

technische Daten

Sprungverteiler

Von der Interessengemeinschaft Heimcomputer der KdT des IZ der TU-Dresden wurde ein Sprungverteiler vereinbart, der zur Einbindung von Erweiterungen des „Z 1013“-Monitorprogramms in Anwenderprogrammen dient. Das Ziel der Vereinbarung bestand in einer weitestgehenden Portabilität von Programmen fuer den „Z 1013“, auch wenn diese z.B. einen Sprung zu einem Druckertreiber enthalten sollen. Der Verteiler beginnt auf der Adresse FFFFH abwärtsführend und enthaelt Sprungbefehle zu den entsprechenden Unterprogrammen der Systemerweiterung, welche mit einem RET-Befehl enden muessen, um die Rueckkehr in das Anwenderprogramm zu erreichen. Nicht vorhandene Unterprogramme sind durch RET (C9H) kurzzuschliessen.

Sprungverteiler fuer Z 1013-Erweiterungen:

Adr.	Aufruf	Bemerkung
FFFDH	JMP INKEY	holt ein Zeichen von Tastatur in den Akku; kommt beim 2. Aufruf nur zurueck, wenn Taste zwischendurch losgelassen wurde
FFFAH	JMP POLL	bringt immer ein Zeichen im Akku zurueck, egal ob Taste losgelassen wurde oder nicht
FFF7H	JMP STAT	uebergibt Tastaturstatus im Akku A=0 - keine Taste gedruickt A=FFH - Taste gedruickt die Abfrage erfolgt ohne Ruecksicht, ob die Taste schon vor dem Aufruf gedruickt war und hinterlaesst trotz gedruickter Taste den Status 'letztes Zeichen war 0' ((Zelle 4)=0) um eine evtl. nachfolgenden INKEY-Routine nicht zu sperren
FFF4	JMP SARUF	ruft die SAVE-Routine des Headersave ! zerstoert 1. Registersatz + AF' Parameteruebergabe: Zellen 1BH - anfad., 1DH - endadr., 23H - strtadr. Akku 3AH - Wiederholen der SAVE-Funktion mit gleichem Kopf H(IY) Typvorgabe (in ASCII), sonst 0
FFF1H	JMP LORUF	ruft LOAD-Routine des Headersave ! zerstoert 1. Registersatz + AF' Parameteruebergabe: Zellen 1BH - neue Anfangsadresse des Files sonst 0 Akku 0 - ohne signifikante Kontrolle, 4EH - mit signifikanter Kopfkontrolle (Typ) + Namenabfrage H(IY) 0 - Typ wird abgefragt H(IY) = Typkennzeichen (in ASCII) - keine Typabfr. L(IY) 20H - Freigabe Autostart bei COM-Files
FFEEH	JMP ZMINI	Initialisierung der Z-Monitorrufe auf B0H
FFEBH	JMP DRDEL	setzt den logischen Druckertreiber zurueck
FFE8H	JMP DRAKK	uebergibt den Akkuinhalt an den logischen Druckertreiber
FFE5H	JMP BSDR	druckt den Inhalt des BWS und kehrt in das rufende Programm zurueck
FFE2H	JMP HARDC	uebergibt den Akkuinhalt an logischen Druckertreiber wenn ein Flag im Rechner gesetzt ist; wandelt CR (0DH) in NL (1EH/0DH-0AH) ! nur verwenden, wenn Programm eine eigene Bildschirmverwaltung hat

