Analoge Ausgabe
4 = Digitale Ein-/Ausgabe
5 = Analoge Ein-/Ausgabe

Hinweis:

Bit: gerade Zahl (0, 2, 4)
Wort: ungerade Zahl (1, 3, 5)

- d3: Byte-Anfangsadresse (Low Byte)
- d4: Byte-Anfangsadresse (High Byte)
- d5: Bitposition innerhalb des Bytes 0...7
- d6...d8: nicht benutzt

– **OK-Antwort**

- status: 51
- a1: Bitwert (0 oder 1)
- a2...a7: 0

• **1...4 Bytes schreiben (07 KR 91, 07 KT 92, 07 KT 93 und 07 KT 94)**

– **Auftrag**

- grn: Gruppennummer 0...63
- code: 65 = 1 Byte schreiben
- 66 = 2 Bytes schreiben
- 67 = 3 Bytes schreiben
- 68 = 4 Bytes schreiben

- d1: erste Kanalnummer auf dem Modul:
 - 0 = erste Kanalnummer auf dem Modul ist 0 (≤ 7)
 - 8 = erste Kanalnummer auf dem Modul ist 8 (> 7)

- d2: Modultyp:
 - 0 = Digitale Eingabe
 - 1 = Analoge Eingabe
 - 2 = Digitale Ausgabe
 - 3 = Analoge Ausgabe
 - 4 = Digitale Ein-/Ausgabe
 - 5 = Analoge Ein-/Ausgabe

Hinweis:

- Bit: gerade Zahl (0, 2, 4)
- Wort: ungerade Zahl (1, 3, 5)

- d3: Byte-Anfangsadresse (Low-Byte)
- d4: Byte-Anfangsadresse (High-Byte)
- d5: Wert des 1. Bytes
- d6: Wert des 2. Bytes oder nicht benutzt
- d7: Wert des 3. Bytes oder nicht benutzt
- d8: Wert des 4. Bytes oder nicht benutzt

– **OK-Antwort**

- status: 51
- a1...a7: 0

• **1 Bit von einem Byte schreiben**

– **Auftrag**

- grn: Gruppennummer 0...63
- code: 79
- d1: erste Kanalnummer auf dem Modul:
 - 0 = erste Kanalnummer auf dem Modul ist 0 (≤ 7)
 - 8 = erste Kanalnummer auf dem Modul ist 8 (> 7)

- d2: Modultyp:
 - 0 = Digitale Eingabe
 - 1 = Analoge Eingabe
 - 2 = Digitale Ausgabe
 - 3 = Analoge Ausgabe
 - 4 = Digitale Ein-/Ausgabe
 - 5 = Analoge Ein-/Ausgabe

Hinweis:

- Bit: gerade Zahl (0, 2, 4)
- Wort: ungerade Zahl (1, 3, 5)

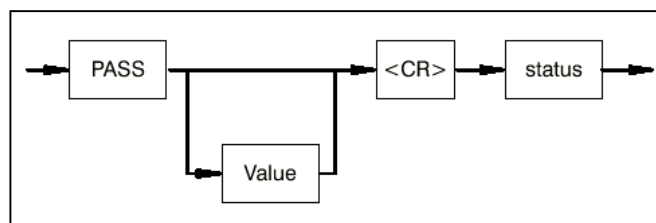
- d3: Byte-Anfangsadresse (Low-Byte)
- d4: Byte-Anfangsadresse (High-Byte)
- d5: Bitposition innerhalb des Bytes 0...7
- d6: Bitwert (0 oder 1)
- d7...d8: nicht benutzt

– **OK-Antwort**

- status: 51
- a1...a7: 0

Paßwort (nur für Serie 30, 40, 50)

Befehl:



Funktion:

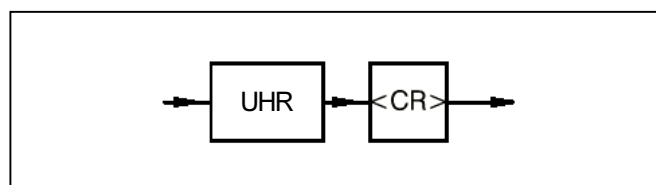
Der Befehl PASS schaltet die Paßwort-Funktion ein bzw. aus. Als Paßwort kann eine beliebige vierstellige Hexadezimalzahl (außer 0000) eingegeben werden. Ist ein Paßwort aktiviert, so sind folgende Funktionen gesperrt: AEND, D, DEEP, FREI, N, NOP, O, S, V.

Value: Eine beliebige vierstellige Hexadezimalzahl.
Achtung: Der Wert 0000 bewirkt nichts.

status: Das Ein-/Ausschalten der Paßwort-Funktion wird angezeigt.

Anzeigen von Uhrzeit und Datum (07 KR 91, 07 KT 92, 07 KT 93 und 07 KT 94)

Befehl:



Funktion:

Die Uhrzeit und das Datum werden in folgender Form auf dem Bildschirm angezeigt:

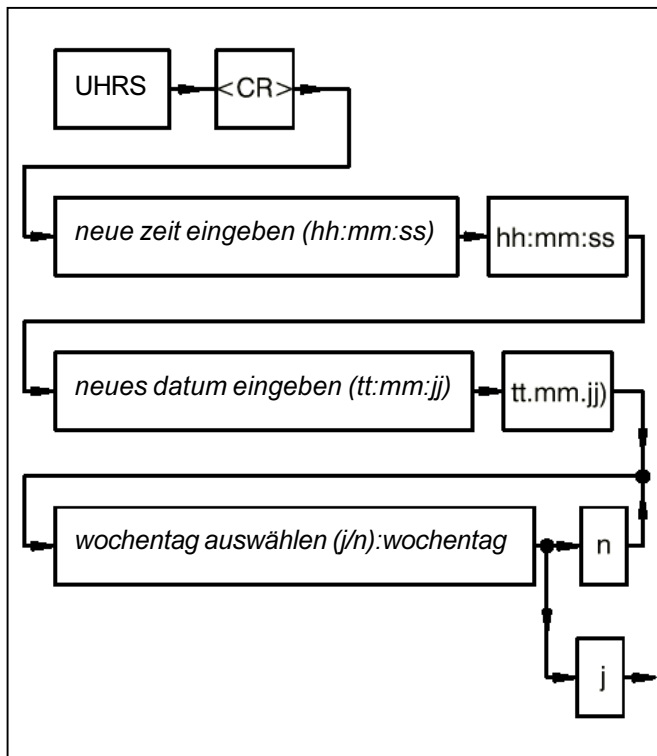
SYSTEMZEIT : HH:MM:SS
SYSTEMDATUM
: WOCHENTAG TT.MM.JJ

dabei ist:

HH: Stunden
MM: Minuten
SS: Sekunden
WOCHENTAG: Name des Wochentags
TT: Tag
MM: Monat
JJ: Jahr

Stellen von Uhrzeit und Datum

Befehl:



Funktion:

Stellen der Uhrzeit und des Datums. Für den Wochentag führt die Uhr intern eine Nummer zwischen 1 und 7. Sie geht beim Umsetzen der Nummer in den Namen davon aus, daß der Montag der erste Tag in der Woche ist (Nummer 1 --> Montag). Falls die Uhr mit dem Baustein UHR (s. a. Bausteinkatalog) gestellt wird, kann dort dem Montag eine andere Nummer zugeordnet werden. In diesem Fall stimmt die Anzeige des Wochentags mit dem Befehl UHR <CR> nicht mehr, da die Anzeigefunktion grundsätzlich davon ausgeht, daß dem Montag die Nummer 1 zugeordnet ist.

hh bzw. hh: Stunden
mm bzw. mm: Minuten
ss bzw. ss: Sekunden
tt bzw. tt: Tag
mm bzw. mm: Monat
jj bzw. jj: Jahr
wochentag: Name des Wochentags
n: Eingabe für 'nein'
j: Eingabe für 'ja'

3 Monitorfunktionen für 07 KR 91, 07 KT 92 bis 07 KT 94

Das Monitorprogramm bietet dem Spezialisten einen Zugang auf hexadezimaler Ebene zum gesamten Adreßbereich der SPS. Es können Speicherbereiche angezeigt und verändert, sowie Hardware-Tests durchgeführt werden.

Monitorbefehle, die Speicherbereiche verändern, können die Funktionalität der SPS gefährden. Aus diesem Grund ist im Umgang mit den Monitorfunktionen Vorsicht geboten.

- Die Geräte arbeiten bei den Monitorbefehlen mit 32-Bit-Adressen. Mit dem Y-Befehl wird das High-Wort der 32-Bit-Adresse eingestellt.

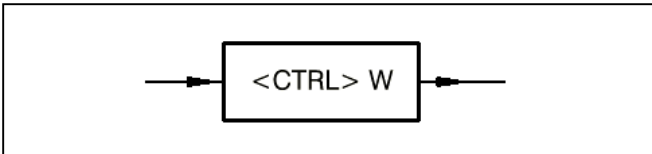
Beispiel:

Y = 20 <CR> } Es wird der Wert der Adresse
 DW0 <CR> } 0020 0000_H angezeigt.

