IDENTIFICATION

PRODUCT CODE: AC-F658A-MC

PRODUCT NAME: CXVTVA0 VTV30J/H-VT30H MODULE

PRODUCT DATE: OCT 1, 1979

MAINTAINER: COMPUTER SPECIAL SYSTEMS
DIGITAL EQUIPMENT CO. LTD.
READING
BERKS. U.K.

VTV30-H/J OR VT30-H DEC/X11 MODULE

PROGRAM DESCRIPTION

1. ABSTRACT

VTV IS AN IOMOD WHICH EXERCISES A VTV30-H/J OR VT30-H COLOUR VIDEO DISPLAY CONTROLLER. THE EXERCISE CONSISTS OF CONTINUALLY UPDATING THE CONTENTS OF THE PICTURE STORE SO AS TO PRODUCE A MOVING PATTERN ON THE SCREEN OF THE T.V. MONITOR. NO SOFTWARE CHECKING IS CARRIED OUT, BUT ERRORS IN THE PICTURE STORE OR CHARACTER STORE AREAS WILL BE CLEARLY VISIBLE ON THE DISPLAY.

2. REQUIREMENTS

2.1 HARDWARE

VTV30-H/J OR VT30-H CONTROLLER WITH T.V. MONITOR

2.2 STORAGE

VTV REQUIRES 500 WORDS OF STORAGE.

3. PASS DEFINITION

ONE PASS OF THE VTV MODULE CONSISTS OF SIX COMPLETE SCREEN COLOUR CHANGES.

4. EXECUTION TIME

ONE PASS OF THE VTV MODULE TAKES APPROXIMATELY ONE MINUTE WHEN RUNNING ALONE.

5. CONFIGURATION REQUIREMENTS

5.1 DEFAULT PARAMETERS

DVADR: 164000
VECTOR: 170
BR1: 4

5.2 REQUIRED PARAMETERS

NONE.

6. DEVICE/OPTION SETUP

ENSURE THAT THE CONTROLLER MODULES ARE CORRECTLY INSTALLED AND CONNECTED TO THE COLOUR T.V. MONITOR. THE MONITOR SHOULD BE POWERED UP.

