TZAKON

Konfigurierprogramm Digitaler Messrechner TZA 4

Bedienungsanleitung	42/18-41 DE	Rev. 02



Inhaltsverzeichnis

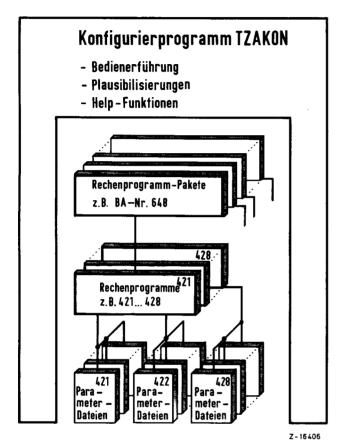
2

	Se	eite			
1	Konfigurierprogramm TZAKON	3	4	Bedieneroberfläche/User-Programme	13
	Allgemeine Beschreibung	3	4.1	Allgemeine Beschreibung	13
	Installation des TZAKON auf einem PC	4	4.2	Programm auf PC erstellen oder ändern	13
	Anforderungen an den PC	4	4.2.1	Neues Programm erstellen	13
	Installation	4	4.3	On-Line-Editor	13
1.2.2	mstallation	7	4.3.1	Terminal/On-Line-Editor	13
1.3	Bedienungsanleitung TZAKON	5	4.3.1	reminar/On-Line-Editor	13
	TZAKON-Funktionstasten	5	_	D. C Pale de DAOIO FO Defelle	4.4
	1. 7		5	Referenzliste der BASIC 52-Befehle	14
	Informationsdaten TZA 4/Initialisierung	_	5.1	Direkt-Modus-Befehle	14
	des TZAKON	6	5.2	RUN und Direkt-Modus-Befehle	14
-	Initialisierung des TZAKON	6	5.3	Arithmetische BASIC 52-Operatoren	22
	TZA 4-Identifikationsdaten auslesen	6	5.4	Logische Operatoren	22
	Freier Programmspeicher im TZA 4	6	5.5	String-Funktionen	22
	Blank-Check Programmspeicher		5.6	Arithmetische Funktionen	22
	im TZA 4	6	5.7	Logarithmische Funktionen	22
1.5.5	Wahl der seriellen COM-Schnittstelle		5.8	Trigonometrische Funktionen	22
	des PC	6	5.9	Spezial-Funktionen	23
1.5.6	Einstellung des Druckers	6	5.10	Basic-Befehle, die nur alternativ alternativ	
	TZA 4-Datum/Zeit auf PC-Datum/Zeit			zu verwenden sind	23
	setzen	6	5.11	Kurzreferenzliste aller BASIC-Befehle	24
	TZAKON-Hilfetext drucken	6			_ :
			6	Konfigurier- und Programmier-Hinweise	
2	Bedienoberfläche/P-Programme	7	•	für TZA 4	26
	Allgemeine Beschreibung	7	6.1	Meß-Funktionen und Signal-Bereiche der	
	Programm/Datei aus PC laden oder		•	Ein- und Ausgänge	26
	parametrieren	7	6.1.1	Analog-Eingänge E1 E6	26
2.2.1	Programm/Datei nach Aufgabengebiet	•	6.1.2	Binär-Eingänge EB1 EB4 bzw. ENI	27
	wählen	7	6.1.3	Analog-Ausgänge	28
	Parameter-Datei-Typ wählen	7	6.2	Skalierung der analogen Ein- und Ausgänge	28
	Datei/Programm nach Auftrags-, Datei-	•	6.3	BASIC-Print-Flag setzen	28
2.2.0	oder Fertigungs-Nr. wählen	7	6.4	Binär-Erweiterungskarte	28
2.2.4	Dateimenü	8	0.4	billar-Li weiterungskarte	20
	Parameter-Datei bearbeiten	O	7	BASIC 52-Programmbeispiele	30
2.2.5	(parametrieren)	8	•	Beispiel 1: Mittelwert-Berechnung	30
2.2.6	Programm/Datei zum TZA 4 transferieren	9			30
	Programm/Datei im TZA 4 starten	9		Beispiel 2: Messen und Skalieren eines	20
		9		physikalischen Meßwertes	30
2.2.8	Programm/Datei im EPROM 2 des TZA 4	^		Beispiel 3: #-Programm erstellen	32
	sichern	9			
2.2.9	Programm/Datei nach Reset automatisch	^	Anhar	-	
	starten	9		-Betriebssystem	
2.3	Datei aus TZA 4 lesen, anzeigen und			herorganisation	
	ändern	10		ory-Map des TZA 4	34
2.3.1	Programm/Datei aus TZA 4-Verzeichnis			52-Interpreter	35
	wählen	10		nlußbelegung der 19″-Steckkarte	
_				-Stecker	36
3	Bedienoberfläche/#-Programme		Gewä	hrleistung und Haftung	37
3.1	Programm aus PC in TZA 4 laden				
3.1.1	Programm wählen				
3.1.2	Startmenü				
3.1.3	Programm-Editor				
3.1.4	Programm abspeichern	11			
3.1.5	Programm unter neuem Namen		Eroön	zondo Druckechrifton:	
	abspeichern		Ligan	zende Druckschriften:	
3.1.6	Programm von Diskette oder Platte lesen	11	Listen	blatt 10/18-5.10	
3.1.7	Programm zum TZA 4 transferieren	12	Gebra	auchsanweisung 42/18-40	
3.1.8	Programm im TZA 4 starten	12			
3.1.9	Programm aus TZA 4 auslesen	12			
3.1.10	Programm vom TZA 4-RAM in das EPROM 2				
	sichern	12			
3.1.11	Programm nach Reset automatisch				
	starten	12			
3.1.12	Programm ausdrucken	12	Technis	che Änderungen vorbehalten.	
3.2	Programm aus TZA 4 starten oder in			uck, Vervielfältigung, Übersetzung, auch auszugsweise, sind	ohne
	Autostart-Liste eintragen	12		nigung nicht gestattet.	

1 Konfigurierprogramm TZAKON

1.1 Allgemeine Beschreibung

Das Konfigurier-Programm TZAKON dient der Konfigurierung, Parametrierung und Programmierung des Digitalen Meßrechners TZA 4.



TZAKON ist lauffähig auf einem IBM-kompatiblen PC, AT- oder PS-2-Computer. Es ist menügeführt und gestattet dem Anwender eine einfache Handhabung. Die Kommunikation zwischen PC und TZA 4 erfolgt über die serielle Schnittstelle (COM1 oder COM2 des PC).

Zum Betrieb wird der TZA 4 mit einem BASIC-Programm geladen. Dazu steht eine Vielfalt konfektionierter, parametrierbarer Programme aus verschiedenen Aufgabengebieten zur Auswahl (siehe auch Übersichtsblatt in Gebrauchsanweisung 42/18-40-). Es können aber auch eigene BASIC-Programme mit Hilfe von TZAKON erstellt und in den TZA 4 geladen werden.

Programme und Parameter-Dateien werden beim Laden im RAM-Speicher des TZA 4 abgelegt. Das Programm kann gestartet, ausgetestet und dann in den BASIC-EPROM-Speicher (EPROM 2) kopiert und dort gesichert (eingebrannt) werden.

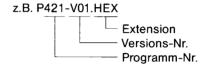
Im EPROM-2-Speicher können mehrere Programme und Parameter-Dateien gesichert werden. Eines dieser Programme wird ausgewählt und in eine AUTO-Start-Liste eingetragen. Diese Liste verweist auf das Programm, das nach Anlegen der Versorgungsspannung oder nach einem Hardware-Reset automatisch startet.

Die Hauptfunktionen des Konfigurierprogrammes TZAKON sind:

- Auswählen eines P-Programms und einer Parameter-Datei aus einer Programm- bzw. Datei-Bibliothek.
- Erstellen bzw. Verändern progammspezifischer Parameter-Dateien.
- User-Programmerstellung
- Laden und Sichern der Programme und Dateien im EPROM 2 des TZA 4
- Auslesen aller im TZA 4 gespeicherten Programme und Parameter-Dateien.
- Informationen über den Status von TZAKON und TZA 4 sowie Initialisierung des TZAKON

TZAKON verwaltet 3 unterschiedliche Programm-Typen:

a) P-Programme



Diese Programme haben folgende Merkmale:

- Es sind konfektionierte Listen-Programme aus verschiedenen Aufgabengebieten (siehe auch Übersichtsblatt in Gebrauchsanweisung 42/18-40-).
- Sie sind im Unterverzeichnis TZAKON/PROGFILE abgespeichert.
- Sie benötigen programmspezifische Parameter-Dateien (im Unterverzeichnis TZAKON/DATFILE abgespeichert).
- Sie sind zu Programm-Paketen zusammengefaßt (max. 9 Programme je Paket).

Zu jedem Paket gehört ein Parametrier-Programm, z.B. P420-000.EXE. Parametrier-Programme sind im Unterverzeichnis TZAKON abgespeichert und dienen der Erstellung und Plausibilisierung von Parameter-Dateien.

b) #-Programme



Diese Programme haben folgende Merkmale:

- Es sind konfektionierte Listen-Programme aus verschiedenen Aufgabengebieten oder kundenspezifische Programme.
- Sie sind im Unterverzeichnis TZAKON/B52DIR abgespeichert
- Sie sind in einer einheitlichen Struktur aufgebaut (siehe Beispiel in Abschnitt 7.3)
- Sie benötigen keine Parameterdatei
- Die Meß- bzw. Signal-Bereiche müssen in das Programm geschrieben werden.
- Die Skalierung erfolgt im Programm.
- Sie haben den Programm-Namen #xxx-yyy.B52 (xxx ähnlich Nr. der P-Programme Pxxx-V00.HEX yyy = Modifikations-Nr.)
- Sie sind unter dem File-Namen #jmmttyy.B52 abgespeichert.

(j = Jahres-Endziffer

mm = Monat

tt = Tag

yy = laufende Nr. des Tages)

Beispiel: 7. Dez. 1992, 5. Auftrag → #2120705

c) USER-Programme

z.B. TESTPROG.B52 Extension File-Namen

Diese Programme haben folgende Merkmale:

- Es sind vom Anwender selbst erstellte BASIC 52-Programme (siehe Abschnitt 4).
- Sie sind im Unterverzeichnis TZAKON/B52DIR abgespeichert.
- Sie haben keine einheitliche Struktur und im Gegensatz zu #-Programmen keine Kopfdaten.
- Sie benötigen keine Parameter-Datei.
- Sie haben einen individuellen File-Namen (max. 8 Zeichen).

1.2 Installation des Konfigurierprogramms TZAKON auf einem PC

1.2.1 Anforderungen an den PC:

- IBM-Kompatibel PC, AT, PS-2

– DOS-Version:

DOS 2.1 aufwärts

- Speichergröße:

≥ 512 KBvte

- Schnittstelle:

RS-232 C

- Diskettenlaufwerk: ≥ 720 KByte

1.2.2 Installation

Das Bedieneroberflächen-Programm TZAKON kann auf 2 verschiedenen Disketten-Arten geliefert werden:

- TZAKON-Diskette Nr. 3100925x, 51/4 Zoll 360 KByte
- TZAKON-Diskette Nr. 3100970x, 31/2 Zoll 720 KByte

Die konfektionierten TZA 4-Programme (P-Programme) werden in Programm-Paketen (nach Aufgabengebieten unterteilt) auf getrennten Disketten geliefert.

2 Disketten-Arten können geliefert werden:

- TZA 4-Diskette Nr. 3100944x...3100967x, 51/4 Zoll 360 KByte
- TZA 4-Diskette Nr. 3100971x...3100994x, 31/2 Zoll 720 KByte

Auf der TZAKON-Diskette ist ein Installierungs-Programm "INSTALL.EXE" abgespeichert. Mit seiner Hilfe wird die gesamte Bediener-Oberfläche TZAKON auf Festplatte oder Diskette installiert.

Vorgehensweise:

- TZAKON-Diskette in Laufwerk A oder B einlegen
- Das Laufwerk mit "A:" oder "B:" anwählen
- INSTALL eingeben und ENTER-Taste drücken

Das INSTALL-Programm blendet folgendes Menü ein:

Anweisungen:

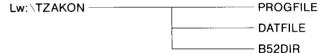
- 1. TZAKON und TZA 4-Programme installieren
- 2. TZAKON ohne TZA 4-Programme installieren
- 3. TZA 4-Programme zusätzlich installieren
- 0. Ende

Eingabe →

Bei Wahl: 1. wird eine Gesamtinstallation von TZAKON einschließlich den auf den TZA 4-Disketten gespeicherten TZA 4-Programmen durchaeführt.

- 2. wird nur TZAKON installiert.
- 3. werden zu einem bereits auf Festplatte oder Diskette installierten TZAKON weitere TZA 4-Programme installiert.

Das INSTALL-Programm legt auf dem Ziel-Laufwerk folgendes Verzeichnis an:



Es werden abgelegt in die Verzeichnisse:

TZAKON:

TZAKON . EXE (Bedieneroberfläche TZAKON)

– INITDAT . DAT

(Initialisierungs-Datei für

- HELPDAT. HLP (Hilfe-Datei für TZAKON)

TZAKON)

- Pxxx-xxx . EXE (Parametrier-Programme für die

Programme)

- Pxxx-xxx . HLP

(Hilfe-Dateien für Parametrier-

Programme)

PROGFILE:

- Pxxx-xxx.HEX (P-Programme)

DATFILE:

xxxxxxxxxx . DAT

(Parameter-Dateien der P-Programme)

B52DIR:

- #xxxxxxx . B52 (#-Programme)

- xxxxxxxxx . B52

(User-Programme)

1.3 Bedienungsanleitung TZAKON:

Nach erfolgreicher Installation schaltet der PC in das Verzeichnis: Lw:\TZAKON. Durch Eingabe von TZAKON und Drücken der ENTER-Taste wird TZAKON aufgerufen. Nach Drücken einer beliebigen Taste wird das START-MENUE angezeigt. Von diesem Menü können alle Hauptfunktionen des TZAKON ausgewählt werden.

Auswahl	Startmenü	
	Aufgabengebiete/ Programme	
	Parameter-Dateien nach Auftrags-Nr.	
	Parameter-Dateien nach Datei-Nr.	
	Parameter-Dateien nach FertigNr.	
	Programme/Dateien aus TZA 4-Verzeichnis	
	Programmerstellung/Terminal-Mode	
	Info TZA 4/Init TZAKON	

<F1> HELP

<ENTER> OK?

<F9> EXIT

Bild 1

Die Tabelle 1 (TZAKON/TZA 4 Handlings-Übersicht) enthält eine Übersicht über alle Funktionen von TZAKON und die dafür benötigten Hilfsmittel. Für jede Funktion werden die Abschnitte angegeben, in der diese Funktionen näher beschrieben sind. Der in den Abschnitten angegebene Menüpfad zeigt die Schritte auf, die ausgeführt werden müssen, um zu der beschriebenen Funktion zu gelangen.

Beispiel: Auswahl eines Programms:

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE → **PROGRAMME**

1.4 TZAKON-Funktionstasten

Zum einfachen Bedienen von TZAKON sind einigen Funktionstasten der PC-Tastatur (F1...F9) spezielle TZAKON-Funktionen zugeordnet. Diese Funktionen werden immer in der Fußzeile des aktuellen Bildes anaezeiat.

Die Bedeutung der Funktionstasten sind:

<F1> HELP:

Hilfetext für das aktuell angewählte Menü-Bild bzw. für das Parametrier-Programm wird ein-

aeblendet.

- PaUp: Hilfetext

vorwärtsblättern

- PgUp: Hilfetext

rückwärtsblättern - ESC: Hilfetext verlassen

<F2> MENUE:

User-Programm-Menü wird ein-

geblendet

<F8> RETURN:

<ENTER> OK?:

Zurück zum vorhergehenden Bild

<F9> FXIT:

Zurück zum Betriebssytem

STARTMENUE: Zurück zum Startmenü DATEIMENUE: Zurück zum Dateimenü

Angewählte Menü-Zeile bzw. der eingegebene Wert wird bestätigt

(weiter).

Hilfs- mittel bzw. Bedien- hinweise	(Pro	Programr ogramme neter-Dat P-Programm parametrieren Datei ausdrucken	mit teien) Parameter- Datei aus	(Prog Parar	Programi gramme i neter-Da	ohne		Usei PC-E	-Program	mme	
mittel bzw. Bedien-	(Pro Paran P-Programm in TZA 4 laden starten	ogramme meter-Dat P-Programm parametrieren Datei	mit teien) Parameter- Datei aus	(Prog Parar	gramme e neter-Da	ohne		PC-E	ditor		
bzw. Bedien-	P-Programm in TZA 4 laden starten	neter-Dat P-Programm parametrieren Datei	teien) Parameter- Datei aus	Paran	neter-Da			PC-E	ditor		
Bedien-	P-Programm in TZA 4 laden starten	P-Programm parametrieren Datei	Parameter- Datei aus	#-Programm		(CICII)					
	in TZA 4 laden starten	parametrieren Datei	Datei aus								
			TZA 4 auslesen drucken	in TZA 4 laden starten sichern (drucken)	#-Programm parametrieren (drucken)	aktuelles #-Programm aus TZA 4 auslesen (drucken)	User- Programm in TZA 4 laden sichern (drucken)	User- Programm erstellen (drucken)	User- Programm ändern (drucken)	User- Programm aus TZA 4 auslesen (drucken)	TZA 4 ON Line-Edito
Hardware:											
TZA 4	ja	nein	ja	ja	nein	ja	ja	nein/(ja)	nein/(ja)	ja	ja
PC	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja
Drucker	nein	(ja)	(ja)	(ja)	(ja)	(ja)	(ja)	(ja)	(ja)	(ja)	nein
Software:											
TZAKON.EXE	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja
P-Programme	ja	ja	ja	nein	nein	nein	nein	nein	nein	nein	nein
Parameter-Dateien	ja	ja	nein	nein	nein	nein	nein	nein	nein	nein	nein
#-Programme	nein	nein	nein	ja	ja	nein	nein	nein	nein	nein	nein
USER-Programme	nein	nein	nein	nein	nein	nein	ja	nein	ja	nein	nein
TZAKON/TZA 4											
Initialisierung	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja
	2.2.1	2.2.1	2.3.1	3.2.1	3.2.1	3.2.2	3.2.2	3.2.2	3.2.2	3.2.2	3.2.2
	2.2.2	2.2.2	2.2.4	oder	oder	3.2.9	3.2.6	4.2.1	3.2.6	3.2.9	4.3.1
	oder	oder	2.2.5	3.2.2	3.2.2	3.2.3	3.2.7	3.2.5	3.2.3	(3.2.12)	ı
Vorgehensweise	2.2.3	2.2.3		3.2.6	3.2.6	(3.2.12)	(3.2.8)	(3.2.7)	3.2.5		ı
siehe in den	2.2.6	2.2.4		3.2.7	3.2.3		3.2.10	(3.2.8)	(3.2.7)		ı
Abschnitten:	(2.2.7)	2.2.5		(3.2.8)	3.2.5		(3.2.11)	(3.2.12)	(3.2.8)		ı
	2.2.8			3.2.10	(3.2.12)		(3.2.12)		(3.2.12)		ı
	(2.2.9)			(3.2.11)							ı

Tabelle 1 TZAKON/TZA 4 Handlings-Übersicht

1.5 Informationsdaten TZA 4

1.5.1 Initialisierung des TZAKON

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Auswahl	Init/Info
	TZA 4-Identifikationsdaten auslesen
	Freier Programm-Speicher im TZA 4
	Blank-Check Programmspeicher-EPROM 2 im TZA 4
	Wahl der seriellen COM-Schnittstelle des PC
	Einstellung des Druckers
	TZA 4-Datum/Zeit auf PC-Datum/Zeit setzen
	TZAKON-Hilfetext drucken

<F1> HELP <ENTER> OK? <F8> RETURN <F9> DATEIMENUE
Bild 2

1.5.2 TZA 4-Identifikationsdaten auslesen

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Es werden aus dem TZA 4 ausgelesen und angezeigt:

- Geräte-Bezeichnung
- Fertigungs-Nr. des TZA 4
- Fertigungs-Datum des TZA 4
- Sach-Nr. des Betriebssystem-EPROM
- Versions-Nr. des Betriebssystem-EPROMs

1.5.3 Freier Programmspeicher im TZA 4

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Anzeig	e freier Programmspeicher im TZA	4-EPROM 2
Freier EPROM 2	-Speicher für Programm/Dateien:	23690 Byte
Freier EPROM 2	-Speicher für Auto-Start-Einträge:	808 Byte
	· · · · · · · · · · · · · · · · · · ·	

Es wird angezeigt:

Bild 3

- Der noch freie Speicherplatz für Programme + Parameterdateien
- Der noch freie Speicherplatz für Einträge in die Auto-Start-Liste im EPROM 2 des TZA 4.

Der Speicherplatz ist in Byte angegeben.

Es werden belegt:

- Bei jedem Eintrag in die Autostart-Liste: 2 Byte
- Beim Sichern eines Programms + Datei: x Byte
 (x ist abhängig von Programm und der Datei-Länge;
 Richtwert ca. 2 KByte).

1.5.4 Blank-Check Programmspeicher-EPROM 2 im TZA 4

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Nach Wechsel des Programmspeicher-EPROM 2 ist ein Blank-Check des neuen EPROM's zwingend notwendig.

