```
                /TD8E Dectape DUMP Program
                /
                / This program will send a Dectape image out the console port.
                / The format of the data sent is 0xff (0377) or 0xfd if read error
                / followed by 128 word  of data for each block.
                / After the last block a 0xfe (0376) is sent
                / with a two byte checksum, low 8 bits first then upper 4.
                / The words in a block are sent as three bytes for each 2 words.
                / Like this: (WvdM)
                /   +--------------------+
                /   ! byte1     ! byte2h !
                /   +--------------------+
                /   ! byte2l !  byte 3   !
                /   +--------------------+
                /
                /   1 = low 8 bits first word (was like this WvdM)
                /   2 = upper 4 bits first and lower 4 bits second
                /   3 = upper 8 bits second word
                /
                / The program (PC) receiving the data should be started before this program
                /
                / To run start at 0200.
                /    SR 11 should be drive, only 0 and 1 supported without reassembling
                /    SR 6-8 should be maximum memory field in computer, needs 8k minimum
                / The receiving program should be running first.
                / At normal exit hitting cont will restart the program
                /
                / Should halt at label finish (140) with number of recoverable errors in AC
                / The current block being read will be displayed in the AC
                / while running.
                /
                / If a unrecoverable error occurs the program will halt with the error in
                / the AC.  Hit continue to dump more or comment out hlt, search for *****.
                / The PC program will print out the bad location if an error occurs
                /
                / We will retry each read up to 16 times on error
                /
                / This transfers the standard 129 word by 1474 blocks used by OS/8 etc.
                / Other formats can be handled by changing constants below

0030            INAD=030                / Address of serial input, 30 for console
6030            KCF2=6000 INAD
6031            KSF2=6001 INAD
6032            KCC2=6002 INAD
6034            KRS2=6004 INAD
6035            KIE2=6005 INAD
6036            KRB2=6006 INAD

0040            OUTAD=040               / Address of serial output, 40 for console
6040            TFL2=6000 OUTAD
6041            TSF2=6001 OUTAD
6042            TCF2=6002 OUTAD
6044            TPC2=6004 OUTAD
6045            TSK2=6005 OUTAD
6046            TLS2=6006 OUTAD
```

```
        /2 TD8E INITIALIZER PROGRAM, V7A
        /
        /COPYRIGHT (C) 1975, 1977
        /DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        /
        /
        /ABSTRACT--
        /       THE ROUTINE DESCRIBED AND LISTED HERE IS A GENERAL
        /DATA HANDLER FOR THE TD8E DECTAPE SYSTEM. THE ROUTINE
        /CONTAINS SEARCH, READ, AND WRITE FUNCTIONS IN A FORMAT
        /WHICH IS COMPATIBLE WITH OS/8 DEVICE HANDLER CALLING
        /SEQUENCES.

        /THIS ROUTINE CAN BE RE-EDITED AND ASSEMBLED TO PRODUCE
        /VARIATIONS ON THE BASIC TD8E SYSTEM. ASSEMBLY PARAMETERS
        /CONTROL:
        /A) WHAT DRIVES (UNITS 0-7) WILL BE USED
        /B) THE ORIGIN OF THE TWO PAGE ROUTINE
        /C) WHAT MEMORY FIELD THE ROUTINE WILL RUN IN
        /D) THE SIZE OF THE DECTAPE BLOCK TO BE READ/WRITTEN

        /FOLLOWING ARE THE PARAMETERS SET UP FOR THE STANDARD
        /DEC VERSION OF THIS ROUTINE:

0010            DRIVE=10        /UNITS 0 AND 1 SELECTED
0600            ORIGIN=600      /ENTER AT ORIGIN, ORIGIN+4
0000            AFIELD=0        /INITIAL FIELD SETTING
0000            MFIELD=00       /AFIELD*10=MFIELD
0201            WDSBLK=201      /129 WORDS PER BLOCK

        /THE USE OF THE PARAMETERS IS AS FOLLOWS:

        / DRIVE: DRIVE DETERMINES WHICH UNITS WILL BE SELECTED
        /       DRIVE=10 IMPLIES UNITS 0 &1
        /       DRIVE=20 IMPLIES UNITS 2&3
        /       DRIVE=30 IMPLIES UNITS 4&5
        /       DRIVE=40 IMPLIES UNITS 6&7

        /ORIGIN: ALTERING ORIGIN CAUSES ASSEMBLY IN A DIFFERENT
        /       MEMORY LOCATION. WHEN CHANGING ORIGIN KEEP IN MIND
        /THAT THIS IS A TWO PAGE ROUTINE.

        /AFIELD: AFIELD DETERMINES THE INITIAL FIELD SETTING FOR THE
        /       LOADER. PERMISSIBLE VALUES FOR AFIELD ARE 0 TO 7.