7. MODULE OPERATION

TEST SEQUENCE

A. SET UP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES.

B. SET DISPLAY CHARACTER MATRIX (6 X 6 OR 8 X 8).

C. LOAD ENTIRE CHARACTER SET WITH DUMMY CHARACTERS (252 PATTERN).

D. LOAD SPECIAL PATTERNS FOR USE IN DISPLAY INTO CHARACTERS 0 - 26 (6 X 6) OR CHARACTERS 0 - 36 (8 X 8).

E. SET INITIAL COLOUR AND CHARACTER COMBINATION.

F. CHECK IF ALL COLOURS DONE. IF SO, REPORT END OF PASS AND GO TO E.

G. DRAW PATTERN ON SCREEN UNTIL FULL OF FOREGROUND COLOUR.

H. INCREMENT COLOUR COUNTER, GO TO F.

VISUAL OPERATION

THE PICTURE STORE IS PRESET TO BLUE BACKGROUND. A RED TRIANGLE IS THEN DRAWN, STARTING AT THE BOTTOM CENTRE OF THE SCREEN AND EXPANDING UPWARDS UNTIL THE SCREEN IS FULL OF RED FOREGROUND. THE SEQUENCE IS THEN REPEATED USING GREEN FOREGROUND ON RED BACKGROUND, BLUE ON GREEN, GREEN ON BLUE, RED ON GREEEN AND BLUE ON RED. AT END OF PASS, THE DISPLAY IS DISABLED THEN RESTART TAKES PLACE AT THE BEGINNING OF THE COLOUR SEQUENCE.

```
1
2                       AT ALL TIMES, THE EDGES OF THE TRIANGLE SHOULD APPEAR
3                       SMOOTH AND MOVING STEADILY.  THE APPEARANCE OF ANY
4                       VERTICALLY STRIPED CHARACTERS ON THE SCREEN INDICATES
5                       THAT AN ILLEGAL CHARACTER IS BEING DISPLAYED AT THAT
6                       POSITION.  ANY OTHER PICTURE STORE OR CHARACTER STORE
7                       MALFUNCTIONS SHOULD BE JUST AS OBVIOUS.
8
9
10              8.      OPERATION OPTIONS
11
12                      SR1 = 0, 6 X 6 CHARACTER MATRIX.
13
14                      SR1 = 1, 8 X 8 CHARACTER MATRIX.
15
16
17              9.      NON STANDARD PRINTOUT
18
19                      NONE.
20
```

```
1                       .ENDR
2 000000                IOMOD   <VTVA >,164000,170,4,0,0,1,0
  000000( ) 1212        .OTUNE  140000,VTVA ,164000,170,4,0,0,1,0
                        .TITLE  VTVA DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      28-NOV-78
                ;       CSSUK
                        .LIST   BIN
                ;*************************************************************
  000000        BEGIN:
  000000    126 MODNAM: .ASCII  /VTVA / ;MODULE NAME.
  000001    124
  000002    126
  000003    101
  000004    040
  000005    000 XFLAG:  .BYTE   OPEN    ;USED TO KEEP TRACK OF WRUFF USAGE
  000006 164000 ADDR:   164000+0        ;1ST DEVICE ADDR.
  000010 000170 VECTOR: 170+0           ;1ST DEVICE VECTOR.
  000012    200 BR1:    .BYTE   PRTY4+0 ;1ST BR LEVEL.
  000013    000 BR2:    .BYTE   PRTY0+0 ;2ND BR LEVEL.
  000014 000001 DVID1:  0+1             ;DEVICE INDICATOR 1.
  000016 000000 SR1:    OPEN            ;SWITCH REGISTER 1
  000020 000000 SR2:    OPEN            ;SWITCH REGISTER 2
  000022 000000 SR3:    OPEN            ;SWITCH REGISTER 3
  000024 000000 SR4:    OPEN            ;SWITCH REGISTER 4
                ;*************************************************************
  000026 140000 STAT:   140000          ;STATUS WORD.
  000030 000620'INIT:   START           ;MODULE START ADDR.
  000032 000224'SPOINT: MODSP           ;MODULE STACK POINTER.
  000034 000000 PASCNT: 0               ;PASS COUNTER.
  000036 000001 ICONT:  1               ;# OF ITERATIONS PER PASS=1
  000040 000000 ICOUNT: 0               ;LOC TO COUNT ITERATIONS
  000042 000000 SOFCNT: 0               ;LOC TO SAVE TOTAL SOFT ERRORS
  000044 000000 HRDCNT: 0               ;LOC TO SAVE TOTAL HARD ERRORS
  000046 000000 SOFPAS: 0               ;LOC TO SAVE SOFT ERRORS PER PASS
  000050 000000 HRDPAS: 0               ;LOC TO SAVE HARD ERRORS PER PASS
  000052 000000 SYSCNT: 0               ;# OF SYS ERRORS ACCUMULATED
