robotron

Systemhandbuch
MON 8000

Anwendungsbeschreibung
Anleitung
für den Programmierer

Streichhahn

r o b o t r o n

Programmtechnische Beschreibung

fuer Buerocomputer A 5120.16

S Y S T E M H A N D B U C H

M O N 8 0 0 0

Teil 1 - Anwendungsbeschreibung

Teil 2 - Anleitung fuer den Programmierer

Ingenieurhochschule Mittweida
— Sektion Informationselektronik —
9925 Mittweida, Platz der DSF 17
Fernruf 580

VEB Robotron-Buchungsmaschinenwerk
Karl-Marx-Stadt 1985
Stand 12/85

Jens Krause
Am Försterweg 32
O-1260 Strausberg

Die vorliegende Dokumentation entspricht dem Stand 12/85.

Nachdruck, jegliche Vervielfaeltigung oder Auszuege daraus sind unzulaessig.

Im Interesse einer staendigen Weiterentwicklung werden alle Leser gebeten, ihre Vorschlaege bzw. Hinweise dem

VEB Robotron-Buchungsmaschinenwerk
9010 Karl-Marx-Stadt
Annabergerstr. 93

mitzuteilen.

Fuer das Betriebssystem MUTOS 8000 wurden folgende Dokumentationen erarbeitet:

- Anwendungsbeschreibung

- Systemhandbuch MUTOS 8000 Teil I,
  . Abschnitt 1 Kommandos
  . Abschnitt 2, 3 Systemrufe, Bibliotheksfunktionen
  . Abschnitt 4, 5, 7, 8 Special Files, Fileformate, Makropakete, Systemunterstuetzung

- Systemhandbuch MUTOS 8000 Teil II,
  . Abschnitt 1 Kommandointerpreter Shell
  . Abschnitt 2 Editor
  . Abschnitt 3 Testhilfsprogramm adb

- Systemhandbuch MUTOS 8000 Teil III,
  . Abschnitt 1 Textverarbeitung nroff

- Sprachbeschreibungen
  . Assemblersprachbeschreibungen U8000
  . Sprachbeschreibung C-Sprache

Das Sachwortverzeichnis ist als Anlage in der Anwendungsbeschreibung enthalten.

Zum Programmentwickeln und -Testen fuer den Prozessor U8000 werden folgende Dokumentationen angeboten:

- Systemhandbuch MON8000
- CROSS-Software U8000 zum Betriebssystem UDOS
- Assembler U8000ASM
- Binder ZLINK
- Absolutbinder IMAGER
- Preprozessor INCLUDE

Inhaltsverzeichnis

1. Anwendungsbeschreibung MON8000

1.1. Zweckbestimmung

Das MON8000-System ist ein Testsystem fuer Software, die fuer Geraete mit dem Prozessor U8000 enwickelt werden soll. Es erfuellt wichtige Leistungen eines Operationssystems.

1.2. Anwendungsbedingungen

1.2.1. Hardware

Fuer die Anwendung des MON8000-Systems ist als technisches Mittel der BC A5120, durch einen entsprechenden Erweiterungsmodul (EM) zum 16-Bit-Programmentwicklungsplatz A5120.16 ergaenzt, einzusetzen.
Die Erweiterungsmodule ermoeglichen durch unterschiedliche CPU-Versionen (U8001/U8002) und RAM-Groessen (64K/256K) eine gute Anpassung an die verschiedensten Anwendungsfaelle.

1.2.2. Software

Das MON8000-System hat folgende Bestandteile:

U880: - UDOS 1526 (logisches System)
- MON8000.8
  * I/O-Prozessor (Wurzel)
  * UDOSRQ (Modul)
  * USERRQ (Anwendereigene U880-Module)
- LOADEM (Lade-Programm fuer MON8000.EM / U8000-Programme)

U8000: - MON8000.EM

1.3. Beschreibung der Loesung

Die Unterstuetzung des Anwenders von MON8000 ist durch beide Prozessoren (U880,U8000) in folgenden Punkten gewaehrleistet:

1.3.1. Nutzung des U880 fuer MON8000

Entsprechend der Hardwareloesung hat der EM64/256 keinen direkten Zugriff auf den U880-Bus, so dass alle notwendigen I/O-Operationen des U8000 indirekt unter Steuerung des U880 ablaufen muessen.
Alle I/O-Forderungen werden UDOS-typisch realisiert. Die Nutzung des U880 als I/O-Prozessor ist so konzipiert, dass jeweils nur ein Request bearbeitet werden kann.

