



# DECUS

## PROGRAM LIBRARY

DECUS NO.	8-393
TITLE	QUEING TC01/TU55 DECTAPE ROUTINES
AUTHOR	James Crapuchettes
COMPANY	Stanford Electronics Labs Stanford, California
DATE	September 29, 1970
SOURCE LANGUAGE	PAL III

### ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

DEBUS

THE GREAT LIBRARY



[Faint, illegible text, possibly bleed-through from the reverse side of the page]

Library

[Faint, illegible text at the bottom of the page, possibly bleed-through]

## QUEING TC01/TU55 DECTAPE ROUTINES

DECUS Program Library Write-up

DECUS NO. 8-393

### New TC01/TU55 DECTape Routines:

These routines, which are a much modified version of DEC-08-FUBO (previously DEC-08-31U), provide the user with the ability to read and write 128 words (one memory page) from/onto standard 129 word DECTape blocks. Successive blocks are transferred into/from successive 128 word areas of memory. The routines will transfer into/from any memory field, will begin searching in either the forward or the reverse direction for the DECTape block at which the transfer will begin, and will queue one read/write request to keep the DECTape in motion (and transferring data) as continuously as possible.

The only restrictions placed on these routines is that they must be located in field 0 of memory and that they will read and write in the forward direction only. Also, as with other DECTape routines, they cannot transfer data across field boundaries, i.e. location 7777 of a field is followed by location 0 of the same field.

### Areas of application for these routines:

These routines can be used to advantage in any program which processes data which is read from or written onto DECTape at a higher rate than the average transfer rate of DECTape transfer. The processing rate, when compared to the maximum DECTape transfer rate of about 7.5 Kwps, falls into three categories:

2.

1. Whenever the processing rate is always higher than the transfer rate, these routines can keep the DECTape in continuous motion, causing the actual transfer rate to approach the maximum rate, and allowing the data on one complete DECTape to be transferred and processed in about 1 minute. This can be particularly important when only a small area of memory is available for DECTape buffers, because other routines would then need a considerable amount of time to search for the start block for the transfer, decreasing the average transfer rate by as much as a factor of 13.

2. Whenever the processing rate is lower than the maximum rate of the DECTape but greater than the average rate (which includes the time spent searching for the transfer start block), using these routines in conjunction with multiple (more than 2) buffers can raise the average rate (compared to the average rate of other routines) as much as is needed for maximum efficiency in the program. Note that more than 2 buffers are needed in order that there be sufficient buffer space after one buffer is filled, for the routines to search for the transfer start block in case the DECTapes have stopped. Since these routines will queue a request, the actual size of the buffers is not necessarily important and several small buffers may work better than 1 or 2 large buffers.

3. Whenever the processing rate is lower than the average transfer rate of the DECTape (when using other routines) but may include high peak rates, the technique of using several small buffers may be used to advantage as described above.

The disadvantages of using these routines, when compared to other DECTape routines in usage, is that they do not attempt to re-try a read or write when an error occurs and that they require about 1 1/4 pages of core. The last disadvantage is somewhat lessened by the ability of the routines to handle small buffers.

#### Calling sequences:

These routines consist of several subroutines. Of these, two (RW128 and DWAIT) have direct use in user programs and two others (DTRAR and DTXAR) may have usage in special instances.

#### Subroutine RW128:

This subroutine is the Read/Write subroutine which controls DECTape transfers. It has a calling sequence:

```
JMS (RE128)    /Effective JMS to RW128
CORE          /Start address in memory for the transfer
USFW         /Unit, Search, Field and Write (see below)
BLKNO        /DECTape block number at which transfer starts
-BLKS        / -(Number of DECTape blocks to transfer)
Return here with DECTape in motion
```

The control word, USFW, is divided into the following fields:

```
U (Bits 0-2) = DECTape unit number, 0 (for 8) through 7
S (bit 3)    = Search start direction. 0 to start in reverse,
              1 to start forward. (Xfer always forward.)
F (Bits 6-8) = Field of memory for the data transfer.
W (Bit 9)    = Write or read. 1 for write, 0 for read.
```

The search direction bit, S, allows the program to specify the direction in which the DECTape should begin moving when it is searching for BLKNO. This can save some time when the desired direction is known.

