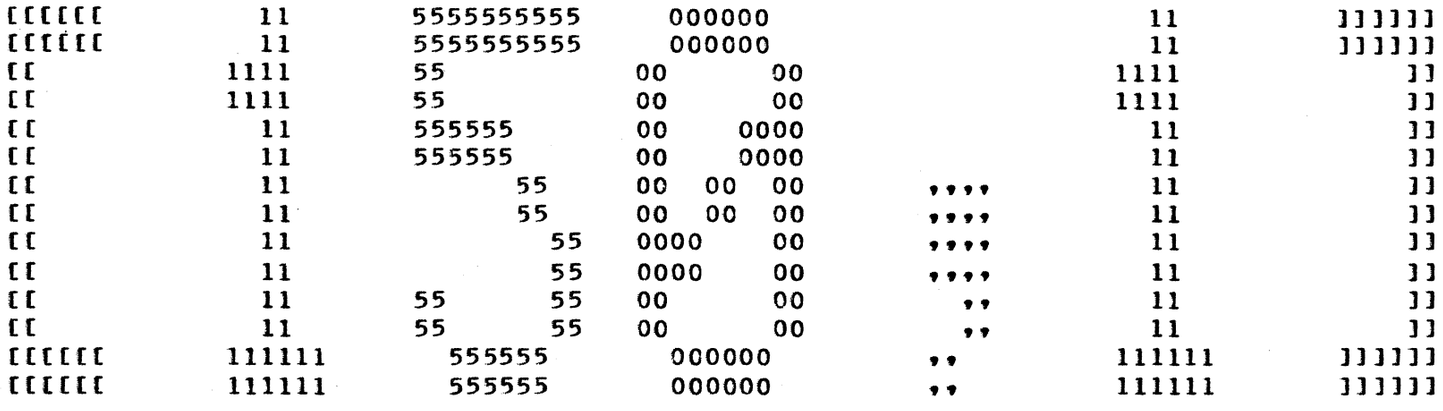

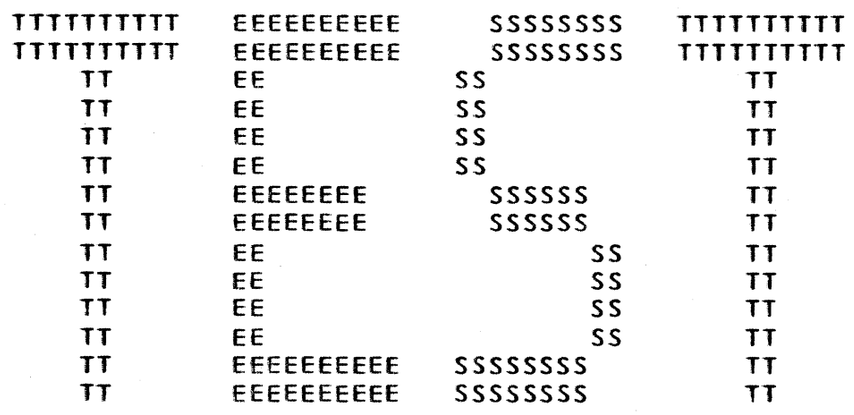

```
1   ********************************************************************************
2   ********************************************************************************
3   ********************************************************************************
4   ***                                                                          ***
5   ***                                                                          ***
6   ***                                                                          ***
7   ***            TEST MODULE FOR  ISI  PRINTERS COMPATIBLE                     ***
8   ***                  with  IBM  3274/6 PROTOCOL                              ***
9   ***                                                                          ***
10  ***                                                                          ***
11  ***                  INTERFACE SYSTEMS, INC.                                 ***
12  ***                  ANN ARBOR, MICHIGAN  48103                              ***
13  ***                                                                          ***
14  ***                                                                          ***
15  ***                                                                          ***
16  ***            FOR MOTOROLA  6809  OR COMPATIBLE MICROPROCESSOR              ***
17  ***                  USING A GENERATION III INTERFACE                        ***
18  ***                                                                          ***
19  ***                                                                          ***
20  ***                                                                          ***
21  ********************************************************************************
22  ********************************************************************************
23  ********************************************************************************
24  *
25  *
26  *                    BY:     INTERFACE SYSTEMS, INCORPORATED
27  *                            462 JACKSON PLAZA
28  *                            ANN ARBOR, MICHIGAN  48103
29  *
30  *                            RICHARD L. COLE
31  *
32  *                            COPYRIGHT 1982, 1983
33  *                                    BY
34  *                            INTERFACE SYSTEMS, INC.