Bedingung:

- Das neue EPROM 2 muß vollständig leer sein, bevor das 1. Programm "eingebrannt" wird.
- Es dürfen nur EPROM's des Typs 27C256 mit 12 V Brennspannung und einer Zugriffszeit ≤ 200 ns verwendet werden.

1.5.5 Wahl der seriellen COM-Schnittstelle des PC

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Anzeige	Serielle COM-Schnittstelle	
	Serielle Schnittstelle COM1**	
	Serielle Schnittstelle COM2	

<F1> HELP
<F8> RETURN
<F9> STARTMENUE
Bild 4

Es kann zwischen der seriellen PC-Schnittstelle COM1 und COM2 gewählt werden. Die mit ** gekennzeichnete

Die Nummer der gewählten Schnittstelle wird in INIT.DAT abgespeichert (siehe Abschnitt 1.2)

1.5.6 Einstellen des Druckers

Schnittstelle ist eingeschaltet.

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Zum Ausdrucken können 68 oder 72 Zeilen pro Seite gewählt werden. Die durch ** gekennzeichnete Seitenlänge ist aktuell eingestellt.

Die gewählte Anzahl wird in INIT.DAT abgespeichert.

1.5.7 TZA 4-Datum/Zeit auf PC-Datum/Zeit setzen

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Die im TZA 4 installierte Echtzeit-Uhr wird mit der PC-Uhr synchronisiert.

PC-Uhr auf DOS-Ebene eventuell vorher kontrollieren und korrigieren:

DOS-Befehl: DATE → Datum TIME → Uhrzeit

ACHTUNG! Bei Spannungsausfall bleibt die TZA 4-Uhr stehen. Sie wird bei Spannungswiederkehr und Restart eines Programmes mit DATE-TIME (1) wieder gestartet.

1.5.8 TZA 4-Hilfetext drucken

Menü-Pfad: → STARTMENUE → Info TZA 4/Init TZAKON

Es wird der gesamte TZAKON-Hilfetext auf Drucker ausgegeben (ca. 20 Seiten DIN A4).

2 Bedienoberfläche/P-Programme

2.1 Allgemeine Beschreibung

Beschreibung der P-Programme siehe Abschnitt 1.1 Nachfolgend sind die Funktionen zur Handhabung der P-Programme beschrieben.

2.2 Programm/Datei aus PC laden oder parametrieren

2.2.1 Programm/Datei nach Aufgabengebiet wählen

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE

Auswahl	Aufgabengebiete		
P100P199	Mathematische/logische Verknüpfungen		
P200P299	Kurvenzug, Sattdampf-Grenzlinie		
P300P399	Füllstand		
P400P499	Durchfluß (Wirkdruck-Verfahren)		
P500 P599	Durchfluß (VolMesser, offene Gerinne)		
P600P699	Wärme-, Kälte-Leistung		
P700P799	Rotor-Temperatur		
P800 P899	Drehzahl, Zeitfunktionen		
P900 P999	Sonderprogramme		

<F1> HELP <ENTER> OK? <F8> RETURN <F9> STARTMENUE
Bild 5

Aufgelistet sind:

- Alle Aufgabengebiete aus denen P-Programme ausgewählt werden können.
- Die Gebietsgrenzen der Programm-Nr. Pxxx...Pxxx
 Ein Aufgabengebiet anwählen und mit ENTER quittieren.
 Programmenü wird ausgegeben.

luswahl	Programme
P631-V01.HEX	Wärmeleistung, Dampf W = f(dP, P, T)
P632-V01.HEX	Wärmeleistung, Dampf $W = f(dP, P)$
633-V01.HEX	Wärmeleistung, Dampf $W = f(dP, T)$
636-V01.HEX	Wärmeleistung, Sattdampf $W = f(dP, P)$
637-V01.HEX	Wärmeleistung, Sattdampf $W = f(dP, T)$
P641-DUM.HEX	Heizleistung, Gas >> P141-Vxx.HEX

<F1> HELP <ENTER> OK? <F8> RETURN <F9> STARTMENUE
Bild 6

Aufgelistet werden:

- Alle Namen der Programme aus dem vorher gewählten Aufgabengebiet. Sind keine Programme aus dem gewählten Aufgabengebiet abgespeichert, erfolgt die Meldung: KEINE PROGRAMME VORHANDEN.
- Kurzbeschreibung der Programme
- Funktionsformel der Programme

Die Programmnamen setzen sich zusammen aus der Programm-Nr., der Versions-Nr. (z.B. V01) und der Extension "HEX". Wenn die Versions-Nr. "DUM" ist, dann werden äquivalente Programme aufgerufen (z.B. für P641-DUM.HEX wird P141-Vxx.HEX aufgerufen).

2.2.2 Parameter-Datei-Typ wählen

Auswahl	Parameter-Datei-Typ Programm: P631-V0		
	Neue Datei		
	Default-Datei		
	Vorhandene Dateien		

<F1> HELP <ENTER> OK? <F8> RETURN <F9> STARTMENUE
Bild 7

In der Kopfzeile des Menüs ist der Name des vorher gewählten Programms angegeben, zu dem eine Parameter-Datei erstellt bzw. eine vorhandene Datei ausgewählt oder verändert werden soll.

Es kann gewählt werden zwischen

- einer neuen Datei
- einer Default-Datei (Datei mit Muster-Daten)
- vorhandenen Dateien

Bei Wahl: "Vorhandene Dateien" werden alle zum gewählten Programm existierenden Parameter-Dateien aufgelistet. Ist noch keine Datei für dieses Programm erstellt worden, wird Meldung ausgegeben. In diesem Fall wird empfohlen auf die Default-Datei zurückzugreifen.

2.2.3 Datei/Programm nach Auftrags-, Datei- oder Fertigungs-Nr. wählen

Menü-Pfad: → STARTMENUE → PARAMETER-DATEIEN NACH AUFTRAGS-NR.

oder: → STARTMENUE → PARAMETER-DATEIEN NACH DATEI-NR.

oder: → STARTMENUE → PARAMETER-DATEIEN NACH FERTIG.-NR.

Auswahl	Parameterdateien n	ach Datei-Nr.	
Datei-Nr.	Auftr./Pos.	ProgrNr.	FertigNr.
90032104	56008086/0001	P421	6.631070.0
90041700	56008087/0002	P111	6.403211.0
90041802	65008123/0001	P112	6.632174.0
90050201	66503423/0001	P311	6.632343.0

<F1> HELP <ENTER> OK? <F8> RETURN <F9> STARTMENUE
Bild 8

Vom Startmenü aus können alle Namen der gespeicherten Dateien, sortiert nach Auftrags-Nr., Datei-Nr. oder nach Fertigungs-Nr. aufgelistet werden (1. Spalte). Von hier aus kann eine Datei mit dem zugehörigen Programm ausgewählt werden.

2.2.4 Dateimenü

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE

→ PROGRAMME → PARAMETER-DATEI-

TYPE

oder: → STARTMENUE → PARAMETER-DATEIEN NACH AUFTRAG-NR.

oder: → STARTMENUE → PARAMETER-DATEIEN NACH DATEI-NR.

oder: → STARTMENUE → PARAMETER-DATEIEN NACH FERTIG.-NR.

Auswahl Dateimenü Programm: P631-V01; Datei 90041705

Parameter-Datei bearbeiten
Programm und Datei zum TZA 4 transferieren
Programm und Datei im TZA 4 starten
Programm/Datei im TZA 4-EPROM 2 sichern
Programm/Datei nach Reset autom. starten

<F1> HELP <ENTER> OK? <F8> RETURN <F9> STARTMENUE BIID 9

Vom Dateimenü aus kann die ausgewählte Datei + Programm den im Menü aufgelisteten, in den folgenden Unterabschnitten beschriebenen Funktionen unterzogen werden.

2.2.5 Parameter-Datei bearbeiten (parametrieren)

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE

→ PROGRAMME → PARAMETER-DATEI-TYPE → DATEIMENUE

 \rightarrow Parameter-Datei bearbeiten

oder: → STARTMENUE → PARAMETER-DATEIEN
NACH AUFTRAGS-NR. → DATEIMENUE

→ Parameter-Datei bearbeiten

oder: → STARTMENUE → PARAMETER-DATEIEN NACH DATEI-NR. → DATEIMENUE

→ Parameter-Datei bearbeiten

oder: → STARTMENUE → PARAMETER-DATEIEN
NACH FERTIG.-NR. → DATEIMENUE

→ Parameter-Datei bearbeiten

Nach Wahl einer Datei + Programm sucht TZAKON das entsprechende Parametrier-Programm aus und startet es.

Es erscheint folgendes Menü:

Auswahl	Programm: 631; Datei-Nr.: 90041705 Blatt 1/1				
Auslegungs	-Daten	anzeigen			
Auslegungs	-Daten	ändern			
Auslegungs	-Daten	drucken			
Kontroll	-Werte	anzeigen			
Kontroll	-Werte	ändern			
Kontroll	-Werte	drucken			
Programm	-Beschreibung	drucken			
90041705.DAT	•	sichern			

<F1> HELP
<ENTER> OK?
<F9> DATEIMENUE

Bild 10

Von diesem Menü aus können die Auslegungs-Daten der Parameter-Datei angezeigt, verändert oder auf Drucker ausgedruckt werden. Vom Parametrier-Programm werden programm- und datei-spezifische Kontrollwerte berechnet. Sie können vom Menü aus angezeigt, verändert oder ausgedruckt werden.

Mit diesen Kontrollwerten kann das TZA 4 nach dem Laden und Starten stimuliert und kontrolliert werden.

Es gibt Parametrier-Programme, für die keine Kontrollwerte im Programm generiert werden. Mit Hilfe von "Kontrollwerte ändern" können jedoch Kontrollwerte durch freigewählte Eingabewerte errechnet werden.

Bei "Datei sichern" werden die durch Eingabewerte erzeugten Kontrollwerte gelöscht.

Daher vor dem Sichern, falls gewünscht, "Kontrollwerte drucken".

Jedes Parametrier-Programm ist individuell auf das entsprechende Programmpaket ausgelegt. Eine genaue Beschreibung kann durch Einblenden einer HELP-Anweisung (Taste F1) auf dem Bildschirm ausgegeben werden.

Bei erstmaligem Erstellen einer neuen Datei wird empfohlen den Helptext auszudrucken (Menü-Zeile: "Programm-Beschreibung drucken").

Beim Ändern der Auslegungs-Daten wird automatisch von TZAKON ein neuer Parameter-Datei-Name vergeben. Er setzt sich zusammen aus dem aktuellen Tages-Datum + einer 2stelligen Nr. Bei "Auslegungs-Daten ändern" müssen alle Parameter-Werte (geänderte und ungeänderte) durch ENTER quittiert werden. Dabei werden die Werte plausibilisiert und eventuell Fehlermeldungen ausgegeben.

Sind alle Werte eingegeben, ist eine Sicherung der Datei über die Menü-Zeile "xxxxxxxxx.DAT sichern" zwingend notwendig. Die neue Parameter-Datei wird vom TZAKON in das Unterverzeichnis DATFILE abgelegt und kann dann vom "DATEIMENUE" aus verschiedenen Operationen unterzogen werden (siehe Abschnitt 2.2.6 bis 2.2.9).

2.2.6 Programm/Datei zum TZA 4 transferieren

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE

- → PROGRAMME → PARAMETER-DATEI-TYPE → DATEIMENUE
- → Programm und Datei zum TZA 4 transferieren
- oder: → STARTMENUE → PARAMETER-DATEIEN NACH DATEI-NR. → DATEIMENUE
 - → Programm und Datei zum TZA 4 transferieren
- TZA 4 über Schnittstelle an PC anschließen.
- Im DATEIMENUE die Menü-Zeile "Programm und Datei zum TZA 4 transferieren" anwählen und mit ENTER quittieren.
- Das Programm und die Datei werden zum TZA 4 transferiert und im RAM des TZA 4 abgelegt.

2.2.7 Programm und Datei im TZA 4 starten

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE

- → PROGRAMME → PARAMETER-DATEI-TYPE → DATEIMENUE
- → Programm und Datei zum TZA 4 transferieren
- → Programm und Datei im TZA 4 starten

oder: → STARTMENUE → PARAMETER-DATEIEN NACH DATEI-NR. → DATEIMENUE

- → Programm und Datei zum TZA 4 transferieren
- → Programm und Datei im TZA 4 starten

oder: → STARTMENUE → PROGRAMME/
DATEIEN AUS TZA 4-VERZEICHNIS

- → DATEIMENUE
- → Programm und Datei im TZA 4 starten
- 1. TZA 4 über Schnittstelle an PC anschließen.
- 2. Im DATEIMENUE die Menü-Zeile "Programm und Datei im TZA 4 starten" anwählen und mit ENTER quittieren.
- 3. TZAKON schaltet in den Terminal-Mode.
- 4. Das BASIC-Print-Flag im TZA 4 wird gesetzt (siehe Abschnitt 6.3).
- 5. Programm u. Datei werden gestartet.
- 6. TZA 4-Software-Überwachung wird aktiviert (gelbe LED auf Frontplatte verlöscht).
- Die vom TZA 4 über die Schnittstelle ausgegebenen Meßergebnisse werden auf dem Bildschirm angezeigt.

Zum Stoppen des Programms im TZA 4 gleichzeitig CTRL und C drücken.

8. Das BASIC-Print-Flag im TZA 4 wird auf 0 gesetzt (siehe Abschnitt 6.3).

2.2.8 Programm/Datei im EPROM 2 des TZA 4 sichern

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE

- → PROGRAMME → PARAMETER-DATEI-TYPE → DATEIMENUE
- → Programm und Datei zum TZA 4 transferieren
- → Programm und Datei im EPROM 2 sichern

oder: → STARTMENUE → PARAMETER-DATEIEN NACH DATEI-NR. → DATEIMENUE

- → Programm und Datei zum TZA 4 transferieren
- → Programm und Datei im EPROM 2 sichern

TZA 4 über Schnittstelle an PC anschließen.

Programm und Datei zum TZA 4 transferieren.

Programm und Datei im EPROM 2 sichern.

TZAKON führt dabei folgende Funktionen aus:

- Teste, ob das zu sichernde Programm bereits im EPROM 2 gespeichert ist. Wenn ja, wird nur die Parameter-Datei gesichert. Ist Programm und Datei bereits gespeichert erfolgt Meldung.
- 2. Teste, ob im EPROM 2 noch genügend Speicherplatz zur Verfügung steht.
- 3. Kopiert Programm und/oder Datei vom RAM ins EPROM 2 und sichert es dort durch "Einbrennen".

Achtung! Folgende Punkte müssen beachtet werden:

- Vor dem Sichern Steckbrücke Br.51 (Brennspannungbrücke) im TZA 4 stecken (siehe Gebrauchsanweisung 42/18-40-).
 - Nach dem Sichern Brücke zurück auf Br52
- Der Binär-Eingang EB1 darf während des Brennvorgangs nicht beschaltet oder muß auf high-Pegel gelegt werden.

2.2.9 Programm/Datei nach Reset automatisch starten

Menü-Pfad: → STARTMENUE → AUFGABENGEBIETE

- → PROGRAMME → PARAMETER-DATEI-TYPE → DATEIMENUE
- → Programm und Datei zum TZA 4 transferieren
- → Programm und Datei im EPROM 2 sichern
- → Programm und Datei nach Reset automatisch starten

oder: → STARTMENUE → PARAMETER-DATEIEN NACH DATEI-NR. → DATEIMENUE

- → Programm und Datei zum TZA 4 trans-
- → Programm und Datei im EPROM 2 sichern
- → Programm und Datei nach Reset automatisch starten

oder: → STARTMENUE → PROGRAMME/
DATEIEN AUS TZA 4-VERZEICHNIS

- → DATEIMENUE
- → Programm und Datei nach Reset automatisch starten

Im EPROM 2 des TZA 4 können mehrere Programme und Dateien gespeichert sein. Nach Einschalten der Versorgungsspannung soll ein Programm/eine Datei automatisch gestartet werden.

Programm/Datei muß dazu in die Autostart-Liste eingetragen werden.

Beim Auslesen des TZA 4-Verzeichnisses ist dieses Programm/Datei mit ** gekennzeichnet.

Nach Auslösen eines Resets über die Reset-Taste bzw. durch Power-On-Reset wird:

- Die Datei vom EPROM 2 in das RAM kopiert
- Das Programm aus dem EPROM 2 heraus gestartet.
- Die TZA 4-Software-Überwachung aktiviert (gelbe LED auf Frontplatte erlöscht).

2.3 Datei aus TZA 4 lesen, anzeigen und ändern

Folgende Funktionen sind möglich:

- Auslesen aller im EPROM 2 gespeicherten Programmund Parameter-Datei-Namen
- Auswahl eines Programms oder einer Datei (siehe Abschnitt 2.3.1)
- Rückdokumentation der Parameter-Datei
- Starten eines Programms bzw. Eintrag in die Auto-Start-Liste

2.3.1 Programm/Datei aus TZA 4-Verzeichnis wählen

Menü-Pfad: → STARTMENUE → PROGRAMME/
DATEIEN AUS TZA 4-VERZEICHNIS

TZAKON liest alle im EPROM 2 gespeicherten Programmund Datei-Namen und listet diese im Menü "PRO-GRAMME UND PARAMETER-DATEIEN AUS TZA 4" auf:

Auswahl	Programme und Parameter-Dateien aus TZA 4						
ProgrNr. #0052201 DEM01	Datei-Nr.	Auftr./Pos.	FertigNr.				
P111-V01	90041700**	53467235/0001	6.403210.0				
P111-V01	90041701	53467235/0001	6.403210.0				
P631-V01 Testprog	90041900	53467235/0001	6.403210.0				

<F1> Help <Enter> OK? <F8> Return <F9> Dateimenue

Die Programm-Namen (Nrn.) werden alphabetisch aufgelistet. Bei P-Programmen werden zu jedem Programm die Auftrags-Nr., Position und die Fertigungs-Nr. mit ausgegeben. Die Zeichen "**" hinter der Datei deuten an, daß dieses Programm bzw. diese Datei in der Autostart-Liste eingetragen ist und nach einem Reset oder Power-On-Reset automatisch gestartet wird.

- Ist das gewählte Programm ein P-Programm, werden die Parameter-Datei-Daten aus dem TZA 4 ausgelesen und als Datei "xxxxxxxxx.GET" im PC-Unterverzeichnis DATFILE des PC abgespeichert. Danach verzweigt TZAKON zum DATEIMENUE.
 - Von hier aus kann die Datei durch Wahl "Parameter-Datei bearbeiten" ausgedruckt bzw. verändert werden.
- Ist das gewählte Programm ein #- oder User-Programm, dann verzweigt TZAKON zum Menü BASIC 52-PRO-GRAMM-START.

Bild 11

3 Bedienoberfläche/#-Programme (Programme ohne Parameter-Dateien)

3.1 Programm aus PC in TZA 4 laden

3.1.1 Programm wählen

a) Nach Aufgabengebiet:

Menü-Pfad: → STARTMENUE → AUFGABEN-GEBIETE → PROGRAMME

Bei Wahl eines Programms nach Aufgabengebieten (siehe Abschnitt 2.2.1) wird dieses in den Programm-Speicher geladen.

Beispiel:

In Bild 12 wird für die Aufgabe P321 das zugehörige #-Programm #321-V01 geladen.

Auswahl	Programme
P311-V01.HEX	Kessel-Wasserstand, P-Korrektur
P312-V01.HEX	Kessel-Wasserstand, P-T-Korrektur
P321-DUM.HEX	lieg. Behälter V=f(h) >> #321-V01.B52

<F1> Help <Enter> OK? <F8> Return <F9> Startmenue Bild 12

Mit F2 kann das BASIC-PROG-MENUE eingeblendet, und das Programm verschiedenen Operationen unterzogen werden.

b) Kundenspezifisches Programm

Bei Wahl eines Programms, das kundenspezifisch erstellt wurde, also nicht im Menü 2.2.1 aufgelistet werden kann, wird das #-Programm aufgerufen über:

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/ TERMINAL-MODE

→ F2 → Programm von Disk/Platte lesen

3.1.2 BASIC-PROG.-MENUE

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/ TERMINAL-MODE

→ F2

Terminal Online-editor

BASIC-PROG.-MENUE

Terminal / On-Line-Editor
Neues Programm erstellen
Programm Editor
Programm abspeichern
Programm unter neuem Namen abspeichern
Programm von Disk / Platte lesen
Programm ins TZA 4 transferieren
Programm im TZA 4 starten
Programm aus TZA 4 auslesen
Programm vom TZA 4-RAM ins EPROM 2 sichern
Programm nach Reset autom. starten
Programm drucken

<F1> HELP <F2> Menue <F8> Return <F9> Startmenue Bild 13

3.1.3 Programm-Editor

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/ TERMINAL-MODE

→ F2 → Programm Editor

Vor Aufruf des Programm-Editors muß über die Menü-Zeile "Programm von Disk/Platte lesen" ein Programm ausgewählt und in den Programm-Speicher geladen werden.

- 1. Das geladene Programm wird angezeigt.
- 2. Der Name des Programms erscheint in der Kopfzeile.
- 3. Das geladene Programm kann verändert werden (siehe Abschnitt 4.2.1).
- 4. Das Programm kann unter seinem Namen im Unterverzeichnis B52DIR abgespeichert werden.
- 5. Das Programm kann unter einem neuen Namen im Unterverzeichnis B52DIR abgespeichert werden.