        /MFIELD: MFIELD IS USED IN A CIF CDF MFIELD INSTRUCTION.
        /       THE VALUE INSERTED FOR MFIELD SHOULD BE 10(8) TIMES
        /       THE VALUE FOR AFIELD. THE PERMISSIBLE VALUES ARE 00-70.

        /WDSBLK: WDSBLK GOVERNS HOW MANY WORDS THE ROUTINE THINKS ARE
        /       IN A DECTAPE BLOCK. THE STANDARD VALUE IS 201(8) OR
        /       129 DECIMAL. NOTE THAT THE FUNCTION WORD BIT 10 CAN
        /       BE USED TO SUBTRACT ONE FROM WDSBLK. THE VALUE USED
        /       FOR WDSBLK SHOULD BE THE NUMBER OF WORDS THE TAPE WAS
```

```
/       FORMATTED TO CONTAIN.

/IF WE WANT A HANDLER FOR UNITS 2&3 TO RESIDE IN
/FIELD 2 AT LOCATION 3000 AND READ/WRITE 256(10) WORDS
/PER BLOCK, THE PARAMETERS WOULD BE:
/       DRIVE=20
/       ORIGIN=3000
/       AFIELD=2
/       MFIELD=20
/       WDSBLK=400
/
/THE CALL TO THE SUBROUTINE FOLLOWS BASICALLY THE
/CALLING SEQUENCE FOR OS/8 DEVICE HANDLERS.
/THE CALLING SEQUENCE IS:

/       CDF CURRENT
/       CIF MFIELD      /MFIELD=FIELD ASSEMBLED IN
/       JMS ENTRY       /AC ON ENTRY=4000 IS UNIT 1
/       ARG1
/       ARG2
/       ARG3
/       ARG4
/       ERROR RETURN
/       NORMAL RETURN

/THE ARGUMENTS ARE:

/ARG1: FUNCTION WORD    BIT0: 0=READ, 1=WRITE
/                       BITS 1-5: UNUSED, WAS # BLOCKS IN OPERATION
/                       BITS 6-8: FIELD OF BUFFER AREA
/                       BIT 9: UNUSED
/                       BIT 10: # OF WORDS/BLOCK.
/                       0= WDSBLK, 1=WDSBLK-1
/                       BIT 11: 1=START FORWARD, 0=REVERSE
/ARG2: # OF BLOCKS IN OPERATION
/ARG3: BUFFER ADDRESS FOR OPERATION
/ARG4: STARTING BLOCK FOR OPERATION

/ERRORS: THE HANDLER DETECTS TWO TYPES OF ERRORS:
/A) FATAL ERRORS- PARITY ERROR, TIMING ERROR,
/               TOO GREAT A BLOCK NUMBER
/       FATAL ERRORS TAKE ERROR RETURN WITH THE
/       AC=4000.
/B) NON-FATAL- SELECT ERROR.
/       IF NO PROPER UNIT IS SELECTED, THE ERROR
/       RETURN IS TAKEN WITH CLEAR AC.
/FATAL ERRORS TRY THREE TIMES BEFORE TAKING ERROR RETURN.
/THE NORMAL RETURN IS TAKEN AFTER ALL INDICATED
/BLOCKS HAVE BEEN TRANSFERRED. THE AC IS CLEAR.

/THE TD8E IOT'S ARE:
6771            SDSS=7001-DRIVE /SKIP ON SINGLE LINE FLAG
6772            SDST=7002-DRIVE /SKIP ON TIMING ERROR
6773            SDSQ=7003-DRIVE /SKIP ON QUAD LINE FLAG
6774            SDLC=7004-DRIVE /LOAD COMMAND REGISTER
```

```
       6775          SDLD=7005-DRIVE /LOAD DATA REGISTER
       6776          SDRC=7006-DRIVE /READ COMMAND REGISTER
       6777          SDRD=7007-DRIVE /READ DATA REGISTER

             /THE IOT'S IN GENERAL ARE 677X,676X,675X,AND 674X.
             /THE OTHERS CONTROL UNITS 2-7.

             /       THIS HANDLER USES DECTAPE BLOCKS NOT OS/8 BLOCKS !

       0600          *ORIGIN

             /       MODIFIED SO BIT 0 ON ENTRY IS UNIT 1
00600  0000  DTA0,   0
00601  3146          DCA UNIT        /SAVE UNIT POSITION
00602  6214          RDF
00603  1132          TAD C6203       /GET DATA FIELD AND SETUP RETURN
00604  3352          DCA LEAVE
00605  1600          TAD I DTA0      /GET FUNCTION WORD
00606  6775          SDLD            /PUT FUNCTION INTO DATA REGISTER
00607  7112          CLL RTR         /AC STILL HAS FUNCTION. PUT # WORDS PER
                                     /BLOCK INTO LINK
00610  7630          SZL CLA         /KNOCK ONE OFF WDSBLK?
