DH11

DEVICE ROUTINE (MPG)

MD-11-DTDHA-B

EP-DTDHA-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DTDHB-B
PRODUCT NAME: DH11 DEVICE ROUTINE FOR MP6
DATE: APRIL 1976
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: M. A. GREENE

%

```
                                .SBTTL  REVISION HISTORY

;       APR 76          DTDH9-B RELEASE
;       JAN 76          ADDED MEMORY MANAGEMENT SUPPORT
;       OCT 75          CREATED FULL SUPPORT DEVICE ROUTINE
;       AUG 75          DTDH9-A INITIAL RELEASE (MINIMUM SUPPORT
;                       DEVICE ROUTINE)
```

```
                                        .SBTTL  STANDARD DEVICE ROUTINE TABLE
                                        .TITLE  MAINDEC-11-DTDH0-B      DH11 DEVICE ROUTINE FOR MPG

                                        ;REVISION B

                                        ;FILENAME OF "TDH0B0.MPG" ON MPG/100P MEDIA

                                        ;MACY11:  DTDH0B,DTDH0B/CRF:SYN/DOC=DTDH0B.P11
                                        ;LINK11:        DTDH0B,MPG/B:0-DTDH0B/E
                                        ;PAPER TAPE:  PUNCH DTDH0B.MPG/FILE:ELEV

            000030'                     .CSECT  DH11
                                        .DSABL  GBL
            177776                      PS=     177776

                                        ;THE FOLLOWING TABLE IS IN THE STANDARDIZED FORMAT REQUIRED
                                        ;TO INTERFACE WITH MPG.

000000' 007522              LOCZ:       .WORD   DVREND-.                ;DEVICE ROUT SIZE IN BYTES
000002' 000000              FLAGWD:     .WORD   0                       ;DEVICE ROUT FLAGWORD

            100000                      DRWAIT= 100000                  ;  DEVICE ROUTINE WAIT FLAG
            040000                      WRTBSY= 40000                   ;  WRITE BUSY FLAG
            020000                      RDRBSY= 20000                   ;  READ BUSY FLAG
            000010                      BRKFLG= 10                      ;  BREAK INST FLAG
            000004                      CLWVCT= 4                       ;  CLEAR WRITE VECTOR FLAG
            000002                      CLRVC=  2                       ;  CLEAR READ VECTOR FLAG
            000001                      LDRFLG= 1                       ;  LEADER FLAG

000004' 000000                          .WORD   0                       ;INTERFACE WORD 0 (NOT USED)
000006' 000000                          .WORD   0                       ;INTERFACE WORD 1 (NOT USED)
000010' 000000                          .WORD   0                       ;INTERFACE WORD 2 (NOT USED)
000012' 000000                          .WORD   0                       ;INTERFACE WORD 3 (NOT USED)
000014' 000000                          .WORD   0                       ;INTERFACE WORD 4 (NOT USED)
000016' 000001                          .WORD   0                       ;INTERFACE WORD 5 (NOT USED)
000020' 000000                          .WORD   0                       ;INTERFACE WORD 6 (NOT USED)
000022' 000000              SIZE:       .WORD   0                       ;# OF BYTES TRANSFERRED / UNIMP FLG
000024' 000000              ERR:        .WORD   0                       ;ERROR ON LAST I/O INDICATOR
000026' 160020              DREG00:     .WORD   160020                  ;FIRST DEVICE REGISTER ADR
000030' 000300              IVCT00:     .WORD   300                     ;INTERRUPT VECTOR ADR
000032' 000240              RSLVL0:     .WORD   240                     ;INT PROC STATUS WORD (BR 5)
000034' 000240              XMTRL0:     .WORD   240                     ;INT PROC STATUS WORD (BR 5)
000036' 000000                          .WORD   HSKEEP-.                ;HOUSEKEEPING ROUT REL ADR
000040' 000000                          .WORD   REPORT-.                ;REPORT ROUT REL ADR
000042' 000000                          .WORD   KILL-.                  ;KILL ROUT REL ADR
000044' 000000                          .WORD   DATAER-.                ;DATA ERROR COUNTER REL ADR
000046' 000000                          .WORD   TOUTER-.                ;TIME OUT ERROR ROUT REL ADR
000050' 000000              CIOBSY:     .WORD   0                       ;I/O BUSY BRANCH ADR
000052' 000000              DUPGER:     .WORD   0                       ;DEVICE ERROR BRANCH ADR
000054' 000000              ULIST:      .WORD   0                       ;USER MODE PRINT ROUTINE BRANCH ADR
000056' 000000              CLIST:      .WORD   0                       ;CMND MODE PRINT ROUTINE BRANCH ADR
000060' 000000              BINASC:     .WORD   0                       ;CONVERT BINARY TO ASCII ROUT BR ADR
000062' 000000              BINASL2:    .WORD   0                       ;CONVERT BINARY TO DECIMAL ASCII BR ADR
000064' 000000              DECASC:     .WORD   0                       ;CONVERT PACKED DECIMAL TO ASCII BR ADR
```

```
                        ;MPG SYSTEM FLAGWORD ADR
                        ;SET INT VECT ROUT BR ADR
                        ;CLEAR INT VECTOR ROUT BR ADR
                        ;TEST INT VECTOR ROUT BR ADR
                        ;RETURN FROM INT ROUT BR ADR
                        ;GET DATA BYTE ROUT BR ADR
                        ;PUT DATA BYTE ROUT BR ADR
                        ;ADR OF DEVICE REGISTER NAMES
                        ;ADR OF DEVICE FUNCTIONS
                        ;ADR OF PACK TBL EXTENSION
                        ;ADR OF MODEL VECTOR TBL EXTEN.
                        ;ADR OF COMPILER TBL EXTEN.
                        ;ADR OF DEV INTERFACE AND SYM TBL

                        ;VALID DEVICE REGISTER NAMES &
                        ;THEIR POSITIONS RELATIVE TO
                        ;THE DEVICE REGISTERS BASE ADDRESS.

                        ;VALID DEVICE FUNCTIONS
                        ;  FLAG BYTE:
                        ;     BIT 7 = NPR DEV
                        ;     BIT 3 = MASSBUS DEV
                        ;     BIT 0 = 2 WORDS FOR ADR
                        ;             (18 BIT ADRS)
```

```
 000254' 002576                          .WORD   BAUD-.
 000256'    365    000                   .BYTE   365,0
 000260' 002606                          .WORD   RBAUD-.
 000262'    364    000                   .BYTE   364,0
 000264' 002612                          .WORD   TBAUD-.
 000266'    363    000                   .BYTE   363,0
 000270' 003052                          .WORD   EVEN-.
 000272'    362    000                   .BYTE   362,0
 000274' 003064                          .WORD   ODD-.
 000276'    361    000                   .BYTE   361,0
 000300' 003076                          .WORD   NOPAR-.
 000302'    360    000                   .BYTE   360,0
 000304' 003106                          .WORD   ONESTP-.
 000306'    357    000                   .BYTE   357,0
 000310' 003116                          .WORD   TWOSTP-.
 000312'    356    000                   .BYTE   356,0
 000314' 003130                          .WORD   BITS-.
 000316'    355    000                   .BYTE   355,0
 000320' 003250                          .WORD   ECHO-.
 000322'    354    000                   .BYTE   354,0
 000324' 003260                          .WORD   NOECHO-.
 000326'    353    000                   .BYTE   353,0
 000330' 003320                          .WORD   PRESET-.
 000332'    352    000                   .BYTE   352,0
 000334' 003332                          .WORD   ALARM-.
 000336' 177777                          .WORD   177777                  ;TABLE TERMINATOR

 000340' 051103 051505 052105  DVPKTE:   .ASCII  /CRESET/                ;PACK TABLE EXTENSION
 000346'    376    000                   .BYTE   376,0
 000350' 047516 040527 052111            .ASCII  /NOWAIT/
 000356'    375    000                   .BYTE   375,0
 000360' 020040 040527 052111            .ASCII  /  WAIT/
 000366'    374    000                   .BYTE   374,0
 000370' 052123 052101 051525            .ASCII  /STATUS/
 000376'    373    000                   .BYTE   373,0
 000400' 047503 047125 051524            .ASCII  /COUNTS/
 000406'    372    000                   .BYTE   372,0
 000410' 042106 050125 054114            .ASCII  /FDUPLX/
 000416'    371    000                   .BYTE   371,0
 000420' 042110 050125 054114            .ASCII  /HDUPLX/
 000426'    370    000                   .BYTE   370,0
 000430' 051440 052105 050125            .ASCII  / SETUP/
 000436'    367    000                   .BYTE   367,0
 000440' 020040 040502 042125            .ASCII  /  BAUD/
 000446'    366    000                   .BYTE   366,0
 000450' 051040 040502 042125            .ASCII  / RBAUD/
 000456'    365    000                   .BYTE   365,0
 000460' 052040 040502 042125            .ASCII  / TBAUD/
 000466'    364    000                   .BYTE   364,0
 000470' 020040 053105 047105            .ASCII  /  EVEN/
 000476'    363    000                   .BYTE   363,0
 000500' 020040 047440 042104            .ASCII  /   ODD/
 000506'    362    000                   .BYTE   362,0
 000510' 047040 050117 051101            .ASCII  / NOPAR/
 000516'    361    000                   .BYTE   361,0
 000520' 047117 051505 050124            .ASCII  /ONESTP/
```

G01


```
000526'    360     000                      .BYTE   360,0
000530'  053524  051517  050124             .ASCII  /TWOSTP/
000536'    357     000                      .BYTE   357,0
000540'  020040  044502  051524             .ASCII  /  BITS/
000546'    356     000                      .BYTE   356,0
000550'  020040  041505  047510             .ASCII  /  ECHO/
000556'    355     000                      .BYTE   355,0
000560'  047516  041505  047510             .ASCII  /NOECHO/
000566'    354     000                      .BYTE   354,0
000570'  051120  051505  052105             .ASCII  /PRESET/
000576'    353     000                      .BYTE   353,0
000600'  040440  040514  046522             .ASCII  / ALARM/
000606'    352     000                      .BYTE   352,0

000610'  000376  001154            DVMVTE:  .WORD   376,LCRST-LOCZ          ;MODEL VECTOR TABLE EXTEN.
000614'  000375  001154                     .WORD   375,LNWAIT-LOCZ
000620'  000374  001154                     .WORD   374,LWAIT-LOCZ
000624'  000373  001154                     .WORD   373,LCOUNT-LOCZ
000630'  000372  001154                     .WORD   372,LCOUNT-LOCZ
000634'  000371  001154                     .WORD   371,LFDPLX-LOCZ
000640'  000370  001154                     .WORD   370,LHDPLX-LOCZ
000644'  000367  001157                     .WORD   367,LSETUP-LOCZ
000650'  000366  001155                     .WORD   366,LRBAUD-LOCZ
000654'  000365  001155                     .WORD   365,LRBAUD-LOCZ
000660'  000364  001155                     .WORD   364,LTBAUD-LOCZ
000664'  000363  001154                     .WORD   363,LEVEN-LOCZ
000670'  000362  001154                     .WORD   362,LODD-LOCZ
000674'  000361  001154                     .WORD   361,LNOPAR-LOCZ
000700'  000360  001154                     .WORD   360,LSTP1-LOCZ
000704'  000357  001154                     .WORD   357,LSTP2-LOCZ
000710'  000356  001155                     .WORD   356,LBITS-LOCZ
000714'  000355  001154                     .WORD   355,LECHO-LOCZ
000720'  000354  001154                     .WORD   354,LNECHO-LOCZ
000724'  000353  001154                     .WORD   353,LPREST-LOCZ
000730'  000352  001155                     .WORD   352,LALARM-LOCZ

                                   ;
                                   ;        COMPILER TABLE EXTENSION
                                   ;
000734'    003     376             DVCPTE:  .BYTE   3,376                   ;CONTROL RESET
000736'  004537  000012                     .WORD   4537,10.
000742'    003     375                      .BYTE   3,375                   ;NO WAIT
000744'  004537  000012                     .WORD   4537,10.
000750'    003     374                      .BYTE   3,374                   ;WAIT
000752'  004537  000012                     .WORD   4537,10.
000756'    004     373                      .BYTE   4,373                   ;STATUS
000760'  004537  000012  001002             .WORD   4537,10.,1002
000766'    004     372                      .BYTE   4,372                   ;COUNTS
000770'  004537  000012  001001             .WORD   4537,10.,1001
000776'    003     371                      .BYTE   3,371                   ;FULL DUPLEX
001000'  004537  000012                     .WORD   4537,10.
001004'    003     370                      .BYTE   3,370                   ;HALF DUPLEX
001006'  004537  000012                     .WORD   4537,10.
001012'    004     367                      .BYTE   4,367                   ;SETUP
001014'  004537  000012  000004             .WORD   4537,10.,4
001022'    004     366                      .BYTE   4,366                   ;TX & RX BAUD RATE
```

```
	001024'	004537	000012	000000		.WORD	4537,10.,0
	001032'	004	365			.BYTE	4,365		;RECEIVE BAUD RATE
	001034'	004537	000012	000000		.WORD	4537,10.,0
	001042'	004	364			.BYTE	4,364		;TRANSMIT BAUD RATE
	001044'	004537	000012	000000		.WORD	4537,10.,0
	001052'	003	363			.BYTE	3,363		;EVEN PARITY
	001054'	004537	000012			.WORD	4537,10.
	001060'	003	362			.BYTE	3,362		;ODD PARITY
	001062'	004537	000012			.WORD	4537,10.
	001066'	003	361			.BYTE	3,361		;NO PARITY
	001070'	004537	000012			.WORD	4537,10.
	001074'	003	360			.BYTE	3,360		;ONE STOP BIT
	001076'	004537	000012			.WORD	4537,10.
