

The main body of the page contains a grid of 12 columns and 10 rows of small, illegible technical diagrams or tables. These appear to be diagnostic steps or component layouts for the printer. The text is too small to read, but the layout is organized into a structured grid.



.REM

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

IDENTIFICATION

PRODUCT CODE: AC-U022A-MC
PRODUCT NAME: CZLCPA0 LCPO1 PRTR DIAG
DATE : JANUARY 15, 1985
MAINTAINER: CSS PRODUCT GROUP DIAGNOSTIC

COPYRIGHT (C) 1985 BY DIGITAL EQUIPMENT CORPORATION

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

16- 646	PARAMETERS
17- 704	MACROS
18- 957	VARIABLES
19- 1006	SETUP SERIAL LINE PARAMETERS
20- 1042	
20- 1043	TEST # DESCRIPTION
20- 1044	-----
20- 1045	TEST 1 SERIAL LINE UNIT TEST
22- 1178	TEST 2 COLOR PRINTER 'SELF' TEST
23- 1260	TEST 3 PRINTER DISPLAY TEST
24- 1317	
24- 1318	END OF TEST SEQUENCE
25- 1351	MISC. SUBROUTINES
26- 1485	KEYBOARD INTERRUPT ROUTINE
27- 1620	LOCAL MESSAGES
28- 1689	SERIAL LINE SETUP ROUTINES
32- 2577	SERIAL LINE SETUP MESSAGES

38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
TABLE OF CONTENTS

CONTENTS

- 1.0 ABSTRACT
 - 1.1 MAINTENANCE HISTORY
- 2.0 REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 SOFTWARE
- 3.0 TEST OVERVIEW
 - 3.1 SECTION DESCRIPTIONS
 - 3.2 SERIAL LINE OPERATIONS
- 4.0 ASSUMPTIONS
- 5.0 OPERATING PROCEDURE
 - 5.1 STARTING ADDRESS OR ADDRESSES
 - 5.2 OPERATIONAL SWITCH SETTINGS - HARDWARE AND SOFTWARE
- 6.0 TEST DESCRIPTIONS
 - 6.1 TEST 1: SERIAL LINE TEST
 - 6.2 TEST 2: COLOR PRINTER SELF TEST
 - 6.3 TEST 3: PRINTER DISPLAY TEST

68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
ABSTRACT

1.0 ABSTRACT

THIS IS A PDP-11 DIAGNOSTIC WHICH IS DESIGNED TO FUNCTIONALLY EXERCISE AN LPC01 PRINTER WHILE ATTACHED TO A PDP-11 PROCESSOR.

THE DIAGNOSTIC CAUSES THE MICROCODE TO EXECUTE SPECIFIC TESTS. THE DIAGNOSTIC THEN MONITORS THE LCPO1 SERIAL LINE OUTPUT, TRACING TEST COMPLETION AND ERROR INDICATIONS.

CZLCP IS AN XXDP+ DIAGNOSTIC.

1.1 MAINTENANCE HISTORY

CZLCP IS A NEW PDP-11 DIAGNOSTIC.

AUTHOR: DIGITAL EQUIPMENT CORPORATION
COMPUTER SPECIAL SYSTEMS
HUDSON, NEW HAMPSHIRE

PREPARED BY:
DICE SYSTEMS, INC.
7 1/2 HARRIS ROAD
NASHUA, NEW HAMPSHIRE

EDIT HISTORY:

NEW VERSION

106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
REQUIREMENTS

2.0 REQUIREMENTS

2.1 EQUIPMENT

THIS DIAGNOSTIC WILL RUN ON ALL PDP 11 FAMILY COMPUTERS WHICH HAVE
EITHER A DL11 OR A DZ11 SERIAL LINE AND AN LCPO1 PRINTER.

2.2 STORAGE

THIS PROGRAM REQUIRES A PDP-11 SYSTEM WITH AT LEAST 28K WORDS OF
MEMORY.

2.3 SOFTWARE

THIS PROGRAM REQUIRES XXDP+ OPERATING SYSTEM SOFTWARE, VERSION 1.2 OR
LATER.

138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
TEST OVERVIEW

3.0 TEST OVERVIEW

3.1 SECTION DESCRIPTIONS

THIS DIAGNOSTIC CONSISTS OF ONE SECTION, CONTAINING THREE TESTS, AS FOLLOWS:

1. SERIAL LINE TEST
2. COLOR PRINTER SELF TEST
3. PRINTER DISPLAY TEST

3.2 SERIAL LINE OPERATIONS

THE PROGRAM HAS BEEN DESIGNED TO ALLOW THE DIAGNOSTIC TO TEST THE LCPO1 PRINTERS WHILE ATTACHED BY SERIAL LINE INTERFACES.

THE FOLLOWING DIALOGUE ALLOWS FOR SERIAL LINE SELECTION:

```
R CZLCP??  
CZLCP.BIN  
  
LCPO1 LINE PRINTER DIAGNOSTIC  
  
SERIAL LINE SELECTION MENU  
1 DL11 SERIAL LINE  
2 DZ11 SERIAL LINE  
TYPE MENU SELECTION <1>?
```

THE APPROPRIATE NUMBER (ONE OR TWO) SHOULD BE ENTERED BY THE OPERATOR. TYPING A CARRIAGE RETURN WILL RESULT IN SELECTION OF THE DEFAULT SERIAL LINE, WHICH IS THE DL11.

3.2.1 DL11 SUPPORT -

THE PROGRAM WILL SUPPORT DL11 OF THE FOLLOWING TYPE:

1. ALL UNIBUS DL11S
2. DLV11
3. DLV11-F

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
TEST OVERVIEW

191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238

THE PROGRAM WILL PROMPT FOR INDIVIDUAL SETUP PARAMETERS AS THEY ARE REQUIRED BY EACH INTERFACE.

3.2.2 DZ11 SUPPORT -

THE PROGRAM WILL SUPPORT BOTH DZ11 AND THE DZV11. THE PROGRAM WILL PROMPT THE OPERATOR FOR THE REQUIRED SETUP PARAMETERS, THESE INCLUDE THE FOLLOWING:

1. CSR ADDRESS
2. THE DZ LINE NUMBER BEING USED (0-7).
3. THE DZ'S BAUD RATE
4. THE NUMBER OF STOP BITS
5. NUMBER OF DATA BITS
6. WHETHER OR NOT PARITY IS BEING USED
7. IF PARITY IS USED, IS IT ODD OR EVEN

4.0 ASSUMPTIONS

THE ONBOARD MICROCODE DIAGNOSTICS HAVE COMPLETE RESPONSIBILITY FOR DEVICE TEST COVERAGE. THE DIAGNOSTIC ONLY VERIFIES THE INTERFACE AND REPORTS ERRORS DETECTED BY THE MICROCODE DIAGNOSTICS.

5.0 OPERATING PROCEDURE

5.1 STARTING ADDRESS OR ADDRESSES

THE INITIAL STARTING ADDRESS TO RUN THE ENTIRE LCPO1 DIAGNOSTIC IS LOCATION 200(8). THE RESTART ADDRESS IS 300 (8).

240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
OPERATING PROCEDURE

5.2 OPERATIONAL SWITCH SETTINGS - HARDWARE AND SOFTWARE

WHEN THE DIAGNOSTIC IS STARTED AT ADDRESS 200(8), IT WILL DETERMINE WHETHER OR NOT THE PROCESSOR HAS A HARDWARE (H/W) SWITCH REGISTER (SWR). IF THERE IS A H/W SWR, THE DIAGNOSTIC WILL USE BOTH THE HARDWARE REGISTER, AT LOCATION 177570, AND THE SOFTWARE (S/W) SWR LOCATED AT ADDRESS 176(8). IF A BIT IS SET IN EITHER REGISTER, EXECUTION WILL BE MODIFIED AS DESCRIBED IN THE 'SWITCH REGISTER BIT DEFINITIONS' SECTION OF THIS DOCUMENT.

5.2.1 CONTROL-G -

THE OPERATOR MAY CHANGE THESE REGISTER VALUES BY ENTERING A CONTROL-G AT THE CONSOLE TERMINAL. THE DIAGNOSTIC WILL PROMPT THE OPERATOR WITH THE MESSAGE:

H/W SWR = XXXXXX SWR = XXXXXX NEW SWR =

AFTER EXECUTION BEGINS, THE OPERATOR MAY CHANGE THE VALUE OF THE SWR, AT ANY TIME, BY ENTERING A CONTROL-G (G) AT THE CONSOLE.

IN RESPONSE TO THE PROMPTS, THE OPERATOR MAY ENTER UP TO SIX (6) OCTAL DIGITS. THE DIGITS MAY BE ANY COMBINATION OF :0,1,2,3,4,5,6,7, OR NO ENTRY AT ALL. ALL SWR VALUES ENTERED WILL BE TRUNCATED TO THE LOWER SIXTEEN (16) BITS. ENTERING MORE THAN SIX DIGITS OR A CHARACTER OTHER THAN A DIGIT RESULTS IN A "?" OUTPUT ON THE CONSOLE AND A REPEAT OF THE PROMPTING MESSAGE.

CARRIAGE RETURN (CR): ENTERS THE NEW SWR VALUE. IF NO DIGITS HAVE BEEN ENTERED, THE SWR VALUE REMAINS UNCHANGED.

5.2.2 CONTROL-U -

ERASES THE SWR VALUE BEING ENTERED. A CARRIAGE RETURN AND LINE FEED WILL BE OUTPUT AT THE CONSOLE. THE CORRECT SWR VALUE MAY THEN BE ENTERED.

5.2.3 CONTROL-H -

PRINTS THE HELP FILE ON THE CONSOLE TERMINAL. THE FOLLOWING INFORMATION IS DISPLAYED:

292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
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

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
OPERATING PROCEDURE

HELP SWITCH REGISTER BIT DEFINITION

(NOTE: <CTRL>G - ALLOWS CHANGE TO "SOFTWARE" SWITCH REG)
15, 14, 13, 12...2, 1, 0
I I I I I I I--LOOP ON SLU TEST 1
I I I I I I-----LOOP ON PRINTER "SELF" TEST
I I I I I-----LOOP ON PRINTER DISPLAY TEST
I I I I-----PAUSE ON ERROR, PAUSE AT END OF PASS
I I I-----INHIBIT ERROR REPORTS
I I-----INHIBIT TEST ERROR AND END OF PASS REPORTS
I-----LOOP ON ERROR (OTHERWISE CONTINUE)

ENTERING ANY CHARACTER BEFORE A CONTROL-G (G) HAS BEEN ENTERED WILL RESULT IN A "?" OUTPUT AT THE CONSOLE.

NOTE

IT IS POSSIBLE FOR THE DIAGNOSTIC TO OUTPUT MESSAGES AT THE CONSOLE BEFORE THE NEW SWR VALUE HAS BEEN ENTERED. SHOULD THIS HAPPEN, THE OPERATOR SHOULD ENTER A CONTROL-U (U) AND THEN ENTER THE CORRECT SWR VALUE.

5.2.4 CONTROL-C -

ENTRY OF A 'CONTROL-C' COMBINATION ABORTS TESTING AND RESTARTS IT AT LOCATION 200 (OCTAL).

5.2.5 SWITCH REGISTER BIT DEFINITIONS -

- BIT0 =1: LOOP ON SLU TEST #1.
- BIT1 =1: LOOP ON PRINTER SELF TEST #2
- BIT2 =1: LOOP ON THE PRINTER DISPLAY TEST #3
- BIT12=1: HALT ON ERROR AND HALT ON END OF PASS, THE OPERATOR MAY CHOOSE TO CONTINUE OR PROCEED BY ENTERINNG THOSE COMMANDS. THE DEFAULT EXECUTION IS TO CONTINUE AFTER AN ERROR OR AN EOP INDICATION IS ENCOUNTERED.
- BIT13=1: INHIBIT ERROR REPORTING.
- BIT14=1: TEST HEADER AND END OF PASS MESSAGES ARE NOT DISPLAYED.
- BIT15=1: WHEN AN ERROR IS ENCOUNTERED, LOOP ON ERROR, IF NO ERROR IS ENCOUNTERED EXECUTE TESTS CONTINUOUSLY.

345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
DEFAULT SECTION TEST DESCRIPTIONS

6.0 TEST DESCRIPTIONS

6.1 TEST 1: SERIAL LINE TEST

THIS TEST IS DESIGNED TO SHOW THE OPERATOR THAT THE SERIAL LINE SETUP HAS BEEN COMPLETED CORRECTLY. THIS WILL UNCOVER MOST SETUP ERRORS, INCLUDING BASIC SERIAL LINE SETUP ERRORS.

FAILURE OF THIS TEST USUALLY SIGNIFIES THAT THE DIAGNOSTIC IS WORKING WITH INCORRECT INFORMATION. FOR INSTANCE, THE CSR ADDRESS SPECIFIED MAY BE WRONG. IF ALL OF THE PROGRAM INFORMATION WAS CORRECT, TEST FAILURE INDICATES THAT THE SERIAL LINE DEVICE FAILED.

THE TEST SEQUENCE OPERATES IN MAINTENANCE MODE, WITH THE LOOPBACK FEATURE SET. ERROR NUMBERS 2 AND 3, LISTED BELOW, DEFINE ERRORS ENCOUNTERED WHILE ATTEMPTING TO FLOAT A ONE THROUGH A FIELD OF ZEROS. ERROR NUMBERS 5 AND 6 WILL BE DISPLAYED IF AN ERROR IS ENCOUNTERED WHILE FLOATING A ZERO THROUGH THE ONES FIELD. SERIAL LINE LOOPBACK FAILURE, ERRORS 4 AND 7, WILL BE DISPLAYED IF THE BYTE RETURNED BY LOOPBACK DOES NOT COMPARE WITH EXPECTED DATA.

ASSUMPTIONS:

FUNCTIONAL COMMUNICATION INTERFACE

TEST STEPS:

1. CHECK FOR SLU ADDRESS VALIDITY
2. TEST THE SLU IN 'LOOPBACK' MAINTENANCE MODE
IF PRESENCE DETETED APPROPRIATELY:
 - A. PERFORM REGISTER TEST - FLOATING ONE BIT
IN ZEROS FIELD
 - B. PERFORM REGISTER TEST - FLOATING ZERO BIT
IN ONES FIELD
3. INTERROGATE DEVICE - VERIFY PRESENCE AND DEVICE TYPE

ERRORS:

1. SERIAL LINE NOT AT THIS ADDRESS
ERROR NUMBER 0001
2. TIMEOUT WAITING FOR OUTPUT DONE
ERROR NUMBER 0002
3. TIMEOUT WAITING FOR INPUT RESPONSE
ERROR NUMBER 0003
4. SERIAL LINE 'LOOPBACK' FAILED
ERROR NUMBER 0004

399
400
401
402
403
404
405
406
407
408
409
410
411

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
DEFAULT SECTION TEST DESCRIPTIONS

5. TIMEOUT WAITING FOR OUTPUT DONE
ERROR NUMBER 0005

6. TIMEOUT WAITING FOR INPUT RESPONSE
ERROR NUMBER 0006

7. SERIAL LINE 'LOOPBACK' FAILED
ERROR NUMBER 0007

413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
DEFAULT SECTION TEST DESCRIPTIONS

6.2 TEST 2: COLOR PRINTER SELF TEST

THIS TEST SEQUENCE EXECUTES THE COLOR PRINTER SELF TEST AND MONITORS ITS EXECUTION. IF AN UNEXPECTED STATUS RESPONSE IS ENCOUNTERED, IT IS REPORTED AS AN ERROR. ADDITIONALLY, TIMEOUT ERRORS, ERROR NUMBER 1 AND 3, WILL OCCUR IF THE PRINTER MICROCODE DOES NOT RESPOND WITHIN THE ALLOTTED TIME PERIOD.

ASSUMPTIONS:

FUNCTIONAL COMMUNICATION INTERFACE

TEST STEPS:

1. INITIATE SELF TEST EXECUTION (TRANSMIT ESCAPE SEQUENCE)
2. READ COLOR PRINTER STATUS
3. IF ERROR PRINT MESSAGE ELSE END OF TEST

ERRORS:

1. TIMEOUT WAITING FOR INPUT RESPONSE
ERROR NUMBER 0010
2. UNEXPECTED RESPONSE TO 'POWER-UP' SELF TEST
ERROR NUMBER 0011
VERI
CPU BAD
000123 156743 023012 203457 143203 156427 012763 003450
3. TIMEOUT WAITING FOR INPUT RESPONSE
ERROR NUMBER 0012

NOTE

THE LAST TWO LINES OF THE POWER-UP DIAGNOSTIC FAILURE MESSAGE (ITEM #2, ERROR #0011, ABOVE), WILL VARY ACCORDING TO THE VALUE RETURNED BY THE LCPO1. IN ALL INSTANCES A COMPONENT LOCATION ("CPU BAD") AND A PORTION OF THE "VERIFIED" MESSAGE WILL BE DISPLAYED, ALONG WITH A REGISTER DUMP.

463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
DEFAULT SECTION TEST DESCRIPTIONS

6.3 TEST 3: PRINTER DISPLAY TEST

THIS TEST IS DESIGNED TO PRINT A "CANNED" DISPLAY FILE. THIS FILE IS PRINTED WHEN THE DIAGNOSTIC SENDS A SPECIAL ESCAPE SEQUENCE TO THE LCPO1'S CONTROLLER ("ESC", "[", "6", ";", "2", AND "Y"). THIS DISPLAY HAS BEEN TESTED AND WILL WORK UNLESS THERE IS A SOFTWARE SETUP ERROR OR A HARDWARE FAILURE. THE TEST PRINTS OUT A RADIANT DISPLAY.

TEST THREE IS ONLY EXECUTED ON THE FIRST TEST PASS. IF SUBSEQUENT TEST PASSES ARE SPECIFIED, IT IS NOT EXECUTED. SETTING BIT #2 IN THE SOFTWARE OR HARDWARE SWITCH REGISTER WILL CAUSE EXECUTION TO LOOP CONTINUOUSLY ON THIS TEST.

ASSUMPTIONS:

FUNCTIONAL COMMUNICATION INTERFACE
WORKING LCPO1 SOFTWARE DRIVER
WORKING LCPO1 PRINTER

TEST STEPS:

1. SEND ESCAPE SEQUENCE TO PRINTER TO REQUEST PRINT

ERRORS:

NO DISTINCT ERROR MESSAGES ARE REQUIRED BY THIS TEST. THE OPERATOR IS ASKED TO VERIFY THE INTEGRITY OF THE PRINTOUT.

NOTE

THE DIAGNOSTIC DOES NOT WAIT FOR PRINT COMPLETION BEFORE CONTINUING. THE PRINT PROCESS TAKES ABOUT TWO MINUTES.

