

SYSTEM-EQUALS

FÜR MUPID 320AF

Table of contents

1.	System-Equals	1-1
1.1.	Adressliste	1-1
1.2.	Routinen	1-7
1.2.1.	ABORT	1-7
1.2.2.	ADDPJ	1-7
1.2.3.	BASIN	1-8
1.2.4.	BRANK	1-8
1.2.5.	BS	1-8
1.2.6.	BSET0	1-8
1.2.7.	BSET1	1-8
1.2.8.	BSET2	1-9
1.2.9.	BSET3	1-9
1.2.10.	BUENI	1-9
1.2.11.	CHSCREEN	1-9
1.2.12.	DOTINV	1-9
1.2.13.	EXNMI	1-10
1.2.14.	HT	1-10
1.2.15.	INTSO	1-10
1.2.16.	KADD	1-11
1.2.17.	OTHDEF	1-11
1.2.18.	PROGEX	1-12
1.2.19.	RST 1	1-12
1.2.20.	RST 2	1-12
1.2.21.	RST 3	1-12
1.2.22.	RST 4	1-13
1.2.23.	RST 5	1-13
1.2.24.	RST 6	1-13
1.2.25.	SCRLUP	1-13
1.2.26.	SCRUEL	1-13
1.2.27.	SP1ACT	1-14
1.2.28.	SWINI	1-14
1.2.29.	SWRES	1-14
1.2.30.	UMWAND	1-14
1.2.31.	WARTE	1-15
1.2.32.	XST2A	1-15
1.2.33.	XST3A	1-15
1.2.34.	XST4A	1-15
1.2.35.	Control Characters	1-16
1.2.36.	SW-Attribut-Tabelle	1-16

1. System-Equals

1.1. Adressliste

Adresswerte hexadezimal, Beschreibung der Routinen s. unter "Routinen". Adressen, falls nicht anders angegeben, in Bank 1.

ABORT	0118	Routine
ADDPI	F505	Erweiterungsbuffer (RTX-Input) Anfangsadresse
ADDPL	0400	Erweiterungsbuffer (RTX-Input) Länge
AP10I	70C3	Auslesepointer BTX Input
AP10O	70C7	Auslesepointer BTX Output
AP12I	70C5	Auslesepointer Keyboard Input
AP12O	70C9	Auslesepointer Keyboard Output
AUTOSC	07	Autoscroll-Flagge in FLAGG
AHSE	05	Select-Status Bit umschalten Synchronisation
BANK	05	Select-Status Bit Memory 2. Bank
BASIN	014B	Basic-Interpreter Start
BBANK	0112	Routine
BILDAB	07	Select-Status Bit Bildabschaltung
BLACK	00	Leerer Bildpunkt
BS	011B	Routine
BSET0	0106	Routine
BSET1	0102	Routine
BSET2	010C	Routine
BSET3	010F	Routine
BTX	10	Portnummer RTX

BTXA 0016 Default-Wert in UZ

LHLD BTXA

SHLD UZ

BTXAD 70D5 Feld für Adresse der Interruptroutine für Zeichen von BTX. 'RET' = LXI HL, BTXC

DCR M

dekrementiere BTXC

JMP OTHDEF

BTXC 70D1 Aktivierungsflagge BTX.

wenn 0 und Zeichen von BTX:

Zeichen über EP10I einspeichern

BTXC inkrementieren

verzweigen zu (BTXAD)

wenn 1. und Zeichen von BTX:

Zeichen über EP10I einspeichern

BTXCI 70D2 Low Byte von BTXC

BTXNR 70F9 Feld für Anfangsadresse Seitenstapel

BUINI 0100 Routine

CHSCREEN 001B Routine

COLOUR 70E1 Konstante, Beginn des Blocks der SW-Attribute
Default Inhalt von YBXS (Aufbau s.
"SW-Attributblock")

Control Characters s. unter "Control Characters" (unten)