3.1.4 Programm abspeichern

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/ TERMINAL-MODE

→ F2 → Neues Programm erstellen

→ F2 → Programm abspeichern

oder: → F2 → Programm von Disk/Platte lesen

→ F2 → Programm Editor (Programm abändern)

→ F2 → Programm abspeichern

Folgende Funktionen werden ausgeführt:

- Das im Programm-Speicher geladene Programm wird unter dem in der Kopfzeile angegeben Namen und der Extension ".B52" im Unterverzeichnis "B52DIR" abgespeichert.
- Existiert bereits ein Programm mit gleichem Namen, wird dieses zuerst unter der Extension "BAK" gesichert.

3.1.5 Programm unter neuem Namen abspeichern

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/ TERMINAL-MODE

→ F2 → Programm von Disk/Platte lesen

→ F2 → Programm Editor (Programm abandern)

→ F2 → Programm unter neuem Namen abspeichern

Folgende Funktionen werden ausgeführt:

- 1. Es wird ein Eingabe-Fenster eingeblendet
- 2. Der neue Programmname muß eingegeben werden

3.1.6 Programm von Diskette oder Platte lesen

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/ TERMINAL-MODE

→ F2 → Programm von Disk/Platte lesen

→ F2 → Programm Editor

Folgende Funktionen werden ausgeführt:

- Alle #-Programme und alle User-Programme die im Unterverzeichnis B52DIR gespeichert sind werden aufgelistet.
- Bei #-Programme werden zusätzlich Auftrags-Nr./Position, Fertigungs-Nr. und Programm-Namen mit ausgegeben.

Beispiel:

Auswahl Programme von Disk/Platte							
File-Name	Auftr./Pos.	Fertig.Nr.	Prog.	Lng.	Datum		
#0032300.B52	56008086/0001	6.631016.0	#722-002	3947	01-03-1990		
#0050300.B52	64108212/0001	6.631017.0	#SUM24ST	3947	01-03-1990		
#0032300.B52	56008086/0001	6.631018.0	Q3+V12/T	3947	01-03-1990		
DEM01 .B52				1033	03-04-1990		
TESTPROG.B52				1240	22-05-1990		

<F1> HELP <ENTER> OK? <F8> RETURN <F9> STARTMENUE
Bild 14

Es kann ein Programm angewählt und in den Programm-Speicher geladen werden.

Im Programm-Editor besteht die Möglichkeit das Programm zu verändern (siehe Abschnitt 3.2.3).

3.1.7 Programm zum TZA 4 transferieren

Menü-Pfad: → STARTMENUE

- → PROGRAMMERSTELLUNG/ TERMINAL-MODE
- → F2 → Programm von Disk/Platte lesen
- → F2 → Programm ins TZA 4 transferieren

Das im Programm-Editor geladene Programm wird zum TZA 4 transferiert und im RAM abgespeichert.

3.1.8 Programm im TZA 4 starten

Menü-Pfad: → STARTMENUE

- → PROGRAMMERSTELLUNG/ TERMINAL-MODE
- → F2 → Programm von Disk/Platte lesen
- → F2 → Programm ins TZA 4 transferieren
- → F2 → Programm im TZA 4 starten

Folgende Funktionen werden ausgeführt:

- Das im TZA 4-RAM geladene Programm wird gestartet
- Das Print-Ausgabeflag im TZA 4 gesetzt
- Meßwerte werden über die Schnittstelle auf dem Bildschirm ausgegeben (falls im Programm vorgesehen).

3.1.9 Programm aus TZA 4 auslesen

Menü-Pfad: → STARTMENUE

- → PROGRAMMERSTELLUNG/ TERMINAL-MODE
- → F2 → Programm aus TZA 4 auslesen

Das im TZA 4 aktive Programm wird zum PC transferiert und im Programmspeicher abgelegt.

3.1.10 Programm vom TZA 4-RAM in das EPROM 2 sichern

Menü-Pfad: → STARTMENUE

- → PROGRAMMERSTELLUNG/ TERMINAL-MODE
- → F2 → Programm von Disk/Platte lesen
- → F2 → Programm ins TZA 4 transferieren
- → F2 → Programm vom TZA 4-RAM ins EPROM 2 sichern

Das zum TZA 4-RAM transferierte Programm (siehe Abschnitt 3.2.7) und dessen Programm-Name werden ins EPROM 2 kopiert und dort gesichert.

Achtung! Folgende Punkte müssen beachtet werden:

- Vor dem Sichern muß die Brücke Br51 (Brennspannungsbrücke) im TZA 4 gesteckt sein.
 Nach dem Sichern Brücke zurück auf Br52.
- Der Binär-Eingang EB1 darf während des Brennvorgangs nicht beschaltet sein oder muß auf high-Pegel gelegt werden.

3.1.11 Programm nach Reset automatisch starten

Menü-Pfad: → STARTMENUE

- → PROGRAMMERSTELLUNG/ TERMINAL-MODE
- → F2 → Programm von Disk/Platte lesen
- → Programm ins TZA 4 transferieren
- → Programm vom TZA 4-RAM ins EPROM 2 sichern
- → Programm nach Reset automatisch starten

oder: → STARTMENUE

- → PROGRAMME/DATEIEN AUS TZA 4-VERZEICHNIS
- → BASIC52-PROGRAMM-START
- → Programm nach Reset automatisch starten

Das im EPROM 2 gesicherte Programm (siehe Abschnitt 3.2.10) wird in die Autostart-Liste eingetragen. Nach Ausfall der Versorgungsspannung und Wiederkehr bzw. nach Betätigung der Reset-Taste wird dieses Programm automatisch gestartet.

3.1.12 Programm ausdrucken

Menü-Pfad: → STARTMENUE

- → PROGRAMMERSTELLUNG/ TERMINAL-MODE
- → F2 → Programm von Disk/Platte lesen

oder: → Programm aus TZA 4 auslesen

→ Programm ausdrucken

Das im TZAKON-Editor-Speicher geladene Programm-Listing wird auf den angeschlossenen Drucker ausgegeben.

3.2 Programm aus TZA 4 starten oder in Autostart-Liste eintragen

Menü-Pfad: → STARTMENUE

- → PROGRAMM/DATEIEN AUS TZA 4-VERZEICHNIS
- → BASIC52-PROGRAMM-START
- → Programm im TZA 4 starten
- oder: → Programm in AUTO-START-LISTE eintragen

Wie in den Abschnitten 2.3 und 2.3.1 beschrieben, kann das Programm-Inhaltsverzeichnis des angeschlossenen TZA 4 ausgelesen und über das Menü (Bild 11) ein #-Programm oder ein User-Programm gewählt werden. Das Programm kann entweder sofort gestartet oder in die Autostart-Liste eingetragen werden (Abschnitte 3.1.10 und 3.1.11 beachten!).

4 Bedieneroberfläche / User-Programme

4.1 Allgemeine Beschreibung.

TZAKON ermöglicht dem Anwender eigene User-Programme zu erstellen. Dazu stellt TZAKON einen Editor zur Verfügung. Die Programme werden zum TZA 4 transferiert, im TZA 4 gestartet, getestet und im EPROM 2 gesichert. Die Programme lassen sich allerdings nicht wie die unter Abschnitt 2 beschriebenen P-Programme parametrieren. Parameter-Werte müssen bei der Programmerstellung als Konstanten mit in das Programm geschrieben werden.

4.2 Programm auf PC erstellen oder ändern

4.2.1 Neues Programm erstellen

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/

TERMINAL-MODE

→ F2 → Neues Programm erstellen

Folgende Funktionen werden ausgeführt:

- Der Programm-Editor wird aufgerufen

- Der Programm-Speicher wird gelöscht
- Das neue Programm erhält den Namen NONAME. Dieser Name kann beim Abspeichern des Programms (siehe Abschnitt 3.2.5) verändert werden.

Mit Hilfe des Programm-Editors kann ein User-Programm erstellt werden:

- Jede BASIC-Zeile beginnt mit einer Zeilen-Nr.
- Die Zeilen können max. 78 Zeichen lang sein und müssen mit RETURN abgeschlossen werden.
- Die max. Programmlänge beträgt 500 Zeilen.

Der TZAKON-Editor unterstützt die Funktions-Tasten:

- INSERT/OVERWRITE
- PgUp/PgDn
- BACKSPACE
- DELETE
- ENDPOS
- POS1
- CURSOR-Tasten
- Blockmanipulationen

Block markieren F3 Block-Anfang, F4 Block-Ende.

Markierung löschen F4
Block kopieren CTRL-C
Block verschieben CTRL-V
Block löschen CTRL-D

Programmierungs-Hinweise und Programmbeispiele siehe Abschnitte 6 und 7

4.3 On-Line-Editor

Menü-Pfad: → STARTMENUE

→ PROGRAMMERSTELLUNG/ TERMINAL-MODE

In diesem Modus ist der PC als Terminal (Ein- u. Ausgabe-Einheit) geschaltet. Über das Terminal können BASIC52-Befehle direkt in den On-Line-Editor des angeschlossenen TZA 4 geschrieben werden.

4.3.1 Terminal / On-Line-Editor

BASIC-Befehle werden vom BASIC-Interpreter sofort übersetzt und ausgeführt.

Beispiel: PRINT 3*4

12

Es können auch mehrere Befehle in eine Zeile geschrieben werden. Jeder Befehl ist durch ":" vom nächsten Befehl zu trennen.

Beispiel: FOR I = 1 TO 6: PRINT I,: NEXT I

123456

Es können auch User-Programme direkt in das RAM eingegeben, mit RUN gestartet und ausgetestet werden. Jede BASIC-Zeile begint mit einer Zeilen-Nr. Die Zeile kann max. 78 Zeichen lang sein und muß mit RETURN quittiert werden.

Beispiel: 10 FOR I = 1 TO 6

20 PRINT I, 30 NEXT I RUN 1 2 3 4 5 6 READY

lst die Zeile quittiert, kann sie nur durch Überschreiben der gesamten Zeile verändert werden. Gelöscht wird eine Zeile durch Eingabe der Zeilen-Nr. und Quittierung.

5 Referenzliste der BASIC 52-Befehle

5.1 Direkt-Modus-Befehle.

BASIC 52 verfügt über 2 Arten von Befehlen:

- RUN-Modus-Befehle
- Direkt-Modus-Befehle

RUN-Modus-Befehle dienen zur Ausführung eines Programms. Sie sind also Bestandteil des Programm-Codes.

Direkt-Modus-Befehle dürfen, im Gegensatz zu RUN-Modus-Befehlen, nicht in einem Programm stehen. Sie werden direkt über den On-Line-Editor ins TZA 4 geschrieben und mit RETURN zur Ausführung gebracht. Diese Befehle sind nachfolgend aufgeführt.

RUN

Startet das geladene BASIC-Programm bei der ersten Zeilen-Nummer. Zuvor werden alle Variablen auf 0 gesetzt und alle Interrupts zurückgesetzt. Gestoppt wird das Programm mit CTRL-C. Das Programm kann auch bei einer bestimmten Zeilen-Nummer (z.B. RUN 100) gestartet werden).

Nach Stoppen eines Programms durch CTRL-C oder STOP-Befehl im Programm, kann dieses mit CONT fortgesetzt werden.

LIST

Gibt das aktuelle BASIC-Programm-Listing aus. Durch CTRL-S wird die Ausgabe angehalten und mit CTRL-Q fortgesetzt. Durch Angabe der Zeilen-Nr. kann ein bestimmter Bereich ausgegeben werden.

Beispiel:

LIST 200 listet ab Zeile 200 bis Programmende LIST 200-900 listet von Zeile 200...900

Löscht den gesamten BASIC-RAM-Speicher, die Variablen werden auf 0 und die Interrupts zurückgesetzt.

ROM, RAM

Mit diesen Kommandos kann zwischen RAM- u. EPROM-Modus umgeschaltet werden. Im RAM-Modus kann das Programm editiert, im ROM-Modus kann ein Programm aus dem EPROM selektiert werden.

ROM 3 selektiert das 3. im EPROM abgespeicherte Programm. Mit RUN wird dieses Programm gestartet.

(Wird ein Programm über TZAKON ausgewählt und gestartet, wird automatisch der richtige Modus gesetzt, siehe Abschnitt 3.2).

Das im EPROM selektierte Programm wird ins RAM kopiert.

5.2 RUN- und Direkt-Modus-Befehle

ADU(1,2,...,6)

AD-Wandlung der Signal-Werte der analogen Eingänge E1... E6 in digitale Signal-Werte. Es sind nur die Eingänge einzutragen, die gewandelt werden sollen. Ablage der digitalen Signalwerte auf dem Stack. Mit POP x1,..,x6 werden die Signalwerte den Variablen x1,..,x6 zugewiesen. Es müssen immer soviele Variablen angegeben werden, wie Eingänge in ADU(...) angegeben wurden.

Eingänge vor Aufruf von ADU() nach der Code-Tabelle (siehe Abschnitt 6.1.1, Tabelle 3) mit:

DEFINP(x, x, x, x, x, x) Eingänge definieren und SKAL0 automatische Skalierung

abschalten

Beispiel:

10 SKALO :rem Skalierung abschalten 20 DEFINP(1, 0, 2, 0, 5, 3) :rem Eingänge definieren 30 ADU(1, 3, 5, 6) :rem Eing.1,3,5,6 messen 40 POP E1, E3, E5, E6 :rem Variablen zuweisen

Setzt alle Variablen auf 0 und setzt alle Interrupts und Stack zurück.

CLEARI

Setzt alle Interrupts zurück

CL FARS

Setzt Stack zurück

CLOCK1

Startet den REALTIME CLOCK, Danach wird alle 5 Millisekunden die BASIC-Variable TIME mit 0.005 inkrementiert. Nach TIME = 65535.995 erfolgt ein Überlauf und TIME wird auf 0.000 gesetzt.

CLOCKO

Stoppt den REAL-TIME-CLOCK.

Ist nur im RUN-Modus einsetzbar.

DATA definiert ein numerisches Datenfeld. Die Daten müssen durch Komma getrennt werden.

Beispiel:

10 FOR A = 1 TO 6 20 READ X: PRINT X, 30 NEXT A 40 DATA 2, 5, 3, 7, 8, 6 RUN 253786

DATETIME(x)

Echtzeit-Uhr.

x=1Echtzeit-Uhr wird gestartet. x=0Echtzeit-Uhr wird gestoppt.

x=RDatum und Zeit wird auf Stack geladen.

Mit POP (z.B. POP Jh, Mt, Tg, St, Mn, Se) kann das Datum und die Zeit den Variablen

zugewiesen werden.

Die RAM-Zelle 6D16H wird jede Sekunde auf 0FFH gesetzt (Sekunden-Flag). Im BASIC52-Programm kann diese Zelle mit IF XBY(6D16H)=0FFH THEN xxx abgefragt werden.

Die Einstellung von DATUM und ZEIT kann auf 2 Arten erfolgen:

1. Durch direktes Beschreiben der RAM-Zellen im TERMI-NAL-Modus.

Beispiele:

XBY(6D15H) = Jh	(z.B. 89)
XBY(6D14H) = Mt	(z.B. 09)
XBY(6D13H) = Tg	(z.B. 27)
XBY(6D12H) = St	(z.B. 14)
XBY(6D11H) = Mn	(z.B. 55)
XBY(6D10H) = Se	(z.B. 00)

2. Durch direktes Laden der PC-Uhrzeit u. Datum in das TZA 4.

Menü-Pfad: → STARTMENUE

→ INFO TZA 4/INIT TZAKON

→ TZA 4-Datum/Zeit auf PC-Datum/Zeit

setzen

Beispiel: Jede Sekunde die Stunde, Minute und Sekunde ausgeben

10 DATETIME(1) :rem starte Echtzeituhr
20 DO: WHILE XBY(6D16H)=0 :rem warte auf Sek.-Flag
40 DATETIME(R) :rem lese Echtzeituhr
50 POP JH,MT,TG,ST,MN,SE :rem Variablen laden
60 PRINT ST,MN,SE :rem Zeit ausgeben
70 XBY(6D16H) = 0 :rem setze Sek.-Flag zurück
80 GOTO 20 :rem wiederhole

DAU(I)

Digitaler Ausgabewert (Digtaler Signalbereich 0..3600) DA-wandeln und auf A1-Ausgang (0...20mA) ausgeben. Ausgabewert x vorher mit PUSH x auf Stack abspeichern.

DAU(U)

Digitaler Ausgabewert (Digitaler Signalbereich 0..3600) DA-wandeln und auf A2-Ausgang (0..10 V) ausgeben. Ausgabewert x vorher mit PUSH x auf Stack abspeichern.

Beispiel:

10 SKALO :rem Skalierung abschalten
20 DEFINP(1, 1, 0, 0, 0, 0) :rem E1 u.E2 → -20 . . +20mA
30 ADU(1, 2) :rem E1 u. E2 messen -3600 . . +3600 Digit
40 POP E1, E2 :rem Variablen laden
50 SGN(E1) : SGN(E2) :rem Vorzeichen eliminieren
60 PUSH E1 : DAU(I) :rem Meßwert auf A1 ausgeben
70 PUSH E2 : DAU(U) :rem Meßwert auf A2 ausgeben
80 GOTO 30

DEFBINP(y)

y=0..8 definiert digitale Meßfunktionen der Binäreingänge EB1.. EB4/ENI. Mit RDIGVAL wird der Meßwert gelesen und durch POP x der Variablen x zugewiesen.

- siehe BASIC-Befehle DEFBINP(1..8)
- siehe Abschnitt 6.1.2 Tabelle 5: Funktionsübersicht
- siehe Abschnitt 5.10 Tabelle 2: BASIC-Befehle nur alternativ einsetzen!

DEFBINP(1, z)

Frequenz-Messung über Eingang: EB4/ENI Frequenzbereich über EB4: 0...100 kHz Frequenzbereich über ENI: 0...10 kHz

Zeitbasis z ist wählbar von 0.01..4.5 · 105 s.

Der Befehl DEFBINP(1, z) startet die Frequenzmeß-Routine. Sie läuft permanent im Hintergrund des aktuell laufenden BASIC-Programms.

Nach Ablauf der Zeitbasis wird ein Interrupt ausgelöst, die Frequenz (Hz) berechnet und im RAM abgespeichert, ein Leseflag auf 55 H gesetzt und die Routine neu gestartet. Mit dem Lese-Befehl "RDIGVAL" kann der Meßwert vom RAM in den Stack gelesen und mit "POP x" der Variablen x zugewiesen werden. Das Leseflag wird dabei wieder auf 0 gesetzt.

Liegt zum Zeitpunkt des Auslesens mit RDIGVAL noch kein neuer Meßwert vor, so wird der alte Meßwert ausgelesen. Durch Abfragen des Leseflags mit

IF XBY(69E7H) = 55H THEN RDIGVAL

kann der Lesebefehl RDIGVAL bedingt, immer nach Vorliegen eines neuen Meßwertes, ausgeführt werden. Beispiel 1: Permanentes Auslesen der gemessenen Frequenz 10

10 DEFBINP(1,2.5): rem Freq.-Messung, Zeitbasis = 2.5 s

20 RDIGVAL: POP FREQ: rem lese Frequenz 30 PRINT "Die Frequenz = ",FREQ, CR,

40 GOTO 20

Beispiel 2: Bedingtes Auslesen der gemessenen Frequenz

10 DEFBINP(1, 2.5): rem Freq.-Messung, Zeitbasis = 2.5 s 20 IF XBY(69E7H) = 55H THEN RDIGVAL: POP FREQ

30 PRINT "Die Frequenz = ",FREQ, CR,

40 GOTO 20

DEFBINP(2)

Zähler (Impulse) über Eingang : EB4/ENI Start/Stop über Eingang : EB1 Reset über Eingang : EB2 Start/Stop-Zeit : $200 \ \mu s \dots \infty$ Min.Impuls-Breite u. -Pause bei EB4 : $5 \ \mu s$ Min.Impuls-Breite u. -Pause bei ENI : $50 \ \mu s$ Max. Zahl der Impulse : $4 \cdot 10^9$ Impulse

Der Befehl DEFBINP(2) startet die Impuls-Zählroutine. Sie läuft permanent im Hintergrund des aktuell laufenden BASIC-Programms.

Liegen der Start-/Stop- und der Reset-Eingang auf high-Pegel, dann werden die am Impulseingang einlaufenden Impulse bei jedem Pegelwechsel low-high gezählt. Wird der Start-/Stop-Eingang auf low-Pegel gezogen, so wird die Zählung unterbrochen und das Leseflag gesetzt.

Ein Pegelwechsel am Reset-Eingang von high nach low setzt den Impulszähler wieder auf Null.

Mit dem Lesebefehl "RDIGVAL" kann zu jeder Zeit der aktuelle Zählerstand ausgelesen und mit dem Befehl "POP x" einer Variablen x zugewiesen werden.

Durch Abfragen des Leseflags mit

IF XBY(69E7H) = 55H THEN RDIGVAL

kann aber auch der Lesebefehl "RDIGVAL", bedingt nach einem Stop-Signal, ausgeführt werden.

Mit "RDIGVAL" wird auch das Leseflag wieder auf 0 gesetzt.