35  *
36  *                            ALL RIGHTS RESERVED
37  *
38  *
39  ********************************************************************************
40  ********************************************************************************
41  *
42  *
43          NAM     TEST MODULE
44  *
45  *
46          EXTERN  BUSY            printer BUSY flag
47          EXTERN  DEVMDN          printer model number
48          EXTERN  DEVPD           device print density
49          EXTERN  ENAFLG          printer ENABLED flag (coax-protocol)
50          EXTERN  EXSTAT          "special status"
51          EXTERN  LSTCOM          last coax command (bottom board only)
52          EXTERN  LSTORD          last ORDER received
```

```
                                53          EXTERN  PIDENT          terminal id field
                                54          EXTERN  SWFLAG          front panel switch pressed flag
                                55          EXTERN  SWVEC           base of switch vector
                                56          EXTERN  VERSIO          software version number
                                57          XREF    DEVTST          Device test print
                                58          XREF    DEVCR           Device carriage return
                                59          XREF    DEVLF           Device line feed
                                60          XREF    DEVOUT          Device output code
                                61          XREF    DEVPUT          Device output line
                                62  *
                                63  *
                                64          INTERN  AUTDMP
                                65          INTERN  CONCHK          return continue/cancel flag for multi-line test*
                                66          INTERN  TEST
                                67          XDEF    TSTBUF          Test print buffer
                                68  *
                                69  *
                                70  *
                                71  *
=C000 =FFF0                     72          SECT    TSTROM,REL,RANGE=0C000H:0FFF0H
=2000 =27FF                     73          SECT    TSTRAM,REL,RANGE=02000H:027FFH
                                74  *
                                75  *
=C000                           76  ROMBEG  EQU     $C000           ROM beginning address
=0000                           77  ROMEND  EQU     $0000           ROM ending address + 1
=FFFF                           78  TRUE.   EQU     -1              boolean "true"
=0000                           79  FALSE.  EQU     0               boolean "false"
=0008                           80  ADCOD   EQU     8               test switch code for auto-hex-dump
=0010                           81  DBC_SP  EQU     $10             DBC code for (space)
=0020                           82  DBC_0   EQU     $20             DBC graphic 0
=0029                           83  DBC_9   EQU     $29             DBC graphic 9
=00A2                           84  DBC_C   EQU     $A2             DBC graphic C
=00AA                           85  DBC_K   EQU     $AA             DBC graphic K
=00B7                           86  DBC_X   EQU     $B7             DBC graphic X
                                87  *
                                88  ********************************************************************************
```

```
                                    90  ********************************************************************************
                                    91  *
                =0000               92          RSECT   TSTRAM
                                    93  *
0000'           =0001               94  GDUMP   RMB     1               "good dump" (not aborted) flag
0001'           =0002               95  LSTADR  RMB     2               address beypnd end of dump
                =0010               96          RMB     16              pre-buffer space
0013'           =0050               97  TSTBUF  RMB     80              test print buffer
```

```
                                99
        =0000                   100             RSECT   TSTROM
                                101     *
0000'   BD 0015'                102     TEST    JSR     TSWIT           get, mask & shift test switch code
0003'   48                      103             LSLA                    double for index
0004'   8E 001D'                104             LDX     #TSTTAB         base of table
0007'   AD 96                   105             JSR     [A,X]           go to appropriate subroutine
0009'   39                      106             RTS
                                107     *
000A'   BD 0015'                108     AUTDMP  JSR     TSWIT           get test switches
000D'   81 08                   109             CMPA    #ADCOD          is auto dump specified?