	001102'	003	357			.BYTE	3,357		;TWO STOP BITS
	001104'	004537	000012			.WORD	4537,10.
	[illegible]	[illegible]	356			.BYTE	[illegible],356	;BITS PER CHAR.
	001112'	004537	000012	000000		.WORD	4537,10.,0
	001120'	003	355			.BYTE	3,355		;ECHO
	[illegible]	004537	000012			.WORD	4537,10.
	[illegible]	003	354			.BYTE	3,354		;NO ECHO
	001130'	004537	000012			.WORD	4537,10.
	001134'	003	353			.BYTE	3,353		;PRESET
	[illegible]	004537	000012			.WORD	4537,10.
	[illegible]	004	352			.BYTE	4,352		;ALARM LEVEL
	001144'	[illegible]537	000012	000000		.WORD	4537,10.,0

				;
				;		DEVICE INTERFACE WORD SYMBOL TABLE
				;
	001152'	177777				DVIWST:	.WORD	177777
				;
				;		MODEL STATEMENT TABLE EXTENSION
				;
	001154'					LEVEN:
	001154'					LODD:
	001154'					LNOPAR:
	001154'					LSTP1:
	001154'					LSTP2:
	001154'					LECHO:
	001154'					LNECHO:
	001154'					LPREST:
	001154'					LCRST:
	001154'					LNWAIT:
	001154'					LWAIT:
	001154'					LSTATS:
	001154'					LCOUNT:
	001154'					LFDPLX:
	001154'		000			LHDPLX:	.BYTE	0
	001155'					LBAUD:
	001155'					LRBAUD:
	001155'					LTBAUD:
	001155'					LBITS:
	001155'		377	000		LALARM:	.BYTE	377,0
	001157'		375	000		LSETUP:	.BYTE	375,0
		001162'					.EVEN
```

```
                                ;STORAGE FOR DEV REG'S AT INT
001162'                 HSKPST= .
001162'                 ISTAT=  .
001162' 000000          SCR:    .WORD   0
001164' 000000          MAR:    .WORD   0
001166' 000000          LWR:    .WORD   0
001170' 000000          CWR:    .WORD   0
001172' 000000          BYCR:   .WORD   0
001174' 000000          BAR:    .WORD   0
001176' 000000          BCR:    .WORD   0
001200' 000000          SSR:    .WORD   0

001202' 000010          CSTAT:  .BLKW   8.              ;DEV REG CURRENT VALUE STORAGE

001222' 000000          BYRD:   .WORD   0               ;BYTES READ COUNT
001224' 000000                  .WORD
001226' 000000          BYWR:   .WORD   0               ;BYTES WRITTEN COUNT
001230' 000000                  .WORD
001232' 000000          RDCNT:  .WORD   0               ;READ CMND COUNT
001234' 000000          WRCNT:  .WORD   0               ;WRITE CMND COUNT
001236' 000000          BKCNT:  .WORD   0               ;BREAK CMND COUNT
001240' 000000          MISCNT: .WORD   0               ;MISC. CMND COUNT (CRESET,
001242' 000000          RICNT:  .WORD   0               ;READ INTERRUPT COUNT
001244' 000000          WICNT:  .WORD   0               ;WRITE INTERRUPT COUNT
001246' 000000          NEMERR: .WORD   0               ;NON-EXIST MEM ERR CNT
                                                        ;FDUPLX,HDUPLX)
001250' 000000          OVRRUN: .WORD   0               ;OVERRUN ERRORS COUNT
001252' 000000          FRAMER: .WORD   0               ;FRAMING ERRORS COUNT
001254' 000000          PARERR: .WORD   0               ;PARITY ERRORS COUNT
001256' 000000          SOFERR: .WORD   0               ;SILO OVERFLOW COUNT
001260' 000000          DATAER: .WORD   0               ;DATA ERROR COUNT
001262' 000000          TOECNT: .WORD   0               ;# OF ENTRIES INTO T/O ERROR ROUT

001264' 000000          FLAG:   .WORD   0               ;FLAGWORD STORAGE


                        HSKPEN= .

                        XXXX=   0                       ;VALUE TO BE TAILORED BY DEV ROUT

                        000001  MMVER=  1               ;SYSTEM FLGWD BIT DEF.
                        120000  PSCONS= 120000          ;INT SRVC VIRT PAGE BASE
                        000002  USMTPS= 2               ;MTPS INST LEGAL FLAG
```

```
                                        .SBTTL  DH11 FUNCTION ROUTINES
                                        ;TIMEOUT ERROR HANDLER
001256' 005267 177770          TOUTER:  INC     TOECNT                  ;INCR TIME OUT ERROR COUNT
001262' 026727 177764 000010           CMP     TOECNT,#8               ;EXCEEDED 8 TIMEOUTS?
001270' 001401                         BEQ     2$                      ;YES - CONTINUE
001272' 000205                         RTS     R5                      ;NO - RETURN
001274' 004067 004646          2$:     JSR     R0,SAVREG               ;SAVE REGISTERS
001300' 004767 004674                  JSR     PC,SUPTAD               ;P TBL ADR TO R3
001304' 042713 000010                  BIC     #INT4IOT,(R3)           ;CLEAR WAIT FOR I/O TERM
001310' 005004                         CLR     R4
001312' 004567 005246                  JSR     R5,PRINT                ;PRINT TIMEOUT ERR MSG
001316' 000026                         .WORD   TOEMSG-.
001320' 000023                         .WORD   19.
001322' 004567 000042                  JSR     R5,KILL                 ;KILL THE PROGRAM
001326' 004767 004360                  JSR     PC,ERDIRG               ;DISPLAY STATUS & STMNT #
001332' 004067 004424                  JSR     R0,RESREG               ;RESTORE REGISTERS
001336' 012605                 TOUTEX:  MOV     (SP)+,R5                ;GO DISPLAY DEVICE REGS
001340' 000177 176474                  JMP     @#USPGER

001344' 044104 030461 052040   TOEMSG: .ASCII  'DH11 TIMEOUT ON I/O'
001352' 046511 047505 051440
001360' 047440 020116 027511
001366' 117
        001400'                        .EVEN
                                        ;KILL USER PROGRAM ROUTINE
001400' 016702 176420          KILL:    MOV     DREGAD,R2               ;GET DEV REG ADR
001404' 032712 020100                  BIT     #20100,(R2)             ;ANY INT EBLS SET ?
001410' 001424                         BEQ     KILLEX                  ;NO-EXIT
001412' 004567 005272          1$:     JSR     R5,TRVECT               ;TEST READ INT VECTOR
001416' 000407                         BR      CWRINT                  ;BRANCH IF NOT ME
001420' 042712 010100                  BIC     #10100,(R2)             ;RESET RO INT EBL
001424' 042767 020000 176350           BIC     #RQBSY,FLAG#0           ;RESET READ BSY IN FLAG #0
001432' 004767 002640                  JSR     PC,RRINTV               ;RESET INT VECTOR INFO
001436' 004567 005276          CWRINT:  JSR     R5,TWVECT               ;TEST WRITE INT VECTOR
001442' 000407                         BR      KILLEX                  ;BRANCH IF NOT ME
001444' 042712 020000                  BIC     #20000,(R2)             ;RESET WR INT EBL
001450' 042767 040000 176324           BIC     #WRBSY,FLAG#0           ;RESET WRITE BSY IN FLAG #0
001456' 004767 002652                  JSR     PC,RWINTV               ;RESET INT VECTOR INFO
001462' 005067 176334          KILLEX:  CLR     ERR                     ;CLEAR ERROR INDICATOR
001466' 005067 002554                  CLR     RCR                     ;CLEAR INTERN CTL REG
001472' 005067 002552                  CLR     ERRFLG                  ;CLEAR INTERN ERR FLAG
001476' 000205                         RTS     R5                      ;RETURN
                                        ;READ COMMAND HANDLER
001500' 010567 004346          READ:    MOV     R5,STMNT                ;SAVE R5
001504' 162767 000004 004340           SUB     #4,STMNT                ;FOR STMNT # REFERENCE
001512' 016704 176306                  MOV     DREGAD,R4               ;DEV REG ADDR TO R4
001516' 032767 020000 176256           BIT     #RQBSY,FLAG#0           ;TEST READ BUSY
001524' 001030                         BNE     6$IPTBS                 ;BRANCH IF SET
001526' 032714 000100                  BIT     #100,(R4)               ;TEST RECV INT EBL
001532' 001403                         BEQ     5$RQBSY                 ;CONTINUE IF NOT SET
```

```
001534' 004577 176306                     JSR     R5,@CIOBSY              ;OTHERWISE RELEASE CONTROL
001540' 000757                            BR      READ
001542' 052767 020000 176232    SRDBSY:   BIS     #RDBSY,FLAGWD           ;SET READ BUSY
001550' 016767 176252 000012              MOV     IVCTRD,1$               ;INT VECTOR ADDR TO CALL
001556' 016767 176246 000006              MOV     RBUSRQ,2$               ;ALSO BUS PRIORITY
001564' 004577 176276                     JSR     R5,@SETVEC              ;GO SET THE VECTOR
001570' 000000                  1$:       .WORD   XXXX
001572' 000000                  2$:       .WORD   XXXX
001574' 000036                            .WORD   RDINT-.
001576' 005067 177460                     CLR     TOECNT                  ;CLEAR TIME OUT ERR CNT
001602' 052714 010100                     BIS     #10100,(R4)             ;SET RECV INT EBL
001606' 004767 004376           GTPTBS:   JSR     PC,SUPTAD               ;GET P TBL BASE IN R3
001612' 005067 176204                     CLR     ERR                     ;CLEAR ERROR INDICATOR
001616' 005063 000030                     CLR     PTOCNT(R3)
001622' 032763 000200 000002              BIT     #SPOPER,POPSW(R3)       ;TEST MAINT BIT IN OPSW
001630' 001403                            BEQ     RDMNCL
001632' 052714 001000                     BIS     #1000,(R4)              ;SET CSR BIT TO SAME STATE
001636' 000402                            BR      RDMNST
001640' 042714 001000           RDMNCL:   BIC     #1000,(R4)
001644' 012567 002364           RDMNST:   MOV     (R5)+,MSADDR            ;GET ADDRESS
001650' 012567 002362                     MOV     (R5)+,LSADDR
001654' 012567 002360                     MOV     (R5)+,BYTES             ;BYTE COUNT AND
001660' 012500                            MOV     (R5)+,R0                ;LINE NBR FROM CALL
001662' 001011                            BNE     TESTRC                  ;BR IF LINE SPECIFIED
001664' 116301 000035                     MOVB    PCURDV(R3),R1           ;GET CURRENT DEV NBR
001670' 012700 000001                     MOV     #1,R0
001674' 105701                            TSTB    R1
001676' 001403                            BEQ     TESTRC
001700' 006300                  CALDNL:   ASL     R0
001702' 005301                            DEC     R1
001704' 001375                            BNE     CALDNL
001706' 030067 002334           TESTRC:   BIT     R0,RCR                  ;ANY DESIRED LINE IN USE ?
001712' 001405                            BEQ     LOADRC                  ;NO - KICK OFF THIS READ
001714' 004577 176126                     JSR     R5,@CIOBSY              ;YES - WAIT
001720' 162705 000006                     SUB     #6,R5
001724' 000665                            BR      READ
001726' 050067 002314           LOADRC:   BIS     R0,RCR                  ;SET LINE NBRS IN RCR
001732' 005267 177274                     INC     RDCNT                   ;INCR READ CMD COUNT
001736' 010702                            MOV     PC,R2
001740' 062702 002312                     ADD     #RCVTBL-.,R2
001744' 012701 000020                     MOV     #16.,R1
001750' 006000                  FRCVTL:   ROR     R0                      ;FILL RECEIVE TABLE
001752' 103007                            BCC     DRCVTL
001754' 016722 002256                     MOV     LSADDR,(R2)+            ;WITH ADDRESS
001760' 016722 002254                     MOV     BYTES,(R2)+             ;AND BYTE COUNT
001764' 005301                            DEC     R1
001766' 001370                            BNE     FRCVTL
001770' 000404                            BR      SETRSE
001772' 062702 000004           DRCVTL:   ADD     #4,R2
001776' 005301                            DEC     R1
002000' 001363                            BNE     FRCVTL
002002' 052763 000010 000000    SETRSE:   BIS     #WT4IOT,PFLGWD(R3)      ;SET WAIT FOR I/O TERM
002010' 032767 100000 175764              BIT     #DRWAIT,FLAGWD          ;TEST DEV ROUT NOWAIT
002016' 001002                            BNE     RDNOWT                  ;BRANCH IF SET
002020' 000167 000610                     JMP     TSTIEB                  ;OTHERWISE WAIT
002024' 042763 000010 000000    RDNOWT:   BIC     #WT4IOT,PFLGWD(R3)      ;CLEAR WAIT FOR I/O TERM
```

```
002032' 000205                          RTS     R5                      ;RETURN INLINE

                                        ;WRITE AND BREAK COMMAND HANDLER

002034' 042767 000010 175740    WRITE:  BIC     #BRFLG,FLAGWD           ;CLEAR BREAK FLAG
002042' 000403                          BR      XMIT
002044' 052767 000010 175730    BREAK:  BIS     #BRFLG,FLAGWD           ;SET BREAK FLAG
002052' 010567 003774           XMIT:   MOV     R5,STMNT                ;SAVE R5
002056' 162767 000004 003766            SUB     #4,STMNT                ;FOR STMNT # REFERENCE
002064' 016704 175734                   MOV     DREGAD,R4               ;DEV REG ADDR TO R4
002070' 032767 040000 175704            BIT     #WRBSY,FLAGWD           ;TEST WRITE BUSY
002076' 001033                          BNE     CLRERR                  ;BRANCH IF SET
002100' 032714 020000                   BIT     #20000,(R4)             ;TEST XMIT INT EBL
002104' 001403                          BEQ     SETBSY                  ;CONTINUE IF NOT SET
002106' 004577 175734                   JSR     R5,@CIOBSY              ;OTHERWISE RELEASE CONTROL
002112' 000757                          BR      XMIT
002114' 052767 040000 175660    SETBSY: BIS     #WRBSY,FLAGWD           ;SET WRITE BUSY
002122' 016767 175700 000020            MOV     VCTRAD,10$              ;INT VECTOR ADDR TO CALL
002130' 062767 000004 000012            ADD     #4,10$                  ;ADJUST FOR WRITE INT
002136' 016767 175670 000006            MOV     BRSTAT,20$              ;ALSO PASS BUS PRIORITY
002144' 004577 175716                   JSR     R5,@SETVEC              ;GO SET THE VECTOR
002150' 000000                  10$:    .WORD   XXXX
002152' 000000                  20$:    .WORD   XXXX
002154' 002176                          .WORD   WRITE-
002156' 005067 177100                   CLR     TOECNT                  ;CLEAR TIME OUT ERR CNT
002162' 052714 020000                   BIS     #20000,(R4)             ;SET XMIT INT EBL
002166' 005067 175630           CLRERR: CLR     ERR                     ;CLEAR ERROR INDICATOR
002172' 005063 000030                   CLR     PTOCNT(R3)
002176' 004767 004006                   JSR     PC,SUPTAD               ;GET P TBL BASE IN R3
002202' 032763 000200 000002            BIT     #SPOPER,POPSW(R3)       ;TEST MAINT BIT IN OPSW
002210' 001403                          BEQ     WRMNCL
002212' 052714 001000                   BIS     #1000,(R4)              ;SET CSR BIT TO SAME STATE
002216' 000402                          BR      WRMNST
002220' 042714 001000           WRMNCL: BIC     #1000,(R4)
002224' 012567 002004           WRMNST: MOV     (R5)+,MSADDR            ;GET MS ADDRESS
002230' 012567 002002                   MOV     (R5)+,LSADDR
002234' 012567 002000                   MOV     (R5)+,BYTES             ;BYTE COUNT AND
002240' 012500                          MOV     (R5)+,R0                ;LINE NBR FROM CALL
002242' 001011                          BNE     TESTTC                  ;BR IF LINE SPECIFIED
002244' 116301 000035                   MOVB    PCURDV(R3),R1           ;GET CURRENT DEV NBR
002250' 012700 000001                   MOV     #1,R0
002254' 105701                          TSTB    R1
002256' 001403                          BEQ     TESTTC
002260' 006300                  DVNMLP: ASL     R0
002262' 005301                          DEC     R1
002264' 001375                          BNE     DVNMLP
002266' 030064 000012           TESTTC: BIT     R0,12(R4)               ;ANY DESIRED LINE IN USE ?