When program control returns from the subroutine, a DECTape will be in motion and the AC will be cleared (= 0). The state of the Link is uncertain. The DECTape unit in motion at the time that program control returns may or may not be the same unit just requested. Furthermore, the unit may be performing one of the following three functions: a) searching for BLKNO on the unit just specified, b) transferring data as specified by the last request made to the program, or c) transferring data as specified by this request. Due to this fact, a small amount of care must be exercised when using these routines, as will be described in the section on usage.

**Subroutine DWAIT:**

This subroutine is used by a calling program (and also by RW128) to find the current status of the DECTape and RW128 queue. Control will not return will not return to the calling program until one of two conditions is true. The desired condition is determined by the setting of the Link when the subroutine is called (the AC should be cleared). If the Link is set (a 1), control will return when (if) no DECTape is in motion. If the Link is clear (a 0), control will return when (if) either no DECTape is in motion or the queue in RW128 is empty. This subroutine has a calling sequence:

```

STL or CLL      /SeT Link or CLeAr Link
JMS (DWAIT)     /Effective JMS to DWAIT
Return here when specified condition true (AC = 0)

```

## Subroutine DTRAR:

This subroutine checks the motion bit in the DECTape A (command) register and returns to the calling address+1 if the bit is a 0 (no DECTape in motion) or to the calling address+2 if the bit is a 1 (a DECTape is in motion): It has a calling sequence:

```
JMS (DTRAR)    /Effective JMS to DTRAR, AC = 0
Return here if no DECTape in motion (AC = 0)
Return here if a DECTape is in motion (AC = 0)
```

This subroutine should be called with the AC cleared and returns with the AC cleared and the Link undisturbed. It can be used by a program to determine the status of the DECTape control.

## Subroutine DTXAR:

This subroutine reads the DECTape A (command) register, masks the AC (by an AND) with the contents of the calling address+1, issues the DTXA iot command (which exclusive-ors the contents of the AC with the contents of the A register), and then returns to the calling address+2 with the AC clear. Any bits set in the AC when the subroutine is called will be inclusive or'ed with the contents of the DECTape A register (the read) before masking. This subroutine has a calling sequence:

```
JMS (DTXAR)    /Effective JMS to DTXAR
MASK           /Mask for AC or'ed with A register.
Return here with AC cleared.
```

This subroutine, with a mask of 200 (octal), will stop any DECTape unit which is in motion. This can be used in a program when something occurs which requires that the DECTape stop immediately. Any request left in the queue in RW128 will be ignored after this is done.

**Usage of the routines:**

Usage of this routine is somewhat similar to that described for its parent in the DEC writeup. If the user is unfamiliar with this routine, it is suggested that he read the writeup available from DEC.

This routine requires three symbols to be defined on page 0 of memory (field 0). These three symbols are MCOM, DIS, and DTERR. MCOM is used by the DECTape interrupt handling routines to point to the current entry point in the interrupt handler. It must be defined by the user in his program but is initialized by the routine. Its usage by the user is described in the DEC writeup. DIS must be a pointer to the user's interrupt dismiss routine. This is also described in the DEC writeup. DTERR must be a pointer to the user's DECTape error processing routine. When a DECTape error occurs, the routines stop the DECTape, load the AC with the contents of the DECTape B (status) register, and then execute a JMP I DTERR. Unlike the DEC version, the status (errors) will be undisturbed in the B register also. A few more comments on error processing will follow shortly.

Unlike the DEC routines, this routine does NOT include an ION instruction to enable the program interrupt. It is important that the user program enable the interrupt shortly after the first time that the routine is called. If it is a problem that the ION is not included, there is room to add this one instruction in the subroutine labeled DSSET, which sets up a search for BLKNO.