Beispiel 1: Permanentes Auslesen des aktuellen Zählerstandes

10 DEFBINP(2) :rem Freigabe Impulszähler 20 RDIGVAL: POP IMP :rem lese aufsummierte Impulse

30 PRINT "Impuls-Zahl = ".IMP

40 GOTO 20

Beispiel 2: Bedingtes Auslesen des Zählerstandes nach Stop-Signal

10 DEFBINP(2): rem Freigabe Impulszähler 20 IF XBY(69E7H) = 55H THEN RDIGVAL: POP IMP

30 PRINT "Impuls-Zahl = ",IMP, CR,

40 GOTO 20

DEFBINP(3)

Diese Routine ist identisch mit der von DEFBINP(2), nur die Eingänge sind anders belegt.

Zähler (-impulse) über Eingang : EB3 Start/Stop über Eingang : EB2 Reset über Eingang : EB1 Start-/Stop-Zeit : $200 \ \mu s \dots \infty$ Min. Impulsbreite und -Pause bei EB3 : $5 \ \mu s$

Max. Zahl der Impulse 4 · 10⁹ Impulse

DEFBINP(4)

Impulsdauer Eingang: EB1

Impulsdauer-Meßbereich : 0.0002..4660 s

Zeit-Auflösung : 1,1 µs

Der Befehl DEFBINP(4) gibt die Zeitmessung über Eingang EB1 frei. Die Routine läuft permanent im Hintergrund des aktuell laufenden BASIC-Programms.

Bei einem Pegelwechsel von low nach high am Zeitmeßeingang wird die interne Meßzeit gestartet und durch den folgenden Pegelwechsel von high nach low wieder gestoppt und das Leseflag gesetzt.

Mit dem Befehl "RDIGVAL" kann die Meßzeit ausgelesen und mit dem Befehl "'POP x" einer Variablen x zugewiesen werden. Liegt zum Zeitpunkt des Aufrufs von 'RDIGVAL' noch kein neuer Meßwert vor, so wird der vorherige Meßwert ausgegeben.

Durch Abfragen des Leseflags mit

IF XBY(69E7H)=55H THEN RDIGVAL

kann aber auch der Lesebefehl "RDIGVAL", bedingt nach Vorliegen einer neuen Messung, ausgeführt werden.

Beispiel 1: Permanentes Auslesen der Meßzeit

rem Freigabe Zeitmessung

20 RDIGVAL: POP ZT:

rem lese Meßzeit

30 PRINT "Meßzeit = ",ZT

40 GOTO 20

Beispiel 2: Bedingtes Auslesen der Meßzeit (nach Stopp der Meßzeit).

10 DEFBINP(4):

rem Freigabe Zeitmessung

20 IF XBY(69E7H) = 55H THEN RDIGVAL : POP ZT

30 PRINT "MeBzeit = ".ZT.CR.

40 GOTO 20

DEFBINP(5)

Routine ist identisch mit der von DEFBINP(4), nur der Eingang ist auf EB2 gelegt.

Impulsdauer Eingang : EB2

Impulsdauer-Meßbereich: 0,0002...4660 s

Meßzeit-Auflösung : 1,1 µs

DEFBINP(6)

Periodendauer-Messung Eingang: EB1

Periodendauer-Meßbereich : 0,0002...4660 s

Zeit-Auflösung : 1,1 µs

Der Befehl DEFBINP(6) gibt die Periodendauer-Messung über Eingang EB1 frei.

Mit der 1. High-Low-Flanke am Eingang wird die interne Meßzeit gestartet und durch die 2. High-Low-Flanke wieder gestoppt und das Leseflag gesetzt.

Mit der 3. High-Low-Flanke wird die nächste Messung wieder gestartet usw.

Mit dem Befehl "RDIGVAL" kann die Meßzeit ausgelesen und mit dem Befehl "POP x" einer Variablen x zugewiesen werden.

Liegt zum Zeitpunkt des Aufrufs von RDIGVAL noch kein neuer Meßwert vor, so wird der vorherige Meßwert ausgegeben.

Durch Abfragen des Leseflags kann aber auch mit IF XBY(69E7H) = 55H THEN RDIGVAL

der Lesebefehl "RDIGVAL" bedingt, nach Ende einer neuen Messung, ausgeführt werden.

Beispiel 1: Permanentes Auslesen der Perioden-Dauer

10 DEFBINP(6):

rem Freigabe Zeitmessung

20 RDIGVAL: POP PD:

rem Lese MeBzeit

30 PRINT "Periodendauer = ",PD

40 GOTO 20

Beispiel 2: Bedingtes Auslesen der Perioden-Dauer (bei Meßende der neuen Messung)

10 DEFBINP(4):

rem Freigabe Zeitmessung

20 IF XBY(69E7H) = 55H THEN RDIGVAL: POP ZT

30 PRINT "Periodendauer = ",ZT, CR,

40 GOTO 20

DEFBINP(7)

Routine ist identisch mit der von DEFBINP(6), nur der Eingang ist auf EB2 gelegt.

Periodendauer-Messung

über Eingang : EB2

Periodendauer-Meßbereich: 0,0002..4660 s

Meßzeit-Auflösung : 1,1 µs

DEFBINP(8)

Perioden-Dauer (in s) und Messung der Impuls-Anzahl

Periodendauer/Impuls-

Messung Eingang · FR1

Periodendauer-Meßbereich: 0.0005...327 s

Zeit-Auflösung : 50 µs

Mit Befehl DEFBINP(8) wird die Periodendauer-Messung und Impulszählung über Eingang EB1 freigegeben.

Die 1. High-Low-Flanke am Eingang EB1 startet die interne Perioden-Meßzeit, und inkrementiert den Impulszähler.

Die 2. High-Low-Flanke stoppt die Meßzeit wieder und setzt das Leseflag.

Mit der 3. High-Low-Flanke wird die nächste Messung wieder gestartet usw.

Mit dem Befehl "RDIGVAL" wird die Meßzeit und die Anzahl der seit dem letzten Aufruf von RDIGVAL eingelaufenen Impulse ausgelesen.

Mit dem Befehl "POP x, y" werden die Meßzeit x und die Impulszahl y den Variablen x,y zugewiesen.

Liegt zum Zeitpunkt des Aufrufs von RDIGVAL noch kein neuer Periodendauer-Meßwert vor, so wird der vorherige Meßwert und Impulszahl = 0 ausgegeben.

Beispiel: Durchflußmessung mit Impulsgeber

(z.B. Ovalrad-Impulsgeber)

Geber EB1: 1 impuls = 0.05 m^3

TZA 4-Zähler-

Ausgang AB1 : 1 Impuls = 1 m^3

Analog-Ausgang A1 : 0..20mA = 0..1000 m³/h

10 DEFBINP(8) : rem Messuna freigeben

20 RDIGVAL: POP PT, IMP: rem lese Periodendauer + Impulszahl

30 IF PT = 0 THEN FRQ = 0 ELSE FRQ = 1/PT: rem Freq.-Berechn.

40 MW = FRQ*3600*0.05: rem MeBwert-Berechn. MW (m3/h)

50 DW = 3600*MW/1000: rem Skalier.(3600 Digit = 1000 m³/h = 20 mA)

60 PUSH DW: DAU(I): rem Ausgabe $0...20 \text{ mA} \rightarrow 0..1000\text{m3/h}$

70 IMPOUT=INT(IMP*0.05+REST): rem berechne die Ausgabe-Impulse

80 REST=(REST+IMP*0.05)-IMPOUT: rem berechne Rest

90 IF IMPOUT <> 0 THEN PWM 500,500,IMPOUT; rem Impuls-Ausgang 100 PRINT U.(####.#), MW: rem zeige Durchfluß auf Monitor an

110 GOTO 20

Schaltet die mit DEFBINP(1..8) definierten Meßfunktion ab.

DEFINP(x, x, x, x, x, x)

Definiert die 6 analogen Eingänge entsprechend den gewünschten Eingangssignalen- bzw. Bereichen. Jedes x repräsentiert einen Eingang, für den der zugehörige Signal-Code eingesetzt werden muß (siehe Abschnitt 6.1.1).

Beispiel:

Definiere:

 $\begin{array}{lll} E1 \rightarrow 0..20 \text{mA} & \rightarrow \text{Code 1} \\ E2 \rightarrow 4..20 \text{mA} & \rightarrow \text{Code 1} \\ E3 \rightarrow -1 \text{V..} +1 \text{V} & \rightarrow \text{Code 3} \\ E6 \rightarrow \text{Pt} 100 \pm 200 \text{'C} & \rightarrow \text{Code 8} \\ E4,E5 \text{ sind unbenutzt.} & \rightarrow \text{Code 0} \end{array}$

 10 SKALO
 :rem Skalierung abschalten

 20 DEFINP(1,1,3,0,0,8)
 :rem definiere E1, E2, E3, E6

 30 ADU(1,2,3,6)
 :rem E1,E2,E3,E6 messen

 40 POP E1, E2, E3, E6
 :rem Variablen laden

 50 PRINT E1, E2, E3, E6
 :rem Ausgabe auf Bildschirm

DIM

Reserviert Speicherplatz für numerische ARRAY-Variablen. Es können max. bis zu 254 Elemente angegeben werden. Jedes Element belegt 6 Byte im RAM.

ARRAY-Variablen unter 10 Elemente brauchen nicht mit DIM dimmensioniert zu werden.

Beispiel:

10 DIM A(34) : rem ARRAY-A mit 34 Elementen

DO UNTIL

Ist nur im RUN-Mode einsetzbar.

Definiert eine Programmschleife. Diese Schleife wird sooft wiederholt, bis die Bedingung von UNTIL erfüllt ist.

Beispiel:

10 A = 0

20 DO

30 A = A+1

40 PRINT A

50 UNTIL A = 8

60 PRINT "Ende"

RUN

1 2 3 4 5 6 7 8 Ende

DO WHILE

Ist nur im RUN-Mode einsetzbar.

Definiert eine Programmschleife. Diese Schleife wird sooft wiederholt, solange die Bedingung von WHILE erfüllt ist.

Beispiel:

10 A = 0

20 DO

30 A = A+1

40 PRINT A.

50 WHILE A < 8

60 PRINT "Ende"

RUN

1 2 3 4 5 6 7 8 Ende

END

Beendet ein laufendes BASIC-Programm.

Beispiel:

10 FOR I = 1 TO 5

20 GOSUB 100

30 NEXT I

40 END

100 PRINT I.

110 RETURN

RUN

12345

FOR TO (STEP) NEXT

For-Next-Schleife. STEP ist optional.

Beispiel 1:

10 FOR A = 0 TO 4: REM A wird immer um 1 erhöht.

20 PRINT A,

30 NEXT A

RUN

01234

Beispiel 2:

Es wird die For-Next-Schleife 6 mal durchlaufen und A jedesmal um 2 erhöht (STEP D).

10 B= 0: C= 10: D= 2

20 FOR A= B TO C STEP D

30 PRINT A.

40 NEXT A

RUN

0246810

FOUT (k; f) 1) Frequenzausgabe einschalten FOUT (0) Frequenzausgabe abschalten

Frequenzausgabe: k = Ausgabekanal 0 bzw. 3...8

f = Frequenz 0,05 Hz...10 kHz

Frequ.-Einstell-Abweichung bei $\,$ 1 Hz: $\,$ 0,00002 $\,$ %

bei 100 Hz: 0,02 % bei 10 kHz: 2 %

Das Impuls-/Pausen-Verhältnis ist fest eingestellt und beträgt ca. 1:1.

Als Ausgabekanäle stehen wahlweise folgende Ausgänge zur Verfügung:

k	Ausgabekanal	Pin	
0	AB1	a16	
3*	Erweiterungskarte Port C Bit O	c1 4	A
4*	Erweiterungskarte Port C Bit 1	c16	
5*	Erweiterungskarte Port C Bit 2	c18	Erweiterungs-
6*	Erweiterungskarte Port C Bit 3	€20	karte
7★	Erweiterungskarte Port C Bit 4	a20	1
8*	Erweiterungskarte Port C Bit 5	a18	lack

* Port C ist zuvor mit XBY (7FC7H) = X auf Ausgang zu schalten (X siehe Abschnitt 6.4, Tabelle 9)

Beispiel:

Frequenz von 100 Hz an Ausgang AB1 ausgeben

10 SKALO

20 F = 100

30 FOUT (0, 100)

40 END

GOSUE

Ist nur im RUN-Mode einsetzbar.

Aufruf eines Unterprogramms über die Zeilen-Nr. X GOSUB X

Die Unterprogramme müssen immer mit dem Statement "RETURN" abgeschlossen werden.

Beispiel: siehe Befehl END

GOTO

Programmverzweigung auf die angegebene Zeilen-Nr.

Beispiel:

10 A = 0

20 A = A+1

30 PRINT A,

40 GOTO 20 Run

1,2,3....

¹⁾ ab Firmware-Version 1.3

GW 0.0

setzt AB1-Ausgang auf 0 (Pin a16)

GW 0.1

setzt AB1-Ausgang auf 1 (Pin a16)

ab Firmware-Version 1.5

GW 1.0

Öffnet den Grenzsignal-Schalter 1

(Pin a24+/c24-)

GW 1 1

Schließt den Grenzsignal-Schalter 1

GW 2.0

Öffnet den Grenzsignal-Schalter 2

(Pin a26+/c26-)

GW 2.1

Schließt den Grenzsignal-Schalter 2

IF THEN ELSE

Bedinate Verzweigung.

- Ist nur im RUN-Mode einsetzbar.
- GOTO hinter THEN und ELSE kann entfallen

Beispiel 1: wenn A = 10 wird A = 0 gesetzt,

wenn A <> 10 wird A inkrementiert.

10 IF A = 10 THEN A = 0 ELSE A = A+1

Beispiel 2: wenn A > = 10 wird A = 0 gesetzt.

10 IF A > = 10 THEN A = 0

Beispiel 3: wenn A = 10 dann wird nach 100 verzweigt, wenn A <> 10 dann wird nach 200 verzweigt.

10 IF A=10 THEN 100 ELSE 200

Beispiel 4: wenn A = 10 dann wird A = 0 gesetzt und nach 100 verzweigt.

10 IF A=10 THEN A=0: GOTO 100

IMPCOUNT

Impulszähler-Ausgang AB1 (PIN 16a+/PIN 16c Null) zum Ansteuern externer Impuls-Zähler.

IMPCOUNT führt folgende Funktionen aus:

- a) Echtzeit-Uhr wird gestartet
- b) Der Meßwert x, der zuvor mit PUSH X auf den Stack geladen wurde, wird zur Summe Y aufsummiert und Meßzähler N um 1 erhöht.
- c) Nach jedem Sekunden-Interrupt wird
- aus der Summe Y der Mittelwert Z berechnet (Z = Y/N)
- Der REST der vorherigen Messung zu Z hinzuaddiert.
- Die Summe Y und Meßzähler N auf 0 gesetzt.
- Der Integer-Anteil von Z als Impulsfolge über AB 1 ausgegeben
- Der Rest berechnet: REST = Z INT(Z)

Die Impulsbreite ist fest und beträgt 500 µs.

Beispiel: E1: Q = $0...1000 \text{ m}^3/\text{h} \rightarrow 0...20 \text{ mA}$ Impulswertigkeit $\rightarrow 10 \text{ m3/Imp}$

Es bedeuten:

ME → physkalisches Meßende (m³/s)

DSW→ Digitaler Signal-Wert (0..3600 Digit für 0..20 mA)
W → Impulswertigkeit

.

10 SKALO : rem Skalierung abschalten

20 ME = 1000/3600 : rem ME mit Meßende laden (in m^3/s)

30 DEFINP(1, 0, 0, 0, 0, 0) : rem Signal-Bereich E1 (0...20 mA)

40 ADU(1) : rem E1 messen

50 POP DSW : rem DSW (0...3600 Digit) laden

60 PUSH((DSW/3600)*(ME/W)) : rem Meßwert auf Stack

70 IMPCOUNT : rem Impulsberechnung und Ausgabe

80 GOTO 40 : rem Wiederholung

Achtung: Folgende Punkte müssen beachtet werden:

- IMPCOUNT muß vom Programm mindestens einmal pro Sekunde aufgerufen werden.
- der Mittelwert Z darf den Wert 255 nicht übersteigen.
- IMPCOUNT, PWM und PORT1 beeinflussen den Ausgang AB1 und dürfen deshalb in einem Programm nur alternativ verwendet werden.

IMPCOUNT (k, n: t) 1)

Impulszähler-Ausgang wahlweise über GW1, GW2 oder Erweiterungskarte mit Parameterübergabe:

k: Ausgabekanal 1...8

k	Ausgabekanal	Pin	
1	Grenzwertausgang GW1	a24	<u>.</u>
2	Grenzwertausgang GW2	a26	
3*	Erweiterungskarte Port C Bit O	c14	A
4*	Erweiterungskarte Port C Bit 1	c16	
5*	Erweiterungskarte Port C Bit 2	c18	Erweiterungs-
6*	Erweiterungskarte Port C Bit 3	c20	karte
7 *	Erweiterungskarte Port C Bit 4	a20	
8*	Erweiterungskarte Port C Bit 5	a18	₩

- * Port C ist zuvor mit XBY (7FC7H) = X auf Ausgang zu schalten (siehe Abschnitt 6.4)
- n: Anzahl d. auszug. Imp., Bereich = 0...255 (Integer)
- t: Impulsdauer einstellbar: = 1...255 ms (Integer)

IMPCOUNT (k1, n1; k2, n2: t) 1)

Wie IMPCOUNT (k, n:t); jedoch zusätzlich mit einer 2. Impulsausgabenfolge n2, zeitgleich mit n1 über Kanal K2. Die Impulsdauer ist für n1 und n2 gleich.

Beispiel:

Eingangsimpulse über EB1 zählen, bewerten und über GW1 wieder ausgeben

10 SKALO : rem autom. Skalierung abschalten 20 DEFBINP (8) : rem Imp.zähl-Messung über EB1 einsch.

30 RDIGVAL : rem Anzahl Impulse lesen

40 POP T,IMP : rem Anzahl Imp. der Variablen IMP zuweisen

50 IMP= IMP*0,9853 : rem Impulse bewerten 60 N=(IMP+ REST) : rem Anzahl Imp. berechnen 70 REST= (IMP+ REST)- N : rem Rest berechnen

80 IMPCOUNT(1,N:100) : rem Impulse über GW1 ausgeben 90 GOTO 30 : rem Rücksprung zu Zeile 30

INPUT

Ist nur im RUN-Mode einsetzbar.

Liest eine oder mehrere Variablen von der Tastatur. Jede Variable muß durch ein Komma getrennt werden.

Beispiel 1:

10 INPUT A, B

20 PRINT A+3, B+3

RUN ? 1, 2 4 5

1) ab Firmware-Version 1.3

Es kann Text mit dem Befehl "INPUT" ausgegeben werden

Der Text wird in " " gesetzt. Werden Text und 1. Variable durch Komma getrennt, so erfolgt die Eingabe hinter dem Text (siehe Beispiel 2). Ohne Komma wird die Eingabe in nächste Zeile geschrieben (siehe Beispiel 3).

Beispiel 2:

10 INPUT "Der Wert X = ", X

20 PRINT X/10

RUN

Der Wert X = 100

10

Beispiel 3:

10 INPUT "Der Wert X = " X

20 PRINT X/10

RUN

Der Wert X =

? 100

10

Es können auch STRINGs mit INPUT eingegeben werden

Beispiel 4:

10 STRING 100, 10

20 INPUT "Name und Alter eingeben: ", \$(1),A

30 PRINT "Hallo ",\$(1),"du bist ",A,"Jahre alt"

RIIN

Name und Alter eingeben: Fritz

? 16

Hallo Fritz du bist 16 Jahre alt

LD@

Mit LD@ X können Floating-Point-Variablen von einer RAM-Speicheradresse in den Argument-Stack und von dort mit POP Yin eine BASIC-Variable Ygeladen werden. X steht für die RAM-Adresse. Jeder Floating-Point-Wert belegt 6 Byte im RAM (z.B bei Adresse 6005H: 6005H, 6004H, 6003H, 6002H, 60001H, 6000H)

Siehe auch Befehl ST@ !!

Beispiel:

10 A = 5.1234

12 PRINT A

20 PUSH A

30 ST@ 6100H

40 LD@ 6100H: POP Y

50 PRINT Y

RUN

5.1234

5.1234

LEDO

Rote LED auf Frontplatte geht aus, Fehler-Signal-Ausgang (PIN 22a) geht auf Low.

LED1

Rote LED auf Frontplatte geht an, Fehler-Signal-Ausgang (PIN 22a) geht auf High.

LKONST

Laden der Parameter-Konstanten (x1..xn) aus einer Parameter-Datei.

Dieser Befehl wird nur bei P-Programmen mit Parameter-Dateien verwendet. Die in der Parameter-Datei abgelegten Konstanten werden in den Stack geladen und mit POP x1..xn den Variablen x1..xn zugewiesen.

ON GOTO

Variablenabhängige Programmverzweigung.

```
Beispiel:
```

10 FOR I = 0 TO 3

20 ON I GOTO 30, 40, 50, 60

30 PRINT "Hans", :GOTO 70

40 PRINT "Peter", :GOTO 70 50 PRINT "Elke", :GOTO 70

60 PRINT "Agathe" :GOTO 70

70 NEXT I

RIIN

Hans Peter Elke Agathe

ON GOSUB

Variablenabhängiger Unterprogramm-Aufruf.

