

1

.REM %

IDENTIFICATION

PRODUCT CODE: AC-E502A-MC
PRODUCT NAME: CZTUTA0 TM03/TU45 UTILITY DRIVER
DATE CREATED: 25 MAY 1978
MAINTAINER: CSS - NASHUA
AUTHOR: J. G. ADAMS/R. J. COLLINS

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) 1975,1976,1977, 1978 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 TU45 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. REQUIREMENTS

2.1 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) TU45 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 TU45 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 TMO2/TMO3 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=EVEN) 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 PDP-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 (TU45 ONLY)
1=556 BPI:NRZI (TU45 ONLY)
0=200 BPI:NRZI (TU45 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 TMO2/TMO3
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 TWOS' COMPLIMENT OF THE NUMBER OF RECORDS DESIRED.

8. LISTING

X

311

.LIST BIN,LOC,SEQ


```
312  
313 .TITLE TU45 UTILITY DRIVER  
314 :CZTUTAO  
315 :25 MAY 78  
316 :J. G. ADAMS/R. J. COLLINS  
317 :REVISED APRIL 1976 BY S. CARPENTER  
318 : 1) SUPPORTS SOFTWARE SWITCH REGISTER  
319 : 2) SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER  
320 :REVISED APRIL 1977 BY J. G. ADAMS  
321  
322 : 1) DOCUMENTATION CHANGES TO REFLECT TM03/TU45 CAPABILITY  
323 :ABS  
324  
325 ;CONSOLE SWITCHES  
326  
327 ;SW 15=1(100000) STOP ON EACH OPERATION  
328 : 0 CONTINUE  
329 ;SW 14=1(040000) STOP AT END OF SEQUENCE  
330 : 0 CONTINUE  
331 ;SW 13=1(020000) IGNORE END OF TAPE (EOT)  
332 : 0 REWIND AT END OF TAPE (EOT)  
333  
334 ;REGISTER EQUIVES  
335  
336 R0=X0  
337 R1=X1  
338 R2=X2  
339 R3=X3  
340 R4=X4  
341 R5=X5  
342 SP=X6  
343 PC=X7  
344  
345 .=46  
346 000046 000170 RESTART: 170 ;ALLOW RESTART WHEN <LF> IS PRESSED  
347 ;DURING CHANGING OF SWREG IF SOFTWARE SWITCH  
348 ;REGISTER IS USED.  
349  
350  
351 ;SOFTWARE SWITCH REGISTER*****  
352  
353 .=176  
354 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER  
355  
356  
357 ;*****  
358  
359 ;THIS PROGRAM SUPPORTS THE SOFTWARE SWITCH REGISTER LOC.176.  
360 ;REFER TO SECTION 5 OF DOCUMENT FOR DESCRIPTION  
361  
362 ;*****  
363  
364 ;STARTING ADDRESS  
365  
366 .=200  
367 000200 000167 001110 JMP SETUP
```

```
368          000600          --J00
369
370          ;TMO2/TMO3 REGISTERS
371
372 000600 172440          C1: 172440
373 000602 172442          WC: 172442
374 000604 172444          BA: 172444
375 000606 172446          FC: 172446
376 000610 172450          CS: 172450
377 000612 172452          DS: 172452
378 000614 172454          ER: 172454
379 000616 172456          AS: 172456
380 000620 172460          CC: 172460
381 000622 172462          DB: 172462
382 000624 172464          MR: 172464
383 000626 172466          DT: 172466
384 000630 172470          SN: 172470
385 000632 172472          C2: 172472
386
387          ;PROCESSOR ADDRESSES
388
389 000634 177776          PSW: 177776          ;PROCESSOR STATUS
390 000636 177570          SWR: 177570          ;SWITCH REGISTER
391
392          ;TTY REGISTERS
393
394 000640 177560          TKS: 177560          ;TTY READER STATUS
395 000642 177562          TKB: 177562          ;TTY READ BUFFER
396 000644 177564          TPS: 177564          ;TTY PUNCH STATUS
397 000646 177566          TPB: 177566          ;TTY PUNCH BUFFER
```