002272' 001405                          BEQ     LOADTC                  ;NO - KICK OFF THIS WRITE
002274' 004577 175546                   JSR     R5,@CIOBSY              ;YES - WAIT
002300' 162705 000006                   SUB     #6,R5
002304' 000643                          BR      XMIT
002306' 010067 001730           LOADTC: MOV     R0,TEMTCR               ;SET LINE NBRS IN TEMP TCR
002312' 032777 000002 175544            BIT     #U$MTPS,@CSYSFW         ;SAVE PS AND INH INT
002320' 001007                          BNE     10$
002322' 113767 177776 001702            MOVB    @#PS,PSSAVE
002330' 152737 000340 177776            BISB    #340,@#PS
```

```
002336' 000404                  BR      20$
002340' 106767 001666    10$:   MFPS    PSSAVE
002344' 106427 000340           MTPS    #340
002350' 016701 001660    20$:   MOV     MSADDR,R1               ;GET MEMORY EXTEN BITS
002354' 000241                  CLC
002356' 006101                  ROL     R1
002360' 006101                  ROL     R1
002362' 006101                  ROL     R1
002364' 006101                  ROL     R1
002366' 062714 000060           BIC     #60,(R4)
002372' 050114                  BIS     R1,(R4)                 ;JAM INTO SCR
002374' 010702           FSNOTB: MOV     PC,R2
002376' 062702 001474           ADD     #SNOTBL-.,R2
002402' 005001                  CLR     R1
002404' 006000           FSNOTL: ROR     R0                      ;TEST LINE BIT
002406' 100030                  BCC     DSNOTL                  ;BYPASS IF ZERO
002410' 042714 000017           BIC     #17,(R4)                ;CLEAR LINE BITS
002414' 050114                  BIS     R1,(R4)                 ;SELECT LINE
002416' 000240                  NOP
002420' 000240                  NOP                             ;TO MAKE LISTING MATCH EVAL VER
002422' 016764 001610 000006    MOV     LSADDR,6(R4)            ;STORE HARD ADDR
002430' 016764 001604 000010    MOV     BYTES,10(R4)            ;STORE HARD B.C.
002436' 005464 000010           NEG     10(R4)                  ;PUT IN TWOS COMP FORM
002442' 016722 001566           MOV     MSADDR,(R2)+            ;STORE SOFT ADDR
002446' 016722 001564           MOV     LSADDR,(R2)+
002452' 016722 001562           MOV     BYTES,(R2)+             ;STORE SOFT B.C.
002456' 005201                  INC     R1
002460' 022701 000020           CMP     #16.,R1
002464' 001347                  BNE     FSNOTL                  ;REPEAT FOR 16 LINES
002466' 000406                  BR      SETTCR
002470' 062702 000004    DSNOTL: ADD     #4,R2                   ;BYPASS THIS LINE
002474' 005201                  INC     R1
002476' 022701 000020           CMP     #16.,R1
002502' 001340                  BNE     FSNOTL                  ;REPEAT IF NOT LAST
002504' 052763 000010 000000 SETTCR: BIS  #MT4IOT,PFLGWD(R3)     ;SET WAIT FOR I/O TERM
002512' 056764 001524 000012    BIS     TEMTCR,12(R4)           ;SET LINE NBRS IN HARD TCR
002520' 056767 001516 001516    BIS     TEMTCR,STCR             ;AND SOFTWARE TCR
002526' 032767 000010 175246    BIT     #BRFLG,FLAGWD           ;IF BREAK FLAG SET
002534' 001406                  BEQ     INCWRC
002536' 056764 001500 000014    BIS     TEMTCR,14(R4)           ;SET LINE NBRS IN BCR
002544' 005267 176446           INC     BKCNT                   ;INCR BREAK CMD COUNT
002550' 000402                  BR      SETSCN
002552' 005267 176456    INCWRC: INC     WRCNT                   ;INCR WRITE CMD COUNT
002556' 032777 000002 175300 SETSCN: BIT  #USHTPS,@CSYSFW        ;RESTORE PS AND EBL INT
002564' 001004                  BNE     10$
002566' 116737 001440 177776    MOVB    PSSAVE,@PS
002574' 000402                  BR      20$
002576' 106467 001430    10$:   MTPS    PSSAVE
002602' 032767 100000 175172 20$:  BIT   #DRWAIT,FLAGWD         ;TEST DEV ROUT NOWAIT
002610' 001411                  BEQ     TSTIEB                  ;IF RESET TEST INT EBL
002612' 042763 000010 000000    BIC     #MT4IOT,PFLGWD(R3)      ;CLEAR WAIT FOR I/O TERM
002620' 000205           RETURN: RTS     R5                      ;OTHERWISE RETURN

                        ;WAIT COMMAND HANDLER

002622' 042767 100000 175152 WAIT:  BIC     #DRWAIT,FLAGWD      ;CLEAR DEV ROUT NOWAIT
```

```
002630' 016704  175170                  MOV     DREGAD,R4               ;POINT R4 AT REG ADDR
002634' 032714  020100          TSTIEB: BIT     #20100,(R4)             ;TEST INT EBLS
002640' 001053                          BNE     RELEAS                  ;IF SET, RELEASE CONTROL
002642' 032767  000002  175132  TRMTST: BIT     #CLRVCT,FLAGWD          ;TEST IF VECTOR CLR REQD
002650' 001403                          BEQ     10$                     ;BRANCH IF NOT
002652' 004567  004032                  JSR     R5,TRVECT               ;TEST READ VECTOR
002656' 000405                          BR      10$                     ;BRANCH IF NOT ME
002660' 004767  002232                  JSR     PC,RRINTV               ;GO RESET THE VECTOR
002664' 042767  000002  175110          BIC     #CLRVCT,FLAGWD          ;CLEAR THE REQ FLAG
002672' 032767  000004  175102  10$:    BIT     #CLWVCT,FLAGWD          ;TEST IF VECTOR CLR REQD
002700' 001410                          BEQ     ERRTST                  ;BRANCH IF NOT
002702' 004567  004032                  JSR     R5,TWVECT               ;TEST WRITE VECTOR
002706' 000405                          BR      ERRTST                  ;BRANCH IF NOT ME
002710' 004767  002220                  JSR     PC,RWINTV               ;GO RESET THE VECTOR
002714' 042767  000004  175060          BIC     #CLWVCT,FLAGWD          ;CLEAR THE REQ FLAG
002722' 005767  001322          ERRTST: TST     ERRFLG                  ;TEST FOR ANY ERROR
002726' 001734                          BEQ     RETURN                  ;RETURN IF NONE
002730' 012767  000001  175064          MOV     #1,ERR                  ;SET ERROR INDICATOR
002736' 004767  003246                  JSR     PC,SUPTAD
002742' 032763  020000  000000          BIT     #PRONER,POPSW(R3)       ;TEST DONT PRINT ON ERR BIT
002750' 001005                          BNE     ERREXT                  ;EXIT IF SET
002752' 005267  002600                  INC     ABBREV                  ;SET ABBREVIATED RPT FLG
002756' 005004                          CLR     R4
002760' 000167  002634                  JMP     ERRRPT
002764' 000177  175060          ERREXT: JMP     @CUPGER
002770' 004577  175052          RELEAS: JSR     R5,@CIOBSY              ;OTHERWISE RELEASE CONTROL
002774' 000717                          BR      TSTIEB

                                ;NOWAIT COMMAND HANDLER

002776' 052767  100000  174776  NOWAIT: BIS     #DRWAIT,FLAGWD          ;SET NOWAIT FLAG
003004' 000705                          BR      RETURN


                                ;CRESET COMMAND HANDLER

003006' 016704  175012          CRESET: MOV     DREGAD,R4
003012' 052714  004000                  BIS     #4000,(R4)              ;SET MASTER CLEAR
003016' 005267  176216                  INC     MISCNT                  ;INCR MISC CMD COUNT
003022' 005014                          CLR     (R4)                    ;CLEAR ALL CONTROL BITS
003024' 005067  174772                  CLR     ERR                     ;CLEAR ERROR INDICATOR
003030' 005067  001214                  CLR     ERRFLG                  ;CLEAR INTERN ERR FLAG
003034' 005067  001206                  CLR     RCR                     ;CLEAR INTERN CTL REG
003040' 000205                          RTS     R5                      ;RETURN

                                ;SETUP COMMAND HANDLER

003042' 012567  000002          SETUP:  MOV     (R5)+,LINE              ;SAVE LINES TO BE USED
003046' 000205                          RTS     R5                      ;RETURN
003050' 000000                  LINE:   .WORD   0

                                ;BAUD COMMAND HANDLER

003052' 004767  000030          BAUD:   JSR     PC,RBAUSB               ;CALL RECV BAUD AND
003056' 004767  000156                  JSR     PC,TBAUSB               ;XMIT BAUD SUBR
003062' 005725                          TST     (R5)+                   ;ADJUST R5
```

```
003064' 000205                          RTS     R5                      ;RETURN
003066' 004767 000014           RBAUD:  JSR     PC,RBAUSB               ;CALL RECV BAUD SUBR
003072' 005725                          TST     (R5)+                   ;ADJUST R5
003074' 000205                          RTS     R5                      ;RETURN
003076' 004767 000136           TBAUD:  JSR     PC,TBAUSB               ;CALL XMIT BAUD SUBR
003102' 005725                          TST     (R5)+                   ;ADJUST R5
003104' 000205                          RTS     R5                      ;RETURN

                                        ;RECEIVE BAUD COMMAND SUBROUTINE

003106' 010567 002740           RBAUSB: MOV     R5,STMNT                ;SAVE R5
003112' 162767 000004 002732            SUB     #4,STMNT                ;FOR STMNT & REFERENCE
003120' 004767 003064                   JSR     PC,SUPTAD
003124' 011500                          MOV     (R5),R0                 ;GET BAUD PARAM
003126' 020027 000017                   CMP     R0,#17                  ;CHECK IF WITHIN RANGE
003132' 101413                          BLOS    10$                     ;BRANCH IF OK
003134' 005004                          CLR     R4
003136' 004567 003432                   JSR     R5,PRINT                ;ELSE PRINT ERROR MESSAGE
003142' [illegible]                     .WORD   BAUMSG-.
003144' [illegible]                     .WORD   24.