000F'   26 03                   110             BNE     99$             branch if no
                                111
0011'   BD 00BF'                112             JSR     FDUMP           yes, do full dump
0014'   39                      113     99$     RTS
                                114     *
                                115     *
                                116     *       get, mask and shift test switch code
                                117     *
0015'   B6 0000"                118     TSWIT   LDA     SWVEC+1         get current code
0018'   44                      119             LSRA                    shift right 4 places (masking in process)
0019'   44                      120             LSRA
001A'   44                      121             LSRA
001B'   44                      122             LSRA
001C'   39                      123             RTS
                                124     *
                                125     *       test decode table
                                126     *
001D'   0033'                   127     TSTTAB  FDB     VERS            0 - user version print
001F'   0033'                   128             FDB     VERS            1 - print version, etc.
0021'   0087'                   129             FDB     STAT            2 - print "special status"
0023'   00BF'                   130             FDB     FDUMP           3 - dump all
0025'   00E6'                   131             FDB     SDUMP           4 - dump stack/var ram only
0027'   000C*                   132             FDB     DEVTST          5 - device dependant test
0029'   0032'                   133             FDB     MEMTST          6 - do a memory check ??????????????????
002B'   0031'                   134             FDB     NULL            7 - normal, no test
002D'   0031'                   135             FDB     NULL            8 - auto-dump, no test
002F'   0171'                   136             FDB     TPRNT           9 - Infinite test print
```

```
                                138   *
                                139   *       no test
                                140   *
                                141   *
0031'  39                       142   NULL    RTS
                                143   *
                                144   ********************************************************************************
                                145   *
                                146   *       memory test  --  to be defined
                                147   *
0032'  39                       148   MEMTST  RTS
```

```
                                150   *
                                151   *       print version, model, checksum (of rom)
                                152   *
0033' BD 0250'                  153   VERS    JSR     NL              test prints should always start with NL
0036' CE 000B*                  154           LDU     #VERSIO         address containing version
0039' 108E 0013'                155           LDY     #TSTBUF         Point to test print buffer
003D' E6 C0                     156           LDB     0,U+            Load version byte count
003F' 34 04                     157           PSHS    B               Save byte count
0041' 27 11                     158           BEQ     20$             No version string
                                159
0043' C1 3E                     160           CMPB    #62             Version string too long ?
0045' 23 04                     161           BLS     10$             No !
                                162
0047' C6 3E                     163           LDB     #62             Load max. byte count
0049' E7 E4                     164           STB     0,S             Save new byte count
004B' A6 C0                     165   10$     LDA     0,U+            Load code
004D' BD 000F*                  166           JSR     DEVOUT          Transfer to test print buffer
0050' 6A E4                     167           DEC     0,S             Last code ?
0052' 26 F7                     168           BNE     10$             No !
                                169
0054' 32 61                     170   20$     LEAS    1,S             Clean up stack
0056' B6 0002*                  171           LDA     DEVMDN          output model number
0059' BD 022F'                  172           JSR     TSTHEX
005C' 86 10                     173           LDA     #DBC_SP         follow with a space
005E' BD 000F*                  174           JSR     DEVOUT
0061' 86 A2                     175           LDA     #DBC_C          a C
0063' BD 000F*                  176           JSR     DEVOUT
0066' 86 AA                     177           LDA     #DBC_K          a K
0068' BD 000F*                  178           JSR     DEVOUT
006B' 86 10                     179           LDA     #DBC_SP         another space
006D' BD 000F*                  180           JSR     DEVOUT
0070' 8E C000                   181           LDX     #ROMBEG         compute ROM checksum by words
0073' CC 0000                   182           LDD     #$0000          clear sum
0076' E3 81                     183   30$     ADDD    ,X++            add a word to sum
0078' 8C 0000                   184           CMPX    #ROMEND         End?