Umschalten

Bedienfunktionen <--> Monitorfunktionen

Befehl:



Befehl bei 07 KR 31 nicht verfügbar.

Funktion:

Durch gleichzeitiges Drücken der Taste <CTRL> und der Taste W gelangt man in das Monitorprogramm der SPS. Befindet man sich im Monitorprogramm, gelangt man durch erneute Eingabe von <CTRL> und W wieder in das Bedienprogramm der SPS.

Erläuterung zur Syntax:

- Das Monitorprogramm meldet sich mit dem Zeichen * und wartet auf eine Eingabe.
- Alle Zahlen sind Hexadezimalzahlen (führende Nullen können entfallen).
- Werden mehr Ziffern als notwendig angegeben, haben nur die letzten Ziffern (bei Byte-Befehlen die letzten 2 Ziffern und bei Wort-Befehlen die letzten 4 Ziffern) Gültigkeit.

- Das Zeichen Blank (Space, Leerstelle, Leerzeichen) wird nicht beachtet und kann zur übersichtlicheren Eingabe benutzt werden.
- Das Zeichen CTRL-C führt zum Abbruch der gerade laufenden Operation.
- Jede Ausgabe auf den Bildschirm kann mit <CTRL>S (XOFF) angehalten und mit <CTRL>Q (XON) weitergeführt werden.
- Wird bei einer Adreßeingabe kein Segment angegeben, wird das Arbeitssegment verwendet (siehe Y-Anweisung).

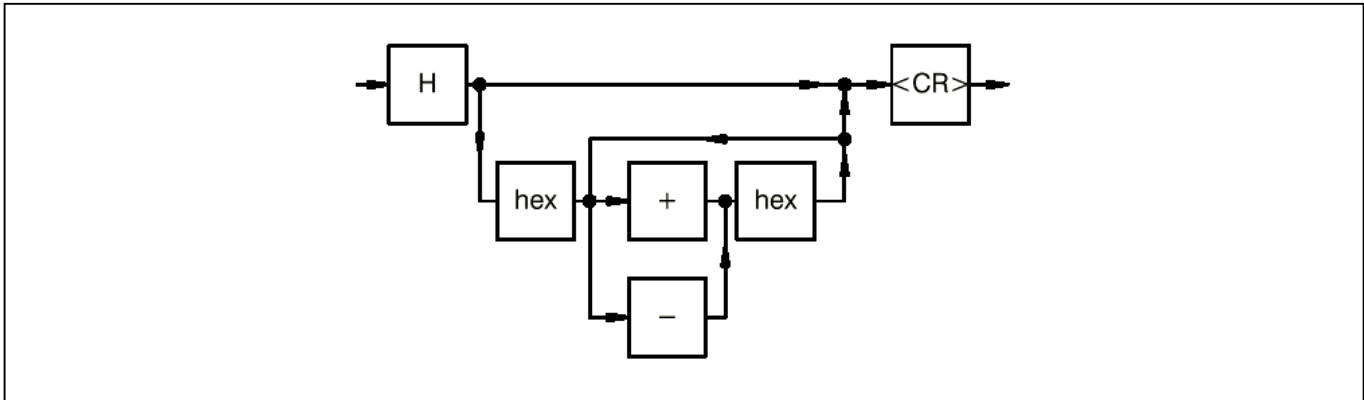
Übersicht über die Monitorfunktionen

Funktion	Erläuterung	Seite
H	Help-Text anzeigen/ Hexadezimal rechnen	42
D	Speicherinhalt anzeigen	42
I	Speicherbereich mit einem Wert füllen	43
M	Übertragen von Speicherbereichen	43
P	Port lesen/schreiben	44
S	Speicherinhalt anzeigen/ändern	45
Y	Arbeitssegment anzeigen/ändern	45
ZG	Gleichzeitige Ausgabe von 3 Werten	46
ZR	RAM-Test	46
ZZA	Ausgabe von 3 Werten nach Eingabe eines Semikolons (;)	47
ZZF	String suchen	47
ZZV	Speicherbefehle wortweise vergleichen	48
R	Intel-Hex-File lesen	49
W	Intel-Hex-File schreiben	49

Help-Text anzeigen / Hexadezimal rechnen

Nach der Eingabe des Befehls H <CR> werden auf dem Bildschirm alle verfügbaren Funktionen des Monitors angezeigt. Außerdem ermöglicht dieser Befehl die Berechnung von einfachen hexadezimalen arithmetischen Ausdrücken.

Befehl:

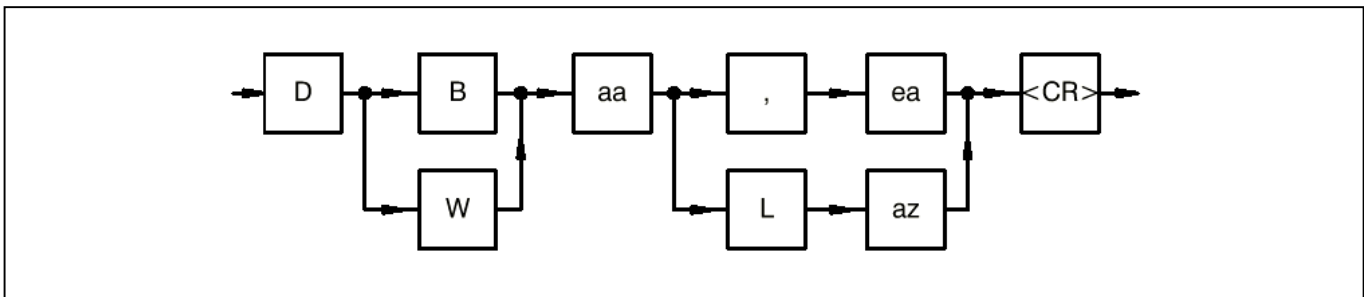


hex: Hexadezimalzahl

Speicherinhalt anzeigen

Der Speicherinhalt kann byte- oder wortweise angezeigt werden.

Befehl:



- B: Byteweise (Schlüsselwort)
- W: Wortweise (Schlüsselwort)
- aa: Anfangsadresse, ab der der Speicherinhalt angezeigt werden soll
- ,: Schlüsselwort (Trennzeichen)
- ea: Endadresse des auszugebenden Speicherinhalts
- L: Länge (Schlüsselwort)
- az: Anzahl der auszugebenden Bytes/Worte

Beispiel:

DB 0:0L2<CR> Speicherinhalt byteweise anzeigen

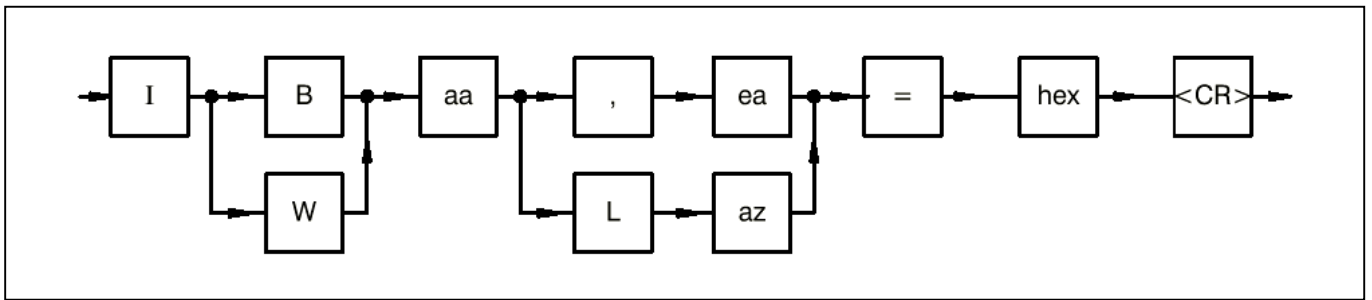
0000:0000 02 00 Bildschirmausgabe

DW 0,2<CR> Speicherinhalt wortweise anzeigen

0000:0000 0002 0000 Bildschirmausgabe

Speicherbereich mit einem Wert füllen

Befehl:



B: Byteweise (Schlüsselwort)

W: Wortweise (Schlüsselwort)

aa: Anfangsadresse, ab der der Speicherinhalt mit dem angegebenen Wert gefüllt werden soll