Die wichtigsten Funktionen von MON8000.8 sind:

- Anfangsladen/Restart des MON8000
- I/O-Prozessor des MON8000
- UDOS-Request nach Anfoderung von U8000-Programmen
- Transfer-Operationen zwischen U880-RAM und EM-RAM fuer spezielle Anwenderprogramme (USERRQ)

1.3.2. Nutzung des U8000 fuer MON8000

MON8000.EM besteht aus:

- Monitor
- spezielle Routinen fuer I/O-Verkehr mit U880

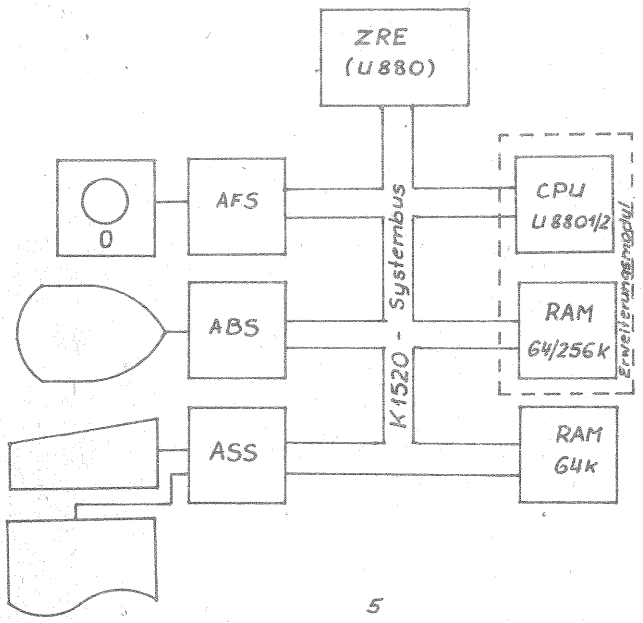
Nach Neustart des MON8000 wird automatisch ein Speichertest ausgefuehrt. Anschliessend wird der Monitor-Status erreicht. Mit dem Monitor koennen alle Kommandos zur Programmentwicklung und zum Programmtest ausgefuehrt werden, wie z.B.:

- Laden und Starten von Programmen
- Anzeige und Aendern von Speicherinhalten und Registern
- verschiedene Funktionen fuer den Programmtest (GO,BREAK, NEXT,...)

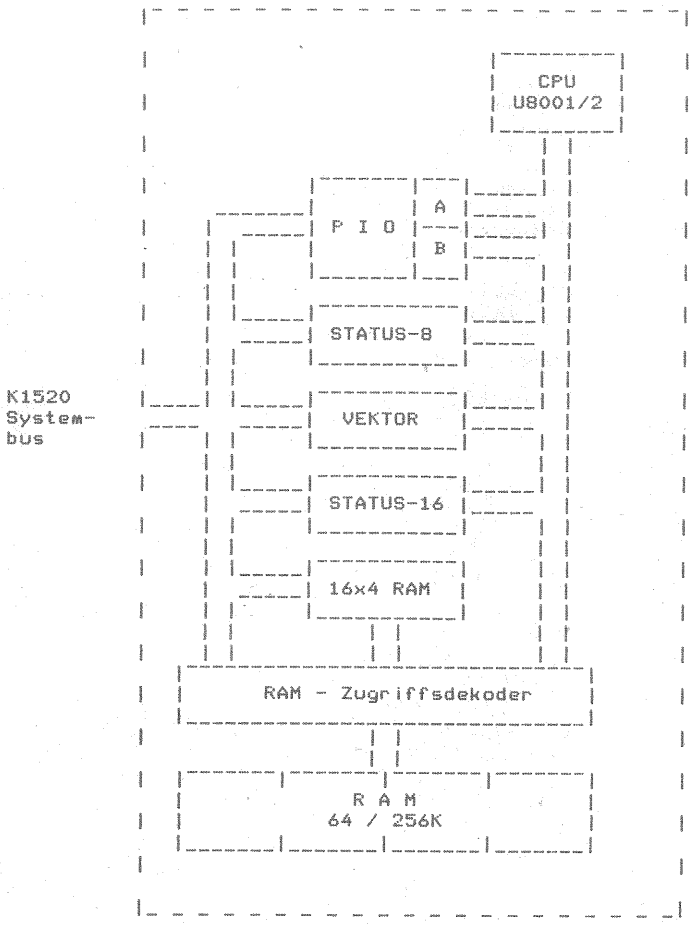
Fuer die einfache Nutzung der U880-Peripherie sind umfangreiche System-Rufe realisiert. Diese ermoeglichen bei entsprechendem Einbinden in Anwenderprogramme eine echte Parallelarbeit beider Prozessoren.