  000054 000000 RANNUM:0                ;HOLDS RANDOM# WHEN RAND MACRO IS CALLED
  000056        CONFIG:                 ;RESERVED FOR MONITOR USE
  000056 000000 RES1:   0               ;RESERVED FOR MONITOR USE
  000060 000000 RES2:   0               ;RESERVED FOR MONITOR USE
  000062 000000 SVR0:   OPEN            ;LOC TO SAVE R0.
  000064 000000 SVR1:   OPEN            ;LOC TO SAVE R1.
  000066 000000 SVR2:   OPEN            ;LOC TO SAVE R2.
  000070 000000 SVR3:   OPEN            ;LOC TO SAVE R3.
  000072 000000 SVR4:   OPEN            ;LOC TO SAVE R4.
  000074 000000 SVR5:   OPEN            ;LOC TO SAVE R5.
  000076 000000 SVR6:   OPEN            ;LOC TO SAVE R6.
  000100 000000 CSRA:   OPEN            ;ADDR OF CURRENT CSR.
  000102        SBADR:                  ;ADDR OF GOOD DATA, OR
  000102 000000 ACSR:   OPEN            ;CONTENTS OF CSR.
  000104        WASADR:                 ;ADDR OF BAD DATA, OR
  000104 000000 ASTAT:  OPEN            ;STATUS REG CONTENTS.
  000106        ERRTYP:                 ;TYPE OF ERROR
  000106 000000 ASB:    OPEN            ;EXPECTED DATA.
  000110 000000 AWAS:   OPEN            ;ACTUAL DATA.
  000112 001216'RSTRT:  RESTRT          ;RESTART ADDRESS AFTER END OF PASS
  000114 000000 WDTO:   OPEN            ;WORDS TO MEMORY PER ITERATION
```



```
  000116 000000 WDFR:   OPEN            ;WORDS FROM MEMORY PER ITERATION
  000120 000000 INTR:   OPEN            ;# OF INTERRUPTS PER ITERATION
  000122 000000 IDNUM:  0               ;MODULE IDENTIFICATION NUMBER=0
         000040         .REPT   SPSIZ   ;MODULE STACK STARTS HERE.
                        .NLIST
                        .WORD   0
                        .LIST
                        .ENDR
  000224        MODSP:
                ;*************************************************************
```

```
1                       ;
2                       ;CONSTANTS AND VARIABLES USED IN PROGRAM.
3                       ;
4                               .NLIST  BEX
5  000224 000000 ACNT:  OPEN            ;VARIABLES LOADED AT RUN-TIME.
6  000226 000000 BCNT:  OPEN
7  000230 000000 CHARA: OPEN
8  000232 000000 CHARB: OPEN
9  000234 000000 ACHAR: OPEN
10 00236 000000 BCHAR: OPEN
11 00240 000000 CCHAR: OPEN
12 00242 000000 DCHAR: OPEN
13 00244 000000 MAXY:  OPEN
14 00246 000000 MAXX:  OPEN
15 00250 000000 LX:    OPEN
16 00252 000000 RX:    OPEN
17 00254 000000 CHROW: OPEN
18 00256 000000 CHCNT: OPEN
19 00260 000014 M6X6:  14            ;CONSTANTS FOR 6X6 CHARACTER MATRIX.
20 00262 000013        13
21 00264 000350'       CHARA6
22 00266 000371'       CHARB6
23 00270 000412'       ACHAR6
24 00272 000420'       BCHAR6
25 00274 000426'       CCHAR6
26 00276 000434'       DCHAR6
27 00300 000057        57
28 00302 000120        120
29 00304 000047        47
30 00306 000050        50
31 00310 000006        6
32 00312 000005        5
33 00314 000020 M8X8:  20            ;CONSTANTS FOR 8X8 CHARACTER MATRIX.
34 00316 000017        17
35 00320 000442'       CHARA8
36 00322 000471'       CHARB8
37 00324 000520'       ACHAR8
38 00326 000530'       BCHAR8
39 00330 000540'       CCHAR8
40 00332 000550'       DCHAR8
41 00334 000045        45
42 00336 000100        100
43 00340 000037        37
44 00342 000040        40
45 00344 000010        8.
46 00346 000007        7
47 00350     000 CHARA6: .BYTE   0,0,0,0,0,0     ;6X6 CHARACTER DATA
48 00356     004         .BYTE   4,14,34,74,174,374
49 00364     374         .BYTE   374,374,374,374,374
50 00371     000 CHARB6: .BYTE   0,0,0,0,0
51 00376     200         .BYTE   200,300,340,360,370,374
52 00404     374         .BYTE   374,374,374,374,374