00611  7001          IAC             /YES
00612  1102          TAD MWORDS
00613  3142          DCA WCT         /STORE MASTER WORD COUNT
00614  2200          ISZ DTA0        /TO BLOCK COUNT
00615  1600          TAD I DTA0
00616  7041          CIA
00617  3362          DCA PGCT
00620  2200          ISZ DTA0        /TO BUFFER
00621  1600          TAD I DTA0
00622  3137          DCA BUFF        /SAVE ADDRESS
00623  2200          ISZ DTA0        /TO BLOCK NUMBER
00624  1600          TAD I DTA0
00625  3141          DCA BLOCK       /INITIAL BLOCK
00626  2200          ISZ DTA0        /POINT TO ERROR EXIT
00627  6203          CIF CDF MFIELD  /TO ROUTINES DATA FIELD
00630  6777          SDRD
00631  0122          AND C70         /GET FIELD FOR XFER
00632  1131          TAD C6201       /FORM CDF N
00633  3251          DCA XFIELD      /IF=0 AND DF=N AT XFER.
00634  1146          TAD UNIT        /TEST FOR SELECT ERROR
00635  6774          SDLC
00636  7200          CLA             /WAIT FOR SLOW SELECT
00637  1100          TAD RETRY
00640  3363          DCA TRYCNT      /3 ERROR TRIES
00641  6776          SDRC
00642  0123          AND C100
00643  7640          SZA CLA
00644  5345          JMP FATAL-1
00645  6777          SDRD            /PUT FUNCT INTO PG 0
00646  3143          DCA FUNCT
00647  6777          SDRD            /GET INITIAL MOTION BIT TO LINK
00650  7110          CLL RAR
00651  7402  XFIELD, HLT             /SET DATA FIELD FOR NEXT PAGE
```

```
00652  5264          JMP GO          /AND START THE MOTION.

00653  6772  RWCOM,  SDST            /ANY CHECKSUM ERRORS?
00654  7440          SZA             /OR CHECKSUM ERRORS?
00655  5334          JMP TRY3        /PLEASE NOTE THAT THE LINK IS ALWAYS
                                     /SET AT RWCOM. GETCHK SETS IT.
00656  3147          DCA CHKFTL      /CLEAR LAST CHECKSUM ERROR
00657  2362          ISZ PGCT        /ALL REQUESTED PAGES DONE?
00660  7410          SKP             / (DJG)
00661  5344          JMP EXIT        /ALL DONE. GET OUT
00662  2141          ISZ BLOCK       /NEXT BLOCK TO XFER
00663  7120          CLL CML         /FORCES MOTION FORWARD
00664  7232  GO,     CLA CML RTR     /LINK BECOMES MOTION BIT
00665  1127          TAD C1000
00666  1146          TAD UNIT        /PUT IN 'GO' AND UNIT #
00667  6774          SDLC            /LOOK FOR BLOCK NO.
00670  7200          CLA
00671  1137          TAD BUFF
00672  3136          DCA OLDBUF      /SAVE BUFFER ADDR FOR ERROR RETRY
00673  6214          RDF
00674  1131          TAD C6201
00675  3336          DCA OLDFLD
00676  4354          JMS RDQUAD      /WAIT AT LEAST 6 LINES TO LOOK
00677  4354          JMS RDQUAD
00700  6771  SRCH,   SDSS
00701  5300          JMP .-1         /WAIT FOR SINGLE LINE FLAG
00702  6776          SDRC
00703  7106          CLL RTL         /DIRECTION TO LINK. INFO BITS ARE SHIFTED.
00704  0140          AND C374        /ISOLATE MARK TRACK BITS
00705  1323          TAD M110        /IS IT END ZONE?
00706  7450          SNA             /THE LINK STAYS SAME THRU THIS
00707  5325          JMP ENDZ
00710  1134          TAD M20         /CHECK FOR BLOCK MARK
00711  7640          SZA CLA
00712  5300          JMP SRCH
00713  6777          SDRD            /GET THE BLOCK NUMBER
00714  7430          SZL             /IF WE ARE IN REVERSE, LOOK FOR 10
                                     /BLOCKS BEFORE TARGET BLOCK. THIS
                                     /ALLOWS TURNAROUND AND UP TO SPEED.
00715  1120          TAD C10         /REVERSE
00716  7040          CMA
00717  1141          TAD BLOCK
00720  7040          CMA             /IS IT RIGHT BLOCK?
00721  7450          SNA
00722  5331          JMP FOUND       /YES..HOORAY!
00723  7670  M110,   SZL SNA CLA     /NO, BUT ARE WE HEADED FOR IT?