1.4. Hardware - Uebersicht

1.4.1. BC A5120.16



1.4.2. Erweiterungsmodul (EM64/256)

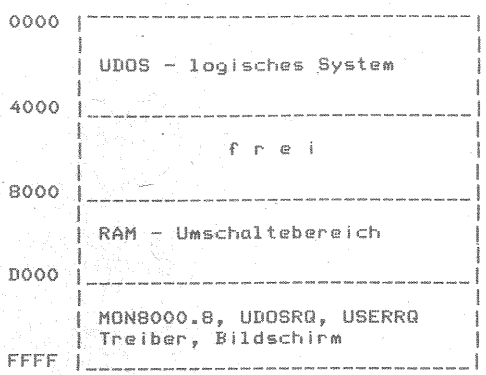


2. Anleitung fuer Programmierer

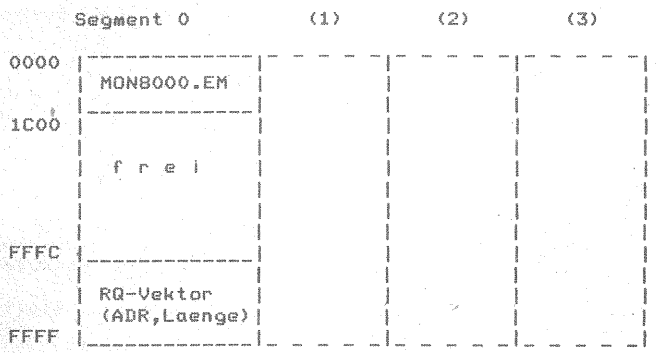
2.1. Systemdaten

2.1.1. Speicherbelegung

U880-RAM (64 K)



EM-RAM (64/256 K)



2.1.2 I/O-Belegung fuer EM64/256

Der Erweiterungsmodul belegt 8 hintereinanderliegende I/O-Adressen. Ueber Wickelbruecken laesst sich die Moduladresse (MODADR) einstellen.
Bei MON8000 ist A8 als MODADR eingestellt.

Es sind folgende I/O-Adressen vorhanden:

MODADR	PIOA	- Daten (Bit-Modus)
MODADR+1	PIOB	- Daten (Bit-Modus)
MODADR+2	PIOA	- Steuerwort
MODADR+3	PIOB	- Steuerwort
MODADR+4	STATUS-8	- ein Byte U880 ==) U8000
MODADR+5	VEKTOR	- ein Byte U880 ==) U8000 (fuer VI)
MODADR+6	STATUS-16	- ein Byte U880 (== U8000
MODADR+7	16x4 RAM	- 4K-Umschaltung (virtuelle ADR)

2.2. Lademoeglichkeiten

2.2.1. LOADEM

LOADEM ist unter UDOS zum Laden von Prozedurfiles auf den EM64/256 nutzbar.

```
%LOADEM filename (segment-nr.)
```

Ist keine Segment-Nr. angegeben, wird auf das Segment 0 geladen.

2.2.2. MON8000.8

Durch das Kommando

```
%MON8000.8
```

wird MON8000.8 (U880) geladen.
Wird das Kommando ohne Restart-Option gegeben, erfolgt durch MON8000.8 (LOADEM MON8000.EM) das Laden des U8000-Monitor-Programms. Die Schnittstelle zwischen U880 und EM64/256 wird initialisiert.
Anschliessend erfolgt durch ein U8000-Reset der Start des MON8000.EM.