Adr.	Aufruf	Bemerkung
FFDFH	JMP DRZEL	wie DRAKK, nur das der Inhalt von 1BH uebergeben wird (vorgesehen, um im BASIC mit POKE zu drucken)
FFDCH	JMP BEEP	erzeugen eines kurzen Signals
FFD9H	JMP ASTA	Ausgabe Akkuinhalt als ASCII-Zeichen an PUNCH
FFD6H	JMP BSTA	Ausgabe Akkuinhalt als Byte an PUNCH
FFD3H	JMP AIN	Eingabe eines ASCII-Zeichens vom LBL in den Akku
FFD0H	JMP BIN	Eingabe eines Bytes vom LBL in den Akku
FFCDH	JMP DRINI	Initialisierung des logischen Druckertreibers
FFCAH	JMP ZEIDR	uebergibt ein Zeichen im Akku an physischen Druckertreiber
FFC7H	JMP BLMK	Lesen eines Blocks vom Headersave Parameteruebergabe: Zellen 25H/26H * Kopfinhalt des zu lesenden Bl. HL * Ladeadresse des Blocks Return: Zellen 25H/26H * Kopfinhalt + 20H HL * HL:=HL+20H Abbruch des Lesens bei Kopfinhalt=0FFFFH oder DMA > Endadr in ARG2 (1DH)
FFC4H	JMP BSMK	Schreiben eines Blocks im Headersave Parameteruebergabe: HL * Quelladresse Block IX * Kopfinhalt DE * Anzahl der Sync.-Bits Return: HL * HL:=HL+20H
FFC1H	JMP SUCHK	Suchen eines Kopfblocks und Uebergabe des Inhalts im Kopfpuffer (E0-FF), keine Auswertung
FFBEH	JMP AKP	Aufbereitung Kopfpuffer mit Namenabfrage Parameteruebergabe wie bei SARUF
FFBBH	JMP GETST	Abfrage der Joysticks und Uebergabe des Ergebnisses in BC (B-links,C-rechts) mit folgenden Bit-Bedeutungen (Belegung mit 1): Bit 0 - links 1 - rechts 2 - runter 3 - hoch 4 - Aktionstaste Z-Flag=1, wenn keine Betaetigung vorliegt CY-Flag=1, wenn Spielhebel nicht angeschlossen
FFB8H	JMP SOUND	Ausgabe einer vollen Periode auf die Tonbandbuchse, sowie auf Bit 7 vom Systemport (User-P) Uebergabe der Periodendauer in C mit $T=n*33\mu s+20\mu s$ (2MHz)

Systemzellen

ADR	BYTE	FUNKTION	ZUSATZFKT.
0000	3	Sprung zum Warmstart	
0003	1	Befehlscode bei RST 20H	Geraetezuweisg. (I/O-Byte)
0004	1	letztes Zeichen Tastatur	
0005	3	CALL 0005H	
0008	3	RST 8H	
000B	2	Breakpointadr.	

ADR	BYTE	FUNKTION	ZUSATZFKT.
000D	3	Operandenfolge bei Breakpoint	
0010	3	RST 10H	
0013	2	Adr. bei INHEX	
0015	1		Merkzelle HSave; (SHILO)
0016	2	SOIL	
0018	3	RST 18H	Sprung zu Monitor 2/3
001B	2	ARG 1	
001D	2	ARG 2	
001F	1	vom Cursor verdecktes Zeichen	
0020	3	RST 20H	
0023	2	ARG 3	
0025	2	Cursorpos. nach Kommandoknv.	
0027	1	ASCII/Graphikschalter	
0028	3	RST 28H	
002B	2	akt. Cursorposition	
002D	3	BER 3 frei	2D:Zeilenzaehler 2E:max. Zeilenzahl 2F:Seitenlaenge
0030	3	RST 30H	
0033	2	Kenntonlaenge	
0035	3	Tastaturcodetab.	
0038	3	RST 38H	
003B	12	Tastaturcodetab.	
0047	2	WINDOW-Laenge	
0049	2	-„- -Anfang	
004B	2	-“- -Ende	
004D	2	Reg.retteber.:HL'	
004F	2	-„- DE'	
0051	2	-“- BC'	
0053	2	-„- AF'	
0055	2	-“- HL	
0057	2	-„- DE	
0059	2	-“- BC	
005B	2	-„- AF	
005D	2	-“- IY	
005F	2	-„- IX	
0061	2	-“- PC	
0063	2	Stackpointer(90H nach RESET)	
0065	1	BER 1 frei	Spaltenzaehler
0066	3	NMI	
0069	1	Merkz. fuer NEXT (gegen EI getauschtes Byte)	
006A	2	SP-Zwischenspeicher	
006C	2	Zwischenspeicher bei FIND	
006E-008FH		Anwenderstack	
0090-00AFH		Systemstack	

ADR	BYTE	FUNKTION	ZUSATZFKT.
00B0-00DFH		frei fuer Kommandoschleife	
00E0-00FFH		Puffer fuer Header S/L	