53                       .EVEN
54 00412     001 ACHAR6: .BYTE   1,2,3,4,5,6     ;6X6 CHARACTER TABLE.
55 00420     007 BCHAR6: .BYTE   7,10,11,12,13,13
56 00426     014 CCHAR6: .BYTE   14,15,16,17,20,21
57 00434     022 DCHAR6: .BYTE   22,23,24,25,26,26
```



```
58 00442     000 CHARA8: .BYTE   0,0,0,0,0,0,0,0 ;8X8 CHARACTER DATA
59 00452     001         .BYTE   1,3,7,17,37,77,177,377
60 00462     377         .BYTE   377,377,377,377,377,377,377
61 00471     000 CHARB8: .BYTE   0,0,0,0,0,0,0,0
62 00500     200         .BYTE   200,300,340,360,370,374,376,377
63 00510     377         .BYTE   377,377,377,377,377,377,377
64                       .EVEN
65 00520     001 ACHAR8: .BYTE   1,2,3,4,5,6,7,10        ;8X8 CHARACTER TABLE
66 00530     011 BCHAR8: .BYTE   11,12,13,14,15,16,17,17
67 00540     020 CCHAR8: .BYTE   20,21,22,23,24,25,26,27
68 00550     030 DCHAR8: .BYTE   30,31,32,33,34,35,36,36
69 00560     241 COL:    .BYTE   241,212,224,242,221,214
70                       .EVEN
71                       .LIST   BEX
72 00566 001702'MSG1:    MSG66
73 00570 177777          177777
74 00572 001725'MSG2:    MSG88
75 00574 177777          177777
76 00576 000000 ITCNT:   0
77 00600 000000 CSR:     OPEN            ;CONTROL & STATUS REGISTER ADDRESS
78 00602 000000 DBUF:    OPEN            ;DATA BUFFER REGISTER ADDRESS
79 00604 000000 CAR:     OPEN            ;CURSOR REGISTER ADDRESS
80 00606 000000 CHSR:    OPEN            ;CHARACTER STORE REGISTER ADDRESS
81 00610 000000 START1:  OPEN            ;STARTING COORDINATE FOR LEFT DIAGONAL
82 00612 000000 START2:  OPEN            ;       DITTO           FOR RIGHT DIAGONAL
83 00614 000000 XY1:     OPEN            ;RUNNING COORDINATE FOR LEFT DIAGONAL
84 00616 000000 XY2:     OPEN            ;       DITTO           FOR RIGHT DIAGONAL
```

```
1                       ;
2                       ;INITIALISE AND LOAD CHARACTER SET.
3                       ;
4  000620 016700 START: MOV     ADDR,R0          ;GET BASE ADDRESS
          177162
5  000624 010067        MOV     R0,CSR           ;LOAD UP TABLE; CONTROL & STATUS
          177750
6  000630 005720        TST     (R0)+
7  000632 010067        MOV     R0,DBUF          ;DATA BUFFER
          177744
8  000636 005720        TST     (R0)+            ;
9  000640 010067        MOV     R0,CAR           ;CURSOR REGISTER
          177740
10 00644  005720        TST     (R0)+            ;
11 00646  010067        MOV     R0,CHSR          ;CHARACTER STORE REGISTER
          177734
12 00652  016700        MOV     VECTOR,R0        ;NOW LOAD INTERRUPT VECTOR.
          177132
13 00656  012720        MOV     #RDYINT,(R0)+    ;SERVICE ROUTINE
          001164'
14 00662  005010        CLR     (R0)
15 00664  116710        MOVB    BR1,(R0)         ;AND PRIORITY.
          177122
16 00670  032767        BIT     #1,SR1           ;SEE WHICH MMATRIX REQUIRED.
          000001
          177120
17 00676  001417        BEQ     2$               ;BRANCH IF 6 X 6.
18 00700  005077        CLR     @CSR             ;SET UP FOR 8 X 8.
          177674
19 00704  012700        MOV     #ACNT,R0         ;GET START OF VARIABLE TABLE
          000224'
20 00710  012701        MOV     #M8X8,R1         ;GET START OF CONSTANT TABLE
          000314'
21 00714  012702        MOV     #14.,R2          ;COUNT OF TABLE ENTRIES
          000016
22 00720  012120 1$:    MOV     (R1)+,(R0)+      ;LOAD AN ENTRY
23 00722  005302        DEC     R2               ;COUNT.
24 00724  001375        BNE     1$               ;LOOP UNTIL TABLE FINISHED.
25 00726  104403        MSGN$,BEGIN,MSG2         ;ASCII MESSAGE CALL WITH COMMON HEADER
