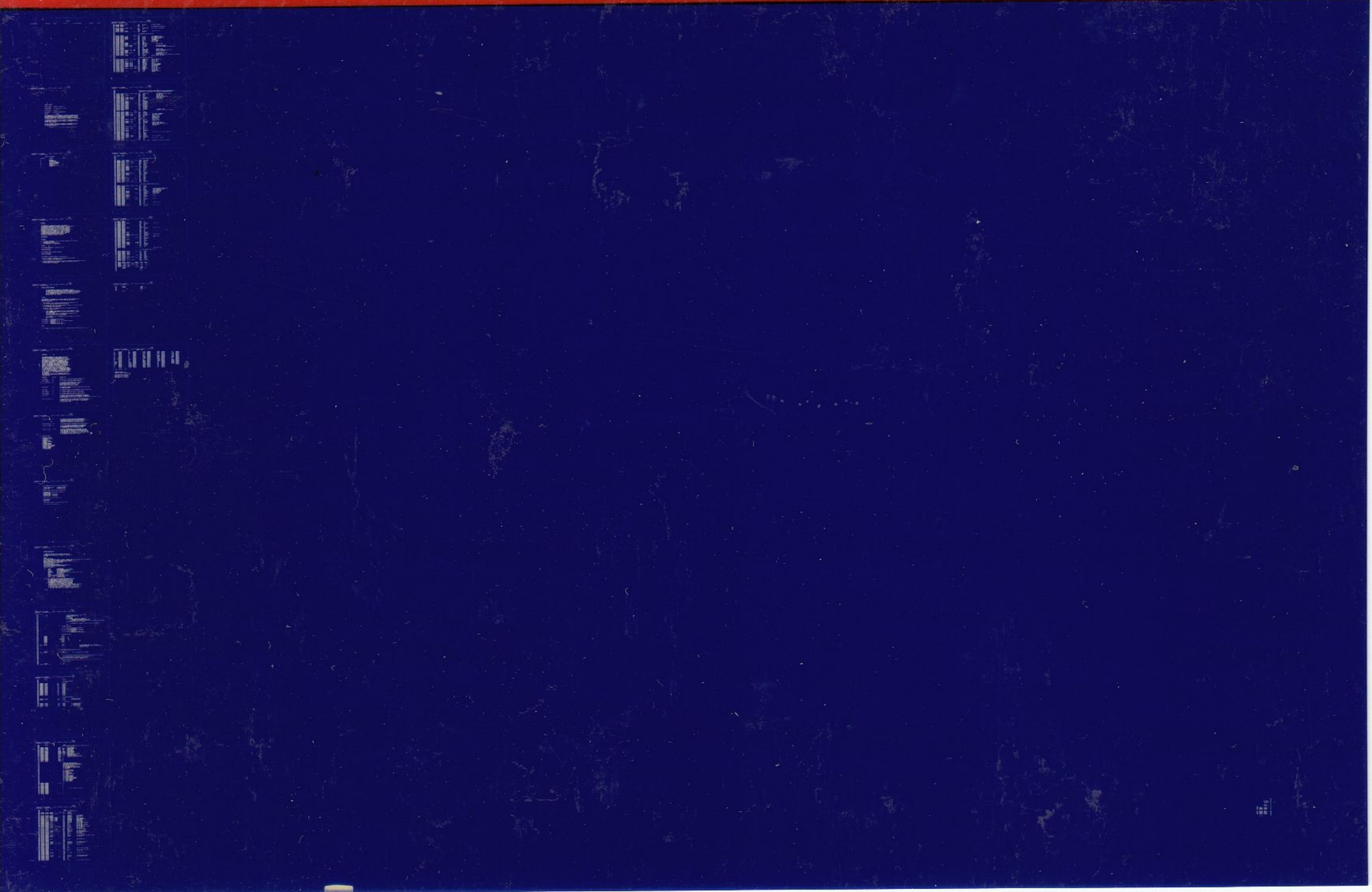


# TE16

UTILITY DRIVER  
MD-11-DZTEF-A

EP-DZTEF-A-DL-A  
COPYRIGHT © 1977  
FICHE 1 OF 1

JUN 1977  
**digital**  
MADE IN USA



B01

EOF1DZTEERSE0

00010000

770608

PDP10 411

5:H0R1DZTEFASE0

00010000

770608

1

.REM X

## IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTEF-A-D  
PRODUCT NAME: TUI6/TE16 UTILITY DRIVER  
DATE CREATED: 21 APRIL 77  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: J. G. ADAMS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMEN CORPORATION

;TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	1
2.	REQUIREMENTS	1
3.	LOADING PROCEDURE	1
4.	STARTING PROCEDURE	1
5.	CONSOLE SWITCHES	1
6.	OPERATION	2
7.	PROGRAM DESCRIPTION	5
8.	LISTING	