I/O-Adressen

Auch bei den I/O-Adressen wurde neben den vom Hersteller festgelegten Adressen einige mit Funktionen belegt und als Standard für die Anwender vorgeschlagen (1.Tagung in Dresden)

ADR.	D-Bit	I/O-PORT bzw.FUNKTION	BEMERKUNGEN
00H,01H		PIO Z-1013 PORT A	USER für Spielhebel, Drucker usw.
02H,03H		PIO Z-1013 PORT B	02H Daten, 03H Steuerung
PETERS-PLATINE			
04H (-07h)	DB7	UMSCHALTUNG 32/64 BS	Bildschirm von 32 auf 64 Zchn. umschalten, und umgekehrt.
	DB6	UMSCHALTUNG TAKTFREQUENZ	Taktfrequenz kann mit diesem I/O-Signal von 2 auf 4 MHz und umgekehrt umgeschaltet werden.
	DB5	UMSCHALTUNG ZEICHENGENERATOR	Wenn verschiedene Zeichen-Generatoren (z.B. ASCII-Code m.Grafikzeichen und DIN-Zeichensatz-m.Umlaute und SZ, sowie gesetzten 7.Bit /invers)
	DB4	ROM-ABSCHALTUNG	Wenn z.B.externer ROM verwendet werden soll, oder ein Urlader.
	DB3	freiprogrammierbarerZeichengenerator	
	DB2 u. DB1	Schreibschutz fuer 4K-RAM-Bereich (z.B. Urlader)	DB2: RAM-Bereich F000H-F7FFH, DB1: F800H-FFFFH
	DB0	frei	freihalten fuer wichtige Verwendung.
	rueckgesetztes Datenbit ist normaler Zustand (Z1013-Original), dabei ist der Schreibschutz eingeschaltet. Bitte bei Anwendung in Programmen nur die jeweils benoetigten Bits setzen oder ruecksetzen. !! NICHT GENERELL '0' BEIM RUECKSETZEN VERWENDEN !!		
05H	DB0-7	frei fuer Anwender	
08H-0Fh		TASTATUR-SPALTEN-TREIBER	
FUER ERWEITERUNGEN:			
E/A-Modul Riesa			
30H,31H		PIO 1 PORT A	Fuer EPROM-Programmierung u.a.
32H,33H		PIO 1 PORT B	
34H,35H		PIO 2 PORT A	V.24-Interface (Riesa)
36H,37H		PIO 2 PORT B	frei
CTC-Modul			

ADR.	D-Bit	I/O-PORT bzw.FUNKTION	BEMERKUNGEN
38H,39H,3AH,3BH		CTC 1, 4 KANAELE	(Bei eigener CTC Installation, 1. CTC als Systemuhr vorgeschlagen)
3CH,3DH,3EH,3FH		CTC 2, 4 KANAELE	
98H		RAM-FLOPPY A	(Adressen fuer 2 RAM-Floppy)
58H		RAM-FLOPPY B	
FUER ROM-FLOPPY STEHT DIE ADRESSE NOCH NICHT FEST, WIRD ABER WIE RAM-FLOPPY ANGESTEUERT, LIEGT ALSO ETWA IM GLEICHEN BEREICH.			
D0H,D1H,D2H		(nach Kramer)	
F0H,F1H,F2H		(nach Brosig)	fuer DISK-FLOPPY genutzt.

Standard-Adressen fuer Vollgrafik festzulegen, hält die IG-HC noch fuer verfrüht, da noch kein für alle akzeptables System vorliegt.