398 000700

.=700

;SET PARAMETERS DESIRED FOR UNIT UNDER TEST*****

399

400

401 000700 000000

DRVN: 0 ;DRIVE NUMBER

402 000702 002300

UDES: 2300 ;SLAVE DESCRIPTION

403 000704 177760

FCNT: -20 ;FRAME COUNT

404 000706 177770

WCNT: -10 ;WORD COUNT

405 000710 004000

RADDR: 4000 ;READ ADDRESS

406 000712 005000

WADDR: 5000 ;WRITE ADDRESS

407 000714 100000

RDYDLY: 100000 ;READY DELAY

408 000716 000001

RDYDX: 1 ;READY DELAY MULTIPLIER

409 000720 010000

OPDLY: 10000 ;OPERATION DELAY

410 000722 000001

OPDX: 1 ;OPERATION DELAY MULTIPLIER

411 000724 000001

OPNUM: 1 ;NUMBER OF OPERATION (1 TO 15)

412 000726 000000

TIB: 0

413 000730 000000

TOB: 0

414 000732 000000

COUNT: 0

415 000734 000000

RDSW: 0

416 000736 000000

TEMPST: 0

417

;OPERATION TABLE*****

418

;ENTER OPERATION SEQUENCE DESIRED.

419

;MUST HAVE AT LEAST 1 OPERATION, AND

420

;MAY HAVE UP TO 15(8).

421

;SET THE OPERATION COUNTER EQUAL

422

;TO THE NUMBER OF OPERATIONS IN

423

;THE SEQUENCE.

424

425

;