1. ABSTRACT

THIS PROGRAM IS INTENDED AS A BRUTE FORCE ROUTINE TO EXECUTE AN OPERATION OR SERIES OF OPERATIONS, CONTINUOUSLY REGARDLESS OF THE RESULTS OF THE OPERATION. BECAUSE OF THE COMPLEXITY OF THE TU16/TE16 MAG TAPE SYSTEM AS OPERATED ON THE MASSBUS, IT IS NOT ALWAYS POSSIBLE TO PROVIDE FOR EVERY CONTINGENCY IN THE NORMAL PROGRAMS. THEREFORE THIS UTILITY DRIVER WILL ALLOW AN OPERATOR TO EXECUTE ANYTHING DESIRED IN ANY ORDER. THERE ARE NO ERROR CHECKS OR PRINTOUTS MADE, AND ANY VARIATION FROM PRESET SEQUENCES AND VALUES ARE MADE BY CHANGING THE APPROPRIATE MEMORY LOCATIONS.

2. REQUIREMENTS2.1 HARDWARE:

- A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT HARDWARE SWITCH REGISTER.
- B. RM MASSBUS CONTROLLER
- C. TMO2/TMO3 MAG TAPE CONTROLLER
- D. AT LEAST ONE (1) TU16/TE16 SLAVE

2.2 STORAGE:

THIS PROGRAM REQUIRES AT LEAST 3K OF CORE

3. LOADING PROCEDURE:

USE STANDARD BINARY LOADING PROCEDURE

4. STARTING PROCEDURE

THE PROGRAM IS ALWAYS STARTED AT LOCATION 200 (8)

\*\*\*LOC. 176 (SWREG) IS DEFINED AS THE SOFTWARE SWITCH REGISTER  
(REFER TO SECTION 5 FOR MORE DETAIL)

\*\*\*IF THE SOFTWARE SWITCH REGISTER IS USED THE DIAGNOSTIC TYPES OUT THE FOLLOWING  
MESSAGE: SWR=XXXXXX NEW= (REFER TO SECTION 5 FOR OPERATOR OPTIONS)  
AT THE START OF THE PROGRAM.

5. CONSOLE SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

## CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G (<↑G>); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE "NEW=" HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
  - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED)  
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
  - B) IF A CONTROL U (<↑U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

SW15(100000): 1=STOP AFTER EACH OPERATION

0=PROCEED

SW14(040000): 1=STOP AT THE END OF THE OPERATION SEQUENCE

0=PROCEED

SW13(020000): 1=IGNORE END OF TAPE (EOT)

0=REWIND AT END OF TAPE (EOT)

5.1 HALT  
\*\*\*\*

\*\*\*TO CHANGE THE CONTENTS OF SWREG TYPE (<↑G>) BEFORE PRESSING CONTINUE AFTER A HALT.\*\*\*

6. OPERATION

THE PROGRAM OPERATION IS QUITE SIMPLE, BUT DOES REQUIRE THE OPERATOR TO HAVE KNOWLEDGE OF THE TU16/TE16 TAPE SYSTEM AS OPERATED ON THE RH MASSBUS CONTROLLER. THE OPERATOR MUST BE ABLE TO DECIDE WHICH SEQUENCE OF OPERATION IS REQUIRED, AND WHAT VALUES TO ASSIGN TO THE VARIOUS PARAMETERS REQUIRED TO EXECUTE THEM. THE OPERATION SEQUENCE IS SET UP BY LOADING A TABLE WITH THE FUNCTION CODES OF THE DESIRED OPERATIONS AND SETTING THE NUMBER OF OPERATIONS IN A COUNTER. THE PROGRAM IS SET UP TO DO A WRITE OF TEN (8) WORDS OF ALL ONES DATA TO SLAVE ZERO (0) ON DRIVE ZERO (0) IN PE (1600 BPI) WITH A NINE TRACK NORMAL DATA FORMAT. THE DATA ADDRESS IS 3000 (8). THE OPERATION SEQUENCE IS SET TO DO A SINGLE WRITE. IF LOADED AND STARTED AT 200 (8) WITH NO CHANGES MADE AND SWITCH 14 AND 15 SET TO A ZERO (0), THIS OPERATION WILL BE EXECUTED CONTINUOUSLY.