Bei Restart des MON8000 durch das Kommando

```
%MON8000.8 R
```

wird nur MON8000.8 geladen. Der EM64/256 befindet sich im gleichen Zustand wie zum definierten Systemaustritt (siehe "E"-Kommando).

2.3. Nutzung des U880 - I/O-Prozessors

2.3.1. Realisierungshinweise

notwendiger Speicherbereich des MON8000.8 :
D000...(MON8000.8.OBJ + UDOSRQ.OBJ + USERRQ.OBJ)
RAM-Umschaltebereich (keine Interrupt-Routinen, usw.) :
8000...CFFF
notwendige Objekte :
MON8000.8.OBJ
UDOSRQ.OBJ
USERRQ.OBJ
notwendige Definitionen in USERRQ des Anwenders :

```
global     USERRQ              ;Anwender-Routine
external   IOBER               ;Eingabebereich
```

genutzte ISR-Vektor-Adressen :
OFEA - ISR-ADR EM-PIOA (INT-16)
OFEC - ISR-ADR EM-PIOB (/PER)
bei "Systemabsturz" des MON8000 infolge Programmtest ist der Austritt aus MON8000 durch Betaetigen der ESC-Taste moeglich (Neustart notwendig).
zur Unterscheidung der Forderungsarten werden vom U8000 ueber EM-PIOA Bit0...2 folgende Kennungen uebergeben:

0 - Monitorbetrieb
2 - Anwenderspezifischer Request (USERRQ)
3 - UDOS-typischer Request (UDOSRQ)
4 - definierter Uebergang in UDOS (Exit)

- die Nutzung des U880 als I/O-Prozessor ist in MON8000 so konzipiert, dass jeweils nur ein Request bearbeitet werden kann.
- die Uebernahme / Uebergabe aller notwendigen Parameter und Daten im Modus 2/3 erfolgt ueber RAM-Umschaltung und /BUSRQ des U880.

2.3.2. Konsolbetrieb

- Eingabe eines Bytes ueber Tastatur
  * Test, ob Zeichen ueber UDOS-INA (0B8E) eingegeben wurde (Polling-Betrieb)
  * Ausgabe des Bytes an EM durch STATUS-8
  * Ausloesen von VI-0 durch VEKTOR
- Ausgabe eines Bytes an Monitor
  * Interrupt von EM durch INT-16 (PIOA4)
  * Uebernahme des Bytes von STATUS-16
  * Ausgabe des Bytes an UDOS-OUTA (0BEB)
  * Fertigmeldung an EM durch VI-2 ueber VEKTOR

2.3.3. UDOS-typische I/O-Nutzung

In diesem Modus wird der I/O-Verkehr ueber UDOS-typische Requestvektoren mit einer speziellen Erweiterung realisiert. Dem U880 wird die Adresse und die Laenge des UDOS-Requestvektors auf dem EM-RAM ((0)FFFC) uebergeben. Ueber RAM-Umschaltungen uebernimmt der U880 den Requestvektor und gegebenenfalls den dazugehoerigen zusaetzlichen Parametervektor (SPV) und auszugebene Daten auf die U880-Seite und fuehrt den geforderten Request aus. Nach Abschluss des Requests werden der aktualisierte Requestvektor und gegebenenfalls Daten auf den EM-RAM uebertragen. War die Operation fehlerhaft, so erfolgt eine Mitteilung auf das Terminal. Ueberschreitet die geforderte Datenlaenge des Requests die maximal festgelegte Transferrate (1000H Byte), so wird der Request in einer Anzahl von Teiloperationen ausgefuehrt.

UDOS-Kommandos koennen ueber LUN 0 abgearbeitet werden. Dabei ist zu beruecksichtigen, dass diese Kommandos nicht den Speicherraum des MON8000.8 ueberschneiden duerfen.

Beachte:

Die wortorientierte Verarbeitung des U8000 bringt Probleme beim Anpassen an den U880-Modus mit sich. Der Anwender muss stets dafuer sorgen, dass Wortdaten, wie z.B. Adressen, in der 8-Bit-typischen Form der L-, H-Vertauschung der Bytes fuer den U880 bereitgestellt werden.