505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547

CZLCP DIAGNOSTIC PROGRAM USER'S DOCUMENT
END OF DOCUMENT

END OF DOCUMENT

```

549
550
551
552
559
560      .LIST  SEQ,BIN,LOC
561      ;*****;
562      ;
563      .TITLE CZLCPA COLOR PRINTER DIAGNOSTIC
564      ;(DECSPEC-11-BDFAD-A-D)
565      ;
566      ; CZLCP-A-0
567      ; CZLCPA COLOR PRINTER DIAGNOSTIC
568      ; UNIT IS TEX
569      ;
570      ;COPYRIGHT (C) 1984 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
571      ;
572      ;*****;
573      000000      R0=#0
574      000001      R1=#1
575      000002      R2=#2
576      000003      R3=#3
577      000004      R4=#4
578      000005      R5=#5
579      000006      R6=#6
580      000007      R7=#7
581      000006      SP=R6
582      000007      PC=R7
583
584      100000      BIT15 =100000
585      040000      BIT14 =40000
586      020000      BIT13 =20000
587      010000      BIT12 =10000
588      004000      BIT11 =4000
589      002000      BIT10 =2000
590      001000      BIT9 =1000
591      000400      BIT8 =400
592      000200      BIT7 =200
593      000100      BIT6 =100
594      000040      BIT5 =40
595      000020      BIT4 =20
596      000010      BIT3 =10
597      000004      BIT2 =4
598      000002      BIT1 =2
599      000001      BIT0 =1
600
601      000036      SL=36      ;START VFU LOAD
602      000037      EL=37      ;END VFU LOAD

```



```
604
605 000000      ;;;*** .PSECT ABS
606              .ENABLE AMA,ABS
607              .DSABLE GBL
608 000000
609            000000      BEGIN:
610              .=.+0
611
612
613            000030      .=BEGIN+30
614
615 000030 004332      TYP
616 000032 000340      340
617
618
619            000042      .=BEGIN+42
620
621 000042 000000      0
622
623            000046      .=BEGIN+46
624              LOGICAL
625              .=BEGIN+52
626 000052 040000      BIT14
627
628
629            000060      .=BEGIN+60
630 000060 004754      TKINT
631 000062 000300      ;KEYBOARD INTERRUPT ROUTINE
632
633
634            000100      .=BEGIN+100
635
636              LKSRV
637 000100 000340      ;LINE CLOCK SERVICE ROUTINE
638              340
639
640              CONVRT
641              340
642
643 000174 000174      .=BEGIN+174
644 000176 000000      DISPREG: 0
645 000176 000000      SWREG: 0
```

```

646
647
648
649
650
651 000200
652          000200
653
654
655
656
657
658 000200 012706 001000
659
660 000204 000137 001306
661
662          000300
663 000300 000137 001464
664
665          001000
666
667
668
669 001000 177514
670
671
672
673
674 001002 177516
675
676
677
678 001004 177570
679 001006 177570
680 001010 177776
681 001012 177570
682 001014 177566
683 001016 177562
684 001020 177564
685 001022 177560
686 001024 172542
687 001026 172540
688 001030 177546
689
690 001032 176500
691 001034 176500
692 001036 160100
693 001040 160100
694 001042 000300
695 001044 000300
696 001046 000200
697 001050 000200
698 001052 000202
699          000240
700          000001
701          000002
702          000001

.SBTTL PARAMETERS
;*****
;
; BEGINNING OF PROGRAM-
;
; START::
;   . =BEGIN+200
;
; THE FOLLOWING INSTRUCTION "MOV #1000,#6" IS OVER LAID BY THE LP11
; INTERRUPT TESTS. DON'T CHANGE IT WITH OUT BEING AWARE OF THE
; PROBLEMS THAT MAY OCCUR.
;*****
MOV    #1000,#6
;*****
10$:  JMP    SETUP
;*****
;   . =BEGIN+300
;   JMP    TEST1
;
;   . =BEGIN+1000
;
;LINE PRINTER HARDWARE REGISTERS
LPS:   177514      ;STATUS REGISTER
;BIT 15=ERROR
;BIT 7=READY
;BIT 6=INTERRUPT ENABLE
LPB:   177516      ;DATA BUFFER REGISTER
;BITS 0-6=7 BIT ASCII CHARACTER BUFFER
;BITS 7-15=NOT USED
SWR:   177570
DISPLAY:177570
PSW:   177776
HWSWR: 177570
TPB:   177566
TKB:   177562
TPS:   177564
TKS:   177560
CSBR:  172542
PLKS:  172540
LKS:   177546
DLCSRC: .WORD    176500      ;DL'S DEFAULT CSR
DLCSR:  .WORD    176500      ;DL'S DEFAULT CSR
DZCSR:  .WORD    160100      ;DZ'S DEFAULT CSR
DZCSRC: .WORD    160100      ;DZ'S DEFAULT CSR
DLVEC:  .WORD    300         ;DL'S DEFAULT VECTOR ADDRESS
DZVEC:  .WORD    300         ;DZ'S DEFAULT VECTOR ADDRESS
PTRC:   .WORD    200         ;LP'S DEFAULT VECTOR ADDRESS
PTRVEC: .WORD    200
PTRPSW: .WORD    202
NOP     =240
N       =1
M       =2
W       =1

```

704
 705
 706
 707
 708
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 760

```

.SBTTL MACROS
;*****
;
;   -- MACROS --
;
;   ;MACRO FOR SETTING UP ERROR COUNT
;
;   ; $ERROR = CALL
;   ;   X = ERROR NUMBER
;   ;   Y = LOOP ADDRESS IF SWR BIT SET
;   .LIST ME
;
;   .MACRO $ERROR X,Y
ERR'X': MOV    #X,    ERCOUNT    ;SET UP ERROR COUNT X
;   .NLIST ME
;   N=N+1
;   .NLIST
;   .LIST ME
;   .LIST
;   .LIST
;   JSR    #5,STAER    ;REPORT ERROR SET
;   .NLIST ME
;   .IF    NB Y
;   .NLIST
;   .LIST ME
;   .LIST
;
;   $TSWRG #100000    ;CK SW REG
;   BEQ    CN'X'      ;CONTINUE IF BIT 15 = 0
;   JMP    Y          ;OTHERWISE LOOP
;
;   CN'X':
;   .NLIST ME
;   .ENDC
;   .NLIST
;   .LIST ME
;   .LIST
;   .ENDM $ERROR
;
;   ;MACRO FOR PRINTING TEST NUMBER AT START OF TEST
;
;   .MACRO $PRTSN Y
;   EMT    +0
;   TYO'Y'
;   .ENDM $PRTSN
;   ;PRINT TEST NUMBER
;   ;TEST NUMBER MESSAGE
;
;   ; WAIT MACROS - -
;   ; Z = ERROR # (USUALLY "W")
;   ; X = ERROR JMP ADDRESS, IF NONE - DEFAULT LOOPS BACK TO WAIT'Z'
;   ; Y = LOOP ON ERROR, IF BIT 15 OF SWR SET
;   ; #T = 2ND PASS TIME (1 COUNT = ABOUT 1/2 SEC), IF NONE = 100 OCTAL
;
;   .MACRO $WAITI Z,X,Y,T
;   ;WAIT FOR INPUT, OR TIMEOUT
;
;   WAIT'Z':
;   .NLIST ME
;   .IF    NB T

```

```

761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817

```

.NLIST			
.LIST	ME		
.LIST			
	MOV	#100,WORK	;LOAD CNTR
	CMP	#1,CYCCNT	;2ND PASS?
	BNE	WDD'Z'	;NO, LEAVE COUNT AS IS
	MOV	T,WORK	;LOAD CNTR WITH LONG CNT
.NLIST	ME		
	.ENDC		
	.IF	B T	
.NLIST			
.LIST	ME		
.LIST			
	MOV	#100,WORK	;LOAD CNTR
.NLIST	ME		
	.ENDC		
.NLIST			
.LIST	ME		
.LIST			
WDD'Z' :	MOV	#-1,WORKA	;LOAD COUNT
	DEC	WORK	;BUMP CNTR
	BEQ	WER'Z'	;ERROR IF TIMEOUT
WDE'Z' :	DEC	WORKA	;BUMP COUNT
	BEQ	WDD'Z'	;LOOP IF COUNT 0
	TSTB	@DLLPS	;CK STATUS
	BPL	WDE'Z'	;LOOP TIL DONE
	BR	WEX'Z'	;CONTINUE
WER'Z' :	\$TSWRG	\$20000	;CK SW REG
			;TEST FOR INHIBIT ERROR MSG
	BNE	.+6	;BRANCH IF NO MSG WANTED
	EMT	+0	
	ETIM		
.NLIST	ME		;TIMED OUT ERROR
	.IF	NB Y	
.NLIST			
.LIST	ME		
.LIST			
	\$ERROR	\N,Y	
.NLIST	ME		
	.ENDC		
	.IF	B Y	
.NLIST			
.LIST	ME		
.LIST			
	\$ERROR	\N	
.NLIST	ME		
	.ENDC		
	.IF	B X	
.NLIST			
.LIST	ME		
.LIST			
	BR	WAIT'Z'	;LOOP
.NLIST	ME		

```

818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874

```

```

      .ENDC
      .IF      NB X
      .NLIST
      .LIST   ME
      .LIST
      .NLIST  JMP      X
      .LIST   ME
      .LIST   .ENDC
      .NLIST
      .LIST   ME
      .LIST
      WEX'Z':
      .NLIST  ME
      .LIST   W=W+1
      .NLIST
      .LIST   ME
      .LIST
      .ENDM  $WAITI

      ;MACRO FOR WAITING FOR OUPUT, DONE

      .MACRO $WAITO Z,X,Y
      WAIT'Z': MOV     #20,TIME
      WDD'Z':  MOV     #-1,TIMER
      WDE'Z':  DEC     TIMER
      WER'Z':  BEQ     WER'Z'
      WDE'Z':  DEC     TIMER
      WDD'Z':  BEQ     WDD'Z'
      WER'Z':  TSTB   @LPS
      WDE'Z':  BPL     WDE'Z'
      WER'Z':  BR      WER'Z'
      WER'Z':  $TSWRG #20000
      BNE     .+6
      EMT     +0
      ETIMO
      .NLIST  ME
      .IF      NB Y
      .NLIST
      .LIST   ME
      .LIST
      $ERROR  \N,Y
      .NLIST  ME
      .ENDC
      .IF      B Y
      .NLIST
      .LIST   ME
      .LIST
      $ERROR  \N
      .NLIST  ME

```

```

;WAIT FOR OUPUT, OR TIMEOUT
;10 SEC CNTR
;LOAD COUNT
;BUMP CNTR
;ERROR IF TIMEOUT
;BUMP COUNT
;LOOP IF COUNT 0
;CK STATUS
;LOOP TIL DONE
;CONTINUE
;CK SW REG
;TEST FOR INHIBIT ERROR MSG
;BRANCH IF NO MSG WANTED
;TIMED OUT ERROR

```

```

875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931

```

```

      .ENDC
      .IF      B X
.NLIST
.LIST ME
.LIST
.NLIST BR      WAIT'Z'           ;LOOP
      ME
      .ENDC
      .IF      NB X
.NLIST
.LIST ME
.LIST
.NLIST JMP      X           ;EXIT
      ME
      .ENDC
.NLIST
.LIST ME
.LIST
      WEX'Z':
.NLIST ME
      W=W+1
.NLIST
.LIST ME
.LIST
      .ENDM    $WAITO

;
;MACRO FOR ENABLING KEYBOARD INTERUPT
;
;MACRO $ENABLE
.NLIST ME
;:::000 CMP      #176,SWR           ;S/W SWR ?
;:::000 BNE      .+10             ;NO- CONTINUE
.NLIST
.LIST ME
.LIST
      BIS      #100,@TKS           ;ENABLE KEYBOARD INTERRUPT
      MOV      -(SP),-(SP)
      MOV      #0,2(SP)
      MOV      #.+6,(SP)
      RTI
      .ENDM    $ENABLE

;MACRO USED TO LOAD THE PSW WITH THE
;CORRECT PROCESSOR PRIORITY LEVEL
;
;MACRO $SETPSW
MOV      PC,-(SP)           ;MOVE PRESENT LOCATION TO STACK
ADD      #6,(SP)           ;SET UP FOR NEXT INSTRUCTION
RTI           ;LOAD PSW
.ENDM    $SETPSW

;
;MACRO USED TO PRINT MESSAGE TO LINE PRINTER

```

932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955

```
;  
; .MACRO $PRINT V  
MOV #'V',PRTMSG ;LOAD MESSAGE ADDRESS  
JSR %4,PRINE ;PRINT IT  
; .ENDM $PRINT  
  
;  
; MACRO TO TYPE MESSAGE ON THE TERMINAL  
;  
; .MACRO $TYPE G  
EMT +0 ;Call "TYP" interrupt  
G ;address of message  
; .ENDM $TYPE  
  
;  
; MACRO TO TEST HARDWARE AND SOFTWARE SWITCH REGISTERS  
;  
; .MACRO $TSWRG MSK  
CMP #176,SWR ;HW SWITCH REG THERE?  
BEQ .+12 ;NO, SKP HW CHECK  
BIT MSK,@HWSWR ;YES, CK HW REG  
BNE .+10 ;IF SET, SKP SW REG CK  
BIT MSK,SWREG ;OTHERWISE, ALSO CK SW REG  
; .ENDM
```

```

957
958
959
960
961
962 001054 000000
963 001056 000000
964 001060 000000
965 001062 000000
966 001064 000000
967
968 001066 000000
969 001070 000000
970 001072 000000
971
972 001074 000000
973 001076 000000
974 001100 000000
975 001102 000000
976 001104 000000
977 001106 000000
978
979 001110 000000
980 001112 000000
981
982 001114 000000
983 001116 000000
984 001120 000000
985 001122 000000
986
987 001124 000000
988 001126 000000
989 001130 000000
990 001132 000000
991 001134 000000
992 001136 000000
993 001140 000000
994 001142 000000
995 001144 000000
996 001146 000000
997 001150 000000
998 001152 000000
999 001154 000000
1000 001156 000000
1001 001160 000000
1002 001162 000000
1003 001164 000000
1004 001166

.SBTTL VARIABLES
;*****
;MEMORY LOCATIONS USED AS PROGRAM FLAGS AND COUNTERS
;
SEG CNT: 0
CHR CNT: 0
CHR GEN: 0
LIN CNT: 0
CYC CNT: 0

DZ TCR: .WORD 0
DZ LPR: .WORD 0
DZ LNE: .WORD 0

DLL PR: .WORD 0
DLL PS: .WORD 0
DLR BUF: .WORD 0
BRATE: .WORD 0
DLR RATE: .WORD 0
DL HERE: .WORD 0

DZ TCRA: .WORD 0
EIA: .WORD 0

DL TYPE: .WORD 0
DZ R BUF: .WORD 0
DZ CSRH: .WORD 0
MAINT B: .WORD 0

WORK: 0
WORKA: 0
TIME: 0
TIMER: 0
SAVE: 0
ERCOUNT: 0
STRCHR: 0
STRCNT: 0
LEGCHR: 0
NUMCHR: 0
OFFSET: 0
DIGITS: 0
SIGNAL: 0
SET: 0
CHAR: 0
OCT: 0
PASSA: 0
BUFF: .BLKB 80.

;HOLDS DZ'S TCR REGISTER
;HOLDS DZ'S LPR REGISTER
;HOLDS DZ'S LINE #

;HOLDS DL'S BAUD RATE BITS
;DL'S RECV REG
;DL'S RECV BUFFER
;HOLD BAUD RATE BITS
;(DL) HOLD BAUD RATE BITS
;SHOWS DL11 PRESENCE

;ADDRESS OF DZ DTR REGISTER
;LINE TYPE 0=20MA 1=EIA

;DL TYPE 0=ALL OTHERS 1=E/F
;DZ'S RECEIVER BUFFER ADDRESS
;HOLDS DZ'S CSR ADDRESS
;MAINTENANCE BIT, IF NOT 0
    
```



```

1006
1007
1008
1009
1010
1011
1012 001306 004437 004314
1013 001312 005037 001064
1014 001316 005037 001154
1015 001322 000005
1016 001324 013746 000004
1017 001330 013746 000006
1018 001334 012737 001350 000004
1019 001342 005777 177444
1020 001346 000407
1021 001350
1022 001350 012737 000176 001004
1023 001356 012737 000174 001006
1024 001364 022626
1025 001366 012637 000006
1026 001372 012637 000004
1027 001376
      001376 052777 000100 177416
      001404 014646
      001406 012766 000000 000002
      001414 012716 001422
      001420 000002
1028 001422 005737 001164
1029 001426 001011
1030 001430 112737 000001 001164
1031 001436 104000
1032 001440 006153
1033 001442 104000
1034 001444 006167
1035 001446 104000
1036 001450 006230
1037 001452 004737 010234
1038 001456 000005
1039 001460 104000
1040 001462 006246

.SBTTL SETUP SERIAL LINE PARAMETERS
;*****
;ROUTINE TO TEST THE MECH. OPERATION OF THE LXY11/21-LXV11
;
SETUP: JSR      #4,TYPINT      ;PRESET POINTERS
        CLR      CYCCNT      ;CLEAR PASS COUNT
        CLR      SIGNAL      ;CTRL G FLAG
        RESET     ;CLEAR WORLD
        MOV      4,-(SP)      ;SAVE CURRENT VECTORS
        MOV      6,-(SP)
        MOV      #1$,4
        TST      @HWSWR      ;SET UP TIMEOUT VECTOR
        BR       2$          ;TRY TO ACCESS HARDWARE SWR
                                ;IF THERE, GO TO 2$
1$:     MOV      #SWREG,SWR    ;POINT TO SOFTWARE SWR
        MOV      #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY
        CMP      (SP)+,(SP)+  ;RESTORE STACK
2$:     MOV      (SP)+,6      ;RESTORE TIMEOUT VECTORS
        MOV      (SP)+,4
        #ENABLE
        BIS      #100,@TKS    ;ENABLE KEYBOARD INTERRUPT
        MOV      -(SP),-(SP)
        MOV      #0,2(SP)
        MOV      #.+6,(SP)
        RTI
        TST      PASSA
        BNE      3$          ;SKP
        MOVB    #1,PASSA
        EMT     +0
        MES1    ;TYPE DIAGNOSTIC TITLE
        EMT     +0
        MES2    ;TYPE NAME
        EMT     +0
        MES3
3$:     JSR      #7,SETSER    ;CHECK FOR SERIAL LINE SELECTION
        RESET   ;REQUIRED INSURANCE
        EMT     +0
        MES4    ;TYPE RESTART ADDRESS INFO
  
```

```

1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066 001464 052777 000100 177330
1067 001472
      001472 022737 000176 001004
      001500 001404
      001502 032777 040000 177302
      001510 001003
      001512 032737 040000 000176
1068 001520 001002
1069 001522
      001522 104000
      001524 005571
1070
1071 001526 013746 000004
1072 001532 013746 000006
1073 001536 012737 001552 000004
1074 001544 105777 177230
1075 001550 000447
1076
1077 001552
1078 001552 022626
1079 001554 012637 000006
1080 001560 012637 000004
1081 001564
      001564 022737 000176 001004
      001572 001404
      001574 032777 020000 177210
      001602 001003
      001604 032737 020000 000176
1082 001612 001002
1083 001614 104000
1084 001616 006704
1085 001620
      001620 012737 000001 001136

```

```

.SBTTL
.SBTTL TEST #  DESCRIPTION
.SBTTL -----
.SBTTL TEST 1  SERIAL LINE UNIT TEST
*****
:
:                               Test 1 Serial Line Unit test
:
:   Test 1-A
:
:   - Insure SLU is there, at correct address.
:
:   Test 1-B
:
:   - If Loopback feature (echo) is present, Test the SLU:
:
:       a - with floating one bit field
:       b - with floating zero bit field
:
:
:   CHECK FOR PRINTER (SLU) ON BUS
:
:   .ENABL LSB
:   BIS      @100,@TKS      ;ENABLE KEYBOARD
:   $TSWRG   @40000        ;CK SW REG
:   CMP      @176,SWR      ;HW SWITCH REG THERE?
:   BEQ      .+12          ;NO, SKP HW CHECK
:   BIT      @40000,@HWSR  ;YES, CK HW REG
:   BNE      .+10          ;IF SET, SKP SW REG CK
:   BIT      @40000,SWREG  ;OTHERWISE, ALSO CK SW REG
:   BNE      1$           ;IF SET, SKIP TST # HDR MSG
:   $PRTSN   1
:   EMT      +0
:   TY01
:                               ;PRINT TEST NUMBER
:                               ;TEST NUMBER MESSAGE
:
:   1$:  MOV      @@4,-(SP)  ;SAVE VECTORS
:        MOV      @@6,-(SP) ;SAVE
:        MOV      @2$,4     ;RELOAD VECTOR
:        TSTB    @LPS      ;IS PRINTER THERE?
:        BR      4$        ;YES, SKIP TRAP PROCESSING
:
:   2$:  CMP      (SP),-(SP) ;IF YES, NEVER GET HERE.
:        MOV      (SP),6    ;RESTORE STACK
:        MOV      (SP),4    ;RESTORE VECTORS
:        $TSWRG  @20000    ;RESTORE
:        CMP      @176,SWR  ;CK SW REG
:        BEQ      .+12      ;HW SWITCH REG THERE?
:        BIT      @20000,@HWSR ;NO, SKP HW CHECK
:        BNE      .+10      ;YES, CK HW REG
:        BIT      @20000,SWREG ;IF SET, SKP SW REG CK
:        BNE      .+6       ;OTHERWISE, ALSO CK SW REG
:        EMT      +0        ;IF INHIBIT ERR MSG
:
:   ERMS1
:   $ERROR   \N,1$
:   MOV      @1, ERCOUNT ;SLU NOT THERE
:                               ;NOTHING THERE
:                               ;SET UP ERROR COUNT 1