THE FOLLOWING IS THE LIST OF PARAMETERS WHICH MAY BE VARIED AND A DESCRIPTION OF EACH ALONG WITH THEIR CORE LOCATION:

PARAMETER	LOCATION	DESCRIPTION
RH ADDRESS	600	ADDRESS OF RH (THE FIRST REGISTER ADDRESS: CS1)
DRIVE NUMBER	700	SET TO SELECT TMD2/TMD3 NUMBER ADDRESS 0-7
UNIT DESCRIPTION	702	SET SELECTED SLAVE NUMBER (0-7) IN BITS 0,1,2 SELECT PARITY IN BIT 3 (0=ODD 1=EVN) SELECT DATA FORMAT IN BITS 4,5,6,7 SELECT DENSITY IN BITS 8,9,10
FRAME COUNT	704	SET NUMBER OF FRAMES TO WRITE PER WORD COUNT AND FORMAT IN TWOS' COMPLIMENT
WORD COUNT	706	SET NUMBER OF WORDS TO BE TRANSFERRED IN TWOS' COMPLIMENT
READ ADDRESS	710	SET DESIRED ADDRESS FOR START OF READ BUFFER.
WRITE ADDRESS	712	SET DESIRED ADDRESS FOR START OF WRITE BUFFER.
READY DELAY	714	THIS DELAY VALUE IS USED BY THE PROGRAM TO ESTABLISH A MAXIMUM TIME TO AWAIT THE COMPLETION OF AN OPERATION BEFORE PROCEEDING TO THE NEXT. ** (DEFAULT IS APPROX 435 MS FOR PDP-11/20) **
READY MULTIPLIER	716	IF THE VALUE SET INTO 714 DOES NOT ALLOW ENOUGH TIME, INCREASE THE SIZE OF THE MULTIPLIER. EACH INCREMENT OF THE MULTIPLIER WILL CAUSE THE 714 DELAY TO BE EXECUTED THAT MANY MORE TIMES.

OPERATION DELAY	720	THIS DELAY IS USED TO ALLOW FOR SOME AMOUNT OF TIME BETWEEN THE EXECUTION OF EACH OPERATION. IT IS LOADED AND USED JUST AS IN THE READY DELAY(714) ** (DEFAULT IS APPROX 54 MS FOR POP-11/20) **
OPER MULTIPLIER	722	THIS IS USED JUST AS THE READY DELAY MULTIPLIER(716)
OPERATION NUMBER	724	THIS IS THE NUMBER OF OPERATIONS TO BE PERFORMED IN A SEQUENCE AND SHOULD REFLECT THE NUMBERS OF OPERATIONS SET INTO THE OPERATION TABLE.
OPERATION TABLE	740-770	THIS TABLE (CONSISTING OF 15 LOCATIONS) IS TO BE LOADED WITH THE FUNCTION CODES FOR EACH OPERATION TO BE PERFORMED IN SEQUENCE. THE NUMBER OF ENTIRES MAY BE FROM ONE (1) TO FIFTEEN (15). MAKE SURE THAT THE NUMBER OF FUNCTION CODES SET IN THE TABLE IS REFLECTED BY THE NUMBER IN LOCATION 724 (OPNUM)

## 6.1 FUNCTION CODES

20=READ IN PRESET  
 02=REWIND-OFF LINE  
 06=REWIND  
 10=DRIVE CLEAR  
 26=WRITE TAPE MARK  
 24=ERASE  
 30=SPACE FORWARD  
 32=SPACE REVERSE  
 50=WRITE CHECK FORWARD  
 56=WRITE CHECK REVERSE  
 60=WRITE FORWARD  
 70=READ FORWARD  
 76=READ REVERSE



6.2 DATA FORMATS (BIT 7,6,5,4 OF UNIT DESCRIPTION)

14=NINE TRACK NORMAL: 2 FRAMES PER WORD  
15=CORE DUMP: 4 FRAMES PER WORD

6.3 DENSITY (BITS 10,9,8 OF UNIT DESCRIPTION)

4=1600 BPI:PE (PE USES ONLY ODD PARITY)  
3=800 BPI:NRZI  
2=800 BPI:NRZI (TU16 ONLY)  
1=556 BPI:NRZI (TU16 ONLY)  
0=200 BPI:NRZI (TU16 ONLY)

6.4 PARITY (BIT 3 OF UNIT DESCRIPTION)

1=EVEN PARITY  
0=ODD PARITY

6.5 SLAVE SELECT (BITS 2,1,0 OF UNIT DESCRIPTIONS)

SET TO DEVICE SLAVE ADDRESS (0-7)

7. PROGRAM DESCRIPTION

IN ORDER TO MAINTAIN THE CONTINUOUS EXECUTION OF THE OPERATIONS DESCRIBED THE PROGRAM IS ORGANIZED AS FOLLOWS:

START  
INITIALIZE THE RM  
SET UP TAPE PARAMETERS (DENSITY, PARITY, FORMAT: WORD COUNT, FRAME COUNT, BUS ADDRESS)  
SELECT DEVICE TO TEST (DRIVE NUMBER, SLAVE NUMBER)  
EXECUTE OPERATION (SET FUNCTION AND FROM OP TABLE AND SET GO=1)  
AWAIT END OF OPERATION (READY DELAY)  
STOP IF SWITCH 15=1  
DO OPERATION DELAY (OP DELAY)  
STOP IF LAST OPERATION IN SEQUENCE AND SWITCH 14=1  
POINT TO NEXT FUNCTION CODE IN OP TABLE  
JUMP BACK TO START