Beispiel:

10 FOR I = 0 TO 3

20 ON I GOSUB 100, 200, 300, 400

30 NEXT I

100 PRINT "Subrout.1": RETURN

200 PRINT "Subrout.2": RETURN

300 PRINT "Subrout.3": RETURN 400 PRINT "Subrout.4": RETURN

RUN

Subrout.1

Subrout.2

Subrout.3

Subrout.4

ONERR

Ist nur im RUN-Mode einsetzbar.

Wenn während der Programmausführung Fehlermeldungen auftreten, die durch ARITH. OVERFLOW, ARITH. UNDERFLOW, DIVIDE BY ZERO u. BAD ARGUMENT, hervorgerufen wurden, so verzweigt das Programm auf die mit ONERR X angegebene Zeilen-Nr. (X = Zeilen-Nr.) In der externen Speicherzelle 257 ist der Fehlercode abgelegt. Dieser kann mit XBY(257) gelesen und im Programm behandelt werden.

Fehler-Code = 10 → Divide by Zero

Fehler-Code = 20 → Arith. Overflow

Fehler-Code = 30 → Arith. Underflow

Fehler-Code = 40 → Bad Argument

Beispiel:

10 ONERR 100

20 FOR I=5 TO 0 STEP-1

30 A = 10/I

40 PRINT I

50 NEXT I

60 END

100 IF XBY(257)=10 THEN 110 ELSE 120

110 PRINT "Fehler! Division durch Null" : GOTO 60

120 PRINT "Fehler-Code 20, 30 oder 40!" : GOTO 60

ONEX1

Ist nur im RUN-Mode einsetzbar.

ONEX1 X gibt den externen Interrupt des Binär-Eingangs EB2 frei. Beim Auftreten eines Interrupts (EB2 wird auf low-Pegel gezogen), springt das Programm auf die unter X angegebene Zeilen-Nr. der Interrupt-Unterroutine. In dieser Routine muß der Interrupt behandelt und mit dem Befehl RETI abgeschlossen werden.

ONTIME

Ist nur im RUN-Mode einsetzbar.

ONTIME T,X gibt den Timerinterrupt frei. Mit T wird die Zeit in Sekunden (1... 65535) übergeben, bei der ein Interrupt ausgelöst werden soll; mit X wird die Zeilen-Nr. der Interrupt-Routine übergeben.

Die Interrupt-Routine ist mit dem Befehl "RETI" zu beenden.

ONTIME arbeitet zusammen mit dem Befehl CLOCK1 und der BASIC-Variablen TIME. CLOCK1 aktiviert den internen Sekunden-Zähler.

TIME wird alle 5 ms mit 0.005 inkrementiert. Nach Aufruf von ONTIME T,X , wird jedesmal, wenn TIME inkrementiert wird ein Vergleich zwischen TIME und dem übergenenen Wert T ausgeführt. Wenn TIME \geq T (nur Integer-Anteil von TIME) dann wird die Interrupt-Routine (Zeilen-Nr. X) aufgerufen. Bei TIME \geq 65536 wird TIME wieder auf 0 gesetzt.

Beispiel:

10 TIME=0: CLOCK1: ONTIME 2,100:D0

20 WHILE TIME < 8: END

100 PRINT "Timer-Interrupt = ".TIME." Sekunde"

110 ONTIME TIME+2,100

120 RETI Run

Timer-Interrupt = 2.045 Sekunde Timer-Interrupt = 4.045 Sekunde Timer-Interrupt = 6.045 Sekunde Timer-Interrupt = 8.045 Sekunde

PHO.

Der Befehl PH0. printet numerische Variablen im Hexadezimal-Format (00H..0FFH). Es werden 2 Stellen und die Hex.-Kennzeichnung H ausgegeben. Ist die Variable eine Floating-Point-Variable, wird nur der Integer-Anteil ausgegeben.

PH1.

Der Befehl PH1. ist gleich dem PH0.-Befehl mit der Ausnahme, daß die Ausgabe mit 4 Hexadezimal-Stellen erfolgt (0000H...0FFFFH).

Beispiele:

 PHO. 2*2
 PH1. 2*2
 PRINT 99H

 04H
 0004H
 153

 PHO.1000
 PH1.1000
 PH0. PI

 3E8H
 03E8H
 03H

PRINT

Der PRINT-Befehl gibt Variablen und Texte über die Schnittstelle aus und printet sie auf Bildschirm.

PRINT kann auch in gekürzter Form wie P. oder? geschrieben werden. Sollen mehrere Variable in einer Zeile ausgegeben werden, so sind die Variablen durch Kommas zu trennen.

Beispiel 1:

PRINT A,B,C

3 7 4

Beispiel 2:

PRINT " A = ",A

A = 3

Mit speziellen Format-Befehlen lassen sich verschiedene Print-Ausgabe-Formate erzielen:

 TAB(x) → tabellarische Ausgabe (springt auf Position x).

Beispiel

PRINT TAB(5), "Hans", TAB(13), "Peter"

Hans Pete

- SPC(x) \rightarrow x Leerzeichen zwischen den Ausgaben

Beispiel

PRINT "Hans", SPC(5), "Peter"

Hans Peter

- CR

→ Nach Befehls-Ausführung wird ein CR, aber kein LF durchgeführt d.h. es wird immer in die gleiche Zeile geschrieben. Beispiel:

10 FOR I=1 TO 100

20 PRINT I,CR,

30 NEXT I

- USING(Fx)

Angabe von speziellen Floating-Point-Formaten:

USING(Fx)

Floating-Point-Ausgabe mit x-Stellen im E-Format

 $(x \rightarrow 3 \dots 8)$.

Beispiel:

PRINT USING(F3),1,2,3

1.00 E 0 2.00 E 0 3.00 E 0

USING(F0):

Floating-Point-Ausgabe im E-Format. Es werden nur soviele Stellen, wie notwendig sind, ausgegeben (max.8 Stellen)

Beispiel:

PRINT USING(F0),1.23, 23235.34, 0.0123

1.23 E 0 2.3235,34 E 4 1.23 E-2

USING(##.###):

Floating-Point-Ausgabe mit angegebenen Stellen ohne E-Format

Beispiel:

PRINT USING(###.###),11,-12,13

11.000 -12.000 13.000

USING(0):

Standard-Ausgabe-Format. Liegt der Wert zwischen \pm 99999999 und \pm 0.1, erfolgt die Ausgabe im gleichen Format, liegt der Wert außerhalb der angegeben Grenzen erfolgt die Ausgabe im USING(F0)-Format

PUSH

PUSH X speichert die Konstante oder Variablen x auf den Argument-Stack. Sollen mehrere Werte abgespeichert werden, sind diese durch Komma zu trennen.

Beispiel:

PUSH A,B,1.234,SQR(25)

POP

POP X liest Floating-Point-Werte vom Stack in die Variable x. Der Wert der zuletzt mit PUSH auf den Stack geladen wurde, wird zuerst mit POP gelesen.

Beispiel:

10 A=10: B=20

20 PRINT A,B

30 PUSH A,B: POP A,B

40 PRINT A,B

RUN

10 20 20 10

PWM X,Y,N

Impulsbreiten-Modulator-Ausgang AB1 (PIN 16a).

Mit diesem Befehl können n Impulse definierter Breite und Pause über PIN 16a ausgegeben werden. X,Y müssen Integer-Werte sein und zwischen 25 und 65535 liegen. X bestimmt die Impulspause (X*1.085 μ s) und Y die Impulsbreite (Y*1.085 μ s).

N (Integer Wert) steht für Anzahl der Impulse und kann zwischen 1 und 65535 liegen.

Beispiel: PWM 2000,1000,20000

Es werden 20000 Impulse von 1.085 ms Breite und 2.17 ms Pause über PIN 16a ausgegeben.

Achtung: PWM, IMPCOUNT u. PORT1 beeinflussen den Ausgang AB1 und dürfen nur alternativ verwendet werden.

RDBINP

Lesen der 4 Binär-Eingänge EB1,..,EB4 oder ENI und Ablegen als BYTE (BIT 0..3) auf Stack.

Mit POP X wird dieses BYTE einer Variablen x zugewiesen. Durch eine AND-Verknüpfung können die Eingänge EB1..EB4 bzw ENI selektiert werden (siehe Abschnitt 6.1.2).

RDIGVAL

Lesen des digitalen Meßwertes des an den Binär-Eingängen anliegenden Meßsignals.

Mit DEFBINP(y) bzw.DEFBINP(y,z) müssen zuvor die Binäreingänge für die gewünschte Meßfunktion definiert werden (siehe BASIC-Befehle DEFBINP(1..8);

Tabelle 5 Funktionsübersicht;

Tabelle 2 BASIC-Befehle)

Beispiel:

Impulse zählen über ENI-Eingang:

10 DEFBINP(2) : rem Freigabe Impulszähler EB4/ENI 20 RDIGVAL : rem Iese eingelaufene Anzahl Impulse 25 POP IMP : rem weise Anzahl der Variabl. IMP zu 30 PRINT "Impuls-Zahl = ",IMP: rem drucke Anzahl aus

40 GOTO 20

READ

Ist nur im RUN-Mode einsetzbar.

READ liest eine Date aus einem mit DATA definierten Datenfeld

Beispiel:

10 FOR A=1 TO 5 20 READ X: PRINT X, 30 NEXT A

40 DATA 2,5,3,7,8,6

RUN

253782

REM

REM ist die Kurzform von REMark. REM erlaubt dem Anwender Kommentare ins BASIC-Programm einzufügen. Es können ganze Kommentarzeilen geschrieben oder Kommentare an eine BASIC-Zeile angehängt werden. Kommentarzeilen ohne vorangestellte Zeilen-Nr. werden beim Transferieren des BASIC-Programms nicht im TZA 4 abgespeichert und belegen daher keinen TZA 4-Speicherplatz.

Beispiel:

REM Bei diesem Programm werden diese Zeilen nicht im REM TZA 4 abgespeichert. Hier kann eine ausführliche

REM Programmbeschreibung folgen, die keinen

REM Speicherplatz im TZA 4 belegt.

10 REM Das ist ein Testprogramm REMTEST
20 FOR I=1 TO 1000: REM Schleife 1..1000
30 PRINT USING(####.#),I,: REM Ausgabe mit 5 Stellen
40 NEXT I

RETI

Ist nur im RUN-Mode einsetzbar.

Der RETI-Befehl wird gebraucht zum Abschließen einer Interrupt-Routine, z.B. beim Verlassen der ONTIME- bzw. ONEX1-Interrupt-Routine.

RESTORE

Ist nur im RUN-Mode einsetzbar.
RESTORE setzt den READ-DATEN-POINTER zurück.

Beispiel:

10 FOR A=1 TO 5

20 READ X: PRINT X.

30 NEXT A

40 RESTORE

50 READ X: PRINT X

60 DATA 2,5,3,7,8,6

RUN

253782

RROM

Mit RROM X kann aus einem laufenden BASIC-Programm heraus ein anderes BASIC-Programm, das im EPROM 2 abgespeichert ist, gestartet werden. Dies ist aber nur bei #-Programmen und User-Programmen möglich. Mit X wird die Programm-Platz-Nr. im EPROM 2 angegeben

Hinweis: Beim Suchen eines bestimmten Programms kann mit dem Befehl "ROM X" ein Programm angewählt und mit dem Befehl "LIST" aufgelistet werden

SKAL1

Skalierung der Meßwerte wird eingeschaltet. Bei P-Programmen werden bei der AD-Wandlung vom "ADU", mit Hilfe der Parameter-Datei-Angaben für Anfangs- und Endwerte, die elektrischen Eingangssignale in physikalische Meßwerte und die physikalischen Ausgabe-Werte in elektrische Ausgangssignale umgerechnet.

SKAL0

Skalierung wird abgeschaltet.

Analoge Einganssignale werden vom "ADU" in digitale Eingangssignale (-3600...0...+3600 Digit) umgewandelt

Ausnahme: Bei PT100 direkt (Signalcode 8 und 9) erfolgt die Umrechnung immer in °C (siehe Tabelle 3).

ST@

Mit ST@ Y kann eine Floating-Point-Variable X die zuvor mit PUSH X im Argument-Stack gespeichert wurde, auf eine RAM-Speicher-Adresse Y geschoben werden. Jeder Floating-Point-Wert belegt 6 Byte im RAM

(z.B Y=6005H : 6005H, 6004H, 6003H, 6002H, 6001H, 6000H); (siehe auch Befehl LD@)

Beispiel:

10 A=5.1234

12 PRINT A

20 PUSH A

30 ST@ 6100H

40 LD@ 6100H: POP X

50 PRINT X

RUN

5.1234

5.1234

STOP

Ist nur im RUN-Mode einsetzbar.

Der STOP-Befehl erlaubt dem Anwender ein BASIC-Programm an einem bestimmten Punkt im Programm anzuhalten. Es können dann im Kommando-Mode die Variablen mit PRINT ausgelesen, verändert und das Programm durch Eingabe von CONT mit der Zeilen-Nr. nach dem STOP fortgesetzt werden.

STRING

Mit STRING X,Y wird RAM-Speicher für Strings reserviert. Mit X wird der gesamte im RAM zu reservierende Speicher in Bytes für alle Strings angegeben.

Mit Y wird die Anzahl der Charakters pro String angegeben. Die Formel zum Berechnen des gesamten Speicherbedarfs X lautet:

$$X = ((nChr+1) * nStr)+1$$

Es bedeuten: nChr → Anzahl der Characters/String nStr → Anzahl der Strings

STRING 100,10 reserviert 100 Byte RAM-Speicher für 9 String-Variablen von \$(0)..\$(8) mit je 10 Charakters

Beispiel:

10 STRING 100,10

20 LET \$(0)="Hallo"

30 LET \$(1)="Franz"

40 PRINT \$(0),\$(1)

RUN

Hallo Franz

WDOG

Programm-Überwachung in einem laufenden BASIC52-Programm.

Durch den Befehl "WDOG" wird die Programmüberwachung aktiviert.

Gelbe LED auf Frontplatte geht aus. Der Befehl WDOG muß mindestens einmal pro Sekunde aufgerufen werden. Bei Zeitüberschreitung wird ein Hardware-Reset ausgeführt, der BASIC-RAM-Speicher gelöscht und das in der Autostart-Liste eingetragene Programm im EPROM 2 gestartet. Mit CTRL-C wird das BASIC52-Programm abgebrochen und die Watchdog in den inaktiven Zustand geschaltet.

5.3 Arithmetische BASIC 52-Operatoren

+	Addition.	z.B. PRINT 3 + 5	\rightarrow	8			
/	Division	z.B. PRINT 100/5	\rightarrow	20			
**	Potenzierung	z.B. PRINT 2**3	\rightarrow	8			
*	Multiplikation	z.B. PRINT 3*3	\rightarrow	9			
_	Subtraktion	z.B. PRINT 9-6	→	3			
=; $<$; $>$; $<$ $=$; $>$ = Vergleichs-Operatoren							

5.4 Logische Operatoren

.AND.Logisches UND	z.B. PRINT 3.AND.2 →	2
.OR. Logisches ODER	z.B. PRINT 1.OR.4 →	5
.XOR.Logisches Exclusiv-OR	z.B. PRINT 7.XOR.6 →	1

5.5 String-Funktionen

ASC(x

Liefert den Integer-Wert eines ASCII-Charakters, z.B. PRINT ASC(A) \rightarrow 65

10 \$(1)= "Das ist ein String"

20 PRINT \$(1)

30 PRINT ASC(\$(1),1)

RUN

Das ist ein String

68

CHR(x)

Konvertiert eine ASCII-Nummer in einen ASCII-Charakter, z.B. PRINT CHR(65) $\,$ A

5 String 19,17

10 \$(1)="TZA 4-Messrechner"

20 FOR I=1 TO 17: PRINT CHR(\$(1),I),: NEXT I

30 PRINT

40 FOR I=17 TO 1 STEP-1

50 PRINT CHR(\$(1),1).

60 NEXT I

RUN

TZA 4-Messrechner

renbcersseM-4AZT

5.6 Arithmetische Funktionen

ABS(x)

Absolutwert-Bildung z.B. PRINT ABS(-5) $\rightarrow 5$

NOT(x)

Komplement-Bildung zum z.B. PRINT

Wert 65535 oder 0FFFFH NOT(65000) → 535

INT(x)

Integer-Wert-Bildung z.B. PRINT INT(3.7) \rightarrow 3

SGN(x)

Umwandlung von x in 1, 0, -1

wenn x > 0 dann SGN(x) \rightarrow 1

wenn x = 0 dann $SGN(x) \rightarrow 0$

wenn x < 0 dann SGN(x) $\rightarrow -1$

z.B. PRINT SGN(52) \rightarrow 1

SQR(x)

Radizierung z.B. PRINT $SQR(9) \rightarrow 3$

RND

Zufallsgenerator. Es wird ein Werte zwischen 0...1 geliefert. Der Genarator generiert 65536 verschiedene Werte. Diese werden berechnet aus 0/65535...65535/65535

PΙ

Pi ist eine Konstante und beträgt 3.1415926

5.7 Logarithmische Funktionen

LOG(x)

Natürlicher Logarithmus,

z.B. PRINT LOG(12) → 2.484906

z.B. PRINT LOG(EXP(1)) \rightarrow 1

EXP(x)

Exponentialfunktion zur Basis "e" (2.7182818)

z.B. PRINT EXP(1) \rightarrow 2.7182818

z.B. PRINT EXP(LOG(2)) \rightarrow 2

5.8 Trigonometrische Funktionen

SIN(x)

Liefert den Sinus von x in Radiant

Beispiel: Winkel W=60°

PRINT SIN(60*PI/180) → 0.8660254

COS(x)

Liefert den Cosinus von x in Radiant

Beispiel: Winkel W=45°

PRINT COS(45*PI/180) → 0.7071067

TAN(x)

Liefert den Tangens von x in Radiant

Beispiel: Winkel W=30°

PRINT TAN(30*PI/180) → 0.57735027

ATN(x)

Liefert den Arkus-Tangens von x in Radiant

Beispiel: Winkel W=180°

PRINT ATN(180*PI/180) → 1.2626272

5.9 Spezial-Funktionen

DBY(x)

Direkter Zugriff auf ein BYTE im internen μ P-RAM. Die Adresse x muß zwischen 0..255 liegen.

Reisniel:

PRINT DBY(9) → Liest Argument-Stack-Pointer
DBY(47H)=0 → Setzt den ms-Zähler auf 0

XBY(x)

Direkter Zugriff auf ein BYTE im externen RAM bzw. Hardware-Adressen.

Beispiel:

PRINT XBY(6D10H) → Liest den Sekunden-Speicher XBY(6D11H)=0 → Setzt den Minuten-Speicher auf 0

CBY(x)

Direkter Zugriff auf ein BYTE des Programm-Speichers

Beispiel:

PRINT CBY(7FBFH) → Checksumme des Betriebs-

System-EPROM 1 wird

ausgelesen

GET

Es wird getestet, ob ein Zeichen von der Tastatur über die Schnittstelle vom TZA 4 empfangen wurde.

Wenn ja, liefert GET die ASCII-Nr. des Zeichens. Wenn nein, liefert GET 0 zurück.

Beispiel:

10 A=GET

20 IF A <> 0 THEN PRINT A

30 GOTO 10

RUN

65 → es wurde die Taste "A" gedrückt
49 → es wurde die Taste "1" gedrückt
24 → es wurde die Taste "CTRL-X" gedrückt
50 → es wurde die Taste "2" gedrückt

TIME

TIME ist eine vordefinierte Variable in BASIC 52.

Sie wird durch den Interrupt der Real-Time-Clock alle 5 ms um 0.005 erhöht. Mit CLOCK1 wird der Interrupt der Real-Time-Clock freigegeben und mit CLOCK0 wieder gesperrt.

Beim Überschreiben von TIME wird nur der Integer-Anteil von TIME überschrieben (nur die Sekunden). Der Millisekunden Speicher bleibt davon unberührt. Dieser Speicher kann seperat über die Adresse 71 im internen μ P-RAM gelesen oder beschrieben werden.

Beispiel:

10 CLOCKO → Stoppt den Real-Time-Clock

20 PRINT TIME 3.415 → Zufallswert

30 TIME = 0 → TIME wird auf 0 gesetzt 40 PRINT TIME 0.415 → Nur die Sekunden sind gelöscht 50 DBY(71)=0 → Millisekunden werden auch gelöscht

60 PRINT TIME 0 → TIME ist exakt auf 0

5.10 BASIC-Befehle, die nur alternativ zu verwenden sind

Die BASIC-Befehle DEFBINP(y) bzw. DEFBINP(y,z), FOUT (), IMPCOUNT, DATETIME(1), CLOCK1, PWM und ONEXI verwenden zum Teil die gleichen internen µP-Zähler bzw. die gleichen externen Interrupt-Eingänge.

Tabelle 2 gibt an, welche der oben genannten BASIC-Befehle nur wahlweise zusammen in einem BASIC-Programm eingesetzt werden dürfen.