,: Schlüsselwort (Trennzeichen)

ea: Endadresse des Speicherbereichs

L: Länge (Schlüsselwort)

az: Anzahl der zu füllenden Bytes/Worte

hex: Hexadezimalwert, mit dem der Speicherbereich gefüllt werden soll

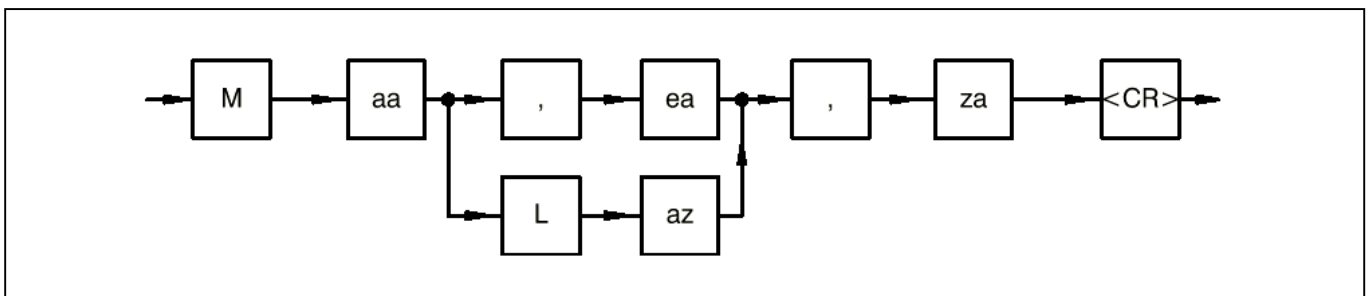
Beispiel:

IB 8000:80L3=FF<CR> Der Speicherinhalt von 8000:80_H, 8000:81_H und 8000:82_H wird mit FF überschrieben

Übertragen von Speicherbereichen

Ein Speicherbereich kann auf einen anderen Bereich kopiert werden. Die Übertragung geschieht wortweise, wobei die Anzahl bei der Eingabe aber in Byte angegeben wird (d. h. bei az = 3 wird 1 Wort übertragen).

Befehl:



aa: Anfangsadresse, ab der der Speicherinhalt kopiert werden soll

,: Schlüsselwort (Trennzeichen)

ea: Endadresse des Speicherbereichs

L: Länge (Schlüsselwort)

az: Anzahl der zu kopierenden Bytes

za: Zieladresse des Speicherbereichs

Beispiel:

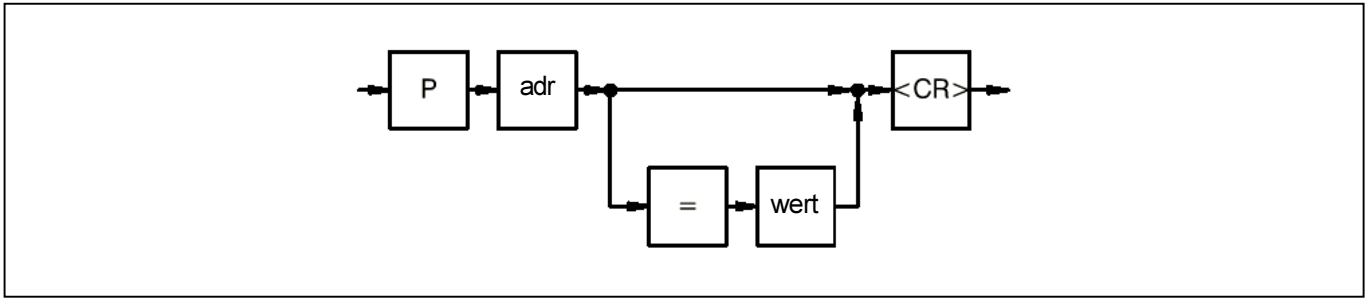
M 8000:80L4,8000:90<CR> oder M 8000:80,84,8000:90<CR>

Kopiert wird:

8000 _H :80 _H	→	8000 _H :90 _H
:81 _H	→	:91 _H
:82 _H	→	:92 _H
:83 _H	→	:93 _H

Port lesen/schreiben

Befehl:



adr: I/O-Adresse

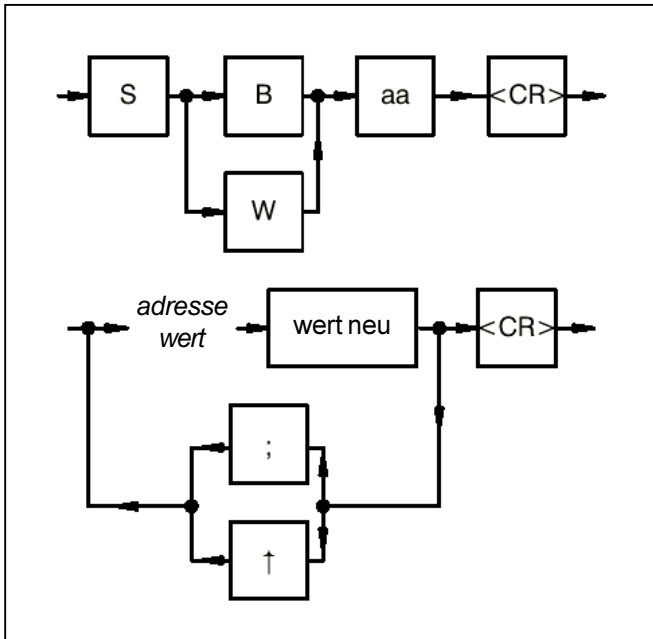
wert: Byte-Wert, der nach der I/O-Adresse geschrieben wird

=: Schlüsselwort

Speicherinhalt anzeigen/ändern

Der Speicherinhalt kann byte- oder wortweise angezeigt und geändert werden.

Befehl:



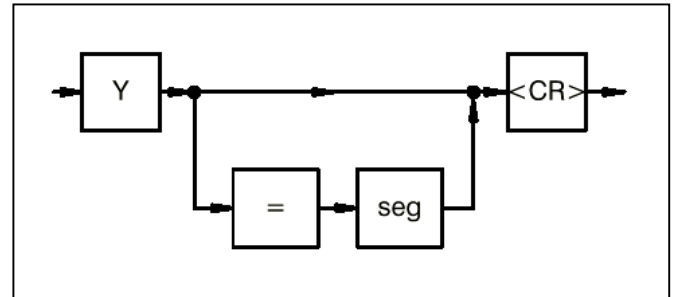
- B: Byteweise (Schlüsselwort)
W: Wortweise (Schlüsselwort)
aa: Anfangsadresse, ab der der Speicherinhalt angezeigt/geändert werden soll
adresse: Adresse des Speicherinhalts
wert: Wert des Speicherinhalts
wert neu: neuer Wert des Speicherinhalts (Anwendereingabe)
;: durch Eingabe eines Semikolons wird die Adresse um 1 (Befehl SB) oder um 2 (Befehl SW) inkrementiert
↑: durch Eingabe eines Hochpfeils (bei PC ^) wird die Adresse um 1 (Befehl SB) oder um 2 (Befehl SW) dekrementiert

Arbeitssegment anzeigen/ändern

Wird bei einer Adreßangabe kein Segment, sondern nur ein Offset angegeben, wird als Segment das Arbeitssegment Y verwendet. Der Defaultwert des Arbeitssegments ist Null.

Die Geräte arbeiten bei den Monitorbefehlen mit 32-Bit-Adressen. Mit dem Y-Befehl wird das High-Wort der 32-Bit-Adresse eingestellt.

Befehl:



seg: neue Segmentadresse des Arbeitssegments
=: Schlüsselwort

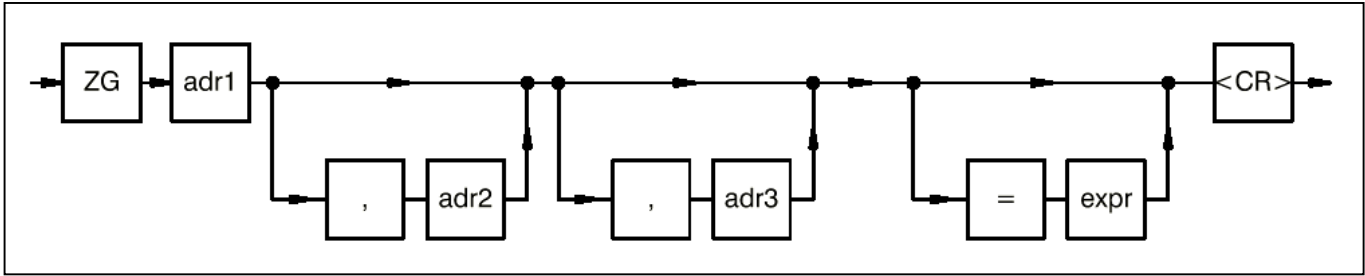
Beispiel:

- Y = 20 <CR> Anwenderbefehl: High-Wort der 32-Bit-Adresse ändern
DW0 <CR> Speicherinhalt wortweise anzeigen
0020:0000 0200 Bildschirmausgabe

Gleichzeitige Ausgabe von 3 Werten

Mit dem ZG-Befehl ist es möglich, die Werte von maximal 3 Adressen anzuzeigen. Immer, wenn sich der Wert der 1. Adresse ändert, werden die Werte auf dem Bildschirm aktualisiert. Der Ausdruck "expr" gibt an, wie oft die Aktualisierung der Werte unterdrückt werden soll.