CURON 00 Cursor-On Flagge in FLAGG

DBE 03 Select-Status Bit. Default Bildeinstreuung 1... TV
ein/ 0...TV aus

DOMAIN 7040 Speicherstelle, 6 Bytes für 1 PDI-Koordinatenpaar

DOTINV 012A Routine

DSR 07 I/O-Status Bit

DTR 01 I/O-Status Bit

EDITKW 81F0 Speicherstelle Editierkennwort 4 Bytes

EP10I 70B8 Einlesepointer BTX Input

EP10O 70BF Einlesepointer BTX Output

EP12I 70BD Einlesepointer Keyboard Input

EP12O 70C1 Einlesepointer Keyboard Output

EXNMI 002E Non Maskable Interrupt (Routine)

GRAM 02 Select-Status Bit graphischer RAM

HT 011E Routine

I/O-Status-bits s. ORDY, IRDY, DSR, DTR

INICOFF 00 Default Column Offset für Bildschirmspeicher (ab 8000)

INILOFF 00 Default Line Offset für Bildschirmspeicher

INTSO 0023 Routine

IRDY 01 I/O-Status Bit

KADD 0115 Routine

KENNW 80F6 Speicherstelle für BTX-Kennwort, Länge LKENNW

KEY 12 Portnummer Keyboard

KEYAD 70D7 Feld für Adresse der Interruptroutine für Zeichen von Keyboard.

RET = LXI HL,KEYC

DCR M

JMP 0THDEF

KEYC 70D3 Aktivierungsflagge Keyboard

Wenn 0 und Zeichen von Keyboard:

Zeichen über EP12I einspeichern

KEYC inkrementieren

verzweigen zu (KEYAD)

Wenn 1 und Zeichen von Keyboard:

Zeichen über EP12I einspeichern

KEYCI 70D4 Low Byte von KEYC

KLN F905 Polygonkantenpuffer Anfangsadresse

LEDITKW 04 Konstante, Länge Editierkennwort

LKENNW 04 Konstante, Länge des BTX-Kennworts

LPERKW 04 Konstante, Länge des pers. Kennworts in BTX

LTLNR 06 Konstante, Länge der BTX-Teilnehmernummer

ORDY 00 I/O-Status-Bit

OTHDEF 0033 Maskable Interrupt Return (Routine)

OTHINT 70DD Feld für Adresse der Interruptbehandlungsroutine für Benutzerinterrupt (zur Zeit nicht verwendet, freies Pin) . default: OTHDEF.

OVI 70CB Adressfeld (2 Byte) für Pufferüberlaufbearbeitung (kommt zurück mit JMP OTHDEF)

PERKW 80FA Speicherstelle für persönliches Kennwort, Länge LPERKW

PFARB 712F Pointer auf Farbtabelle (bei punktissem Facsimile)

Port-Nr. s. BTX, KEY, SELECT

Postdictionary-Verweise:

BASIC 1

BASIO 2

INDEX 3
 ASSEMB 6
 EDITOR 7
 PASCAL 8

PL10I 0A BTX-Input Puffer Standardlänge
 PL100 0A BTX-Output Puffer Standardlänge
 PL12I 0A Keyboard Input Puffer Standardlänge
 PL120 0A Keyboard Output Puffer Standardlänge
 PODICA 70FB Speicherbereich mit der PostdictionarySeitennummer,
 (8 Byte) default *1501\$ (Österreich)
 PROGBA 70F7 Feld für Programmbasis bei Laden von Code mit Si V.
 PROGEX 0006 PROM Adresse (Routine)
 PROTAB 7103 Programmtabelle (20 Bytes)
 PROTL 10 Protablänge (konstant)
 PU10I 708F BTX-Input-Puffer Anfangsadresse
 PU100 7099 BTX-Output-Puffer Anfangsadresse
 PU12I 70A3 Keyboard-Input-Puffer Anfangsadresse
 PU120 70AD Keyboard-Output-Puffer Anfangsadresse
 QUICK 01 Select Status Bit Refresh ein
 R75CLK 70D9 Uhr
 RAM0 7000 Anfangsadresse des freien RAMs für Systemvariable,
 Puffer, Pointer und Stack
 SCRLUP 0121 Routine
 SCRDEL 0124 Routine
 SELECT 16 Internes Output-Port. Verwendung: Zusammenstellen
 Select-Status-Bits , auf ST16 speichern, OUT SELECT

