

01234567890123456789	** RSX-11M V3.2 **	DK0:[140,3]HOLD.LST;2	6-MAY-83	16:41:27	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	COPY 1 OF 1	6-MAY-83	16:41:27	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	DELETION NOT SPECIFIED	6-MAY-83	16:41:27	01234567890123456789

HH	HH	000000	LL	DDDDDDDD	
HH	HH	000000	LL	DDDDDDDD	
HH	HH	00 00	LL	DD DD	
HH	HH	00 00	LL	DD DD	
HH	HH	00 00	LL	DD DD	
HH	HH	00 00	LL	DD DD	
HH	HH	00 00	LL	DD DD	
HH	HH	00 00	LL	DD DD	
HH	HH	00 00	LL	DD DD	....
HH	HH	00 00	LL	DD DD	....
HH	HH	000000	LLLLLLLLLL	DDDDDDDD	....
HH	HH	000000	LLLLLLLLLL	DDDDDDDD	....

LL	SSSSSSSS	TTTTTTTTTT	;;;;	222222
LL	SSSSSSSS	TTTTTTTTTT	;;;;	222222
LL	SS	TT	;;;;	22 22
LL	SS	TT	;;;;	22 22
LL	SS	TT		22
LL	SS	TT		22
LL	SSSSSS	TT	;;;;	22
LL	SSSSSS	TT	;;;;	22
LL	SS	TT	;;;;	22
LL	SS	TT	;;;;	22
LL	SS	TT	;;	22
LL	SS	TT	;;	22
LLLLLLLLLL	SSSSSSSS	TT	;;	2222222222
LLLLLLLLLL	SSSSSSSS	TT	;;	2222222222

01234567890123456789	** RSX-11M V3.2 **	DK0:[140,3]HOLD.LST;2	6-MAY-83	16:41:27	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	COPY 1 OF 1	6-MAY-83	16:41:27	01234567890123456789
01234567890123456789	** RSX-11M V3.2 **	DELETION NOT SPECIFIED	6-MAY-83	16:41:27	01234567890123456789

```
1 *****
2 *
3 *
4 *   H O L D   P R I N T   P R O C E S S O R   M O D U L E
5 *
6 *                               F O R
7 *
8 *               I S I   P R I N T E R S   C O M P A T I B L E   W I T H   I B M
9 *               3 2 7 4 / 7 6   P R O T O C O L
10 *
11 *
12 *
13 *****
14 *
15 *
16 *   N A M       H O L D P R I N T
17 *
18 *   I N C L U D   D K 3 : [ 1 4 0 , 2 ] R A N G E . A C T
19 *
20 *   . R A M B .   E Q U       $ 2 0 0 0           t a r g e t / t a r g e t   v a r i a b l e   R A M
21 *   . R A M E .   E Q U       $ 2 7 F F
22 *   . R O M B .   E Q U       $ C 0 0 0           t a r g e t   p r o g r a m   b a s e   a d d r e s s
23 *   . R O M E .   E Q U       $ F F F F           t a r g e t   p r o g r a m   l i m i t
24 *
25 *
26 *   S E C T      H L D R A M , R E L , R A N G E = . R A M B . : . R A M E .   v a r i a b l e   s t o r a g e
27 *   S E C T      H L D R O M , R E L , R A N G E = . R O M B . : . R O M E .   p r o g r a m   a n d   c o n s t a n t   a r e a
28 *
29 *
30 *   I N T E R N   H O L D
31 *   I N T E R N   T M O D E 1           t e s t   f o r   m o d e   i s   L U 1   ( s u b r o u t i n e )
32 *   I N T E R N   H L D F L G           " h o l d   a c t i v e "   f l a g
33 *   I N T E R N   C A N F L G           " c a n c e l "   f l a g
34 *   I N T E R N   C A N R E Q
35 *   I N T E R N   R P T F L G           " b u f f e r   r e p r i n t "   f l a g
36 *   I N T E R N   R P T R E Q
37 *   I N T E R N   H T O F L G           " i n t e r v e n t i o n   r e q u i r e d "   ( t i m e o u t )   f l a g
38 *
39 *   E X T E R N   B U S Y           "( p r i n t )   j o b   a c t i v e "   f l a g
40 *   E X T E R N   B S Y F L G           " p h y s i c a l   p r i n t e r "   b u s y
41 *   E X T E R N   C N F L 1           c o n f i g u r a t i o n   f l a g   --   7 3 6 q
42 *   E X T E R N   D E V H P           d e v i c e   c u r r e n t   h o r i z o n t a l   p o s i t i o n   ( 1 6   b i t )
43 *   E X T E R N   D E V F W           d e v i c e   c u r r e n t   f o r m s   w i d t h   ( 1 6   b i t )
44 *   E X T E R N   D E V M P L           d e v i c e   f o r m s   l e n g t h   ( M P L )
45 *   E X T E R N   D E V M P P           d e v i c e   f o r m s   w i d t h   M P P ,   #   c o l u m n s
46 *   E X T E R N   M A X M P L           d e v i c e   l i m i t   f o r   f o r m s   l e n g t h
47 *   E X T E R N   M A X M P P           d e v i c e   l i m i t   f o r   f o r m s   w i d t h
48 *   E X T E R N   R O M M P L           d e f a u l t   f o r m s   l e n g t h
49 *   E X T E R N   R O M M P P           d e f a u l t   f o r m s   w i d t h
50 *   E X T E R N   F L T R E Q           d e v i c e   f a u l t   d e t e c t
51 *   E X T E R N   F L T F L G           d e v i c e   f a u l t   t i m e d - o u t
52 *   E X T E R N   H L D R E Q           " h o l d   r e q u e s t "   f l a g
```

=2000  
=27FF  
=C000  
=FFFF

=2000 =27FF  
=C000 =FFFF

```
53          EXTERN  HLDTIM          hold time limit in device module
54          EXTERN  SWFLAG          "momentary switch activated" flag
55          EXTERN  SWVEC           beginning of switch vector
56          EXTERN  SWBYT3          code for activated momentary switch
57          EXTERN  PSTATS          current line status
58          EXTERN  PSWSTA          current switch status
59          EXTERN  PORDER          current order
60          EXTERN  PMODE           current line specified mode
61          EXTERN  CHNFLG          LUI "in chain" flag
62          *
63          *
64          *****
65          *
66          *
67          *          EQUATES
68          *
=0005      69  HLDCOD  EQU    5          code for HOLD lamp
=0004      70  RDYCOD  EQU    4          code for READY lamp
=0006      71  CASCOD  EQU    6          code for DUAL (case) lamp
=0007      72  DSPCOD  EQU    7          code for DOUBLE (spacing) lamp