In some programs it may be necessary to process a DECTape error with the program interrupt on (as in the case of buffered Teletype

output). If it is desired to do this, it is necessary to make a small change in the interrupt skip chain to properly ignore an illegal DECTape flag. This illegal flag is caused by the DTXA instruction which stops the DECTape unit. In the case of a select error (which is caused by either selecting an off-line unit or by trying to write on a write-locked unit), this DTXA instruction will cause the error to occur again. Simply checking for a DECTape flag (with a DTSF) to determine the device which caused an interrupt will cause the program to go to the DECTape interrupt handler with possible disastrous results. To solve this problem, the DTXA instruction which stops the unit also clears the bit in the DECTape A register which allows the DECTape flag to cause a program interrupt. The interrupt skip chain should therefore check this bit before going to the DECTape handler. An example follows:

```

/Save the AC and Link after the interrupt...
DTSF          /Is the DECTape flag set?
JMP .+5       /No, try the next device
DTRA          /Yes, is control connected to P.I.?
AND P4        /C(P4) = 4
SZA CLA       /If AC = 0, not connected, try next device
JMP I MCOM    /All OK, go handle DECTape interrupt
... /Continue with device identification routine

```

This modified routine will properly handle "true" DECTape interrupts while ignoring the flag if it could not have caused the interrupt.

Because RW128 will queue a DECTape read/write request, care must be used to keep from using a buffer before the transfer into or out of it is finished. The table on the following page shows the possible functions that can be in progress after repeated calls to RW128 and to DWAIT. It is suggested that the user read and attempt to understand the

table before using the routine. The following words are used in the table:

<b>SEARCH</b>	Signifies that the routine is searching the DECTape for BLKNO.
<b>XFER</b>	Signifies that a data transfer (read/write) is taking place.
<b>Q</b>	Signifies that the Queue in RW128 (which has a depth of one request) is full. If it is necessary for RW128 to perform a SEARCH for a specified request, the Queue will remain full until the actual XFER begins for that request.
$\bar{Q}$	Signifies that the Queue in RW128 is empty.
<b>n-1, n, n+1,</b>	Signify the number of the current request. In each of the examples shown, the current request is number n, the last request is n-1, etc.

Should there be any questions about the routine or its use, I will do what I can to be of help.

Jim Crapuchettes  
Stanford Electronics Labs  
Stanford, Calif.

Use of RW128:

"(wait)" means that program control did not return from the routine immediately

<u>Function</u>	<u>Status of DECTape &amp; Queue</u>		
before call →	STOP, $\bar{Q}$	SEARCH, $Q, n-1$	XFER, $\bar{Q}, n-1$
JMS RW128 . (n) . . .		(wait)	
on return → CLL JMS DWAIT	SEARCH, $Q, n$	XFER, $Q, n-1$	XFER, $Q, n-1$
on return → (note that status is same in all cases.)	XFER, $\bar{Q}, n$	XFER, $\bar{Q}, n$	XFER, $\bar{Q}, n$

before call →	STOP, $\bar{Q}$	SEARCH, $Q, n-1$	XFER, $\bar{Q}, n-1$
JMS RW128 . (n) . . .		(wait)	
on return →	SEARCH, $Q, n$	XFER, $Q, n-1$	XFER, $Q, n-1$
. . .			
JMS RW128 . (n+1) . . .	(wait)	(wait)	(wait)
on return →	XFER, $Q, n$	XFER, $Q, n$	XFER, $Q, n$
. . .			
JMS RW128 . (n+2) . . .			
on return →	XFER, $Q, n+1$	XFER, $Q, n+1$	XFER, $Q, n+1$

RW128, QUEING DECTAPE ROUTINES: SOURCE, BIN &amp; LISTING 15 OCT 1970

NAME	BLK#	BLKS	S.A.	CORE LOCS	
INDEX	7777	8	7777	07400-07577	
START	0007	9	7667	07400-07577	
UPDATE	0020	7	7600	06000-07577	
CHANGE	0027	5	7600	06400-07577	
GETSYS	0034	5	7600	06200-07377	
XSYS	0041	13	7667	04400-07577	
XEDIT	0056	11	0147	00000-02577	
XPAL	0071	31	0200	00000-07577	
XLOAD	0130	5	7200	06400-07577	
XMOVE	0135	11	0200	00000-02577	
XINDEX	0150	9	0200	00000-02177	
XODT8	0161	3	7000	07000-07577	
XLIST	0164	12	0200	00000-02777	
XDUP	0200	4	0200	00000-00777	
XTAPE	0204	24	0177	00000-05777	
DECU31	0234	12	7667	07400-07577	
SRWTAP	0250	20	7667	07400-07577	
CONVT	0274	31	0200	00000-07577	
SRW128	0333	30	7667	07400-07577	Source file
BRW128	0371	6	7667	07400-07577	Binary file
LRW128	0377	50	7667	07400-07577	Listing file
RW128T	0461	3	7667	00000-00577	Test program