Select-Status-Bits s. VC, QUICK, GRAM, DBE, VC2, BANK, AHSE,
BILDA9

SEND4 000E Default Adresse für Starten Protah-Programme L4LD
SEND4 SHLD PROTAB(1)

SESTA 88FF Bank 0 !!!, Seitennummernstapel Anfang

SP1ACT 0026 Routine

STAP1 71C0 Anfangsadresse des freien RAMs (bis ca. 73C0)

ST10I 70B8 BTX-Input-Statusbyte (Bits s. I/O-Statusbits)

ST100 70B7 BTX-Output-Statusbyte

ST12I 70BA Keyboard-Input-Statusbyte (nicht verändert ??)

ST120 70B9 Keyboard-Output-Statusbyte

ST16 70CD Kopie des Zustands des Status-Ports

ST16S 70CE Sicherung von ST16

STACK0 7400 Initialisierungswert für Stackpointer

SWINI 014E Routine

SWRES 0148 Routine

SWTIN 70DB Feld für Adresse der Time-Overflow-Routine. Darf
nur im Speicherbereich unter 8000H liegen !
default: EXNMI innerhalb einer eigenen Routine sind
nur HL und AF verwendbar. 'RET' = JMP EXNMI

TELNR 81F4 Speicherstelle für Telefonnummer, bis max. 82FF,
beendet durch 00

TLNR 80FD Speicherstelle für Teilnehmernummer, Länge LTLNR

UMWAND 012D Routine

U7 70CF Adressfeld, wohin springen, wenn 'K' gedrückt ?
default: BTXA, 'RET': JMP OTHDEF

VC 00 Select-Status-Bit Video-Conv
VC2 04 Select-Status-Bit (freier Pin)
WART 002B Routine
XST2A 0013 Routine
XST3A 000B Routine
XST4A 0003 Routine
YBXS 712B Pointer auf Bildschirmparameterblock default für
COLOUR
YWERT EF Offset des letzten Punktes einer Punktspalte im
BS-Bereich
ZEND 1C Bildschirm Zeilenende

1.2. Routinen

1.2.1. ABORT

Funktion: Auslesen BTX-Input-Puffer, Löschen Bildschirm und
ausgehen "Fehler bei Übertragung" auf Bildschirm.

Veränd. Reg.: AF, HL, DE, BC

IY wird verwendet

Achtung: BTXC muß ungleich 0 sein Bank 0 wird selektiert

Aufruf: CALL ABORT

1.2.2. ADDPU

Funktion: Erweitern PU101 (BTX-Input Puffer) auf F505 - F904

Veränd. Reg.: AF, DE, HL

Aufruf: CALL ADDPU

1.2.3. BASIN

Funktion: starten Basic-Interpreter
Veränd. Reg.: alles
Aufruf: JMP BASIN

1.2.4. BBANK

Funktion: Selektieren Bank 0 oder Bank 1 nach Zero-Flag
Eingabe: Zero-Flag: 0 ... Bank 1, 1... Bank 0
Ausgabe: keine
Veränd. Reg.: A
Aufruf: CALL BBANK
Achtung: Dieser Aufruf muß auf einer Adresse unter 8000H erfolgen
!

1.2.5. BS.

Funktion: Ausführung eines "Backspace" am Bildschirm.
Veränd. Reg.: AF, HL, DE, BC
Aufruf: CALL BS
Achtung: IY wird verwendet

1.2.6. BSET0

Funktion: Selektieren Bank 0
Veränd. Reg.: AF
Aufruf: CALL BSET0
Achtung: Dieser Aufruf muß auf einer Adresse unter 8000H erfolgen
!