7.1 FLOW:    START:           HOUSEKEEPING  
          INIT:            CLEAR MASSBUS AND TMD2/TMD3  
          SET UP:          SET UP REQUIRED REGISTERS  
          EXECUTE:         SET FUNCTION AND GO=1  
          AWAIT END:       LOOP ON DRY=1 AS LONG AS ALLOWED BY READY DELAY  
          STOP:            IF SWITCH 15=1  
          DELAY:           PER OP DELAY  
          END OF RSEQUENCE? IF NOT JUMP TO START  
          STOP:            IF SWITCH 14=1  
          JUMP TO START    RESTART SEQUENCE

7.2 VARIATIONS: THERE ARE TWO VARIATIONS MADE FROM THIS FLOW.  
BOTH ARE CAUSED BY A PARTICULAR FUNCTION CODE.  
IF A READ REVERSE IS TO BE EXECUTED, THEN THE  
BUS ADDRESS IS INCREMENTED BY THE SIZE OF THE  
RECORD BECAUSE THE DATA IS LOADED INTO MEMORY  
IN REVERSE (I.E: HIGH ADDRESS TO LOW ADDRESS)  
THE SECOND VARIATION IS CAUSED BY A SPACE (FORWARD OR REVERSE)  
OPERATION AND IT IS THAT THE FRAME COUNTER IS SET TO A -1  
SO THAT ONLY ONE (1) RECORD IS SPACED OVER. IF YOU WISH  
TO SPACE OVER MORE THAN ONE (1) RECORD, SET LOCATION 1100 (8)  
TO THE TWO'S COMPLIMENT OF THE NUMBER OF RECORDS DESIRED.

8. LISTING

310

%

.LIST BIN,LOC,SEQ

311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366

```
.TITLE TU16/TE16 UTILITY DRIVER
:MAINDEC-11-DZTUE-E-D
:15 APR 77
:J. G. ADAMS
:REVISED APRIL 1976 BY S. CARPENTER
:1) SUPPORTS SOFTWARE SWITCH REGISTER
:2) SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
:REVISED APRIL 1977 BY J. G. ADAMS
```

```
:1)DOCUMENTATION CHANGES TO REFLECT TMO3/TE16 CAPABILITY
:ABS
```

;CONSOLE SWITCHES

```
;SW 15=1(100000) STOP ON EACH OPERATION
:0 CONTINUE
;SW 14=1(040000) STOP AT END OF SEQUENCE
:0 CONTINUE
;SW 13=1(020000) IGNORE END OF TAPE (EOT)
:0 REMIND AT END OF TAPE (EOT)
```

;REGISTER EQUIVES

```
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7
```

```
.=46
RESTART: 170 ;ALLOW RESTART WHEN <LF> IS PRESSED
;DURING CHANGING OF SWREG IF SOFTWARE SWITCH
;REGISTER IS USED.
```

;SOFTWARE SWITCH REGISTER\*\*\*\*\*

```
.=176
SWREG: 0 ;SOFTWARE SWITCH REGISTER
```

\*\*\*\*\*

```
;THIS PROGRAM SUPPORTS THE SOFTWARE SWITCH REGISTER LOC.176.
;REFER TO SECTION 5 OF DOCUMENT FOR DESCRIPTION
```

\*\*\*\*\*

;STARTING ADDRESS

```
.=200
JMP SETUP
```

```
000200 000200
000167 001110
```

367 000600 . =600

368  
369 ; TM02/TM03 REGISTERS

370				
371	000600	172440	CI:	172440
372	000602	172442	MC:	172442
373	000604	172444	BA:	172444
374	000606	172446	FC:	172446
375	000610	172450	CS:	172450
376	000612	172452	DS:	172452
377	000614	172454	ER:	172454
378	000616	172456	AS:	172456
379	000620	172460	CC:	172460
380	000622	172462	DB:	172462
381	000624	172464	MR:	172464
382	000626	172466	DT:	172466
383	000630	172470	SN:	172470
384	000632	172472	C2:	172472