426

;20 = READ IN PRESET

427

;02 = REWIND-OFFLINE

428

;06 = REWIND

429

;10 = DRIVE CLEAR

430

;26 = WRITE TAPE MARK

431

;24 = ERASE

432

;30 = SPACE FORWARD

433

;32 = SPACE REVERSE

434

;50 = WRITE CHECK FORWARD

435

;56 = WRITE CHECK REVERSE

436

;60 = WRITE FORWARD

437

;70 = READ FORWARD

438

;76 = READ REVERSE

439

440 000740 000060

OPTBL: 60

441 000742 000000

0

442 000744 000000

0

443 000746 000000

0

444 000750 000000

0

445 000752 000000

0

446 000754 000000

0

;FILL WITH OPERATION SEQUENCE

447 000756 000000

0

448 000760 000000

0

449 000762 000000

0

450 000764 000000

0

451 000766 000000

0

452 000770 000000

0

```

453          001000          . = 1000
454          ; START OF PROGRAM*****
455
456 001000 012706 000500 START: MOV #500,SP
457 001004 012777 000340 177622 MOV #340,@PSW
458
459 001012 016700 177706 MOV OPNUM,R0 ; SET COUNTER
460 001016 012701 000740 MOV #OPTBL,R1 ; SET POINTER
461 001022 012777 000040 177560 A: MOV #40,@CS ; INIT
462 001030 016777 177644 177552 MOV DRVN,@CS ; DRIVE NUMBER
463 001036 016777 177640 177566 MOV UDES,@C2 ; UNIT DESCRIPTION
464 001044 016777 177636 177530 MOV WCNT,@WC ; WORD COUNT
465 001052 016777 177626 177526 MOV FCNT,@FC ; FRAME COUNT
466 001060 012102 MOV (R1)+,R2 ; SET OP CODE
467 001062 022702 000030 CMP #30,R2 ; SEE IF SPACE FORWARD
468 001066 001403 BEQ AA ; IF SO: BR
469 001070 022702 000032 CMP #32,R2 ; SEE IF SPACE REVERSE
470 001074 001003 BNE A0 ; IF NOT: BR
471 001076 012777 177777 177502 AA: MOV #-1,@FC ; SET TO SPACE ONE RECORD
472 001104 022702 000060 A0: CMP #60,R2 ; SEE IF READ OP
473 001110 103404 BLO A1 ; IF SO: BR
474 001112 016777 177574 177464 MOV WADDR,@BA ; SET WRITE ADDRESS
475 001120 000413 BR A3
476 001122 016777 177562 177454 A1: MOV RADDR,@BA ; SET READ ADDRESS
477 001130 022702 000070 CMP #70,R2 ; SEE IF READ OPERATION
478 001134 001405 BEQ A3 ; IF SO: BR
479 001136 016703 177542 MOV FCNT,R3 ; GET FRAME COUNT
480 001142 005403 NEG R3
481 001144 060377 177434 ADD R3,@BA ; SET BUS ADDRESS FOR READ REVERSE
482 001150 052702 000001 A3: BIS #1,R2 ; SET GO BIT
483 001154 000240 NOP
484 001156 000240 NOP
485 001160 010277 177414 MOV R2,@C1 ; START OPERATION
486 001164 000240 NOP
487 001166 000240 NOP
488 001170 016704 177522 MOV RDYDX,R4 ; SET DELAY MULTIPLIER
489 001174 016703 177514 B0: MOV RDYDLY,R3 ; SET READY DELAY
490 001200 032777 000200 177404 B: BIT #200,@DS
491 001206 001005 BNE C ; IF DRY: BR
492 001210 005303 DEC R3
493 001212 001372 BNE B
494 001214 005304 DEC R4
495 001216 001366 BNE B0 ; DELAY FOR DRIVE READY
496 001220 000240 NOP
497 001222 005777 177410 C: TST @SWR ; SEE IF STOP ON OPERATION
498 001226 100001 BPL D ; IF NOT: BR
499 001230 000000 HALT
500 001232 004767 000302 D: JSR PC,CKSWR ; CHECK FOR CNTL G
501 001236 000240 NOP
502 001240 000240 NOP
503 001242 016704 177454 E0: MOV OPDX,R4 ; SET DELAY MULTIPLIER
504 001246 016703 177446 E: MOV OPDLY,R3 ; SET OPERATION DELAY
505 001252 005303 DEC R3
506 001254 001376 BNE E
507 001256 005304 DEC R4
508 001260 001372 BNE E0 ; DELAY BETWEEN OPERATIONS
    
```

```

509 001262 004767 000152 JSR PC,RWMD ;GO SEE IF REWIND
510 001266 005300 DEC RO ;
511 001270 001254 BNE A ;IF SEQUENCE NOT DONE: BR
512
513 001272 032777 040000 177336 BIT #40000,@SWR ;SEE IF HALT ON SEQUENCE
514 001300 001401 BEQ 1$ ;