Basic-Befehle	DEFBINP(0)	DEFBINP(1,x)	DEFBINP(2)	DEFBINP(3)	DEFBINP(4)	DEFBINP(5)	DEFBINP(6)	DEFBINP(7)	DEFBINP(8)	DATETIME(1)	IMPCOUNT	CLOCK1	PWM	ONEX1	ONTIME	FOUT()
DEFBINP(0)																
DEFBINP(1,x)			*	*	*	*	*	*	*	*	*	*	*		*	*
DEFBINP(2)		*		*	*	*	*	*	*	*	*	*		*	*	
DEFBINP(3)		*	*		*	*	*	*	*				*	*		*
DEFBINP(4)		*	*	*		*	*	*	*	*	*	*	!		*	*
DEFBINP(5)		*	*	*	*		*	*	*				*	*		*
DEFBINP(6)		*	*	*	*	*		*	*	*	*	*	!		*	*
DEFBINP(7)		*	*	*	*	*	*		*				*	*		*
DEFBINP(8)		*	*	*	*	*	*	*					!	*	*	*
DATETIME(1)		*	*		*		*					*	!		*	
IMPCOUNT		*	*		*		*					*	!			
CLOCK1		*	*		*		*			*	*		!			
PWM	Ī	*		*	!	*	!	*	!	!	!	!				*
ONEX1			*	*		*		*	*		-					
ONTIME		*	*		*		*		*	*	*					
FOUT()		*		*	*	*	*	*	*				*			

Tabelle 2

- * Die BASIC-Befehle der X/Y-Koordinate schließen sich in einem BASIC-Programm gegenseitig aus.
- Die PWM Impuls-Breite/Pause kann bis zu 130 μs verfälscht werden

(Interrupt-Bearbeitungszeit für DEFBINP(4 bzw. 6) oder Zeitinterrupt bei IMPCOUNT, CLOCK1 oder DATETIME).

5.11 Kurzreferenz-Liste aller BASIC-Befehle

Kommandos	Funktion	Beispiel
ADU	AD-Wandlung der Analog-Eing. E1 E6	ADU(1,2,5,6)
CLEAR	löscht Variable, Interr. u. Strings	CLEAR
CLEARI	löscht Interrupts	CLEARI
CLEARS	löscht Stack	CLEARS
CLOCK0	stoppt REAL TIME CLOCK	CLOCK0
CLOCK1	startet REAL TIME CLOCK	CLOCK1
CONT	Programm nach STOP o. Ctrl-C fortetzen	CONT
DATA	spezifiziert ein Datenfeld	DATA 10,24
DATETIME	Echtzeituhr (1)start; (0) stopp; (R)lesen	DATETIME(R)
DAU(I) bzw. DAU(U)	DA-Wandlung auf Ausgang A1 (20mA) bzw. Ausgang A2 (10 V)	DAU(I) bzw. DAU(U)
DEFBINP(1,y)	Frequenzmessung über Binär-Eing. y= Zeitbasis in s	DEFBINP(1,10.222)
DEFBINP(2)	Impulszähler über Eingang EB4/ENI	DEFBINP(2)
DEFBINP(3)	Impulszähler über Eingang EB3	DEFBINP(3)
DEFBINP(4)	Zeitmessung (ImpBreite) Eing. EB1	DEFBINP(4)
DEFBINP(5)	Zeitmessung (ImpBreite) Eing. EB2	DEFBINP(5)
DEFBINP(6)	Periodendauermessung über Eing. EB1	DEFBINP(6)
DEFBINP(7)	Periodendauermessung über Eing. EB2	DEFBINP(7)
DEFBINP(8)	Periodendauer- und Impulsmessung	DEFBINP(8)
DEFBINP(0)	keine Funktion der Binär-Eing.	DEFBINP(0)
DEFINP	Definiere Analog-EingMeß-Bereiche	DEFINP(1,1,0,3,6,0)
DIM	reserviert Speicher für Array	DIM A(20)
DO	Schleife DO WHILE oder DO UNTIL	DO
END	Programm-Ende	END
FOR TO (STEP)	For-Next-Schleife	FOR A=1 TO 6 STEP 2
FOUT(k, f)	Frequenzausgabe (k=Kanal, f=Frequenz)	- · · · ·
GOSUB	Aufruf einer Unterroutine	FOUT(1, 100)
GOTO	Programmverzweigung	GOSUB 1000
		GOTO 200
GW0,0	AB1-Ausgang wird auf Low-Pegel gesetzt	GW0,0
GW0,1	AB1-Ausgang wird auf High-Pegel gesetzt	GW0,1
GW1,0	Grenzwert 1 wird geöffnet	GW1,0
GW1,1	Grenzwert 1 wird geschlossen	GW1,1
GW2,0	Grenzwert 2 wird geöffnet	GW2,0
GW2,1	Grenzwert 2 wird geschlossen	GW2,1
IF THEN (ELSE)		IF A>B THEN A=0
IMPCOUNT	Impuls-Zählausgang bedienen	IMPCOUNT
IMPCOUNT(k, n:t)	Impuls-Zählausgang mit Parameter-Übergabe	IMPCOUNT(1, 25:100)
IMPCOUNT(k1, n1;k2, n2:t)		IMPCOUNT(1,25;2,20:100)
INPUT	Wert- oder String-Eingabe	INPUT "Peter = ",X
LD@	fl.Point-Wert von RAM-Adr. auf Stack.	LD@ 1000
LED0	Rote LED=aus; Fehler Sign-Ausg.= Low	LED0
LED1	Rote LED=an; Fehler Sign-Ausg.= High	LED1
LET	bezeichnet String od Variab.(optinal)	LET A=10
LIST	listet Programm-Code auf	LIST 1-1000
LKONST	ladet Konst.v. Param.Dat. auf Stack	LKONST
NEW	löscht den Programm-Speicher	NEW
NEXT	FOR-NEXT-Schleife	NEXT A
ON GOTO	Variablenabhängige Verzweigung	ON A GOTO 5,20
ON GOSUB	Variablenabhängige UnterprogrVerzw.	ON A GOSUB 5,20
ONERR	Programm wird nach Fehler fortgesetzt	ONERR 1000
ONEX1	springe auf Zeile x wenn Interr. INT1	ONEX1 1000
ONTIME	Interrupt wenn TIME > ONTIME, dann GOTO Zeilen-Nr.	ONTIME 10,1000
PH0.	Wert-Ausgabe in HexFormat (BYTE)	PH0.A
PH1.	Wert-Ausgabe in HexFormat (WORD)	PH1.A
POP	FlPoint- oder IntWert von Stack	POP A,B,C
PRINT	Ausgabe von Variablen u.Strings	PRINT "Wert=",A
PUSH	FIPoint- oder IntWert auf Stack	PUSH A,B,C
PWM	Puls-Breiten-Modulation	PWM 50,50,100
RAM	schaltet auf RAM-Mode um	RAM
	liest Binär-EingPegel auf Stack	RDBINP
RDBINP	ag. / ogo! du! Otdok	
RDBINP RDIGVAI	liest den digitalen Meßwert der mit DEERIND(v) defin	RDIGVAI
RDBINP RDIGVAL	liest den digitalen Meßwert der mit DEFBINP(x) defin.	RDIGVAL
RDIGVAL	Meßfunktion der Binär-Eingänge auf den Stack	
RDIGVAL READ	Meßfunktion der Binär-Eingänge auf den Stack liest eine Date aus Datenfeld	READ I
	Meßfunktion der Binär-Eingänge auf den Stack	

Kommandos	Funktion	Beispiel
RETI	zurück von BASIC-Interrupt-Routine	RETI
RETURN	Unterroutine beenden	RETURN
RROM	Startet ein anderes Programm.	RROM 3
RUN	Programm starten	
SKAL1		RUN
SKALO	Schaltet Skalierung ein	SKAL1
	Schaltet Skalierung aus	SKAL0
ST@	Floating Point-Wert vom Stack auf RAM-Adr.	ST@ 1000
STOP	stoppt die Programmausführung	STOP
STRING	reserviert RAM für STRINGS	STRING 50,10
UNTIL	Test ob Bedingung erfüllt	UNTIL $A = 10$
WDOG	Setzt Programmüberwachung	WDOG
WHILE	wiederhole bis Bedingung erfüllt	WHILE A <> 10
XFER	kopiert Programm vom EPROM ins RAM	XFER
Operanden	Tunkking.	· · · · · · · · · · · · · · · · · ·
Operanden	Funktion	Beispiel
=,>,>=<,<=,<>	Vergleichsoperatoren	IF A \leq = B THEN 25
+	Addition	23+55
_	Subtraktion	55-23
/	Division	65/5
*	Multiplikation	4*8
**	Potenzierung	4**8
.AND.	logisches UND	10.AND.3
.OR.	logisches ODER	
.XOR.	_	X.OR.Y
	logisches Exclusiv-ODER	A.XOR.B
ABS()	absoluter Wert	ABS(-3)
ASC()	liefert den ASCII-Code eines Charakt.	ASC(A)
ATN()	liefert den Arc-Tangens von x	ATN(1)
CHR()	wandelt ASCII-Code in Charakter	CHR(65)
COS()	liefert den Cosinus von x	COS(0)
EXP()	Exponentialfunktion zur Basis "e"	EXP(10)
INT()	Integer-Wandlung	INT(3.2)
LOG()	natürlicher Logarithmus	LOG(10)
NOT()	Komplement	NOT(0)
RND	Zufallsgenerator	RND
SGN()	Vorzeichen	
SQR()	Quadrat-Wurzel	SGN(-5)
	• • • • • • • • • • • • • • • • • • • •	SQR(25)
SIN()	liefert den Sinus von x	SIN(3.14)
TAN()	liefert den Tangens von x	TAN(0.707)
Spez.Print-Formate	Funktion	Beispiel
TAB()	Stellt Cursor auf angegebene Position PRINT TAB(5),"X"	
SPC()	Printet Blanks	PRINT A,SPC(4),B
CR	Printet CR ohne Line-Feed	PRINT A,CR,
USING()	Printet im angegebenen Format	PRINT USING(###.##),A
oom(d()	ExpFormat mit 5 Stellen	
		PRINT USING(F5),A
	Integer bzw.ExpFormat	PRINT USING(0),A
Spezialfunktionen	Funktion	Beispiel
CBY()	lesen eines EPROM-Bytes	PH0.CBY(8100H)
DBY()	lesen/schreiben eines µP-Daten-Bytes	PRINT DBY(25)
XBY()	lesen/schreiben eines RAM-Daten-Bytes	PRINT XBY(1200H)
GET	lesen von Tastatur	PRINT GET
IE.	lesen µP-IE-Register	
IP.	•	PRINT IE
PCON	lesen µP-IP-Register	PRINT IP
	lesen µP-PCON-Register	PRINT PCON
TCON	lesen µP-TCON-Register	PRINT TOON
TIME	Real-Time-Clocks	PRINT TIME
TIMER0	μP-Timers0	TIMER0=0
TIMER1	μP-Timers1	TIMER1=0
TIMER2	μP-Timers2	TIMER2=0
MTOP	Max. RAM-BASIC-Speicher	PRINT MTOP
LEN	BASIC-Programmlänge	PRINT LEN
FREE	Freier BASIC-Speicher	PRINT FREE
PI	PI=3.1415926	A=PI
PORT1	8052-Prozessor-Port 1	
. 51111	3302 1 10263301-1 Off 1	PRINT PORT1

6 Konfigurier- und Programmier-Hinweise

6.1 Meß-Funktionen und Signal-Bereiche der Ein- und Ausgänge

6.1.1 Analog-Eingänge E1..E6

Bei der AD-Wandlung werden durch den Befehl ADU(..) die Eingangssignale der Analog-Eingänge in digitale Signalwerte umgewandelt.

Die Umwandlung ist abhängig von dem durch den Befehl "DEFINP(x,x,x,x,x,x)" gewählten Code der Eingangsgrößen und Meßbereiche nach folgender Tabelle:

Code Nr.	Eingangs- Signale	Digital- Signal-Werte	Digital- Analog- Faktor KD ¹)
0 1 2 3 4 5 * 6 * 7 * ! 8 *	Eingang inaktiv - 200 + 20 mA -2000 + 200 mV - 10 + 1 V - 100 + 10 V 0 100 Ω 0 1000 Ω 0 1000 Ω -200 + 200 °C Pt100 -200 + 800 °C Pt100	-3600+3600 -3600+3600 -3600+3600 0+3600 0+3600 0+3600 -200+200 -200+800	180 18 3600 3600 18 18

Tabelle 3 Code-Tabelle der Analog-Eingänge

Hinweise

Zu Tabelle 3

 Bei mehreren Eingängen darf dieser Code nur alternativ vergeben werden (Speisestrom Ik für alle Eingänge gemeinsam).

Beispiel: DEFINP(1,1,3,8,8,8) → erlaubt DEFINP(1,1,3,5,6,8) → nicht erlaubt!

! Bei Eingängen, die für PT100 definiert wurden (Code 8 u. 9) wird die Skalierung in °C bereits bei der AD-Wandlung vorgenommen.

Bei der 2- bzw. 3-Leiterschaltung kann auf Wunsch ein fester Leitungsabgleich mit eingerechnet werden $(R_{Abql}=10~\Omega)$.

Beispiel:

50 PUSH 5:@ST69FAH :REM 5Ω Leitungswiderstand werden eingerechnet

Zu Tabelle 4

- ! Löt-Brücken: 409,413 bei anal.Eing.E5; 410,414 bei bin.Eing.EB1 u.EB2
- ! Löt-Brücken: 415,419 bei anal.Eing.E6; 416,420 bei bin.Eing.EB3 u.EB4
- () keine Brücken;
- # mehrere Temp.-Eingänge (Pt100) oder mehrere Eingänge von Widerstandsgebern nur innerhalb eines Bereiches möglich.

Eing.	E1	A E2	nalog-E	Eingäng E4	e E5!	E6!	lk	EB1	Binär-E	ingäng EB3	e EB4	ENI
Bereich								!	!	!	!	
+-20 mA	401	403	405	407	411	417	()	()	()	()	()	()
+-200mV	()	()	()	()	()	()	()	()	()	()	()	()
+- 1 V	()	()	()	()	()	()	()	()	()	()	()	()
+-10 V	402	404	406	408	412	418	()	()	()	()	()	()
0 100 Ω# 0 1000 Ω# 010000 Ω#	()	()	()	()	()	()	72 81 82	()	()	()	()	()
-200+200°C # -200+800°C #	()	()	()	()	()	()	61 71	()	()	()	()	()
wenn unbenutzt	401	403	405	407	411	417	71	()	()	()	()	()

Tabelle 4 Steckbrücken bei analogen Eingangsgrößen

¹⁾ KD ist der Digital-Analog-Faktor zum Umrechnen von elektrischen in physikalische Meßwerte nach Skalierungsformel

6.1.2 Binär-Eingänge EB1.. EB4 bzw. ENI

Allgemeine Beschreibung

Die Binär-Eingänge EB1.. EB4/ENI sind Eingänge für binäre bzw. digitale Signale.

EB4/ENI kann alternativ als Binäreingang EB4 oder als Eingang für HF-Sensoren (ENI) benutzt

werden.

EB1..EB3

und ENI liegen im unbeschalteten Zustand auf high-Signal und können z.B. über einen externen potentialfreien Kontakt gegen System-Null (Low-Signal) gezogen werden.

EB4 liegt im unbeschalteten Zustand auf Low-Signal.

Wegen der begrenzten Kontaktzahl des Steckers müssen bei Benutzung der Binäreingänge EB1. EB4 die Anschlüsse der analogen Eingänge E5/E6 durch Umlöten von Lötbrücken auf der Grundleiterplatte auf die Binär-Eingänge umgeschaltet werden (siehe Tabelle 6) und Anschlüsse bei der 19"-Steckkarte beachten (Tabelle 11).

Angaben über die elektrischen Eingangsdaten siehe Gebrauchsanweisung 42/18-40- .

Binär-Zustandserfassung der Binär-Eingänge EB1..-EB4/ENI

Mit dem Befehl RDBINP können, jeder Zeit und unabhängig, welche Funktion mit DEFBINP() definiert wurde, die Binär-Eingangs-Pegel (low oder high) von EB1..EB4/ENI gelesen werden.

Alle 4 Binär-Eingänge werden als Block gelesen und im Stack abgelegt. Mit POP x wird der Block den Bit's 0..3 der Byte- Variablen 'x' zugewiesen.

BIT-Zuordnung der Bin.-Eing.-Variablen x

128	64	32	16	8	4	2	1
0	0	0	0	EB2	EB1	EB3	EB4 ENI

lst ein bestimmter Eingang auf dessen Binärzustand zu prüfen, muß dieser mit einer .AND.-Verknüpfung selektiert werden.

Selektions-Werte:

Eing.	Selektion
EB1	→ EB1 = X.AND.04H
EB2	→ EB2 = X.AND.08H
EB3	→ EB3 = X.AND.02H
EB4	→ EB4 = X.AND.01H
ENI	→ ENI = X.AND.01H

Beispiel:

Signalisierung über GW1 wenn EB1 = "H"

10 RDBINP : POP EB

20 IF (EB.AND.04H) = 04H THEN GW1,1 ELSE GW1,0 30 GOTO 10

Digitale Meßfunktionen über Binär-Eingänge EB1...EB4/ENI

DEFBINP(y) definiert folgende digitalen Meßfunktionen über die Binär-Eingänge EB1..EB4/ENI: (siehe auch DEFBINP() und RDIGVAL unter Abschnitt 2.6 "BASIC-Befehle").

Code 'y'	Meßfunktion	Binär- Digitale Eingang min.		Signale max. (DIM)	KD
0	Funkt.abgeschalt.				
1	Frequenz	EB4/ENI	0	100000 Hz	1
2	Zähler (Impulse)	EB4/ENI	0	16 ⋅ 10 ⁶ lmp	1
3	Zähler (Impulse)	EB3	0	16 • 10 ⁶ lmp	1
4	Impuls-Dauer	EB1	0.0002	4660 s	1
5	Impuls-Dauer	EB2	0.0002	4660 s	1
6	Perioden-Dauer	EB1	0.0002	4660 s	1
7	Perioden-Dauer	EB2	0.0002	4660 s	1
8	Perioden-Dauer und	EB1	0.0005	327 s	1
	Imp.pro Meßzyklus		0	65535 Imp	

Tabelle 5 Funktions-Übersicht

Eingangsgröße (Eingang)	BASIC- Befehl	Eingang Brücke	409	5 413	415	6 419	EB1 410	EB2 414	EB3 416	EB4 420	ENI
Pegel (ENI)	RDBINP		()	()	()	()	()	()	()	()	
Pegel (EB3,EB4/ENI)	RDBINP		()	()	()	()	()	()	()	()	
Pegel (EB1EB4/ENI)	RDBINP		()	()	()	()	()	()	()	()	
Frequenz (ENI)	DEFBINP(1,z)		()	()	()	()	()	()	()	()	
Frequenz (EB4)	DEFBINP(1,z)		()	()	()	()	()	()	()	()	
Zählen (ENI)	DEFBINP(2)		()	()	()	()	()	()	()	()	
Zählen (EB4)	DEFBINP(2)		()	()	()	()	()	()	()	()	
Zählen (EB3)	DEFBINP(3)		()	()	()	()	()	()	()	()	
Zeit (EB1)	DEFBINP(4)		()	()	()	()	()	()	()	()	
Zeit (EB2)	DEFBINP(5)		()	()	()	()	()	()	()	()	
Periode (EB1)	DEFBINP(6)		()	()	()	()	()	()	()	()	
Periode (EB2)	DEFBINP(7)		()	()	()	()	()	()	()	()	
Periode/Imp.(EB1)	DEFBINP(8)		()	()	()	()	()	()	()	()	

Tabelle 6 Lötbrücken bei digitalen Eingangsgrößen

Lötbrücke geschlossen

[!] Die Analog-Eingänge E5 bzw. E6 können nur dann verwendet werden, wenn beide Lötbrücken (409,413 bzw. 415,419) geschlossen sind

6.1.3 Analog-Ausgänge

A1 → Stromausgang

A2 → Spannungsausgang

A1 und A2 sind getrennte Ausgänge. Es können verschiedene Meßgrößen über diese Ausgänge ausgegeben werden.

Ausg.	Ausgangs-	Digitalsignal-	Digital-/Analog-				
	Signale	Bereiche	Faktor KD				
A1	020 mA	0+3600	180				
A2	010 V	0+3600	360				

Tabelle 7 Bereichs-Tabelle der Analog-Ausgänge A1 und A2

6.2 Skalierung der analogen Ein- und Ausgänge

Bei P-Programmen wandeln die Befehle ADU(..), RDIG-VAL und DAU(..) die digitalen Signalwerte mit Hilfe der Parameter-Datei-Daten direkt in physikalische Meßwerte um.