Befehl:



- adr1: 1. Adresse, deren Wert auf dem Bildschirm angezeigt wird. Ändert sich der Wert von adr1, werden die Werte auf dem Bildschirm aktualisiert.
- adr2: 2. Adresse, deren Wert auf dem Bildschirm angezeigt wird.
- adr3: 3. Adresse, deren Wert auf dem Bildschirm angezeigt wird.
- expr: Anzahl, wie oft bei einer Wertänderung von adr1 die Aktualisierung der Werte auf dem Bildschirm unterdrückt werden soll.
- ,: Schlüsselwort (Trennzeichen)
- =: Schlüsselwort

Beispiel:

ZG 1000:0, 1000:100 <CR>

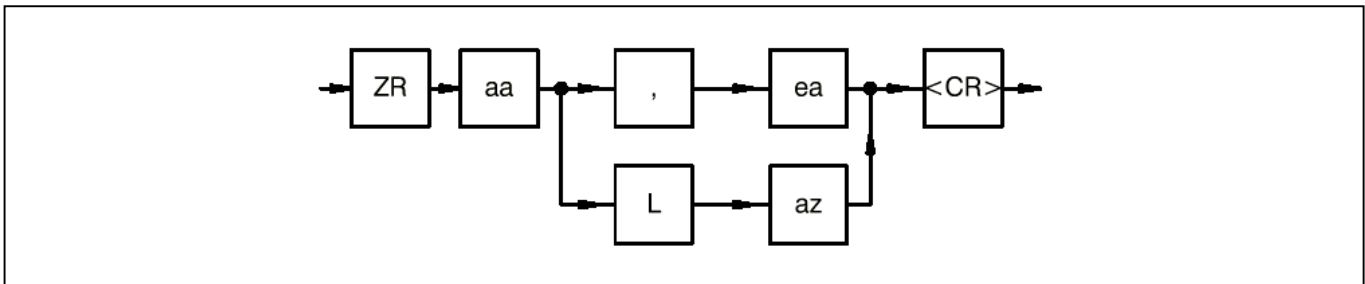
Die Werte der Adressen 1000:0_H und 1000:100_H werden auf dem Bildschirm angezeigt. Ändert sich der Wert von Adresse 1000:0_H, werden die Werte der beiden Adressen auf dem Bildschirm aktualisiert.

RAM-Test

Der angegebene Bereich wird mit einem Testmuster beschrieben (FFFF, 5555, AAAA), danach wird geprüft, ob die Testwerte fehlerlos im angegebenen Bereich stehen. Wird ein Fehler festgestellt, werden Adresse, Istwert und Sollwert ausgegeben. Mit einem beliebigen Tastendruck (außer <SPACE>) kann der Test fortgesetzt werden. Mit CTRL C wird der Test abgebrochen.

Es werden 3 Testzyklen mit in der Reihenfolge vertauschten Testwerten durchgeführt. Der 4. Testzyklus besteht darin, auf die Anfangsadresse einen Zähler abzuspeichern, auf fehlerloses Abspeichern zu prüfen und den Test mit dem dekrementierten Zähler zu wiederholen bis dieser Null erreicht. Danach wird der RAM-Test mit der Monitormeldung (*) beendet.

Befehl:



- aa: Anfangsadresse des RAM-Bereichs
- ea: Endadresse des RAM-Bereichs
- L: Länge (Schlüsselwort)
- az: Anzahl der Bytes des RAM-Bereichs
- ,: Schlüsselwort (Trennzeichen)

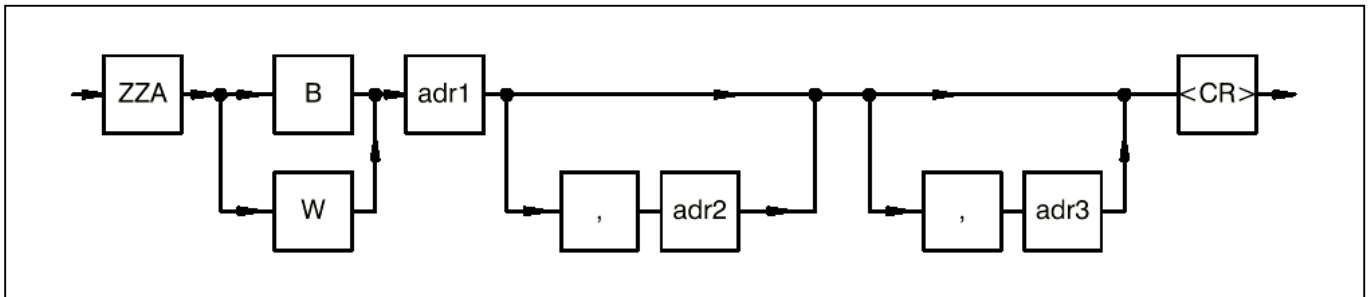
Beispiel:

ZR 1000:0L100<CR> RAM-Test über den angegebenen Speicherbereich.

Ausgabe von 3 Werten nach Eingabe eines Semikolons (;)

Mit dem ZZA-Befehl ist es möglich, nach jeder Eingabe eines Semikolons (;) die Werte (Byte oder Wort) von maximal 3 Adressen anzuzeigen. Der Befehl kann mit <CR> abgebrochen werden.

Befehl:



B: Byteweise (Schlüsselwort)

W: Wortweise (Schlüsselwort)

adr1: 1. Adresse, deren Wert auf dem Bildschirm angezeigt wird.

adr2: 2. Adresse, deren Wert auf dem Bildschirm angezeigt wird.

adr3: 3. Adresse, deren Wert auf dem Bildschirm angezeigt wird.

,: Schlüsselwort (Trennzeichen)

Beispiel:

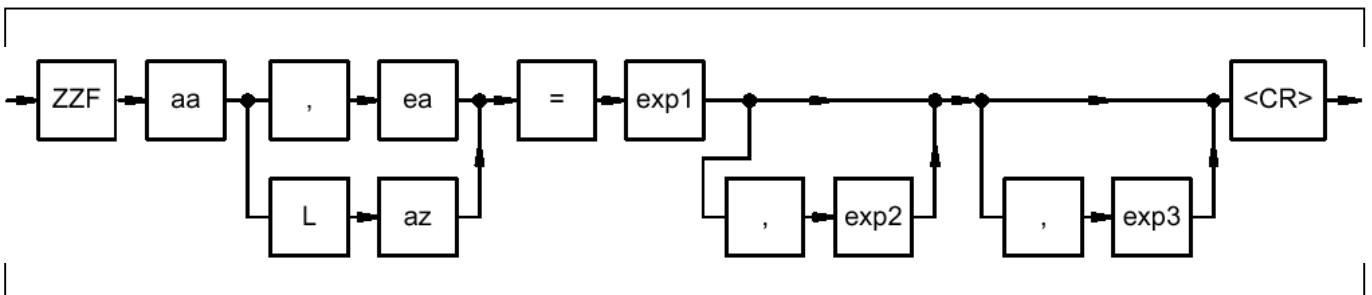
ZZA 1000:0, 1000:100 <CR>

Nach Eingabe eines Semikolons (;) werden die Werte der Adressen 1000:0_H und 1000:100_H auf dem Bildschirm angezeigt.

String suchen

Mit dem Befehl ZZF kann ein String mit maximal 3 Worten im angegebenen Speicherbereich gesucht werden. Wird der String gefunden, wird die Adresse auf dem Bildschirm ausgegeben. Durch Eingabe eines Semikolons (;) wird die Suche fortgesetzt. Wird der String nicht gefunden, wird die Monitormeldung <#07> ausgegeben.

Befehl:



aa: Anfangsadresse des Speicherbereichs

ea: Endadresse des Speicherbereichs

L: Länge (Schlüsselwort)

az: Anzahl der Worte des Speicherbereichs

exp1: 1. Wort des Strings

exp2: 2. Wort des Strings

exp3: 3. Wort des Strings

,: Schlüsselwort (Trennzeichen)

Beispiel:

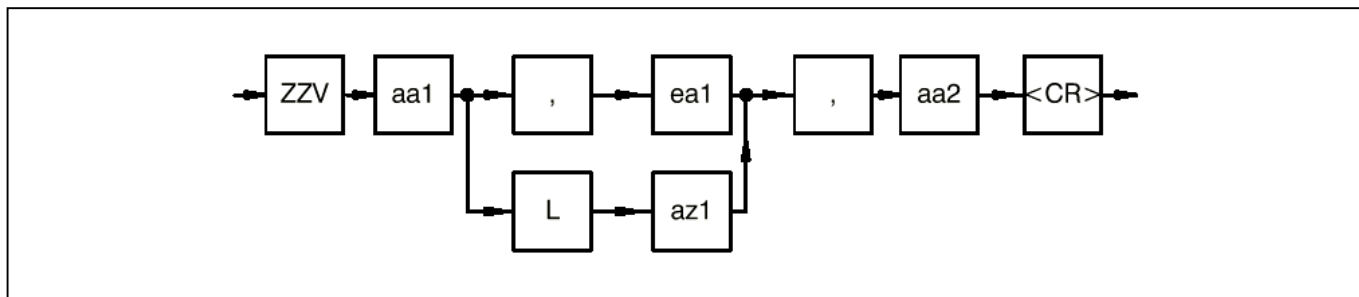
ZZF 1000:0, 100 = AAAA, BBBB <CR>

Der eingegebene String (AAAA_H, BBBB_H) wird im Bereich von 1000:0_H bis 1000:100_H gesucht.