00724  5300          JMP SRCH        /YES
00725  6776  ENDZ,   SDRC            /WE ARE IN THE END ZONE
00726  7106          CLL RTL         /DIRECTION TO LINK
00727  7200          CLA             /ARE WE IN REVERSE?
00730  5264          JMP GO          /YES..TURN US AROUND

00731  7630  FOUND,  SZL CLA         /RIGHT BLOCK. HOW ABOUT DIRECTION?
00732  5264          JMP GO          /WRONG..TURN AROUND
00733  5761          JMP I CXGO      /OK GO READ/WRITE
```

```
00734  7002  TRY3,   BSW
00735  3147          DCA CHKFTL      /KEEP LAST CHECKSUM ERROR
00736  7000  OLDFLD, NOP
00737  1136          TAD OLDBUF
00740  3137          DCA BUFF
00741  2363          ISZ TRYCNT
00742  5264          JMP GO          /TRY 3 TIMES
00743  5346          JMP FATAL       /LINK OFF MEANS AC=4000 ON RETURN
00744  2200  EXIT,   ISZ DTA0
00745  7120          CLL CML         /AC=0 ON NORMAL RETURN
00746  1146  FATAL,  TAD UNIT
00747  6774          SDLC            /STOP THE UNIT
00750  7230          CLA CML RAR
00751  1147          TAD CHKFTL
00752  7402  LEAVE,  HLT             /GETS CIF CDF RETURN
00753  5600          JMP I DTA0

00754  0000  RDQUAD, 0               /READ A 12 BIT WORD
00755  6773          SDSQ
00756  5355          JMP .-1
00757  6777          SDRD            /READ DATA
00760  5754          JMP I RDQUAD

00761  1000  CXGO,   GOON            /HERE WE REALLY GO
00762  0000  PGCT,   0               /TOTAL PAGES TO TRANSFER
00763  7775  TRYCNT, -3              /COUNTER FOR TRIES

       1000         *ORIGIN+200
01000  6202  GOON,   CIF MFIELD
01001  6776          SDRC
01002  6774          SDLC
01003  6771  REVGRD, SDSS
01004  5203          JMP .-1         /LOOK FOR REVERSE GUARD
01005  6776          SDRC
01006  0263          AND K77
01007  1135          TAD M32         /IS IT REVERSE GUARD?
01010  7640          SZA CLA
01011  5203          JMP REVGRD      /NO.KEEP LOOKING
01012  1142          TAD WCT
01013  3374          DCA WORDS       /WORD COUNTER
01014  1143          TAD FUNCT       /GET FUNCTION  READ OR WRITE
01015  7700  K7700,  SMA CLA
01016  5264          JMP READ        /NEG. IS WRITE
01017  6776  WRITE,  SDRC
01020  0125          AND C300        /CHECK FOR WRITE LOCK AND SELECT ERROR
01021  7120          CLL CML         /LOCK OUT AND SELECT ARE AC 0 ERRORS
01022  7640          SZA CLA
01023  5772          JMP I CFATAL    /FATAL ERROR. LINK MUST BE ON
01024  4771          JMS I XQUAD     /NO ONE EVER USES THIS WORD!
01025  7600  C7600,  7600
01026  1130          TAD C1400
01027  1146          TAD UNIT        /INITIATE WRITE MODE
01030  6774          SDLC
01031  7240          CLA CMA
```

```
01032  4331          JMS WRQUAD      /PUT 77 IN REVERSE CHECKSUM
01033  7240          CLA CMA
01034  3145          DCA CHKSUM
01035  1537  WRLP,   TAD I BUFF      /GLORY BE! THE ACTUAL WRITE!
01036  4331          JMS WRQUAD
01037  2137          ISZ BUFF        /BUMP CORE POINTER
01040  1137          TAD BUFF
01041  1124          TAD C200
01042  7640          SZA CLA         /ARE WE AT OS/8 BOUNDARY?
01043  5251          JMP STFLD1+1    /NOT AT END OF FIELD (DJG)
01044  3137          DCA BUFF        /OK, TO NEXT FIELD
01045  6214          RDF
01046  1133          TAD C6211
01047  3250          DCA STFLD1
01050  7000  STFLD1, NOP
01051  2374          ISZ WORDS       /DONE THIS BLOCK?
01052  5235          JMP WRLP        /NOT YET..LOOP A WHILE
01053  1143          TAD FUNCT       /IS THE OPERATION FOR WDSBLK PER BLOCK?