   00730  000000'
   00732  000572'
26 00734  000417        BR      4$               ;THEN EXIT.
27 00736  012777 2$:    MOV     #1400,@CSR       ;SET UP FOR 6 X 6
          001400
          177634
28 00744  012700        MOV     #ACNT,R0         ;GET START OF VARIABLE TABLE.
          000224'
29 00750  012701        MOV     #M6X6,R1         ;GET START OF CONSTANT TABLE.
          000260'
30 00754  012702        MOV     #14.,R2          ;COUNT OF TABLE ENTRIES.
          000016
31 00760  012120 3$:    MOV     (R1)+,(R0)+      ;LOAD AN ENTRY.
32 00762  005302        DEC     R2               ;COUNT.
33 00764  001375        BNE     3$               ;LOOP UNTIL TABLE FINISHED.
34 00766  104403        MSGN$,BEGIN,MSG1         ;ASCII MESSAGE CALL WITH COMMON HEADER
   00770  000000'
   00772  000566'
```



```
35 00774  005077 4$:    CLR     @CAR             ;ZERO CURSOR
          177604
36 01000  016700        MOV     CHSR,R0          ;GET HIGH BYTE ADDRESS
          177602
37 01004  105060        CLRB    1(R0)            ;CLR HIGH BYTE OF CHSR.
          000001
38                      ;
39                      ;LOAD ALL DUMMY CHARACTERS FIRST.
40                      ;
41 01010  012700 LDCH:  MOV     #128.,R0         ;NUMBER OF CHARACTERS
          000200
42 01014  016701 1$:    MOV     CHROW,R1         ;BYTES PER CHARACTER
          177234
43 01020  112777 2$:    MOVB    #252,@CHSR       ;LOAD A BYTE.
          000252
          177560
44 01026  005301        DEC     R1               ;CHARACTER DONE?
45 01030  001373        BNE     2$               ;BRANCH IF NOT
46 01032  005300        DEC     R0               ;ALL CHARACTERS DONE?
47 01034  001367        BNE     1$               ;BRANCH IF NOT.
48                      ;
49                      ;DUMMY CHARACTERS LOADED, NOW LOAD SPECIALS.
50                      ;
51 01036  016700        MOV     CHSR,R0          ;GET HIGH BYTE ADDRESS
          177544
52 01042  105060        CLRB    1(R0)            ;ZERO CHAR ADDRESS.
          000001
53 01046  016700        MOV     CHARA,R0         ;GET DATA POINTER.
          177156
54 01052  016702        MOV     ACNT,R2          ;NUMBER OF CHARACTERS.
          177146
55 01056  010001 3$:    MOV     R0,R1            ;COPY POINTER.
56 01060  016703        MOV     CHROW,R3         ;BYTES PER CHARACTER.
          177170
57 01064  112177 4$:    MOVB    (R1)+,@CHSR      ;LOAD A BYTE.
          177516
58 01070  005303        DEC     R3               ;CHARACTER DONE?
59 01072  001374        BNE     4$               ;BRANCH IF NOT.
60 01074  005200        INC     R0               ;UPDATE POINTER
61 01076  005302        DEC     R2               ;THIS LOT DONE?
62 01100  001366        BNE     3$               ;BRANCH IF NOT.
63 01102  016700        MOV     CHARB,R0         ;NEXT POINTER
          177124
64 01106  016702        MOV     BCNT,R2          ;NUMBER OF CHARACTERS.
          177114
65 01112  010001 5$:    MOV     R0,R1            ;COPY POINTER.
66 01114  016703        MOV     CHROW,R3         ;BYTES PER CHARACTER.
          177134
67 01120  112177 6$:    MOVB    (R1)+,@CHSR      ;LOAD A BYTE
          177462
68 01124  005303        DEC     R3               ;CHAR DONE?
69 01126  001374        BNE     6$
70 01130  005200        INC     R0               ;UPDATE POINTER.
71 01132  005302        DEC     R2               ;ALL DONE?
72 01134  001366        BNE     5$               ;BRANCH IF NOT
73 01136  112777        MOVB    #242,@DBUF       ;SET GREEN ON BLUE.
          000242
```

```
          177436
74 01144 112777          MOVB    #300,@DBUF      ;CLEAR BLINK CONTROL.
