



PERMANENT
MEMORANDUM

Paul Lambert

M -1131

PAGE 1 OF 16

DATE November 16, 1961

SUBJECT
TO SIMPLE TAPE CONTROL PROGRAM
PDP-1 Distribution List

PRELIMINARY

ABSTRACT

A routine for command magnetic tape to read forward, backward, space, write, read check, and other features. The calling sequence requires six registers in core memory.

FROM Steve Lambert

APPROVED BY

Harlan E. Anderson

INTRODUCTION

This program allows the operator to:

1. Read forward either binary or bcd for any set block length (block gap must be 3/4" or more with the exception when a tape transport is in the continue operation).
2. Read backward either binary or bcd for any set block length.
3. Space forward or backward a stated number of blocks.
4. Write forward either binary or bcd in standard IBM format.
5. Read check forward or back.
6. Rewind the tape transport back to load point.
7. Operate tape transports in a continuous mode, i.e., read or write on a given tape unit without stopping the transport between blocks.
8. Read or write with an early completion, i.e., the transport stops at end of block but programmer is given 3 extra milliseconds.
9. Examine status of a tape dec without interrupting tape control.
10. Give an illegal tape operation command without effecting the tape control unit.
11. Monitor the block count and status bits after a normal return.
12. Check for parity or mischaracter errors during read operations.
13. Control the operations of 1 to 3 tape units.

The calling sequence is such:

```
law or lac command  
jda tape  
initial address back or forward (always smaller value)
```

final address & 1 back or forward (always larger value)
hlt non normal return
hlt normal return

<u>BITS</u>	<u>COMMANDS</u>
6	continue different unit - early complete - forward or back
7	its your gap - forward or back
8	read check - forward or back
9	space - forward or back
10	operate* - forward or back
11	rewind if a (1) - normal operation if (0)
12	back if (1) - forward if (0)
13	write if (1) - read if (0)
14	even parity (bcd) alpha if (1) - odd parity (binary) if (0)
15	not used
16	unit number
17	unit number

*Operate bit must be on to initiate a tape operation, otherwise, a non normal return will occur.

<u>BITS IN IO</u>	<u>STATUS</u>
0	ready when (0)
1	rewind when (0)
2	write lock out when (0)

- | | |
|---|---------------------|
| 3 | load point when (0) |
| 4 | full reel when (0) |
| 5 | low reel when (0) |
| 6 | end point when (0) |
| 7 | auto when (0) |

When the ready bit is off, it means that the tape unit in question may be used. Bit 7, auto, says that the tape unit exists, is turned on, the tape control unit is plugged in and auto switch is in an auto position.

OPERATING PROCEDURE

Turn on the desired tape units and tape control. Be sure to press the stop button to clear out motor control flip-flops and other associated logic. Begin tape operation using the calling sequence stated above. The initial address must be greater than 0 but not equal to 0 and the final address must be less than 7777 but not equal to 7777. Be sure to make the final address & 1.

NOTES

A non normal return will occur if the operate bit is off. At this time, the IO will have the status of the tape unit in question, or if the last operation was a continue type, it will have the status of the tape dec that is in operation. Normally, the accumulator will be cleared except when the last operation was a continue type. Then the AC will contain the command of the last operation. If the rewind command is given a normal return will occur with the accumulator cleared and the status of the tape dec just commanded in the IO. After a read or a write operation, the normal return will show the block count in the accumulator and the status of the tape transport in the IO. A write operation will always return to a normal return. However, in a read operation, it is possible that a parity or a mischaracter will occur. If this happens, a non normal return will result. If a parity error was detected, Flag 4 will be set, the IO will contain the status bits, and the current address will appear in the AC.

If a parity error and mischaracter error occur, the IO contains status, AC contains current address and Flag 4 and 6 are set. Mischaracter sets Flag 6.

Flag 2 is used to detect characters.

One should not try to begin reading in the middle of a block. Flag 2 will not get set immediately because the tape transport will not have had a chance to move for about 1.8 milliseconds. A non normal return will occur with the result of no characters being read.

Writing bcd with all zeros is the same as saying "writing blank tape". This program will assume that a character was missed if it sees blank tape within a block of characters. One should be careful when writing bcd to be sure all zeros are coded unless the purpose is to write blank tape.

When working near end point it is suggested that blank tape be written at end point to clear out undesirable characters that might interfere with read programs.

It is suggested that the programmer should not try to replace blocks as there is a possibility that erroneous results will appear when reading.

If continue operations are desired, bit 7, the programmer must generate the proper timing for interrecord gaps during writing. However, read operations may be flexible. In this mode, it is necessary to allot approximately 1.0 millisecond to the tape control program. This leaves about 9.0 milliseconds program time for transition of IBM 3/4" interrecord gaps.