Speicherbereiche wortweise vergleichen

Mit dem Befehl ZZV wird ein Speicherbereich 1 wortweise mit einem Speicherbereich 2 verglichen. Wird eine Abweichung festgestellt, werden die Adresse 1, der Inhalt 1, die Adresse 2 und der Inhalt 2 auf dem Bildschirm ausgegeben. Der Vorgang kann mit CTRL C abgebrochen werden.

Befehl:



aa1: Anfangsadresse des Speicherbereichs 1

ea1: Endadresse des Speicherbereichs 1

L: Länge (Schlüsselwort)

az1: Anzahl der Worte des Speicherbereichs 1

aa2: Anfangsadresse des Speicherbereichs 2

,: Schlüsselwort (Trennzeichen)

Beispiel:

ZZVA000:0 L 100, 8000:0 <CR>

Der Speicherbereich 1 zwischen A000:0_H und A000:100_H wird mit dem Speicherbereich 2 ab 8000:0_H verglichen.

INTEL-HEX-File lesen

Mit dem R-Befehl ist es möglich, einen INTEL-HEX-File über die serielle Schnittstelle COM2 des Zentraleinheiten 07 KT 92/93/94 einzulesen und die HEX-File-Daten in der SPS abzulegen.

Dabei werden folgende Records akzeptiert:

- Adreßerweiterungs-Record
- Daten-Record
- End-Record

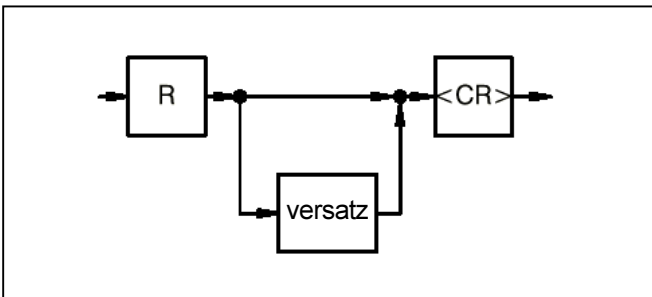
Es gilt dabei folgendes Übertragungsformat:

- 8 Datenbits
- kein Paritybit
- 1 Stoppsbit

Die Daten des INTEL-HEX-Files werden in der SPS ab folgender Adresse abgelegt:

- Die Segmentadresse wird durch die Adresse im Adreßerweiterungs-Record des INTEL-HEX-Files bestimmt. Wird bei der Eingabe des Befehls ein Versatz (versatz) angegeben, so wird dieser Versatz zur Segmentadresse im Adreßerweiterungs-Record addiert. Es entsteht somit eine neue Segmentadresse, ab der die Daten des HEX-Files abgelegt werden. Damit kann der Ablagebereich der HEX-File-Daten in der SPS vorgegeben werden.
- Die Offset-Adresse wird durch die Adresse im Daten-Record des INTEL-HEX-Files bestimmt.

Befehl:



versatz: Versatz (ergibt durch Addition mit der Segmentadresse des Adreßerweiterungs-Records die neue Segmentadresse)

Beispiel:

- R <CR> die SPS ist bereit, ein INTEL-HEX-File zu empfangen
- R 2F00 <CR> die SPS ist bereit, ein INTEL-HEX-File zu empfangen. Zur Segmentadresse des Adreßerweiterungs-Records wird der HEX-Wert 2F00_H addiert. Die dadurch entstehende neue Segmentadresse ist für die Ablage der HEX-File-Daten maßgebend.

INTEL-HEX-File schreiben

Mit dem W-Befehl ist es möglich, einen Datenbereich der SPS als INTEL-HEX-File über die serielle Schnittstelle COM2 des Zentraleinheiten 07 KT 92/93/94 auszugeben.

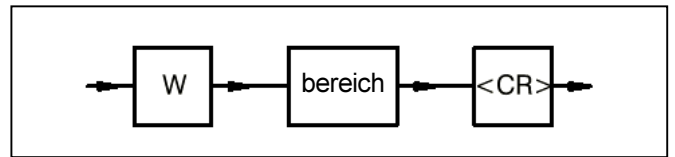
Dabei werden folgende Records erzeugt:

- Adreßerweiterungs-Record
- Daten-Record
- End-Record

Es gilt dabei folgendes Übertragungsformat:

- 8 Datenbits
- kein Paritybit
- 1 Stoppsbit

Befehl:



bereich: Speicherbereich, der als Intel-Hex-File ausgegeben werden soll

Beispiel:

W 8000:0,FFFF <CR> Der Speicherbereich von 8000:0_H bis einschließlich 8000:FFFF_H wird als INTEL-HEX-File über die serielle Schnittstelle COM2 der SPS ausgegeben.

W 8000:0LFFFF <CR> Der Speicherbereich von 8000:0_H bis einschließlich 8000:FFFE_H wird als INTEL-HEX-File über die serielle Schnittstelle COM2 der SPS ausgegeben.

4 Speicherübersichten

4.1 Speicherübersichten für 07 KT 94

Achtung:

Bei Bausteinen, die mit Segmenten arbeiten (COPY, WAES, DWAES, WOL, DWOL, WOS, DWOS), muß bei den Segmenten der Wert des Descriptors angegeben werden.

Beispiel:

Bei der 07 KT 94 soll der Merker MW 0,00 nach MW 1,00 kopiert werden:

COPY Baustein:

Wert am Eingang: FREI = 1_D
 ANZ = 1_D
 QOFF = 8A80_H (35456_D)
 QSEG = B8_H (184_D)
 ZOFF = 8AA0_H (35488_D)
 ZSEG = B8_H (184_D)

4.1.1 Anwenderprogramm-RAM

Begriffserläuterung:

- ARCNET-Zwischenspeicher: Descriptor 176_D (B0_H)
- Organisationsverzeichnis: Descriptor 168_D (A8_H)
 - SPS-spezifisch: Darin sind Organisationsdaten abgelegt, die die ganze SPS betreffen.
 - für AWP 1: Darin sind Organisationsdaten abgelegt, die den Programmspeicher 1 betreffen.
 - für AWP 2: Darin sind Organisationsdaten abgelegt, die den Programmspeicher 2 betreffen.
- Programm-Identifikation: 16 Bytes für eine Identifikationskennung, z. B. Projektname
- Anwenderprogrammspeicher 1 bzw. 2: Speicher für das SPS-Programm
 - Anwenderprogrammspeicher 1: Descriptor 328_D bis 384_D (148_H bis 180_H)
 - Anwenderprogrammspeicher 2: Descriptor 600_D bis 656_D (258_H bis 290_H)
- Turbo-RAM-Programm 1 bzw. 2: Maschinencode für Anwenderprogrammspeicher 1 bzw. 2
- Konstanten für Programm 1 bzw. 2: In diesem Bereich liegen die indirekten Konstanten des Anwenderprogrammspeichers 1 bzw. 2, Konstanten für Programm 1: Descriptor 448_D (1C0_H)
 Konstanten für Programm 2: Descriptor 464_D (1D0_H)
- Operanden-Speicher: Descriptor 184_D (B8_H)

Turbo-RAM Programm 2 30000 _H Byte Descriptor 768 _D bis 784 _D (300 _H bis 310 _H)	002FFFFF
Reserved	002D0000
Reserved	002CFFFF 002C0000
CS31-EA-Copy 800 _H Byte	002BF7FF
Reserved	002BF000
Reserved	002BEFFF 002B0000
Turbo-RAM Programm 2 40000 _H Byte Descriptor 736 _D bis 760 _D (2E0 _H bis 2F8 _H)	002AFFFF
Turbo-RAM Programm 1 70000 _H Byte Descriptor 392 _D bis 440 _D (188 _H bis 1B8 _H)	00270000
Operandenspeicher	0026FFFF
Konstanten für Programm 2 1142 _H Byte	00200000
Anwenderprogramm-Speicher 2 3EE40 _H Byte	001FFFFF 001F0000
Konstanten für Programm 1 1142 _H Byte	001C0000
Anwenderprogramm-Speicher 1 3EE40 _H Byte	001BEE3F
Programm-Identifikation	00180000
Organisationsverzeichnis für Programm 2	00170000
Organisationsverzeichnis für Programm 1	0016EE3F
Organisationsverzeichnis SPS-spezifisch	00130000
Steuerblock 0...2	0012FF7F 0012FF70
ARCNET-Zwischenspeicher 6220 _H Byte	0012FF5A
	0012FF44
	0012FF30
	0012FF00
	0011621F
	00110000
Anwenderprogramm-RAM	