          000300
          177430
75 01152 052777          BIS     #110,@CSR       ;SET INT. ENAB. AND PRESET.
          000110
          177420
76 01160 104400          EXIT$,BEGIN     ;EXIT TO MONITOR, WAIT FOR INTERRUPT.
   01162 000000'
77 01164 042777 RDYINT:  BIC     #100,@CSR       ;CLEAR INT. ENABLE.
          000100
          177406
78                       ;*******************************************************
   01172 000004          PIRQ$,BEGIN,1$  ; QUEUE  CONTINUE AT 1$ AND RTI
   01174 000000'
   01176 001200'
                         ;*******************************************************
79 01200 032777 1$:      BIT     #BIT10,@CSR     ;CHECK FOR 525 LINES.
          002000
          177372
80 01206 001403          BEQ     RESTRT          ;IF NOT, OK
81 01210 162767          SUB     #10,MAXY        ;YES,REDUCE MAXIMUM Y COORD.
          000010
          177026
```



```
1 001216 052777 RESTRT:  BIS     #1,@CSR         ;TURN DISPLAY ON.
          000001
          177354
2 001224 012777          MOV     #TIMINT,@VECTOR ;SET FOR TIMER SERVICE
          001356'
          176556
3 001232 005067          CLR     ITCNT           ;RESET COLOUR COUNTER.
          177340
4 001236 026727 PASS:    CMP     ITCNT,#6        ;ALL COLOURS DONE YET?
          177334
          000006
5 001244 001005          BNE     1$              ;IF NOT, CARRY ON.
6 001246 042777          BIC     #40101,@CSR     ;DISPLAY OFF, DISABLE INTS.
          040101
          177324
7 001254 104413          ENDIT$,BEGIN    ;SIGNAL END OF ITERATION.
  001256 000000'
                                         ;MONITOR SHALL TEST END OF PASS
8 001260 016701 1$:      MOV     ITCNT,R1        ;GET COLOUR INDEX.
          177312
9 001264 116177          MOVB    COL(R1),@DBUF   ;SET COLOUR.
          000560'
          177310
10 01272 005267          INC     ITCNT           ;UPDATE COUNTER FOR NEXT TIME.
          177300
11 01276 116767          MOVB    LX,START1       ;SET 1ST X COORD (LEFT).
          176746
          177304
12 01304 116767          MOVB    MAXY,START1+1   ;SET 1ST Y COORD (LEFT).
          176734
          177277
13 01312 116767          MOVB    RX,START2       ;SET 1ST X COORD (RIGHT).
          176734
          177272
14 01320 116767          MOVB    MAXY,START2+1   ;SET 1ST Y COORD (RIGHT).
          176720
          177265
15 01326 005001 STROW:   CLR     R1              ;R1 IS NOW CHARACTER INDEX.
16 01330 016767 FIRST:   MOV     START1,XY1      ;SET RUNNING COORDINATES
          177254
          177256
17 01336 016767          MOV     START2,XY2      ;
          177250
          177252
18 01344 052777          BIS     #BIT14,@CSR     ;ENABLE TIMER
          040000
          177226
19 01352 104400          EXIT$,BEGIN     ;EXIT TO MONITOR, WAIT FOR INTERRUPT.
   01354 000000'
20 01356 042777 TIMINT:  BIC     #BIT15!BIT14,@CSR       ;DISABLE TIMER.
          140000
          177214
21                       ;*******************************************************
   01364 000004          PIRQ$,BEGIN,CONT        ; QUEUE  CONTINUE AT CONT AND RTI
   01366 000000'
   01370 001372'
                         ;*******************************************************
```

```
22 01372 126767 CONT:   CMPB    XY1,MAXX        ;AT MAXIMUM X ?
         177216
         176646
23 01400 103013         BHIS    11$
24 01402 016777         MOV     XY1,@CAR        ;SET FOR 1ST LEFT
         177206
         177174
25 01410 105777 1$:     TSTB    @CSR            ;CHECK READY
         177164
26 01414 100375         BPL     1$              ;WAIT IF NECESSARY.
27 01416 016702         MOV     ACHAR,R2        ;GET CHARACTER ADDRESS