385  
386 ; PROCESSOR ADDRESSES

387				
388	000634	177776	PSW:	177776 ; PROCESSOR STATUS
389	000636	177570	SWR:	177570 ; SWITCH REGISTER

390  
391 ; TTY REGISTERS

392				
393	000640	177560	TKS:	177560 ; TTY READER STATUS
394	000642	177562	TKB:	177562 ; TTY READ BUFFER
395	000644	177564	TPS:	177564 ; TTY PUNCH STATUS
396	000646	177566	TPB:	177566 ; TTY PUNCH BUFFER



```

452          001000          . = 1000
453          ; START OF PROGRAM*****
454
455 001000 012706 000500 START: MOV #500, SP
456 001004 012777 000340 177622 MOV #340, @PSW
457
458 001012 016700 177706          MOV OPNUM, R0          ; SET COUNTER
459 001016 012701 000740          MOV #OPTBL, R1        ; SET POINTER
460 001022 012777 000040 177560 A:  MOV #40, @CS          ; INIT
461 001030 016777 177644 177552          MOV DRVN, @CS        ; DRIVE NUMBER
462 001036 016777 177640 177566          MOV UDES, @C2        ; UNIT DESCRIPTION
463 001044 016777 177636 177530          MOV WCNT, @MC        ; WORD COUNT
464 001052 016777 177626 177526          MOV FCNT, @FC        ; FRAME COUNT
465 001060 012102          MOV (R1), R2          ; SET OP CODE
466 001062 022702 000030          CMP #30, R2          ; SEE IF SPACE FORWARD
467 001066 001403          BEQ AA              ; IF SO: BR
468 001070 022702 000032          CMP #32, R2          ; SEE IF SPACE REVERSE
469 001074 001003          BNE AD              ; IF NOT: BR
470 001076 012777 177777 177502 AA:  MOV #-1, @FC          ; SET TO SPACE ONE RECORD
471 001104 022702 000060          AD:  CMP #60, R2          ; SEE IF READ OP
472 001110 103404          BLO A1              ; IF SO: BR
473 001112 016777 177574 177464          MOV WADDR, @BA      ; SET WRITE ADDRESS
474 001120 000413          BR A3
475 001122 016777 177562 177454 A1:  MOV RADDR, @BA      ; SET READ ADDRESS
476 001130 022702 000070          CMP #70, R2          ; SEE IF READ OPERATION
477 001134 001405          BEQ A3              ; IF SO: BR
478 001136 016703 177542          MOV FCNT, R3          ; GET FRAME COUNT
479 001142 005403          NEG R3
480 001144 060377 177434          ADD R3, @BA          ; SET BUS ADDRESS FOR READ REVERSE
481 001150 052702 000001          A3:  BIS #1, R2          ; SET GO BIT
482 001154 000240          NOP
483 001156 000240          NOP
484 001160 010277 177414          MOV R2, @C1          ; START OPERATION
485 001164 000240          NOP
486 001166 000240          NOP
487 001170 016704 177522          MOV ROYDX, R4          ; SET DELAY MULTIPLIER
488 001174 016703 177514          BO:  MOV ROYDLY, R3      ; SET READY DELAY
489 001200 032777 000200 177404 B:   BIT #200, @DS
490 001206 001005          BNE C                ; IF DRY: BR
491 001210 005303          DEC R3
492 001212 001372          BNE B
493 001214 005304          DEC R4
494 001216 001366          BNE B0              ; DELAY FOR DRIVE READY
495 001220 000240          NOP
496 001222 005777 177410          C:   TST @SWR          ; SEE IF STOP ON OPERATION
497 001226 100001          BPL D                ; IF NOT: BR
498 001230 000000          HALT
499 001232 004767 000302          D:   JSR PC, @CKSWR   ; CHECK FOR CNTL G
500 001236 000240          NOP
501 001240 000240          NOP
502 001242 016704 177454          EO:  MOV OPDX, R4          ; SET DELAY MULTIPLIER
503 001246 016703 177446          E:   MOV OPDLY, R3      ; SET OPERATION DELAY
504 001252 005303          DEC R3
505 001254 001376          BNE E
506 001256 005304          DEC R4
507 001260 001372          BNE E0              ; DELAY BETWEEN OPERATIONS