=0008      73  LPICOD  EQU    8          code for 8LPI lamp
=0009      74  TSTCOD  EQU    9          code for TEST lamp
=0020      75  DISPI   EQU   $20         code for units (1st of 3) 7-segment display
=FFFF     76  TRUE.   EQU   -1         boolean TRUE
=0000     77  FALSE.  EQU    0         boolean FALSE
=0006     78  TICK    EQU    6         clock uses 6ms ticks
=2710     79  TICKM   EQU  10000       ticks for 1 minute
=0003     80  PRTORD  EQU    3         code for print order
=0008     81  IRBIT   EQU    8         intervention required
82          *
83          *
84          *****
85          *
86          *
=0000     87          RSECT  HLDRAM
88          *
89          *
0000'    =0001     90  HLDFLG  RMB    1          "hold active" flag
0001'    =0001     91  CANREQ  RMB    1          "cancel" flag
0001'    =0001     92  CANFLG  RMB    1          "cancel" flag
0002'    =0001     93  RPTREQ  RMB    1          "buffer reprint" flag
0002'    =0001     94  RPTFLG  RMB    1          "buffer reprint" flag
0003'    =0001     95  HLDLMP  RMB    1          "hold" lamp control cell
0004'    =0001     96  RDYLMP  RMB    1          "ready" lamp control cell
0005'    =0001     97  HTOFLG  RMB    1          "hold timed out" flag
0006'    =0001     98  FIRST   RMB    1          HOLD timeout control word
0007'    =0001     99  SWSAV   RMB    1          save cell for switches
0008'    =0001    100  SSFLG   RMB    1          second mode flag
0009'    =0001    101  STATE   RMB    1          memory for alternate functions
000A'    =0003    102  TEMP    RMB    3          temporary for incremental switches
000D'    =0003    103  SWCON   RMB    3          temporary for switch status control
0010'    =0001    104  LMPRO   RMB    1          left margin procedure NOT allowed flag
```

```
105 *
106 *
107 *****
108 INCLUD DK3:[140,2]CALL.MAC
109 *      *CALL* MACRO DEFINED
171 *
172 *      THE FOLOWING ARE MACRO DEFINITIONS FOR USE BY THE SUPERVISOR
173 *
174 *
175 *      CLEAR CLOCK FLAG (6522 VIA)
176 *
177 *      Use is:
178 *          CCF
179 *
=0001 180          IFNDEF TIC.L
=0001 181          IFNDEF VIA
=8000 182 VIA      EQU      $8000
183          ENDIF
=8004 184 TIC.L   EQU      VIA+4      define timer 1 ccounter, low order byte
185          ENDIF
186 *
187 CCF      MACRO
188          TST      TIC.L          clears the clock flag
189          ENDM
190 *
191 *
192 *
193 *      INTERRUPT OFF
194 *      INTERRUPT ON
195 *
196 *      Use is:
197 *          IOF      <F,I,B> for FIRQ, IRQ, BOTH
198 *          ICON     <F,I,B> for FIRQ, IRQ, BOTH
199 *          B is equivalent to F,I
200 *
201 .IOXAR   MACRO *ARGS
202 .MSK     SET      0              initial value of mask variable
203          IFNB     <*ARGS>        do no more if no arguments
204          IRP      *ARG,<*ARGS>
205          IFIDN    <*ARG>,<B>     check for a "B"
206 .MSK     SET      .MSK!+$50     F,I set
207          ENDIF
208          IFIDN    <*ARG>,<I>     check for an "I"
209 .MSK     SET      .MSK!+$10     I set
210          ENDIF
211          IFIDN    <*ARG>,<F>     check for an "F"
212 .MSK     SET      .MSK!+$40     F set
213          ENDIF
214          ENDR
215          ENDIF
216          ENDM
217 *
```

```

218 *
219 *
220 IOF      MACRO  *ARGS
221          NLIST
222          .IOXAR <*ARGS>      form required constant
223          LIST
224          ORCC   #.MSK        turn *ARGS interrupts off
225          ENDM
226 *
227 *
228 ION      MACRO  *ARGS
229          NLIST
230          .IOXAR <*ARGS>      form required constant
231          LIST
232          ANDCC  #!N.MSK      turn *ARGS interrupts on
233          ENDM
234 *
235          INCLUD DK3:[140,2]CALL.MAC
236          'CALL' MACRO DEFINED
295 *
297 *****
298 *
299 *
=0000
300          RSECT  HLDROM
301 *
302 *
0000' 34 70      303 HOLD  PSHS   X,Y,U      save caller's registers
0002' 7D 0000'   304          TST    HLDFLG    check for first entry under hold_request
0005' 26 3A      305          BNE    1$
306 *
307 *          new hold request -- enter HOLD state, timers, lamps, etc.
308 *
0007' C6 FF      309          LDB    #TRUE.
0009' F7 0000'   310          STB    HLDFLG    hldflg indicates hold state
000C' F7 0006'   311          STB    FIRST    first phase of timeout
000F' 7F 0009'   312          CLR    STATE    initialize alternate function control
0012' 7F 0008'   313          CLR    SSFLG
0015' 86 0014*   314          LDA    PSWSTA    save switch status byte
0018' 87 0007'   315          STA    SWSAV
0018' 86 05      316          LDA    #HLDCOD   code for HCLD lamp
317+          CALL   LEDSET
322A          EXTERN LEDSET
001D' BD 0018*   326A          JSR    LEDSET    (GO TO LEDSET)
0020' F7 0003'   346          STB    HLDLMP    hold lamp starts out "on"
0023' 53
0024' 86 04      347          COMB
348          LDA    #RDYCOD   code for "READY" lamp
349+          CALL   LEDSET
0026' BD 0018*   358A          JSR    LEDSET    (GO TO LEDSET)
0029' F7 0004'   378          STB    RDYLMP    ready lamp starts out "off"
002C' 86 00      379          LDA    #FALSE.