007B' 26 F9                     185           BNE     30$
                                186
007D' BD 021D'                  187           JSR     TST4HX          output checksum
0080' 86 10                     188           LDA     #DBC_SP         output a space
0082' BD 000F*                  189           JSR     DEVOUT
0085' 20 07                     190           BRA     STAT1           jump into STAT, below
```

```
                                192   *
                                193   *       "special status"        --      Xxxyzab
                                194   *
                                195   *               X is an indicator character
                                196   *               xx represents the "special status" byte
                                197   *               y represents the BUSY flag, F or 0
                                198   *               z represents the ENABLED flag, F or 0
                                199   *               a represents the last ORDER, 0 to 3
                                200   *               b represents the last COMMAND, 0 to 3
                                201   *
0087' BD 0250'                  202   STAT    JSR     NL              begin with a new line
008A' 108E 0013'                203           LDY     #TSTBUF         Point to test print buffer
                                204
008E' 86 B7                     205   STAT1   LDA     #DBC_X          X
0090' BD 000F*                  206           JSR     DEVOUT
0093' B6 0005*                  207           LDA     EXSTAT          the "special status" byte
0096' BD 0224'                  208           JSR     TST2HX
0099' B6 0001*                  209           LDA     BUSY            BUSY flag
009C' BD 022F'                  210           JSR     TSTHEX
009F' B6 0004*                  211           LDA     ENAFLG          ENABLED flag
00A2' BD 022F'                  212           JSR     TSTHEX
00A5' B6 0007*                  213           LDA     LSTORD          last ORDER
00A8' 84 03                     214           ANDA    #3
00AA' BD 022F'                  215           JSR     TSTHEX
00AD' B6 0006*                  216           LDA     LSTCOM          last COMMAND
00B0' 84 03                     217           ANDA    #3
00B2' BD 022F'                  218           JSR     TSTHEX
00B5' BD 023D'                  219           JSR     TSTPUT          Output test print line
00B8' BD 0250'                  220           JSR     NL
00BB' BD 0250'                  221           JSR     NL
00BE' 39                        222           RTS
```

```
                                224   *
                                225   *       full dump  --  print buffer, eab, stack & var
                                226   *
00BF'  BD 0033'                 227   FDUMP   JSR     VERS            print version/status as header
00C2'  8E 0000                  228           LDX     #$0000          dump starting at addr=0
00C5'  B6 0000"                 229           LDA     PIDENT+2        buffer size (high byte)
00C8'  5F                       230           CLRB
00C9'  1F 03                    231           TFR     D,U             to buffer_size-1
00CB'  BD 00F6'                 232           JSR     DUMP            dump ((X)) to ((U))
00CE'  4D                       233           TSTA                    was dump aborted?
00CF'  27 21                    234           BEQ     DMPEND          branch if yes
                                235   *
                                236   *       eab?
                                237   *
00D1'  7D 0000"                 238           TST     PIDENT+1        eab present? (ID byte 1, bit 0)
00D4'  2A 10                    239           BPL     SDUMP           branch if no
00D6'  8E 1000                  240           LDX     #$1000          eab always starts here
00D9'  1F 10                    241           TFR     X,D             copy to D
00DB'  BB 0000"                 242           ADDA    PIDENT+2        calculate end
00DE'  1F 03                    243           TFR     D,U
00E0'  BD 00F6'                 244           JSR     DUMP
00E3'  4D                       245           TSTA                    was dump aborted?
00E4'  27 0C                    246           BEQ     DMPEND          branch if yes
                                247   *
                                248   *       stack and variable ram  ************************************************
                                249   *
00E6'  BD 0033'                 250   SDUMP   JSR     VERS            print version/status as header
00E9'  8E 2000                  251           LDX     #$2000          beginning
00EC'  CE 2800                  252           LDU     #$2800          end
00EF'  BD 00F6'                 253           JSR     DUMP
00F2'  BD 0250'                 254   DMPEND  JSR     NL              skip a line after dump
00F5'  39                       255           RTS                     done
```

```
                                257   *
                                258   *
                                259   *       DUMP SUBROUTINE
                                260   *
                                261   *       dumps memory from (X) to (U)
                                262   *
                                263   *
00F6'  86 FF                    264   DUMP    LDA     #TRUE.          initialize "good (unaborted) dump" flag
00F8'  B7 0000'                 265           STA     GDUMP
00FB'  FF 0001'                 266           STU     LSTADR          Save last address
00FE'  BD 0250'                 267           JSR     NL              ok to skip a line before dumping
                                268   *
                                269   *       dump a line, then check for cancel (TEST switch pressed)
                                270   *
0101'  108E 0013'               271           LDY     #TSTBUF         Point to test print buffer
0105'  BD 0144'                 272   10$     JSR     PNTLIN
0108'  BD 025A'                 273           JSR     CONCHK          check for front panel abort
0108'  27 26                    274           BEQ     40$
                                275   *
                                276   *       check for last word dumped.  check for lines to be skipped.