01054  7112          CLL RTR         /IF NO, WRITE A 0 WORD
01055  7630          SZL CLA
01056  4331          JMS WRQUAD      /WRITE A WORD OF 0
01057  4355          JMS GETCHK      /DO THE CHECK SUM
01060  4331          JMS WRQUAD      /WRITE FORWARD CHECKSUM
01061  4331          JMS WRQUAD      /ALLOW CHECKSUM TO BE WRITTEN
01062  5773          JMP I CRWCOM
01063  0077  K77,    77              /ABOVE MAY SKIP (NOT ANYMORE DJG)
01064  4771  READ,   JMS I XQUAD
01065  4771          JMS I XQUAD
01066  4771          JMS I XQUAD      /SKIP CONTROL WORDS
01067  0263          AND K77
01070  1215          TAD K7700       /TACK 7700 ONTO CHECKSUM.
01071  3145          DCA CHKSUM      /CHECKSUM ONLY LOW 6 BITS ANYWAY
01072  4771  RDLP,   JMS I XQUAD
01073  4340          JMS EQUFUN      /COMPUT CHECKSUM AS WE GO
01074  3537          DCA I BUFF      /IT GETS CONDENSED LATER
01075  1537          TAD I BUFF      /CHECK FOR MEMORY ERRORS
01076  1144          TAD EQUTMP
01077  7001          IAC
01100  7640          SZA CLA
01101  5176          JMP MEMERR
01102  2137          ISZ BUFF        /AT END OF FIELD?
01103  1137          TAD BUFF
01104  1124          TAD C200
01105  7640          SZA CLA         /ARE WE AT OS/8 BOUNDARY?
01106  5314          JMP STFLD2+1    /NOT AT END OF FIELD (DJG)
01107  3137          DCA BUFF        /OK, TO NEXT FIELD
01110  6214          RDF
01111  1133          TAD C6211
01112  3313          DCA STFLD2
01113  7000  STFLD2, NOP
01114  2374          ISZ WORDS       /DONE THIS OP?
01115  5272          JMP RDLP        /NO SUCH LUCK
01116  1143          TAD FUNCT       /IF OP WAS FOR WDSBLK-1, READ AND
01117  7112          CLL RTR         /CHECKSUM THE LAST TAPE WORD
01120  7620          SNL CLA
```

```
01121  5324          JMP RDLP2
01122  4771          JMS I XQUAD     /NOT NEEDED FOR WDSBLK/BLOCK
01123  4340          JMS EQUFUN      /CHECKSUM IT
01124  4771  RDLP2,  JMS I XQUAD     /READ CHECKSUM
01125  0215          AND K7700
01126  4340          JMS EQUFUN
01127  4355          JMS GETCHK      /GET SIX BIT CHECKSUM
01130  5773          JMP I CRWCOM

01131  0000  WRQUAD, 0               /WRITE OUT A 12 BIT WORD
01132  4340          JMS EQUFUN      /ADD THIS TO CHECKSUM
01133  6773          SDSQ            /SKIP ON QUADLINE FLAG
01134  5333          JMP .-1
01135  6775          SDLD            /LOAD DATA  ONTO BUS
01136  7200          CLA             /SDLD DOESN'T CLEAR AC
01137  5731          JMP I WRQUAD

01140  0000  EQUFUN, 0               /COMPUTE EQUIVALENCE CHECKSUM
01141  7040          CMA
01142  3144          DCA EQUTMP      /ACTUALLY CHECKSUMS ON DECTAPE ARE
01143  1144          TAD EQUTMP      /EQUIVALENCE OF ALL WORDS IN A RECORD
01144  0145          AND CHKSUM      /SIX BITS AT A TIME. BUT SINCE EQUIVALENCE
01145  7041          CIA             /IS ASSOCIATIVE, WE CAN DO IT 12
01146  7104          CLL RAL         /BITS AT A TIME AND CONDENSE LATER.