NUMBER OF FILES: 21  
 NEXT FREE BLOCK: 0464  
 NEXT FREE WORD: 6236  
 WORDS LEFT: 0341

```

0001          /NEW DECTAPE ROUTINES: WILL QUEUE ONE REQUEST TO
0002          / KEEP A DECTAPE IN MOTION IF POSSIBLE.
0003          /JIM CRAPUCHETTES, STANFORD UNIVERSITY, AUG 1970
0004          /
0005          /CALLING SEQUENCE:
0006          /      JMS RW128      (EFFECTIVE)
0007          /      CORE      START ADDR FOR DATA XFER
0008          /      USFW      U (BITS 0-2) = UNIT #
0009          /      S (BIT 3) = SEARCH DIRECTION. 0 TO
0010          /      START IN REVERSE, 1 FOR FORWARD.
0011          /      F (BITS 6-8) = DATA FIELD
0012          /      W (BIT 9) = READ (0) OR WRITE (1)
0013          /      BLKNO     START BLOCK #
0014          /      -BLKS     -NUMBER OF BLKS TO XFER
0015          /      RETURN HERE WITH DECTAPE IN MOTION.
0016          /NOTE: "-BLKS" MUST BE THE LAST ITEM IN THE CALLING
0017          / SEQUENCE BECAUSE IT IS USED TO SIGNIFY SOMETHING
0018          / IN THE QUEUE.
0019
0020 0200 0000 RW128, 0      /READ OR WRITE 128 WORD BLOCKS
0021 0201 7300      CLA CLL
0022 0202 4763      JMS I DRWAIT      /WAIT FOR ROOM IN QUEUE
0023 0203 7240      STA
0024 0204 4332      JMS DGET      /GET (START LOC-1) FOR NEXT
0025 0205 3367      DCA DNSTAD      / TRANSFER
0026 0206 4332      JMS DGET
0027 0207 3364      DCA DNUSFW      /NEXT TRANSFER INFO
0028 0210 4332      JMS DGET
0029 0211 3366      DCA DNSBLK      /NEXT XFER START BLOCK
0030 0212 4332      JMS DGET
0031 0213 3365      DCA DNMBLK      /NEXT -(# BLKS TO XFER)
0032 0214 4761      JMS I DRTRAR      /ANYTHING IN MOTION?
0033 0215 4336      JMS DSSET      /NO, SET UP THE SEARCH
0034 0216 5600      JMP I RW128      /OK, NOW RETURN
0035
0036 0217 6772 DLOOK, DTRB      /LOOK FOR END ZONE
0037 0220 7006      RTL
0038 0221 7710      SPA CLA
0039 0222 5323      JMP DTURNX      /IN END ZONE; TURN AROUND
0040 0223 6772      DTRB
0041 0224 7710      SPA CLA
0042 0225 5325      JMP DER      /ERROR FLAG
0043 0226 6761      DTRA      /PUT DIRECTION BIT IN LINK
0044 0227 7006      RTL
0045 0230 7006      RTL
0046 0231 7600 D7600, 7600      /OPERATE 2 CLA
0047 0232 1372      TAD DTBLK      /AT THE RIGHT BLOCK?
0048 0233 7041      CIA
0049 0234 1366      TAD DNSBLK
0050 0235 7450      SNA
0051 0236 5245      JMP DTFIND      /YES, FOUND BLOCK, CHECK DIR
0052 0237 7041      CIA
0053 0240 7420      SNL      /IF IN REV, GO 2 BLKS TOO FAR
  