380+          CALL   ALARM     turn off alarm
385A          EXTERN ALARM

```

```

002E'  BD 0019*      389A      JSR      ALARM          (GO TO ALARM)
                   409+      CALL     STIMER,<HOLDTO,HLDTIM>
                   414A      EXTERN   STIMER
0031'  BD 001A*      418A      JSR      STIMER          (GO TO STIMER)
0034'  03D4'         429B      FDB     HOLDTO
0036'  000F*         434B      FDB     HLDTIM
                   443+      CALL     STIMER,<FLASH,FLSHTM>
0038'  BD 001A*      452A      JSR      STIMER          (GO TO STIMER)
003B'  03AC'         463B      FDB     FLASH
003D'  009C'         468B      FDB     FLSHTM
003F'  20 16         477       BRA     3$
                   478      *
                   479      *      force device status update
                   480      *
                   481+     1$      CALL     DEVSTA
                   486A     EXTERN   DEVSTA
0041'  BD 0018*      490A      JSR      DEVSTA          (GO TO DEVSTA)
                   510      *
                   511      *      check for momentary switch activated (pushed)
                   512      *
0044'  B6 0010*      513       LDA     SWFLAG
0047'  27 0E         514       BEQ     3$              switch flag not set, exit
                   515      *
                   516      *      switch pressed, dispatch to appropriate handler
                   517      *
0049'  F6 0012*      518       LDB     SWBYT3          get switch code
004C'  86 00         519       LDA     #FALSE.       clear switch flag
004E'  B7 0010*      520       STA     SWFLAG
0051'  58           521       ASLB                    double switch code for table index
0052'  8E 005A'      522       LDX     #TRAVEC-2     table has no zero element
0055'  AD 95         523       JSR     [B,X]          go to switch handler
0057'  35 70         524       PULS   X,Y,U        restore caller's registers
0059'  39           525       RTS                    (NOTE: this RTS is also for 'null', below)
                   526      *
                   527      *      switch trap table
                   528      *
005A'  0059'         529       FDB     4$              0  'null' switch
005C'  00A5'         530      TRAVEC  FDB     INDEX          1  index switch
005E'  00D5'         531       FDB     REPR           2  reprint
0060'  00C5'         532       FDB     PA            3  PA 1
0062'  00C5'         533       FDB     PA            4  PA 2
0064'  02EF'         534       FDB     PRHLD          5  hold print
0066'  00F5'         535       FDB     CANCEL         6  cancel
0068'  0111'         536       FDB     TESTSW         7  test
006A'  00B5'         537       FDB     FRMFD          8  form feed
006C'  0301'         538       FDB     PRENB          9  enable print (exit hold)
006E'  0298'         539       FDB     LPISW          A  lines/inch
0070'  0281'         540       FDB     SPCING         B  line spacing
0072'  0267'         541       FDB     SWCASE         C  case
0074'  0238'         542       FDB     IN100         D  100's increment
0076'  025E'         543       FDB     IN10           E  10's  "
0078'  0263'         544       FDB     IN1            F  1's   "

```

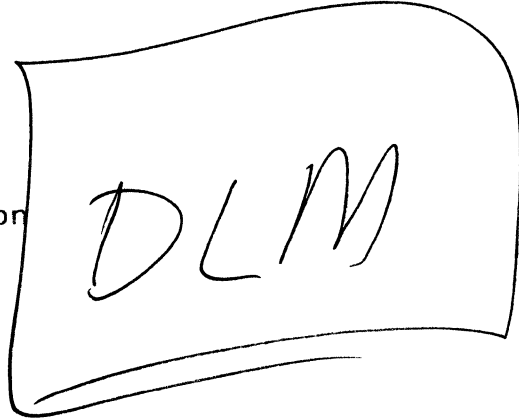
007A'	014F'	545	FDB	SETUP	10	setup print
007C'	0233'	546	FDB	RESET	11	reset
007E'	0181'	547	FDB	CONFIG	12	configure (NVM save)
0080'	0195'	548	FDB	DIAGN	13	diagnose ????
0082'	0201'	549	FDB	STMP	14	mpp set
0084'	01CF'	550	FDB	STMPL	15	form length set
0086'	00A4'	551	FDB	NULL	16	not defined code
0088'	00A4'	552	FDB	NULL	17	
008A'	00A4'	553	FDB	NULL	18	
008C'	00A4'	554	FDB	NULL	19	
008E'	00A4'	555	FDB	NULL	1A	
0090'	00A4'	556	FDB	NULL	1B	
0092'	00A4'	557	FDB	NULL	1C	
0094'	00A4'	558	FDB	NULL	1D	
0096'	00A4'	559	FDB	NULL	1E	
0098'	02B5'	560	FDB	PRIMSW	1F	primary function switch
009A'	02D8'	561	FDB	SECSW	20	secondary function switch
		562	*			
		563	*			
009C'	0053	564	FLSHTM	FDB	500/TICK	.5 sec @ 6 ms intervals
		565	*			
		566+	STACHK	CALL	DEVSTA	check for device error
009E'	BD 001B*	575A		JSR	DEVSTA	(GO TO DEVSTA)
00A1'	BA 0002*	595		ORA	BSYFLG	... or busy
00A4'	39	596	NULL	RTS		
		597	*			
		598	*			

```
600 *****
601 *****
602 *
603 *
604 *      switch handlers
605 *
606 *
607 *****
608 *
609 *
610 *      INDEX
611 *
00A5' 7D 0009' 612 INDEX  TST      STATE
00A8' 1026 0209 613          LBNE     PRIMSW
614 *
00AC'  BD 009E' 615          JSR      STACHK   check for legal
00AF'  26 03    616          BNE      1$
617+          CALL     DEVLFF   perform a line feed
622A          EXTERN   DEVLFF
00B1'  BD 001C* 626A          JSR      DEVLFF   (GO TO DEVLFF)
00B4'  39      646 1$      RTS
647 *
648 *****
649 *
650 *      FORM FEED
651 *
00B5' 7D 0009' 652 FRMFD  TST      STATE
00B8' 1026 01F9 653          LBNE     PRIMSW
654 *
00BC'  BD 009E' 655          JSR      STACHK   check for legal
00BF'  26 03    656          BNE      1$
657+          CALL     DEVFF   perform a form feed
662A          EXTERN   DEVFF
00C1'  BD 001D* 666A          JSR      DEVFF   (GO TO DEVFF)
00C4'  39      686 1$      RTS
687 *
688 *****
689 *
690 *      PA switch 1 or 2
691 *
00C5' 7D 0009' 692 PA      TST      STATE   check for legal
00C8' 1026 01E9 693          LBNE     PRIMSW
694 *
00CC'  BD 03E7' 695          JSR      TMODE1   must be LU1 mode for PA switches
00CF'  27 03    696          BEQ      1$      branch if MODE not LU1
697+          CALL     PASWIT   SSA module knows how to handle PA switches
702A          EXTERN   PASWIT
00D1'  BD 001E* 706A          JSR      PASWIT   (GO TO PASWIT)
00D4'  39      726 1$      RTS
727 *
728 *****
729 *
```



```
730 * reprint switch
731 *
00D5* 7D 0009* 732 REPR TST STATE check for legal
00D8* 1026 01D9 733 LBNE PRIMSW
734 *
00DC* 8D 009E* 735 JSR STACHK check for legal
00DF* 26 13 736 BNE 1$
00E1* 86 0001* 737 LDA BUSY must be busy to reprint