**4.1.2 Anwenderprogramm
Flash-EPROM**

Anwenderdaten Segment 3 7860 _H Byte	02470000
Anwenderdaten Segment 2 7860 _H Byte	02460000
Anwenderdaten Segment 1 7860 _H Byte	02450000
Anwenderdaten Segment 0 7860 _H Byte	02440000
Checksum 0 bis 3	0243FFF8
frei 6 _H Byte	0243FFF2
Anwenderprogramm Teil 3 EE40 _H Byte	024311B2
Konstanten 1142 _H Byte	02430070
frei 20 _H Byte	02430050
Programm-Identifikation	02430040
Organisationsverzeichnis für Programm 2	0243002A
Organisationsverzeichnis für Programm 1	02430014
Organisationsverzeichnis SPS-spezifisch	02430000
Anwenderprogramm Teil 2 10000 _H Byte	02420000
Anwenderprogramm Teil 1 10000 _H Byte	02410000
Anwenderprogramm Teil 0 10000 _H Byte	02400000

- Anwenderdaten Segment 0 bis Segment 3:
In diesem Bereich werden die Anwenderdaten Segment 0 bis Segment 3 abgelegt. Die Anwenderdaten sind mit einer Prüfsumme gesichert.

4.1.3 Operandenspeicher 07 KT 94

1FF200		SEG:F200
1FF100	Stack 2 100 _H	SEG:F100
1FF000	ASAS 100 _H	SEG:F000
1FEA80	580 _H	SEG:EA80
1FDA80	S 1000 _H	SEG:DA80
1FCA80	MD 1000 _H	SEG:CA80
1F8A80	MW 4000 _H	SEG:8A80
1F6A80	M 2000 _H	SEG:6A80
1F6420	AW 660 _H	SEG:6420
1F55E0	A E40 _H	SEG:55E0
1F4F80	EW 660 _H	SEG:4F80
1F4140	E E40 _H	SEG:4140
1F3B40	KD 600 _H	SEG:3B40
1F3000	KW B40 _H	SEG:3000
1F2FFE	K 2 _H	SEG:2FFE
1F2EF0	Free Pool 10E _H	SEG:2EF0
1F2200	CF0 _H	SEG:2200
1F0200	VWS 2000 _H	SEG:0200
1F0100	Stack 1 100 _H	SEG:0100
1F0000	ASAS 1 100 _H	SEG:0000

Begriffserläuterung:

ASAS 1: Arbeitsspeicher Programm 1

Stack1: Stack für Programm 1

VWS: Vergangenheitswert-Speicher

K: Indirekte Konstanten BINÄR
K 00,00...K 00,01

KW: Indirekte Konstanten WORT
KW 00,00...KW 89,15

KD: Indirekte Konstanten DOPPELWORT
KD 00,00...KD 23,15

E: Prozeßabbild der Eingänge BINÄR
E 00,00...E 99,15
E 100...E 163,15
E 200...E 263,15

EW: Prozeßabbild der Eingänge WORT
EW 00,00...EW 34,15
EW 100,00...EW 107,15
EW 200,00...EW 207,15

A: Prozeßabbild der Ausgänge BINÄR
A 00,00...A 99,15
A 100,00...A 163,15
A 200,00...A 263,15

AW: Prozeßabbild der Ausgänge WORT
AW 00,00...AW 34,15
AW 100,00...AW 107,15
AW 200,00...AW 207,15

M: Merker BINÄR
M 00,00...M 511,15

MW: Merker WORT
MW 00,00...MW 511,15

MD: Merker DOPPELWORT
MD 00,00...MD 63,15

S: Schrittketten
S 00,00...S 255,15

ASAS2: Arbeitsspeicher für Programm 2

Stack 2: Stack für Programm 2

Achtung:

Bei Bausteinen, die mit Segmenten arbeiten (COPY, WAES, DWAES, WOL, DWOL, WOS, DWOS), muß bei den Segmenten der Wert des Descriptors angegeben werden.

Beispiel siehe unter 4.1

SEG=001F_H
(High Word der 32-Bit-Adresse,
Descriptor 0B8_H = 184_{DEZ})

4.1.3 Dual-Port-RAM:
CS31

CS31: AW 08,00...AW 15,15	SEG:05FF
CS31: EW 08,00...EW 15,15	SEG:0500 SEG:04FF
CS31-Status (EW 07,15)	SEG:0400 SEG:03FF
Read real time clock (EW 07,08...EW 07,14)	SEG:03FE SEG:03FD
Spontaneous mail box (EW 07,04...EW 07,07)	SEG:03F0 SEG:03EF
Receive mail box EW 07,00...EW 07,03	SEG:03E8 SEG:03E7
CS31: EW 00,00...EW 05,15	SEG:03E0 SEG:03DF
reserved	SEG:0300
Send mail box	SEG:02FF
reserved	SEG:02FE
CS31: AW 00,00...AW 05,15	SEG:02F4 SEG:02F3 SEG:02E0 SEG:02DF
CS31: E 00,00...E 61,15	SEG:0200 SEG:01FF
reserved	SEG:0180 SEG:017F
CS31: A 00,00...A 61,15	SEG:0080 SEG:007F
	SEG:0000

SEG=0050_H
(High Word der 32-Bit-Adresse,
Descriptor 0C8_H = 200_{DEZ})

4.1.4 Dual-Port-RAM:
Ein- und Ausgänge der Zentraleinheit

Alle nicht belegten Speicherbereiche sind reserviert.	
EW 06,00...EW 06,15	SEG:006F
	SEG:0050
AW 06,00...AW 06,15	SEG:00AF
E 62,00...E 63,15	SEG:0090 SEG:0033
E 64,00...E 64,07	SEG:0030
E 64,00...E 64,07	SEG:0034
A 62,00...A 63,15	SEG:0042
	SEG:0040

SEG=0060_H
(High Word der 32-Bit-Adresse,
Descriptor 058_H = 88_{DEZ})

4.2 Speicherübersichten für 07 KR 91

4.2.1 Anwenderprogramm-RAM

frei	38C20 38C12
Konstanten für Programm 2 702 _H Byte	38510
Turbo-RAM Programm 2 12EB0 _H Byte	25660
Anwenderprogramm-Speicher 2 7800 _H Byte	1DE60
frei	1DE52
Konstanten für Programm 1 702 _H Byte	1D750
Turbo-RAM Programm 1 12EB0 _H Byte	0A8A0
Anwenderprogramm-Speicher 1 7800 _H Byte	030A0
frei	03080
Programm-Identifikation	03070
Organisationsverzeichnis für Programm 2	0305A
Organisationsverzeichnis für Programm 1	03044
Organisationsverzeichnis SPS-spezifisch	03030
Steuerblock 0...2	03000

Begriffserläuterung:

- Organisationsverzeichnis
 - SPS-spezifisch: Darin sind Organisationsdaten abgelegt, die die ganze SPS betreffen.
 - für AWP 1: Darin sind Organisationsdaten abgelegt, die den Programmspeicher 1 betreffen.
 - für AWP 2: Darin sind Organisationsdaten abgelegt, die den Programmspeicher 2 betreffen.
- Programm-Identifikation:
16 Bytes für eine Identifikationskennung, z. B. Projektname

4.2.2 Anwenderprogramm Flash-EPROM

Checksum	A7FFE
frei 8C _H Byte	A7F72
Anwenderprogramm 7800 _H Byte	A0772
Konstanten 702 _H Byte	A0070
frei 20 _H Byte	A0050
Programm-Identifikation	A0040
Organisationsverzeichnis für Programm 2	A002A
Organisationsverzeichnis für Programm 1	A0014
Organisationsverzeichnis SPS-spezifisch	A0000

- Anwenderprogrammspeicher 1:
Speicher für das SPS-Programm
- Turbo-RAM Programm 1:
Maschinencode für
Anwenderprogrammspeicher 1
- Konstanten für Programm 1:
In diesem Bereich liegen die indirekten Konstanten des Anwenderprogrammspeichers 1.
- Anwenderprogrammspeicher 2:
Speicher für das SPS-Programm
- Turbo-RAM Programm 2:
Maschinencode für
Anwenderprogrammspeicher 2
- Konstanten für Programm 2:
In diesem Bereich liegen die indirekten Konstanten des Anwenderprogrammspeichers 2.