1.2.7. BSET1

Funktion: Selektieren Bank 1
Veränd. Reg.: AF
Aufruf: CALL BSET1
Achtung: Dieser Aufruf muß auf einer Adresse unter 8000H erfolgen
!

1.2.8. BSET2

Funktion: Selektieren Bank 2

Veränd. Reg.: AF

Aufruf: CALL BSET2

Achtung: Dieser Aufruf muß auf einer Adresse unter 8000H erfolgen
!

1.2.9. BSET3

Funktion: Selektieren Bank 3

Veränd. Reg.: AF

Aufruf: CALL BSET3

Achtung: Dieser Aufruf muß auf einer Adresse unter 8000H erfolgen
!

1.2.10. BUINI

Funktion: Initialisieren alle APxxx, EPxxx, PUxxx auf
Standardgröße.

Veränd. Reg.: A, HL, DE, BC

Aufruf: CALL BUINI

1.2.11. CHSCREEN

Funktion: Einzelzeichen Anzeige

Eingabe: C: das Zeichen

Ausgabe: keine

Veränd. Reg.: AF, HL, DE, BC, IX, IY

Aufruf: MVI C, zeichen

CALL CHSCREEN

Achtung: Nur am Bildschirm darstellbare Zeichen ! (und
Cursorbewegungen)

1.2.12. DOTINV

Funktion: invertiere Bildpunkt am Bildschirm

Eingabe: BC: X-Wert zwischen 0 und 319 E: Y-Wert zwischen 0 und
239

Ausgabe: keine

Veränd. Reg.: AF, HL, DE, BC, AF

Aufruf: LXI BC,X-Wert
 MVI E,Y-Wert
 CALL DOTINV

1.2.13. EXNMI

Funktion: Rücksetzen der Register nach Clock-Interrupt ("Non Maskable Interrupt-Return")

Clock-Interrupt: R75CLK springt auf 0

Aufruf: JMP EXNMI

Veränd. Reg.: AF, HL (in dieser Reihenfolge am Stack)

Beispiel: ...
 LXI HL,INTERRUPT
 SHLD SWTIN

...
 INTERRUPT

Routine zur Behandlung
 benutzbar: AF, HL !

...
 JMP EXNMI

EXNMI dient als 'RET' der, eventuell vom Benutzer definierten Interruptbehandlung, deren Adresse in SWTIN steht.

1.2.14. HT

Funktion: Ausführen eines "Horizontal Tabulation"

Veränd. Reg.: AF, HL, DE, BC

Aufruf: CALL HT

Achtung: IY wird verwendet

1.2.15. INTSO

Funktion: Zeichenbuffer ausgeben auf Bildschirm

Eingabe: Puffer mit geändertem AP10I

Ausgabe: keine

Aufruf: DI

LDED AP10I
 LYI HL,bufferanfangsadresse
 SHLD AP10I
 LXI HL,bufferendeadresse+1
 MVI M,80H
 INY HL
 MOV M,E
 INY HL

```

MOV M,D
CALL INTSO
Aufruf 2:
LXI HL,bufferendadresse + 1
MVI M,0
LXI DE,bufferanfangsadresse
...
DI
LHLD AP10I          sichern AP10I
SDED AP10I
CALL INTSO
SHLD AP10I
Veränd. Reg.: AF', HL', DE', BC'

```

1.2.16. KADD

Funktion: Kontrollroutine, simuliert alle !K! x -Eingaben außer !K! C

Eingabe: A: Zeichen, z.B. "V" für Verbindungsaufbau

Ausgabe: keine

Veränd. Reg.: AF, HL, DE, BC

Aufruf: MVI A, Zeichen
CALL KADD

1.2.17. OTHDEF

Funktion: Rücksetzen der Register nach Interruptbehandlung (aus Stack)
für Maskable Interrupt

Aufruf: JMP OTHDEF

Veränd. Reg.: AF, HL, DE, BC (in dieser Reihenfolge am Stack)

Anmerkung: OTHDEF dient als 'RET' der, eventuell vom Benutzer gesetzten Interruptbehandlungen, deren Adressen auf IZ, OVI, BTXAD, KEYAD stehen.