00E4* 27 0E 738 BEQ 1$
00E6* 86 0015* 739 LDA PORDER must be printing to reprint
00E9* 84 07 740 ANDA #7
00EB* 81 03 741 CMPA #PRTORD
00ED* 26 05 742 BNE 1$
00EF* 86 FF 743 LDA #TRUE. set REPRINT flag
00F1* 87 0002* 744 STA RPTFLG
00F4* 39 745 1$ RTS
746 *
747 *****
748 *
749 * cancel switch
750 *
00F5* 7D 0009* 751 CANCEL TST STATE
00F8* 1026 01B9 752 LBNE PRIMSW
753 *
00FC* 8D 009E* 754 JSR STACHK check for legal
00FF* 26 0F 755 BNE 1$
0101* 8D 03E7* 756 JSR TMODE1 must be in LUI mode to cancel
0104* 27 0A 757 BEQ 1$
0106* 86 0017* 758 LDA CHNFLG must also be "in chain"
0109* 27 05 759 BEQ 1$
760 *
010B* 86 FF 761 LDA #TRUE. mode=LUI, set CANCEL request
010D* 87 0001* 762 STA CANFLG
0110* 39 763 1$ RTS
764 *
765 *****
766 *
767 * test switch
768 *
0111* 7D 0009* 769 TESTSW TST STATE
0114* 1026 019D 770 LBNE PRIMSW
771 *
0118* 8D 009E* 772 JSR STACHK check for legal
011B* 26 31 773 BNE 1$
774 *
775 * check for forms width procedure (736q)
776 *
011D* 7D 0003* 777 TST CNFL1 check configuration flag 1
0120* 27 1B 778 BEQ 10$
0122* 7D 0010* 779 TST LMPRO check for too late
0125* 26 16 780 BNE 10$
0127* 86 0000" 781 LDA SWVEC+2 first HOLD PRINT, check permission switch
```

```
012A* 85 20          782          BITA    %#00100000    --switch 10
012C* 27 0F          783          BEQ     10$
                                784          *
                                785          *    legal to do forms width procedure -- do it
                                786          *
012E* 86 0000"      787          LDA     SWVEC+4      save current MPL
0131* 34 02          788          PSHS   A
                                789+         CALL   SETDFW        "device" does the action
                                794A        EXTERN SETDFW
                                798A        JSR    SETDFW        (GO TO SETDFW)
0133* BD 001F*      818          PULS   A            restore current MPL
0136* 35 02          819          STA   SWVEC+4
0138* B7 0000"      820          BRA   1$            exit
013B* 20 11          821          *
                                822          *    ordinary tests
                                823          *
013D* 86 09          824          10$    LDA     #TSTCOD      turn on TEST LED
013F* C6 FF          825          LDB   #TRUE.
                                826+         CALL   LEDSET
0141* BD 0018*      835A        JSR    LEDSET        (GO TO LEDSET)
                                855+         CALL   TEST
                                860A        EXTERN TEST
0144* BD 0020*      864A        JSR    TEST        (GO TO TEST)
0147* 86 09          884          LDA     #TSTCOD      turn off TEST LED
0149* C6 00          885          LDB   #FALSE.
                                886+         CALL   LEDSET
0148* BD 0018*      895A        JSR    LEDSET        (GO TO LEDSET)
                                915          *
014E* 39             916          1$    RTS
                                917          *
                                918          *****
                                919          *
                                920          *    SETUP switch pressed -- print a test character for forms alignment
                                921          *
014F* 7D 0009*      922          SETUP TST     STATE      check for legal
0152* 1026 015F     923          LBNE  PRIMSW
                                924          *
0156* BD 009E*      925          JSR    STACHK
0159* 26 13          926          BNE   1$
                                927          *
                                928          *    if 'carrage' is not beyond MPP, print a test character
                                929          *
015B* FC 0004*      930          LDD   DEVHP        current position
015E* 10B3 0005*    931          CMPD  DEVPW
0162* 2D 03          932          BLT   2$
                                933+         CALL   DEVCR        return 'carriage' before print
                                938A        EXTERN DEVCR
0164* BD 0021*      942A        JSR    DEVCR        (GO TO DEVCR)
0167* 108E 017F'    962          2$    LDY   #SUBUF        send the setup print buffer
                                963+         CALL   DEVPUT
                                968A        EXTERN DEVPUT
0168* BD 0022*      972A        JSR    DEVPUT        (GO TO DEVPUT)
```



```

          992 *
016E' 39 993 1$ RTS
          994 *
          995 * device coded buffer for "setup" print -- 1 char at current position
          996 *
016F' 00 00 00 00 00 00 997 FCB 0,0,0,0,0,0,0,0 pad 16 bytes before count
0175' 00 00
0177' 00 00 00 00 00 00 998 FCB 0,0,0,0,0,0,12,0 required by DEVPUT
017D' 0C 00
017F' 01 999 SUBUF FCB 1 1 character
0180' A7 1000 FCB $A7 DBC "H"
1001 *
1002 *****
1003 *
1004 * CONFIG switch pressed -- load configuration to NVM
1005 *
0181' 7D 0009' 1006 CONFIG TST STATE check for legal
0184' 1026 012D 1007 LBNE PRIMSW
1008 *
1009+ CALL NVMRD get copy of current NVM
1014A EXTERN NVMRD
0188' BD 0023* 1018A JSR NVMRD (GO TO NVMRD)
1038+ CALL COPSW copy current settings to array
1043A EXTERN COPSW
018B' BD 0024* 1047A JSR COPSW (GO TO COPSW)
1067+ CALL NVMWR write it back out
1072A EXTERN NVMWR
018E' BD 0025* 1076A JSR NVMWR (GO TO NVMWR)
1096+ CALL NVMREL release copy memory
1101A EXTERN NVMREL
0191' BD 0026* 1105A JSR NVMREL (GO TO NVMREL)
1125 *
0194' 39 1126 1$ RTS
1127 *
1128 *****
1129 *
1130 * DIAGNOSE switch pressed -- select test
1131 *
0195' B6 0009' 1132 DIAGN LDA STATE see if mode set for selection
0198' 26 12 1133 BNE 20$
1134 *
019A' 86 03 1135 LDA #3 set state
019C' B7 0009' 1136 STA STATE
019F' F6 0000" 1137 LDB SWVEC+1 get current value
01A2' 54 1138 LSRB
01A3' 54 1139 LSRB
01A4' 54 1140 LSRB
01A5' 54 1141 LSRB
01A6' 4F 1142 CLRA
01A7' BD 0370' 1143 JSR SETDIS set up display of value, etc.