01147  1144          TAD EQUTMP      /THIS ROUTINE USES THESE IDENTITIES:
01150  1145          TAD CHKSUM      /A+B=(A.XOR.B)+2*(A.AND.B)
01151  3145          DCA CHKSUM      /A.EQU.B=.NOT.(A.XOR.B)=A.XOR.(.NOT.B)
01152  1144          TAD EQUTMP      /A.EQU.B=(A+(.NOT.B))-2*(A.AND.(.NOT.B))
01153  7040          CMA
01154  5740          JMP I EQUFUN

01155  0000  GETCHK, 0               /FORM 6 BIT CHECKSUM
01156  7200          CLA
01157  1145          TAD CHKSUM
01160  7040          CMA
01161  7106          CLL RTL
01162  7006          RTL
01163  7006          RTL
01164  4340          JMS EQUFUN
01165  7320          CLA CLL CML     /FORCES LINK ON AT RWCOM
01166  1145          TAD CHKSUM
01167  0215          AND K7700
01170  5755          JMP I GETCHK

01171  0754  XQUAD,  RDQUAD
01172  0746  CFATAL, FATAL
01173  0653  CRWCOM, RWCOM           /BLOCK DONE, BACK TO FIRST PAGE
01174  0000  WORDS,  0

       0100          *100
00100  7775  RETRY,  7775            /RETRY UP TO 3 TIMES
00101  2702  NUMBLK, 2702            /TOTAL NUMBER OF BLOCKS ON DECTAPE
00102  7577  MWORDS, -WDSBLK         /WORDS PER BLOCK FOR SETUP
00103  0036  BLKFLD, 36              /30 129 WORD BLOCKS PRESERVES OS8
                                     /WRAPPING PAST END OF LAST FIELD DOESN'T WORK
```

```
00104  0000  DRVSEL, 0              /SELECTED DRIVE
00105  0000  FIELDS, 0              /MINUS NUMBER OF FIELDS TO USE
00106  0000  ERRCN2, 0              /COUNTING ERRORS
00107  0000  RDSIZE, 0              /NUMBER BLOCKS PER READ
00110  0000  CBLOCK, 0              /CURRENT DECTAPE BLOCK TO XFER
00111  0000  CHKSM,  0
00112  0000  READST, 0
00113  0000  LOC,    0
00114  0000  LEN,    0
00115  0000  BCNT,   0              /BLOCKS TO SEND TO PC
00116  0000  REASON, 0
00117  0000  TEMP,   0

00120  0010  C10,    10
00121  0017  C17,    17
00122  0070  C70,    70
00123  0100  C100,   100
00124  0200  C200,   200
00125  0300  C300,   300
00126  0360  C360,   360
00127  1000  C1000,  1000
00130  1400  C1400,  1400
00131  6201  C6201,  6201
00132  6203  C6203,  6203
00133  6211  C6211,  6211
00134  7760  M20,    -20
00135  7746  M32,   -32

00136  0000  OLDBUF, 0              /USED BY DTA0 ROUTINE - RETRY BUFFER
00137  0000  BUFF,   0              /USED BY DTA0 ROUTINE
00140  0374  C374,   374            /USED BY DTA0 ROUTINE
00141  0000  BLOCK,  0              /USED BY DTA0 ROUTINE
00142  0000  WCT,    0              /USED BY DTA0 ROUTINE - MASTER WORDCOUNT
00143  0000  FUNCT,  0              /USED BY DTA0 ROUTINE - FUNCTION
00144  0000  EQUTMP, 0              /USED BY DTA0 ROUTINE
00145  0000  CHKSUM, 0              /USED BY DTA0 ROUTINE
00146  0000  UNIT,   0              /USED BY DTA0 ROUTINE
00147  0000  CHKFTL, 0              /USED BY DTA0 ROUTINE


       0176          *176
00176  6214  MEMERR, RDF            /SIGNAL MEMORY ERROR!!!
00177  7402  FINISH, HLT            /NORMAL GOOD HALT OR SELERR
00200  6201  START,  CDF 0
00201  6007          CAF
00202  7704          CLA CLL OSR    /GET DRIVE
00203  0377          AND (1
00204  7012          RTR
00205  3104          DCA DRVSEL
00206  7704          CLA CLL OSR    /GET MAX FIELD TO USE
00207  7012          RTR
00210  7010          RAR
00211  0376          AND (7
00212  7450          SNA
00213  7402          HLT            /MUST HAVE AT LEAST 1 FIELD FOR BUFFER
```

```
00214  7041          CIA
00215  3105          DCA FIELDS
00216  3106          DCA ERRCN2
00217  1103  RDSZLP, TAD BLKFLD      /MULTIPLY BY NUMER OF FIELDS AVAILABLE
00220  2105          ISZ FIELDS
00221  5217          JMP RDSZLP
00222  3107          DCA RDSIZE      /TOTAL NUMBER OF BLOCKS IN ONE GO = BATCH
00223  3110          DCA CBLOCK      /START WITH BLOCK 0
00224  3111          DCA CHKSM

00225  7200  DUMPLP, CLA
00226  1107          TAD RDSIZE
00227  1110          TAD CBLOCK      /(TOTAL - BATCH) - CURRENT
00230  7041          CIA
00231  1101          TAD NUMBLK      /MORE BLOCKS LEFT THAN BATCH?
00232  7500          SMA             /NO, NUMBER OF BLOCKS LEFT
00233  7200          CLA             /YES, ONLY BATCH SIZE
00234  1107          TAD RDSIZE
00235  7450          SNA             /ANY MORE BLOCKS?
00236  5267          JMP DONE        /NO, DO FINISH STUFF
00237  3245          DCA ARGSZ       /TO NUMBER OF BLOCKS REQUEST
00240  1110          TAD CBLOCK      /TO CURRENT BLOCK TO START WITH
00241  3247          DCA ARGBK
00242  1104          TAD DRVSEL
00243  4775          JMS I (DTA0
00244  0010          0010            /READ STARTING IN FIELD 1
00245  0000  ARGSZ,  0
00246  0000          0
00247  0000  ARGBK,  0
00250  5316          JMP ERRRET      /SOME DUST ON THE TAPE?