         176612
28 01422 060102         ADD     R1,R2           ;ADD INDEX
29 01424 111277         MOVB    (R2),@DBUF      ;AND SEND IT.
         177152
30 01430 126767 11$:    CMPB    XY2,MAXX        ;AT MAXIMUM X ?
         177162
         176610
31 01436 103013         BHIS    12$
32 01440 016777         MOV     XY2,@CAR        ;MOVE CURSOR TO 1ST RIGHT.
         177152
         177136
33 01446 105777 2$:     TSTB    @CSR            ;CHECK READY
         177126
34 01452 100375         BPL     2$
35 01454 016702         MOV     CCHAR,R2        ;GET CHARACTER ADDRESS
         176560
36 01460 060102         ADD     R1,R2           ;ADD INDEX
37 01462 111277         MOVB    (R2),@DBUF      ;AND SEND IT.
         177114
38 01466 126767 12$:    CMPB    XY1,LX          ;IS X AT CENTRE?
         177122
         176554
39 01474 001452         BEQ     NXCHAR          ;IF SO, SKIP 2ND PAIR.
40 01476 105267         INCB    XY1             ;UPDATE COORDINATES
         177112
41 01502 105367         DECB    XY2             ;FOR 2ND PAIR.
         177110
42 01506 126767         CMPB    XY1,MAXX        ;AT MAXIMUM X ?
         177102
         176532
43 01514 103013         BHIS    13$
44 01516 016777         MOV     XY1,@CAR        ;SET CURSOP FOR LEFT.
         177072
         177060
45 01524 105777 3$:     TSTB    @CSR            ;CHECK FOR READY
         177050
46 01530 100375         BPL     3$
47 01532 016702         MOV     BCHAR,R2        ;GET CHARACTER ADDRESS
         176500
48 01536 060102         ADD     R1,R2           ;ADD INDEX
49 01540 111277         MOVB    (R2),@DBUF      ;AND SEND IT.
         177036
50 01544 126767 13$:    CMPB    XY2,MAXX        ;AT MAXIMUM X ?
         177046
         176474
51 01552 103013         BHIS    14$
```



```
52 01554 016777         MOV     XY2,@CAR        ;SET CURSOR FOR RIGHT.
         177036
         177022
53 01562 105777 4$:     TSTB    @CSR            ;CHECK FOR READY.
         177012
54 01566 100375         BPL     4$
55 01570 016702         MOV     DCHAR,R2        ;GET CHARACTER ADDRESS
         176446
56 01574 060102         ADD     R1,R2           ;ADD INDEX
57 01576 111277         MOVB    (R2),@DBUF      ;AND SEND IT.
         177000
58 01602 105767 14$:    TSTB    XY1+1           ;Y REACHED TOP?
         177007
59 01606 001405         BEQ     NXCHAR          ;IF SO, CHANGE CHARS.
60 01610 105367         DECB    XY1+1           ;NO, UPDATE COORDINATES
         177001
61 01614 105367         DECB    XY2+1           ;
         176777
62 01620 000664         BR      CONT            ;AND CONTINUE ROW.
63 01622 020167 NXCHAR: CMP     R1,CHCNT        ;ALL CHARACTERS DONE?
         176430
64 01626 001402         BEQ     NXTROW          ;IF SO, START NEW ROW.
65 01630 005201         INC     R1              ;NO, NEXT SET OF CHARS
66 01632 000636         BR      FIRST           ;GO BACK TO ROW START.
67 01634 126767 NXTROW: CMPB    START2,MAXX     ;RUN OUT OF X'S YET?
         176752
         176404
68 01642 001405         BEQ     NEXTY           ;IF SO, CHANGE Y STARTS.
69 01644 105367         DECB    START1          ;NO, CHANGE X STARTS.
         176740
70 01650 105267         INCB    START2
         176736
71 01654 000624         BR      STROW           ;START NEW ROW.
72 01656 105767 NEXTY:  TSTB    START1+1        ;Y START REACHED TOP?
         176727
73 01662 001002         BNE     1$
74 01664 000167         JMP     PASS            ;IF SO, GO TO PASS COUNT.
         177346
75 01670 105367 1$:     DECB    START1+1        ;NO, CHANGE Y STARTS
         176715
76 01674 105367         DECB    START2+1        ;