Der U880 als I/O-Prozessor nimmt dieses Bytevertauschen automatisch bei der Uebernahme der Requestvektoradresse und im Requestvektor selbst fuer DTA, DL und SPI vor (Anwender traegt 16-bit-seitig die Original-Wortdaten ein).

Bei der Uebergabe des aktualisierten Requestvektors an den U8000 werden diese Daten 16-bit-typisch dargestellt.

Zum Vereinfachen der Request-Auswertung durch den U880-I/O-Prozessor wurde der Standard-Requestvektor (RQV) erweitert:

```
RQV+0  byte      LUN   logical Unit ,
   +1  byte      RC    Requestcode
   +2  word      DTA   Data Transfer Address
   +4  word      DL    Data Length
   +6  word :=0  CRA   Completion Return Address
   +8  word :=0  ERA   Error Return Address
   +10 byte      CC    Completion Code
   +11 word      SPI   Supplemental Information
   +13 byte      IOI   I/O Information
```

Dabei hat das Zusatzbyte IOI folgende Bedeutung:

```
Bit7...5          nicht belegt
   4              1 - READ-Daten von I/O-Prozessor
                  0 - WRITE-Daten zum I/O-Prozessor
   3              1 - SPI enthaelt Adresse eines Zusatz-
                      parametervektor
                  0 - keine Zusatzparameter
   2              1 - Fuer die DTA des Requestvektor gilt
                      die Segmentnummer der Bits 0 und 1
                      von IOI dls Segmentnummer
                  0 - Segmentnummer des I/O - Rufes ist
                      gueltiges Segment fuer DTA
   1 / 2            - Gueltige Segmentnummer fuer DTA,
                      wenn Bit2 = 1
```

2.3.4. Anwenderprogrammaufruf auf U880-Basis

Um dem Anwender die Nutzung von U880-Routinen fuer spezielle Probleme zu ermoeglichen, wurde dieser Aufruf geschaffen. Dem U880 werden Adresse und Laenge eines beliebigen Datenblocks (max. 1000H Byte) uebermittelt. Dieser Block wird ueber RAM-Umschaltung auf IOBER uebertragen (s.2.2.1.). Durch ein anwenderspezifisches Programm, das unter dem Namen USERRQ in MON8000.8 eingebunden werden muss, kann dieser Datenblock ausgewertet und behandelt werden. Nach Abschluss dieser Routine wird der aktualisierte Datenblock auf den EM-RAM rueckuebertragen.

2.4. Nutzung U8000 (EM64/256)

2.4.1. Kommandointerpreter

Nach Neustart des MON8000 meldet sich der Kommandointerpreter mit:

```
MON8000, Vers. X.Y
```

Anschliessend erfolgt automatisch ein RAM-Speichertest, nach dessen Abschluss sich MON8000 eingabebereit meldet:

```
RAM DIANOSTICS COMPLETE
MAX SEG.....
@
```

- interne Kommandos des MON8000

  EXIT   - Definierter Uebergang in UDOS-Betrieb
  MONITOR - Uebergang in Monitor-Modus
  XEQ   - Starten/Wiederholen des zuletzt geladenen Kommandos

- externe Kommandos des MON8000

  Alle Zeichenketten, die keine internen Kommandos sind, werden als externe Kommandos interpretiert und fuehren zu einem Ladeversuch von Diskette.

     filename ((segment-nr.))

  Laden des Prozedurfiles "filename", Voreinstellen der Register und Start der Prozedur. (Filename kann mit Drive-Nr. angegeben werden, z.B. 1/prog).
  Wird keine Segmentnummer angegeben, so wird auf das jeweils aktuelle Nutzersegment geladen.

     filename ((segment-nr.)),

  Laden des Prozedurfiles "filename", Voreinstellen der Register und Rueckkehr in den Kommandointerpreter.
  Wird keine Segmentnummer angegeben, so wird auf das jeweils aktuelle Nutzersegment geladen.

  Fehler beim Laden externer Kommandos werden im UDOS-typischen Fehlercode angezeigt.
  Eine Uebergabe von Parametern und das Nutzen von Kommandoketten gemaess UDOS-Konventionen ist nicht moeglich.