```

```

508 001262 004767 000152      JSR    PC,RWIND      ;GO SEE IF REWIND
509 001266 005300              DEC    RO
510 001270 001254              BNE    A              ;IF SEQUENCE NOT DONE: BR
511
512 001272 032777 040000 177336  BIT    #40000,2SMR   ;SEE IF HALT ON SEQUENCE
513 001300 001401              BEQ    IS
514 001302 000000              HALT
515 001304 004767 000230      1S:   JSR    PC,CKSMR   ;CHECK FOR CNTL G
516 001310 000167 177464              JMP
517
518                                ;RM REGISTER SETUP*****
519
520                                SETUP:  NOP
521 001314 000240              MOV    C1,R1        ;GET ADDRESS OF CS1
522 001316 016701 177256      MOV    #15,RO       ;SET NUMBER OF REGISTERS
523 001322 012700 000015      MOV    #WC,R2       ;GET FIRST ADDRESS
524 001326 012702 000602      MOV    #2,R1        ;INCREMENT
525 001332 062701 000002      SETA:  ADD    R1,(R2)+ ;LOAD ADDRESS
526 001336 010122              MOV    RO           ;SEE IF DONE
527 001340 005300              DEC    RO           ;IF NOT: BR
528 001342 001373              BNE    SETA
529 001344 012706 000500      SUSMR: MOV    #500,SP     ;SAVE VECTORS
530 001350 013746 000006      MOV    #6,-(SP)
531 001354 013746 000004      MOV    #4,-(SP)
532 001360 012737 001400 000004  MOV    #15,#4       ;SET UP FOR TIMEOUT
533 001366 022777 177777 177242  CMP    #1,2SMR     ;REFERENCE HARDWARE SWITCH REGISTER
534 001374 001402              BEQ    2S
535 001376 000404              BR     3S
536 001400 022626              1S:  CMP    (SP)+,(SP)+ ;ADJUST STACK
537 001402 012767 000176 177226  2S:  MOV    #SMREG,SMR ;POINT TO SOFTWARE SWITCH REG
538 001410 012637 000004      3S:  MOV    (SP)+,#4    ;RESTORE VECTORS
539 001414 012637 000006      MOV    (SP)+,#6
540 001420 023727 000636 000176  CMP    #SMR,#SMREG ;IS SOFTWARE REG USED
541 001426 001002              BNE    GO           ;BRANCH IF NO
542 001430 004767 000156      JSR    PC,CNTLU    ;ALLOW SOFTWARE SWITCH REGISTER TO BE CHANGED
543 001434 000167 177340      GO:   JMP    START      ;ELSE GO START EXECUTION
544
545                                ;REWIND FROM EOT (PER SW13)
546 001440 032777 020000 177170  RWIND: BIT    #20000,2SMR  ;SEE IF IGNORE EOT
547 001446 001033              BNE    RWINDX      ;IF SO: BR
548 001450 032777 002000 177134  BIT    #2000,2DS    ;SEE IF AT EOT
549 001456 001427              BEQ    RWINDX      ;IF NOT: BR
550 001460 012777 000040 177122  MOV    #40,2CS     ;INIT
551 001466 016777 177206 177114  MOV    DRVN,2CS    ;SET DRIVE NUMBER
552 001474 016777 177202 177130  MOV    UDES,2C2    ;SET SLAVE NUMBER
553 001502 012777 000007 177070  MOV    #7,2C1      ;START REWIND
554 001510 032777 000200 177074  RWINDA: BIT    #200,2DS ;SEE IF DRY
555 001516 001774              BEQ    RWINDA     ;IF NOT: BR
556 001520 032777 020000 177064  RWINDB: BIT    #20000,2DS ;SEE IF PIP RESET
557 001526 001374              BNE    RWINDB     ;IF NOT: BR
558 001530 005726              TST   (SP)+       ;RESET STACK
559 001532 000167 177242      JMP    START      ;RESTART SEQUENCE
560 001536 000207      RWINDX: RTS    PC ;RETURN

```

```

561
562
563
564
565 001540 022767 000176 177070 CKSWR:  CMP      #SWREG,SWR      ;SOFTWARE SWITCH REG PRESENT
566 001546 001041          BNE      OUT      ;NO, GET OUT
567 001550 105777 177064  TSTB    #TKS      ;YES, WAIT FOR
568 001554 100036          BPL      OUT      ;READY, GET CHARACTER
569 001556 017767 177060 177142  MOV     #TKB,TIB  ;AND STRIP OFF
570 001564 042767 177600 177134  BIC     #177600,TIB ;THE GARBAGE
571 001572 022767 000007 177126  CMP     #7,TIB    ;IS IT A <IG>
572 001600 001024          BNE      OUT
573 001602 012704 002512  MOV     #SCNTG,R4
574 001606 004767 000242  JSR    PC,TTOUT
575 001612 012704 002516  CNTLU:  MOV     #SWR,R4
576 001616 004767 000232  JSR    PC,TTOUT
577 001622 017703 177010  MOV     #SWR,R3
578 001626 004767 000354  JSR    PC,CTPE
579 001632 012704 002525  MOV     #SWNEW,R4
580 001636 004767 000212  JSR    PC,TTOUT
581 001642 005037 000736  CLR     #TEMPST
582 001646 004767 000002  JSR    PC,SREAD
583 001652 000207          OUT:    RTS
584
585 001654 005067 177056  SREAD:  CLR     TEMPST
586 001660 012767 000007 177044  MOV     #7,COUNT
587 001666 004767 000546  1S:    JSR    PC,TTIN  ;GO READ A CHARACTER
588 001672 042767 177600 177026  BIC     #177600,TIB ;STRIP OFF GARBAGE
589 001700 122767 000025 177020  CMPB   #25,TIB
590 001706 001002          BNE     2S
591 001710 005726          TST    (SP)+
592 001712 000737          BR     CNTLU
593 001714 122767 000015 177004  2S:    CMPB   #15,TIB
594 001722 001013          BNE     4S
595 001724 012767 000200 177002  MOV     #200,RDSW
596 001732 004767 000150  JSR    PC,TCRLF
597 001736 022767 000007 176766  CMP     #7,COUNT
598 001744 001037          BNE     7S
599 001746 005726          8S:    TST    (SP)+
600 001750 000740          BR     OUT
601 001752 122767 000060 176746  4S:    CMPB   #60,TIB
602 001760 003004          BGT     5S
603 001762 122767 000067 176736  CMPB   #67,TIB
604 001770 003005          BGT     6S
605 001772 012704 002535  5S:    MOV     #SQUEST,R4
606 001776 004767 000052  JSR    PC,TTOUT
607 002002 000742          BR     3S
608 002004 006367 176726  6S:    ASL    TEMPST
609 002010 006367 176722  ASL    TEMPST
610 002014 006367 176716  ASL    TEMPST
611 002020 142767 000060 176700  BICB   #60,TIB  ;GET NITTY-GRITTY
612 002026 156767 176674 176702  BICB   TIB,TEMPST
613 002034 005367 176672  DEC    COUNT
614 002040 001754          BEQ    5S
615 002042 000711          BR     1S
616 002044 016777 176666 176564  7S:    MOV     TEMPST,#SWR ;CHANGE SWITCH REGISTER CONTENTS