515 001302 000000 HALT ;
516 001304 004767 000230 1$: JSR PC,CKSWR ;CHECK FOR CNTL G
517 001310 000167 177464 JMP START ;
518
519 ;RH REGISTER SETUP*****
520
521 001314 000240 SETUP: NOP ;
522 001316 016701 177256 MOV C1,R1 ;GET ADDRESS OF C1
523 001322 012700 000015 MOV #15,RO ;SET NUMBER OF REGISTERS
524 001326 012702 000602 MOV #VC,R2 ;GET FIRST ADDRESS
525 001332 062701 000002 SETA: ADD #2,R1 ;INCREMENT
526 001336 010122 MOV R1,(R2)+ ;LOAD ADDRESS
527 001340 005300 DEC RO ;SEE IF DONE
528 001342 001373 BNE SETA ;IF NOT: BR
529 001344 012706 000500 MOV #500,SP ;
530 001350 013746 000006 SUSWR: MOV @#6,-(SP) ;SAVE VECTORS
531 001354 013746 000004 MOV @#4,-(SP) ;
532 001360 012737 001400 000004 MOV #1$,@#4 ;SET UP FOR TIMEOUT
533 001366 022777 177242 CMP #1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
534 001374 001402 BEQ 2$ ;
535 001376 000404 BR 3$ ;
536 001400 022626 1$: CMP (SP)+,(SP)+ ;ADJUST STACK
537 001402 012767 000176 177226 2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
538 001410 012637 000004 3$: MOV (SP)+,@#4 ;RESTORE VECTORS
539 001414 012637 000006 MOV (SP)+,@#6 ;
540 001420 023727 000636 000176 CMP @#SWR,#SWREG ;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
547 001440 032777 020000 177170 RWMD: BIT #20000,@SWR ;SEE IF IGNORE EOT
548 001446 001033 BNE RWMDX ;IF SO: BR
549 001450 032777 002000 177134 BIT #2000,@DS ;SEE IF AT EOT
550 001456 001427 BEQ RWMDX ;IF NOT: BR
551 001460 012777 000040 177122 MOV #40,@CS ;INIT
552 001466 016777 177206 177114 MOV DRVN,@CS ;SET DRIVE NUMBER
553 001474 016777 177202 177130 MOV UDES,@C2 ;SET SLAVE NUMBER
554 001502 012777 000007 177070 MOV #7,@C1 ;START REWIND
555 001510 032777 000200 177074 RWMDA: BIT #200,@DS ;SEE IF DRY
556 001516 001774 BEQ RWMDA ;IF NOT: BR
557 001520 032777 020000 177064 RWMDB: BIT #20000,@DS ;SEE IF PIP RESET
558 001526 001374 BNE RWMDB ;IF NOT: BR
559 001530 005726 TST (SP)+ ;RESET STACK
560 001532 000167 177242 JMP START ;RESTART SEQUENCE
561 001536 000207 RWMDX: RTS PC ;RETURN
    
```

```

562
563
564           ;CKSWR ROUTINE THAT ALLOWS THE LOADING OF LOC.176, SWREG*****
565           ;FROM THE TTY AT SELECTED POINTS WITHIN THE PROGRAM*****
566 001540 022767 000176 177070 CKSWR: CMP      #SWREG,SWR      ;SOFTWARE SWITCH REG PRESENT
567 001546 001041                BNE      OUT          ;NO, GET OUT
568 001550 105777 177064        TSTB     @TKS          ;YES, WAIT FOR
569 001554 100036                BPL      OUT          ;READY, GET CHARACTER
570 001556 017767 177060 177142 MOV      @TKB,TIB      ;AND STRIP OFF
571 001564 042767 177600 177134 BIC      #177600,TIB   ;THE GARBAGE
572 001572 022767 000007 177126 CMP      #7,TIB        ;IS IT A < G>
573 001600 001024                BNE      OUT
574 001602 012704 002512        MOV      #SCNTG,R4
575 001606 004767 000242        JSR      PC,TTOUT
576 001612 012704 002516        CNTLU:  MOV      #SMSWR,R4
577 001616 004767 000232        JSR      PC,TTOUT
578 001622 017703 177010        MOV      @SWR,R3
579 001626 004767 000354        JSR      PC,OCTPE
580 001632 012704 002525        MOV      #SMNEW,R4
581 001636 004767 000212        JSR      PC,TTOUT
582 001642 005037 000736        CLR      @TEMPST
583 001646 004767 000002        JSR      PC,$READ
584 001652 000207                OUT:    RTS          ;GO READ A LINE