```

0054	0241	7001		IAC	
0055	0242	7620		SNL CLA	
0056	0243	1375	DTURN,	TAD D400	/CHANGE DIRECTION IF HERE
0057	0244	5271		JMP DRXOR	/XOR TO A STATUS AND DISMIS
0058	0245	7620	DTFIND,	SNL CLA	/TEST MOTION
0059	0246	5271		JMP DRXOR	/DON'T TURN YET
0060	0247	2037		ISZ MCOM	/POINT INTERRUPT TO DATA
0061	0250	4762	DNSET,	JMS I DRTXAR	/CLEAR CURRENT FUNCTION
0062	0251	0070		70	
0063	0252	1364		TAD DNUSFW	/SET UP A TRANSFER
0064	0253	6774		DTLB	/SET FIELD
0065	0254	1364		TAD DNUSFW	
0066	0255	0376		AND D7000	
0067	0256	7041		CIA	
0068	0257	3370		DCA DTUNIT	/= -(# OF UNIT DOING XFER)
0069	0260	1365		TAD DNMBLK	/MOVE -# BLKS TO COUNTER
0070	0261	3371		DCA DTMBLK	
0071	0262	3365		DCA DNMBLK	/ & CLEAR OLD INFO (QUEUE OK)
0072	0263	1367		TAD DNSTAD	
0073	0264	3755		DCA I DCAA	/SET UP XFER ADDRESS
0074	0265	1364		TAD DNUSFW	/READ OR WRITE?
0075	0266	7006		RTL	/BIT 9 = 1 FOR WRITE
0076	0267	0373		AND D20	/MASK OFF BIT
0077	0270	1373		TAD D20	/READ = 20, WRITE = 40
0078	0271	6764	DRXOR,	DTXA	
0079	0272	1231		TAD D7600	/SET WORD COUNT FOR 128(10)
0080	0273	3756		DCA I DWCA	
0081	0274	5443		JMP I DIS	/EXIT
0082					
0083	0275	5217		JMP DLOOK	/COME HERE THRU MCOM ON SEARCH
0084	0276	6772	DINT,	DTRB	/READ STATUS B
0085	0277	2372		ISZ DTBLK	/BUMP CURRENT BLOCK NUMBER
0086	0300	7710		SPA CLA	
0087	0301	5325		JMP DER	/ERROR
0088	0302	2371		ISZ DTMBLK	/COUNT BLOCKS, DONE?
0089	0303	5271		JMP DRXOR	/NO, CONTINUE OPERATION
0090	0304	1365		TAD DNMBLK	/YES, IS SOMETHING IN QUEUE?
0091	0305	7650		SNA CLA	
0092	0306	5323		JMP DTURNX	/NO, STOP & EXIT
0093	0307	1364		TAD DNUSFW	/IS QUEUED UNIT NUMBER THE
0094	0310	0376		AND D7000	/ SAME AS THE UNIT WHICH
0095	0311	1370		TAD DTUNIT	/ JUST FINISHED XFER?
0096	0312	7640		SZA CLA	
0097	0313	5321		JMP DNOT	/NO, STOP THIS & RESET SEARCH
0098	0314	1366		TAD DNSBLK	/OK, IS TAPE AT THE RIGHT
0099	0315	7041		CIA	
0100	0316	1372		TAD DTBLK	/ BLOCK NOW?
0101	0317	7650		SNA CLA	
0102	0320	5250		JMP DNSET	/YES, SET UP NEXT XFER
0103	0321	4336	DNOT,	JMS DSSET	/STOP LAST UNIT & SET UP
0104	0322	5443		JMP I DIS	/ NEXT SEARCH, THEN LEAVE
0105					
0106	0323	1340	DTURNX,	TAD D200	/COMPLEMENT MOTION
0107	0324	5243		JMP DTURN	/ & DIRECTION BITS

```

0108
0109 0325 7326 DER,   CLA STL RTL   /= 2, DON'T CLEAR ERRORS
0110 0326 4762      JMS I DRTXAR  /STOP TAPE & XOR OUT INTERRUPT
0111 0327 0206      206
0112 0330 6772      DTRB          /ERROR STATUS B
0113 0331 5442      JMP I DTERR   /GO TO ERROR ROUTINE
0114
0115 0332 0000 DGET,   0           /PICK UP ARGUEMENTS
0116 0333 1600      TAD I RW128
0117 0334 2200      ISZ RW128
0118 0335 5732      JMP I DGET
0119
0120 0336 0000 DSSET,   0           /SET UP SEARCH ON A UNIT
0121 0337 4762      JMS I DRTXAR  /STOP ANYTHING IN MOTION
0122 0340 0200 D200,   200
0123 0341 1357      TAD DCINT     /SET UP INTERRUPT POINTER
0124 0342 3037      DCA MCOM
0125 0343 1364      TAD DNUSFW    /GET INFORMATION
0126 0344 0231      AND D7600    /MASK OFF UNIT & DIRECTION
0127 0345 1374      TAD D214     /ADD: GO, SEARCH, INTERRUPT
0128 0346 6766      DTCA DTXA    /SET IT UP
0129 0347 1375      TAD D400     /XOR THE DIRECTION BIT SO
0130 0350 6764      DTXA         / DNUSFW3 = 0 MEANS REVERSE
0131 0351 6774      DTLB        /0 => FIELD
0132 0352 1360      TAD DBLK     /SET UP SEARCH C.A.
0133 0353 3755      DCA I DCAA
0134 0354 5736      JMP I DSSET
0135
0136 0355 7755 DCAA,   7755    /POINTER TO CURRENT ADDRESS
0137 0356 7754 DWCA,   7754    /POINTER TO WORD COUNT
0138 0357 0275 DCINT,   DINT-1
0139 0360 0372 DBLK,   DTBLK
0140 0361 0412 DRTRAR, DTRAR
0141 0362 0422 DRTXAR, DTXAR
0142 0363 0400 DRWAIT, DWAIT
0143
0144          /NEXT 4 ARE NEXT INFO (QUEUE)
0145 0364 0000 DNUSFW, 0           /UNIT, SEARCH, FIELD & WRITE
0146 0365 0000 DNMBLK, 0           / -(NO. BLKS)
0147 0366 0000 DNSBLK, 0           /START BLK
0148 0367 0000 DNSTAD, 0           /CORE ADDR
0149
0150          /NEXT 3 ARE CURRENT INFO
0151 0370 0000 DTUNIT, 0           / -(UNIT NO.)
0152 0371 0000 DTMBLK, 0           / -(NO. BLKS) COUNTER
0153 0372 0000 DTBLK, 0           /BLOCK NUMBER
0154 0373 0020 D20,   20
0155 0374 0214 D214,   214
0156 0375 0400 D400,   400
0157 0376 7000 D7000, 7000
  