01AA' 20 22 1144 BRA 1$ and exit
1145 *
```

```

1146 * non-zero state. were we setting test?
1147 *
01AC' 81 03 1148 20$ CMPA #3 3 is state number for test set
01AE' 26 1B 1149 BNE 2$
1150 *
01B0' F6 0000" 1151 LDB SWVEC+1 set default value
01B3' C4 0F 1152 ANDB #$0F
01B5' F7 0000" 1153 STB SWVEC+1
1154 *
01B8' 8D 0396' 1155 JSR GETNUM convert entered number to binary
01BB' 1083 0009 1156 CMPD #9 check for greater than allowed
01BF' 2E 0A 1157 BGT 2$ if illegal value, go to primary
1158 *
01C1' 58 1159 LSLB shift value
01C2' 58 1160 LSLB
01C3' 58 1161 LSLB
01C4' 58 1162 LSLB
01C5' FA 0000" 1163 ORB SWVEC+1 merge with other bits
01C8' F7 0000" 1164 STB SWVEC+1
01CB' 7E 02B5' 1165 2$ JMP PRIMSW revert to primary mode
1166 *
01CE' 39 1167 1$ RTS
1168 *
1169 *****
1170 *
1171 * SET FL switch pressed -- set form length (MPL)
1172 *
01CF' B6 0009' 1173 STMP L LDA STATE check for function legal
01D2' 26 0E 1174 BNE 20$
1175 *
01D4' 86 02 1176 LDA #2 set state
01D6' B7 0009' 1177 STA STATE
01D9' F6 0000" 1178 LDB SWVEC+4 get current value
01DC' 4F 1179 CLRA extend
01DD' 8D 0370' 1180 JSR SETDIS set up display of value, etc.
01E0' 20 1E 1181 BRA 1$ and exit
1182 *
1183 * non-zero state. were we setting MPL (Form Length)?
1184 *
01E2' 81 02 1185 20$ CMPA #2 2 is state number for MPL set
01E4' 26 17 1186 BNE 3$
1187 *
01E6' F6 000A* 1188 LDB ROMMPL set default value
01E9' F7 0000" 1189 STB SWVEC+4
1190 *
01EC' 8D 0396' 1191 JSR GETNUM convert entered number to binary
01EF' 4D 1192 TSTA check for greater than allowed
01F0' 26 08 1193 BNE 2$ cannot have high order bits
01F2' F1 0008* 1194 CMPB MAXMPL compare with MAX
01F5' 22 03 1195 BHI 2$ if illegal value, go to primary
1196 *
01F7' F7 0000" 1197 STB SWVEC+4 store new value

```

```
1198 *
1199+ 2$ CALL SETDFL update device
1204A EXTERN SETDFL
01FA* BD 0027* 1208A JSR SETDFL (GO TO SETDFL)
01FD* 7E 02B5* 1228 3$ JMP PRIMSW revert to primary mode
1229 *
0200* 39 1230 1$ RTS
1231 *
1232 *****
1233 *
1234 * SET MPP switch pressed -- set maximum line length
1235 *
0201* B6 0009* 1236 STMPPLDA STATE check for function legal
0204* 26 0E 1237 BNE 20$
1238 *
0206* 86 01 1239 LDA #1 set state
0208* B7 0009* 1240 STA STATE
020B* F6 0000" 1241 LDB SWVEC+5 get current value
020E* 4F 1242 CLRA
020F* BD 0370* 1243 JSR SETDIS set up display of value, etc.
0212* 20 1E 1244 BRA 1$ and exit
1245 *
1246 * non-zero state. were we setting MPP?
1247 *
0214* 81 01 1248 20$ CMPA #1 1 is state number for MPP set
0216* 26 17 1249 BNE 3$
1250 *
0218* F6 0008* 1251 LDB ROMMPP set default value
021B* F7 0000" 1252 STB SWVEC+5
1253 *
021E* BD 0396* 1254 JSR GETNUM convert entered number to binary
0221* 4D 1255 TSTA check returned value -- no high bits allowed
0222* 26 08 1256 BNE 2$
0224* F1 0009* 1257 CMPB MAXMPP compare with MAX value
0227* 22 03 1258 BHI 2$ if illegal value, go to primary
1259 *
0229* F7 0000" 1260 STB SWVEC+5 store new value
1261+ 2$ CALL SETDFW update device
022C* BD 001F* 1270A JSR SETDFW (GO TO SETDFW)
022F* 7E 02B5* 1290 3$ JMP PRIMSW revert to primary mode
1291 *
0232* 39 1292 1$ RTS
1293 *
1294 *****
1295 *
1296 * RESET -- ???