```

```
001626 004537 004606          JSR      #5,STAER          ;REPORT ERROR SET
001632                                $TSWRG #100000          ;CK SW REG
001632 022737 000176 001004    CMP      #176,SWR          ;HW SWITCH REG THERE?
001640 001404                                BEQ      .+12              ;NO, SKP HW CHECK
001642 032777 100000 177142    BIT      #100000,@MWSWR   ;YES, CK HW REG
001650 001003                                BNE      .+10              ;IF SET, SKP SW REG CK
001652 032737 100000 000176    BIT      #100000,SWREG   ;OTHERWISE, ALSO CK SW REG
001660 001402                                BEQ      CN1              ;CONTINUE IF BIT 15 = 0
001662 000137 001526                                JMP      1$                ;OTHERWISE LOOP
001666                                CN1:
1086                                ;
1087 001666 000607          BR      SETUP
1088                                ;
1089                                ; YES, SLU IS HERE
1090                                ;
1091 001670 012637 000006    4$:    MOV      (SP)+,6          ;RESTORE VECTORS
1092 001674 012637 000004    MOV      (SP)+,4          ;RESTORE
```

```

1094
1095
1096
1097
1098
1099
1100
1101
1102
1103 001700 005737 001122
1104 001704 001002
1105 001706 000137 003016
1106
1107
1108
1109
1110
1111
1112
1113 001712 053777 001122 177060
1114 001720 012700 000001
1115 001724 010037 001134
1116
1117 001730
      001730 012737 000020 001130
      001736 012737 177777 001132
      001744 005337 001130
      001750 001407
      001752 005337 001132
      001756 001767
      001760 105777 177014
      001764 100372
      001766 000442
      001770
      001770 022737 000176 001004
      001776 001404
      002000 032777 020000 177004
      002006 001003
      002010 032737 020000 000176
      002016 001002
      002020 104000
      002022 006007
      002024
      002024 012737 000002 001136
      002032 004537 004606
      002036
      002036 022737 000176 001004
      002044 001404
      002046 032777 100000 176736
      002054 001003
      002056 032737 100000 000176
      002064 001402
      002066 000137 001712

*****
:
:   Test 1-B
:
:   - If Loopback feature (echo) is present, Test the SLU:
:
:       a - with floating one bit field
:       b - with floating zero bit field
:
:   TST    MAINIB      ;NONZERO IF LOOPBACK FEATURE PRESENT
:   BNE    9$          ;HERE, CONTINUE
:   JMP    40$         ;NOT HERE, SKIP SUBTEST
:
-----
:
:   Test 1-B-a
:
:   FLOAT A ONE BIT THRU ALL ZERO BYTE
:
:9$:   BIS    MAINTB,@LPS      ;SET MAINTENANCE (LOOPBACK) BIT
:      MOV    #1,RO          ;SET BIT #0
:10$:  MOV    RO,SAVE        ;SAVE IT
:
:12$:  $WAITO \W,.9$
:
:      WAIT1: MOV    #20,TIME      ;WAIT FOR OUPUT, OR TIMEOUT
:      WDD1:  MOV    #-1,TIMER     ;10 SEC CNTR
:           DEC    TIME          ;LOAD COUNT
:           BEQ    WER1          ;BUMP CNTR
:           DEC    TIMER        ;ERROR IF TIMEOUT
:           BEQ    WDD1          ;BUMP COUNT
:           TSTB  @LPS          ;LOOP IF COUNT 0
:           BPL  WDE1            ;CK STATUS
:           BR    WEX1           ;LOOP TIL DONE
:           $TSWRG #20000        ;CONTINUE
:           CMP   #176,SWR       ;CK SW REG
:           BEQ  .+12            ;HW SWITCH REG THERE?
:           BIT  #20000,@HWSWR   ;NO, SKP HW CHECK
:           BNE  .+10            ;YES, CK HW REG
:           BIT  #20000,SWREG    ;IF SET, SKP SW REG CK
:           BNE  .+6             ;OTHERWISE, ALSO CK SW REG
:           EMT  +0              ;TEST FOR INHIBIT ERROR MSG
:           ETIMO                ;BRANCH IF NO MSG WANTED
:
:           ;TIMED OUT ERROR
:
:      ERR2: $ERROR \N,9$
:           MOV    #2, ERCOUNT  ;SET UP ERROR COUNT 2
:           JSR   #5,STAER       ;REPORT ERROR SET
:
:      $TSWRG #100000           ;CK SW REG
:      CMP   #176,SWR          ;HW SWITCH REG THERE?
:      BEQ  .+12                ;NO, SKP HW CHECK
:      BIT  #100000,@HWSWR     ;YES, CK HW REG
:      BNE  .+10                ;IF SET, SKP SW REG CK
:      BIT  #100000,SWREG      ;OTHERWISE, ALSO CK SW REG
:      BEQ  CN2                 ;CONTINUE IF BIT 15 = 0
:      JMP  9$                  ;OTHERWISE LOOP
  
```



```

002266 004537 004606          JSR      #5,STAER          ;REPORT ERROR SET
002272
002272 022737 000176 001004    $TSWRG #100000          ;CK SW REG
002300 001404          CMP      #176,SWR      ;HW SWITCH REG THERE?
002302 032777 100000 176502    BEQ      .+12          ;NO, SKP HW CHECK
002310 001003          BIT      #100000,@HWSWR ;YES, CK HW REG
002312 032737 100000 000176    BNE      .+10          ;IF SET, SKP SW REG CK
002320 001402          BIT      #100000,SWREG ;OTHERWISE, ALSO CK SW REG
002322 000137 001712          BEQ      CN4          ;CONTINUE IF BIT 15 = 0
002326          JMP      9$          ;OTHERWISE LOOP
1129 002326 013700 001134    CN4:   MOV      SAVE,RO      ;RESTORE, RO
1130 002332 000137 001730    JMP      12$          ;TRY AGAIN
1131
1132 002336 000241          16$:   CLC              ;CLEAR CARRY
1133 002340 006100          ROL      RO            ;SHIFT BIT
1134 002342 105700          TSTB     RO
1135 002344 100402          BMI      .+6
1136 002346 000137 001724    JMP      10$          ;NOT FINISHED
1137
1138
1139
1140
1141
1142
1143
1144 002352 012700 000176
1145 002356 142700 000200
1146 002362 010037 001134
1147
1148 002366          20$:   MOV      #176,RO      ;SET ALL EXCEPT BIT #0
          BICB     #200,RO    ;CLR BIT #7
          MOV      RO,SAVE  ;SAVE IT
          22$:   $WAITO  \W,,20$
          ;WAIT FOR OUPUT, OR TIMEOUT
          ;10 SEC CNTR
          ;LOAD COUNT
          ;BUMP CNTR
          ;ERROR IF TIMEOUT
          ;BUMP COUNT
          ;LOOP IF COUNT 0
          ;CK STATUS
          ;LOOP TIL DONE
          ;CONTINUE
          ;WAIT3: MOV      #20,TIME
          ;WDD3: MOV      #-1,TIMER
          ;DEC      TIME
          ;BEQ      WER3
          ;WDE3: DEC      TIMER
          ;BEQ      WDD3
          ;TSTB    @LPS
          ;BPL     WDE3
          ;BR      WEX3
          ;WER3: $TSWRG #20000          ;CK SW REG
          ;CMP      #176,SWR      ;HW SWITCH REG THERE?
          ;BEQ      .+12          ;NO, SKP HW CHECK
          ;BIT      #20000,@HWSWR ;YES, CK HW REG
          ;BNE      .+10          ;IF SET, SKP SW REG CK
          ;BIT      #20000,SWREG ;OTHERWISE, ALSO CK SW REG
          ;BNE      .+6          ;TEST FOR INHIBIT ERROR MSG
          ;EMT     +0          ;BRANCH IF NO MSG WANTED
          ;ETIMO
          ;TIMED OUT ERROR
          ;ERR5: $ERROR  \N,20$
          ;MOV      #5,          ERCOUNT
          ;JSR      #5,STAER      ;SET UP ERROR COUNT 5
          ;REPORT ERROR SET
002474          $TSWRG #100000          ;CK SW REG
  
```

```

002474 022737 000176 001004      CMP      #176,SWR      ;HW SWITCH REG THERE?
002502 001404                      BEQ      .+12        ;NO, SKP HW CHECK
002504 032777 100000 176300      BIT      #100000,@HWSWR ;YES, CK HW REG
002512 001003                      BNE      .+10        ;IF SET, SKP SW REG CK
002514 032737 100000 000176      BIT      #100000,SWREG ;OTHERWISE, ALSO CK SW REG
002522 001402                      BEQ      CN5        ;CONTINUE IF BIT 15 = 0
002524 000137 002356                      JMP      20$        ;OTHERWISE LOOP
002530                                CNS:
002530 000716                      BR       WAIT3      ;LOOP
002532                                WEX3:
1149 002532 110077 176244      MOV      RO,@LPB    ;XMIT BYTE
1150                                24$:
1151 002536                                $WAITI \W,,20$
002536                                WAIT4:
002536 012737 000100 001124      MOV      #100,WORK  ;LOAD CNTR
002544 012737 177777 001126      WDD4:  MOV      #-1,WORKA ;LOAD COUNT
002552 005337 001124                      DEC      WORK        ;BUMP CNTR
002556 001407                      BEQ      WER4        ;ERROR IF TIMEOUT
002560 005337 001126      WDE4:  DEC      WORKA    ;BUMP COUNT
002564 001767                      BEQ      WDD4        ;LOOP IF COUNT 0
002566 105777 176304      TSTB    @DLLPS    ;CK STATUS
002572 100372                      BPL      WDE4        ;LOOP TIL DONE
002574 000442                      BR       WEX4        ;CONTINUE
002576                                WER4:
002576 022737 000176 001004      $TSWRG #20000      ;CK SW REG
002604 001404                      CMP      #176,SWR  ;HW SWITCH REG THERE?
002606 032777 020000 176176      BEQ      .+12        ;NO, SKP HW CHECK
002614 001003                      BIT      #20000,@HWSWR ;YES, CK HW REG
002616 032737 020000 000176      BNE      .+10        ;IF SET, SKP SW REG CK
                                ;OTHERWISE, ALSO CK SW REG
002624 001002                      BIT      #20000,SWREG ;TEST FOR INHIBIT ERROR MSG
                                ;BRANCH IF NO MSG WANTED
002626 104000                      BNE      .+6
002630 005731                      EMT      +0
                                ;TIMED OUT ERROR
002632                                ERR6:
002632 012737 000006 001136      $ERROR  \N,20$
002640 004537 004606      MOV      #6, ERCCOUNT ;SET UP ERROR COUNT 6
                                JSR      #5,STAER   ;REPORT ERROR SET
002644                                $TSWRG #100000
002644 022737 000176 001004      CMP      #176,SWR  ;CK SW REG
002652 001404                      BEQ      .+12        ;HW SWITCH REG THERE?
002654 032777 100000 176130      BIT      #100000,@HWSWR ;NO, SKP HW CHECK
002662 001003                      BNE      .+10        ;YES, CK HW REG
002664 032737 100000 000176      BIT      #100000,SWREG ;IF SET, SKP SW REG CK
002672 001402                      BEQ      CN6        ;OTHERWISE, ALSO CK SW REG
002674 000137 002356                      JMP      20$        ;CONTINUE IF BIT 15 = 0
002700                                CN6:
                                ;OTHERWISE LOOP
002700 000716                      BR       WAIT4      ;LOOP
002702                                WEX4:
1152                                WEX4:
1153 002702 117701 176172      MOV      @DLRBUF,R1 ;GET BYTE IN

```

```

1154 002706 120001
1155 002710 001431      CMPB   R0,R1      ;SAME?
1156                   BEQ    26$      ;OK
1157 002712 104000
1158 002714 006756      EMT
1159 002716           $ERMS2   +0
      002716 012737 000007 001136 ERR7: $ERROR \N,20$      ;"LOOPBACK FAILED" MSG
      002724 004537 004606           MOV   #7,   ERCOUNT ;LOOPBACK FAILED
      002730           JSR   #5,STAER ;SET UP ERROR COUNT 7
      002730           ;REPORT ERROR SET
      002730           $TSWRG   #100000 ;CK SW REG
      002730 022737 000176 001004      CMP   #176,SWR    ;HW SWITCH REG THERE?
      002736 001404           BEQ   .+12         ;NO, SKP HW CHECK
      002740 032777 100000 176044      BIT   #100000,@HWSWR ;YES, CK HW REG
      002746 001003           BNE   .+10         ;IF SET, SKP SW REG CK
      002750 032737 100000 000176      BIT   #100000,SWREG ;OTHERWISE, ALSO CK SW REG
      002756 001402           BEQ   CN7          ;CONTINUE IF BIT 15 = 0
      002760 000137 002356           JMP   20$        ;OTHERWISE LOOP
      002764           ;
1160 002764 013700 001134      MOV   SAVE,R0     ;RESTORE, R0
1161 002770 000137 002366      JMP   22$        ;TRY AGAIN
1162
1163 002774 000261           26$: SEC
1164 002776 006100           ROL   R0          ;SET CARRY
1165 003000 105700           TSTB  R0          ;SHIFT BIT
1166 003002 100002           BPL
1167 003004 000137 002356      JMP   .+6
1168 003010 043777 001122 175762      BIC   MAINTB,@LPS ;NOT FINISHED
1169                   ;CLEAR MAINTENANCE (LOOPBACK) BIT
1170 003016           ;
1171 003016           40$: $TSWRG   #1      ;CK SW REG
      003016 022737 000176 001004      CMP   #176,SWR    ;HW SWITCH REG THERE?
      003024 001404           BEQ   .+12         ;NO, SKP HW CHECK
      003026 032777 000001 175756      BIT   #1,@HWSWR   ;YES, CK HW REG
      003034 001003           BNE   .+10         ;IF SET, SKP SW REG CK
      003036 032737 000001 000176      BIT   #1,SWREG    ;OTHERWISE, ALSO CK SW REG
1172 003044 001402           BEQ   TEST2       ;CONTINUE IF = 0
1173 003046 000137 001464      JMP   TEST1      ;IF SET, LOOP ON TEST
1174
1175
1176

```



```

1178 .SBTTL TEST 2 COLOR PRINTER 'SELF' TEST
1179 ;*****
1180 ;
1181 ; Test 2 Initialization Test
1182 ;
1183 ; -A Invoke "Confidence (Self) Test" of Color Printer
1184 ;
1185 ; -B Monitor and report unexpected response:
1186 ;
1187 ; Issue request for device "state" or "status" and check for
1188 ; correct response:
1189 ;
1190 ; "VERIFIED" message = I'm OK
1191 ; Anything else = I'm Not OK
1192 ;
1193 ;
1194 003052 TEST2: $TSWRG #40000 ;CK SW REG
      003052 022737 000176 001004 CMP #176,SWR ;HW SWITCH REG THERE?
      003060 001404 BEQ .+12 ;NO, SKP HW CHECK
      003062 032777 040000 175722 BIT #40000,@HWSWR ;YES, CK HW REG
      003070 001003 BNE .+10 ;IF SET, SKP SW REG CK
      003072 032737 040000 000176 BIT #40000,SWREG ;OTHERWISE, ALSO CK SW REG
1195 003100 001002 BNE 61$ ;IF SET, SKIP TST # HDR MSG
1196 003102 $PRTSN 2 ;PRINT TEST NUMBER
      003102 104000 EMT +0 ;TEST NUMBER MESSAGE
      003104 005630 TY02
1197 ;
1198 ; SEND "ESCAPE" SEQUENCE TO THE PRINTER
1199 ;
1200 ;
1201 ;
1202 003106 61$: $PRINT LCPS ;INVOKE CONFIDENCE TEST
      003106 012737 005542 004312 MOV #LCP5,PRMSG ;LOAD MESSAGE ADDRESS
      003114 004437 004250 JSR #4,PRINE ;PRINT IT
1203 ;
1204 ; EXPECT "VERIFIED" MESSAGE FROM PRINTER
1205 ;
1206 ;
1207 ;
1208 ;
1209 003120 62$: MOV #BUFF,R3 ;GET RESPONSE
      003124 105013 CLRB (R3) ;INPUT BUFFER ADDRESS
1210 003126 012701 005556 MOV #LCP7,R1 ;NULL BYTE
1211 003132 111100 64$: MOVB (R1),R0 ;EXPECTED STRING ADDRESS
1212 003134 001002 BNE .+6 ;GET EXPECTED CHAR
1213 003136 000137 003604 JMP 69$ ;END OF MSG, EXIT
1214 003142 $WAITI \W.,TEST2,#600 ;WAIT FOR INPUT, OR TIMEOUT
1215 ;
      003142 WAIT5: ;WAIT FOR INPUT, OR TIMEOUT
      003142 012737 000100 001124 MOV #100,WORK ;LOAD CNTR
      003150 022737 000001 001064 CMP #1,CYCCNT ;2ND PASS?
      003156 001003 BNE WDD5 ;NO, LEAVE COUNT AS IS
      003160 012737 000600 001124 MOV #600,WORK ;LOAD CNTR WITH LONG CNT
      003166 012737 177777 001126 WDD5: MOV #-1,WORKA ;LOAD COUNT
      003174 005337 001124 DEC WORK ;BUMP CNTR
      003200 001407 BEQ WERS ;ERROR IF TIMEOUT
      003202 005337 001126 WDE5: DEC WORKA ;BUMP COUNT

```

```

003206 001767
003210 105777 175662
003214 100372
003216 000442
003220
003220 022737 000176 001004 WERS: $TSWRG #20000 ;CK SW REG
003226 001404 CMP #176,SWR ;HW SWITCH REG THERE?
003230 032777 020000 175554 BEQ .+12 ;NO, SKP HW CHECK
003236 001003 BIT #20000,@HWSWR ;YES, CK HW REG
003240 032737 020000 000176 BNE .+10 ;IF SET, SKP SW REG CK
;OTHERWISE, ALSO CK SW REG
003246 001002 BIT #20000,SWREG ;TEST FOR INHIBIT ERROR MSG
;BRANCH IF NO MSG WANTED
003250 104000 EMT +0
003252 005731 ETIM ;TIMED OUT ERROR
003254
003254 012737 000010 001136 ERR10: $ERROR \N,TEST2
003262 004537 004606 MOV #10, ERCOUNT ;SET UP ERROR COUNT 10
JSR #5,STAER ;REPORT ERROR SET
003266
003266 022737 000176 001004 $TSWRG #100000 ;CK SW REG
003274 001404 CMP #176,SWR ;HW SWITCH REG THERE?
003276 032777 100000 175506 BEQ .+12 ;NO, SKP HW CHECK
003304 001003 BIT #100000,@HWSWR ;YES, CK HW REG
003306 032737 100000 000176 BNE .+10 ;IF SET, SKP SW REG CK
003314 001402 BIT #100000,SWREG ;OTHERWISE, ALSO CK SW REG
003316 000137 003052 BEQ CN10 ;CONTINUE IF BIT 15 - 0
003322 CN10: JMP TEST2 ;OTHERWISE LOOP
003322 000707 BR WAIT5 ;LOOP
003324
1216 003324 117702 175550 WEX5:
1217 003330 110223 MOV B @DLRBUF,R2 ;GET CHAR IN
1218 003332 105013 MOV B R2,(R3)+ ;STORE IT
1219 003334 112100 CLRB (R3) ;NULL NEXT BYTE
1220 003336 120002 MOV B (R1)+,R0 ;GET EXPECTED CHAR
1221 003340 001674 CMPB R0,R2 ;COMPARE
1222 BEQ 64$ ;OK, WAIT FOR ANOTHER
;
; IGNORE UP TO 4 GARBAGE BYTES. LOOKING FOR "V"
1225 003342 020327 001172 CMP R3,#BUFF+4 ;R3 POINTS TO GARBAGE STORED
1226 003346 003003 BGT 66$ ;MORE THAN 4 MISCOMPARES
1227 003350 105741 TSTB -(R1) ;RESET TO BEGINING OF MSG
1228 003352 000137 003132 JMP 64$ ;TRY AGAIN
1229
1230 003356 104000 66$: EMT +0
1231 003360 007120 ERMS4 ;UNEXPECTED RESPONSE
1232 003362
003362 012737 000011 001136 ERR11: $ERROR \N
003370 004537 004606 MOV #11, ERCOUNT ;SET UP ERROR COUNT 11
JSR #5,STAER ;REPORT ERROR SET
1233
1234 ; Print incomplete "VERIFIED" message from "Power-up" diagnostic
1235 ;
1236 003374 104000 EMT +0
1237 003376 001166 BUFF ;Stored message
  
```

```

1238 003400 012737 000005 001062      MOV      #5,LINCNT      ;Load line count
1239                                     ;
1240                                     ; Type all subsequent info sent by printer = error info
1241                                     ;
1242 003406      67$:      $WAITI  \W,79$,TEST?
                                ;WAIT FOR INPUT, OR TIMEOUT
                                ;
                                ;
003406      003406 012737 000100 001124      WAIT6:
003414 003414 012737 177777 001126      WDD6:  MOV      #100,WORK      ;LOAD CNTR
003422 003422 005337 001124      WDD6:  MOV      #-1,WORKA     ;LOAD COUNT
003426 003426 001407      WDD6:  DEC      WORK          ;BUMP CNTR
003430 003430 005337 001126      WDE6:  BEQ      WER6          ;ERROR IF TIMEOUT
003434 003434 001767      WDE6:  DEC      WORKA       ;BUMP COUNT
003436 003436 105777 175434      WDE6:  BEQ      WDD6          ;LOOP IF COUNT 0
003442 003442 100372      WDE6:  TSTB     @DLLPS       ;CK STATUS
003444 003444 000443      WDE6:  BPL      WDE6         ;LOOP TIL DONE
003446      WDE6:  BR       WEX6          ;CONTINUE
003446 003446 022737 000176 001004      WER6:  $TSWRG  #20000      ;CK SW REG
003454 003454 001404      WER6:  CMP      #176,SWR     ;HW SWITCH REG THERE?
003456 003456 032777 020000 175326      WER6:  BEQ      .+12         ;NO, SKP HW CHECK
003464 003464 001003      WER6:  BIT      #20000,@HWSWR ;YES, CK HW REG
003466 003466 032737 020000 000176      WER6:  BNE      .+10         ;IF SET, SKP SW REG CK
                                ;OTHERWISE, ALSO CK SW REG
                                ;TEST FOR INHIBIT ERROR MSG
                                ;BRANCH IF NO MSG WANTED
003474 003474 001002      WER6:  BIT      #20000,SWREG
                                ;
                                ;
003476 003476 104000      WER6:  BNE      .+6
003500 003500 005731      WER6:  EMT      +0
                                ;
                                ;
                                ;TIMED OUT ERROR
003502
003502 003502 012737 000012 001136      ERR12: $ERROR  \N,TEST2
003510 003510 004537 004606      ERR12: MOV      #12,  ERCOUNT ;SET UP ERROR COUNT 12
                                ;REPORT ERROR SET
                                ;
003514      $TSWRG  #100000      ;CK SW REG
003514 003514 022737 000176 001004      ERR12: CMP      #176,SWR     ;HW SWITCH REG THERE?
003522 003522 001404      ERR12: BEQ      .+12         ;NO, SKP HW CHECK
003524 003524 032777 100000 175260      ERR12: BIT      #100000,@HWSWR ;YES, CK HW REG
003532 003532 001003      ERR12: BNE      .+10         ;IF SET, SKP SW REG CK
003534 003534 032737 100000 000176      ERR12: BIT      #100000,SWREG ;OTHERWISE, ALSO CK SW REG
003542 003542 001402      ERR12: BEQ      CN12        ;CONTINUE IF BIT 15 = 0
003544 003544 000137 003052      ERR12: JMP      TEST2       ;OTHERWISE LOOP
003550      CN12:
003550 003550 000137 003604      ERR12: JMP      79$
                                ;EXIT
003554      WEX6:
1243
1244 003554 117737 175320 004464      WEX6:  MOVB     @DLRBUF,TYPDAT ;Character from printer
1245 003562 004737 004400      WEX6:  JSR      #7,TYPD      ;Type it
1246 003566 022737 000015 004464      WEX6:  CMP      #15,TYPDAT   ; CR?
1247 003574 001304      WEX6:  BNE      67$         ;No, loop
1248
1249 003576 005337 001062      WEX6:  DEC      LINCNT      ;Bump line count
1250 003602 001301      WEX6:  BNE      67$         ;Loop if not done
1251
1252
1253 003604      WEX6:
1254 003604      WEX6:  69$:
                                79$:  $TSWRG  #2
                                ;CK SW REG