Ports aufgesammelt

Peters-Platine Port 04h

04TAB:

```
DB 01110000B ;"1" 64x16 Zeichen
DB 10000000B ;"2"
DB 10110000B ;"3" 4 Mhz
DB 01000000B ;"4"
DB 11010000B ;"5" 2.ter Zeichensatz
DB 00100000B ;"6"
```

EPROMMer:

```
;Hardware Eprom-Programmiermodul des CC Leipzig (mit PIO)
PIOAD: EQU 0FCH ;EPROM-DATEN PORT
PIOAC: EQU 0FDH ;STW-DATENPORT
PIOBD: EQU 0FEH ;EPROM-STEUERPORT
PIOBC: EQU 0FFH ;STW-STEUERPORT
;
;Hardware Eprom-Programmiermodul IGD ueber PIO-Modul
;PIOAD EQU 030H ;EPROM-DATEN PORT
;PIOAC EQU 031H ;STW-DATENPORT
;PIOBD EQU 032H ;EPROM-STEUERPORT
;PIOBC EQU 033H ;STW-STEUERPORT
```

NANOS-Ram-Disk

Adresse	Wert
0D821H DISK A	DB RAMDI 0C4H
0D822H	DB READDI 0C6H
0D823H	DB LDAH 0C0H

0D824H	DB LDBB	0C2H
0D825H	DA WINDOW	0F700H

0D827H DISK B	DB RAMDI	0C4H
0D828H	DB READDI	0C6H
0D829H	DB LDAH	0C0H
0D82AH	DB LDBB	0C2H
0D82BH	DA WINDOW	0F700H

CCJena-Floppy ??

CFDC:	EQU	94H	;STEUERUNG FDC
DFDC:	EQU	95H	;DATEN FDC
TC:	EQU	92H	;TERMINAL COUNT (ENDE-IMPULS)

???

CFDC:	EQU	7CH	;STEUERUNG FDC
DFDC:	EQU	7DH	;DATEN FDC
TC:	EQU	78H	;TERMINAL COUNT

NANOS ???

;FDC-Adressen			
;			
CFDC:	EQU	0F0H	;Steuerung
DFDC:	EQU	0F1H	;Daten
TC:	EQU	0F8H	;Terminal count
MON:	EQU	0F6H	;Motor ein
MOFF:	EQU	0F2H	;Motor aus
RFDC:	EQU	0FAH	;Softreset
;			

CCJena GDC

Port 18H

ist aber einstellbar:

Dil-1	=	OUT 0,1
2	=	8,9
3	=	10,11
4	=	18,19 ---> Stellung f}r BIOS und Urlader
5	=	20,21
6	=	28,29
7	=	30,31

BIOS Cottbus

; RAM - FLOPPY GRUNDADRESSEN

```

GADDA EQU 98H ;GRUND-ADR. 1 RAM-FLOPPY
GADDB EQU 58H ;GRUND-ADR. 2 RAM-FLOPPY
GADDC EQU 68H ;GRUND-ADR. 3 RAM-FLOPPY
GADDD EQU 88H ;GRUND-ADR. 4 RAM-FLOPPY
RAFDDR EQU 88H ;GRUND-ADR. DER RAF
VIS EQU 0E0H ;I/O-ADRESSE VIS 3 A
BAGDC EQU 18H ;I/O-ADRESSE GDC-KARTE 18
oder BAGDC EQU 20H ;I/O-ADRESSE GDC-KARTE 20
STGDC EQU BAGDC
RDGDC EQU BAGDC+1
WDGDC EQU BAGDC
WCGDC EQU BAGDC+1

PIOD EQU 34H ;ADR.V.24 E/A-MOD.
PIOC EQU 35H

;FDC
CFDC EQU 0F0H ;STEUERUNG FDC
DFDC EQU 0F1H ;DATEN FDC
MOAUS EQU 0F2H ;FDC-PORT-ADRESSE
MOEIN2 EQU 0F4H ;2 PHYSISCHE FLOPPY
MOEIN EQU 0F6H
TC EQU 0F8H
FDCRES EQU 0FAH
odwr
CFDC EQU 7CH ;STEUERUNG FDC
DFDC EQU 7DH ;DATEN FDC
TC EQU 78H ;TERMINAL-COUNT
RESFDC EQU 7AH ;RESET FDC

CTC1 EQU 38H ;CTCUHR- KANAL 1
CTC2 EQU 39H ;CTCUHR- KANAL 2
CTC3 EQU 3AH ;CTCUHR- KANAL 3

```

From:

<https://hc-ddr.hucki.net/wiki/> - Homecomputer DDR

Permanent link:

https://hc-ddr.hucki.net/wiki/doku.php/z1013/technische_daten?rev=1283845509

Last update: 2010/09/06 22:00