4.2.3 Operandenspeicher

40000			SEG:F3E0
3FFF0	frei	10 _H	SEG:F3D0
3FE60	E/A-Konfigurierliste 2	190 _H	SEG:F240
3FCD0	E/A-Konfigurierliste 1	190 _H	SEG:F0B0
3FCC8	frei	8 _H	SEG:F0A8
3FA40	E/A-Force-Listen	288 _H	SEG:EE20
3FA30	frei	10 _H	SEG:EE10
3F930	Stack 2	100 _H	SEG:ED10
3F830	ASAS 2	100 _H	SEG:EC10
3E030	VWS	1800 _H	SEG:D410
3D830	S	800 _H	SEG:CC10
3D030	MD	800 _H	SEG:C410
3B030	MW	2000 _H	SEG:A410
3A030	M	1000 _H	SEG:9410
39F30	AW	100 _H	SEG:9310
39B30	A	400 _H	SEG:8F10
39A30	EW	100 _H	SEG:8E10
39630	E	400 _H	SEG:8A10
39430	KD	200 _H	SEG:8810
38F30	KW	500 _H	SEG:8310
38F2E	K	2 _H	SEG:830E
38E20	Free Pool	10E _H	SEG:8200
38D20	Stack 1	100 _H	SEG:8100
38C20	ASAS 1	100 _H	SEG:8000
			SEG=30C2

Begriffserläuterung:

- ASAS 1: Arbeitsspeicher Programm 1
- Stack 1: Stack für Programm 1
- K: Indirekte Konstanten BINÄR
- KW: Indirekte Konstanten WORT
- KD: Indirekte Konstanten DOPPELWORT
- E: Prozeßabbild der Eingänge BINÄR
- EW: Prozeßabbild der Eingänge WORT
- A: Prozeßabbild der Ausgänge BINÄR
- AW: Prozeßabbild der Ausgänge WORT
- M: Merker BINÄR
- MW: Merker WORT

- MD: Merker DOPPELWORT
- S: Schrittketten
- VWS: Vergangenheitswert-Speicher
- ASAS 2: Arbeitsspeicher für Programm 2
- E/A-Force-Listen:

Hier werden die zu forcenden E/A-Signale und ihre Force-Werte eingetragen.

E/A-Konfigurierliste 1:

Hier werden die im Programm 1 projektierten E/A-Signale eingetragen, damit sie bei der Erstellung und Ausgabe des Prozeßabbildes berücksichtigt werden.

E/A-Konfigurierliste 2:

Hier werden die im Programm 2 projektierten E/A-Signale eingetragen, damit sie bei der Erstellung und Ausgabe des Prozeßabbildes berücksichtigt werden.

4.2.4 Dual-Port-RAM

CS31-Status (EW 07,15)	C03FF C03FE C03FD
Read real time clock EW 07,08...EW 07,14	C03F0 C03EF
Spontaneous mail box EW 07,04...EW 07,07	C03E8 C03E7
Receive mail box EW 07,00...EW 07,03	C03E0 C03DF
Direct: EW 06,00...EW 06,15 CS31: EW 00,00...EW 05,15	C0300
reserved	C02FF
Send mail box	C02FE C02F4 C02F3 C03E0 C02DF
reserved	
Direct: AW 06,00...AW 06,15 CS31: AW 00,00...AW 05,15	C0200 C01FF
Direct: E 62,00...E 63,15 CS31: E 00,00...E 61,15	C0180 C017F C0100 C00FF
reserved	
Read/write permission Read/write request	C00FE C00FD
reserved	C0080 C007F
Direct: A 62,00...A 63,15 CS31: A 00,00...A 61,15	C0000

5 Funktionen in der Anweisungsliste für 07 KR 91, 07 KT 92 bis 07 KT 94

5.1 Texte in der Anweisungsliste

Einige SPS-Bausteine (DRUCK, EMAS) arbeiten mit Texten, die im Anwenderprogramm abgelegt sind.

Eingabe der Texte in das Anwenderprogramm

Die Eingabe eines Textes mit einem Terminal erfolgt eingebettet in die Schlüsselzeichen # und ". Das Schlüsselzeichen # kennzeichnet den Anfang und das Schlüsselzeichen " das Ende eines Textstrings.

Eingegeben werden können alle ASCII-Zeichen zwischen 0_H und 7_FH.

Ablage der Texte im Anwenderprogramm

Jedes eingegebene Textzeichen belegt im Anwenderprogramm ein Wort. Dabei wird im Low-Byte die ASCII-Kennung des Textzeichens und im High-Byte das Präfix FA abgelegt.

Beispiel:

Texteingabe und Ablage ab Adresse 100 im SPS-Proramm:

Eingabe: S 100 <CR>
00100 NOP
#"ABB"#<CR>

Ablage: 00100 FA41
00101 FA42
00102 FA42

Übersicht über die möglichen Textzeichen, ihre Eingabe und ihre Ausgabe auf dem Bildschirm

ASCII-Zeichen	Hex-Wert	Anwendereingabe	Ausgabe Ausgabe	ASCII-Zeichen	Hex-Wert	Anwendereingabe	Ausgabe	ASCII-Zeichen	Hex-Wert	Anwendereingabe	Ausgabe
NUL	00	<CTRL><SP>	<NUL>*	0	30	0	0	`	60	`	`
SOH	01	<CTRL>A	<SOH>	1	31	1	1	a	61	a	a
STX	02	<CTRL>B	<STX>	2	32	2	2	b	62	b	b
ETX	03	<CTRL>C	<ETX>	3	33	3	3	c	63	c	c
EOT	04	<CTRL>D	<EOT>	4	34	4	4	d	64	d	d
ENQ	05	<CTRL>E	<ENQ>	5	35	5	5	e	65	e	e
ACK	06	<CTRL>F	<ACK>	6	36	6	6	f	66	f	f
BEL	07	<CTRL>G	<BEL>	7	37	7	7	g	67	g	g
BS	08	<CTRL>H	<BS>	8	38	8	8	h	68	h	h
HT	09	<CTRL>I	<HT>	9	39	9	9	i	69	i	i
LF	0A	<CTRL>J	<LF>	:	3A	.	.	j	6A	j	j
VT	0B	<CTRL>K	<VT>	;	3B	;	:	k	6B	k	k
FF	0C	<CTRL>L	<FF>	<	3C	<	<	l	6C	l	l
CR	0D	<CTRL>M	<CR>	=	3D	=	=	m	6D	m	m
SO	0E	<CTRL>N	<SO>	>	3E	>	>	n	6E	n	n
SI	0F	<CTRL>O	<SI>	?	3F	?	?	o	6F	o	o
DLE	10	<CTRL>P	<DLE>	@	40	@	@	p	70	p	p
DC1	11	<CTRL>Q	<DC1>	A	41	A	A	q	71	q	q
DC2	12	<CTRL>R	<DC2>	B	42	B	B	r	72	r	r
DC3	13	<CTRL>S	<DC3>	C	43	C	C	s	73	s	s
DC4	14	<CTRL>T	<DC4>	D	44	D	D	t	74	t	t
NAK	15	<CTRL>U	<NAK>	E	45	E	E	u	75	u	u
SYN	16	<CTRL>V	<SYN>	F	46	F	F	v	76	v	v
ETB	17	<CTRL>W	<ETB>	G	47	G	G	w	77	w	w
CAN	18	<CTRL>X	<CAN>	H	48	H	H	x	78	x	x
EM	19	<CTRL>Y		I	49	I	I	y	79	y	y
SUB	1A	<CTRL>Z	<SUB>	J	4A	J	J	z	7A	z	z
ESC	1B	<CTRL>[<ESC>	K	4B	K	K	{	7B	{	{
FS	1C	<CTRL>\	<FS>	L	4C	L	L		7C		
GS	1D	<CTRL>]	<GS>	M	4D	M	M	}	7D	}	}
RS	1E	<CTRL>~	<RS> *	N	4E	N	N	~	7E	~	~
US	1F	<CTRL>?	<US> *	O	4F	O	O	DEL	7F		
SP	20	<SP>	<SP> **	P	50	P	P				
!	21	!	!	Q	51	Q	Q				
"	22	"	"	R	52	R	R				
#	23	#	#	S	53	R	R				
\$	24	\$	\$	T	54	T	T				
%	25	%	%	U	55	U	U				
&	26	&	&	V	56	V	V				
'	27	'	'	W	57	W	W				
(28	((X	58	W	W				
)	29))	Y	59	Y	Y				
*	2A	*	*	Z	5A	Z	Z				
+	2B	+	+	[5B	[[
,	2C	,	,	\	5C	\	\				
-	2D	-	-]	5D]]				
.	2E	.	.	^	5E	^	^				
/	2F	/	/	_	5F	_	_				

* Bei älteren Terminals galt:

NUL 00 <CTRL>@ <NUL>
RS 1E <CTRL>^ <RS>
US 1F <CTRL>_ <US>

** Bei der *Texteingabe* wird ein SPACE als Leerzeichen angezeigt, bei der *Ausgabe* des Anwenderprogramms zur besseren Erkennung als <SP>.