Bei #- und bei User-Programmen müssen die digitalen Signal-Werte im Programm in physikalische Meßwerte umgerechnet (skaliert) werden. Die Skalierungs-Formel für die analogen Eingänge E1.. E6 und für die binär Eingänge EB/ENI lautet:

$$MW(n) = DW(n) * K3(n) + K2(n)$$

Die Skalierungs-Formel für die analogen Ausgänge A1. A2 lautet:

$$DW(n) = (MW(n) - K2(n)) / K3(n)$$

$$K1(n) = \frac{ME(n) - MA(n)}{SE(n) - SA(n)}$$

$$K2(n) = MA(n) - K1(n) * SA(n)$$

K3(n) = K1(n) / KD(n)

Die Variablen sind in ARRAYS aufgebaut:

 $n=1..6 \quad \rightarrow \text{Analog-Eingänge E1} \dots \text{E6}$

 $n = 7 \text{ u.8 } \rightarrow \text{Analog-Ausgänge A1 u. A2}$

n = 9 → Binär-Eingänge EB/ENI

MW(n) = physikalischer Meßwert

DW(n) = digitaler Signalwert

ME(n) = physikalischer Meßbereich-Ende

MA(n) = physikalischer Meßbereich-Anfang

SE(n) = Signalbereich Ende

SA(n) = Signalbereich Anfang

KD(n) = Digital-/Analog-Faktor (s. Tabellen 3, 5, 7)

Programm-Beispiele siehe Abschnitte 7.2 und 7.3

6.3 BASIC-Print-Flag setzen

Das BASIC-Print-Flag ist ein Register mit der TZA 4-RAM-Adresse 6FF5H. Es wird beim Starten eines Programms über TZAKON (siehe Abschnitte 2.2.7, 3.2.8 u. 3.3) auf 55H gesetzt. Alle P- und #-Programme testen, ob das Flag gesetzt. Wenn ja, werden Meßergebnisse über die Schnittstelle auf dem Bildschirm ausgegeben.

Bei User-Programmen kann der Anwender in seinen Programmen die Meßwert-Ausgabe durch Abfragen des Print-Flags steuern.

Beispiel:

60 IF XBY(6FF5H)=55H THEN GOTO 100

100 PRINT "Meßwert E1 = ".E1

Bei einem Start des TZA 4 über Power-On-Reset wird das Print-Flag nicht gesetzt und es werden keine Meßergebnisse über die Schnittstelle ausgegeben.

6.4 Binär-Erweiterungskarte

Die Erweiterungskarte wird auf die Grundleiterplatte aufgesteckt und stellt zusätzlich 24 binäre Ein/Ausgänge, unterteilt in 3 mal 8 PORTS A, B, C, zur Verfügung.

ВІТ			Zuor zu de		g der chluß		;	
PORT	7	6	5	4	3	2	1	0
A B C	a24	a26	a06 a28 a18	a30	c30	c28	c26	c24

Tabelle 8 PORT-Anschlüsse bei 19" Steckkarte mit D-Stecker (für F-Stecker z = a; d = c)

Die Erweiterungskarte kann in 2 verschiedenen Betriebsarten eingesetzt werden:

Betriebsart 1:

 Jeder PORT A,B u.C ist als Eingang oder Ausgang programmierbar. Dazu wird in das INIT-Register eine Code-Nr. geschrieben. Die Ports sind nur Byteweise lesbar oder beschreibbar.

INIT-Reg.Adresse: → 7FC7H
PORT-Adressen: PORT A → 7FC4H

PORT B → 7FC5H PORT C → 7FC6H

INIT-I = Eingang O = AusgangPORT A Code-Nr. PORT B PORT C 80H O 0 0 90H 0 0 82H 0 0 92H 1 0 89H O 0 1 99H 0 Т Т 8BH O 1 1 9BH 1

Tabelle 9 INIT-Code-Nummern
I = Eingang O = Ausgang

Beispiel: XBY(7FC7H)=99H definiert

PORT A,C → Eingänge PORT B → Ausgänge

- Lesen eines PORTs.

Es werden alle 8 Eingänge eines PORTs als BYTE gelesen.

Um zu prüfen, ob bestimmte Eingänge auf High-Pegel liegen, müssen die zugehörigen Bits mit .AND. verknüpft werden.

Beispiel:

Teste, ob PIN a28 und c28 auf High-Pegel.

rem Pin a28 \rightarrow BIT5 \rightarrow 20H rem Pin c28 \rightarrow BIT2 \rightarrow 04H rem BIT5 + BIT2 = 24H

10 PB=XBY(7FC5H) : rem Port B in Variable PB 20 IF (PB.AND.24H) = 24H THEN GW1,1 ELSE GW1,0

- Schreiben in PORTs.

Es werden alle 8 Ausgänge eines PORTs als BYTE beschrieben.

Beispiel:

XBY(7FC5H)=05H {setze BIT 0 u.2 (PIN c24,c28) von

PORT B auf High; alle anderen BIT's auf Low

Betriebsart 2:

Direkte BIT-Manipulation nur über PORT C: PORT C muß zuvor, wie unter Betriebsart 1 beschrieben, als Ausgangs-Port programmiert werden. Jedes einzelne PORT-BIT von Port C kann dann durch Schreiben einer Code-Nr. in das INIT-Reg. (Adr. 7FC7H) gesetzt bzw. rückgesetzt werden (siehe Tabelle 10).

PIN	C.	14	c1	16	C.	18	c2	20	a2	20	a¹	8	a1	16	a1	14
CODE	00Н	01H	02H	03H	04H	05H	06Н	07H	08Н	09Н	0AH	0ВН	осн	ODH	0EH	0FH
PEGEL	L	Н	L	Н	L	Н	L	Н	L	Н	L	Н	L	Н	L	Н

Tabelle 10

Beispiel:

XBY(7FC7H)=92H rem defin. Port A,B als Eing.

Port C als Ausg.

XBY(7FC7H)=07H rem PIN c20 wird auf "H" gesetzt XBY(7FC7H)=06H rem PIN c20 wird auf "L" gesetzt

7 BASIC 52-Programmbeispiele

Beispiel 1:

Aufgabe: Mittelwert-Berechnung von 6 Eingangsgrößen (jeweils im Meßbereich von 0...+20 mA).

- Ergebnis auf Strom-Ausgang A1 ausgeben.
- Wenn der Mittelwert > 80 %, dann Grenzwert 1 einschalten.
- Spitzenwert des Mittelwertes berechnen und auf Spannungsausgang A2 ausgeben.

```
rem Skalierung ausschalten
                                              rem alle 6 Eing. für -20..+20 mA definieren
20 DEFINP(1,1,1,1,1,1):
                                              rem Maxwert auf 0 setzen
30 MAXWERT=0:
                                              rem Totmann-Zelle bedienen
 40 WDOG:
                                              rem alle 6 Eing. AD-Wandeln (0..3600 Digit)
50 ADU(1,2,3,4,5,6):
                                              rem Meßwerte den Variabl. E1.. E6 zuweisen
60 POP E1,E2,E3,E4,E5,E6:
                                              rem Mittelwert aus 6 Meßwerten berechnen
70 X=(E1+E2+E3+E4+E5+E6)/6:
 80 PUSH X:
                                              rem Mittelwert auf Stack schieben
                                              rem Mittelwert auf I-Ausg. ausgeben.
90 DAU(I):
100 IF X > 3600\star0.8 THEN GW1,1 ELSE GW1,0: rem Teste ob Wert= 80 \%
                                              rem Teste ob X > Max.
110 IF X > MAXWERT THEN MAXWERT=X:
                                              rem Max. Wert auf Stack
120 PUSH(MAXWERT):
                                              rem Max. Wert auf Spannungsausg.ausgeben.
130 DAU(U):
140 PRINT U.(#######,#), "MITTELWERT = ",X," MAX. WERT = ",MAXWERT"
150 GOTO 40:
                                              rem nächste Messung.
```

Beispiel 2:

10 SKALO

Aufgabe: Es soll über Eingang 1 ein physikalischer Meßwert jede Sekunde gemessen, skaliert und auf Bildschirm und Ausgang A2 ausgegeben werden.

```
- Signalbereich-Eingang 

- Meßbereich-Eingang 

- Signalbereich-Ausgang 

- Meßbereich-Ausgang 

- Meßbereich-Ausgang 

- Meßbereich-Ausgang 

- Meßbereich-Eingang 

- A2 = + 0... + 20 mA 

E1 = + 4... + 20 mA 

E1 = - 100... + 900 t 

A2 = - 100... + 900 t
```

10	SKALU	Hem Okanerung ausschaften
20	DEFINP(1,0,0,0,0,0)	rem Eingänge definieren:
30	DATETIME(1)	:rem Echtzeituhr einschalten
40	SA(1)=4: SE(1)=20	:rem E1 SignAnf.uEnde festlegen
50	SA(8)=0: SE(8)=10	:rem A2 SignAnf.uEnde festlegen
60	MA(1)=-100: ME(1)=900	:rem E1 phys.Meß-Anf. uEnde festlegen
70	MA(8)=-100: ME(8)=900	:rem A2 phys.Meß-Anf. uEnde festlegen
80	KD(1)=180	rem Digital-Analog-Faktor für 20 mA Eing.
90	KD(8)=360	:rem Digital-Analog-Faktor für 10 mA Ausg.
100	K1(1)=(ME(1)-MA(1))/(SE(1)-SA(1))	:rem K1-Skal.Vorberechnung E1
110	K2(1)=MA(1)-K1(1)*SA(1)	:rem K2-Skal.Vorberechnung E1
120	K3(1)=K1(1)/KD(1)	:rem K3-Skal.Vorberechnung E1
130	K1(8)=(ME(8)-MA(8))/(SE(8)-SA(8))	:rem K1-Skal.Vorberechnung A2
140	K2(8)=MA(8)-K1(8)*SA(8)	:rem K2-Skal.Vorberechnung A2
150	K3(8)=K1(8)/KD(8)	:rem K3-Skal.Vorberechnung A2
REM	Beginn Programm-Schleife	
160	SECFLAG=6D16H	rem Adr. Sekunden-Flag laden
170	XBY(SECFLAG)=0	:rem Sekunden-Flag auf 0 setzen
180	DO	rem wiederhole solange, bis
190	WHILE XBY(SECFLAG)=0	:rem SECFLAG von Uhr gesetzt.
200	WDOG	rem Programmüberwachung abfragen:
210	XBY(SECFLAG)=0	:rem SECFLAG wieder auf 0 setzen
220	ADU(1):POP DW(1)	:rem AD-Wandlung; Digit.Sign-Wert nach DW(1)
230	MW(1)=DW(1)*K3(1)+K2(1)	rem E1-Skalierung durchführen:
240	PRINT USING(####.#),"Der Meßwert beträgt:	: ",MW(1)," t",CR,
242	MW(8)=MW(1)	
250	DW(8) = (MW(8) - K2(8))/K3(8)	:rem Skalierung Ausg. A2
260	PUSH DW(8): DAU(U)	rem DA-Wandlung und Ausgabe A2
270	GOTO 160	:rem wiederhole Messung

:rem Skalierung ausschalten

Beispiel 3: Erstellung eines eigenen #-Programms

Zur Erstellung eines #-Programms empfiehlt es sich, ein vorhandenes Programm zu kopieren, zu überschreiben

und unter neuem Namen #jmmttyy.B52 (Datum+Nr.) abzuspeichern. Falls noch kein Programm existiert, sollte das allgemeine Programm #111-V01.B52 (Summierung) kopiert werden.

10	REM File		:#111-V01.B52
20	REM Datum		:20.09.1990
30	REM Programm		:#111-V01.B52
40	REM Auftr/Pos		:12345678/0001
50	REM F-Nr.		:6.000000.0
60	REM Bearbeiter		: xxxxx
70	REM FB-Nr.		: 000000
80	REM Bestell-Nr.		: 18015-0-000000
90	REM Aufgabe		: Summierung von 6 bewerteten Eingängen
100	REM		: mit Zählimpulsausgang und Grenzwertsignal
REM	HINWEISE FÜR PROC		
REM	Empteniung: Erst Pro	gra	mm ausdrucken, unter neuem Namen abspeichern. REM-Zeilen
REM	(nicht di	ie U	berschriften) löschen,dann ändern. Zeile/Ziffer/Variable fehlender
REM	Eingang)e//	Ausgänge löschen; betrifft: 200250, 330380, 10101080, 11001110.
REM			KON <f1> beachten für Anschluß-Pins 'Brücken und Tabelle für</f1>
REM			gnal-Code
	Zeilen 10,30,40,50	→	TZAKON liest den Text hinter ":" und bildet daraus eine gemeinsame Menüzeile.
REM	Zeile 10		Daher beim Überschreiben Anzahl der Zeichen beachten.
	Zelle 10	→	File-Namen überschreiben = #+Jahres-Endziffer+Monat+Tag+Zählnr +.B52
REM	7-:1- 00		z.B. 0092033.B52 ='0 +09 +20 +33 +.B52 für 1990. 09. 20. Zählnr 33
	Zeile 20		:10 Zeichen überschreiben
	Zeile 30		:12 Zeichen überschreiben
	Zeilen 40 Zeilen 50		:13 Zeichen überschreiben
	Zeilen 60100		:10 Zeichen überschreiben
	Zeilen 200270		Text ergänzen, ändern
REM	Zelleli 200270	7	SA(),SE() = Signal-Anfang/-Ende
REM			MA(),ME() = Meß-Anfang/-Ende
REM			COD() = Eingangssignal-Code siehe HELP
REM			(1)(6) = Eingänge E1E6
REM	Zeilen 200250	_	(7),(8) = Ausgänge A1,A2
REM	Zelieli 200230	_	SA,SE,MA,ME,COD() überschreiben
REM			Zeilen für nicht belegte Eingänge löschen COD(n) = x
REM			$x = 1$ bedeutet E(n) liegt innerhalb \pm 20mA $x = 4$ bedeutet E(n) liegt innerhalb \pm 10V
REM	Zeilen 260,270		SA,SE,MA,ME,COD() überschreiben
REM	201011 200,270		Zeilen ! nicht ! löschen
REM			COD(7) = 1 bedeutet A1 belegt
REM			COD(8) = 0 bedeutet A1 belegt
REM			wenn COD() = 0, dann SA,MA = 0, SE,ME = 1
REM	Zeile 280,285	→	Konstanten (+-) ergänzen, überschreiben
	Zeile 290		Eingangsignal-Code x für E(n) überschreiben
REM			DEFINP(x,0,x,0,x,x),(6 Code-Angaben für E1 E6)
REM			Code = x, 0, x, 0, x, x; 0, wenn E nicht belegt
REM			für E1,E2,E3,E4,E5,E6
REM	Zeilen 300320	→	nicht ändern
REM	Zeilen 330380		Zeilen für nicht belegte Eingänge löschen (n = 5 → Eingang E5)
REM	Zeilen 390640	→	nicht ändern (Skalier-Unterprogramme für Ein-/Ausgänge)
REM	Zeilen 800820		" Text " eventuell ändern (Text = Überschrift für Bildschirm-Anzeige)
REM	Zeilen 850860		Variablen, Print-Format eventuell ändern (für Bildschirm-Anzeige)
REM			z.B. using (F14), (0), (###.####)
REM	Zeilen 10001150	\rightarrow	Programm ändern
REM	Zeile 1000		Wdog muß im Programm einmal vorkommen
REM	Zeile 1010		Ziffer für nicht belegte Eingänge weglassen (ADU(1,3) → Eingänge E1,E3 belegt)
REM	Zeile 1020		Variable für nicht belegte Eingänge weglassen
REM			(POP SW1,SW3; SW1,SW3=Variable für E1,E3)
REM			(Anzahl der Variablen = belegte Eingänge)
REM	Zeilen 10301080	\rightarrow	Zeile für nicht belegten Eingang löschen.
REM			(wandelt dig.Wert [SW] in phys.Wert [MW] um)
REM	Zeilen 11001110	\rightarrow	Zeile für nicht belegten Ausgang löschen.
REM			(wandelt phys.Ergebnis A in dig.Wert um und liefert analogen Ausgang)
REM			1100 → A1; 1110 → A2

```
REM Zeile 1120.1130
                          → ohne Impulsausgang: Zeilen löschen. (Impulszähler, ZW=Anzahl der Impulse/s)
REM Zeile 1140
                           → nicht ändern, liefert Bildschirmanzeige
      Zeile 1150
                          → nicht ändern, Schleifenende
REM
REM
      Zeile 1200..1240
                           → ändern oder ergänzen
REM
      Signalanfang SA>0
                          → bei Eingang 4..20 mA ist unter 4 mA → MW«MA.
                             daher → IF MW1<MA(1) THEN MW1=MA(1)
REM
REM
                             PUSH ZW: IMPCOUNT
      Zählimpulsausgang
REM
                             Es muß sein ZW = Zmax/ZFA \le 1 (Zmax = Z für 100%)
REM
                             wähle z.B.: Zmax/ZFA = 1, 0.1, 0.01, 0.001
      PROGRAMM-BESCHREIBUNG
REM
REM A=C1*E1+C2*E2+C3*E3+C4*E4+C5*E5+C6*E6+k7
REM E1...E6
                             : Physikalische Eingänge
REM C1...C6
                             : Bewertungsfaktoren
REM A
                             : Physikalischer Ausgang
     ZSCHW
REM
                             : Zschwelle
                             : Zählfaktor = 3600 · Impulswertigkeit
REM
      ZFA
      Impulswertigkeit
                             : Phys.Wert/Impuls z.B. 10 kg/Imp.
REM
REM
      MWmin
                             : unterer Grenzwert für Grenzwertschalter
REM
      IF MWn<MA(n) THEN MWn=MA(n)
                                              → z.B. unter 4 mA bleibt Meßwert=0
                                              → unter Zschwelle wird nicht gezählt
REM
      IF Z<ZSCHW THEN Z=0
REM IF MW«MWmin THEN GW1,1 ELSE GW1,0 → Schalter GW1 schaltet ein bei < MWmin
REM BEISPIEL
      E1 = 0... 1000 \text{ kg/h};
                             c1 = 1.0
                                              → 4..20 mA
                                                                             COD(1)=1
REM
                                              → 4..20 mA
REM
      E2 = 0... 4000 \text{ kg/h};
                             c2 = 1.0
                                                                             COD(2)=1
      E3 = 0... 5000 \text{ kg/h};
                                              → 4..20 mA
REM
                             c3 = 1.0
                                                                             COD(3)=1
                                              → 4..20 mA
REM
      E4 = 0... 2000 \text{ kg/h};
                             c4 = 1.0
                                                                             COD(4) = 1
                                              → 4..20 mA
      E5 = 0... 4000 \text{ kg/h};
                             c5 = 1.0
                                                                             COD(5)=1
REM
      E6 = 0... 1000 \text{ kg/h};
                             c6 = 1.0
                                              → 4..20 mA
                                                                             COD(6) = 1
REM
REM A1 = 0...10000 \text{ kg/h};
                             K7 = 0
                                              → 4..20 mA
                                                                             COD(7)=1
      A2 = 0...10000 \text{ kg/h}
                                              → 0..10 V
REM
                                                                             COD(8) = 1
      Zähler
                                              → 10 kg/lmp
                                                                           → ZFA=3600*10
RFM
REM
      Zählerschwelle
                                              → ZSCHW
                                                                             100 kg/h
REM
      Unterer Grenzwert für E1
                                              → MW1min
                                                                           = 200 \text{ kg/h}
REM
      KONTROLL-WERTE
                                   E3/phys.
                                                               E5/phys.
                                                                              E6/phys.
REM
      E1/phys.
                    E2/phys.
                                                 E4/phys.
                                                                                            A/phvs.
      1000.00
                        0.00
                                      0.00
                                                     0.00
                                                                   0.00
                                                                                 0.00
                                                                                              1000.00
REM
REM
         0.00
                    4000.00
                                      0.00
                                                     0.00
                                                                   0.00
                                                                                 0.00
                                                                                              4000.00
REM
         0.00
                        0.00
                                   5000.00
                                                     0.00
                                                                   0.00
                                                                                 0.00
                                                                                              5000.00
                                                 2000.00
REM
         0.00
                        0.00
                                      0.00
                                                                   0.00
                                                                                 0.00
                                                                                              2000.00
                                                                4000.00
                        0.00
                                                                                              4000.00
REM
         0.00
                                      0.00
                                                     0.00
                                                                                 0.00
                                                                              1000.00
                                                                                              1000.00
REM
         0.00
                        0.00
                                      0.00
                                                     0.00
                                                                   0.00
REM
      1000.00
                        0.00
                                   5000.00
                                                     0.00
                                                                4000.00
                                                                                 0.00
                                                                                            10000.00
                    4000.00
                                                 2000.00
                                                                              1000.00
                                                                                              7000.00
REM
         0.00
                                      0.00
                                                                   0.00
       250.00
                    1000.00
                                   1250.00
                                                  500.00
                                                                1000.00
                                                                               250.00
                                                                                              4250.00
REM
REM
                   E2/mA
REM
      E1/mA
                               E3/mA
                                            E4/mA
                                                        E5/mA
                                                                     E6/mA
                                                                                  A1/mA
                                                                                              A2/V
                                4.000
                                             4.000
                                                          4.000
                                                                      4.000
                                                                                                1.000
      20.000
                    4.000
                                                                                   5.600
REM
                                                                                  10.400
       4.000
                   20.000
                                4.000
                                             4.000
                                                          4.000
                                                                      4.000
                                                                                                4.000
REM
       4.000
                    4.000
                               20.000
                                             4.000
                                                          4.000
                                                                      4.000
                                                                                  12.000
                                                                                                5.000
REM
REM
       4.000
                    4.000
                                4.000
                                            20.000
                                                          4.000
                                                                      4.000
                                                                                   7.200
                                                                                                2.000
REM
       4.000
                    4.000
                                4.000
                                             4.000
                                                         20.000
                                                                      4.000
                                                                                  10.400
                                                                                                4.000
RFM
       4.000
                    4.000
                                4.000
                                             4.000
                                                          4.000
                                                                     20.000
                                                                                   5.600
                                                                                                1.000
REM
      20.000
                    4.000
                               20.000
                                             4.000
                                                         20.000
                                                                      4.000
                                                                                  20.000
                                                                                               10.000
                                                          4.000
REM
       4.000
                   20.000
                                4.000
                                            20.000
                                                                     20.000
                                                                                  15.200
                                                                                                7.000
                    8.000
                                8.000
                                                                      8.000
                                             8.000
                                                          8.000
REM
       8.000
                                                                                  10.800
                                                                                                4.250
REM PROGRAMM-BEGINN
REM
      PARAMETERDATEN-EINGABE
200
                                                                  :ME(1)=1000
      SA(1)=4
                          :SE(1)=20
                                              :MA(1)=0
                                                                                       :COD(1)=1
210
      SA(2)=4
                          :SE(2)=20
                                              :MA(2)=0
                                                                  :ME(2) = 4000
                                                                                       :COD(2)=1
      SA(3)=4
                                              :MA(3)=0
                                                                  :ME(3) = 5000
220
                          :SE(3)=20
                                                                                       :COD(3)=1
230
      SA(4) = 4
                          :SE(4)=20
                                              :MA(4)=0
                                                                  :ME(4)=2000
                                                                                       :COD(4)=1
240
      SA(5) = 4
                          :SE(5)=20
                                              :MA(5)=0
                                                                  :ME(5)=4000
                                                                                       :COD(5)=1
250
      SA(6)=4
                          :SE(6)=20
                                              :MA(6)=0
                                                                  :ME(6)=1000
                                                                                       :COD(6)=1
260
      SA(7) = 4
                          :SE(7)=20
                                              :MA(7)=0
                                                                  :ME(7)=10000
                                                                                       :COD(7)=1
```

```
270
     SA(8) = 0
                        :SE(8)=10
                                         :MA(8)=0
                                                           :ME(8)=10000
                                                                             :COD(8)=1
280
                C2=1:
     C1=1:
                           C3=1:
                                                 C5=1:
                                                            C6=1:
                                                                      K7=0
285
     ZSCHW=100:
                       ZFA=3600*10:MW1min=200
290
     DEFINP(1,1,1,1,1,1)
     SKALIER-VORBERECHNUNG
RFM
300
     SKAL0
310
     Test=XBY(6FF5H):
                            IF TEST=55H THEN GOSUB 800
320
     KD(7)=180: KD(8)=360
330
     n=1:
                 GOSUB 500 :GOSUB 600
340
     n=2:
                 GOSUB 500 :GOSUB 600
350
     n=3:
                 GOSUB 500 :GOSUB 600
360
     n=4.
                 GOSUB 500 :GOSUB 600
370
     n=5:
                 GOSUB 500 :GOSUB 600
380
     n=6:
                 GOSUB 500 : GOSUB 600
390
     n=7:
                 GOSUB 600
400
     n=8:
                 GOSUB 600
     GOTO 1000
410
500
     ON COD(n) GOTO 530, 540, 510, 520, 540, 540, 540, 580, 580
510
     KD(n)= 3600:RETURN
520
     KD(n)= 360: RETURN
530
     KD(n)= 180: RETURN
540
     KD(n)= 18: RETURN
580
     RETURN
600
     IF COD(n) < 1 .OR. COD(n) > 7 THEN K3(n)=1:RETURN
610
     K1(n) = (ME(n)-MA(n))/(SE(n)-SA(n))
620
     K2(n) = MA(n)-K1(n) * SA(n)
630
     K3(n) = K1(n)/KD(n)
640
     RETURN
800
     PRINT "E1
                    E2
                            E3
                                    E4
                                            E5",
     PRINT "E6
810
                    A"
820
     RETURN
850
     PRINT USING(F4), MW1, MW2, MW3, MW4, MW5, MW6, A, cr,
860
     RETURN
REM HAUPTPROGRAMM
1000 WDOG
1010 ADU(1,2,3,4,5,6)
1020 POP SW1,SW2,SW3,SW4,SW5,SW6
1030 MW1=K3(1)*SW1+K2(1)
1040 MW2=K3(2)*SW2+K2(2)
1050 MW3=K3(3)*SW3+K2(3)
1060 MW4=K3(4)*SW4+K2(4)
1070 MW5=K3(5)*SW5+K2(5)
1080 MW6=K3(6)*SW6+K2(6)
1090 GOSUB 1200
1100 PUSH(A-K2(7))/K3(7): DAU(I)
1110 PUSH(A-K2(8))/K3(8): DAU(U)
1120 PUSH ZW
1130
     IMPCOUNT
1140 IF TEST=55H THEN GOSUB 850
1150 GOTO 1000
REM UNTERPROGRAMM
1200 IF MW1<MA(1) THEN MW1=MA(1)
1210 IF MW1<MW1min THEN GW1,1 ELSE GW1,0
1220 A=C1*MW1+C2*MW2+C3*MW3+C4*MW4+C5*MW5+C6*MW6+k7
1230 Z=A
1240 IF Z<ZSCHW THEN Z=0
1250 ZW=Z/ZFA
1260 RETURN
```