003146' 005725                          TST     (R5)+
003150' 005726                          TST     (SP)+
003152' 004767 002552                   JSR     PC,ERRSNM               ;AND TAKE ERROR EXIT
003156' 000177 17[illegible]            JMP     @CLPGER
003162' 012701 000006           10$:    MOV     #6,R1                   ;MOVE RECEIVE
003166' 000241                          CLC
003170' 006100                  20$:    ROL     R0                      ;BAUD CODE
003172' 005301                          DEC     R1                      ;INTO POSITION
003174' 001375                          BNE     20$                     ;SET UP
003176' 012701 001700                   MOV     #1700,R1                ;PARAMETER MASK
003202' 004567 003070                   JSR     R5,LPAUPD               ;GO UPDATE PARAM REGS
003206' 000207                          RTS     PC                      ;RETURN
003210' [illegible] 042125 051440 BAUMSG: .ASCII /BAUD SELECTION CODE > 15/
003216' 046105 041505 [illegible]
003224' 047117 041440 042117
003232' 020105 020076 032461
                                        .EVEN

                                        ;TRANSMIT BAUD COMMAND SUBROUTINE

003240' 010567 002606           TBAUSB: MOV     R5,STMNT                ;SAVE R5
003244' 162767 000004 002600            SUB     #4,STMNT                ;FOR STMNT & REFERENCE
003252' 004767 002732                   JSR     PC,SUPTAD
003256' 011500                          MOV     (R5),R0                 ;GET BAUD PARAM
003260' 020027 000017                   CMP     R0,#17                  ;CHECK IF WITHIN RANGE
003264' 101413                          BLOS    10$                     ;BRANCH IF OK
003266' 005004                          CLR     R4
003270' 004567 003300                   JSR     R5,PRINT                ;ELSE PRINT ERR MSG
003274' 177714                          .WORD   BAUMSG-.
003276' 000030                          .WORD   24.
003300' 005725                          TST     (R5)+
003302' 005727                          TST     (PC)+
003304' 004767 002420                   JSR     PC,ERRSNM               ;AND TAKE ERROR EXIT
003310' 000177 174534                   JMP     @CLPGER
003314' 012701 000012           10$:    MOV     #10.,R1                 ;MOVE TRANSMIT
003320' 000241                          CLC
```

```
                                ;BAUD CODE
                                ;INTO POSITION
                                ;SET UP
                                ;PARAMETER MASK
                                ;GO UPDATE PROGRAM REGS
                                ;RETURN

                        ;EVEN COMMAND HANDLER

                                ;EVEN COMMAND HANDLER

                        ;ODD COMMAND HANDLER

                                ;ODD COMMAND HANDLER

                        ;NO PARITY COMMAND HANDLER

                                ;NO PARITY COMMAND HANDLER

                        ;ONE STOP BIT COMMAND HANDLER

                                ;ONE STOP BIT COMMAND HANDLER

                        ;TWO STOP BITS COMMAND HANDLER

                                ;TWO STOP BITS COMMAND HANDLER

                        ;BITS PER CHARACTER COMMAND HANDLER

                                ;SAVE R5
                                ;FOR STRNT 0 REFERENCE
                                ;GET NBR OF BITS
                                ;SUBTRACT 5
                                ;IF LESS THAN 4
                                ;ENTRY OK
                                ;ELSE PRINT ERR MSG
                                ;THEN TAKE ERR EXIT
```

```
    003514' 000177  174330           JMP     @CUPGER
    003520' 012701  000003   10$:    MOV     #3,R1                   ;GO UPDATE
    003524' 004567  002546           JSR     R5,LPARUPD              ;PARAMETER REGS
    003530' 000205                   RTS     R5                      ;RETURN
    003532' 020043  043117  041440  BITMSG: .ASCII  /# OF CHAR BITS NOT 5,6,7 OR 8/
    003540' 040510  020122  044502
    003546' 051524  047040  052117
    003554' 032440  033054  033454
    003562' 047440  020122     070
            003570'                          .EVEN

                                             ;ECHO COMMAND HANDLER

    003570' 012700  100000           ECHO:   MOV     #100000,R0
    003574' 010001                           MOV     R0,R1
    003576' 004567  002474                   JSR     R5,LPARUPD
    003602' 000205                           RTS     R5

                                             ;NO ECHO COMMAND HANDLER

    003604' 005000                   NOECHO: CLR     R0
    003606' 012701  100000                   MOV     #100000,R1
    003612' 004567  002460                   JSR     R5,LPARUPD
    003616' 000205                           RTS     R5

                                             ;HALF DUPLEX COMMAND HANDLER

    003620' 012700  040000           HDUPLX: MOV     #40000,R0
    003624' 010001                           MOV     R0,R1
    003626' 004567  002444                   JSR     R5,LPARUPD
    003632' 000205                           RTS     R5

                                             ;FULL DUPLEX COMMAND HANDLER

    003634' 005000                   FDUPLX: CLR     R0
    003636' 012701  040000                   MOV     #40000,R1
    003642' 004567  002430                   JSR     R5,LPARUPD
    003646' 000205                           RTS     R5

                                             ;PRESET COMMAND HANDLER

    003650' 012700  006307           PRESET: MOV     #6307,R0
    003654' 012701  177777                   MOV     #177777,R1
    003660' 004567  002412                   JSR     R5,LPARUPD
    003664' 000205                           RTS     R5

                                             ;ALARM COMMAND HANDLER

    003666' 010567  002160           ALARM:  MOV     R5,STANT                ;SAVE R5
    003672' 162767  000004  002152           SUB     #4,STANT                ;FOR STANT @ REFERENCE
    003700' 004767  002304                   JSR     PC,SUPTAD
    003704' 011500                           MOV     (R5),R0                 ;GET ALARM LEVEL
    003706' 012701  000006                   MOV     #6,R1                   ;SET UP BIT COUNT
    003712' 005700                           TST     R0                      ;ALARM LEVEL = 0?
    003714' 001426                           BEQ     40$                     ;YES - THATS ALLOWED
    003716' 020027  000040                   CMP     R0,#32.                 ;ALARM LEVEL > 32?
```

```
    003722' 101010                          BMI     25$                     ;THATS A NO - NO
    003724' 006000                  10$:    ROR     R0                      ;TEST FOR A BIT = 1
    003726' 103402                          BCS     20$                     ;FOUND IT!
    003730' 005301                          DEC     R1                      ;DECR BIT COUNT
    003732' 001374                          BNE     10$                     ;TRY AGAIN
    003734' 005301                  20$:    DEC     R1                      ;ARE WE DONE?
    003736' 001415                          BEQ     40$                     ;YES-GO SET ALARM
    003740' 006000                          ROR     R0                      ;CHECK IF MORE BITS SET
    003742' 103374                          BCC     20$                     ;IF NOT KEEP LOOKING
    003744' 005004                  25$:    CLR     R4
    003746' 004567 002622                   JSR     R5,PRINT                ;PRINT ERR MSG
    003752' 000042                          .WORD   ALMMSG-.
    003754' 000056                          .WORD   46.
    003756' 062705 000002                   ADD     #2,R5
    003762' 004767 001742                   JSR     PC,ERRSNM               ;TAKE ERROR EXIT
    003766' 000177 174056                   JMP     @CUPGER
    003772' 012500                  40$:    MOV     (R5)+,R0                ;GET ALARM LEVEL AGAIN
    003774' 016704 174024                   MOV     DREGAD,R4               ;DEV REG ADDR TO R4
    004000' 042764 000037 000016            BIC     #37,16(R4)              ;CLEAR ALARM
    004006' 050064 000016                   BIS     R0,16(R4)               ;UPDATE TO NEW VALUE
    004012' 000205                          RTS     R5                      ;EXIT
    004014' 044523 047514 040440    ALMMSG: .ASCII  /SILO ALARM LEVEL OTHER THAN 0,1,2,4,8,16 OR 32/
    004022' 040514 046522 046040
    004030' 053105 046105 047440
    004036' 044124 051105 052040
    004044' 040510 020116 026060
    004052' 026061 026062 026064
    004060' 026070 033061 047440
    004066' 020122 031063
                                            .EVEN
```

```
                                        ;WRITE & READ CONTROL TABLES

 004072'  000000  000000  000000  SNDTBL: .WORD   0,0,0           ;LINE  0   ADDR/ADDR/BYTE CNT
 004100'  000000  000000  000000          .WORD   0,0,0           ;      1
 004106'  000000  000000  000000          .WORD   0,0,0           ;      2
 004114'  000000  000000  000000          .WORD   0,0,0           ;      3
 004122'  000000  000000  000000          .WORD   0,0,0           ;      4
 004130'  000000  000000  000000          .WORD   0,0,0           ;      5
 004136'  000000  000000  000000          .WORD   0,0,0           ;      6
 004144'  000000  000000  000000          .WORD   0,0,0           ;      7
 004152'  000000  000000  000000          .WORD   0,0,0           ;      8
 004160'  000000  000000  000000          .WORD   0,0,0           ;      9
 004166'  000000  000000  000000          .WORD   0,0,0           ;      10
 004174'  000000  000000  000000          .WORD   0,0,0           ;      11
 004202'  000000  000000  000000          .WORD   0,0,0           ;      12
 004210'  000000  000000  000000          .WORD   0,0,0           ;      13
 004216'  000000  000000  000000          .WORD   0,0,0           ;      14
 004224'  000000  000000  000000          .WORD   0,0,0           ;      15

 004232'  000000                  PSSAVE: .WORD   0               ;PROC STATUS SAVE
 004234'  000000                  MSADDR: .WORD   0
 004236'  000000                  LSADDR: .WORD   0
 004240'  000000                  BYTES:  .WORD   0
 004242'  000000                  TEMTCR: .WORD   0               ;TEMP XMIT CONT REG
 004244'  000000                  STCR:   .WORD   0               ;XMIT SOFT CONT REG
 004246'  000000                  RCR:    .WORD   0               ;RECEIVE CONT REG
 004250'  000000                  ERRFLG: .WORD   0               ;ERROR FLAG


 004252'  000000  000000          RCVTBL: .WORD   0,0             ;LINE  0   ADDR/BYTE CNT
 004256'  000000  000000                  .WORD   0,0             ;      1
 004262'  000000  000000                  .WORD   0,0             ;      2
 004266'  000000  000000                  .WORD   0,0             ;      3
 004272'  000000  000000                  .WORD   0,0             ;      4
 004276'  000000  000000                  .WORD   0,0             ;      5
 004302'  000000  000000                  .WORD   0,0             ;      6
 004306'  000000  000000                  .WORD   0,0             ;      7
 004312'  000000  000000                  .WORD   0,0             ;      8
 004316'  000000  000000                  .WORD   0,0             ;      9
 004322'  000000  000000                  .WORD   0,0             ;      10
 004326'  000000  000000                  .WORD   0,0             ;      11
 004332'  000000  000000                  .WORD   0,0             ;      12
 004336'  000000  000000                  .WORD   0,0             ;      13
 004342'  000000  000000                  .WORD   0,0             ;      14
 004346'  000000  000000                  .WORD   0,0             ;      15
```

```
                                                ;WRITE INTERRUPT HANDLER

004352' 004067  001600          WRINT:  JSR     R0,SAVREG               ;SAVE REGISTERS
004356' 004567  001646                  JSR     R5,STSTAT               ;STORE DEVICE REG CONTENTS
004362' 174400                          .WORD   ISTAT-.
004364' 005267  174664                  INC     WICNT                   ;INCR WRITE INT COUNT
004370' 004767  001614                  JSR     PC,SUPTAD               ;SET UP INTRNAL PTRS
004374' 032714  100000                  BIT     #100000,(R4)            ;XMTR INT?
004400' 001005                          BNE     WHODON                  ;YES - WHO?
004402' 152767  000010  177640          BISB    #8.,ERRFLG              ;REMEMBER NEM ERROR
004410' 005267  174632                  INC     NEMERR                  ;INCR NEM ERROR CNT
004414' 016400  000012          WHODON: MOV     12(R4),R0               ;GET BUFFER ACTIVE REG
004420' 016701  177620                  MOV     STCR,R1                 ;GET PGM TCR
004424' 040001                          BIC     R0,R1                   ;FIND OUT WHAT CHANGED
004426' 001443                          BEQ     WTERM                   ;BRANCH IF NO CHANGE
004430' 040167  177610                  BIC     R1,STCR                 ;RESET WHOEVER CHANGED
004434' 040164  000014                  BIC     R1,14(R4)               ;IN STCR AND BAR REG
004440' 010700                          MOV     PC,R0                   ;POINT R0
004442' 062700  177430                  ADD     #SNDTBL-.,R0            ;AT WRITE ADDR TBL
004446' 000241                          CLC
004450' 005002                          CLR     R2
004452' 006001                  10$:    ROR     R1                      ;TEST LINE # BIT
004454' 103404                          BCS     30$
004456' 062700  000006          20$:    ADD     #6,R0                   ;FIND ADDR ENTRY
004462' 005202                          INC     R2                      ;AND LINE NBR
004464' 000772                          BR      10$
004466' 042714  000017          30$:    BIC     #17,(R4)                ;CLEAR LINE BITS
004472' 050214                          BIS     R2,(R4)                 ;SELECT LINE
004474' 016403  000010                  MOV     10(R4),R3               ;GET CURRENT BYTE CNT
004500' 005403                          NEG     R3                      ;COMPLIMENT IT
004502' 016067  000004  000120          MOV     4(R0),WRBCNT            ;GET INIT BYTE CNT
004510' 160367  000114                  SUB     R3,WRBCNT               ;SUB CUR FROM INIT
004514' 010703                          MOV     PC,R3
004516' 062703  174512                  ADD     #BYWR+2-.,R3
004522' 066713  000102                  ADD     WRBCNT,(R3)             ;UPDATE BYTES
004526' 005543                          ADC     -(R3)                   ;WRITTEN COUNT
004530' 005701                          TST     R1                      ;ANY MORE LINES DONE?
004532' 001361                          BNE     20$                     ;YES
004534' 000727                          BR      WHODON                  ;MAYBE
004536' 042714  100000          WTERM:  BIC     #100000,(R4)            ;RESET XMIT INT
004542' 016367  000002  173250          MOV     2(R3),SIZE              ;UPDATE ACTUAL BYTES XFERRED
004550' 005767  177470                  TST     STCR                    ;ALL LINES QUIET?