Beispiel:

```

...
LXI HL, INTERRUPT
...
SHLD xx          d.h. IZ, OVI, BTXAD, KEYAD
...
INTERRUPT
...
AF, BC, DE, HL verwendbar, alle anderen
Register MÜSSEN
... gesichert werden.
JMP OTHDEF

```

Beispiel 2: Pufferüberlauf soll ignoriert werden. ... LXI
HL, OTHDEF SHLD OVI

1.2.18. PROGEX

Funktion: Beenden eines Benutzerprogramms,
Initialisierung aller
interner Parameter außer TLN-Nummer (TLNR), Seitenstapel
(SESTAY)
PROTAB und Interfaces.
Aufruf: LHLD PROGEX
PCHL
bzw. LD HL, (PROGEX) / LD PC, (HL)

1.2.19. RST 1

Funktion: Zugriff auf Speicherbank 0 (Lesen) (8000 - FFFF)
Eingabe: HL: Adresse
Ausgabe: B: Inhalt des Bytes Adresse in Bank 0
Veränderte Register: AF

1.2.20. RST 2

Funktion: Zugriff auf Speicherbank 0 (Schreiben)
Eingabe: HL: Adresse (zwischen 8000 und FFFF)
B: Wert für Byte
Ausgabe: keine
Veränd. Reg.: AF
Anmerkung: Die Speicherbank 0 enthält den größten Teil des
Bildschirmspeichers, d.h. mit RST 1 und 2 liest bzw.
verändert man
den Inhalt des Bildschirmspeichers.

1.2.21. RST 3

Funktion: Senden 1 Zeichen auf die BTX-Leitung
Eingabe: C: zu sendendes Zeichen in ASCII-Code
Ausgabe: keine
Veränd. Reg.: AF

1.2.22. RST 4

Funktion: Sende 1 Zeichen an die Tastaturschnittstelle
Eingabe: C: zu sendendes Zeichen
Ausgabe: keine
Veränd. Reg.: AF

1.2.23. RST 5

Funktion: Warte und hole ein Zeichen von der BTX-Leitung
Eingabe: keine
Ausgabe: A: empfangenes Zeichen
Veränd. Reg.: AF
Achtung: BTXC muß ungleich 0 sein

1.2.24. RST 6

Funktion: Warte und hole ein Zeichen von der
 Tastaturschnittstelle
Eingabe: keine
Ausgabe: A: empfangenes Zeichen
Veränd. Reg.: AF
Achtung: KEYC muß ungleich 0 sein

1.2.25. SCRLUP

Funktion: Scrollen eines Anzeigefensters am BS nach oben
Eingabe: LINBEG: (erste Zeile - 1)*10+INILOFF
 LINANZ: anzahl der zu scrollenden zeilen - 1
Veränd. Reg.: alle
Aufruf: LIYD YBXS
 MVIY (erste zeile - 1)*10+INILOFF,LINBEG
 MVIY anzahl der zu scrollenden zeilen - 1,LINANZ
 CALL SCRLUP

1.2.26. SCRDEL

Funktion: Löschen einer Zeile am Bildschirm
Eingabe: E: (zeilennummer - 1)*10+INILOFF
Veränd. Reg.: alle
Aufruf: CALL DELLINE mit DELLINE s.u.

```

DELLINE MVI E, (zeilennummer - 1) * 10 + INILOFF
        LDA ST16
        PUSH PSW
        EXAF
        EXX
        PUSH PSW
        PUSH BC
        JMP SCRUEDEL
    
```

1.2.27. SP1ACT

Funktion: Start von BASIO (Basic Input/ Output System)

Veränd. Reg.: alle

Aufruf: LXI HL, PROTAB + BASIO * 2 + 1
 LIXD SP1ACT
 PCIX

1.2.28. SWINI

Funktion: Software Initialisierung, setzt die Systemvariablen, BUINI und ADDPU

Veränd. Reg.: AF, HL, DE, BC

Aufruf: CALL SWINI
 XRA A
 STA BTXC

1.2.29. SWRES

Funktion: Software Reset, Enable BTX-Input-Port, sendet "*", liest danach den BTX-Input-Puffer aus und sendet danach "*" oder "***", sodaß danach ein definierter Status im BTX erreicht wird.