Writing records with the "your gap" bit on adds a new dimension to the method of writing. A means must be provided for getting tape up to speed before a write command is initiated. This is due to the fact that the write start delay is taken away when bit 7 is on. Thus, one must make the last operation a "your gap" type before a write calling sequence (with bit 7 on) is used. Example:

```
law 2602           ,space forward - your gap - unit 2
jda tape
l
4000              ,not used with spacing
hlt
```

```

jsp 9 milliseconds
law 2222           ,write forward - normal stop - unit 2
jda tape
1
4000
hlt
hlt               ,final stop after writing two blocks

```

Normally TCP will generate a 3/4" gap during all operations. If one wishes to have additional program time, bit 6 will take away the necessary 3 milliseconds transport stopping time and give it to the programmer. If this time is not used up before another command is given, the 3/4" IRG will become smaller when writing.

During a rewind operation, the initial and final addresses of the programmers calling sequence are not used. At this time, the block count is set to zero.

Read checking does not destroy the contents of memory. When a character does not readcheck, Flag 6 will be set and a non normal return will occur with the AC containing the address of the word that did not read check. The IO will contain the status bits. This program does not readcheck the last address when block length is equivalent to the difference between the initial and final addresses of the calling sequence. However, if the block length is greater than this difference, the last word will be checked.

When a spacing command is given, the initial address represents the number of blocks forward or back of the final address is not used. Flag 6 will be set during this operation but may be ignored.

Here are some examples of octal commands that can be executed.

```

000202           ,read forward binary - unit 2
000242           ,read back binary - unit 2
000341           ,rewind - unit 1
000222           ,write binary - unit 2
000601           ,space forward - unit 1
000642           ,space backward - unit 2

```

SYMBOLIC PROGRAM

In the following program, the first few registers are used to determine the initial and final addresses and where it says operate