004554' 001401                          BEQ     STOPWR                  ;YES - STOP ALL WRITES
004556' 000420                          BR      INTEXT                  ;RETURN FROM INTERRUPT
004560' 042767  040000  173214  STOPWR: BIC     #WRBSY,FLAGWD           ;RESET WRITE BUSY
004566' 042714  020000                  BIC     #20000,(R4)             ;RESET XMIT INT EBL
004572' 004767  001412                  JSR     PC,SUPTAD
004576' 032714  000100                  BIT     #100,(R4)               ;RECEIVE INT STILL UP ?
004602' 001003                          BNE     GVECAD                  ;YES - CONTINUE
004604' 042763  000010  000000          BIC     #WT4IOT,PFLGWD(R3)      ;NO - RESET WAIT FOR IOT
004612' 052767  000004  173162  GVECAD: BIS     #CLWVCT,FLAGWD          ;SET CLR VECT REQUEST
004620' 004067  001346          INTEXT: JSR     R0,RESREG               ;RESTORE REGISTERS
004624' 000177  173244                  JMP     @RTNINT                 ;RETURN FROM INTERRUPT
004630' 000000                  WRBCNT: .WORD   0                       ;WRITE BYTE COUNT
```

```
                                        ;READ INTERRUPT HANDLER
                                RDINT:  JSR     R0,SAVREG               ;SAVE REGISTERS
                                        JSR     R5,STSTAT               ;STORE DEVICE REG CONTENTS
                                        .WORD   .STAT-.
                                        INC     RICNT                   ;INCR READ INT COUNT
                                        JSR     PC,SUPTRD               ;SET UP INTERN PTRS
                                        BIT     #40000,(R4)             ;SILO OVERFLOW?
                                        BNE     REMOFL                  ;YES-REMEMBER IT
                                        MOV     @R1,R1                  ;GET STORED CHAR
                                        BR      FRSBYT                  ;PROCESS IT
                        177352  REMOFL: BISB    #16.,ERRFLG             ;REMEMBER SILO OVERFL
                                        INC     SOFERR                  ;INCR SILO OVERFL COUNT
                                        BR      STOPRD                  ;ABORT READ
                                GTBYTE: MOV     2(R4),R1                ;GET DATA FROM RBUF
                                FRSBYT: BPL     CHKLVL                  ;BR IF [illegible] NOT VALID
                                        MOV     R1,R2                   ;SAVE IN R2
                                        SWAB    R1
                                        BIC     #177760,R1              ;FILTER OUT OTHER BITS
                                        ASL     R1
                                        ASL     R1                      ;USE AS TABLE INDEX
                                        MOV     PC,R3
                                        ADD     @RCVTBL-.,R3            ;POINT R0 AT TBL ENTRY
                                        ADD     R1,R3                   ;ADD INDEX
                                        TST     2(R3)                   ;TEST BYTE COUNT
                                        BEQ     RTERM                   ;BRANCH IF ZERO
                                        MOV     PC,R1
                                        ADD     @BYRD+2-.,R1
                                        ADD     #1,(R1)                 ;INCR BYTES READ BY 1
                                        ADC     -(R1)
                                        BIT     #100000,R2              ;TEST FOR PARITY ERROR
                                        BEQ     NOPERR                  ;BRANCH IF NOT
                                        INC     PARERR                  ;OTHERWISE INCR COUNT
                        177250          BISB    #1,ERRFLG               ;REMEMBER PARITY ERROR
                        020000  NOPERR: BIT     #20000,R2               ;TEST FOR FRAMING ERROR
                                        BEQ     NOFERR                  ;BRANCH IF NOT
                                        INC     FRAMER                  ;OTHERWISE INCR COUNT
                        177230          BISB    #2,ERRFLG               ;REMEMBER FRAMING ERROR
                        040000  NOFERR: BIT     #40000,R2               ;TEST FOR OVERRUN ERROR
                                        BEQ     NOOERR                  ;BRANCH IF NOT
                                        INC     OVRRUN                  ;OTHERWISE INCR COUNT
                        177210          BISB    #4,ERRFLG               ;REMEMBER OVERRUN ERROR
                                NOOERR: MOV     (R3)+,R0
                                        MOVB    R2,R1
                        173030          JSR     PC,@PUTBYT              ;MOVE BYTE TO MEMORY
                        177776          INC     -2(R3)                  ;INCREMENT ADDRESS
                                        DEC     (R3)                    ;DECREMENT BYTE COUNT
                                        BEQ     RTERM                   ;TERMINATE IF BC ZERO
                                        BR      CHKLVL
                                RTERM:  MOV     PC,R1
                        174140          ADD     @BYRD+2-.,R1
                        172724          MOV     (R1),SIZE               ;UPDATE ACTUAL BYTES XFERRED
                        000001          MOV     #1,R1
                                        SWAB    R2
                        177760          BIC     #177760,R2              ;FILTER ALL BUT LINE NBR
                                RTRMLP: TSTB    R2
```

```
                                                BEQ     STRSCN
                                                ASL     R1                      ;SET BIT AS PER LINE NBR
                                                DEC     R2
                                                BNE     RTRYLP
005120' 040167  177122          STRSCN:         BIC     R1,RCR                  ;RESET LINE NBR BIT
005124' 005767  177116                          TST     RCR                     ;ALL LINES DISABLED ?
005130' 001404                                  BEQ     STOPRD                  ;YES - STOP ALL READS
005132' 116401  000017          CHKLVL:         MOVB    17(R4),R1               ;GET SILO FILL LEVEL
005136' 001262                                  BNE     GTBYTE                  ;GO AGAIN IF NOT EMPTY
005140' 000027                                  BR      INTEXT                  ;RETURN FROM INTERRUPT
005142' 042767  020000  172632  STOPRD:         BIC     #RDBSY,FLAG40           ;RESET READ BUSY
005150' 042714  010100                          BIC     #10100,(R4)             ;RESET READ INT ENBL
005154' 004767  001030                          JSR     PC,SUPTAB
005160' 032714  020000                          BIT     #20000,(R4)             ;TEST IF WRITE STILL UP
005164' 001003                                  BNE     COMBER                  ;IF YES-GO COMBINE ERRORS
005166' 042763  000010  000000                  BIC     #WAIT4IOT,PFLAG40(R3)   ;RESET WAIT FOR IOT
005174' 032763  000400  000002  COMBER:         BIT     #DDERCK,PDPSW(R3)       ;TEST DONT ERROR CK BIT
005202' 001000                                  BNE     RERVEC                  ;BRANCH IF SET
005204' 052767  000002  172570  RERVEC:         BIS     #CLRVCT,FLAG40          ;SET CLR RD VECT REQUEST
005212' 000167  177402                          JMP     INTEXT                  ;EXIT


005216' 016767  172604  000004  RRINTV:         MOV     IVCTAD,IOS              ;RESET READ VECT SUBR
005224' 004577  172640                          JSR     R5,@CLRVEC
005230' 000000                  IOS:            .WORD   XXXX
005232' 000207                                  RTS     PC


00523N' 016767  172566  000012  RWINTV:         MOV     IVCTAD,IOS              ;RESET WRITE VECT SUBR
005242' 062767  000004  000004                  ADD     #4,IOS
005250' 004577  172614                          JSR     R5,@CLRVEC
005254' 000000                  IOS:            .WORD   XXXX
005256' 000207                                  RTS     PC
```

```
                                .SBTTL  DH11 SUPPORT ROUTINES

                                ;DEVICE ROUTINE HOUSEKEEPING

                                        ;JSR    R5,HSKEEP       S/R CALL
                                        ;.WORD  0 OR 1          0 = DO HSKP PER OPSW
                                        ;                       1 = UNCOND. DO HSKP
                                        ;R2 = PROG'S OPSW
                                        ;DESTROYS R0,R1

005260' 005725          HSKEEP: TST     (R5)+                   ;UNCONDITIONALLY DO HSKP?
005262' 001003                  BNE     10$                     ;N,Y-10$
005264' 032702 000004           BIT     #HSKPEP,R2              ;OPSW SPECIFY EACH PASS HSKP?
005270' 001010                  BNE     30$                     ;Y,N-30$
005272' 010700          10$:    MOV     PC,R0                   ;SET UP FIRST WD ADR
005274' 062700 173666           ADD     #HSKPST-.,R0
005300' 012701 000042           MOV     #HSKPEN-HSKPST/2,R1     ;SET UP # OF WORDS
005304' 005020          20$:    CLR     (R0)+                   ;HSKP ALL NECESSARY AREAS
005306' 005301                  DEC     R1
005310' 001375                  BNE     20$
005312' 000205          30$:    RTS     R5                      ;EXIT IN-LINE

                                ;DH11 REPORT ROUTINE

                                        ;JSR    R5,REPORT       S/R CALL
                                        ;.WORD  FLAGWORD        FLAGWORD
                                        ;                         BIT 15 = CMND MODE CALL
                                        ;                         BIT  9 = PROG STRNT CALL
                                        ;                         BIT  1 = DO STATUS REPORT
                                        ;                         BIT  0 = DO COUNTS REPORT

005314' 005067 000236   REPORT: CLR     ABBREV                  ;CLR ABBREVIATED RPT FLAG
005320' 004067 000632           JSR     R0,SAVREG               ;SAVE REG'S R0 - R5
005324' 032715 177776           BIT     #177776,(R5)            ;DISPLAYING CNTS AT END OF
005330' 001012                  BNE     8$                      ;PROG PASS? (Y,N-8$)
005332' 010700                  MOV     PC,R0                   ;SET UP ADR OF CNTS
005334' 062700 173666           ADD     #BYTO-.,R0
005340' 012701 000020           MOV     #16.,R1                 ;GET # OF CNT WORDS
005344' 005720          25:     TST     (R0)+                   ;THIS CNT WORD = 0?
005346' 001003                  BNE     8$                      ;Y,N-8$
005350' 005301                  DEC     R1                      ;DECR WORD CNT
005352' 001374                  BNE     25                      ;CK'ED ALL WORDS? (Y,N-25)
005354' 000474                  BR      DVREX                   ;GO TO EXIT -- ALL CNTS ARE 0'S
005356' 004767 000626   8$:     JSR     PC,SUPTAD               ;SET UP PROG TBL ADR IN R3
005362' 012504                  MOV     (R5)+,R4                ;GET FLAGWORD
005364' 032704 000002           BIT     #2,R4                   ;GOING TO DO STATUS DISPLAY?
005370' 001403                  BEQ     10$                     ;Y,N-10$
005372' 004567 000632           JSR     R5,STSTAT               ;GO STORE STATUS REG'S
005376' 173604                  .WORD   CSTAT-.
005400' 004767 001010   10$:    JSR     PC,DISUNM               ;DISPLAY CURR UNIT #
005404' 032704 000002           BIT     #2,R4                   ;DISPLAY DEV STATUS?
005410' 001416                  BEQ     DISCNT                  ;Y,N-DISCNT
005412' 004567 001156           JSR     R5,PRINT                ;ISSUE "AT INT" MSG
005416' 001365                  .WORD   AT!MSG-.
```

```
005420' 000014                  .WORD   12.
005422' 004567 001056           JSR     R5,DISPST       ;GO DISPLAY STATUS AT LAST INT
005426' 173524                  .WORD   1STAT-.
005430' 004567 001140           JSR     R5,PRINT        ;ISSUE "CURRENTLY" MSG
005434' 001363                  .WORD   CURMSG-.
005436' 000012                  .WORD   10.
005440' 004567 001040           JSR     R5,DISPST       ;GO DISPLAY CURRENT STATUS
005444' 173546                  .WORD   CSTAT-.
005446' 032704 000001 DISCNT:   BIT     #1,R4           ;DISPLAY COUNTS?
005452' 001431                  BEQ     RPTEND          ;Y,N=RPTEND
005454' 012700 000020           MOV     #16.,R0         ;SET UP # OF WORDS
005460' 010701                  MOV     PC,R1           ;SET UP ADR OF CNTS
005462' 062701 173540           ADD     #BYRD-.,R1
005466' 010702                  MOV     PC,R2           ;SET UP TBL ADR
005470' 062702 000070           ADD     #REPTBL-.,R2
005474' 012267 000012 RPTLP:    MOV     (R2)+,RPTBAS    ;MOV MSG ADR TO S/R LINKAGE
005500' 004467 000452           JSR     R0,SAVREG       ;SAVE ALL REG'S
005504' 011100                  MOV     (R1),R0         ;GET CURRENT COUNT
005506' 004567 172344           JSR     R5,2BINASC      ;CONVERT IT TO ASCII
005512' 000000 RPTBAS:  .WORD   XXXX
005514' 004467 000452           JSR     R0,RESREG       ;RESTORE REG'S
005520' 005721                  TST     (R1)+           ;POINT AT NXT CNT
005522' 005300                  DEC     R0              ;DONE ALL WORDS?
005524' 001363                  BNE     RPTLP           ;Y,N=RPTLP
005526' 004567 001042           JSR     R5,PRINT        ;GO ISSUE COUNTS MSG
005532' 001366                  .WORD   CNTSMG-.
005534' 000412                  .WORD   CNTSEN-CNTSMG
005536' 004567 001032 RPTEND:   JSR     R5,PRINT        ;ISSUE "END OF REPORT" MSG
005542' 001367                  .WORD   REMSG-.
005544' 177763                  .WORD   -13.