                                277   *
010D'  BC 0001'                 278           CMPX    LSTADR          Last byte ?
0110'  27 24                    279           BEQ     90$             Yes, exit !
                                280
0112'  1F 13                    281           TFR     X,U             starting scan point
0114'  A6 1F                    282           LDA     -1,X            last byte dumped
                                283   *
                                284   *       scan loop
                                285   *
0116'  11B3 0001'               286   20$     CMPU    LSTADR          last line passed?
011A'  24 04                    287           BCC     30$             Yes, print last line !
                                288
011C'  A1 C0                    289           CMPA    0,U+            check a byte
011E'  27 F6                    290           BEQ     20$             if equal, go look some more
                                291   *
                                292   *       found something to dump  --  should a line be skipped?
                                293   *
0120'  33 5F                    294   30$     LEAU    -1,U            back up to data to dump
0122'  1F 30                    295           TFR     U,D             compute base address of its line
0124'  C4 F0                    296           ANDB    #$F0
0126'  34 06                    297           PSHS    D
0128'  AC E1                    298           CMPX    0,S++           same as already pointed to by X?
012A'  27 D9                    299           BEQ     10$             if yes, go dump it
                                300
012C'  1F 01                    301           TFR     D,X             no, update X and skip line
012E'  BD 0250'                 302           JSR     NL
0131'  20 D2                    303           BRA     10$
                                304   *
                                305   *       dump aborted
                                306   *
0133'  B7 0000'                 307   40$     STA     GDUMP           set good dump false
                                308   *
```

```
                                309    *       dump is complete
                                310    *
0136'  B6 0000'                 311    90$     LDA     GDUMP           return no abort/abort flag
0139'  39                       312            RTS
                                313    *
                                314    *       control table
                                315    *
013A'  00 00 00 FF FF 00        316    TAB2    FCB     FALSE.,FALSE.,FALSE.,TRUE.,TRUE.,FALSE.,FALSE.,FALSE.
0140'  00 00
0142'  FF 00                    317            FCB     TRUE.,FALSE.
```

```
                                319   *
                                320   *       dump line routine  --  display (X), then ((X))...((X+15)), then NL
                                321   *
0144'  1F 10                    322   PNTLIN  TFR     X,D             print base address
0146'  BD 021D'                 323           JSR     TST4HX
0149'  86 10                    324           LDA     #DBC_SP         4 spaces
014B'  BD 000F*                 325           JSR     DEVOUT
014E'  BD 000F*                 326           JSR     DEVOUT
0151'  BD 000F*                 327           JSR     DEVOUT
0154'  BD 000F*                 328           JSR     DEVOUT
0157'  A6 80                    329   10$     LDA     0,X+            get a byte
0159'  BD 0224'                 330           JSR     TST2HX
015C'  86 10                    331           LDA     #DBC_SP         2 spaces
015E'  BD 000F*                 332           JSR     DEVOUT
0161'  BD 000F*                 333           JSR     DEVOUT
0164'  1F 10                    334           TFR     X,D             check for line done
0166'  C4 0F                    335           ANDB    #$F
0168'  26 ED                    336           BNE     10$             No !
                                337
016A'  BD 023D'                 338           JSR     TSTPUT          Output test print line
016D'  BD 0250'                 339           JSR     NL              new_line
0170'  39                       340           RTS
```

```
                                342   *
                                343   *
                                344   *       test print  --  alphabets numbers and special characters
                                345   *
                                346   *               indefinate repeat until cancelled
                                347   *
                                348   *
0171'  BD 0250'                 349   TPRNT   JSR     NL              test prints should always start with NL
0174'  108E 01A1'               350   10$     LDY     #TP1            print the test line
0178'  BD 0010#                 351           JSR     DEVPUT
017B'  BD 000D#                 352           JSR     DEVCR           return to beginning of line
017E'  108E 0202'               353           LDY     #TP2            print underlines
0182'  BD 0010#                 354           JSR     DEVPUT
0185'  BD 0250'                 355           JSR     NL              new line
0188'  BD 0250'                 356           JSR     NL
018B'  BD 025A'                 357           JSR     CONCHK          continue?