```

RW128 - SPECIAL DECTAPE ROUTINES 30 AUG 1970  
RECORD 2

```

0001          *RW128+200
0002
0003          /ROUTINE TO WAIT FOR EITHER NO MOTION FROM ANY
0004          / DECTAPE (LINK = 1) OR FOR EITHER NO MOTION
0005          / OR ROOM IN THE ROUTINE QUEUE (LINK = 0).
0006  0400  0000  DWAIT,  0
0007  0401  4212          JMS DTRAR          /IS ANYTHING IN MOTION?
0008  0402  5600          JMP I DWAIT          /NO, EXIT
0009  0403  7630          SZL CLA          /YES, WAIT FOR ALL DONE?
0010  0404  5201          JMP DWAIT+1        /YES
0011  0405  1611          TAD I DNMBKI        /NO, ROOM IN QUEUE?
0012  0406  7640          SZA CLA
0013  0407  5201          JMP DWAIT+1        /NO, KEEP WAITING
0014  0410  5600          JMP I DWAIT        /YES, EXIT
0015  0411  0365  DNMBKI, DNMBLK
0016
0017  0412  0000  DTRAR,  0          /GET MOTION BIT FROM DT A REG
0018  0413  6761          DTRA
0019  0414  6761          DTRA
0020  0415  0221          AND DT200        /MASK IT OFF
0021  0416  7640          SZA CLA          /IS IT IN MOTION?
0022  0417  2212          ISZ DTRAR        /YES, RETURN TO CALL+2
0023  0420  5612          JMP I DTRAR        / ELSE RETURN TO CALL+1
0024  0421  0200  DT200,  200
0025
0026          /READ D.T. A REG, MASK IT BY C(CALL+1), THEN
0027          / XOR THE MASKED BITS, & RETURN TO CALL+2.
0028  0422  0000  DTXAR,  0
0029  0423  6761          DTRA          /READ (OR WITH AC)
0030  0424  0622          AND I DTXAR        /AND IT
0031  0425  2222          ISZ DTXAR
0032  0426  6764          DTXA          /XOR BITS
0033  0427  5622          JMP I DTXAR