005546' 004067 000420 DVREX:    JSR     R0,RESREG       ;RESTORE REGISTERS
005552' 005725                  TST     (R5)+           ;SET UP RETURN POINT
005554' 000205                  RTS     R5              ;EXIT IN-LINE

005556' 000000 ABBREV:  .WORD   0

005560' 001412 REPTBL:  .WORD   BCNRO-RPTBAS
005562' 001420                  .WORD   BCNRO+6-RPTBAS
005564' 001434                  .WORD   BCNAR-RPTBAS
005566' 001442                  .WORD   BCNAR+6-RPTBAS
005570' 001467                  .WORD   CNOCRO-RPTBAS
005572' 001502                  .WORD   CNOCBK-RPTBAS
005574' 001516                  .WORD   CNOCBK-RPTBAS
005576' 001533                  .WORD   CNOCHS-RPTBAS
005600' 001565                  .WORD   RDINHS-RPTBAS
005602' 001606                  .WORD   WRINHS-RPTBAS
005604' 001627                  .WORD   ERCHEN-RPTBAS
005606' 001647                  .WORD   ERCOVR-RPTBAS
005610' 001667                  .WORD   ERCFRM-RPTBAS
005612' 001706                  .WORD   ERCPAR-RPTBAS
005614' 001746                  .WORD   ERCSOF-RPTBAS
005616' 002002                  .WORD   ERCDTA-RPTBAS
```

```
                                        ;DH11 ERROR REPORT ROUTINE

005620' 004767 000004          ERRRPT: JSR     PC,ERRDIS
005624' 000177 172220                  JMP     @#KDUMGER
005630' 010701                 ERRDIS: MOV     PC,R1                   ;POINT R1 AT ERR MSG
005632' 062701 000236                  ADD     #EMSGBF-.,R1
005636' 012767 000014 000054           MOV     #12.,ERRBCT
005644' 010700                         MOV     PC,R0                   ;POINT R0 AT ERR MSG TBL
005646' 062700 000144                  ADD     #ERRCOTB-.,R0
005652' 105710                 1$:     TSTB    (R0)
005654' 001416                         BEQ     ERTBEN                  ;BRANCH IF R0 AT TBL END
005656' 132067 176366                  BITB    (R0)+,ERRFLG            ;TEST FOR PARTICULAR ERR
005662' 001003                         BNE     3$                      ;BRANCH IF FOUND
005664' 062700 000005          2$:     ADD     #5,R0
005670' 000770                         BR      1$
005672' 012702 000005          3$:     MOV     #5,R2
005676' 112021                 4$:     MOVB    (R0)+,(R1)+             ;MOVE MSG CODE TO ERR MSG
005700' 005267 000014                  INC     ERRBCT                  ;BUMP BYTE COUNT
005704' 005302                         DEC     R2
005706' 001373                         BNE     4$
005710' 000760                         BR      1$                      ;CHECK IF MORE
005712' 004567 000656          ERTBEN: JSR     R5,PRINT                ;PRINT ERROR MSG
005716' 000136                         .WORD   EMSGHD-.
005720' 000014                 ERRBCT: .WORD   12.
005722' 004567 000556          ERDIAG: JSR     R5,DISPST               ;DISPLAY DEVICE REGS
005726' 173234                         .WORD   ISTAT-.
005730' 016300 000022          ERRSNM: MOV     PSRCST(R3),R0           ;GET ADDR OF SRC STMNTS
005734' 111001                 10$:    MOVB    (R0),R1                 ;SAVE STMNT LENGTH
005736' 026067 003004 000106           CMP     4(R0),STMNT             ;ERROR OCCUR ON THIS STMNT?
005744' 001402                         BEQ     20$                     ;YES - BRANCH
005746' 060100                         ADD     R1,R0                   ;POINT AT NEXT STATEMENT
005750' 000771                         BR      10$                     ;GO CK NEXT STMNT
005752' 005720                 20$:    TST     (R0)+                   ;SET UP ADR OF STMNT # DATA
005754' 010701                         MOV     PC,R1                   ;SET UP DATA OUTPUT ADDR
005756' 062701 000172                  ADD     #STMNUM-.,R1
005762' 004577 172074                  JSR     R5,$DECASC              ;CONVERT IT TO ASCII
005766' 012767 020040 000160           MOV     #20040,STMNUM+4         ;SET 2 LOW DIGITS TO SPACES
005774' 004567 000574                  JSR     R5,PRINT                ;ISSUE STMNT # MSG
006000' 000140                         .WORD   STMNMG-.
006002' 177762                         .WORD   -14.
006004' 005067 176240                  CLR     ERRFLG                  ;CLEAR ERROR FLAG
006010' 000207                         RTS     PC

006012' 020001 050040 051101   ERRCDTB: .ASCII  <001>/  PAR/           ;ERROR MSG  CODE TABLE
006020' 020002 043040 046522           .ASCII  <002>/  FRM/
006026' 020004 047440 051126           .ASCII  <004>/  OVR/
006034' 020010 047040 046505           .ASCII  <010>/  NEM/
006042' 020020 051440 043117           .ASCII  <020>/  SOF/
006050' 000                            .BYTE   0
        006052'                        .EVEN

006052' 000000                 STMNT:  .WORD   0                       ;SAVED R5 FOR STMNT #

006054' 044104 030461 042440   EMSGHD: .ASCII  /DH11 ERROR: /
        006062' 051122 051117 020072
```

```
 1201  006070' 000050                    EMSGBF: .BLKB   40.
 1202  006140' 052123  047115  020124    STMNMG: .ASCII  /STMNT #/
       006146' 020043
 1203  006150' 054130  054130  054130    STMNUM: .ASCII  /XXXXXX/
```

```
                                        .SBTTL  SUBROUTINES FOR DH11 DEVICE ROUTINE

                                        ;SAVE REGISTERS R0 THRU R5

                                                ;JSR    R0,SAVREG       S/R CALL

006154' 010146                  SAVREG: MOV     R1,-(SP)                ;SAVE R0 THRU R5
006156' 010246                          MOV     R2,-(SP)
006160' 010346                          MOV     R3,-(SP)
006162' 010446                          MOV     R4,-(SP)
006164' 010546                          MOV     R5,-(SP)
006166' 010046                          MOV     R0,PC                   ;EXIT IN-LINE

                                        ;RESTORE REGISTERS R0 THRU R5

                                                ;JSR    R0,RESREG       S/R CALL

006170' 005726                  RESREG: TST     (SP)+                   ;RESTORE R4 THRU R0
006172' 012605                          MOV     (SP)+,R5
006174' 012604                          MOV     (SP)+,R4
006176' 012603                          MOV     (SP)+,R3
006200' 012602                          MOV     (SP)+,R2
006202' 012601                          MOV     (SP)+,R1
006204' 000200                          RTS     R0                      ;EXIT IN-LINE

                                        ;SET PROGRAM'S PROG TABLE ADR IN R3

                                                ;JSR    PC,SUPTAD       S/R CALL

006206' 010703                  SUPTAD: MOV     PC,R3                   ;SET UP LOCATION ZERO ADR
006210' 062703  171566                  ADD     #LOC2-.,R3
006214' 166303  177776                  SUB     -2(R3),R3               ;SUBTRACT PROG TBL LENGTH
006220' 016704  171576                  MOV     DREGAD,R4               ;PUT DEV REG ADR IN R4
006224' 000207                          RTS     PC                      ;EXIT IN-LINE

                                        ;STORE DEVICE'S STATUS REGISTERS

                                                ;JSR    R5,STSTAT       S/R CALL
                                                ;.WORD  STADR-.         REL STORAGE ADR
                                                ;DESTROYS R0,R1,R2

006230' 016701  171570          STSTAT: MOV     DREGAD,R1
006234' 010500                          MOV     R5,R0                   ;GET REL STORAGE ADR & MAKE
006236' 062500                          ADD     (R5)+,R0                ;IT ABSOLUTE
006240' 012746  000010                  MOV     #OVREGE-OVREGS/6,-(SP)  ;GET # OF REG'S TO STORE
006244' 010702                          MOV     PC,R2                   ;GET ADR OF 1ST REG DISPLACEMENT
006246' 062702  171654                  ADD     #OVREGS+4-.,R2
006252' 011201                  10$:    MOV     (R2),R1                 ;GET REG DISPLACEMENT
006254' 066701  171544                  ADD     DREGAD,R1               ;ADD IN REG'S BASE ADR
006260' 011120                          MOV     (R1),(R0)+              ;STORE REGISTER VALUE
006262' 062702  000006                  ADD     #6,R2                   ;POINT AT NXT DISPLACEMENT
006266' 005316                          DEC     (SP)                    ;DECR REG CNT
006270' 001370                          BNE     10$                     ;DONE ALL? (Y,N-10$)
```

```
006272' 005726                              TST     (SP)+                   ;CLEAN UP THE STACK
006274' 000205                              RTS     R5                      ;EXIT IN-LINE

                                    ;LINE PARAMETER REGISTER UPDATE SUBROUTINE

                                            ;JSR    R5,LPRUPD       S/R CALL
                                            ;
                                            ;R0 CONTAINS PARAMETER
                                            ;R1 CONTAINS BIT MASK
                                            ;DESTROYS R2,R3,R4

006276' 010067  [illegible]106      LPRUPD: MOV     R0,PARAM                ;SAVE PARAMETER
[illegible]302' 010167  [illegible]04       MOV     R1,PMASK                ;AND PARAM MASK
[illegible]306' 016704  1[illegible]        MOV     DREGAD,R4               ;POINT R4 AT DEV REGS
[illegible]312' 004767  17[illegible]       JSR     PC,SUPTAD               ;POINT R3 AT P TABLE
[illegible]316' 016700  17[illegible]       MOV     LINE,R0                 ;CHECK SIMULTANEOUS LINES
[illegible]322' 001003                      BNE     10$                     ;BRANCH IF ANY SET
[illegible]   116301  000035                MOVB    PCURDV(R3),R1           ;ELSE GET ASSIGNED LINE
[illegible]   [illegible]                   BR      20$
[illegible]   [illegible]           10$:    CLR     R1                      ;CONVERT BIT POSITION
[illegible]   012702  000001                MOV     #1,R2                   ;REPRESENTING LINE #
[illegible]   [illegible]           20$:    BIT     R2,R0                   ;TO
[illegible]   [illegible]                   BNE     30$                     ;BINARY NBR IN R1
[illegible]   [illegible]                   INC     R1
[illegible]   [illegible]                   ROL     R2
[illegible]   [illegible]                   BCC     20$
[illegible]   [illegible]           30$:    BIC     R2,R0                   ;CLEAR LINE BIT
[illegible]   [illegible]714  000017        BIC     #17,(R4)                ;CLEAR LINE SELECT
[illegible]   [illegible]114                BIS     R1,(R4)                 ;JAM INTO LINE SELECT
[illegible]   016701  000022                MOV     PARAM,R1                ;GET PARAMETER
[illegible]   016702  [illegible]           MOV     PMASK,R2                ;AND MASK
[illegible]   [illegible]  [illegible]      BIC     R2,4(R4)                ;CLEAR PARAM BIT IN P REG
[illegible]   [illegible]  000004           BIS     R1,4(R4)                ;JAM PARAM IN P REG
[illegible]402' 005700                      TST     R0                      ;ANY MORE LINES?
[illegible]404' 001[illegible]              BNE     10$                     ;YES - DO IT AGAIN
006406' 000205                              RTS     R5                      ;NO - EXIT
006410' 000000                      PARAM:  .WORD   0
006412' 000000                      PMASK:  .WORD   0

                                    ;DISPLAY CURRENT UNIT #

                                            ;JSR    PC,DISUNM       S/R CALL
                                            ;R3 MUST CONTAIN PROG TBL ADR
                                            ;DESTROYS R0,R1,R2

006414' 012767  000022  000056      DISUNM: MOV     #18.,DISUML             ;INIT TO NORM MSG LNTH
006422' 116300  000035                      MOVB    PCURDV(R3),R0           ;GET CURRENT UNIT #
006426' 020027  000020                      CMP     R0,#16.                 ;VALID UNIT #?
006432' 101007                              BHI     DISUIV                  ;Y,N-DISUIV
006434' 004577  171420                      JSR     R5,@OTASLZ              ;CONVERT # TODECIMAL ASCII
006440' 000426                              .WORD   UNASCI-.
006442' 016767  000424  000416              MOV     UNASCI+4,UNASCI         ;MOVE ASCII # TO 1ST TWO DIGITS
006450' 000410                              BR      DISUPR
006452' 012767  000026  000020      DISUIV: MOV     #22.,DISUML             ;SET UP ERR COND MSG LNTH
006460' 042700  177400                      BIC     #177400,R0              ;RESET HIGH BYTE
```

```
006464' 004577 171366            JSR     R5,2$BINASC             ;CONVERT BINARY TO ASCII
006470' 000376                   .WORD   UNASCI-.
006472' 004567 000076    DISUPR: JSR     R5,PRINT                ;GO ISSUE UNIT # MSG
006476' 000350                   .WORD   UNITMG-.
006500' 000020           DISUNL: .WORD   16.
006502' 000207                   RTS     PC                      ;EXIT INLINE




                                 ;TAILOR STATUS MSG & PRINT IT

                                         ;JSR    R5,DISPST       S/R CALL
                                         ;WORD   STATADR-.       REL ADR OF STATUS DATA
                                         ;DESTROYS R0,R1,R2

006504' 010502           DISPST: MOV     R5,R2                   ;GET REL DATA ADR
006506' 062502                   ADD     (R5)+,R2                ;MAKE IT ABS
006510' 010701                   MOV     PC,R1                   ;SET UP ADR OF REG NAMES IN ASCII
006512' 062701 171404            ADD     #DVREGS-.,R1
006516' 012700 000010            MOV     #DVREGE-DVREGS/6,R0     ;GET # OF REGISTERS TO DISPLAY
006522' 012167 000346    10$:    MOV     (R1)+,DVRGMG            ;MOVE REG NAME TO MSG
006526' 012167 000344            MOV     (R1)+,DVRGMG+2
006532' 005721                   TST     (R1)+                   ;BYPASS DISP VALUE
006534' 004067 177416            JSR     R0,SAVREG               ;SAVE REG'S R0 - R5
006540' 011200                   MOV     (R2),R0                 ;GET REG'S STORED VALUE
006542' 004577 171310            JSR     R5,2$BINASC             ;CONVERT IT TO ASCII
006546' 000334                   .WORD   DVRGDT-.