- Registervoreinstellung

  Alle notwendigen speziellen Register des U8000 werden eingestellt. Die allgemeinen Register R0...R15 sind davon nicht betroffen. Die vorgegebenen Werte sind im Monitor-Modus aenderbar:

  SG - entsprechend angegebener Segmentnummer, sonst keine Aenderung
  PC - entsprechend Entry-Point des Procedurfiles
  FC - %1000 (nicht segmentiert, Normalmodus, Enable vektorisierter Interrupt)
  NS - entsprechend angegebener Segmentnummer, sonst keine Aenderung
  NO - %FFFC (RQV-Adresse und Laenge fuer UDOSRQ bzw. USERRQ)

2.4.2. Monitor des MONB000

Bei Eingabe der Kommandos ist das erste Zeichen signifikant.

Kommandoumfang:

BREAK (adr)(cnt)

Setzen bzw. Loeschen von Unterbrechungspunkten
Das Kommando ohne Parameter bewirkt das Loeschen des BREAK.
Die Angabe eines Schleifenzaehlers (cnt = 0...FF) ist moeglich.

COMPARE (adr1)(adr2)(n)

Vergleich von Speicherbereichen

DISPLAY (adr)(cnt)((longwords words bytes))(L W B)

Anzeige und Veraendern von Speicherinhalten.
Das Lesen bzw. Schreiben einzelner Speicherplaetze ist ohne Angabe von "cnt" moeglich.
Die Eingabe von "-" bewirkt ein Dekrementieren der Adresse.
Durch "Q" wird dieser Modus verlassen.

EXIT Definierter Uebergang zu UDOS 1526

FILL (adr1)(adr2)(data)

Fuellen von Speicherbereichen mit einem Hexa-word.

GO Starten (Sprung) einer Prozedur ab aktuellem PC.

IOPORT (portadr)(W P)

Lesen bzw. Schreiben von/auf E/A-Ports

JUMP (adr)

Sprung zur angegebenen Adresse.

LOAD filename ((segno))

Laden einer Prozedur von UDOS-Diskette in den EM-RAM.

MOVE (adr1)(adr2)(n)

Transfer von Daten zwischen zwei Speicherbereichen.

NEXT (cnt)

Einzelschrittbetrieb

Die Anzahl der abzuarbeitenden Befehle kann durch "cnt"(1...FF) angegeben werden. Nach Aktivieren von NEXT koennen weitere N-Kommandos durch (CR) aufgerufen werden.

QUIT Uebergang in den internen MON8000-Betrieb

REGISTER (registername)

Anzeige bzw. Veraendern von Registerinhalten.

SEND filename beginadr length E=entrypoint

Speicherinhalte des Erweiterungsmoduls werden ab der angegebenen Adresse als Prozedurdatei auf UDOS-Diskette abgelegt.

TEST (R)

Speichertest des EM256.
Bei Angabe von "R" wird der Test laufend wiederholt. Durch Eingabe von (CR) erfolgt der Abbruch des Tests.

VERIFIKATION (R)

Speichertest des EM64 wie bei TEST.

2.4.3. Trap-/Interruptbehandlung

Fuer MON8000 ist ein Trap- und Interruptbehandlungssystem zum Realisieren des I/O-Verkehrs bzw. zur Testunterstuetzung vorhanden:

- Extended Instruction Trap

  Anzeige: nichtimplementierter Kommandocode, Segment und Offset des Auftretens (wird fuer B-Kommando genutzt). Der aktuelle Registerstand wird gerettet und ist im Monitormodus aenderbar.

- Privileged Instruction Trap

  Anzeige: Kommandocode, Segment und Offset des Auftretens, Abbruch des Programms und Rueckkehr in MON8000.

- System Call Trap (SC-Trap)

  Wird zur Nutzung allgemein verwendbarer Routinen des MON8000 benoetigt. Eine Aufstellung der eingebundenen SC-Routinen befindet sich im Anhang A.

- Segment Trap

  Standardfehlerroutine

- NMI

  Anzeige der Adresse des Paritaetsfehlers.

- NVI

  Wird zur Realisierung des N-Kommandos benoetigt.

- VI

  Dient zum Realisieren des I/O-Verkehrs U880-U8000.