00251  1374          TAD (377        /ALL BLOCKS GOOD
00252  3112          DCA READST
                                     /Send data, each block starts with FF
00253  7300          CLA CLL         / then 2 12 bit words in 3 bytes
00254  3113          DCA LOC         / ERRRET DUPLICATES SOME OF THIS
00255  1245          TAD ARGSZ
00256  7041          CIA
00257  3115          DCA BCNT        /SET UP COUNTER OF NUMBER OF BLOCKS XFERRED
00260  6211          CDF 10          /START IN FIELD 1
00261  4773  OUTBL1, JMS I (OUTBLK      /SEND A BLOCK
00262  2110          ISZ CBLOCK      /NEXT BLOCK
00263  2115          ISZ BCNT        /BATCH XFERRED?
00264  5261          JMP OUTBL1      /NO
00265  6201          CDF 0
00266  5225          JMP DUMPLP      /GO READ NEXT BATCH


00267  7200  DONE,   CLA             / Send FE and -checksum of all words
00270  1372          TAD (376
00271  4771          JMS I (PUN
00272  1111          TAD CHKSM       / Send checksum in two bytes, low bits first
00273  7041          CIA
00274  4771          JMS I (PUN
00275  1111          TAD CHKSM
00276  7041          CIA
```

```
00277  7012          RTR
00300  7012          RTR
00301  7012          RTR
00302  7012          RTR
00303  0121          AND C17
00304  4771          JMS I (PUN
00305  1104          TAD DRVSEL
00306  4775          JMS I (DTA0     / REWIND TAPE
00307  0010          0010
00310  0001          1
00311  0000          0
00312  0000          0
00313  7000          NOP
00314  1106          TAD ERRCN2      / Leave AC with # of errors
00315  5177          JMP FINISH

00316  7440  ERRRET, SZA
00317  5322          JMP NOSEL
00320  7240          CLA CMA
00321  5177          JMP FINISH      /SELECT ERROR WITH AC = 7777
00322  3116  NOSEL,  DCA REASON      /ERROR CODE
00323  6211          CDF 10          /SEND GOOD BLOCKS READ WITH GOOD BLOCK FLAG
00324  3113          DCA LOC
00325  1110          TAD CBLOCK
00326  7041          CIA
00327  1141          TAD BLOCK       /Get - number good blocks read
00330  7041          CIA             /Last was bad
00331  7450          SNA
00332  5342          JMP FSTBAD      /First block is bad, no good to send
00333  3115          DCA BCNT
00334  1374          TAD (377
00335  3112          DCA READST
00336  4773  OUTBL2, JMS I (OUTBLK   /Send good blocks
00337  2110          ISZ CBLOCK
00340  2115          ISZ BCNT
00341  5336          JMP OUTBL2
00342  1116  FSTBAD, TAD REASON
00343  7004          RAL
00344  7640          SZA CLA         /WAS IT CHECKSUM ERROR?
00345  7240          CLA CMA         /SEND 374 FOR CHECKSUM ERROR
00346  1370          TAD (375        /NOW SEND BAD BLOCK (TIMING ERROR)
00347  3112          DCA READST
00350  4773          JMS I (OUTBLK
00351  2110          ISZ CBLOCK
00352  2106          ISZ ERRCN2
00353  6201          CDF 0
00354  5225          JMP DUMPLP      /And read from here on

00370  0375
00371  0476
00372  0376
00373  0400
00374  0377
00375  0600
00376  0007
```

```
00377  0001
       0400          PAGE
00400  0000  OUTBLK, 0               /Send a block of data out serial port
00401  7200          CLA
00402  1142          TAD WCT
00403  3114          DCA LEN         /COUNT FOR ONE BLOCK
00404  1112          TAD READST      /Send good/bad flag
00405  4276          JMS PUN
00406  7300  OUT,    CLA CLL
00407  1513          TAD I LOC
00410  1111          TAD CHKSM       / Keep checksum of all words sent
00411  3111          DCA CHKSM
00412  1513          TAD I LOC       / Send 2 words as 3 bytes
00413  7112          CLL RTR         / WvdM: Left 8 Bits first
00414  7012          RTR             / WvdM:
00415  4276          JMS PUN
00416  7300          CLA CLL
00417  1513          TAD I LOC
00420  0121          AND C17
00421  7106          CLL RTL         / WvdM: Low 4 Bits to high byte 2
00422  7006          RTL             / WvdM
00423  3117          DCA TEMP
00424  2113          ISZ LOC
00425  1113          TAD LOC
00426  1124          TAD C200
00427  7640          SZA CLA         /ARE WE AT OS/8 BOUNDARY?