1297 *
0233* 7D 0009* 1298 RESET TST STATE
0236* 1026 007B 1299 LBNE PRIMSW
1300 *
1301 *
023A* 39 1302 1$ RTS
```

```
1303 *
1304 *****
1305 *
1306 *      HUNDREDS switch pressed -- increment hundreds value
1307 *
023B' C6 02 1308 IN100 LDB #2 relative address of hundreds temporary
1309 *
023D' B6 0009' 1310 DIGIT LDA STATE see if incrementing is even legal
0240' 27 19 1311 BEQ 1$ STATE must be non-zero but not > max allowed
0242' 81 04 1312 CMPA #4
0244' 2C 15 1313 BGE 1$
1314 *
0246' 8E 000A' 1315 LDX #TEMP
0249' A6 85 1316 LDA B,X get current value
024B' 4C 1317 INCA
024C' 1C FE 1318 ANCC #!N1 clear carry for adjust
024E' 19 1319 DAA
024F' 84 0F 1320 ANDA #$F keep digit part only
0251' A7 85 1321 STA B,X store new value
1322 *
0253' 1E 89 1323 EXG A,B swap registers to match LEDSET requirements
0255' 8B 14 1324 ADDA #20 compute address of digit to load
1325+ CALL LEDSET send new value to display
0257' BD 0018* 1334A JSR LEDSET (GO TO LEDSET)
025A' 39 1354 RTS
1355 *
1356 *      incrementing not legal
1357 *
025B' 7E 02B5' 1358 1$ JMP PRIMSW
1359 *
1360 *****
1361 *
1362 *      TENS switch pressed -- increment tens value
1363 *
025E' C6 01 1364 IN10 LDB #1 relative address of tens temporary
0260' 7E 023D' 1365 JMP DIGIT process
1366 *
1367 *****
1368 *
1369 *      UNITS switch pressed -- increment units value
1370 *
0263' 5F 1371 IN1 CLRB relative address of units temporary
0264' 7E 023D' 1372 JMP DIGIT process
1373 *
1374 *****
1375 *
1376 *      CASE switch pressed -- toggle dual/mono mode
1377 *
0267' 7D 0009' 1378 SWCASE TST STATE check for legal
026A' 1026 0047 1379 LBNE PRIMSW
1380 *
026E' F6 0000" 1381 LDB SWVEC+1 get current
```

```
0271* C8 04      1382      EORB   #4      toggle
0273* F7 0000"   1383      STB    SWVEC+1
0276* C4 04      1384      ANDB   #4      detect bit for LED
0278* 86 06      1385      LDA    #CASCOD address DUAL LED
                   1386+      CALL   LEDSET
027A* BD 0018*   1395A      JSR    LEDSET   (GO TO LEDSET)
                   1415+      CALL   SETCAS   force device to update case flag
                   1420A      EXTERN SETCAS
027D* BD 0028*   1424A      JSR    SETCAS   (GO TO SETCAS)
0280* 39         1444      RTS
                   1445      *
                   1446      *****
                   1447      *
                   1448      *      SPACING switch pressed -- toggle single/double mode
                   1449      *
0281* 7D 0009*   1450      SPCING TST     STATE
0284* 1026 002D  1451      LBNE   PRIMSW
                   1452      *
0288* F6 0000"   1453      LDB    SWVEC+1   get current
028B* C8 02      1454      EORB   #2      toggle
028D* F7 0000"   1455      STB    SWVEC+1   store new
0290* C4 02      1456      ANDB   #2      detect bit for LED
0292* 86 07      1457      LDA    #DSPCOD  address DOUBLE LED
                   1458+      CALL   LEDSET
0294* BD 0018*   1467A      JSR    LEDSET   (GO TO LEDSET)
                   1487+      CALL   SETDFL   force device to update form size
0297* BD 0027*   1496A      JSR    SETDFL   (GO TO SETDFL)
029A* 39         1516      RTS
                   1517      *
                   1518      *****
                   1519      *
                   1520      *      "LPI" switch pressed -- toggle 6/8 LPI mode
                   1521      *
0298* 7D 0009*   1522      LPISW TST     STATE
029E* 1026 0013  1523      LBNE   PRIMSW
                   1524      *
02A2* F6 0000"   1525      LDB    SWVEC+1   get current
02A5* C8 01      1526      EORB   #1      toggle
02A7* F7 0000"   1527      STB    SWVEC+1   store new
02AA* C4 01      1528      ANDB   #1      detect bit for LED
02AC* 86 08      1529      LDA    #LPICOD  address LPI LED
                   1530+      CALL   LEDSET
02AE* BD 0018*   1539A      JSR    LEDSET   (GO TO LEDSET)
                   1559+      CALL   SETDFL   force device to update form size
02B1* BD 0027*   1568A      JSR    SETDFL   (GO TO SETDFL)
02B4* 39         1588      RTS
                   1589      *
                   1590      *****
                   1591      *
                   1592      *      "primary function" switch pressed
                   1593      *      or other switch defaults to PRIMSW because of illegal action
                   1594      *
```

```
02B5' 86 00          1595 PRIMSW LDA    #FALSE.    set switches to "1st" mode
                   1596+ CALL    SWMODE
                   1601A EXTERN  SWMODE
02B7' BD 0029*      1605A JSR    SWMODE    (GO TO SWMODE)
02BA' B7 0008'      1625 STA    SSFLG    do not flash HOLD light
02BD' 7F 0009'      1626 CLR    STATE    clear 2nd function state
02C0' C6 FF         1627 LDB    #TRUE.    make sure HOLD light is on
02C2' 86 05         1628 LDA    #HLDCOD
                   1629+ CALL    LEDSET
02C4' BD 0018*      1638A JSR    LEDSET    (GO TO LEDSET)
02C7' F7 0003'      1658 STB    HLDLMP
02CA' 86 20         1659 LDA    #DISP1    blank digit display
                   1660+ CALL    LEDSET
02CC' BD 0018*      1669A JSR    LEDSET    (GO TO LEDSET)
02CF' 4C            1689 INCA    next digit
                   1690+ CALL    LEDSET
02D0' BD 0018*      1699A JSR    LEDSET    (GO TO LEDSET)
02D3' 4C            1719 INCA    next digit
                   1720+ CALL    LEDSET
02D4' BD 0018*      1729A JSR    LEDSET    (GO TO LEDSET)
02D7' 39            1749 RTS
                   1750 *
                   1751 *****
                   1752 *
                   1753 *      "secondary function" switch pressed
                   1754 *
02D8' 86 FF         1755 SECSW LDA    #TRUE.    set switches to "2nd" mode
                   1756+ CALL    SWMODE
02DA' BD 0029*      1765A JSR    SWMODE    (GO TO SWMODE)
02DD' B7 0008'      1785 STA    SSFLG    internal switch set -- flash HOLD light
                   1786 *
02E0' C6 0A         1787 LDB    #A        load display with indicator
02E2' 86 20         1788 LDA    #DISP1
                   1789+ 1$ CALL    LEDSET
02E4' BD 0018*      1798A JSR    LEDSET    (GO TO LEDSET)
02E7' C3 0065       1818 ADDD   #101     next digit, different character
02EA' 81 22         1819 CMPA   #DISP1+2 check for done
                   1820 *
02EC' 2F F6         1821 BLE    1$
02EE' 39            1822 RTS
                   1823 *
                   1824 *****
                   1825 *
                   1826 *      HOLD PRINT switch pressed -- clear alarm
                   1827 *
02EF' 7D 0009'      1828 PRHLD TST    STATE
02F2' 1026 FFBF     1829 LBNE   PRIMSW
                   1830 *
02F6' 86 00         1831 LDA    #FALSE.