Anmerkung:

Zusammenhang zwischen Wert und ASCII-Zeichen:

Dualwert im Rechner:

1011 (0B_H, 11_{DEC})

Darstellung als:

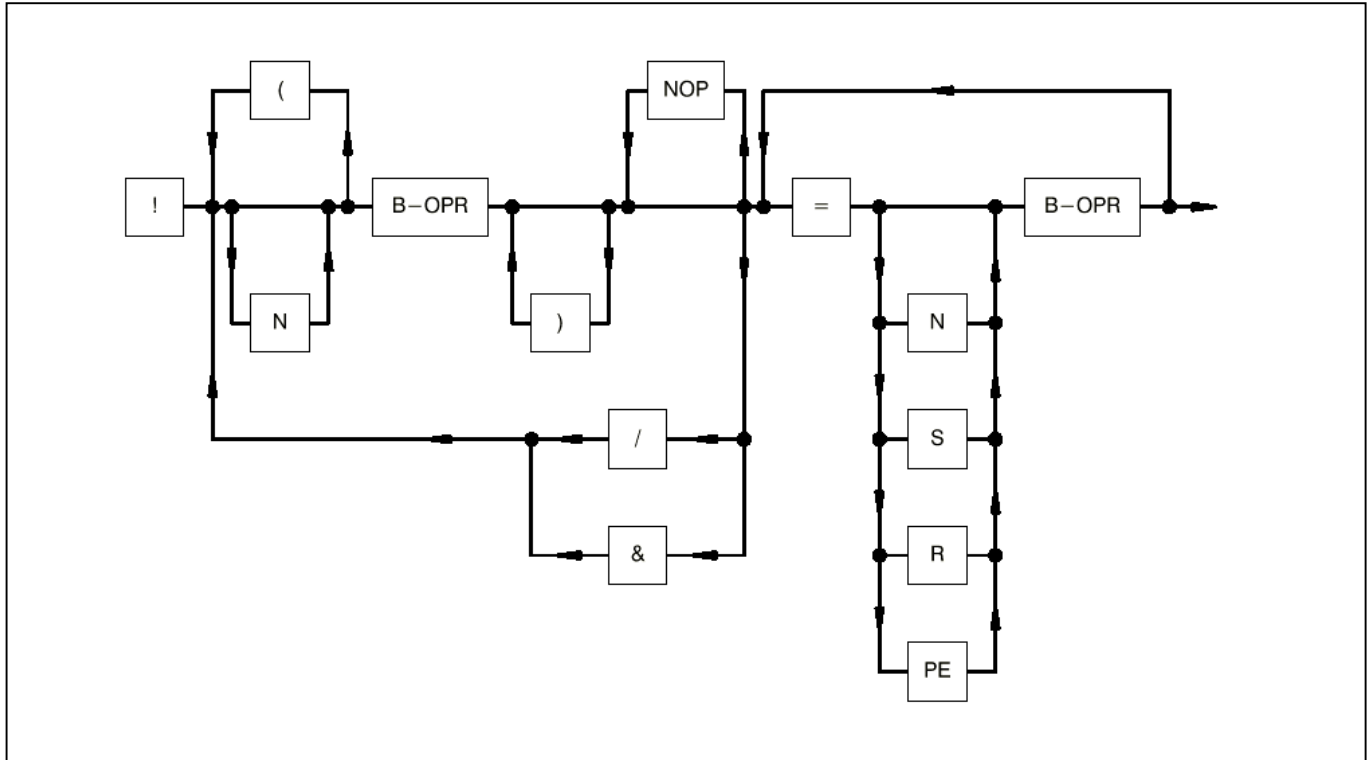
- Dezimal ASCII: 31_H, 31_H

- Hexadezimal ASCII: 42_H

Ausgabe durch DRUCK
als Hexwert: 1011

5.2 Syntaxdiagramme für Anweisungsliste (AWL)

5.2.1 Syntaxdiagramm: BOOL'SCHER SATZ



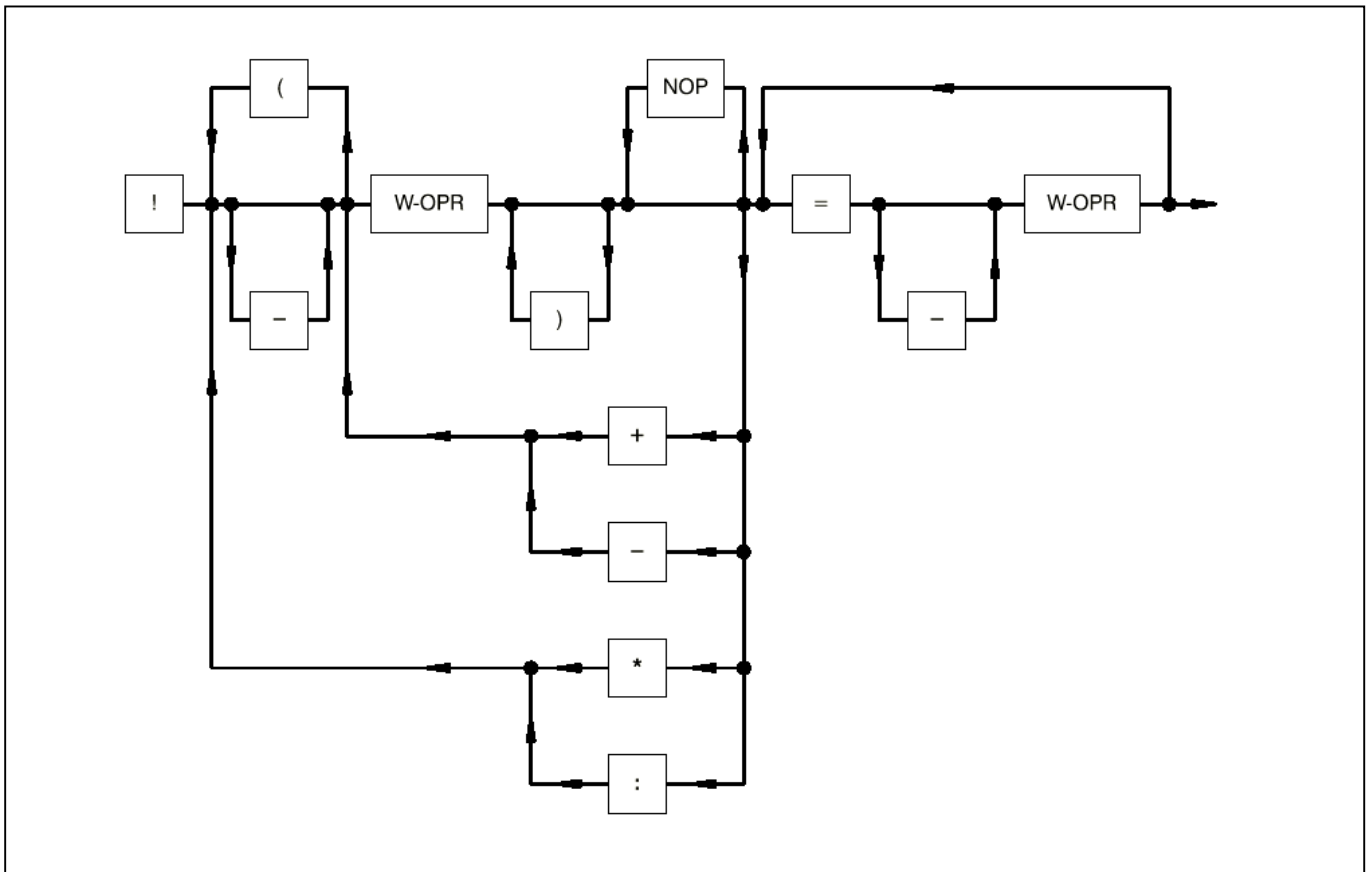
Signalfluß: In Pfeilrichtung, sonst von links nach rechts

Klammern: Summe "KLAMMER AUF" = Summe "KLAMMER ZU", Klammertiefe: 15

B-OPR: Binär-Operand (E, A, M, S, K)

Beispiele: E 00,03 A 07,15 M 05,01 S 05,04 K 00,01

5.2.2 Syntaxdiagramm: ARITHMETIK-SATZ



Signalfluß: In Pfeilrichtung, sonst von links nach rechts

Klammern: Summe "KLAMMER AUF" = Summe "KLAMMER ZU", Klammertiefe: 15

W-OPR: Wort-Operand (EW, AW, MW, KW)

Beispiele: EW 03,05 AW 11,12 MW 22,15 KW 09,06



Druck auf chlorfrei gebleichtem Papier

ABB Schalt- und Steuerungstechnik GmbH
Eppelheimer Straße 82 Postfach 10 50 09
D-69123 Heidelberg D-69040 Heidelberg

Telefon (06221) 777-0
Telefax (06221) 777-111
E-Mail desst.helpline@de.abb.com
Internet <http://www.abb-sst.de>

Printed in the Federal Republic of Germany (09.99)