```

	003604	022737	000176	001004	CMP	#176,SWR	;HW SWITCH REG THERE?
	003612	001404			BEQ	+.12	;NO, SKP HW CHECK
	003614	032777	000002	175170	BIT	#2,@HWSWR	;YES, CK HW REG
	003622	001003			BNE	+.10	;IF SET, SKP SW REG CK
	003624	032737	000002	000176	BIT	#2,SWREG	;OTHERWISE, ALSO CK SW REG
1255	003632	001402			BEQ	TEST3	;CONTINUE IF = 0
1256	003634	000137	003052		JMP	TEST2	;IF SET, LOOP ON TEST
1257							
1258							

```

1260 .SBTTL TEST 3 PRINTER DISPLAY TEST
1261 ;*****
1262 ;
1263 ;
1264 ; Test 3 Display Test
1265 ;
1266 ; -A Send "Escape Sequence" to color printer
1267 ; to initiate Pattern display
1268 ;
1269 003640 005737 001064 TEST3: TST CYCCNT ;CK PASS COUNT
1270 003644 001416 BEQ 82$ ;IF 1ST PASS, EXECUTE TEST ALWAYS
1271 ;
1272 003646 $TSWRG #4 ;ONLY IF LOOP ON TST BIT, CONTINUE
003646 022737 000176 001004 CMP #176,SWR ;HW SWITCH REG THERE?
003654 001404 BEQ .+12 ;NO, SKP HW CHECK
003656 032777 000004 175126 BIT #4,@HWSWR ;YES, CK HW REG
003664 001003 BNE .+10 ;IF SET, SKP SW REG CK
003666 032737 000004 000176 BIT #4,SWREG ;OTHERWISE, ALSO CK SW REG
1273 003674 001002 BNE 82$ ;SET, CONTINUE TEST
1274 003676 000137 004070 JMP TSEND ;OTHERWISE, EXIT>>>>>>>>>
1275 ;
1276 003702 82$: $TSWRG #40000 ;CK SW REG
003702 022737 000176 001004 CMP #176,SWR ;HW SWITCH REG THERE?
003710 001404 BEQ .+12 ;NO, SKP HW CHECK
003712 032777 040000 175072 BIT #40000,@HWSWR ;YES, CK HW REG
003720 001003 BNE .+10 ;IF SET, SKP SW REG CK
003722 032737 040000 000176 BIT #40000,SWREG ;OTHERWISE, ALSO CK SW REG
1277 003730 001002 BNE 83$ ;IF SET, SKIP TST # HDR MSG
1278 003732 $PRTSN 3 ;PRINT TEST NUMBER
003732 104000 EMT .0 ;TEST NUMBER MESSAGE
003734 005673 TY03
1279 ;
1280 ; DELAY FOR A SEC
1281 ;
1282 ;
1283 003736 012737 000004 001124 83$: MOV #4,WORK ;2 SEC CNTR
1284 003744 012737 177777 001126 84$: MOV #-1,WORKA ;LOAD COUNT
1285 003752 005337 001124 DEC WORK ;BUMP CNTR
1286 003756 001404 BEQ 87$ ;TIMEOUT
1287 003760 005337 001126 86$: DEC WORKA ;BUMP COUNT A
1288 003764 001767 BEQ 84$ ;LOOP IF COUNT 0
1289 003766 000774 BR 86$ ;LOOP TIL DONE
1290 ;
1291 ; SEND "ESCAPE" SEQUENCE TO THE PRINTER
1292 ;
1293 ;
1294 003770 87$: $PRINT LCP4 ;INVOKE DISPLAY TEST
1295 ;
1296 003770 012737 005533 004312 MOV #LCP4,PRMSG ;LOAD MESSAGE ADDRESS
003776 004437 004250 JSR #4,PRINE ;PRINT IT
1297 ;
1298 ; DELAY FOR A COUPLE OF MINUTES
1299 ;
1300 ;
1301 004002 012737 000400 001124 93$: MOV #400,WORK ;TWO MIN CNTR
1302 004010 012737 177777 001126 94$: MOV #-1,WORKA ;LOAD COUNT

```

```
1303 004016 005337 001124          DEC      WORK          ;BUMP CNTR
1304 004022 001404                   BEQ      99$           ;TIMEOUT
1305 004024 005337 001126          96$:    DEC      WORKA        ;BUMP COUNT A
1306 004030 001767                   BEQ      94$           ;LOOP IF COUNT 0
1307 004032 000774                   BR       96$           ;LOOP TIL DONE
1308
1309
1310 004034                   ;
      004034 022737 000176 001004    99$:    $TSWRG #4          ;CK SW REG
      004042 001404                   CMP      #176,SWR      ;HW SWITCH REG THERE?
      004044 032777 000004 174740    BEQ      .+12          ;NO, SKP HW CHECK
      004052 001003                   BIT      #4,@HWSWR     ;YES, CK HW REG
      004054 032737 000004 000176    BNE      .+10          ;IF SET, SKP SW REG CK
1311 004062 001402                   BIT      #4,SWREG      ;OTHERWISE, ALSO CK SW REG
1312 004064 000137 003640          BEQ      TSEND         ;CONTINUE IF = 0
1313                                JMP      TEST3         ;IF SET, LOOP ON TEST
1314
1315                                .DSABL LSB
```

```

1317
1318
1319
1320
1321
1322
1323 004070 005237 001064
1324 004074
      004074 022737 000176 001004
      004102 001404
      004104 032777 040000 174700
      004112 001003
      004114 032737 040000 000176
1325 004122 001045
1326 004124 004537 004466
1327 004130 001064
1328 004132 006475
1329 004134 000004
1330 004136 104000
1331 004140 006457
1332 004142
      004142 022737 000176 001004
      004150 001404
      004152 032777 010000 174632
      004160 001003
      004162 032737 010000 000176
1333 004170 001002
1334 004172 000137 001464
1335
1336 004176 104000
1337 004200 006416
1338 004202 105777 174614
1339 004206 100375
1340 004210 117700 174602
1341 004214 142700 000140
1342 004220 122700 000004
1343 004224 001004
1344 004226 004737 014066
1345
1346 004232 000137 001306
1347
1348 004236 000137 001464
1349

```

```

.SBTTL
.SBTTL END OF TEST SEQUENCE
:*****
:
:   END OF TEST SEQUENCE. WAIT FOR KEY INPUT.
:
TSEND:  INC      CYCCNT      ;BUMP PASS COUNT
        $TSWRG  #40000      ;CK SW REG
        CMP     #176,SWR    ;HW SWITCH REG THERE?
        BEQ     .+12        ;NO, SKP HW CHECK
        BIT     #40000,@HWSR ;YES, CK HW REG
        BNE     .+10        ;IF SET, SKP SW REG CK
        BIT     #40000,SWREG ;OTHERWISE, ALSO CK SW REG
        BNE     TSRST       ;DON'T PRINT IF = SET
        JSR     #5,CONV     ;CONVERT PASS #
        CYCCNT
        MES9
        &
        .MT      +0
        MES8      ;PRINT IT
        $TSWRG  #10000     ;CK SW REG
        CMP     #176,SWR   ;HW SWITCH REG THERE?
        BEQ     .+12        ;NO, SKP HW CHECK
        BIT     #10000,@HWSR ;YES, CK HW REG
        BNE     .+10        ;IF SET, SKP SW REG CK
        BIT     #10000,SWREG ;OTHERWISE, ALSO CK SW REG
        BNE     TSEAB      ;PAUSE IF = SET
        JMP     TEST1      ;LOOP TO BEGINING OF TEST

TSEAB:  EMT      +0
        MES7
TSEDA:  TSTB     @TKS
        BPL     TSEDA      ;ANY KEY TO RESTART
        MOVB    @TKB,RO    ;KEY INPUT?
        BICB    #140,RO    ;IF NOT LOOP
        CMPB    #4,RO      ;GET INPUT CHAR
        BNE     TSRST      ;CLR UPPER 2 BITS
        JSR     #7,CLRTTY  ;CTRL D, S, D, OR d
                          ;IF NOT
                          ;CLR PENDING CHARACTERS

TSRSS:  JMP      SETUP
TSRST:  JMP      TEST1

```

```

1351 .SBTTL MISC. SUBROUTINES
1352 ;*****
1353 ;
1354 ; MISC. ROUTINES
1355 ;
1356 ;*****
1357 ;
1358 ; ROUTINE TO OUTPUT ASCII MESSAGES ON THE LINE PRINTER
1359 ;
1360 004242 012737 006570 004312 PRNT: MOV #MES14,PRMSG ;PRINT TEST NUMBER
1361 004250 004737 014122 PRINE: JSR #7,ERCHK ;TEST FOR ERROR
1362 004254 100007 BPL RINT ;BRANCH IF OK
1363 004256 104000 EMT +0
1364 004260 007305 ERMS6 ;STATUS ERROR
1365 004262 012737 000013 001136 ERR13: MOV #13, ERRCOUNT ;SET UP ERROR COUNT 13
1366 004270 004537 004606 JSR #5,STAER ;REPORT ERROR SET
1367 004274 013737 001000 001020 RINT: MOV LPS,TPS ;SET VECTORS -
1368 004302 013737 001002 001014 MOV LPB,TPB ;TO PRINT ON LINE PRINTER
1369 004310 104000 EMT +0 ;PRINT
1370 004312 006570 PRTMSG: MES14 ;MESSAGE
1371 004314 012737 177564 001020 TYPINT: MOV #177564,TPS ;RESET VECTORS
1372 004322 012737 177566 001014 MOV #177566,TPB ;FOR TTY
1373 004330 000204 RTS #4 ;RETURN
1374 ;*****
1375 ;
1376 ; INTERRUPT CALLED ROUTINE TO OUTPUT ASCII MESSAGES ON TELETYPE PRINTER
1377 ;
1378 ; EMT +0
1379 ; POINTER TO MESSAGE
1380 ;
1381 004332 011600 TYP: MOV @#6,#0 ;GET ADDR. THAT CONTAINS MESS.
1382 004334 062716 000002 ADD #2,@#6 ;SET UP EXIT
1383 004340 011000 MOV @#0,#0 ;ADDRESS OF MESSAGE IN RO
1384 004342 112037 004464 TYPA: MOVB (0),TYPDAT ;GET CHARACTER
1385 004346 001001 BNE TYPC ;BRANCH IF NOT DONE
1386 004350 000002 RTI ;EXIT
1387 004352 122737 000045 004464 TYPC: CMPB #45,TYPDAT ;CHECK FOR "*"
1388 004360 001426 BEQ TYPF ;BRANCH IF "*"
1389 004362 122737 000043 004464 CMPB #43,TYPDAT ;CHECK FOR "@"
1390 004370 001427 BEQ TYPG ;BRANCH IF "@"
1391 004372 004737 004400 JSR #7,TYPD ;TYPE CHARACTER IN TYPDAT
1392 004376 000761 BR TYPA ;NEXT CHAR IN MESSAGE
1393 004400 113777 004464 174406 TYPD: MOVB TYPDAT,@TPB ;OUTPUT CHARACTER TO PRINTER
1394 004406 023737 001020 001120 CMP TPS,DZCSRH ;ARE WE REALLY TALKING TO A DZ
1395 004414 001004 BNE TYPD0 ;BR, IF NOT DZ
1396 004416 005777 174376 TST @TPS ;IF DZ BIT 15 IS READY BIT NOT BIT 7
1397 004422 100375 BPL .-4 ;WAIT UNTIL DONE IS SET
1398 004424 000403 BR TYPD01 ;SKIP OTHER "TSTB"
1399 004426 105777 174366 TYPD0: TSTB @TPS
1400 004432 100375 BPL .-4
1401 004434 000207 TYPD01: RTS #7 ;CHAR. TYPED EXIT
1402 004436 112737 000012 004464 TYPF: MOVB #12,TYPDAT ;OUTPUT LF
1403 004444 004737 004400 JSR #7,TYPD ;GO TYPE CHAR.
1404 004450 112737 000015 004464 TYPG: MOVB #15,TYPDAT ;OUTPUT CR
1405 004456 004737 004400 JSR #7,TYPD ;GO TYPE CHAR.

```



```

1406 004462 000727
1407 004464 000000
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419 004466 010137 004600
1420 004472 010237 004602
1421 004476 010337 004604
1422
1423 004502 013537 004576
1424 004506 012501
1425 004510 012502
1426 004512 060201
1427 004514 013703 004576
1428 004520 042703 177770
1429 004524 062703 000060
1430 004530 110341
1431 004532 000241
1432 004534 006037 004576
1433 004540 000241
1434 004542 006037 004576
1435 004546 000241
1436 004550 006037 004576
1437 004554 005302
1438 004556 001356
1439 004560 013701 004600
1440 004564 013702 004602
1441 004570 013703 004604
1442
1443 004574 000205
1444
1445 004576 000000
1446 004600 000000
1447 004602 000000
1448 004604 000000
1449
1450
1451
1452
1453
1454 004606 010537 004752
1455 004612 004537 004466
1456 004616 001136
1457 004620 006104
1458 004622 000003
1459 004624
      004624 022737 000176 001004
      004632 001404
      004634 032777 020000 174150

```

```

          BR      TYPDAT: 0
          TYPDA
          ;*****
          ;ROUTINE TO CONVERT OCTAL TO ASCII
          ;
          ;ENTER ROUTINE AS FOLLOWS
          ;   JSR      #5,CONV
          ;   XXXXXX=ADDRESS OF NUMBER TO BE CONVERTED
          ;   XXXXXX=ADDRESS OF ASCII MESSAGE
          ;   XXXXXX=NUMBER OF OCTAL NO.'S TO BE CONVERTED
          ;
CONV:     MOV      R1,CONR1      ;SAVE REG
          MOV      R2,CONR2      ;SAVE REG
          MOV      R3,CONR3      ;SAVE REG
          ;
          MOV      @5+,ACNVX     ;ADDRS OF NO. TO BE CONVERTED
          MOV      (5)+,#1       ;ADDRESS OF MESSAGE
          MOV      (5)+,#2       ;NUMBER OF ASCII CHARACTERS
          ADD      #2,#1         ;FIRST CHAR ADDRESS
ACVN:     MOV      ACNVX,#3      ;STORE NUMBER
          BIC      #177770,#3    ;ISOLATE LEAST SIGNIFICANT BIT
          ADD      #60,#3        ;SET UP ASCII CHARACTER
          MOVB    #3,-(1)       ;STORE CHARACTER
          CLC
          ROR     ACNVX         ;GET NEXT SIGNIFICANT BIT ...
          CLC
          ROR     ACNVX
          CLC
          ROR     ACNVX
          DEC     #2            ;-1 FROM ASCII CHAR. CNT
          BNE     ACVN          ;CONVERT NEXT CHARACTER
          MOV     CONR1,R1      ;RESTORE REG
          MOV     CONR2,R2      ;RESTORE REG
          MOV     CONR3,R3      ;RESTORE REG
          RTS      #5          ;EXIT! CONVERSION DONE
          ;
ACNVX:    0                    ;WORK REGISTER
CONR1:    0
CONR2:    0
CONR3:    0
          ;*****
          ;ROUTINE TO REPORT ERROR COUNT
          ;
STAER:    MOV      #5,STARN     ;SAVE R5
          JSR      #5,CONV     ;CONVERT OCTAL TO ASCII
          ERCOUNT
          HED2
          3
          $TSWRG #20000        ;CK SW REG
          CMP     #176,SWR     ;HW SWITCH REG THERE?
          BEQ     +12          ;NO, SKP HW CHECK
          BIT     #20000,@HWSR ;YES, CK HW REG

```

```

004642 001003
004644 032737 020000 000176      BNE      .+10      ;IF SET, SKP SW REG CK
1460      BIT      #20000,SWREG      ;OTHERWISE, ALSO CK SW REG
1461 004652 001002      BNE      .+6      ;TEST FOR INHIBIT ERROR MSG
1462 004654 104000      EMT      +0      ;BRANCH IF SET
1463 004656 000063      HED1      ;TYPE ERROR MESSAGE
1464
1465 004660      $TSWRG #10000      ;CK SW REG
004660 022737 000176 001004      CMP      #176,SWR      ;HW SWITCH REG THERE?
004666 001404      BEQ      .+12      ;NO, SKP HW CHECK
004670 032777 010000 174114      BIT      #10000,@HWSWR ;YES, CK HW REG
004676 001003      BNE      .+10      ;IF SET, SKP SW REG CK
004700 032737 010000 000176      BIT      #10000,SWREG ;OTHERWISE, ALSO CK SW REG
1466      BEQ      STEXT      ;TEST FOR HALT ON ERROR
1467 004706 001416      :      ;BRANCH IF NO HALT WANTED
1468      HALT      ;HALT ON ERROR
1469 004710 104000      EMT      +0      ;PRINT IT
1470 004712 006504      MES10
1471
1472 004714 105777 174102      STEDA: TSTB @TKS      ;KEY INPUT?
1473 004720 100375      BPL      STEDA      ;IF NOT LOOP
1474 004722 117705 174070      MOVB @TKB,R5      ;GET INPUT CHAR
1475 004726 142705 000140      BICB #140,R5      ;CLR UPPER 2 BITS
1476 004732 122705 000004      CMPB #4,R5      ;CTRL D, $, D, OR d
1477 004736 001002      BNE      STEXT      ;IF NOT
1478 004740 004737 014066      JSR      #7,CLRTTY ;CLR PENDING CHARACTERS
1479
1480 004744 013705 004752      STEXT: MOV      STARN,#5 ;RESTORE R5
1481 004750 000205      RTS      #5      ;RETURN
1482      .EVEN
1483 004752 000000      STARN: .WORD 0      ;RETURN STORAGE