                   1832+ CALL    ALARM
02F8' BD 0019*      1841A JSR    ALARM    (GO TO ALARM)
                   1861 *
```



```
02FB* 7D 0003*      1862      TST      CNFL1      check for 736q
02FE* 26 01         1863      BNE      PRENB     if 736, hold means enable
0300* 39            1864      RTS
1865      *
1866      *****
1867      *
1868      *      enable switch pressed -- leave hold state
1869      *
0301* 7D 0009*      1870      PRENB   TST      STATE
0304* 1026 FFAD     1871      LBNE    PRIMSW
1872      *
0308* BD 009E*      1873      JSR     STACHK     check for legal
030B* 26 60         1874      BNE     4$
1875+    CALL     CTIMER,<HOLDTO>      clear hold timeout timer
1880A    EXTERN  CTIMER
030D* BD 002A*      1884A    JSR     CTIMER     (GO TO CTIMER)
0310* 03D4*         1895B    FDB     HOLDTO
1904+    CALL     CTIMER,<FLASH>      flasher is now superfluous
0312* BD 002A*      1913A    JSR     CTIMER     (GO TO CTIMER)
0315* 03AC*         1924B    FDB     FLASH
0317* C6 00         1933      LDB     #FALSE.
0319* 34 01         1934      PSHS    CC          timer interrupt not allowed here
1935+    IOF     <I>
0318* 1A 10         1962A    ORCC    #.MSK       turn I interrupts off
031D* F7 000E*      1963      STB     HLDREQ
0320* F7 0000*      1964      STB     HLDFLG
0323* 35 01         1965      PULS    CC          restore existing interrupt masks
0325* F7 000D*      1966      STB     FLTFLG
0328* 86 05         1967      LDA     #HLDCOD    hold lamp off
1968+    CALL     LEDSET
032A* BD 0018*      1977A    JSR     LEDSET     (GO TO LEDSET)
032D* C6 FF         1997      LDB     #TRUE.     ready lamp on
032F* 86 04         1998      LDA     #RDYCOD
1999+    CALL     LEDSET
0331* BD 0018*      2008A    JSR     LEDSET     (GO TO LEDSET)
0334* F7 0010*      2028      STB     LMPRO      disable (further) margin setting
2029      *
2030      *      see if intervention required needs cleared
2031      *
0337* B6 0013*      2032      LDA     PSTATS
033A* 84 08         2033      ANDA   #IRBIT
033C* 27 0E         2034      BEQ     2$          branch if no
2035      *
033E* 86 01         2036      LDA     #1          disable code
0340* 8E 036E*      2037      LDX     #CIRVEC
0343* 3F            2038      SWI
0344* 5D            2039      TSTB
0345* 26 F7         2040      BNE     3$          if failed, retry
0347* 86 00         2041      LDA     #FALSE.    clear HTOFLG
0349* B7 0005*      2042      STA     HTOFLG
2043      *
2044      *      see if switch status change needs to be reported
```

```

2045 *
034C' B6 0000" 2046 2$ LDA SWVEC+1 the way it should be
034F' B8 0007' 2047 EDRA SWSAV the way it was
0352' 84 07 2048 ANDA #7 bits currently supported
0354' 27 17 2049 BEQ 4$ branch if no changes
2050 *
0356' B6 0000" 2051 LDA SWVEC+1 update control vector
0359' 84 07 2052 ANDA #7
035B' B7 000F' 2053 STA SWCON+2
2054 *
035E' 86 01 2055 21$ LDA #1 "switch transition" bit & disable code
0360' B7 000D' 2056 STA SWCON
0363' 7F 000E' 2057 CLR SWCON+1
0366' 8E 000D' 2058 LDX #SWCON address of control vector to X
0369' 3F 2059 SWI
036A' 5D 2060 TSTB
036B' 26 F1 2061 BNE 21$ loop if failed (already disabled)
2062 *
036D' 39 2063 4$ RTS
2064 *
036E' 0008 2065 CIRVEC FDB IRBIT clear IR control
2066 *
2067 *****
```

```
2069 *****
2070 *
2071 * routine converts D for display in 7-segment display and for update
2072 * in TEMP...TEMP+2
2073 *
0370' 108E 000A' 2074 SETDIS LDY #TEMP address at which to store BCD
0374' 8E 03AB' 2075 LDX #TEN address of divisor (10)
2076+ CALL DIVIDE (D)/((X))
2081A EXTERN DIVIDE
0377' 8D 002B* 2085A JSR DIVIDE (GO TO DIVIDE)
037A' A7 A0 2105 STA ,Y+ store remainder as first digit
037C' 4F 2106 CLRA clear high order and divide again
2107+ CALL DIVIDE
037D' 8D 002B* 2116A JSR DIVIDE (GO TO DIVIDE)
0380' A7 A0 2136 STA ,Y+ ten's digit
0382' E7 A4 2137 STB ,Y hundred's
2138 *
0384' 86 22 2139 LDA #DISP1+2 display address for hundred's
2140+ CALL LEDSET
0386' 8D 0018* 2149A JSR LEDSET (GO TO LEDSET)
0389' E6 A2 2169 LDB , -Y ten's
038B' 4A 2170 DECA
2171+ CALL LEDSET
038C' 8D 0018* 2180A JSR LEDSET (GO TO LEDSET)
038F' E6 A2 2200 LDB , -Y unit's
0391' 4A 2201 DECA
2202+ CALL LEDSET
0392' 8D 0018* 2211A JSR LEDSET (GO TO LEDSET)
0395' 39 2231 RTS return with BCD in TEMP and displayed
2232 *
2233 * convert BCD in TEMP to binary in D
2234 *
0396' F6 000C' 2235 GETNUM LDB TEMP+2 hundred's digit
0399' 86 0A 2236 LDA #10