  Folgende VI-Werte sind durch MON8000 belegt:

  VI0 - Zeichen des U880 empfangen (GETA)

  VI2 - Zeichen des U8000 vom U880 uebernommen (PUTA)
  (fuer Monitorbetrieb)

  VI4 - Requestanforderungen des U8000 vom U880 empfangen

  VI6 - Request beendet
  (fuer I/O-Verkehr)

  Alle anderen VI sind nicht belegt.

2.4.4. Spezielle Routinen des MON8000

Um eine einfache I/O-Nutzung durch den Anwender, sowie eine echte Parallelarbeit zwischen U880 und U8000 zu gewaehrleisten, sind eine Vielzahl von Systemroutinen nutzbar (s. Anhang A).

Anhang A

Uebersicht anwendernutzbarer Routinen des MON8000

Bezeichnung	SC-Nr.	Parameter	Kurzbeschreibung
GETA	0	Return:r10-ASCII-Zeichen	Einzelzeicheneingabe mit Warten auf Zeichen
PUTA	1	Input:r10-ASCII-Zeichen Return:Z-Flag-Zeichen war <CR>	Einzelzeichenausgabe zum U880
SYSMOD	11		Uebergang des Anwenderprogramms in den Systemmodus
NMOD	12		Uebergang des Anwenderprogramms in den Nutzermode
SGNM	13	Return:r0-Segmentnummer	Bereitstellen der Segmentnummer des rufenden Programms; Segmentnummer wird U8000-typisch dargestellt.
SEGMODE	14		Uebergang des Anwenderprogramms in den Segmentmode
NORMMODE	15		Uebergang des Anwenderprogramms in den Nichtsegmentmode
IOREQ	16	Input:r0-Adr. Datenbereich r1-Datenlaenge r2-Transferkennung	Ausloesen einer I/O-Anforderung mit sofortiger Rueckkehr ins rufende Programm Transferkennung: 2 anwenderspezifische I/O (USERRQ) 3 UDOS-typischer Request (UDOSRQ)

IOWAIT	17		Warten auf Requestendemeldung des U880
IOREWA	18	Input:r0-Adr.Datenbereich r1-Datenlaenge r2-Transferkennung	Ausloesen einer I/O-Anforderung mit Warten auf Requestendemeldung des U880
IOCTRL	19	Return:r0=-1 Request laeuft noch r0=0 Request beendet	Kontrolle Requestendemeldung des U880

Neben diesen allgemein nutzbaren Routinen gibt es noch folgende Funktionen, die nur im Segment 0 benutzt werden koennen und zur Vereinfachung der Konsolarbeit dienen.

Bezeichnung	SC-Nr.	Parameter	Kurzbeschreibung
GETLIN	2	Input: r1-max.Eingabelaenge r2-Adr.Eingabebereich Return: r1-eingelesene Laenge Z-Flag-max.Laenge erreicht	Eingabe einer Zeile von Console mit Zeichenreflektion bis <CR> oder max.Laenge
CRLF	3		Ausgabe <CR><LF> auf Console
PUTSTR	4	Input: r2-Adr. Zeichenkette	Ausgabe einer Zeichenkette zur Konsole Zeichenkette: word:=<laenge zeichenfolge> array[*byte]:='<zeichenfolge>'
BTOH16	5	Input: r5-Binaerwert	Konvertieren Binaerwort in hexadezimale ASCII-Zeichenkette, mit Ausgabe auf Ausgabepuffer des MON8000

STROCRCON	6		Ausgabe des Ausgabepuffers des MON8000 auf Console mit Anfuegen (CR); Loeschen Ausgabepuffer und Pufferzeiger
BTOH8	7	Input: r15-Binaerwert	Konvertieren Binaerwert in hexadezimale ASCII-Zeichen mit Ausgabe auf Ausgabepuffer des MON8000
SGKONV	8	Input: r15-Binaerwert	Konvertieren Binaerwert in ASCII-Segmentnummer (mit eckigen Klammern) mit Ausgabe auf Ausgabepuffer des MON8000
OUTBF1Z	9	Input: r11-ASCII-Zeichen	Eintragen eines Zeichens in den Ausgabepuffer des MON8000 mit Weiterstellen Pufferzeiger
CMDLOP	10		Ruecksprung zur zentralen Kommandoeingabeschleife des MON8000

Kv 460/86 - V 3/15 - 445