Anhang

TZA 4-Betriebssystem

Das Betriebssystem des Meßrechners TZA 4 ist im EPROM 1 abgelegt. Es übernimmt folgende Funktionen:

- Steuerung der Multiplex-Schalter der analogen Einund Ausgänge
- Steuerung des AD- bzw. DA-Wandlers.
- Korrektur der AD- bzw. DA-Signalwerte mit Hilfe der im EPROM 1 abgelegten ADU-Korrekturdaten (Nullpunkt und Endwert)
- Steuerung der Grenzsignalausgänge GW1 und GW2
- Steuerung der EPROM-Brennautomatik zum "Einbrennen" von Programmen und Parameter-Dateien in das EPROM 2 sowie der ADU-Korrekturdaten in das EPROM 1
- Steuerung der gesamten Speicherverwaltung des BASIC-EPROM 2

Speicherorganisation

Der Meßrechner TZA 4 hat 5 unterschiedliche Speicher.

- der interne µP-RAM-Speicher. Er dient in erster Linie dem Betriebssystem und dem BASIC-Interpreter beim Ausführen der Instruktionen.
- der interne µP-ROM-Speicher. In ihm ist der BASIC-Interpreter mit den BASIC 52-Standard-Befehlen abgespeichert.
- der externe RAM-Speicher. Er unterteilt sich in einen gepufferten und einen ungepufferten Bereich. Im gepufferten Bereich werden alle Daten abgelegt, die bei Spannungsausfall erhalten werden müssen; z.B. Daten Echtzeit-Uhr oder Meßdaten. Das aktuelle Programm und die Parameter-Datei werden im ungepufferten Bereich abgespeichert.
- der Betriebssystem-EPROM-Speicher
- der BASIC-Programm-Speicher. Er ist in einen unteren und einen oberen Bereich unterteilt. Der untere Bereich ist mit den Programmen und Parameter-Dateien, der obere Bereich mit dem Programmverzeichnis und der Autostart-Liste belegt. Im Programmverzeichnis sind alle Namen der im EPROM 2 abgespeicherten Programme abgelegt. Der letzte Eintrag der Autostart-Liste verweist auf das zu startende Programm und die Datei nach einem Reset bzw. Power-On-Reset.

Speicherbelegung im internen µP-RAM

Die Speicherplätze mit den folgenden Adressen befinden sich im internen μP -RAM und können im BASIC mit PRINT DBY(Adresse) ausgelesen werden.

Adress	e Bedeutung
0AH	: BASIC-Text-Pointer H-Byte
09H	: Argument-Stack-Pointer L-Byte (H-Byte immer 1)
13H	: Start-Adr. des aktiven BASIC-Programms H-Byte
14H	: Start-Adr. des aktiven BASIC-Programms L-Byte
47H	: Millisekunden-Zähler der Echtzeit-Uhr
08H	: BASIC-Text-Pointer L-Byte
48H	: Sekunden-Zähler der Echtzeit-Uhr H-Byte
49H	: Sekunden-Zähler der Echtzeit-Uhr L-Byte

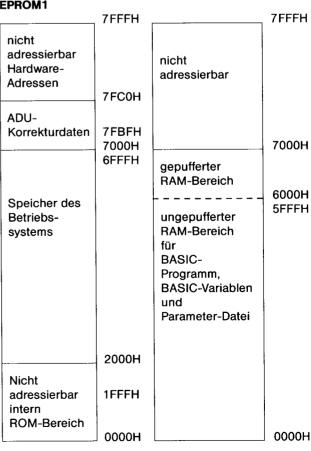
Memory-Map des TZA4

BASIC-EPROM 2

FFFFH Jeder neue Prog.Name wird 1. Progr.-Name n. Progr./Name unter den letzten Namen "eingebrannt". Oberster Prog. Eintrag wird n. Progr./Datei nach Reset automatisch gestartet 1. Progr./Datei OFC00H 1. Param.-Datei **OFBFFH** 2. Param.-Datei Parameter-Datei-Bereich. Jede neue Datei wird unter n. Param.-Datei der zuletzt abgespeicherten Datei "eingebrannt". BASIC-Programme werden n. BASIC-Progr. ab Adr.8000H von unten nach oben im EPROM 2. BASIC-Progr. "eingebrannt". 1. BASIC-Progr. 8000H

BETRIEBSSYSTEM-EPROM1

RAM-SPEICHER



Speicherbelegun	ig des externen RAM	Adresse	Bedeutung				
Diese Speicherpla können im BAS (Adresse) ausgele	ätze befinden sich im externen RAM und BIC-Kommand-Mode mit PRINT XBY esen werden.	6F61H 6F67H 6F6DH 6F8BH	Digitaler DA-Ausgabe-Wert A1 Digitaler DA-Ausgabe-Wert A2 Digitaler Signalwert EE/ENI				
Adresse	Bedeutung	6F8CH6FEAH	Digitaler Ausgabe-Wert BA Betriebssystem-Daten				
	: Diverse BASIC-Verwaltungs- Speicher	6FEBH :	Leitungsabgleich Pt 100-Reg. (10 Ω) wenn auf 55H gesetzt und der Signalwert < als Signal-Anfangswert,				
0200H5FFFH	: Argument-Stack-Bereich: BASIC-Programm-Speicher: Gepufferter Anwender-Speicher-		so wird Signalwert auf Signal-Anfang gesetzt (gilt nur bei Strom-Eingang)				
	bereich. Hier können gegen Spannungsausfall geschützte Daten		GW 2-Register BIT 05 → E1E6; BIT 6 u.7 → A1 u.A2 GW 1-Register BIT 05 → E1E6;				
6500H68FAH	X abgelegt werden (z.B. XBY(6000H)= X) : Reserviert		BIT 6,7 → A1 u.A2 wenn Grenzwert über- bzw. unter-				
68FBH69D2H			schritten, wird das entsprechende BIT gesetzt.				
	: Speicher der Binäreingangs- Funktionen : Lese-Flag der Binäreingangs-Funk-		GW 2-Register BIT 0 = 1 → GW 2 von EB gesetzt				
002711	tionen, wenn 55H, kann neue Mes- sung mit RDIGVAL gelesen werden.		GW 1-Register BIT 0 = 1 → GW 1 von EB gesetzt				
69F3H	: Floating-Point-Wert der Zeitbasis bei Frequenzmessung		GW-Enable-Reg. BIT 0 = 1 → GW für EB enable				
	Parameter-Datei-Bereich. Datei beginnt bei 6CBEH		GW-Enable-Reg. BIT 05 → E1 E6; BIT 6,7 → A1,A2				
6D00H6D08H :	: BASIC-Programm-Namen	6FF5H :	BASIC-Print-Flag. Wird von TZAKON				
	Sekunden-Speicher der Echtzeit-Uhr	6FF6H ;	bei Programm-Start auf 55H gesetzt Programmzeiger: Zeigt auf das aktu-				
6D11H :	Minuten-Speicher der Echtzeit-Uhr		elle Programm				
6D12H :	Stunden-Speicher der Echtzeit-Uhr	6FF7H :	Dateizeiger: Zeigt auf die aktuelle				
6D13H :	: Tag-Speicher der Echtzeit-Uhr	CECOLI	Datei 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
	: Monat-Speicher der Echtzeit-Uhr : Jahr-Speicher der Echtzeit-Uhr	6FF8H :	Skalierung-Flag: Wenn 5AH, werden				
	Sekunden-Flag. wird jede Sekunde auf FFH gesetzt, muß vom BASIC-	SEEON SEEEN .	die Meßsignale der Ein- u. Ausgänge skaliert. Betriebssystem-Daten				
	Programm wieder auf 0 gesetzt werden.	0113110FFFH .	Detrieussystem-Daten				
69D7H :	Wochentag 17 (1 = Sonntag)						
6D18H6D1FH :	Reserviert	BASIC 52-Inter	preter				
	Betriebssystem-Daten						
	Physikalischer Meßwert des Eingangs E1	Modus und im RU	erpreter arbeitet in 2 Modi: Im Direkt- IN-Modus.				
	Physikalischer Meßwert des Eingangs E2	Im Direkt-Modus werden BASIC-Befehle nach ihrer gabe und Drücken der RETURN-Taste sofort interpre					
	Physikalischer Meßwert des Eingangs E3	und ausgeführt. Beispiel:					
	Physikalischer Meßwert des Eingangs E4 Physikalischer Meßwert des	Eingabe: PRINT 3 Antwort: 12	*4				
	Eingangs E5 Physikalischer Meßwert des		ird ein BASIC-Programm Zeile für Zeile usgeführt Jede Zeile beginnt mit einer				
	Eingangs E6 Physikalischer Meßwert des	Zeilen-Nr., gefolgt	von den BASIC-Befehlen.				
6EDAH :	Ausgangs A1 Physikalischer Meßwert des	Beispiel: 10 FOR I =1 TO 10 20 PRINT I	00				
6EE0H :	Ausgangs A2 Physikalischer Meßwert der Eingänge EB/ENI	30 NEXT I RUN					
6EE1H6F54H :	Reserviert	1 2 3 100					
6F3DH :	Digitaler ADU-Signalwert E1	Folgende BASIC ! nutzt werden:	52-Befehle dürfen im TZA 4 nicht be-				
6F43H :	Digitaler ADU-Signalwert E2						
6F49H :	Digitaler ADU-Signalwert E3	LIST#, LIST@,	ROM, PROG, PROG1PROG6,				
6F55H :	Digitaler ADU-Signalwert E4 Digitaler ADU-Signalwert E5		G6 BAUD, PRINT#, PRINT@, PH0.#, I.@, UI1, UI0, NULL, PGM, IDLE, XTAL,				
6F5BH :	Digitaler ADU-Signalwert E6	RCAP2, T2CON	OII, OIO, NOLE, FUIN, IDLE, ATAL,				

BASIC 52-Datenformate.

- Floating-Point-Daten:

Diese werden auf 8 Dezimal-Stellen berechnet. Die 8. Stelle ist gerundet.

Bereich: ± 1E-127...± 0.9999999E+127

Die Eingabe der Daten kann in 4 verschiedenen Formaten erfolgen:

Beispiele:

Integer, Dezimal, Hexadezimal, Exponential 129, 34.89, 0A6EH, 1.23456E+3

- Integer-Daten:

Bereich: 0...65535 (0FFFFH)

- Konstanten:

Konstanten können Floating-Point-Werte oder Integer-Werte sein.

Hinweise für Variablen

Bei Verwendung von Variablen-Namen ist zu beachten:

- Ein Variablen- Name darf max. 8 Zeichen betragen.
- Das 1. Zeichen muß immer ein Alpha-Zeichen sein.
- Eine Variable wird nur durch das erste- und letzte
 Zeichen und durch die Anzahl der Zeichen unterschieden, z.B. ES1 und EK1 werden nicht unterschieden.
- Ein Variablen-Name darf kein reserviertes Wort enthalten. (z.B. ist in der Variablen LABSUS das reservierte Wort "ABS" enthalten). Dies führt zur Meldung Syntax Error.

Für weitere Informationen wird auf das User-Manual des BASIC-52-PROZESSORS hingewiesen (Order Nr.270010-003 der Firma INTEL).

Anschlußbelegung der 19"-Steckkarte mit D-Stecker

Kanal	Meßgröße	Anschlüsse
Eingang E1	E1 (mA,V,Ω,Pt100)	+ 2a - 2c
Eingang E2	E2 (mA,V,Ω,Pt100)	+ 4a - 4c
Eingang E3	E3 (mA,V, Ω ,Pt100)	+ 6a — 6c
Eingang E4	E4 (mA,V, Ω ,Pt100)	+ 8a — 8c
Eingang E5	E5 (mA,V,Ω,Pt100)	+10a -10c
Eingang E6	E6 mA,V, Ω ,Pt100)	+12a -12c
Eingang EB1	EB1 Binär-Eing.1	+10a -16c22c 1)
Eingang EB2	EB2 Binär-Eing.2	+10c -16c22c 1)
Eingang EB3	EB3 Binär-Eing.3	+12a -16c22c ²)
Eingang EB4	EB4 Binär-Eing.4	+12c -16c22c ²)
Eingang ENI	ENI HF-Schalter-Eing	+14c -16c22c 3)
Ausgang A1	A1 (mA)	+18a -16c22c
Ausgang A2	A2 (V)	+20a -16c22c
Impulsausgang f	AB1 (Imp/h)	+16a -16c22c
Konstantstrom	lk (für Pt100)	+14a -16c22c
Grenzw.schalter(Min)	GW1 (24V/0.1A)	+24a -24c
Grenzw.schalter(Max)	GW2 (24V/0.1A)	+26a -26c
Energie-Versorgung	24V~/115V~/230V~	+c30 −32c
Schutzleiter	Erde	c28
Schutzieren	CIUE	620

Tabelle 11 Anschlüsse bei 19"-Steckkarte mit D-Stecker (für F-Stecker z = a; d = c)

- alternativ zu Analogeingang E5 → Binäreingänge EB1 u. EB2
- alternativ zu Analogeingang E6 → Binäreingänge EB3 u. EB4
- 3) alternativ ENI oder EB4

Gewährleistung und Haftung

- Nach dem Stand der Technik sind trotz intensiver Prüfung Fehler in der Software nicht auszuschließen.
- Ein nachvollziehbarer Programmfehler, den der Erwerber Hartmann & Braun schriftlich mitteilt, wird von Hartmann & Braun in einer angemessenen Zeit kostenlos korrigiert.
- Sollte dies Hartmann & Braun trotz wiederholter Bemühungen nicht gelingen und kann der Erwerber das Programm deshalb nicht einsetzen, so kann er unter Ausschluß weitergehender Ansprüche das Recht auf Minderung der Vergütung oder das Recht auf Wandlung geltend machen.
- Die Gewährleistung beginnt mit dem Zeitpunkt der Lieferung und endet nach 6 Monaten. Hartmann & Braun übernimmt keine Gewährleistung für die korrekte Speicherung der Daten auf Disketten und übernimmt keine Haftung für Schäden, die durch fehlerhafte Disketten entstehen.
- Schadensersatzansprüche des Erwerbers jeglicher Art sind ausgeschlossen, inbesondere auf Ersatz von Folgeschäden, wie z. B. bei Verlust von Daten oder entgangenem Gewinn, Ansprüche aus Unmöglichkeit positiver Forderungsverletzung, wegen Nichterfüllung und unerlaubter Handlung. Dieser Haftungsausschluß gilt nicht bei Vorsatz oder grober Fahrlässigkeit von gesetzlichen Vertretern oder leitenden Angestellten.

Bedingungen für die Nutzung der Programme

Die Lieferung der Programme erfolgt unter folgenden Bedingungen:

- Die Programme einschließlich eventuell kundenseitig erstellter Kopien dürfen nur auf einem Rechner benutzt werden
- Die gelieferten Programme dürfen nur in maschinenlesbarer oder gedruckter Form vervielfältigt werden, wenn die Kopie dazu dient, die Programme gegen Verlust zu schützen.
 - Die in den Programmen enthaltenen Copyright-Vermerke dürfen weder in der Originaldiskette noch in Kopien entfernt werden.
- 3. Die Änderung und über den vorgesehenen Umfang hinausgehende Benutzung erfordert die ausdrückliche Zustimmung von Hartmann & Braun.
- 4. Der Erwerber kann die Nutzungsrechte an Dritte übertragen. Dabei ist explizit auf die Lieferbedingungen von Hartmann & Braun hinzuweisen. Mit der Übertragung auf Dritte erlöschen alle Nutzungsrechte des Veräußerers. Kopien, die nicht mitveräußert wurden, sind zu vernichten.

Technische Änderungen vorbehalten.	
Diese Bedienungsanleitung ist urheberrechtlich geschützt. Die Übersetzung so – auch als Bearbeitung oder in Auszügen –, insbesondere als Nachdruck, pl Form der Speicherung in Datenverarbeitungsanlagen oder Datennetzen ohne oden zivil- und strafrechtlich verfolgt.	notomechanische oder elektronische Wiedergabe oder in
ABB	
ABB Automation Products GmbH Borsigstraße 2 D-63755 Alzenau Tel. (0 60 23) 92 - 0 Fax (0 60 23) 92 - 33 00 http://www.abb.de	Technische Änderungen vorbehalten Printed in the Fed. R. of Germany 42/18-41 DE Rev. 02 Ausgabe 02.01