```

```

1485
1486
1487
1488
1489
1490
1491
1492 004754 010046
1493 004756 010146
1494 004760 010246
1495 004762 010346
1496 004764 010446
1497 004766 010546
1498
1499
1500
1501 004770 017737 174022 001160
1502 004776 042737 177600 001160
1503 005004 022737 000007 001160
1504 005012 001527
1505 005014 022737 000015 001160
1506 005022 001466
1507 005024 022737 000025 001160
1508 005032 001557
1509 005034 022737 000010 001160
1510 005042 001576
1511
1512 005044 022737 000003 001160
1513 005052 001564
1514
1515 005054 023727 001160 000060
1516 005062 100001
1517 005064 000466
1518 005066 022737 000067 001160
1519 005074 100001
1520 005076 000461
1521 005100 005237 001152
1522 005104 022737 000006 001152
1523 005112 100453
1524 005114 105777 173700
1525 005120 100375
1526 005122 013777 001160 173664
1527 005130 162737 000060 001160
1528 005136 022737 000001 001152
1529 005144 001411
1530 005146 000241
1531 005150 006137 001162
1532 005154 000241
1533 005156 006137 001162
1534 005162 000241
1535 005164 006137 001162
1536 005170 063737 001160 001162
1537 005176 000503
1538 005200 005737 001152
1539 005204 001470
1540 005206 013737 001162 000176
1541 005214 000464

.SBTTL KEYBOARD INTERRUPT ROUTINE
;*****
;KEYBOARD INTERRUPT ROUTINE
;FOR ACCESS TO THE S/W SWITCH REGISTER
;
TKINT:  MOV    %0,-(SP)      ;SAVE REGISTERS
        MOV    %1,-(SP)      ;
        MOV    %2,-(SP)      ;
        MOV    %3,-(SP)      ;
        MOV    %4,-(SP)      ;
        MOV    %5,-(SP)      ;
;
:::000  TST    SIGNAL        ;PREVIOUS CONTROL-G INPUT ?
:::000  BEQ    CNTRLG        ;'O-
        MOV    @TKB,CHAR     ;GET INPUT CHARACTER
        BIC    #177600,CHAR  ;STRIP OFF PARITY BIT
        CMP    #7,CHAR       ;CONTROL-G INPUT?
        BEQ    TYP5WR        ;YES-PRINT HEADER
        CMP    #15,CHAR      ;CARRIAGE RETURN ?
        BEQ    DGTS          ;YES-TERMINATE SWR CHANGE
        CMP    #25,CHAR      ;CONTROL-U INPUT ?
        BEQ    TK4           ;YES-CLEAR PREVIOUS ENTRY
        CMP    #10,CHAR      ;CONTROL-H INPUT ?
        BEQ    TKHLP         ;YES-PRINT HELP
;
        CMP    #3,CHAR       ;CONTROL-C INPUT ?
        BEQ    TK9           ;YES-ABORT
;
        CMP    CHAR,#60      ;ILLEGAL # CHECK: LESS THAN 60 ?
        BPL    TK1           ;NO-CONTINUE
        BR     WT3           ;YES-PRINT "?"
        CMP    #67,CHAR      ;ILLEGAL # CHECK: GREATER THAN 67 ?
        BPL    TK2           ;NO-CONTINUE
        BR     WT3           ;YES-PRINT "?"
        INC    DIGITS        ;NEXT DIGIT OF SWR INPUT
        CMP    #6,DIGITS     ;MORE THAN SIX DIGITS ?
        BMI    WT3           ;YES-PRINT "?"
        TSTB   @TPS          ;TTY PRINTER READY ?
        BPL    WT2           ;NO-WAIT
        MOV    CHAR,@TPB     ;PRINT CHARACTER
        SUB    #60,CHAR      ;CONVERT TO OCTAL
        CMP    #1,DIGITS     ;FIRST DIGIT ?
        BEQ    TK5           ;YES-CONTINUE
        CLC                    ;ROTATE LEFT THREE
        ROL    OCT           ;TIMES
        CLC                    ;THIS WILL SHIFT
        ROL    OCT           ;SWR VALUE ONE
        CLC                    ;PLACE LEFT
        ROL    OCT           ;OCTAL.
        ADD    CHAR,OCT      ;NEW VALUE OF SWR
        BR     TK6           ;RETURN FROM INTERRUPT
        TST    DIGITS        ;SWR VALUE CHANGED ?
        BEQ    TK3           ;NO-RETURN ,NO CHANGE TO SWR
        MOV    OCT,SWREG     ;YES-ENTER NEW SWR VALUE
        BR     TK3           ;RETURN FROM INTERRUPT

```

```

1542 005216 017737 173574 001160 CNTRLG: MOV @TKB,CHAR ;GET CHARACTER
1543 005224 042737 177600 001160 BIC #177600,CHAR ;STRIP OFF PARITY BIT
1544 005232 022737 000007 001160 CMP #7,CHAR ;CONTROL-G INPUT ?
1545 005240 001414 BEQ TYP5WR ;YES-PRINT HEADER
1546 005242 105777 173552 WT3: TSTB @TPS ;TTY PRINTER READY ?
1547 005246 100375 BPL WT3 ;NO-WAIT
1548 005250 013777 001160 173536 MOV CHAR,@TPB ;PRINT CHARACTER
1549 005256 104000 EMT +0 ;PRINT "?"
1550 005260 006614 MES22
1551 005262 005737 001154 TST SIGNAL ;BAD VALUE ?
1552 005266 001001 BNE TYP5WR ;YES-PRINT HEADER
1553 005270 000442 BR TK7 ;RETURN FROM INTERRUPT
1554 005272 012737 000001 001154 TYP5WR: MOV #1,SIGNAL ;SET FLAG: CONTROL-G ENTERED
1555 005300 104000 EMT +0
1556 005302 006612 MES21
1557 005304 022737 000176 001004 CMP #176,SWR ;CR
1558 005312 001411 BEQ TYP5WX ;H/W SW REG THERE?
1559 005314 104000 EMT +0 ;NO, SKIP H/W REG DUMP
1560 005316 006656 MES26
1561 005320 004537 004466 JSR #5,CONV ;H/W SW REG HDR
1562 005324 177570 177570 MES27
1563 005326 006672 6
1564 005330 000006 EMT +0
1565 005332 104000 MES27
1566 005334 006672 TYP5WX: EMT +0 ;PRINT HEADER
1567 005336 104000 MES23
1568 005340 006620 JSR #5,CONV ;CONVERT SWR VALUE TO ASCII
1569 005342 004537 004466 176
1570 005346 000176 MES25
1571 005350 006647 6
1572 005352 000006 EMT +0 ;PRINT SWR VALUE
1573 005354 104000 MES25
1574 005356 006647 EMT +0 ;PRINT HEADER
1575 005360 104000 MES24
1576 005362 006630 BR TK7
1577 005364 000404 TK3: CLR SIGNAL ;RETURN FROM INTERRUPT
1578 005366 005037 001154 TK4: EMT +0 ;CLEAR CONTROL-G FLAG
1579 005372 104000 MES21 ;PRINT LINE FEED AND CARRIAGE RETURN
1580 005374 006612 TK7: CLR DIGITS ;CLEAR DIGIT COUNT
1581 005376 005037 001152 CLR OCT ;CLEAR SWR INPUT
1582 005402 005037 001162 TK6: MOV (SP)+,#5 ;RESTORE REGISTERS
1583 005406 012605 MOV (SP)+,#4
1584 005410 012604 MOV (SP)+,#3
1585 005412 012603 MOV (SP)+,#2
1586 005414 012602 MOV (SP)+,#1
1587 005416 012601 MOV (SP)+,#0
1588 005420 012600 RTI ;RETURN FROM INTERRUPT
1589 005422 000002
1590
1591 005424 012746 000340 TK9: MOV #340,-(SP) ;ABORT AND GO TO 200, RESTART
1592 005430 012746 005440 MOV #.+10,-(SP)
1593 005434 000137 001306 JMP SETUP
1594
1595 ; Help - reached by <ctrl>H
1596
1597 005440 104000 TKHLP: EMT +0
1598 005442 007350 HLP0

```

1599	005444	104000	EMT	+0
1600	005446	007420	HLP1	
1601	005450	104000	EMT	+0
1602	005452	007514	HLP2	
1603	005454	104000	EMT	+0
1604	005456	007541	HLP3	
1605	005460	104000	EMT	+0
1606	005462	007611	HLP4	
1607	005464	104000	EMT	+0
1608	005466	007672	HLP4A	
1609	005470	104000	EMT	+0
1610	005472	007752	HLP5	
1611	005474	104000	EMT	+0
1612	005476	010040	HLP6	
1613	005500	104000	EMT	+0
1614	005502	010100	HLP7	
1615	005504	104000	EMT	+0
1616	005506	010163	HLP8	
1617	005510	000726	BR	TK3
1618				

```

1620 .SBTTL LOCAL MESSAGES
1621 .NLIST BEX
1622 .EVEN
1623 ;*****
1624 ;
1625 ;   DEVICE "ESCAPE SEQUENCES"
1626 ;
1627 ;   AND OPERATOR MESSAGES.
1628 ;
1629 005512      033      133      143  DAR1:  .ASCIZ  <33>\[c\           ;What are you?
1630 005516      033      133      060  DAR2:  .ASCIZ  <33>\[0c\          ;What are you?
1631 005523      033      133      077  LCP3:  .ASCIZ  <33>\[??;0c\       ;<answer> I am a LCPO (Color printer).
1632 005533      033      133      066  LCP4:  .ASCIZ  <33>\[6;2y\       ;Display pattern
1633 005542      033      133      066  LCP5:  .ASCIZ  <33>\[6;1y\       ;Run Confidence test on clr ptr.
1634 005551      033      133      065  DAR6:  .ASCIZ  <33>\[5n\          ;Report your status.
1635 005556      126      105      122  LCP7:  .ASCIZ  \VERIFIED\        ;<answer> I'm OK (Color Printer).
1636 005567      021      000      DAR9:  .ASCIZ  <21>                ;XON
1637 005571      045      124      105  TY01:  .ASCIZ  \*TEST 1, SERIAL LINE UNIT TEST\
1638 005630      045      124      105  TY02:  .ASCIZ  \*TEST 2, COLOR PRINTER 'SELF' TEST\
1639 005673      045      124      105  TY03:  .ASCIZ  \*TEST 3, PRINTER DISPLAY TEST\
1640
1641
1642 005731      045      105      122  ETIM:  .ASCIZ  \*ERROR - TIMEOUT, WAITING FOR INPUT, RESPONSE\
1643 006007      045      105      122  ETIMO: .ASCIZ  \*ERROR - TIMEOUT, WAITING FOR OUTPUT, DONE$\
1644 006063      045      040      055  HED1:  .ASCII  /* - ERROR NUMBER /
1645 006104      040      040      040  HED2:  .ASCIZ  / * /
1646 006114      045      123      124  MESDD: .ASCIZ  /*STARTING EVFU PRINTING TESTS*/
1647 006153      045      045      103  MES1:  .ASCIZ  \*CZLCP-A-0\
1648 006167      045      103      132  MES2:  .ASCIZ  \*CZLCP0 COLOR PRINTER DIAGNOSTIC\
1649 006230      045      125      116  MES3:  .ASCIZ  \*UNIT IS TEX*\
1650 006246      045      122      105  MES4:  .ASCIZ  \*RESTART ADDRESS 200*\
1651 006274      045      116      157  MES5:  .ASCIZ  \*Now begins the Color Printer Diagnostic*\
1652 006347      045      103      157  MES6:  .ASCIZ  \*Color Printer Diagnostic Completed*\
1653 .EVEN
1654 006416      045      120      162  MES7:  .ASCIZ  \*Press Any Key to Restart test..\
1655 006457      045      105      116  MES8:  .ASCII  \*END OF PASS: \
1656 006475      040      040      040  MES9:  .ASCIZ  \ * \
1657 006504      045      120      101  MES10: .ASCIZ  \*PAUSE (HALT) ON ERROR, Press Any Key to Continue..\
1658
1659
1660 006570      012      012      124  MES14: .ASCIZ  <12><12>\TEST NUMBER \
1661 006607      040      040      000  MES20A: .ASCIZ  / /
1662 .EVEN
1663 006612      045      000      MES21: .ASCIZ  /*/
1664 006614      040      077      045  MES22: .ASCIZ  / ?*/
1665 006620      123      127      122  MES23: .ASCIZ  /*SWR = /
1666 006630      040      040      040  MES24: .ASCIZ  / NEW SWR = /
1667 006647      040      040      040  MES25: .ASCIZ  / /
1668 006656      050      110      057  MES26: .ASCIZ  \ (H/W SWR = \
1669 006672      040      040      040  MES27: .ASCIZ  \ ) , \
1670 .EVEN
1671 006704      045      105      122  ERMS1: .ASCIZ  \*ERROR - SERIAL LINE NOT AT THIS ADDRESS*\
1672 006756      045      105      122  ERMS2: .ASCIZ  \*ERROR - SERIAL LINE "LOOPBACK" FAILED*\
1673 007026      045      105      122  ERMS3: .ASCIZ  \*ERROR - UNEXPECTED RESPONSE, TO "WHAT ARE YOU?" REQUEST*\
1674 007120      045      105      122  ERMS4: .ASCIZ  \*ERROR - UNEXPECTED RESPONSE, TO "POWER-UP" (SELF) TEST*\
1675 007211      045      105      122  ERMS5: .ASCIZ  \*ERROR - UNEXPECTED RESPONSE, TO "POWER-UP" STATUS REQUEST*\
1676 007305      045      117      125  ERMS6: .ASCIZ  \*OUTPUT, SERIAL LINE STATUS ERROR*\
    
```

```
1677
1678 007350 045 045 040 hlp0: .asciz \*\ HELP, Switch Register Bit definition\
1679 007420 045 040 040 hlp1: .asciz \* (Note: <CTRL>G - Allows change to "Software" Switch Reg)\
1680 007514 045 061 065 hlp2: .asciz \*15)14,13,12...2,1,0\
1681 007541 045 040 174 hlp3: .asciz \* ) ) ) ) ) ) )_Loop on SLU Test 1\
1682 007611 045 040 174 hlp4: .asciz \* ) ) ) ) ) ) )_Loop on Printer "Self" Test 2\
1683 007672 045 040 174 hlp4a: .asciz \* ) ) ) ) ) ) )_Loop on Printer Display Test 3\
1684 007752 045 040 174 hlp5: .asciz \* ) ) ) )_Pause on Error, and Pause at End of Pass\
1685 010040 045 040 174 hlp6: .asciz \* ) ) )_Inhibit Error Reports\
1686 010100 045 040 174 hlp7: .asciz \* ) )_Inhibit Test Number and End of Pass Reports\
1687 010163 045 040 174 hlp8: .asciz \* )_Loop On Error (otherwise continue)\
```

```

1689 .SBTTL SERIAL LINE SETUP ROUTINES
1690 .EVEN
1691 ;*****
1692 ;
1693 ; THIS SUBROUTINE SETS UP THE SERIAL LINE INTERFACE IF THERE IS ONE
1694 ;
1695 ;*****
1696 010234 SETSER:
1697 010234 005037 001120 CLR DZCSRH ;CLEAR DZ'S CSR HOLDER
1698 010240 005037 001066 CLR DZTCR ;CLEAR PSEUDO DZ TCR REG
1699 010244 005037 001070 CLR DZLPR ;CLEAR PSEUDO DZ LPR REG
1700 010250 005037 001106 CLR DLHERE ;CLEAR DL'S PRESENCE AREA
1701 010254 005037 001102 CLR BRATE ;CLEAR DZ'S BAUD RATE HOLDER
1702 010260 20$: ; CMP SERSW,#1 ;HAVE WE ALREADY SET UP SLU
1703 ; BNE 30$ ;BR, NOT SET UP
1704 ; JMP 200$ ;LEAVE ROUTINE ALREADY SET UP
1705 010260 042777 000100 170534 30$: BIC #100,@TKS ;SHUT OFF IE IN TTY
1706 ; MOV #1,SERSW ;SET SW TO BEEN SET UP
1707 010266 012737 000010 001122 MOV #10,MAINTB ;PRELOAD BIT 3 SET
1708 010274 104000 EMT +0 ;PRINT MESSAGE CALL
1709 010276 015723 MENU1 ;PRINT MESSAGE CALL
1710 010300 104000 EMT +0 ;PRINT MESSAGE CALL
1711 010302 015760 MENU2 ;PRINT MESSAGE CALL
1712 010304 104000 EMT +0 ;PRINT MESSAGE CALL
1713 010306 016006 MENU3 ;PRINT MESSAGE CALL
1714 010310 104000 EMT +0 ;PRINT MESSAGE CALL
1715 010312 016034 MENU4 ;PRINT MESSAGE CALL
1716 010314 105777 170502 40$: TSTB @TKS ;READ TTY'S STATUS
1717 010320 100375 BPL 40$ ;BR, IF NOT DONE
1718 010322 117700 170470 MOVB @TKB,#0 ;PICK UP CHAR. TYPED
1719 010326 105777 170466 42$: TSTB @TPS ;CHECK FOR BUSY
1720 010332 100375 BPL 42$ ;LOOP IF TTY IS BUSY (BR)
1721 010334 110077 170454 MOVB #0,@TPB ;ECHO CHARACTER TO TTY
1722 010340 042700 177700 BIC #177700,#0 ;ONLY 6 BITS ALLOWED
1723 010344 120027 000061 CMPB #0,#61 ;WAS IT AN ASCII 1
1724 010350 001422 BEQ 47$ ;BR, IF IT WAS (DL11 SEL.)
1725 010352 120027 000062 CMPB #0,#62 ;WAS IT AN ASCII 2
1726 010356 001007 BNE 45$
1727 010360 004737 013642 JSR #7,GETCR ;GO GET LF OR CR
1728 010364 020027 123456 CMP #0,#123456 ;CHECK FOR BAD INPUT
1729 010370 001733 BEQ 30$ ;TRY AGAIN BAD INPUT (NOT CR)
1730 010372 000137 011340 JMP 300$ ;BR, IF IT WAS (DZ11 SEL.)
1731 010376 122700 000012 45$: CMPB #12,#0 ;CHK FOR <LF> DEFAULT
1732 010402 001412 BEQ 49$ ;BR, IF DEFAULT (DL11 SEL.)
1733 010404 122700 000015 CMPB #15,#0 ;CHK FOR <CR> DEFAULT
1734 010410 001407 BEQ 49$ ;BR, IF DEFAULT (DL11 SEL.)
1735 010412 000137 010260 JMP 30$ ;WASN'T CORRECT TYPE-IN
1736 ;
1737 ; IT'S A DL11
1738 ;
1739 010416 004737 013642 47$: JSR #7,GETCR ;GO GET LF OR CR
1740 010422 020027 123456 CMP #0,#123456 ;CHECK FOR BAD INPUT
1741 010426 001714 BEQ 30$ ;TRY AGAIN BAD INPUT (NOT CR)
1742 010430 012737 000001 001106 49$: MOV #1,DLHERE ;SHOW DEC PRESENCE
1743 010436 005037 001102 CLR BRATE ;JUST TO BE SURE
1744 010442 004737 014066 JSR #7,CLRTTY ;PICK UP PENDING CHARACTERS
1745 010446 012737 000004 001122 MOV #4,MAINTB ;PRELOAD BIT # 2
    
```


1746	010454	104000				EMT	+0		
1747	010456	016070				MENUD1			;PRINT MESSAGE CALL
1748	010460	104000				EMT	+0		
1749	010462	016156				MENUD2			;PRINT MESSAGE CALL
1750	010464	104000				EMT	+0		
1751	010466	016221				MENUD3			;PRINT MESSAGE CALL
1752	010470	104000				EMT	+0		
1753	010472	016263				MENUD4			;PRINT MESSAGE CALL
1754	010474	105777	170322		51#:	TSTB	@TKS		;READ TTY'S STATUS
1755	010500	100375				BPL	51#		;BR, IF NOT DONE
1756	010502	117700	170310			MOVB	@TKB,#0		;PICK UP CHAR. TYPED
1757	010506	105777	170306		52#:	TSTB	@TPS		;CHECK FOR BUSY
1758	010512	100375				BPL	52#		;LOOP IF TTY IS BUSY (BR)
1759	010514	110077	170274			MOVB	#0,@TPB		;ECHO CHARACTER TO TTY
1760	010520	042700	177700			BIC	#177700,#0		;ONLY 6 BITS ALLOWED
1761	010524	120027	000061			CMPB	#0,#61		;WAS IT AN ASCII 1
1762	010530	001424				BEQ	54#		;BR, IF IT WAS (DL11 SEL.)
1763	010532	120027	000062			CMPB	#0,#62		;WAS IT AN ASCII 2
1764	010536	001011				BNE	53#		
1765	010540	004737	013642			JSR	#7,GETCR		;GO GET LF OR CR
1766	010544	020027	123456			CMP	#0,#123456		;CHECK FOR BAD INPUT
1767	010550	001727				BEQ	49#		;TRY AGAIN BAD INPUT (NOT CR)
1768	010552	005037	001122			CLR	MAINTB		;NO MAINT FEATURE
1769	010556	000137	010614			JMP	55#		;BR
1770	010562	122700	000012		53#:	CMPB	#12,#0		;CHK FOR <LF> DEFAULT
1771	010566	001412				BEQ	55#		;BR, IF DEFAULT (DL11 SEL.)
1772	010570	122700	000015			CMPB	#15,#0		;CHK FOR <CR> DEFAULT
1773	010574	001407				BEQ	55#		;BR, IF DEFAULT (DL11 SEL.)
1774	010576	000137	010430			JMP	49#		;WASN'T CORRECT TYPE-IN
1775									
1776	010602	004737	013642		54#:	JSR	#7,GETCR		;GO GET LF OR CR
1777	010606	020027	123456			CMP	#0,#123456		;CHECK FOR BAD INPUT
1778	010612	001706				BEQ	49#		;TRY AGAIN BAD INPUT (NOT CR)
1779	010614	012737	000001	001106	55#:	MOV	#1,DLHERE		;SHOW DEC PRESENCE
1780	010622	005037	001102			CLR	BRATE		;JUST TO BE SURE
1781	010626	004737	014066			JSR	#7,CLRTTY		;PICK UP PENDING CHARACTERS
1782									
1783	010632	104000				EMT	+0		;PRINT MESSAGE TO TTY
1784	010634	016317				DLCSRM			
1785	010636	105777	170160		56#:	TSTB	@TKS		;CHK TTY IN STATUS
1786	010642	100375				BPL	56#		;WAIT FOR DONE
1787	010644	117700	170146			MOVB	@TKB,#0		;PICK UP CHARACTER TYPED IN
1788	010650	105777	170144		57#:	TSTB	@TPS		;CHECK FOR BUSY
1789	010654	100375				BPL	57#		;LOOP IF TTY IS BUSY (BR)
1790	010656	110077	170132			MOVB	#0,@TPB		;ECHO CHARACTER
1791	010662	042700	177700			BIC	#177700,#0		;ONLY 6 BIT PASS
1792	010666	122700	000012			CMPB	#12,#0		;WAS DEFAULT SEL. <LF>
1793	010672	001551				BEQ	60#		;BR, IF DEFAULT SELECTED
1794	010674	122700	000015			CMPB	#15,#0		;WAS DEFAULT SEL. <CR>
1795	010700	001546				BEQ	60#		;BR, IF DEFAULT SELECTED
1796	010702	122700	000061			CMPB	#61,#0		;WAS IT AN ASCII 1
1797	010706	001403				BEQ	59#		;BR, IT WAS A 1
1798	010710	122700	000067			CMPB	#67,#0		;WAS IT AN ASCII 7
1799	010714	001332				BNE	54#		;BR, IF IT WASN'T A 1 OR 7
1800	010716	005037	001124		59#:	CLR	WORK		;CLEAR WORK AREA
1801	010722	052737	100000	001124		BIS	#100000,WORK		;ALL THAT FOR BIT 15
1802	010730	004737	013560			JSR	#7,GETOCT		;GO GET AN OCTAL NUMBER