006550' 004567 000020            JSR     R5,PRINT                ;PRINT THE STATUS MSG
006554' 000320                   .WORD   DVRGMG-.
006556' 000014                   .WORD   12.
006560' 004067 177406            JSR     R0,RESREG               ;RESTORE R0 - R4
006564' 005722                   TST     (R2)+                   ;POINT AT NXT REG VALUE
006566' 005300                   DEC     R0                      ;DECR REG CNT
006570' 001354                   BNE     10$                     ;DONE ALL? (Y,N-10$)
006572' 000205                   RTS     R5                      ;EXIT IN-LINE
```

```
                                        ;ISSUE MSG TO LIST DEVICE
                                        ;
                                        ;JSR    R5,PRINT        S/R CALL
                                        ;.WORD  MSGADR-.        REL ADR OF MSG
                                        ;.WORD  BYTCNT          MSG BYTE CNT (IF NEGATIVE,
                                        ;                         RESET PRT DEV DEDICATED.)
                                        ;R3 = PROG TBL ADR
                                        ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
                                        ;DESTROYS R0,R1,R2
006574' 010500                  PRINT:  MOV     R5,R0           ;GET MSG ADR & MAKE IT ABS
006576' 062700                          ADD     (R5)+,R0
006602' 011501                          MOV     (R5)+,R1        ;GET BYTE COUNT
006604' 005704                          TST     R4              ;USE CMND MODE PRINT?
006606' 100003                          BPL     40$             ;Y,N=40$
006610' 010702                          MOV     PC,R2           ;SET UP LINK INFO ADR
006612' 062702  000040                  ADD     #20$-.,R2
006616' 160200                          SUB     R2,R0           ;MAKE MSG ADR REL
006620' 010012                          MOV     R0,(R2)+        ;STORE MSG ADR
006622' 010112                          MOV     R1,(R2)         ;STORE MSG'S BYTE COUNT
006624' 100001                          BPL     10$             ;CNT REG=.+2,N=10$)
006626' 005412                          NEG     (R2)            ;MAKE +1 POSITIVE
006630' 016302  000006  000144  10$:    MOV     PRSCIN(R3),PRUNIT       ;STORE PROG'S # IN MSG
006636' 004567  171214                  JSR     R5,2CLIST       ;ISSUE PROG #
006642' 000136                          .WORD   PNMSG-.
006644' 000005                          .WORD   5
006646' 004567  171204                  JSR     R5,2CLIST       ;ISSUE MSG SPECIFIED
006652' 000000                  20$:    .WORD   XXXX
006654' 000000                          .WORD   XXXX
006656' 004567  171174                  JSR     R5,2CLIST       ;ISSUE A <CR> & <LF>
006662' 000142                          .WORD   CRLF-.
006664' 000002                          .WORD   2
006666' 000410                          BR      PRTEX           ;GO TO EXIT
006670' 012767  000010  000006  40$:    MOV     #10,6           ;STORE MSG'S ABS ADR
006676' 012767  000006  000004          MOV     #6,4            ;STORE ITS BYTE CNT
006704' 004567  171150                  JSR     R5,2ALIST       ;GO TO MPG TO ISSUE THE MSG
006710' 000000                  50$:    .WORD   XXXX
006712' 000000                  60$:    .WORD   XXXX
006714' 000205                  PRTEX:  RTS     R5              ;EXIT IN-LINE

                                        ;TEST READ INTERRUPT VECTOR S/R
006716' 016767  171112  000010  TRVECT: MOV     IVCTR0,20$      ;GET CURR INT VECT ADR
006724' 016746  000004          MOV     PFWORD(R3),-(SP)        ;STORE FLAG ADR TO IDENTIFY ME
006730' 004567  171144          JSR     R5,2TSTVEC              ;DO I HAVE VECTOR CONTROL?
006734' 000000                  20$:    .WORD   XXXX            ;  MPG WILL TELL ME SINCE I CAN'T
006736' 175202                  .WORD   @#INT-.                 ;  GET AT LOWER MEM IF MEM MGMT
006740' 000401                  BR      TRVEXT                  ;BR IF I DON'T HAVE CONTROL
006742' 005726                  TST     (R5)+                   ;BYPASS BR INST IN S/R CALL
006744' 000205                  TRVEXT: RTS     R5              ;EXIT IN-LINE

                                        ;TEST WRITE INTERRUPT VECTOR S/R
006746' 016767  171062  000016  TWVECT: MOV     IVCTR0,20$      ;GET CURR INT VECT ADR
006754' 062767  000004  000010          ADD     #4,20$          ;ADJUST FOR WRITE INT
```

```
1410  006754' 016346 000004          MOV     PFVADR(R3),-(SP)        ;STORE FLAG ADR TO IDENTIFY ME
1411  006760' 004577 171106          JSR     R5,@TSTVEC              ;DO I HAVE VECTOR CONTROL?
1412  006764' 000000                 208:    .WORD   XXXX            ;   MPG WILL TELL ME SINCE I CAN'T
1413  006766' 173644                  .WORD   NRINT-.                 ;   GET AT LOWER MEM IF MEM RIGHT
1414  006770' 000401                  BR      TWEXT                   ;BR IF I DONT HAVE CONTROL
1415  006772' 005725                  TST     (R5)+                   ;BYPASS BR INST IN S/R CALL
1416  006774' 000205                  TWEXT:  RTS     R5              ;RETURN IN-LINE
1417
1418
```

```
                                        .SBTTL  MESSAGE STORAGE AREA

                                        .NLIST  BEX

                                        .EVEN
006776' 021520                  PRMMSG: .ASCII  /P#/
007000' 054130     011          PROGMR: .ASCII  /XX/<011>
007003'    101  020124  040514  ATIMSG: .ASCII  /AT LAST INT:/
007017'    103  051125  042522  CURMSG: .ASCII  /CURRENTLY:/
007031'    105  042116  047440  RENDMG: .ASCII  /END OF REPORT/
                                        .EVEN
007046' 025052  025052  042040  UNITMG: .ASCII  /**** DH11 UNIT: /
                                        .EVEN
007066' 054130  054130  054130  UNASCI: .ASCII  /XXXXXX/
007074' 054130  054130  020075  DVAGMS: .ASCII  /XXXX= /
007102' 054130  054130  054130  DVAGDT: .ASCII  /XXXXXX/

007110' 054502  042524  035123  CNTSMG: .ASCII  /BYTES:  RD= /
007124' 054130  054130  054130  BCMRD:  .ASCII  /XXXXXXXXXXXX  WR= /
007146' 054130  054130  054130  BCMWR:  .ASCII  /XXXXXXXXXXXX/
007162' 005015                  CRLF:   .ASCII  <015><012>

007164' 041411  047115  051504          .ASCII  <011>/CMNDS:  RD= /
007201'    130  054130  054130  CMDCRD: .ASCII  /XXXXXX WR= /
007214' 054130  054130  054130  CMDCWR: .ASCII  /XXXXXX BRK= /
007230' 054130  054130  054130  CMDCBK: .ASCII  /XXXXXX MISC= /
007245'    130  054130  054130  CMDCMS: .ASCII  /XXXXXX/<015><012>
007255'    011  047111  042524          .ASCII  <011>/INTERRUPTS:  RD= /
007277'    130  054130  054130  RDINMS: .ASCII  /XXXXXX WR= /
007312' 054130  054130  054130  WRINMS: .ASCII  /XXXXXX/<015><012>

007322' 042411  051122  051117  ERRMSG: .ASCII  <011>/ERRORS:  NEM= /
007341'    130  054130  054130  ERCNEM: .ASCII  /XXXXXX OVERRUN= /
007361'    130  054130  054130  ERCOVR: .ASCII  /XXXXXX FRAMING= /
007401'    130  054130  054130  ERCFRM: .ASCII  /XXXXXX PARITY= /
007420' 054130  054130  054130  ERCPAR: .ASCII  /XXXXXX/<015><012>
007430' 004411  020040  020040          .ASCII  <011><011>/        SILO OVERFLOW= /
007460' 054130  054130  054130  ERCSOF: .ASCII  /XXXXXX/
007466' 020011  020040  020040          .ASCII  <011>/          DATA ERRORS = /
007514' 054130  054130  054130  ERCDTA: .ASCII  /XXXXXX/
        007522'                 CNTSEN= .
                                        .EVEN

                                        .LIST   BEX

        007522'                 DVAEND= .
```

```
                                .SBTTL  FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

                        ;       PROGRAM TABLE FORMAT

000242                  PTLGTH= 162.    ;PROGRAM TABLE LENGTH - NON MEM MGMNT VERSION OF MPG
                        ;(PTLGTH= 212.  ;PROGRAM TABLE LENGTH - MEM MGMNT VERSION OF MPG)

000000                  PFLGWD= +0.     ;PROGRAM FLAG WORD - 1 WORD

000002                          USRSTP= 2       ; 1 = USER HAS STOPPED THIS PROGRAM
000004                          ERSTOP= 4       ; 1 = AN ERROR HAS STOPPED THIS PROGRAM
000010                          WTNIOT= 10      ; 1 = WAITING FOR I/O TERMINATION
000020                          CTPRIO= 20      ; 1 = CONSOLE OR PRINTER I/O IN PROGRESS
000040                          SETDED= 40      ; 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
000100                          OCPRES= 100     ; 1 = OBJ CODE IS PRESENT
000200                          USEUBM= 200     ; 1 = THIS PROG USES THE UNIBUS MAP (MEM MGMNT ONLY)
100000                          ACTIVE= 100000  ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)

000002                  POPSW=  +2.     ;PROGRAM'S OPERATION SWITCHES - 1 WORD

100000                          STOPER= 100000  ; 1 = STOP PROG EXECUTION UPON ERROR
040000                          CYCPRG= 40000   ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
020000                          PRNOER= 20000   ; 1 = DO NOT PRINT ON ERROR
010000                          BIT12=  10000   ; 0 = NOT USED
004000                          BIT11=  4000    ; 0 = NOT USED
002000                          CYCDVL= 2000    ; 1 = CYCLE THE DEVICE LIST
001000                          STNXTD= 1000    ; 1 = CYCLE ON SAME DEVICE UPON ERROR
000400                          DOERCK= 400     ; 1 = DON'T DO ERROR CHECKING
000200                          SPOPER= 200     ; 1 = DEVICE SPECIAL OPERATION
000100                          BIT6=   100     ; 0 = NOT USED
000040                          DOIOT=  40      ; 1 = DO NOT PERFORM I/O TIMEOUT
000020                          AUTORP= 20      ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
000010                          AUPPEP= 10      ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
000004                          HSKPEP= 4       ; 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
000002                          PFBBOV= 2       ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
000001                          NOCOMP= 1       ; 1 = DO NOT PRINT PROG COMPLETED MSG

000004                  PFWADR= +4.     ;+0;PROGRAM FLAGWORD ADDRESS - 1 WORD

000006                  PASCIN= +6.     ;PROGRAM'S NUMBER IN ASCII - 1 WORD

000010                  PNAME=  +8.     ;PROGRAM'S NAME IN ASCII - 6 BYTES

000016                  PRDIOA= +14.    ;ADDRESS OF READ I/O AREA - 1 WORD

000020                  PWRIOA= +16.    ;ADDRESS OF WRITE I/O AREA - 1 WORD

000022                  PSRCST= +18.    ;SOURCE STATEMENTS START ADDRESS - 1 WORD

000024                  POBJST= +20.    ;OBJECT CODE START ADDRESS - 1 WORD

000026                  PLNGTH= +22.    ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD

000030                  PTOCNT= +24.    ;I/O TIMEOUT COUNT - 1 WORD
```

```
000032                  PMDLCD= +26.    ;DEV ROUT MODEL # CODE - 1 WORD
000034                  PDPNTR= +28.    ;CURRENT DEVICE NUMBER POINTER - 1 BYTE
000035                  PCUROV= +29.    ;CURRENT DEVICE # - 1 BYTE
000036                  PDNUMS= +30.    ;DEVICE NUMBERS - 16 BYTES
000056                  PTEM0=  +46.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000060                  PTEM1=  +48.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000062                  PTEM2=  +50.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000064                  PTEM3=  +52.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000066                  PTEM4=  +54.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000070                  PTEM5=  +56.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000072                  PTEM6=  +58.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000074                  PTEM7=  +60.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000076                  PTEM8=  +62.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000100                  PTEM9=  +64.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000102                  PTEM10= +66.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000104                  PTEM11= +68.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000106                  PTEM12= +70.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000110                  PTEM13= +72.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000112                  PTEM14= +74.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000114                  PTEM15= +76.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
000116                  PNBR=   +78.    ;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
000120                  PSRC=   +80.    ;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
000122                  PDST=   +82.    ;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
000124                  PSTKCT= +84.    ;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
000126                  PSTKSV= +86.    ;STACK WORDS STORAGE AREA - 30 WORDS
000222                  PSVREG= +146.   ;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
000236                  PUSRPC= +158.   ;USER'S CURRENT PROGRAM COUNTER - 1 WORD
```

```
                                ;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION
                               ;(PRDIOX= .160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
                               ;(PRDIOV= .164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
                               ;(PWRIOX= .168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
                               ;(PWRIOV= .172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