00430  5236          JMP STFLD3+1    /NOT AT END OF FIELD (DJG)
00431  3113          DCA LOC         /OK, TO NEXT FIELD
00432  6214          RDF             /
00433  1377          TAD (6211       /BUILD CDF
00434  3235          DCA STFLD3
00435  7000  STFLD3, NOP
00436  2114          ISZ LEN         /END IF BUFFER?
00437  7410          SKP             /NO
00440  5273          JMP ENDBK       /YES
00441  1513          TAD I LOC
00442  1111          TAD CHKSM
00443  3111          DCA CHKSM
00444  1513          TAD I LOC
00445  7106          CLL RTL
00446  7006          RTL
00447  7004          RAL             / New: High 4 bits to low byte 2
00450  0121          AND C17         / (WvdM) was AND C360
00451  1117          TAD TEMP
00452  4276          JMS PUN
00453  7300          CLA CLL
00454  1513          TAD I LOC
00455  4276          JMS PUN
00456  2113          ISZ LOC
00457  1113          TAD LOC
00460  1124          TAD C200
00461  7640          SZA CLA         /ARE WE AT OS/8 BOUNDARY?
00462  5270          JMP STFLD4+1    /NOT AT END OF FIELD (DJG)
00463  3113          DCA LOC         /OK, TO NEXT FIELD
00464  6214          RDF
```

```
00465  1377          TAD (6211       /BUILD CDF
00466  3267          DCA STFLD4
00467  7000  STFLD4, NOP
00470  2114          ISZ LEN
00471  5206          JMP OUT
00472  5600          JMP I OUTBLK
00473  1117  ENDBK,  TAD TEMP        /SEND LAST PART OF WORD
00474  4276          JMS PUN
00475  5600          JMP I OUTBLK

00476  0000  PUN,    0               / Send byte out serial port
00477  6026          PLS             / Punch for testing with emulator
              /       TLS2            / Send out console
00500  7300          CLA CLL
00501  1110          TAD CBLOCK
00502  6021          PSF
              /       TSF2            /Wait until character sent
00503  5302          JMP .-1
00504  7200          CLA
00505  5676          JMP I PUN

00577  6211
                     $
```

AFIELD 0000	KSF2 6031	TSF2 6041
ARGBK 0247	K77 1063	TSK2 6045
ARGSZ 0245	K7700 1015	UNIT 0146
BCNT 0115	LEAVE 0752	WCT 0142
BLKFLD 0103	LEN 0114	WDSBLK 0201
BLOCK 0141	LOC 0113	WORDS 1174
BUFF 0137	MEMERR 0176	WRITE 1017
CBLOCK 0110	MFIELD 0000	WRLP 1035
CFATAL 1172	MWORDS 0102	WRQUAD 1131
CHKFTL 0147	M110 0723	XFIELD 0651
CHKSM 0111	M20 0134	XQUAD 1171
CHKSUM 0145	M32 0135	
CRWCOM 1173	NOSEL 0322	
CXGO 0761	NUMBLK 0101	
C10 0120	OLDBUF 0136	
C100 0123	OLDFLD 0736	
C1000 0127	ORIGIN 0600	
C1400 0130	OUT 0406	
C17 0121	OUTAD 0040	
C200 0124	OUTBLK 0400	
C300 0125	OUTBL1 0261	
C360 0126	OUTBL2 0336	
C374 0140	PGCT 0762	
C6201 0131	PUN 0476	
C6203 0132	RDLP 1072	
C6211 0133	RDLP2 1124	
C70 0122	RDQUAD 0754	
C7600 1025	RDSIZE 0107	
DONE 0267	RDSZLP 0217	
DRIVE 0010	READ 1064	
DRVSEL 0104	READST 0112	
DTA0 0600	REASON 0116	
DUMPLP 0225	RETRY 0100	
ENDBK 0473	REVGRD 1003	
ENDZ 0725	RWCOM 0653	
EQUFUN 1140	SDLC 6774	
EQUTMP 0144	SDLD 6775	
ERRCN2 0106	SDRC 6776	
ERRRET 0316	SDRD 6777	
EXIT 0744	SDSQ 6773	
FATAL 0746	SDSS 6771	
FIELDS 0105	SDST 6772	
FINISH 0177	SRCH 0700	
FOUND 0731	START 0200	
FSTBAD 0342	STFLD1 1050	
FUNCT 0143	STFLD2 1113	
GETCHK 1155	STFLD3 0435	
GO 0664	STFLD4 0467	
GOON 1000	TCF2 6042	
INAD 0030	TEMP 0117	
KCC2 6032	TFL2 6040	
KCF2 6030	TLS2 6046	
KIE2 6035	TPC2 6044	
KRB2 6036	TRYCNT 0763	
KRS2 6034	TRY3 0734	

ERRORS DETECTED: 0
LINKS GENERATED: 0