bit off, is where a decision is made to see whether or not one should continue down through the program or give a non normal return. At P102, a non normal return will occur if the operate bit is off. From P104 to P60, a decision is made as to whether the operation was a continue type or not. If it was not a continue, then a jump is made to P60 where the tape dec is started. Then from P103 down to just before P113, initializing is taking place for a read forward operation. However, down at P113, this initializing is altered depending upon whether or not the operation is a read operation or one of the other commands. These decisions are made between P113 and P120. From P120 down to P134 the initial and final addresses are put into the proper subroutines of the tape control program. P134 is the pointer to the different small subroutines used for back reading, forward reading, and writing, etc. Also P131 is used in this same manner. From P132 down to P45 is the read forward subroutine. P45 down to P56 is the jump out spot for all the subroutines. This is where all the decisions are made as to whether or not it is an early complete or parity error or mischaracter. From Bread down to P50 is the back read subroutine. From Write down to Space is the write subroutine. From Space to rdck is the space routine. From rdck down to Q30 is the read check routine. From Q30 to read is where the writing is done. From the word read down to P31 is where all read, readcheck, and spacing programs jsp to. Here the timing is done per character and when a character is missed a jump back to the assigned subroutine will occur at P31. From delay to the word exit is self explanatory, this is the one delay used in the program for all starting and stopping. The registers following exit are all fixed or variable storage area. From T112 down to T120 contains the proper delay timing for 3/4" IBM format interrecord gaps.

```
,TAPE CONTROL PROGRAM S. L.  
,calling sequence  
,law or lac command  
,jda tape  
,initial address back or forward  
,final address back or forward  
,hlt non normal return  
,hlt normal return  
opd msm          720073  
opd mwc          720071  
opd mrc          720072  
opd mec          720034  
opd mcb          720070  
org 7000  
tape             0                ,command  
                dap c & 1  
Q100            lac  
                dap k1  
                idx Q100  
                lac * Q100  
                dap k2  
                idx Q100  
                dap k3  
                lio tape  
                rir s8  
                spi                ,operate bit off  
                jmp P104  
                lac T105  
                sza                ,tape is stopped  
                jmp P102  
                lio tape  
                msm  
                cla  
P102            mec                ,strobe status  
                jmp * k3  
P104            lac tape  
                and T111  
                dac T3  
                lac T105            ,what unit number  
                dac T2  
                and T111  
                sas T3                ,was last operation continue  
                jmp P60  
                mec  
                spi                ,is present unit busy  
                jmp P103
```



```
P60          lio tape
             msm          ,start up tape transport
P103         dzm t105
             dzm t122    ,flag 4
             law p132
             dap p131
             law p131
             dap p134
             law Q6
             dap p122
             dap Q7 & 1
             dap Q7 & 3
             dap Q10 & 2
             law 6
             dap p32
             law Q15
             dap Q13 - 4
             law p44
             dap p133
             law p50
             dap p125
             law p52
             dap p130
             law Q1
             dap Q13
             mcb
             lio tape
             ril s6
             spi          ,early complete different unit
P113         ril s1
             spi          ,its your gap
             jmp p106
P114         ril s1
             spi          ,read check
             jmp p107
P115         ril s1
             spi          ,space back or forward
P116         ril s2
             spi          ,rewind
             jmp p137
             ril s1
             spi          ,forward or reverse
P117         jmp p111
             ril s1
             spi          ,read or write
```

```

      jmp p112
      jmp p120
p105  law p122
      dap p123
      jmp p113
p106  lac tape
      dac t105
      law p122
      dap p133
      dap p125
      dap p130
      jmp p114
p107  law rdck
      dap p134
      jmp p115
p110  law space
      dap p134
      law p134
      dap p133
      dap p125
      lac k1
      cma
      dac t110
      jmp p116
p137  idx k3
      dzm bk
      cli cla
      msm
      mec
      jmp * k3
p111  law bread
      dap p131
      jmp p117
p112  lac t2
      sza *
      jmp ¢ & 4
      law write & 2
      dap p134
      jmp p120
      law write
p120  dap p134
      lac k1
      dap Q31
      law * 1
      add k1
```

```
dap Q6
dap Q15
lac k2
dap Q33
dap Q11
dap t103
cli 7
p134 jmp p131 ,is it write, read, rdck, space
p131 jmp p132 ,is it read forward or back
p132 lac t107
dac Q7
lac Q32 & 1
dac Q2
dac Q4
lac Q30
dac Q5
idx bk
jsp read
p133 jmp p44 ,your gap
p44 lac t113 ,fixed delay for eob
jda delay
p45 lio t3
msm
p123 jmp p46 ,early complete different
p46 lac t114 ,fixed stop delay
jda delay
p122 lac Q6
p32 szf 6
jmp p56
lac t122
sza *
jmp c & 3
stf 4
jmp p56
idx k3
lac bk
mec
jmp * k3
p56 lac * p122
mec
jmp * k3 ,non-normal return
bread lac Q6
dap Q11
```

```

dap t103
lac k2
dap Q6
dap Q15
lac t104
dac Q7
lac t106
dac Q2
dac Q4
dac Q5
law * 1
add bk
dac bk
stf 6
jsp read
p125 jmp p50 ,your gap
p50 lac t115 ,delay back
jda delay
jmp p45
write lac t116
jda delay
idx bk
law Q31
dap p122
jmp Q30
Q35 lac t117
jda delay
mcb
p130 jmp p52 ,your gap
p52 lac t120 ,stopping delay
jda delay
jmp p45
space law p22
dap Q13
isp t110
jmp p131
law p44
dap p133
law p50
dap p125
law 2000
dap p32
jmp p131
rdck law Q6
```

```

                                dap Q13 - 4
                                law Q15
                                dap p122
                                dap Q7 & 1
                                dap Q7 & 3
                                dap Q10 & 2
                                jmp p131
Q30                               nop
                                law * 2
                                dac t2
Q31                               lio
Q32                               mwc
                                ril s6
                                cla
                                jda delay
                                isp t2
                                jmp Q32
                                nop
                                mwc
                                nop
                                idx Q31
                                sad Q33
                                jmp Q35
                                jmp Q30
read                              dap p31
                                mcb
                                lac t112                ,delay before error miss
                                dac t2
p2                                szf 2
                                jmp c & 4
                                isp t2
                                jmp p2
                                jmp p26 - 1            ,tape has no characters
                                law t123
                                cli
                                dap
                                mrc
                                cla
                                nop
Q13                              jmp Q1
Q1                                szf 2
                                jmp Q2
                                szf 2
                                jmp Q2
```

```
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp Q2
szf 2
jmp p22
Q2   ril s6
Q7   jmp Q10
     add Q6
     mrc
     dac Q6
     jmp Q3
Q10  lac Q6
     mrc
     idx Q6
     jmp Q3
Q3   sad Q11
     jmp p26
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     szf 2
     jmp Q4
     jmp Q14
```

```
Q4         ril s6
            mrc
Q5         nop
Q6         dio
            cli clf 6
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q12
            szf 2
            jmp Q14
Q12        lac t123
            mrc
Q15        sas
            jmp p26 - 1
            jmp Q1
Q14        law * 0
            dac t2
            szf 2
            jsp Q16
            isp t2
            jmp Q14 & 2
            jmp p26
Q16        dap c & 3
            stf 6
            mrc
            jmp
p22        szf * 6           ,is it eob
            jmp Q14
            mrc
            clf 7
            jmp p2
            stf 6
```

```
p26          szf 4
             idx t122
             law * 14          ,delay after character
             dac t2
             mrc
p27          szf 2
             jmp p26 & 2
             isp t2
             jmp p27
             cli clf 4
p31          jmp
delay        0
             dap exit
             isp delay
             jmp ¢ - 1
exit        jmp
Q11         dio
Q33         lio
t2          0          ,index counter
t3          0          ,unit no & temp stor
k1          0          ,initial address
k2          0          ,final address
k3          0          ,non norm return
t7          0          ,address equal
t103        sas
t106        rir s6
t104        law * 1
t105        0          ,continue memory
t107        jmp Q10
t110        0          ,space count
bk          0          ,block count
t111        000003    ,unit mask
t112        -1777     ,begining delay onread mis char
t113        -112      ,forward delay read before stop
t114        -306      ,final stop delay read & write
t115        -530      ,back read delay before stop
t116        -447      ,write start delay
t117        -13       ,eob delay write
t120        -603      ,write delay before stop
t122        0         ,flag 4
t123        0         ,storage for read check
end .
```