585                                     ;RETURN TO MAIN BODY OF PROGRAM
586 001654 005067 177056        $READ: CLR      TEMPST
587 001660 012767 000007 177044 MOV      #7,COUNT
588 001666 004767 000546        1$:    JSR      PC,TTIN      ;GO READ A CHARACTER
589 001672 042767 177600 177026 BIC      #177600,TIB   ;STRIP OFF GARBAGE
590 001700 122767 000025 177020 CMPB     #25,TIB
591 001706 001002                BNE      2$          ;IS IT A U?
592 001710 005726                3$:    TST      (SP)+      ;BRANCH IF NOT
593 001712 000737                ;POP THE STACK
594 001714 122767 000015 177004 2$:    BR      CNTLU      ;START OVER
595 001722 001013                CMPB     #15,TIB
596 001724 012767 000200 177002 BNE      4$          ;IS IT A <CR>?
597 001732 004767 000150        MOV      #200,RDSW    ;BRANCH IF NOT
598 001736 022767 000007 176766 JSR      PC,TCRLF
599 001744 001037                ;ECHO IT WITH <LF>
600 001746 005726                8$:    CMP      #7,COUNT  ;WAS IT FIRST CHARACTER
601 001750 000740                BNE      7$          ;CHANGE SWR IF NOT FIRST ONE
602 001752 122767 000060 176746 4$:    TST      (SP)+      ;POP THE STACK
603 001760 003004                BR      OUT          ;GET OUT
604 001762 122767 000067 176736 CMPB     #60,TIB
605 001770 003005                BGT      5$
606 001772 012704 002535        5$:    CMPB     #67,TIB
607 001776 004767 000052        BGT      6$
608 002002 000742                MOV      #SQUEST,R4
609 002004 006367 176726        6$:    JSR      PC,TTOUT
610 002010 006367 176722        BR      3$          ;START OVER IF NOT LEGAL CHARACTER
611 002014 006367 176716        ASL      TEMPST
612 002020 142767 000060 176700 7$:    ASL      TEMPST
613 002026 156767 176674 176702 ASL      TEMPST
614 002034 005367 176672        BICB    #6C,TIB      ;GET NITTY-GRITTY
615 002040 001754                BISB    TIB,TEMPST
616 002042 000711                DEC      COUNT
617 002044 016777 176666 176564 7$:    BEQ     5$          ;ONLY WANT 6 DIGITS
        BR      1$
        MOV     TEMPST,@SWR ;CHANGE SWITCH REGISTER CONTENTS
    
```

```

618 002052 000735 BR 8$
619
620
621 ;TTY OUTPUT SUBROUTINE*****
622
623 002054 112467 176650 TTOUT: MOVB (R4)+,TOB
624 002060 122767 000043 176642 CMPB #43,TOB
625 002066 001446 BEQ TEX
626 002070 122767 000045 176632 CMPB #45,TOB
627 002076 001403 BEQ TCRLF
628 002100 004767 000064 JSR PC,TOG
629 002104 000763 BR TTOUT
630 002106 112767 000015 176614 TCRLF: MOVB #15,TOB
631 002114 004767 000050 JSR PC,TOG
632 002120 012703 000004 MOV #4,R3
633 002124 005067 176600 TCRLFA: CLR TOB
634 002130 004767 000034 JSR PC,TOG
635 002134 005303 DEC R3
636 002136 001372 BNE TCRLFA ;DO FILLERS
637 002140 112767 000012 176562 MOVB #12,TOB
638 002146 004767 000016 JSR PC,TOG
639 002152 105767 176556 TSTB RDSW
640 002156 100401 BMI 1$
641 002160 000735 BR TTOUT
642 002162 005067 176546 1$: CLR RDSW
643 002166 000406 BR TEX
644 002170 105777 176450 TOG: TSTB @TPS
645 002174 100375 BPL TOG
646 002176 116777 176526 176442 MOVB TOB,@TPB
647 002204 000207 TEX: RTS PC
    
```

```

648
649 ;OCTAL OUTPUT SUBROUTINE*****
650
651 002206 012767 000001 000222 OCTPE: MOV #1,OFL
652 002214 010304 MOV R3,R4
653 002216 000410 BR OCTPO
654 002220 005067 000212 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