039B' 3D 2237 MUL compute "hundred's * 10"
039C' FB 000B' 2238 ADDB TEMP+1 compute "hundred's * 10 + ten's"
039F' 86 0A 2239 LDA #10
03A1' 3D 2240 MUL compute "hundred's * 100 + ten's * 10"
03A2' ED E3 2241 STD ,--S save above
03A4' F6 000A' 2242 LDB TEMP unit's
03A7' 4F 2243 CLRA extend
03A8' E3 E1 2244 ADDD ,S++ add back above
03AA' 39 2245 RTS
2246 *
03AB' 0A 2247 TEN FCB 10 constant ten
2248 *
2249 *****
```

```
2251 *****
2252 *
2253 *
2254 *      "flash" interrupt
2255 *
03AC* BD 03E7* 2256 FLASH JSR TMODE1      check for mode=LUI
03AF* 27 03    2257      BEQ 1$          if not mode LUI, set RDY lamp OFF
03B1* B8 0004* 2258      EORA RDYLMP      exor A (true) with RDYLMP to complement it
03B4* B7 0004* 2259 1$ STA RDYLMP      store result
03B7* 1F 89    2260      TFR A,B          state code to B for LEDSET
03B9* 86 04    2261      LDA #RDYCOD      lamp code ro A
2262+          CALL LEDSET
03BB* BD 0018* 2271A     JSR LEDSET      (GO TO LEDSET)
03BE* F6 0003* 2291      LDB HLDLMP      compute state of HOLD lamp
03C1* F8 0008* 2292      EORB SSFLG
03C4* F7 0003* 2293      STB HLDLMP      store it
03C7* 86 05    2294      LDA #HLDCOD
2295+          CALL LEDSET      and load it
03C9* BD 0018* 2304A     JSR LEDSET      (GO TO LEDSET)
2324+          CALL STIMER,<FLASH,FLSHTM> retrigger timer
03CC* BD 001A* 2333A     JSR STIMER      (GO TO STIMER)
03CF* 03AC*   2344B     FDB FLASH
03D1* 009C*   2349B     FDB FLSHTM
03D3* 39      2358      RTS          done
2359 *
2360 *
2361 *      timeout interrupt -- five/ten minutes hold print
2362 *
2363 *
03D4* 73 0006* 2364 HOLDTO COM FIRST      which 5min t/o is this
03D7* 27 06    2365      BEQ 1$
2366 *
2367 *      FIRST became true again -- 10 min t/o elapsed
2368 *
03D9* 86 FF    2369      LDA #TRUE.      set HTOFLG
03DB* B7 0005* 2370      STA HTOFLG
03DE* 39      2371      RTS          return to clock processor
2372 *
2373 *      first t/o -- 5 min elapsed, reset for another 5
2374 *
2375+ 1$      CALL STIMER,<HOLDTO,HLDTIM>
03DF* BD 001A* 2384A     JSR STIMER      (GO TO STIMER)
03E2* 03D4*   2395B     FDB HOLDTO
03E4* 000F*   2400B     FDB HLDTIM
03E6* 39      2409      RTS
2410 *
2411 *
2412 *****
2413 *****
2414 *
2415 *
2416 *      TMODE1 -- a routine to test for MODE = LUI
```

```
2417 *
2418 * routine tests PMODE bits 5,6,7 for value 6, indicating LU1 mode
2419 * return is in A,CC. A is true for LU1, false for other. CC is set
2420 * by a TSTA before the return.
2421 *
2422 * B is also modified
2423 *
2424 *
03E7' 86 FF 2425 TMODE1 LDA #TRUE. preset A
03E9' F6 0000" 2426 LDB PMODE+1 get mode byte w/ required info
03EC' C4 07 2427 ANDB #7 bits 5,6,7
03EE' C1 06 2428 CMPB #6 B=6?
03F0' 27 01 2429 BEQ 255$ branch to exit if equal
03F2' 43 2430 COMA set A false if not equal
03F3' 4D 2431 255$ TSTA set CC for user branch
03F4' 39 2432 RTS
2433 *
2434 *
2435 END
```

ALARM	0019 EX	BSYFLG	0002 EX	BUSY	0001 EX	CALL	Macro	CANCEL	00F5'
CANFLG	0001' IN	CANREQ	0001' IN	CASCOD	0006	CCF	Macro	CHNFLG	0017 EX
CIRVEC	036E'	CNFL1	0003 EX	CONFIG	0181'	COPSW	0024 EX	CTIMER	002A EX
DEVCR	0021 EX	DEVFF	001D EX	DEVFW	0005 EX	DEVHP	0004 EX	DEVLF	001C EX
DEVMP	0006 EX	DEVMP	0007 EX	DEVPUT	0022 EX	DEVSTA	001B EX	DIAGN	0195'
DIGIT	023D'	DISP1	0020	DIVIDE	002B EX	DSPCOD	0007	FALSE.	0000
FIRST	0006'	FLASH	03AC'	FLSHTM	009C'	FLTFLG	000D EX	FLTREQ	000C EX
FRMFD	00B5'	GETNUM	0396'	HLDCOD	0005	HLDFLG	0000' IN	HLDLMP	0003'
HLDREQ	000E EX	HLDTIM	000F EX	HOLD	0000' IN	HOLDTC	03D4'	HTOFLG	0005' IN
INDEX	00A5'	INI	0263'	IN10	025E'	IN100	023B'	IOF	Macro
ION	Macro	IRBIT	0008	LEDSET	0018 EX	LMPRO	0010'	LPICOD	0008
LPISW	029B'	MAXMPL	0008 EX	MAXMPP	0009 EX	NULL	00A4'	NVMRD	0023 EX
NVMREL	0026 EX	NVMWR	0025 EX	PA	00C5'	PASWIT	001E EX	PMODE	0016 EX
PORDER	0015 EX	PRENB	0301'	PRHLD	02EF'	PRIMSW	02B5'	PRTORD	0003
PSTATS	0013 EX	PSWSTA	0014 EX	RDYCOD	0004	RDYLMP	0004'	REPR	00D5'
RESET	0233'	ROMMPL	000A EX	ROMMPP	000B EX	RPTFLG	0002' IN	RPTREQ	0002' IN
SECSW	02D8'	SETCAS	0028 EX	SETDFL	0027 EX	SETDFW	001F EX	SETDIS	0370'
SETUP	014F'	SPCING	0281'	SSFLG	0008'	STACHK	009E'	STATE	0009'
STIMER	001A EX	STMPL	01CF'	STMPP	0201'	SUBUF	017F'	SWBYT3	0012 EX
SWCASE	0267'	SWCON	000D'	SWFLAG	0010 EX	SWMODE	0029 EX	SWSAV	0007'
SWVEC	0011 EX	TEMP	000A'	TEN	03AB'	TEST	0020 EX	TESTSW	0111'
TICK	0006	TICKM	2710	TMODE1	03E7' IN	TRAVEC	005C'	TRUE.	FFFF
TSTCOD	0009	TIC.L	8004	VIA	8000	.IOXAR	Macro	.MSK	=0010
.RAMB.	2000	.RAME.	27FF	.RDMB.	C000	.ROME.	FFFF		

No errors detected