```



```

617 002052 000735 BR BS
618
619
620 ;TTY OUTPUT SUBROUTINE*****
621
622 002054 112467 176650 TTOUT: MOVB (R4)+,TOB
623 002050 122767 000043 176642 CHPB #43,TOB
624 002056 001446 BEQ TEX
625 002070 122767 000045 176632 CHPB #45,TOB
626 002076 001403 BEQ TCRLF
627 002100 004767 000064 JSR PC,TOG
628 002104 000763 BR TTOUT
629 002106 112767 000015 176614 TCRLF: MOVB #15,TOB
630 002114 004767 000050 JSR PC,TOG
631 002120 012703 000004 MOV #4,R3
632 002124 005067 176600 TCRLFA: CLR TOB
633 002130 004767 000034 JSR PC,TOG
634 002134 005303 DEC R3
635 002136 001372 BNE TCRLFA ;DO FILLERS
636 002140 112767 000012 176562 MOVB #12,TOB
637 002146 004767 000016 JSR PC,TOG
638 002152 105767 176556 TSTB RDSW
639 002156 100401 BMI IS
640 002160 000735 BR TTOUT
641 002162 005067 176546 IS: CLR RDSW
642 002166 000406 BR TEX
643 002170 105777 176450 TOG: TSTB #TPS
644 002174 100375 BPL TOG
645 002176 116777 176526 176442 TEX: MOVB TOB,#TPB
646 002204 000207 RTS PC

```

```

647
648 ;OCTAL OUTPUT SUBROUTINE*****
649
650 002206 012767 000001 000222 OCTPE: MOV #1,OFL
651 002214 010304 MOV R3,R4
652 002216 000410 BR OCTPO
653 002220 005067 000212 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
654 002224 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
655 002226 001004 BNE OCTPO ;IF NOT ZERO: BR
656 002230 004767 000162 JSR PC,OCTPG1 ;ELSE PRINT ZERO
657 002234 000167 000120 JMP OCTP3 ;SPACE AND EXIT
658 002240 032704 100000 OCTPO: BIT #100000,R4 ;SEE IF MSD = 1
659 002244 001406 BEQ OCTP1 ;IF NOT: BR
660 002246 012704 000001 MOV #1,R4
661 002252 004767 000116 JSR PC,OCTPG ;PRINT 1
662 002256 000167 000006 JMP OCTP2
663 002262 005004 OCTP1: CLR R4 ;PRINT 0
664 002264 004767 000104 JSR PC,OCTPG
665 002270 010304 OCTP2: MOV R3,R4
666 002272 006004 ROR R4
667 002274 006004 ROR R4
668 002276 006004 ROR R4 ;POSITION DIGIT
669 002300 006004 ROR R4
670 002302 000304 SWAB R4
671 002304 004767 000064 JSR PC,OCTPG ;PRINT DIGIT 2
672 002310 010304 MOV R3,R4