1803	010734	020027	123456	CMP	%0,#123456	
1804	010740	001720		BEQ	54‡	;CHECK FOR NON-OCTAL #
1805	010742	000241		CLC		;BR, IN NO GOOD #
1806		000014		.REPT	12.	;CLEAR CARRY FOR ROTATE
1807				ROL	%0	
1808				.ENDR		;ROTATE RO
	010744	006100		ROL	%0	;ROTATE RO
	010746	006100		ROL	%0	;ROTATE RO
	010750	006100		ROL	%0	;ROTATE RO
	010752	006100		ROL	%0	;ROTATE RO
	010754	006100		ROL	%0	;ROTATE RO
	010756	006100		ROL	%0	;ROTATE RO
	010760	006100		ROL	%0	;ROTATE RO
	010762	006100		ROL	%0	;ROTATE RO
	010764	006100		ROL	%0	;ROTATE RO
	010766	006100		ROL	%0	;ROTATE RO
	010770	006100		ROL	%0	;ROTATE RO
	010772	006100		ROL	%0	;ROTATE RO
1809	010774	060037	001124	ADD	%0,WORK	;ROTATE RO
1810	011000	004737	013560	JSR	%7,GETOCT	; "OR" THE BITS IN
1811	011004	020027	123456	CMP	%0,#123456	;GO GET AN OCTAL NUMBER
1812	011010	001674		BEQ	54‡	;CHECK FOR NON-OCTAL #
1813	011012	000241		CLC		;BR, IN NO GOOD #
1814		000011		.REPT	9.	;CLEAR CARRY FOR ROTATE
1815				ROL	%0	
1816				.ENDR		;ROTATE RO
	011014	006100		ROL	%0	;ROTATE RO
	011016	006100		ROL	%0	;ROTATE RO
	011020	006100		ROL	%0	;ROTATE RO
	011022	006100		ROL	%0	;ROTATE RO
	011024	006100		ROL	%0	;ROTATE RO
	011026	006100		ROL	%0	;ROTATE RO
	011030	006100		ROL	%0	;ROTATE RO
	011032	006100		ROL	%0	;ROTATE RO
	011034	006100		ROL	%0	;ROTATE RO
1817	011036	060037	001124	ADD	%0,WORK	;ROTATE RO
1818	011042	004737	013560	JSR	%7,GETOCT	; "OR" THE BITS IN
1819	011046	020027	123456	CMP	%0,#123456	;GO GET AN OCTAL NUMBER
1820	011052	001653		BEQ	54‡	;CHECK FOR NON-OCTAL #
1821	011054	000241		CLC		;BR, IN NO GOOD #
1822		000006		.REPT	6	;CLEAR CARRY FOR ROTATE
1823				ROL	%0	
1824				.ENDR		;ROTATE RO
	011056	006100		ROL	%0	;ROTATE RO
	011060	006100		ROL	%0	;ROTATE RO
	011062	006100		ROL	%0	;ROTATE RO
	011064	006100		ROL	%0	;ROTATE RO
	011066	006100		ROL	%0	;ROTATE RO
	011070	006100		ROL	%0	;ROTATE RO
1825	011072	060037	001124	ADD	%0,WORK	;ROTATE RO
1826	011076	004737	013560	JSR	%7,GETOCT	; "OR" THE BITS IN
1827	011102	020027	123456	CMP	%0,#123456	;GO GET AN OCTAL NUMBER
1828	011106	001635		BEQ	54‡	;CHECK FOR NON-OCTAL #
1829	011110	000241		CLC		;BR, IN NO GOOD #
1830		000003		.REPT	3	;CLEAR CARRY FOR ROTATE
1831				ROL	%0	
1832				.ENDR		;ROTATE RO

011112	006100			ROL	#0					
011114	006100			ROL	#0					;ROTATE R0
011116	006100			ROL	#0					;ROTATE R0
1833	011120	060037	001124	ADD	#0,WORK					;ROTATE R0
1834	011124	004737	013560	JSR	#7,GETOCT					; "OR" THE BITS IN
1835	011130	020027	123456	CMP	#0,#123456					;GO GET AN OCTAL NUMBER
1836	011134	001622		BEQ	54					;CHECK FOR NON-OCTAL #
1837	011136	060037	001124	ADD	#0,WORK					;BR, IN NO GOOD #
1838	011142	013737	001124	MOV	WORK,DLLPS					; "OR" THE BITS IN
1839	011150	013737	001124	MOV	WORK,DLRBUF					;ADDRESS OF DL'S RECV STATUS
1840	011156	062737	000002	ADD	#2,DLRBUF					;ADDRESS OF DL'S RECV BUF
1841	011164	013737	001124	MOV	WORK,LPS					;ADDRESS OF DL'S RECV BUFFER
1842	011172	062737	000004	ADD	#4,LPS					;NEW ADDRESS FOR CSR
1843	011200	013737	001000	MOV	LPS,LPB					;MUS, POINT TO TRANSMITTER BUF
1844	011206	062737	000002	ADD	#2,LPB					;GET STATUS ADDRESS
1845	011214	000426		BR	65					;POINT TO DATA BUFFER ADDRESS
1846	011216	013737	001032	MOV	DLCSRC,LPS					;SKIP OVER DEFAULT
1847	011224	062737	000004	ADD	#4,LPS					;MOVE DEFAULT CSR IN
1848	011232	013737	001032	MOV	DLCSRC,DLLPS					;TRANSMITTER STATUS
1849	011240	013737	001032	MOV	DLCSRC,DLRBUF					;ADDRESS OF DL'S RECV STATUS
1850	011246	062737	000002	ADD	#2,DLRBUF					;ADDRESS OF DL'S RECV BUF
1851	011254	013737	001000	MOV	LPS,LPB					;ADDRESS OF DL'S RECV BUFFER
1852	011262	062737	000002	ADD	#2,LPB					;SET UP THE DATA BUFFER
1853	011270	000407		BR	67					;SET TO CORRECT ADDRESS
1854	011272	004737	013642	JSR	#7,GETCR					;SKIP OVER CR GET
1855	011276	020027	123456	CMP	#0,#123456					;GO GET LF OR CR
1856	011302	001004		BNE	66					;CHECK FOR BAD INPUT
1857	011304	000137	010602	JMP	54					;BR, IF CR RECEIVED (GOOD)
1858	011310	004737	014066	JSR	#7,CLRTTY					;JMP, IF BAD CHARACTER RECD.
1859	011314									;PICK UP PENDING CHARACTERS
1860	011314	004737	014364							
1861	011320	004737	014066	JSR	#7,DLSET					;FIND OUT WHAT TYPE DL11
1862	011324	052777	000100	JSR	#7,CLRTTY					;PICK UP PENDING CHARACTERS
1863	011332	104000		BIS	#100,@TKS					;TURN BACK ON
1864	011334	016543		EMT	.0					;MESSAGE ADDRESS
1865	011336	000207		DLCRLF						;MESSAGE ADDRESS
1866				RTS	#7					;RETURN TO CALLEE
1867										
1868										
1869	011340									
1870	011340	004737	014066							
1871	011344	104000		JSR	#7,CLRTTY					;PICK UP PENDING CHARACTERS
1872	011346	016431		EMT	.0					;PRINT MESSAGE TO TTY
1873	011350	005037	001106	DZCSRH						
1874	011354	105777	167442	CLR	DLHERE					
1875	011360	100375		TSTB	@TKS					;JUST TO BE SURE
1876	011362	117700	167430	BPL	355					;CHK TTY IN STATUS
1877	011366	105777	167426	MOVB	@TKB,#0					;WAIT FOR DONE
1878	011372	100375		TSTB	@TPS					;PICK UP CHARACTER TYPED IN
1879	011374	110077	167414	BPL	356					;CHECK FOR BUSY
1880	011400	042700	177700	MOVB	#0,@TPB					;LOOP IF TTY IS BUSY (BR)
1881	011404	122700	000012	BIC	#177700,#0					;ECHO CHARACTER
1882	011410	001545		CMPB	#12,#0					;ONLY 6 BIT PASS
1883	011412	122700	000015	BEQ	360					;WAS DEFAULT SEL. <LF>
1884	011416	001542		CMPB	#15,#0					;BR, IF DEFAULT SELECTED
1885	011420	122700	000061	BEQ	360					;WAS DEFAULT SEL. <CR>
1886	011424	001403		CMPB	#61,#0					;BR, IF DEFAULT SELECTED
				BEQ	357					;WAS IT AN ASCII 1
										;BR, IT WAS A 1

1887	011426	122700	000067		CMPB	#67,#0		
1888	011432	001342			BNE	350\$; WAS IT AN ASCII 7
1889	011434	042700	177770	357\$:	BIC	#177770,#0		; BR, IF IT WASN'T A 1 OR 7
1890	011440	005037	001124		CLR	WORK		; ONLY 3 BITS NEEDED
1891	011444	052737	100000	001124	BIS	#100000,WORK		; CLEAR WORK AREA
1892	011452	004737	013560		JSR	#7,GETOCT		; ALL THAT FOR BIT 15
1893	011456	020027	123456		CMP	#0,#123456		; PICK UP AN OCTAL NUMBER
1894	011462	001726			BEQ	350\$; CHECK FOR BAD DATA
1895	011464	000241			CLC			; BR, IF DATA WAS BAD
1896		000014			.REPT	12.		; CLEAR CARRY FOR ROTATE
1897					ROL	#0		
1898					.ENDR			; ROTATE R0
	011466	006100			ROL	#0		; ROTATE R0
	011470	006100			ROL	#0		; ROTATE R0
	011472	006100			ROL	#0		; ROTATE R0
	011474	006100			ROL	#0		; ROTATE R0
	011476	006100			ROL	#0		; ROTATE R0
	011500	006100			ROL	#0		; ROTATE R0
	011502	006100			ROL	#0		; ROTATE R0
	011504	006100			ROL	#0		; ROTATE R0
	011506	006100			ROL	#0		; ROTATE R0
	011510	006100			ROL	#0		; ROTATE R0
	011512	006100			ROL	#0		; ROTATE R0
	011514	006100			ROL	#0		; ROTATE R0
1899	011516	060037	001124		ADD	#0,WORK		; ROTATE R0
1900	011522	004737	013560		JSR	#7,GETOCT		; "OR" THE BITS IN
1901	011526	020027	123456		CMP	#0,#123456		; GO GET AN OCTAL NUMBER
1902	011532	001702			BEQ	350\$; CHECK FOR NON-OCTAL #
1903	011534	000241			CLC			; BR, IN NO GOOD #
1904		000011			.REPT	9.		; CLEAR CARRY FOR ROTATE
1905					ROL	#0		
1906					.ENDR			; ROTATE R0
	011536	006100			ROL	#0		; ROTATE R0
	011540	006100			ROL	#0		; ROTATE R0
	011542	006100			ROL	#0		; ROTATE R0
	011544	006100			ROL	#0		; ROTATE R0
	011546	006100			ROL	#0		; ROTATE R0
	011550	006100			ROL	#0		; ROTATE R0
	011552	006100			ROL	#0		; ROTATE R0
	011554	006100			ROL	#0		; ROTATE R0
	011556	006100			ROL	#0		; ROTATE R0
1907	011560	060037	001124		ADD	#0,WORK		; ROTATE R0
1908	011564	004737	013560		JSR	#7,GETOCT		; "OR" THE BITS IN
1909	011570	020027	123456		CMP	#0,#123456		; GO GET AN OCTAL NUMBER
1910	011574	001661			BEQ	350\$; CHECK FOR NON-OCTAL #
1911	011576	000241			CLC			; BR, IN NO GOOD #
1912		000006			.REPT	6		; CLEAR CARRY FOR ROTATE
1913					ROL	#0		
1914					.ENDR			; ROTATE R0
	011600	006100			ROL	#0		; ROTATE R0
	011602	006100			ROL	#0		; ROTATE R0
	011604	006100			ROL	#0		; ROTATE R0
	011606	006100			ROL	#0		; ROTATE R0
	011610	006100			ROL	#0		; ROTATE R0
	011612	006100			ROL	#0		; ROTATE R0
1915	011614	060037	001124		ADD	#0,WORK		; ROTATE R0
1916	011620	004737	013560		JSR	#7,GETOCT		; "OR" THE BITS IN
								; GO GET AN OCTAL NUMBER


```

2085 012616 023727 001124 000000 527$: CMP      WORK,#0
2086 012624 001616          BEQ      498$      ;CHECK FOR NOT SET
2087 012626 004737 013642          JSR      #7,GETCR  ;BR, IF NUMBERS NOT VALID
2088 012632 020027 123456          CMP      #0,#123456 ;GO GET LF OR CR
2089 012636 001611          BEQ      498$      ;CHECK FOR BAD INPUT
2090 012640          530$:          ;TRY AGAIN BAD INPUT (NOT CR)
2091 012640 013737 001124 001102 540$: MOV      WORK,BRATE ;STORE BAUD RATE FOR LATER
2092 012646 053737 001124 001070 BIS      WORK,DZLPR  ;PUT BAUD RATE IN PLACE
2093          ;
2094          ;
2095          ;
2096 012654          600$:          THIS CODE SETS UP THE NUMBER OF STOP BITS
2097 012654 004737 014066          JSR      #7,CLRTTY
2098 012660 104000          EMT      +0          ;PICK UP PENDING CHARACTERS
2099 012662 017153          STOPM          ;CALL PRINT ROUTINE
2100 012664 105777 166132 604$: TSTB     @TKS          ;"TYPE NUMBER OF STOP BITS ETC."
2101 012670 100375          BPL      604$      ;CK TTY IN STATUS
2102 012672 117700 166120          MOVB     @TKB,#0    ;WAIT FOR DONE
2103 012676 105777 166116 607$: TSTB     @TPS          ;PICK UP CHARACTER
2104 012702 100375          BPL      607$      ;CHECK FOR BUSY
2105 012704 110077 166104          MOVB     #0,@TPB    ;LOOP IF TTY IS BUSY (BR)
2106 012710 042700 177700          BIC      #177700,#0 ;ECHO CHARACTER
2107 012714 122700 000012          CMPB     #12,#0     ;ONLY 6 BIT PASS
2108 012720 001413          BEQ      610$      ;CHK FOR <LF> DEFAULT
2109 012722 122700 000015          CMPB     #15,#0     ;BR, IF DEFAULT (ONE STOP BIT)
2110 012726 001410          BEQ      610$      ;CHK FOR <CR> DEFAULT
2111 012730 122700 000061          CMPB     #61,#0    ;BR, IF DEFAULT (ONE STOP BIT)
2112 012734 001011          BNE      620$      ;WAS AN ASCII 1 TYPED
2113 012736 004737 013642          JSR      #7,GETCR  ;BR, IF IT WASN'T A ONE
2114 012742 020027 123456          CMP      #0,#123456 ;GO GET LF OR CR
2115 012746 001742          BEQ      600$      ;CHECK FOR BAD INPUT
2116 012750 042737 000040 001070 610$: BIC      #40,DZLPR  ;TRY AGAIN BAD INPUT (NOT CR)
2117 012756 000413          BR       630$      ;1 STOP BIT = 0
2118 012760 122700 000062 620$: CMPB     #62,#0     ;SKIP OVER
2119 012764 001333          BNE      600$      ;CHECK FOR A TWO
2120 012766 004737 013642          JSR      #7,GETCR  ;BR, IF NOT A TWO (ERROR)
2121 012772 020027 123456          CMP      #0,#123456 ;GO GET LF OR CR
2122 012776 001726          BEQ      600$      ;CHECK FOR BAD INPUT
2123 013000 052737 000040 001070 BIS      #40,DZLPR  ;TRY AGAIN BAD INPUT (NOT CR)
2124 013006          630$:          ;2 STOP BIT = 1
2125 013006 004737 014066          JSR      #7,CLRTTY
2126          ;
2127          ;
2128          ;
2129 013012          700$:          THIS CODE SETS UP THE NUMBER OF DATA BITS
2130 013012 004737 014066          JSR      #7,CLRTTY
2131 013016 104000          EMT      +0          ;PICK UP PENDING CHARACTERS
2132 013020 017224          DATAM          ;CALL PRINT ROUTINE
2133 013022 105777 165774 704$: TSTB     @TKS          ;"TYPE NUMBER OF DATA BITS ETC."
2134 013026 100375          BPL      704$      ;CK TTY IN STATUS
2135 013030 117700 165762          MOVB     @TKB,#0    ;WAIT FOR DONE
2136 013034 105777 165760 707$: TSTB     @TPS          ;PICK UP CHARACTER
2137 013040 100375          BPL      707$      ;CHECK FOR BUSY
2138 013042 110077 165746          MOVB     #0,@TPB    ;LOOP IF TTY IS BUSY (BR)
2139 013046 042700 177700          BIC      #177700,#0 ;ECHO CHARACTER
2140 013052 122700 000012          CMPB     #12,#0     ;ONLY 6 BIT PASS
2141 013056 001407          BEQ      710$      ;CHK FOR <LF> DEFAULT
                ;BR, IF DEFAULT (7 DATA BITS)
    
```



```

2250
2251
2252
2253
2254
2255
2256 013560
2257 013560 105777 165236
2258 013564 100375
2259 013566 117700 165224
2260 013572 105777 165222
2261 013576 100375
2262 013600 110077 165210
2263 013604 042700 177700
2264 013610 120027 000070
2265 013614 002007
2266 013616 120027 000060
2267 013622 002404
2268 013624 042700 177770
2269 013630 000137 013640
2270 013634 012700 123456
2271 013640 000207
2272
2273
2274
2275
2276 013642
2277 013642 105777 165154
2278 013646 100375
2279 013650 117700 165142
2280 013654 105777 165140
2281 013660 100375
2282 013662 110077 165126
2283 013666 042700 177700
2284 013672 120027 000015
2285 013676 001405
2286 013700 120027 000012
2287 013704 001402
2288 013706 012700 123456
2289 013712 004737 014066
2290 013716 000207
2291 013720 004737 014066
2292 013724 104000
2293 013726 015613
2294 013730 104000
2295 013732 015643
2296 013734 104000
2297 013736 015654
2298 013740 104000
2299 013742 015667
2300 013744 105777 165052
2301 013750 100375
2302 013752 117700 165040
2303 013756 105777 165036
2304 013762 100375
2305 013764 110077 165024
2306 013770 042700 177700

;*****
;
; OTHER SUBROUTINES
;
GETOCT:
1$: TSTB @TKS ;CK TTY IN STATUS
    BPL 1$ ;WAIT FOR DONE
    MOVB @TKB,%0 ;PICK UP CHARACTER
5$: TSTB @TPS ;CHECK FOR BUSY
    BPL 5$ ;LOOP IF TTY IS BUSY (BR)
    MOVB %0,@TPB ;ECHO CHARACTER
    BIC #177700,%0 ;ONLY 6 BIT PASS
    CMPB %0,#70 ;ERROR IF 8 OR MORE
    BGE 10$
    CMPB %0,#60 ;ERROR IF LESS THAN 0
    BLT 10$
    BIC #177770,%0 ;ONLY THREE BITS PASS
    JMP 20$ ;OCTAL # OK
10$: MOV #123456,%0 ;WAS NOT OCTAL #
20$: RTS #7
;*****
;
; THIS ROUTINE WAITS FOR A CR OR LF
;
GETCR:
1$: TSTB @TKS ;CK TTY IN STATUS
    BPL 1$ ;WAIT FOR DONE
    MOVB @TKB,%0 ;PICK UP CHARACTER
5$: TSTB @TPS ;CHECK FOR BUSY
    BPL 5$ ;LOOP IF TTY IS BUSY (BR)
    MOVB %0,@TPB ;ECHO CHARACTER
    BIC #177700,%0 ;ONLY 6 BIT PASS
    CMPB %0,#15 ;WAS IT A CR
    BEQ 20$ ;BR, IF IT WAS
    CMPB %0,#12 ;WAS IT A LF
    BEQ 20$ ;BR, IF IT WAS
10$: MOV #123456,%0 ;WAS NOT OCTAL #
20$: JSR #7,CLRTTY ;CLEAR OUT ANY WAITING CHARA
    RTS #7
EIACHK: JSR #7,CLRTTY ;PICK UP PENDING CHARACTERS
    EMT +0 ;CALL TO THE TTY PRINTER
    MENU10 ;"LINE TYPE"
    EMT +0 ;CALL TO THE TTY PRINTER
    MENU20 ;" 1 EIA"
    EMT +0 ;CALL TO THE TTY PRINTER
    MENU30 ;" 20MA"
    EMT +0 ;CALL TO THE TTY PRINTER
    MENU40 ;"SERIAL LINE TYPE <1>?"
1$: TSTB @TKS ;CHK TTY IN STATUS
    BPL 1$ ;WAIT FOR DONE
    MOVB @TKB,%0 ;PICK UP CHARACTER TYPED IN
5$: TSTB @TPS ;CHECK FOR BUSY
    BPL 5$ ;LOOP UNTIL NOT BUSY
    MOVB %0,@TPB ;ECHO CHARACTER
    BIC #177700,%0 ;ONLY 6 BIT PASS
    
```

```

2307 013774 122700 000012          CMPB    #12,#0          ;WAS DEFAULT SEL. <LF>
2308 014000 001426                BEQ     20$             ;BR, IF DEFAULT SELECTED
2309 014002 122700 000015          CMPB    #15,#0          ;WAS DEFAULT SEL. <CR>
2310 014006 001423                BEQ     20$             ;BR, IF DEFAULT SELECTED
2311 014010 042700 177770          BIC     #177770,#0      ;ONLY THREE BITS PASS
2312 014014 122700 000001          CMPB    #1,#0          ;CHECK FOR A "1" TYPED
2313 014020 001006                BNE     10$            ;BR, IF IT WASN'T A ONE
2314 014022 012737 000001 001112  MOV     #1,EIA          ;SET FOR EIA
2315 014030 004737 013642          JSR     #7,GETCR        ;WAIT FOR CR
2316 014034 000413                BR      100$           ;EXIT
2317 014036 122700 000002          CMPB    #2,#0          ;CHECK FOR 20MA
2318 014042 001340                BNE     1$             ;BR, NEITHER
2319 014044 005037 001112          CLR     EIA            ;SET TO 20 MA
2320 014050 004737 013642          JSR     #7,GETCR        ;WAIT FOR CR
2321 014054 000403                BR      100$           ;EXIT
2322 014056 012737 000001 001112  MOV     #1,EIA          ;CR = DEFAULT = EIA
2323 014064 000207                BR      100$           ;EXIT SUBROUTINE
2324
2325
2326
2327
2328
2329
2330
2331 014066
2332 014066 012703 040000          CLRTTY:
2333 014072 105777 164724          1$:    MOV     #40000,#3  ;BIG COUNTER
2334 014076 100007                TSTB   @TKS            ;CHECK FOR RECEIVER DONE SET
2335 014100 117700 164712          BPL     5$             ;BR, IF DONE IS NOT SET
2336 014104 105777 164710          MOVB   @TKB,#0         ;PICK UP THE CHARACTER
2337 014110 100375                TSTB   @TPS            ;CHECK FOR TRANSMITTER READY
2338 014112 110077 164676          BPL     3$             ;WAIT UNTIL READY
2339 014116 077313                MOVB   #0,@TPB         ;ECHO THE "EXTRA" CHARACTER
2340 014120 000207                SOB    #3,1$           ;LOOP COUNTER
                                RTS     #7                     ;RETURN TO CALLEE