                               ;(PUPRRS= .176. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
                               ;(PUPDRS= .192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
                               ;(PUBMAP= .208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
                                ;END OF MEM MGMNT ONLY ENTRIES
000240                         PTSIZE= .160.   ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT
                               ;(PTSIZE= .210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)
000242                         PTEND=  .162.   ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION
                               ;(PTEND=  .212. ;END OF PROGRAM TABLE - MEM MGMNT VERSION)
```

```
                                ;       DEVICE ROUTINE TABLE

000116                          DRTLTH= 78.     ;DEVICE ROUTINE TABLE LENGTH
                                ;
000000                          DEVRS2= +0.     ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
000002                          DEVFW0= +2.     ;DEVICE ROUTINE FLAGWORD - 1 WORD
000004                          DEVIW1= +4.     ;DEVICE INTERFACE WORD # 1 - 1 WORD
000006                          DEVIW2= +6.     ;DEVICE INTERFACE WORD # 2 - 1 WORD
000010                          DEVIW3= +8.     ;DEVICE INTERFACE WORD # 3 - 1 WORD
000012                          DEVIW4= +10.    ;DEVICE INTERFACE WORD # 4 - 1 WORD
000014                          DEVIW5= +12.    ;DEVICE INTERFACE WORD # 5 - 1 WORD
000016                          DEVIW6= +14.    ;DEVICE INTERFACE WORD # 6 - 1 WORD
000020                          DEVIW7= +16.    ;DEVICE INTERFACE WORD # 7 - 1 WORD  (SIZE)
000022                          DEVIW8= +18.    ;DEVICE INTERFACE WORD # 8 - 1 WORD  (ERR)
000024                          DEVDRA= +20.    ;DEVICE REGISTERS ADDRESS - 1 WORD
000026                          DEVIVA= +22.    ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
000030                          DEVRPS= +24.    ;DEVICE READ PROCESSOR STATUS WORD (BUS REG) - 1 WORD
000032                          DEVWPS= +26.    ;DEVICE WRITE PROC STATUS WORD (BUS REG) - 1 WORD
000034                          DHKPAD= +28.    ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
000036                          DERPAD= +30.    ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
000040                          DKILAD= +32.    ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
000042                          DECTAD= +34.    ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
000044                          DTOERD= +36.    ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
000046                          DEVIOB= +38.    ;DEVICE I/O BUSY BRANCH ADDRESS (IOBSY) - 1 WORD
000050                          DEVOER= +40.    ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
000052                          DVUPRT= +42.    ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
000054                          DVCPRT= +44.    ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
000056                          DEVBTA= +46.    ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
000060                          DVBTDA= +48.    ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASL2) - 1 WORD
```

```
000062                  DVPOTA= +50.    ;CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
000064                  DVSFWD= +52.    ;MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
000066                  DVSVEC= +54.    ;SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
000070                  DVCVEC= +56.    ;CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
000072                  DVTVEC= +58.    ;TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
000074                  DVRINT= +60.    ;RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
000076                  DVGETB= +62.    ;GET DATA BYTE BR ADR (GETBYT) - 1 WORD
000100                  DVPUTB= +64.    ;PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
000102                  DEVSTP= +66.    ;DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
000104                  DEVETP= +68.    ;DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
000106                  DVPTEP= +70.    ;PACK TABLE EXTEN. REL POINTER - 1 WORD
000110                  DVVTEP= +72.    ;VECTOR TABLE EXTEN. REL POINTER - 1 WORD
000112                  DVCTEP= +74.    ;COMPILER TBL EXTEN. REL POINTER - 1 WORD
000114                  DVIWSP= +76.    ;DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
000116                  DRTEND= +78.    ;END OF DEVICE ROUTINE TABLE


000001                          .END
```

```
ABBREV  005556R    002   DATAER  001260R    002   DVPTEP= 000106          HSKPEP= 000004          PARAM   006410R    002
ACTIVE= 100000           DECASC  007062R    002   DVPVTB= 000100          HSKPST= 001162R    002   PAREAR  001254R    002
ALARM   003446R    002   DECTAD= 000042           DVREGE= 000176R    002   INCWAC  002552R    002   PASCIN= 000006
ALMMSG  004014R    002   DERPAD= 000036           DVREGS  000116R    002   INTEXT  004620R    002   PC     =%000007
ATIMSG  007003R    002   DEVBTA= 000056           DVREND= 007522R    002   ISTAT = 001162R    002   PCURDV= 000036
AURPEP= 000010           DEVDER= 000050           DVREX   005546R    002   IVCTAD  000024R    002   PDNUMS= 000036
AUTORP= 000020           DEVDRA= 000024           DVRGDT  007102R    002   KILL    001400R    002   PDPNTR= 000034
BAR     001174R    002   DEVETP= 000104           DVRGMG  007074R    002   KILLEX  001466R    002   PDST  = 000122
BAUD    003052R    002   DEVFWD= 000002           DVRINT= 000074          LALARM  001155R    002   PFBBDV= 000002
BAUMSG  003210R    002   DEVIOB= 000046           DVSFWD= 000064          LBAUD   001155R    002   PFLGWD= 000000
BCMRD   007124R    002   DEVIVA= 000026           DVSVEC= 000066          LBITS   001155R    002   PFWRDR= 000004
BCMWR   007146R    002   DEVIM1= 000004           DVTVEC= 000072          LCOUNT  001154R    002   PLNGTH= 000026
BCR     001176R    002   DEVIM2= 000006           DVUPRT= 000052          LCRST   001154R    002   PMASK   006412R    002
BINASC  000056R    002   DEVIM3= 000010           DVVTEP= 000110          LDRFLG= 000001          PMDLCD= 000032
BITMSG  003532R    002   DEVIM4= 000012           ECHO    003570R    002   LECHO   001154R    002   PNAME = 000010
BITS    003444R    002   DEVIM5= 000014           EMSWF   006070R    002   LEVEN   001154R    002   PNBR  = 000116
BIT11 = 004000           DEVIM6= 000016           ERSHD   006054R    002   LFDPLX  001154R    002   PNMMSG  006776R    002
BIT12 = 010000           DEVIM7= 000020           ERCDTA  007514R    002   LHDPLX  001154R    002   POBJST= 000024
BIT6  = 000100           DEVIM8= 000022           ERCDTB  006012R    002   LINE    003050R    002   POPSW = 000002
BKCNT   001236R    002   DEVMPS= 000030           ERCFRM  007401R    002   LNECHO  001154R    002   PRCIOA= 000016
BREAK   002044R    002   DEVRSZ= 000000           ERCMEM  007341R    002   LNOPAR  001154R    002   PRESET  003650R    002
BRFLG = 000010           DEVSTP= 000102           ERCOVR  007361R    002   LNWAIT  001154R    002   PRINT   006574R    002
BTASLZ  000060R    002   DEVWPS= 000032           ERCPAR  007420R    002   LOADAC  001726R    002   PROGNM  007000R    002
BYCR    001172R    002   DHKPAD= 000034           ERCSOF  007460R    002   LOADTC  002306R    002   PRONER= 020000
BYRD    001222R    002   DISCNT  005746R    002   ERDIAG  005722R    002   LOC7    000000R    002   PRTEX   006706R    002
BYTES   004250R    002   DISPST  006504R    002   ERMBCT  005720R    002   LODD    001154R    002   PS    = 177776
BYWP    001226R    002   DISUIV  006452R    002   ERR     000022R    002   LPR     001166R    002   PSRC  = 000120
CALDNL  001700R    002   DISUAL  006500R    002   ERRDIS  005630R    002   LPREST  001154R    002   PSRCST= 000022
CAR     001170R    002   DISUNM  006412R    002   ERREXT  002764R    002   LPRUPD  004276R    002   PSSAVE  004232R    002
CHKLVL  005132R    002   DISUPR  006472R    002   ERRFLG  004250R    002   LRBAUD  001155R    002   PSTKCT= 000124
CIOBSY  000046R    002   DKILAD= 000040           ERRMSG  007322R    002   LSADDR  004236R    002   PSTKSV= 000126
CLIST   000054R    002   DOERCK= 000400           ERROPT  005620R    002   LSETUP  001157R    002   PSVREG= 000222
CLRERR  002166R    002   DOIOT = 000040           ERRSMR  005730R    002   LSTATS  001154R    002   PTEM0 = 000056
CLRVCT= 000002           DRCVTL  001772R    002   ERRTST  002722R    002   LSTP1   001154R    002   PTEM1 = 000060
CLRVEC  000070R    002   DREGAD  000024R    002   ERSTOP= 000004          LSTP2   001154R    002   PTEM10= 000102
CLWVCT= 000004           DRTEND= 000116           ERTBEN  005712R    002   LTBAUD  001155R    002   PTEM11= 000104
CMDCBK  007230R    002   DRTLTH= 000116           EVEN    003342R    002   LWAIT   001154R    002   PTEM12= 000106
CMDCMS  007254R    002   DRWAIT= 100000           FDUPLX  003634R    002   MISCNT  001240R    002   PTEM13= 000110
CMDCRD  007201R    002   DSNDTL  002470R    002   FLAG    001264R    002   MMVER = 000001          PTEM14= 000112
CMDCWR  007214R    002   DTDEAD= 000044           FLAGWD  000002R    002   MSADDR  004234R    002   PTEM15= 000114
CNTSEN= 007522R    002   DVBTDA= 000060           FRAMER  001252R    002   NEMERR  001246R    002   PTEM2 = 000062
CNTSMG  007110R    002   DVCMDS  000176R    002   FRCVTL  001750R    002   NOCOMP= 000001          PTEM3 = 000064
COMBER  005174R    002   DVCPRT= 000054           FRSBYT  004710R    002   NOECHO  003604R    002   PTEM4 = 000066
CRESET  003446R    002   DVCPTE  000734R    002   FSNDTB  002374R    002   NOFERR  005020R    002   PTEM5 = 000070
CRLF    007162R    002   DVCTEP= 000112           FSNDTL  002404R    002   NODERR  005040R    002   PTEM6 = 000072
CSTAT   001202R    002   DVCVEC= 000070           GETBYT  000076R    002   NOPAR   003376R    002   PTEM7 = 000074
CSYSFW  000064R    002   DVGETB= 000076           GTBYTE  004704R    002   NOPERR  005000R    002   PTEM8 = 000076
CTPRIO= 000020           DVIWSP= 000114           GTNXTD= 001000          NOWAIT  002776R    002   PTEM9 = 000100
CUPGER  000050R    002   DVIWST  001152R    002   GTPTBS  001606R    002   NRC     001164R    002   PTEND = 000242
CURMSG  007017R    002   DVMVTE  000610R    002   [illegible]SCAD  004612R    002   OCPRES= 000100          PTLGTH= 002242
CWPINT  001436R    002   DVNMLP  002260R    002   [illegible]PLX  003620R    002   ODD     003360R    002   PTOCNT= 000030
CYCDVL= 002000           DVPDTA= 000062           [illegible]EEP  005260R    002   ONESTP  003412R    002   PTSIZE= 000240
CYCPRG= 040000           DVPKTE  000340R    002   HSKPEN= 001266R    002   OVRRUN  001250R    002   PUSRPC= 000236
```

```
PUTBYT  000100R    002   RERVEC  005204R    002   SETDED= 000040         STSTAT  006230R    002   UNITMG  007046R    002
PWRIOA= 000020           RESREG  006172R    002   SETRSE  002002R    002   SUPTAD  006210R    002   URSTOP= 000002
PSCONS= 120000           RETJAN  072620R    002   SETSCN  002256R    002   TBAUD   003076R    002   USEURP= 000200
RBAUD   003066R    002   RICNT   001242R    002   SETTCR  002504R    002   TBAUSB  003240R    002   USMTPS= 010002
RBAUSB  003106R    002   RPTBAS  005512R    002   SETUP   003042R    002   TEMTCR  004252R    002   WAIT    002522R    002
RBUSRQ  000030R    002   RPTEND  005536R    002   SETVEC  000066R    002   TESTRC  001706R    002   WBUSRQ  000032R    002
RCR     004256R    002   RPTLP   005474R    002   SIZE    000020R    002   TESTTC  002266R    002   WDOON   004414R    002
RCVTBL  004252R    002   RRINTV  005216R    002   SNDTBL  004072R    002   TOECNT  001262R    002   WICNT   001244R    002
RDBSY = 020000           RTERM   005062R    002   SOFERR  001256R    002   TOEMSG  001354R    002   WRBCNT  004630R    002
RDCNT   001232R    002   RTNINT  000074R    002   SP     =%000006          TOUTER  001266R    002   WRBSY = 040000
RDINMS  007272R    002   RTRMLP  005106R    002   SPOPER= 000200         TOUTEX  001376R    002   WRCNT   001234R    002
RDINT   004632R    002   RWINTV  005234R    002   SRDBSY  001542R    002   TRMTST  002642R    002   WRINMS  007312R    002
RDMNCL  001640R    002   R0     =%000000          SSR     001200R    002   TRVECT  006710R    002   WRINT   004352R    002
RDMNST  001644R    002   R1     =%000001          STCR    004244R    002   TRVEXT  006736R    002   WRITE   002034R    002
RDNOWT  002024R    002   R2     =%000002          STMMSG  006140R    002   TSTIEB  002634R    002   WRMNCL  002220R    002
READ    001500R    002   R3     =%000003          STMNT   006052R    002   TSTVEC  000072R    002   WRMNST  002224R    002
RELEAS  002770R    002   R4     =%000004          STMNUM  006150R    002   TWOSTP  003426R    002   WTERM   004536R    002
REMOFL  004670R    002   R5     =%000005          STONER= 100000         TWVECT  006740R    002   WT4IOT= 000010
REMMSG  007031R    002   SAVREG  006156R    002   STOPRD  005142R    002   TWVEXT  006774R    002   XMIT    002352R    002
REPORT  005314R    002   SCR     001162R    002   STOPWR  004560R    002   ULIST   000052R    002   XXXX  = 000000
REPTBL  005560R    002   SETBSY  002114R    002   STRSCN  005120R    002   UNASCI  007066R    002   .     = 007522R    002

. ABS.  000000     000
        000000     001
DH11    007522     002

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DTDHAB/NL:TOC/DOC=DTDHAB.P11
 RUN-TIME: 4 10 1 SECONDS
 RUN-TIME RATIO: 40/16=2.4
 CORE USED:  5K  (9 PAGES)

 DOCUMENT PAGES:  39
```