018E'  26 E4                    358           BNE     10$             branch if yes
                                359
0190'  39                       360           RTS                     no, stop
```

```
                                    362   *
                                    363   *       device coded buffer for test print  ALPHAalphanumbersspecials
                                    364   *
                                    365   *               character density is coded for default  --  10cpi
                                    366   *
0191' 00 00 00 00 00 00             367           FCB     0,0,0,0,0,0,0,0
0197' 00 00
0199' 00 00 00 00 01 51             368           FCB     0,0,0,0,1,81,0,0
019F' 00 00
01A1' 50                            369   TP1     FCB     80              buffer has 80 chars
      =0010                         370           RADIX   16
01A2' A0 A1 A2 A3 A4 A5             371           FCB     $A0,$A1,$A2,$A3,$A4,$A5,$A6,$A7,$A8,$A9,$AA,$AB,$AC     A-M
01A8' A6 A7 A8 A9 AA AB
01AE' AC
01AF' AD AE AF B0 B1 B2             372           FCB     $AD,$AE,$AF,$B0,$B1,$B2,$B3,$B4,$B5,$B6,$B7,$B8,$B9     N-Z
01B5' B3 B4 B5 B6 B7 B8
01BB' B9
01BC' 80 81 82 83 84 85             373           FCB     80,81,82,83,84,85,86,87,88,89,8A,8B,8C                   a-m
01C2' 86 87 88 89 8A 8B
01C8' 8C
01C9' 8D 8E 8F 90 91 92             374           FCB     8D,8E,8F,90,91,92,93,94,95,96,97,98,99                   n-z
01CF' 93 94 95 96 97 98
01D5' 99
01D6' 20 21 22 23 24 25             375           FCB     20,21,22,23,24,25,26,27,28,29                            0-9
01DC' 26 27 28 29
01E0' 0D 11 12 13 14 18             376           FCB     0D,11,12,13,14,18,19,1A,2E                       (='"/?!$%
01E6' 19 1A 2E
01E9' 30 31 32 33 34 35             377           FCB     30,31,32,33,34,35,$BE,$BF,0C                     &-.,:+;*)
01EF' BE BF 0C
      =000A                         378           RADIX   10
                                    379   *
                                    380   *       underline string
                                    381   *
01F2' 00 00 00 00 00 00             382           FCB     0,0,0,0,0,0,0,0
01F8' 00 00
01FA' 00 00 00 00 01 1B             383           FCB     0,0,0,0,1,27,0,0
0200' 00 00
0202' 1A                            384   TP2     FCB     26
      =0010                         385           RADIX   16
0203' 2F 2F 2F 2F 2F 2F             386           FCB     2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F
0209' 2F 2F 2F 2F 2F 2F
020F' 2F
0210' 2F 2F 2F 2F 2F 2F             387           FCB     2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F,2F
0216' 2F 2F 2F 2F 2F 2F
021C' 2F
      =000A                         388           RADIX   10
```

```
                                390   *
                                391   *       SERVICE ROUTINES FOR DUMPS AND OTHER TEST PRINTS
                                392   *
                                393   *       NOTE:  the following routines are strung together to save space
                                394   *              at the expense of some readability
                                395   *
                                396   *       output (D) as 4 hex digits
                                397   *
021D'  34 04                    398   TST4HX  PSHS    B               save lo-order
021F'  BD 0224'                 399           JSR     TST2HX          output hi-order
0222'  35 02                    400           PULS    A               get lo-order back
                                401   *       drop into out2 to output lo-order byte
                                402   *
                                403   *       output (A) as 2 hex digits
                                404   *
0224'  34 02                    405   TST2HX  PSHS    A               save
0226'  44                       406           LSRA                    right justify hi-digit
0227'  44                       407           LSRA
0228'  44                       408           LSRA
0229'  44                       409           LSRA
022A'  BD 022F'                 410           JSR     TSTHEX          output A bits 7-4
022D'  35 02                    411           PULS    A
                                412   *       drop into HEX to output lo-digit
                                413   *
                                414   *       output a (hex) digit from A3-A0
                                415   *
022F'  84 0F                    416   TSTHEX  ANDA    #$F             mask
0231'  8B 20                    417           ADDA    #DBC_0          convert to DBC
0233'  81 29                    418           CMPA    #DBC_9          digit 0 to 9?