```

THIS SUBROUTINE PICKS CHARACTERS OUT OF THE TTY RECEIVE BUFFER
 I.E. GETS RID OF THE "EXTRA" ONES

CLRTTY:

```

2342
2343
2344
2345
2346
2347
2348
2349
2350
2351 014122
2352 014122 005737 001102
2353 014126 001403
2354 014130 005737 014144
2355 014134 000207
2356 014136 005777 164636
2357 014142 000774
2358 014144 000200
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369 014146
2370 014146 005737 001106
2371 014152 001047
2372 014154 005737 001102
2373 014160 001416
2374 014162 005737 001112
2375 014166 001404
2376 014170 137737 164714 001066
2377 014176 001465
2378 014200 037727 164574 000200 5$:
2379 014206 001006 10$:
2380 014210 005777 164564 20$:
2381 014214 000207 50$:
2382 014216 105777 164556
2383 014222 000774
2384 014224 017705 164666 60$:
2385 014230 042705 177700
2386 014234 120527 000023
2387 014240 001363
2388 014242 037727 164532 000200 70$:
2389 014250 001774
2390 014252 017705 164640
2391 014256 042705 177700
2392 014262 120527 000021
2393 014266 001365
2394 014270 000747
2395 014272 032777 000200 164576 80$:
2396 014300 001746
2397 014302 117705 164572
2398 014306 042705 177700

```

```

;*****
;
; TEST FOR ERROR,
; THIS ROUTINE TAKES THE PLACE OF ALL THE "TST @LPS" CODES.
; THIS WAS REQUIRED FOR THE DZ11. IF THE DZ IS NOT SELECTED
; THE ROUTINE SIMPLY DOES THE "TST @LPS" AND RETURNS. IF THERE
; IS A DZ11 IT ONLY CHECKS A DUMMY WORD TO SHOW NO ERROR.
; THERE ARE NO TRANSMIT ERRORS ON THE DZ11.
ERCHK:
    TST     BRATE
    BEQ     50$ ;CHECK FIRST FOR DZ
    TST     TSTWRD ;BR, IF NO DZ SELECTED
    RTS     *7 ;DUMMY CONDITION CODE SETUP
    TST     @LPS ;RETURN WITH NO MINUS SET
    BR      20$ ;NOT DZ SO DO REGULAR CHECK
TSTWRD: .WORD 000200 ;RETURN WITH CODES ETC
;NO MINUS BIT SET
;*****
;
; TEST FOR READY,
; THIS ROUTINE TAKES THE PLACE OF ALL THE "TSTB @LPS" CODES.
; THIS WAS REQUIRED FOR THE DZ11. IF THE DZ IS NOT SELECTED
; THE ROUTINE SIMPLY DOES THE "TSTB @LPS" AND RETURNS. IF THERE
; IS A DZ11 IT CHECKS THE REAL READY BIT (BIT15).
; THE ROUTINE DOES A "TST @LPS" WHICH SETS THE CORRECT CONDITION
; CODES
ERCHKB:
    TST     DLHERE
    BNE     80$ ;ARE WE TALKING TO A DL
    TST     BRATE ;BR, IF DL IS BEING USED
    BEQ     50$ ;CHECK FIRST FOR DZ
    TST     EIA ;BR, IF NO DZ SELECTED
    BEQ     5$ ;WHAT MODE ARE WE IN
    BITB    @DZTCRA,DZTCR ;BR, IF IN 20 MA MODE
    BEQ     100$ ;IS DATA TERMINAL READY SET
    BIT     @LPS,#200 ;BR IF IT IS NOT SET (ERROR)
    BNE     60$ ;RECEIVER DONE SET
    TST     @LPS ;BR, IF RECVR HAS CHARACTER
    RTS     *7 ;LOOKS AT REAL READY BIT
    TSTB    @LPS ;RETURN WITH NO MINUS SET
    BR      20$ ;NOT DZ SO DO REGULAR CHECK
    MOV     @DZRBUF,#5 ;RETURN WITH CODES ETC
    BIC     #177700,#5 ;PICK UP CHARACTER
    CMPB    #5,#23 ;ONLY 6 BITS PASS
    BNE     10$ ;CHK FOR XOFF (CNTL S)
    BIT     @LPS,#200 ;BR, IF NOT XOFF (RETURN)
    BEQ     70$ ;LOOK FOR ANOTHER CHARACTER
    MOV     @DZRBUF,#5 ;WAIT HERE FOR NEXT CHARACTER
    BIC     #177700,#5 ;IT ARRIVED
    CMPB    #5,#21 ;ONLY SIX BITS PASS
    BNE     70$ ;WAS IT XON (CNTL Q)
    BR      10$ ;BR, IF IT WASN'T XON (LOOP)
    BIT     #200,@DLLPS ;IT WAS XON RETURN
    BEQ     50$ ;HAVE WE RECEIVED A RECV CHARA
    MOVB    @DLRBUF,#5 ;BR, IF WE HAVE NOT
    BIC     #177700,#5 ;PICK UP THE CHARACTER
;ONLY SIX BITS

```



```

2413
2414
2415
2416
2417
2418
2419
2420 014364
2421 014364 104000
2422 014366 015302
2423 014370 104000
2424 014372 015335
2425 014374 104000
2426 014376 015365
2427 014400 104000
2428 014402 015421
2429 014404 105777 164412
2430 014410 100375
2431 014412 117700 164400
2432 014416 105777 164376
2433 014422 100375
2434 014424 110077 164364
2435 014430 042700 177700
2436 014434 122700 000012
2437 014440 001416
2438 014442 122700 000015
2439 014446 001413
2440 014450 120027 000061
2441 014454 001421
2442 014456 120027 000062
2443 014462 001350
2444 014464 004737 013642
2445 014470 020027 123456
2446 014474 001743
2447 014476 004737 014066
2448 014502 000207
2449 014504 004737 013642
2450 014510 020027 123456
2451 014514 001403
2452 014516 000767
2453 014520
2454 014520 004737 013642
2455 014524 104000
2456 014526 015455
2457 014530 012737 000001 001102
2458 014536 105777 164260
2459 014542 100375
2460 014544 117700 164246
2461 014550 105777 164244
2462 014554 100375
2463 014556 110077 164232
2464 014562 042700 177600
2465 014566 122700 000012
2466 014572 001741
2467 014574 122700 000015
2468 014600 001736
2469 014602 122700 000131

;*****
;
; THIS ROUTINE CHECKS FOR DL11S THAT NEED ADDITIONAL PROGRAMMING
; FEATURES I.E. PROGRAMMABLE BAUD RATES ON THE DLV11-E/F MODULES.
;
DLSET:
EMT +0 ;PRINT CALL
DLASK1 ;"DL11 TYPE MENU"
EMT +0 ;PRINT MESSAGE CALL
DLASK2 ;"1 DLV11-F OR DLV11-E"
EMT +0 ;PRINT MESSAGE CALL
DLASK3 ;"2 DLV11, DL11 OR DLV11-J"
EMT +0 ;PRINT MESSAGE CALL
DLASK4 ;"ENTER MENU SELECTION"
40$: TSTB @TKS ;READ TTY'S STATUS
BPL 40$ ;BR, IF NOT DONE
MOVB @TKB,%0 ;PICK UP CHAR. TYPED
45$: TSTB @TPS ;CHECK FOR BUSY
BPL 45$ ;LOOP IF TTY IS BUSY (BR)
MOVB %0,@TPB ;ECHO CHARACTER TO TTY
BIC @177700,%0 ;ONLY 6 BITS ALLOWED
CMPB @12,%0 ;CHK FOR <LF> DEFAULT
BEQ 75$ ;BR, IF DEFAULT (NO)
CMPB @15,%0 ;CHK FOR <CR> DEFAULT
BEQ 75$ ;BR, IF DEFAULT (NO)
CMPB %0,%61 ;WAS IT AN ASCII 1
BEQ 100$ ;BR, IF IT WAS (DL11 SEL.)
CMPB %0,%62 ;WAS IT AN ASCII 2
BNE 40$
JSR %7,GETCR ;GO GET LF OR CR
CMP %0,@123456 ;CHECK FOR BAD INPUT
BEQ 40$ ;TRY AGAIN BAD INPUT (NOT CR)
75$: JSR %7,CLRTTY ;CLEAR OUT THE TTY
RTS %7 ;RETURN TO CALLEE
76$: JSR %7,GETCR ;LOOK FOR CR
CMP %0,@123456 ;WAS THERE AN ERROR
BEQ 105$ ;BR, IF AN ERROR RESTART
BR 75$ ;EXIT NO ERROR
100$: JSR %7,GETCR
105$: EMT +0 ;GET CR
DLASK5 ;CALL PRINT ROUTINE
MOV @1,BRATE ;"DOES DL HAVE PROG BAUD RATE"
110$: TSTB @TKS ;SWITCH FOR BAUD RATE NEEDED
BPL 110$ ;CK TTY IN STATUS
MOVB @TKB,%0 ;WAIT FOR DONE
115$: TSTB @TPS ;PICK UP CHARACTER
BPL 115$ ;CHECK FOR BUSY
MOVB %0,@TPB ;LOOP IF TTY IS BUSY (BR)
BIC @177600,%0 ;ECHO CHARACTER
CMPB @12,%0 ;ONLY 7 BIT PASS
BEQ 75$ ;CHK FOR <LF> DEFAULT
CMPB @15,%0 ;BR, IF DEFAULT (NO)
BEQ 75$ ;CHK FOR <CR> DEFAULT
CMPB @131,%0 ;BR, IF DEFAULT (NO)
;WAS AN ASCII Y TYPED

```

270	014606	001412			BEQ	497\$		
2471	014610	122700	000116	120\$:	CMPB	#116,#0		;BR, IF IT WASN'T A Y
2472	014614	001343			BNE	105\$;CHECK FOR A N
2473	014616	004737	013642		JSR	#7,GETCR		;BR, IF NOT A N (ERROR)
2474	014622	020027	123456		CMP	#0,#123456		;GO GET LF OR CR
2475	014626	001736			BEQ	105\$;CHECK FOR BAD INPUT
2476	014630	000137	014476		JMP	75\$;TRY AGAIN BAD INPUT (NOT CR)
2477	014634	004737	013642	497\$:	JSR	#7,GETCR		;EXIT ROUTINE
2478	014640	020027	123456		CMP	#0,#123456		;GO GET LF OR CR
2479	014644	001727			BEQ	105\$;CHECK FOR BAD INPUT
2480	014646	004737	014066	498\$:	JSR	#7,CLRTTY		;TRY AGAIN BAD INPUT (NOT CR)
2481	014652	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2482	014654	016647			LN1A			;CALL TO THE TTY PRINTER
2483	014656	004737	014066		JSR	#7,CLRTTY		;BAUD RATE MENU PRINTOUTS
2484	014662	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2485	014664	016701			LN2			;CALL TO THE TTY PRINTER
2486	014666	004737	014066		JSR	#7,CLRTTY		;BAUD RATE MENU PRINTOUTS
2487	014672	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2488	014674	016722			LN3			;CALL TO THE TTY PRINTER
2489	014676	004737	014066		JSR	#7,CLRTTY		;BAUD RATE MENU PRINTOUTS
2490	014702	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2491	014704	016743			LN4			;CALL TO THE TTY PRINTER
2492	014706	004737	014066		JSR	#7,CLRTTY		;BAUD RATE MENU PRINTOUTS
2493	014712	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2494	014714	016764			LN5			;CALL TO THE TTY PRINTER
2495	014716	004737	014066		JSR	#7,CLRTTY		;BAUD RATE MENU PRINTOUTS
2496	014722	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2497	014724	017005			LN6			;CALL TO THE TTY PRINTER
2498	014726	104000			EMT	*0		;BAUD RATE MENU PRINTOUTS
2499	014730	017026			LN7			;CALL TO THE TTY PRINTER
2500	014732	004737	014066		JSR	#7,CLRTTY		;BAUD RATE MENU PRINTOUTS
2501	014736	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2502	014740	017047			LN8			;CALL TO THE TTY PRINTER
2503	014742	104000			EMT	*0		;BAUD RATE MENU PRINTOUTS
2504	014744	017070			LN8A			;CALL TO SUBROUTINE
2505	014746	004737	014066		JSR	#7,CLRTTY		; "8" =19200 BAUD"
2506	014752	104000			EMT	*0		;PICK UP PENDING CHARACTERS
2507	014754	017111			LN10			;CALL TO THE TTY PRINTER
2508	014756	105777	164040	500\$:	TSTB	@TKS		;BAUD RATE MENU PRINTOUTS
2509	014762	100375			BPL	500\$;CHK TTY IN STATUS
2510	014764	117700	164026		MOVB	@TKB,#0		;WAIT FOR DONE
2511	014770	105777	164024	505\$:	TSTB	@TPS		;PICK UP CHARACTER TYPED IN
2512	014774	100375			BPL	505\$;CHECK FOR BUSY
2513	014776	110077	164012		MOVB	#0,@TPB		;LOOP UNTIL NOT BUSY
2514	015002	042700	177700		BIC	#177700,#0		;ECHO CHARACTER
2515	015006	122700	000012		CMPB	#12,#0		;ONLY 6 BIT PASS
2516	015012	001475			BEQ	544\$;WAS DEFAULT SEL. <LF>
2517	015014	122700	000015		CMPB	#15,#0		;BR, IF DEFAULT SELECTED
2518	015020	001472			BEQ	544\$;WAS DEFAULT SEL. <CR>
2519	015022	005037	001124		CLR	WORK		;BR, IF DEFAULT SELECTED
2520	015026	042700	177760		BIC	#177760,#0		;CLEAR WORK AREA
2521	015032	122700	000001		CMPB	#1,#0		;ONLY FOUR BITS PASS
2522	015036	001003			BNE	510\$;CHECK FOR A "1" TYPED
2523	015040	012737	040000	001124	MOV	#40000,WORK		;BR, IF IT WASN'T A ONE
2524	015046							;SET FOR 150 BAUD
2525	015046	122700	000002	510\$:	CMPB	#2,#0		;CHECK FOR A TWO TYPED
2526	015052	001003			BNE	520\$;BR, IF IT WASN'T A '2


```

2527 015054 012737 050000 001124      MOV      #50000,WORK      ;SET FOR 300 BAUD
2528 015062      520$:      ;
2529 015062 122700 000003      CMPB     #3,#0           ;CHECK FOR A THREE TYPED
2530 015066 001003      BNE      521$           ;BR, IF IT WASN'T A "3"
2531 015070 012737 060000 001124      MOV      #60000,WORK      ;SET FOR 600 BAUD
2532 015076      521$:      ;
2533 015076 122700 000004      CMPB     #4,#0           ;CHECK FOR A FOUR TYPED
2534 015102 001003      BNE      522$           ;JR, IF IT WASN'T A "4"
2535 015104 012737 070000 001124      MOV      #70000,WORK      ;SET FOR 1200 BAUD
2536 015112      522$:      ;
2537 015112 122700 000005      CMPB     #5,#0           ;CHECK FOR A FIVE TYPED
2538 015116 001003      BNE      523$           ;BR, IF IT WASN'T A "5"
2539 015120 012737 120000 001124      MOV      #120000,WORK     ;SET FOR 2400 BAUD
2540 015126      523$:      ;
2541 015126 122700 000006      CMPB     #6,#0           ;CHECK FOR A SIX TYPED
2542 015132 001003      BNE      524$           ;BR, IF IT WASN'T A "6"
2543 015134 012737 140000 001124      MOV      #140000,WORK     ;SET FOR 4800 BAUD
2544 015142      524$:      ;
2545 015142 122700 000007      CMPB     #7,#0           ;CHECK FOR A SEVEN TYPED
2546 015146 001003      BNE      525$           ;BR, IF IT WASN'T A "7"
2547 015150 012737 160000 001124 543$:      MOV      #160000,WORK     ;SET FOR 9600 BAUD
2548 015156      525$:      ;
2549 015156 122700 000010      CMPB     #8,#0           ;WAS AN ASCII EIGHT TYPED
2550 015162 001015      BNE      526$           ;IT WASN'T AN EIGHT
2551 015164 012737 170000 001124      MOV      #170000,WORK     ;SET FOR 19200KC
2552 015172 004737 013642      JSR      #7,GETCR        ;GO GET LF OR CR
2553 015176 020027 123456      CMP      #0,#123456     ;CHECK FOR BAD INPUT
2554 015202 001621      BEQ      498$           ;BR, IF CR RECEIVED (GOOD)
2555 015204 000415      BR       530$           ;LEAVE
2556 015206 012737 160000 001124 544$:      MOV      #160000,WORK     ;DEFAULT IS 9600
2557 015214 000411      BR       530$           ;LEAVE
2558 015216      526$:      ;
2559 015216 023727 001124 000000 527$:      CMP      WORK,#0        ;CHECK FOR NOT SET
2560 015224 001610      BEQ      498$           ;BR, IF NUMBERS NOT VALID
2561 015226 004737 013642      JSR      #7,GETCR        ;GO GET LF OR CR
2562 015232 020027 123456      CMP      #0,#123456     ;CHECK FOR BAD INPUT
2563 015236 001603      BEQ      498$           ;TRY AGAIN BAD INPUT (NOT CR)
2564 015240      530$:      ;
2565 015240 013737 001124 001102 540$:      MOV      WORK,BRATE     ;STORE BAUD RATE FOR LATER
2566 015246 053737 001124 001074      BIS      WORK,DLLPR     ;PUT BAUD RATE IN PLACE
2567      ;
2568      ;
2569 015254 013703 001000      MOV      LPS,#3         ;GET STATUS REG ADDRESS
2570 015260 062703 000002      ADD      #2,#3         ;POINT TO LPR REGISTER
2571 015264 052737 004000 001074      BIS      #4000,DLLPR    ;SET PBR ENB BIT
2572 015272 013713 001074      MOV      DLLPR,(#3)     ;SET SPEED, LINE, PARITY ETC
2573      ;
2574      ;
2575 015276 000137 014476      JMP      75$           ;NO MORE DL QUESTIONS
  