         176713
77 01700 000612         BR      STROW           ;START NEW ROW.
78                      .NLIST BEX
79 01702     115 MSG66: .ASCIZ  'MATRIX IS 6 BY 6.%'
80 01725     115 MSG88: .ASCIZ  'MATRIX IS 8 BY 8.%'
81                      .LIST BEX
82                      .EVEN
83       000001'        .END                    ;END OF PROGRAM.
```

```
ACHAR   000234R     ACHAR6  000412R     ACHAR8  000520R
ACNT    000224R     ACSR    000102R     ADDR    000006R
ADDR22= 001000      ASB     000106R     ASTAT   000104R
AWAS    000110R     BCHAR   000236R     BCHAR6  000420R
BCHAR8  000530R     BCNT    000226R     BEGIN   000000R
BIT0  = 000001      BIT1  = 000002      BIT10 = 002000
BIT11 = 004000      BIT12 = 010000      BIT13 = 020000
BIT14 = 040000      BIT15 = 100000      BIT2  = 000004
BIT3  = 000010      BIT4  = 000020      BIT5  = 000040
BIT6  = 000100      BIT7  = 000200      BIT8  = 000400
BIT9  = 001000      BREAK$= 104407      BR1     000012R
BR2     000013R     BTOD$ = 104421      CAR     000604R
CCHAR   000240R     CCHAR6  000426R     CCHAR8  000540R
CDATA$= 104412      CHAPA   000230R     CHARA6  000350R
CHARA8  000442R     CHAPB   000232R     CHARB6  000371R
CHARB8  000471R     CHCNT   000256R     CHROW   000254R
CHSR    000606R     COL     000560R     CONFIG  000056R
CONT    001372R     CSR     000600R     CSRA    000100R
DATCK$= 104411      DATER$= 104404      DBUF    000602R
DCHAR   000242R     DCHAR6  000434R     DCHAR8  000550R
DVID1   000014R     ENDIT$= 104413      END$  = 104410
ERRTYP  000106R     EXIT$ = 104400      FIRST   001330R
GETPA$= 104415      GWBUF$= 104414      HRDCNT  000044R
HRDER$= 104405      HRDPAS  000050R     ICONT   000036R
ICOUNT  000040R     IDNUM   000122R     INIT    000030R
INTR    000120R     ITCNT   000576R     LDCH    001010R
LX      000250R     MAP22$= 104416      MAXX    000246R
MAXY    000244R     MODNAM  000000R     MODSP   000224R
MSGN$ = 104403      MSGS$ = 104402      MSG$  = 104401
MSG1    000566R     MSG2    000572R     MSG66   001702R
MSG88   001725R     M6X6    000260R     M8X8    000314R
NEXTY   001656R     NULL  = 000000      NXCHAR  001622R
NXTROW  001634R     OPEN  = 000000      OTOA$ = 104420
PASCNT  000034R     PASS    001236R     PC    =%000007
PIRQ$ = 000004      POPSP = 005726      POPSP2= 022626
PRTY  = 000000      PRTY0 = 000000      PRTY1 = 000040
PRTY2 = 000100      PRTY3 = 000140      PRTY4 = 000200
PRTY5 = 000240      PRTY6 = 000300      PRTY7 = 000340
PS    = 177776      PSW   = 177776      PUSH  = 005746
PUSH2 = 024646      RAND$ = 104417      RANNUM  000054R
RDYINT  001164R     RESTRT  001216R     RES1    000056R
RES2    000060R     RSTRT   000112R     RX      000252R
R0    =%000000      R1    =%000001      R2    =%000002
R3    =%000003      R4    =%000004      R5    =%000005
R6    =%000006      R7    =%000007      SRADR   000102R
SOFCNT  000042R     SOFER$= 104406      SOFPAS  000046R
SP    =%000006      SPOINT  000032R     SFSIZ = 000040
SR1     000016R     SR2     000020R     SR3     000022R
SR4     000024R     START   000620R     START1  000610R
START2  000612R     STAT    000026R     STROW   001326R
SVR0    000062R     SVR1    000064R     SVR2    000066R
SVR3    000070R     SVR4    000072R     SVR5    000074R
SVR6    000076R     SYSCNT  000052R     TIMINT  001356R
TRPDFD= 000022      VECTOR  000100R     WASADR  001104R
WDFR    000116R     WDTO    000114R     XFLAG   000005R
XY1     000614R     XY2     000616R
. ABS.  000000      000
```



```
        001750      001
ERRORS DETECTED: 0
FREE CORE: 11488. WORDS

XVTVA0.OBJ,XVTVA0.SEQ=[40,0]NEWDDX.P11,[40,30]XVTVA0.P11
```