```

RW128 - SPECIAL DECTAPE ROUTINES 30 AUG 1970  
RECORD 3

```

0001          /TEST ROUTINE FOR RW128
0002
0003          *1
0004 0001 3010          DCA AC          /SAVE THE AC
0005 0002 7004          RAL
0006 0003 5020          JMP INTR          /GO TO INTERRUPT ROUTINE
0007
0008          *10
0009 0010 0000          AC, 0
0010 0011 0000          LK, 0
0011
0012          *20
0013 0020 3011          INTR, DCA LK          /SAVE THE LINK
0014 0021 6771          DTSF          /DECTAPE INTERRUPT?
0015 0022 7410          SKP
0016 0023 5437          JMP I MCOM          /YES
0017 0024 6041          TSF          /PRINTER?
0018 0025 5030          JMP TTYI
0019 0026 6042          TCF          /YES, CLEAR FLAG
0020 0027 5443          JMP I DIS
0021 0030 6031          TTYI, KSF          /KEYBOARD?
0022 0031 5034          JMP HALT
0023 0032 6032          KCC          /YES, CLEAR FLAG
0024 0033 5443          JMP I DIS
0025 0034 7240          HALT, STA
0026 0035 7402          HLT          /DIE!
0027 0036 5035          JMP .-1
0028
0029 0037 0000          MCOM, 0          /ROUTINE LINKS
0030 0040 0400          DWAITI, DWAIT
0031 0041 0200          RW128I, RW128
0032 0042 0035          DTERR, HALT+1 /D.T. ERROR
0033
0034 0043 0044          DIS, .+1          /POINTER TO DISMIS ROUTINE
0035 0044 1011          TAD LK          /RESTORE EVERYTHING
0036 0045 7110          CLL RAR
0037 0046 1010          TAD AC
0038 0047 6244          RMF
0039 0050 6001          ION
0040 0051 5400          JMP I 0
0041
0042          *100
0043
0044 0100 7604          START, LAS
0045 0101 3112          DCA B1          /SET B1 = SR
0046 0102 7402          HLT
0047 0103 6001          ION          /TURN ON INTERRUPT
0048 0104 7604          LOOP, LAS
0049 0105 7710          SPA CLA          /KEEP GOING?
0050 0106 5130          JMP STOP          /NO
0051 0107 4441          JMS I RW128I          /TEST THE ROUTINE
0052 0110 1000          BUF1
0053 0111 1000          1000          /UNIT 1, READ

```

```

0054 0112 0000 B1, 0
0055 0113 7773 -NBLKS
0056 0114 1112 TAD B1 /SET UP B2
0057 0115 1137 TAD NBLK
0058 0116 3122 DCA B2
0059 0117 4441 JMS I RW128I /DO ANOTHER
0060 0120 3000 BUF2
0061 0121 1000 1000
0062 0122 0000 B2, 0
0063 0123 7773 -NBLKS
0064 0124 1122 TAD B2 /BUMP B1
0065 0125 1137 TAD NBLK
0066 0126 3112 DCA B1
0067 0127 5104 JMP LOOP
0068 0130 7320 STOP, STL CLA /WAIT FOR ALL DONE
0069 0131 4440 JMS I DWAITI
0070 0132 6002 IOF
0071 0133 1112 TAD B1 /DISPLAY B1
0072 0134 7402 HLT
0073 0135 7200 CLA
0074 0136 5100 JMP START
0075
0076 BUF1= 1000
0077 BUF2= 3000
0078 NBLKS= 5
0079
0080 0137 0005 NBLK, NBLKS
0081
0082 /PATCH TO DWAIT TO DISPLAY A WORD IN THE MQ
0083 *DNMBKI-2
0084 0407 5232 JMP PATCH
0085
0086 *DTXAR+10
0087 0432 7604 PATCH, LAS /READ SR
0088 0433 3222 DCA DTXAR
0089 0434 1622 TAD I DTXAR /USE IT AS ADDRESS
0090 0435 7421 MQL /DISPLAY IN MQ
0091 0436 7200 CLA
0092 0437 5201 JMP DWAIT+1
    
```