```


ACNVX	004576	DLCSRC	001032	HLP4A	007672	MES6	006347	TK9	005424
ACVN	004514	DLCSRM	016317	HLP5	007752	MES7	006416	TPB	001014
BEGIN	000000	DLHERE	001106	HLP6	010040	MES8	006457	TPS	001020
BIT0	= 000001	DLLPR	001074	HLP7	010100	MES9	006475	TSEAB	004176
BIT1	= 000002	DLLPS	001076	HLP8	010163	N	= 000014	TSEDA	004202
BIT10	= 0020C0	DLRATE	001104	HWSR	001012	NOP	= 000240	TSEND	004070
BIT11	= 004000	DLRBUF	001100	LCP3	005523	NUMCHR	001146	TSRSS	004232
BIT12	= 000000	DLSET	014364	LCP4	005533	OCT	001162	TSRST	004236
BIT13	= 020000	DLTYPE	001114	LCP5	005542	OFFSET	001150	TSTWRD	014144
BIT14	= 040000	DLVEC	001042	LCP7	005556	PARITY	017275	TY01	005571
BIT15	= 100000	DLVECM	016364	LEGCHR	001144	PARITZ	017340	TY02	005630
BIT2	= 000004	DZCSR	001036	LINCNT	001062	PASSA	001164	TY03	005673
BIT3	= 000010	DZCSR	001040	LKS	001030	PLKS	001026	TYP	004332
BIT4	= 000020	DZCSRH	001120	LN1	016615	PRINE	004250	TYPA	004342
BIT5	= 000040	DZCSR	016431	LN1A	016647	PRNNT	004242	TYPC	004352
BIT6	= 000100	DZLINE	016547	LN10	017111	PRTMSG	004312	TYPD	004400
BIT7	= 000200	DZLNE	001072	LN2	016701	PSW	001010	TYPDAT	004464
BIT8	= 000400	DZLPR	001070	LN3	016722	PTRC	001046	TYPDO	004426
BIT9	= 001000	DZRBUF	001116	LN4	016743	PTRDTR	015536	TYPD01	004434
BRATE	001102	DZTCR	001066	LN5	016764	PTRPSW	001052	TYPF	004436
BUFF	001166	DZTCRA	001110	LN6	017005	PTRVEC	001050	TYPG	004450
CHAR	001160	DZVEC	001044	LN7	017026	RINT	004274	TYPINT	004314
CHRCNT	001056	DZVECM	016476	LN8	017047	R6	= 000006	TYPSWR	005272
CHRGEN	001060	EIA	001112	LN8A	017070	R7	= 000007	TYPSWX	005336
CLRTTY	014066	EIACHK	013720	LPB	001002	SAVE	001134	W	= 000007
CNTRLG	005216	EL	= 000037	LPS	001000	SEGCNT	001054	WAIT1	001730
CN1	001666	ERCHK	014122	LSTADR	017410	SET	001156	WAIT2	002100
CN10	003322	ERCHKB	014146	M	= 000002	SETSER	010234	WAIT3	002366
CN12	003550	ERCOUN	001136	MAINTB	001122	SETUP	001306	WAIT4	002536
CN2	002072	ERMS1	006704	MENUD1	016070	SIGNAL	001154	WAIT5	003142
CN3	002242	ERMS2	006756	MENUD2	016156	SL	= 000036	WAIT6	003406
CN4	002326	ERMS3	007026	MENUD3	016221	STAER	004606	WDD1	001736
CN5	002530	ERMS4	007120	MENUD4	016263	STARN	004752	WDD2	002106
CN6	002700	ERMS5	007211	MENU1	015723	START	000200 G	WDD3	002374
CN7	002764	ERMS6	007305	MENU10	015613	STEDA	004714	WDD4	002544
CONR1	004600	ERR1	001620	MENU2	015760	STEXT	004744	WDD5	003166
CONR2	004602	ERR10	003254	MENU20	015643	STOPM	017153	WDD6	003414
CONR3	004604	ERR11	003362	MENU3	016006	STRCHR	001140	WDE1	001752
CONV	004466	ERR12	003502	MENU30	015654	STRCNT	001142	WDE2	002122
CSBR	001024	ERR13	004262	MENU4	016034	SWR	001004	WDE3	002410
CYCCNT	001064	ERR2	002024	MENU40	015667	SWREG	000176	WDE4	002560
DAR1	005512	ERR3	002174	MESDD	006114	TEST1	001464	WDE5	003202
DAR2	005516	ERR4	002260	MES1	006153	TEST2	003052	WDE6	003430
DAR6	005551	ERR5	002462	MES10	006504	TEST3	003640	WER1	001770
DAR9	005567	ERR6	002632	MES14	006570	TIME	001130	WER2	002140
DATAM	017224	ERR7	002716	MES2	006167	TIMER	001132	WER3	002426
DGTS	005200	ETIM	005731	MES20A	006607	TKB	001016	WER4	002576
DIGITS	001152	ETIMO	006007	MES21	006612	TKHLP	005440	WER5	003220
DISPLA	001006	GETCR	013642	MES22	006614	TKINT	004754	WER6	003446
DISPRE	000174	GETOCT	013560	MES23	006620	TKS	001022	WEX1	002074
DLASK1	015302	HED1	006063	MES24	006630	TK1	005066	WEX2	002244
DLASK2	015335	HED2	006104	MES25	006647	TK2	005100	WEX3	002532
DLASK3	015365	HLP0	007350	MES26	006656	TK3	005366	WEX4	002702
DLASK4	015421	HLP1	007420	MES27	006672	TK4	005372	WEX5	003324
DLASK5	015455	HLP2	007514	MES3	006230	TK5	005170	WEX6	003554
DL CRLF	016543	HLP3	007541	MES4	006246	TK6	005406	WORK	001124
DLCSR	001034	HLP4	007611	MES5	006274	TK7	005376	WORKA	001126

WT2 005114 WT3 005242

. ABS. 017412 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 4362 WORDS (18 PAGES)

DYNAMIC MEMORY: 7630 WORDS (29 PAGES)

ELAPSED TIME: 00:01:37

CZLCPA,CZLCPA/-SP/CR=CZLCPA

SYMBOL CROSS REFERENCE

CREF V01

SYMBOL	VALUE	REFERENCES
ACNVX	004576	*25-1423
ACVN	004514	#25-1427 25-1427 *25-1432 *25-1434 *25-1436 #25-1445
BEGIN	000000	#25-1427 25-1438 #15-608 15-613 15-619 15-623 15-625 15-629 15-634 15-642 16-652 16-662 #14-599
BIT0	= 00C001	#14-598
BIT1	= 000002	#14-589
BIT10	= 002000	#14-588
BIT11	= 004000	#14-587
BIT12	= 010000	#14-586
BIT13	= 020000	#14-585 15-626
BIT14	= 040000	#14-584
BIT15	= 100000	#14-584
BIT2	= 000004	#14-597
BIT3	= 000010	#14-596
BIT4	= 000020	#14-595
BIT5	= 000040	#14-594
BIT6	= 000100	#14-593
BIT7	= 000200	#14-592
BIT8	= 000400	#14-591
BIT9	= 001000	#14-590
BRATE	001102	#18-975 *28-1701 *28-1743 *28-1780 *28-2091 30-2352 30-2372 *31-2457 *31-2565
BUFF	001166	#18-1004 22-1209 22-1225 22-1237
CHAR	001160	#18-1001 *26-1501 *26-1502 26-1503 26-1505 26-1507 26-1509 26-1512 26-1515 26-1518 26-1526 *26-1527 26-1536 *26-1542 *26-1543 26-1544 26-1548
CHRCNT	001056	#18-963
CHRGEN	001060	#18-964
CLRTTY	014066	24-1344 25-1478 28-1744 28-1781 28-1858 28-1861 28-1870 28-1962 28-1968 28-2011 28-2014 28-2017 28-2020 28-2023 28-2026 28-2029 28-2032 28-2035 28-2097 28-2125 28-2130 28-2161 28-2166 28-2195 28-2200 28-2228 29-2289 29-2291 #29-2331 31-2447 31-2480 31-2483 31-2486 31-2489 31-2492 31-2495 31-2500 31-2505
CNTRLG	005216	#26-1542
CN1	001666	20-1085 #20-1085
CN10	003322	22-1215 #22-1215
CN12	003550	22-1242 #22-1242
CN2	002072	21-1117 #21-1117
CN3	002242	21-1120 #21-1120
CN4	002326	21-1128 #21-1128
CN5	002530	21-1148 #21-1148
CN6	002700	21-1151 #21-1151
CN7	002764	21-1159 #21-1159
CONR1	004600	*25-1419 25-1439 #25-1446
CONR2	004602	*25-1420 25-1440 #25-1447
CONR3	004604	*25-1421 25-1441 #25-1448
CONV	004466	24-1326 #25-1419 25-1455 26-1561 26-1569
CSBR	001024	#16-686
CYCCNT	001064	#18-966 *19-1013 22-1215 23-1269 *24-1323 24-1327
DAR1	005512	#27-1629
DAR2	005516	#27-1630
DAR6	005551	#27-1634
DAR9	005567	#27-1636
DATAM	017224	28-2132 #32-2620

SYMBOL CROSS REFERENCE

CREF V01

SYMBOL	VALUE	REFERENCES
DGTS	005200	26-1506 #26-1538
DIGITS	001152	#18-998 *26-1521 26-1522 26-1528 26-1538 *26-1581
DISPLA	001006	#16-679 *19-1023
DISPRE	000174	#15-643 19-1023
DLASK1	015302	31-2422 #32-2584
DLASK2	015335	31-2424 #32-2585
DLASK3	015365	31-2426 #32-2586
DLASK4	015421	31-2428 #32-2587
DLASK5	015455	31-2456 #32-2588
DLCRLF	016543	28-1864 #32-2606
DLCSR	001034	#16-691
DLCSRC	001032	#16-690 28-1846 28-1848 28-1849
DLCSRM	016317	28-1784 #32-2602
DLHEPE	001106	#18-977 *28-1700 *28-1742 *28-1779 *28-1873 30-2370
DL.LPR	001074	#18-972 *31-2566 *31-2571 31-2572
DLLPS	001076	#18-973 21-1120 21-1151 22-1215 22-1242 *28-1838 *28-1848 30-2395 30-2401
DLRATE	001104	#18-976
DLRBUF	001100	#18-974 21-1122 21-1153 22-1216 22-1244 *28-1839 *28-1840 *28-1849 *28-1850
		30-2397 30-2403
DLSET	014364	28-1860 #31-2420
DLTYPE	001114	#18-982
DLVEC	001042	#16-694
DLVECM	016364	#32-2603
DZCSR	001036	#16-692
DZCSRC	001040	#16-693 28-1935
DZCSRH	001120	#18-984 25-1394 *28-1697 *28-1959
DZCSRM	016431	28-1872 #32-2604
DZLINE	016547	28-1970 #32-2607
DZLNE	001072	#18-970 *28-1990 *28-1992 28-1996 28-1997
DZLPR	001070	#18-969 *28-1699 *28-1996 *28-2092 *28-2116 *28-2123 *28-2147 *28-2149 *28-2159
		*28-2185 *28-2192 *28-2219 *28-2226 *28-2239 28-2240
DZRBUF	001116	#18-983 *28-2238 30-2384 30-2390
DZTCR	001066	#18-968 *28-1698 *28-1993 *28-2004 28-2243 30-2376
DZTCRA	001110	#18-979 *28-2244 *28-2245 30-2376
DZVEC	001044	#16-695
DZVECM	016476	#32-2605
EIA	001112	#18-980 *29-2314 *29-2319 *29-2322 30-2374
EIACHK	013720	#29-2291
EL	000037	#14-602
ERCHK	014122	25-1361 #30-2351
ERCHK8	014146	#30-2369 30-2411
ERCOUN	001136	#18-992 *20-1085 *21-1117 *21-1120 *21-1128 *21-1148 *21-1151 *21-1159 *22-1215
		*22-1232 *22-1242 *25-1365 25-1456
ERMS1	006704	20-1084 #27-1671 28-1950
ERMS2	006756	21-1127 21-1158 #27-1672
ERMS3	007026	#27-1673
ERMS4	007120	22-1231 #27-1674
ERMS5	007211	#27-1675
ERMS6	007305	25-1364 #27-1676
ERR1	001620	#20-1085
ERR10	003254	#22-1215
ERR11	003362	#22-1232

SYMBOL CROSS REFERENCE		CREF V01									
SYMBOL	VALUE	REFERENCES									
ERR12	003502	#22-1242									
ERR13	004262	#25-1365									
ERR2	002024	#21-1117									
ERR3	002174	#21-1120									
ERR4	002260	#21-1128									
ERR5	002462	#21-1148									
ERR6	002632	#21-1151									
ERR7	002716	#21-1159									
ETIM	005731	21-1120	21-1151	22-1215	22-1242	#27-1642					
ETIMO	006007	21-1117	21-1148	#27-1643							
GETCR	013642	28-1727	28-1739	28-1765	28-1776	28-1854	28-1931	28-2005	28-2078	28-2087	
		28-2113	28-2120	28-2150	28-2156	28-2182	28-2189	28-2216	28-2223	28-2276	
		29-2315	29-2320	31-2444	31-2449	31-2454	31-2473	31-2477	31-2552	31-2561	
GETOCT	013560	28-1802	28-1810	28-1818	28-1826	28-1834	28-1892	28-1900	28-1908	28-1916	
		28-1924	#29-2256								
HED1	006063	25-1463	#27-1644								
HED2	006104	25-1457	#27-1645								
HLP0	007350	26-1598	#27-1678								
HLP1	007420	26-1600	#27-1679								
HLP2	007514	26-1602	#27-1680								
HLP3	007541	26-1604	#27-1681								
HLP4	007611	26-1606	#27-1682								
HLP4A	007672	26-1608	#27-1683								
HLP5	007752	26-1610	#27-1684								
HLP6	010040	26-1612	#27-1685								
HLP7	010100	26-1614	#27-1686								
HLP8	010163	26-1616	#27-1687								
HWSWR	001012	#16-681	19-1019	20-1067	20-1081	20-1085	21-1117	21-1117	21-1120	21-1120	
		21-1128	21-1148	21-1148	21-1151	21-1151	21-1159	21-1171	21-1194	22-1215	
		22-1215	22-1242	22-1242	22-1254	23-1272	23-1276	23-1310	24-1324	24-1332	
		25-1459	25-1465								
LCP3	005523	#27-1631									
LCP4	005533	23-1296	#27-1632								
LCP5	005542	22-1202	#27-1633								
LCP7	005556	22-1211	#27-1635								
LEGCHR	001144	#18-995									
LINCNT	001062	#18-965	*22-1238	*22-1249							
LKS	001030	#16-688									
LN1	016615	28-2013	#32-2608								
LN1A	016647	31-2482	#32-2609								
LN10	017111	28-2037	31-2507	#32-2618							
LN2	016701	28-2016	31-2485	#32-2610							
LN3	016722	28-2019	31-2488	#32-2611							
LN4	016743	28-2022	31-2491	#32-2612							
LN5	016764	28-2025	31-2494	#32-2613							
LN6	017005	28-2028	31-2497	#32-2614							
LN7	017026	28-2031	31-2499	#32-2615							
LN8	017047	28-2034	31-2502	#32-2616							
LN8A	017070	31-2504	#32-2617								
LPS	001002	#16-674	21-1118	21-1149	25-1368	*28-1843	*28-1844	*28-1851	*28-1852	*28-1929	
		*28-1930	*28-1936	*28-1937	*28-2235						
		#16-669	20-1074	21-1113	21-1117	21-1148	21-1168	25-1367	*28-1841	*28-1842	

SYMBOL CROSS REFERENCE

CREF V01

SYMBOL	VALUE	REFERENCES
TK7	005376	26-1553 26-1577 #26-1581
TK9	005424	26-1513 #26-1591
TPB	001014	#16-682 *25-1368 *25-1372 25-1393 26-1526 26-1548 28-1721 28-1759 28-1790
		28-1879 28-1976 28-2043 28-2105 28-2138 28-2174 28-2208 29-2262 29-2282
TPS	001020	29-2305 29-2338 31-2434 31-2463 31-2513 *16-684 *25-1367 *25-1371 25-1394 25-1396 25-1399 26-1524 26-1546 28-1719
		28-1757 28-1788 28-1877 28-1974 28-2041 28-2103 28-2136 28-2172 28-2206
		29-2260 29-2280 29-2303 29-2336 31-2432 31-2461 31-2511
TSEAB	004176	24-1333 #24-1336
TSEDA	004202	#24-1338 24-1339
TSEND	004070	23-1274 23-1311 #24-1323
TSRSS	004232	#24-1346
TSRST	004236	24-1325 24-1343 #24-1348
TSTWRD	014144	30-2354 #30-2358
TY01	005571	20-1069 #27-1637
TY02	005630	22-1196 #27-1638
TY03	005673	23-1278 #27-1639
TYP	004332	15-615 #25-1381
TYPA	004342	#25-1384 25-1392 25-1406
TYPC	004352	25-1385 #25-1387
TYPD	004400	22-1245 25-1391 #25-1393 25-1403 25-1405
TYPDAT	004464	*22-1244 22-1246 #25-1384 25-1387 25-1389
TYPDO	004426	25-1395 #25-1399 25-1393 *25-1402 *25-1404 #25-1407
TYPD01	004434	25-1398 #25-1401
TYPF	004436	25-1388 #25-1402
TYPG	004450	25-1390 #25-1404
TYPINT	004314	19-1012 #25-1371
TYPSWR	005272	26-1504 26-1545 26-1552 #26-1554
TYPSWX	005336	26-1558 #26-1567
W	* 000007	#16-702 *21-1117 21-1117 #21-1117 *21-1120 21-1120 #21-1120 *21-1148 21-1148
		#21-1148 *21-1151 21-1151 #21-1151 22-1215 22-1215 #22-1215 *22-1242 22-1242
		#22-1242 #21-1117 21-1117
WAIT1	001730	#21-1117 21-1117
WAIT2	002100	#21-1120 21-1120
WAIT3	002366	#21-1148 21-1148
WAIT4	002536	#21-1151 21-1151
WAIT5	003142	#22-1215 22-1215
WAIT6	003406	#22-1242
WDD1	001736	#21-1117 21-1117
WDD2	002106	#21-1120 21-1120
WDD3	002374	#21-1148 21-1148
WDD4	002544	#21-1151 21-1151
WDD5	003166	22-1215 #22-1215 22-1215
WDD6	003414	#22-1242 22-1242
WDE1	001752	#21-1117 21-1117
WDE2	002122	#21-1120 21-1120
WDE3	002410	#21-1148 21-1148
WDE4	002560	#21-1151 21-1151
WDE5	003202	#22-1215 22-1215
WDE6	003430	#22-1242 22-1242
WER1	001770	21-1117 #21-1117
WER2	002140	21-1120 #21-1120

SYMBOL CROSS REFERENCE

CREF V01

SYMBOL	VALUE	REFERENCES
WER3	002426	21-1148 #21-1148
WER4	002576	21-1151 #21-1151
WER5	003220	22-1215 #22-1215
WER6	003446	22-1242 #22-1242
WEX1	002074	21-1117 #21-1117
WEX2	002244	21-1120 #21-1120
WEX3	002532	21-1148 #21-1148
WEX4	002702	21-1151 #21-1151
WEX5	003324	22-1215 #22-1215
WEX6	003554	22-1242 #22-1242
WORK	001124	#18-987 *21-1120 *21-1120 *21-1151 *21-1151 *22-1215 *22-1215 *22-1215 *22-1242
		*22-1242 *23-1283 *23-1285 *23-1301 *23-1303 *28-1800 *28-1801 *28-1809 *28-1817
		*28-1825 *28-1833 *28-1837 28-1838 28-1839 28-1841 *28-1890 *28-1891 *28-1899
		*28-1907 *28-1915 *28-1923 *28-1927 28-1928 *28-1967 *28-1982 *28-1989 28-1990
		*28-2049 *28-2053 *28-2057 *28-2061 *28-2065 *28-2069 *28-2073 *28-2077 *28-2082
		28-2085 28-2091 28-2092 *31-2519 *31-2523 *31-2527 *31-2531 *31-2535 *31-2539
		*31-2543 *31-2547 *31-2551 *31-2556 31-2559 31-2565 31-2566
WORKA	001126	#18-988 *21-1120 *21-1120 *21-1151 *21-1151 *22-1215 *22-1215 *22-1242 *22-1242
		*23-1284 *23-1287 *23-1302 *23-1305
WT2	005114	#26-1524 26-1525
WT3	005242	26-1517 26-1520 26-1523 #26-1546 26-1547

MACRO CROSS REFERENCE

CREF V01

MACRO NAME	REFERENCES									
\$ENABL	#17-907	19-1027								
\$ERROR	#17-716	20-1085	21-1117	21-1120	21-1128	21-1148	21-1151	21-1159	22-1215	22-1232
	22-1242	25-1365								
\$PRINT	#17-933	22-1202	23-1296							
\$PRTSN	#17-745	#20-1069	#22-1196	#23-1278						
\$SETPS	#17-924									
\$TSWRG	#17-949	20-1067	20-1081	20-1085	21-1117	21-1117	21-1120	21-1120	21-1128	21-1148
	21-1148	21-1151	21-1151	21-1159	21-1171	22-1194	22-1215	22-1215	22-1242	22-1242
	22-1254	23-1272	23-1276	23-1310	24-1324	24-1332	25-1459	25-1465		
\$TYPE	#17-941									
\$WAITI	#17-756	21-1120	21-1151	22-1215	22-1242					
\$WAITO	#17-843	#21-1117	#21-1148							