Veränd. Reg.: A, HL, E

Aufruf: CALL SWRES

1.2.30. UMWAND

Funktion: Koordinatenumwandlung

Eingabe: H, L, C, A ... Koordinatenbytes in PDI-Darstellung

Ausgabe: BC: X-Wert zwischen -319 und 319 DE: Y-Wert zwischen -239 und 239

Veränd. Reg.: AF, HL

Aufruf: MVI H, erstes byte
 MVI L, zweites byte
 MVI C, drittes byte
 MVI A, viertes byte
 CALL UMWAND

1.2.31. WARTE

Funktion: N/75 Sekunden warten
 Eingabe: HL: $(-1)*N$
 Ausgabe: keine
 Veränd. Reg.: AF
 Aufruf: LXI HL, n
 CALL WARTE

1.2.32. XST2A

Funktion: Ein Zeichen einspeichern in Puffer
 Eingabe: C: das Zeichen HL: Adresse des Einlesepointers (High Byte)
 Ausgabe: A: Status: 0 .. ok, sonst Überlauf
 Veränd. Reg.: DE, HL
 Aufruf: LXI HL, EPxxx + 1
 CALL XST2A

1.2.33. XST3A

Funktion: Output Port aktivieren
 Eingabe: C: Portnummer 10H oder 12H
 Ausgabe: keine
 Veränd. Reg.: A, C, HL
 Aufruf: MVI C, portnummer
 CALL XST3A

1.2.34. XST4A

Funktion: Ein Zeichen auslesen aus Puffer
 Eingabe: HL: Adresse des Auslesepointers (High Byte)
 Ausgabe: A: das Zeichen oder 0 (kein Zeichen vorhanden)
 Veränd. Reg.: DE, HL
 Aufruf: LXI HL, APxxx + 1
 CALL XST4A

1.2.35. Control Characters

BSCC	08	
CRCC	0D	
ESCCC	1B	
FFCC	0C	
HOMCC	1E	
HTCC	09	
LFCC	0A	
LOCK	CE	Keyboard lock
NODEL	92	No keyboard delay
OFFCC	014	Cursor off
ONCC	015	Cursor on
REMRES	BF	remote reset keyboard
SICC	0F	
SOCC	0E	
VTCC	0B	

1.2.36. SW-Attribut-Tabelle

Name	Offset	Bedeutung
HOLD	00	Hold Graphics, 80 .. ein, 00 .. aus
ATTR	01	Halbton, 80 .. ein, 00 .. aus
GRAPH	02	Schalter Alpha/ Graphikmodus 00 .. Alpha, .FF .. Mosaikgraphik normal, 77 .. Mosaik separiert
NCOL	03	Farbbyte, 1. Hexstelle für 1. Punkt, 2. Hexstelle für 2. Werte: 0 schwarz 1 blau 2 grün 3 cyan 4 rot 5 magenta 6 gelb 7 weiß
SGR	04	80 .. separierte Graphik, 00 .. normal (! vgl. GRAPH !!)
LASTGZ	05	Letztes Graphikzeichen für Hold Graphics, default 20 = Blank
DHSH	06	03 .. doppelte Höhe, 00 .. normale Höhe
FLAGY	07	Flags für gesichert, dichte Info, CONCEAL, Füllkontrolle, Si F
SCAL	08	Bildschirmadresse (Low Byte) für Textcursor
SCAH	09	Bildschirmadresse (High Byte) Bit 7=1 .. Bank 1 (SELECT BANK1), = 0 .. Bank 0 (SET Bit 7)
CODEL	0A	Anfangsadresse für Tabelle der Zeichenmatrizen (low byte)