```

673	002312	006004			ROR	R4		
674	002314	000304			SWAB	R4		
675	002316	004767	000052		JSR	PC, OCTPG		;PRINT DIGIT 3
676	002322	010304			MOV	R3, R4		
677	002324	006104			ROL	R4		
678	002326	006104			ROL	R4		
679	002330	000304			SWAB	R4		
680	002332	004767	000036		JSR	PC, OCTPG		;PRINT DIGIT 4
681	002336	010304			MOV	R3, R4		
682	002340	006004			ROR	R4		
683	002342	006004			ROR	R4		
684	002344	006004			ROR	R4		
685	002346	004767	000022		JSR	PC, OCTPG		
686	002352	010304			MOV	R3, R4		
687	002354	004767	000014		JSR	PC, OCTPG		;PRINT DIGIT 5
688	002360	012767	000240	176342	MOV	#240, TOB		
689	002366	004767	177576		JSR	PC, TOG		;PRINT SPACE
690	002372	000207			RTS	PC		;EXIT
691	002374	042704	177770		OCTPG:	BIC	#177770, R4	
692	002400	001004			BNE	OCTPGO		
693	002402	005767	000030		TST	OFL		
694	002406	001001			BNE	OCTPGO		
695	002410	000207			RTS	PC		
696	002412	005267	000020		OCTPGO:	INC	OFL	
697	002416	052704	000260		OCTPG1:	BIS	#260, R4	
698	002422	010467	176302		MOV	R4, TOB		
699	002426	004767	177536		JSR	PC, TOG		
700	002432	010304			MOV	R3, R4		
701	002434	000207			RTS	PC		
702	002436	000000			OFL:	0		;FIRST CHAR FLAG
703								
704								;TTY READ SUBROUTINE*****
705								
706	002440	005077	176174		TTIN:	CLR	@TKS	
707	002444	005077	176172			CLR	@TKB	
708	002450	005067	176252			CLR	TIB	
709	002454	005277	176160			INC	@TKS	
710	002460	105777	176154		TTIN1:	TSTB	@TKS	
711	002464	100375				BPL	TTIN1	
712	002466	017767	176150	176232		MOV	@TKB, TIB	
713	002474	105777	176144		TTIN2:	TSTB	@TPS	
714	002500	100375				BPL	TTIN2	
715	002502	116777	176220	176136		MOVB	TIB, @TPB	
716	002510	000207				RTS	PC	
717								
718	002512	057045	021507		SCNTG:	.ASCII	/%G#/	
719	002516	051445	051127	020075	SMSWR:	.ASCII	/%SMR= #/	
720	002524	043						
721	002525	040	047040	053505	SMNEW:	.ASCII	/ NEW= #/	
722	002532	020075	043					
723	002535	077	021445		SQUEST:	.ASCII	/?%#/	
724		004000				.=4000		
725		000100				.REPT	100	
726						0		
727						.ENDR		
728								

TU16/TE16 UTILITY DRIVER  
DZTEFA.P11 06-APR-77 09:55

MACY11 27(1006) 06-APR-77 09:56 PAGE 17

729	005000	.=5000	
730	000100	.REPT	100
731		i77777	
732		.ENDR	
733			
734	000001	.END	

A	001022	DB	000622	OCTP1	002262	RWIDX	001536	TPS	000644
AA	001076	DRVN	000700	OCTP2	002270	SETA	001332	TTIN	002440
AS	000616	DS	000612	OCTP3	002360	SETUP	001314	TTIN1	002460
AO	001104	DT	000626	OFL	002436	SN	000630	TTIN2	002474
A1	001122	E	001252	OPDLY	000720	START	001000	TTOUT	002054
A3	001150	ER	000614	OPDX	000722	SUSWR	001350	UDES	000702
B	001200	EO	001246	OPNUM	000724	SMR	000636	WADDR	000712
BA	000604	FC	000606	OPTBL	000740	SMREG	000176	MC	000602
BO	001174	FCNT	000704	OUT	001652	TCRLF	002106	MCNT	000706
C	001222	GO	001434	PSW	000634	TCRLFA	002124	SCHTG	002512
CC	000620	MR	000624	RADDR	000710	TEMPST	000736	SMNEW	002525
CKSMR	001540	OCTP	002220	RDSW	000734	TEX	002204	SMSWR	002516
CNTLU	001612	OCTPE	002206	RDYDLY	000714	TIB	000726	SQUEST	002535
COUNT	000732	OCTPE1	002224	RDYDX	000716	TKB	000642	SREAD	001654
CS	000610	OCTPG	002374	RESTAR	000046	TKS	000640	.	= 005200
C1	000600	OCTPG0	002412	RWID	001440	TOB	000730		
C2	000632	OCTPG1	002416	RWIDA	001510	TOG	002170		
D	001232	OCTPO	002240	RWIDB	001520	TPB	000646		

. ABS. 005200 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0DZTEFA,DZTEFA/SOL+DZTEFA.P11  
RUN-TIME: .6 1 0 SECONDS  
RUN-TIME RATIO: 32/1=17.3  
CORE USED: 5K (9 PAGES)