655 002224 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
656 002226 001004 BNE OCTPO ;IF NOT ZERO: BR
657 002230 004767 000162 JSR PC,OCTPG1 ;ELSE PRINT ZERO
658 002234 000167 000120 JMP OCTP3 ;SPACE AND EXIT
659 002238 032704 100000 OCTPO: BIT #100000,R4 ;SEE IF MSD = 1
660 002244 001406 BEQ OCTP1 ;IF NOT: BR
661 002246 012704 000001 MOV #1,R4
662 002252 004767 000116 JSR PC,OCTPG ;PRINT 1
663 002256 000167 000006 JMP OCTP2
664 002262 005004 OCTP1: CLR R4
665 002264 004767 000104 JSR PC,OCTPG ;PRINT 0
666 002270 010304 OCTP2: MOV R3,R4
667 002272 006004 ROR R4
668 002274 006004 ROR R4
669 002276 006004 ROR R4 ;POSITION DIGIT
670 002300 006004 ROR R4
671 002302 000304 SWAB R4
672 002304 004767 000064 JSR PC,OCTPG ;PRINT DIGIT 2
673 002310 010304 MOV R3,R4
    
```

```

674 002312 006004 ROR R4
675 002314 000304 SWAB R4
676 002316 004767 000052 JSR PC,OCTPG ;PRINT DIGIT 3
677 002322 010304 MOV R3,R4
678 002324 006104 ROL R4
679 002326 006104 ROL R4
680 002330 000304 SWAB R4
681 002332 004767 000036 JSR PC,OCTPG ;PRINT DIGIT 4
682 002336 010304 MOV R3,R4
683 002340 006004 ROR R4
684 002342 006004 ROR R4
685 002344 006004 ROR R4
686 002346 004767 000022 JSR PC,OCTPG
687 002352 010304 MOV R3,R4
688 002354 004767 000014 JSR PC,OCTPG ;PRINT DIGIT 5
689 002360 012767 000240 176342 OCTP3: MOV #240,TOB
690 002366 004767 177576 JSR PC,TOG ;PRINT SPACE
691 002372 000207 RTS PC ;EXIT
692 002374 042704 177770 OCTPG: BIC #177770,R4
693 002400 001004 BNE OCTPG0
694 002402 005767 000030 TST OFL
695 002406 001001 BNE OCTPG0
696 002410 000207 RTS PC
697 002412 005267 000020 OCTPG0: INC OFL
698 002416 052704 000260 OCTPG1: BIS #260,R4
699 002422 010467 176302 MOV R4,TOB
700 002426 004767 177536 JSR PC,TOG
701 002432 010304 MOV R3,R4
702 002434 000207 RTS PC
703 002436 000000 OFL: 0 ;FIRST CHAR FLAG
704
705 ;TTY READ SUBROUTINE*****
706
707 002440 005077 176174 TTIN: CLR @TKS
708 002444 005077 176172 CLR @TKB
709 002450 005067 176252 CLR TIB
710 002454 005277 176160 INC @TKS
711 002460 105777 176154 TTIN1: TSTB @TKS
712 002464 100375 BPL TTIN1
713 002466 017767 176150 176232 MOV @TKB,TIB
714 002474 105777 176144 TTIN2: TSTB @TPS
715 002500 100375 BPL TTIN2
716 002502 116777 176220 176136 MOVB TIB,@TPB
717 002510 000207 RTS PC
718
719 002512 057045 021507 $CNTG: .ASCII /% G#,
720 002516 051445 051127 020075 $MSWR: .ASCII /%SWR= #/
721 002524 043
722 002525 040 047040 053505 $MNEW: .ASCII / NEW= #/
723 002532 020075 043
724 002535 077 021445 $QUEST: .ASCII /?%#/
725 004000
726 000100 .REPT 100
727 0
728 .ENDR
729
    
```


730	005000 -	.=5000	
731	000100	.REPT	100
732		177777	
733		.ENDR	
734			
735	000001	.END	

A	001022	DB	000622	OCTP1	002262	RWDX	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	SWR	000636	WADDR	000712
BA	000604	FC	000606	OPTBL	000740	SWREG	000176	WC	000602
BO	001174	FCNT	000704	OUT	001652	TCRLF	002106	WCNT	000706
C	001222	GO	001434	PSW	000634	TCRLFA	002124	SCNTG	002512
CC	000620	MR	000624	RADDR	000710	TEMPST	000736	SMNEW	002525
CKSWR	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	RWND	001440	TOB	000730		
C2	000632	OCTPG1	002416	RWDA	001510	TOG	002170		
D	001232	OCTP0	002240	RWDB	001520	TPB	000646		

. ABS. 005200 000

ERRORS DETECTED: 0

.CZTUTA.SEQ/SOL.CZTUTA.P11
 RUN-TIME: 37.3 SECONDS
 RUN-TIME RATIO: 23/11=2.0
 CORE USED: 14K (28 PAGES)