0235'  2F 02                    419           BLE     TSTBYT          branch if yes
                                420
0237'  8B 76                    421           ADDA    #$A0-$2A        convert to upper DBC for A through F
                                422   *       drop into CHAR to output the digit
                                423   *
                                424   *       output DBC char in A to the buffer
                                425   *
0239'  BD 000F*                 426   TSTBYT  JSR     DEVOUT          no, place char in buffer
023C'  39                       427           RTS
```

```
                                429   *
                                430   * 'TSTPUT' output test print line to device
                                431   *
                                432   * Entry : Y point to 'TSTBUF'
                                433   *
                                434   * Exit  : Y points to 'TSTBUF'
                                435   *
023D'  E6 A4                    436   TSTPUT  LDB     0,Y             Load byte count
023F'  27 0E                    437           BEQ     99$             No, line to output !
                                438
0241'  5C                       439           INCB                    Increment to next print position
0242'  E7 3D                    440           STB     -3,Y            Save end print position
0244'  C6 01                    441           LDB     #1              Load first print position
0246'  E7 3C                    442           STB     -4,Y            Save begin print position
0248'  6F 3E                    443           CLR     -2,Y            Use device default print density
024A'  BD 0010*                 444           JSR     DEVPUT          Output test print buffer to device
024D'  6F A4                    445           CLR     0,Y             Null test print buffer
024F'  39                       446   99$     RTS                     Return
                                447   *
                                448   *       generate a "new_line"  --  CR-LF
                                449   *
0250'  34 30                    450   NL      PSHS    Y,X             save X
0252'  BD 000D*                 451           JSR     DEVCR           CR
0255'  BD 000E*                 452           JSR     DEVLF           LF
0258'  35 B0                    453           PULS    X,Y,PC          restore X
```

```
                                455   *
                                456   *       routine checks for TEST switch on front panel pressed
                                457   *               presense causes a return of FALSE. in A with Z set
                                458   *               absense causes a return of TRUE. in A with Z clear
                                459   *               SWFLAG is always cleared by this routine
                                460   *
025A' B6 0009*                  461   CONCHK  LDA     SWFLAG          check for front panel switch
025D' 43                        462           COMA
025E' 26 0B                     463           BNE     99$
                                464   *
                                465   *       lda     #false.         switch flag set, prepare to clear it
                                466   *
0260' F6 0000"                  467           LDB     SWVEC+3         what switch was it?
0263' B7 0009*                  468           STA     SWFLAG          clear the flag
0266' C1 07                     469           CMPB    #7              is switch code = TEST?
0268' 26 F0                     470           BNE     CONCHK          if switch was not TEST, go set A TRUE.
                                471
026A' 4D                        472           TSTA                    set condition code (A remains FALSE.)
026B' 39                        473   99$     RTS
                                474
                                475           END
```

ADCOD	0008		AUTDMP	000A'IN		BUSY	0001 EX		CONCHK	025A'IN		DBC$C	00A2
DBC$K	00AA		DBC$SP	0010		DBC$X	00B7		DBC$0	0020		DBC$9	0029
DEVCR	000D EX		DEVLF	000E EX		DEVMDN	0002 EX		DEVOUT	000F EX		DEVPD	0003 EX
DEVPUT	0010 EX		DEVTST	000C EX		DMPEND	00F2'		DUMP	00F6'		ENAFLG	0004 EX
EXSTAT	0005 EX		FALSE.	0000		FDUMP	00BF'		GDUMP	0000'		LSTADR	0001'
LSTCOM	0006 EX		LSTORD	0007 EX		MEMTST	0032'		NL	0250'		NULL	0031'
PIDENT	0008 EX		PNTLIN	0144'		ROMBEG	C000		ROMEND	0000		SDUMP	00E6'
STAT	0087'		STAT1	008E'		SWFLAG	0009 EX		SWVEC	000A EX		TAB2	013A'
TEST	0000'IN		TPRNT	0171'		TP1	01A1'		TP2	0202'		TRUE.	FFFF
TSTBUF	0013'IN		TSTBYT	0239'		TSTHEX	022F'		TSTPUT	023D'		TSTTAB	001D'
TST2HX	0224'		TST4HX	021D'		TSWIT	0015'		VERS	0033'		VERSIO	000B EX

No errors detected