TECHNICAL MEMORANDUM (NASA) 83

ENHANCED CHARACTER SIZES FOR THE VDM-1 VIDEO DISPLAY BOARD

Described is a software program to provide increased character size on a 3.75-inch diagonal CRT display used in the Loran-C receiver designed by the Avionics Engineering Center at Ohio University.

by

Stanley M. Novacki, III
James D. Nickum

Avionics Engineering Center
Department of Electrical Engineering
Ohio University
Athens, Ohio 45701

December 1982

Prepared for

NASA Langley Research Center
Hampton, Virginia 23665

(Contract NGR 36-009-017)

TABLE OF CONTENTS

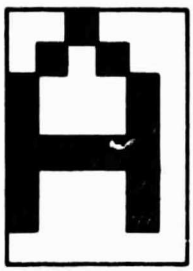
I. INTRODUCTION

This technical memorandum will address a display enhancement implemented on the new VDM-1 video display board that is currently installed in the Ohio University Loran-C navigation receiver. This enhancement provides increased character size for easier viewing of the 3.75-inch diagonal CRT display currently being used with the Loran-C receiver. The enhancement is purely software once the display is placed in the low resolution graphics mode. Additionally, certain functions are capable, such as displaying a course deviation indicator (CDI) at the bottom of the display.

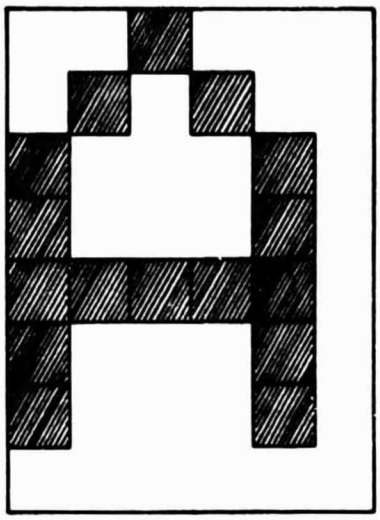
II. APPROACH

It was decided to investigate the success of artificially drawing the alpha-numeric symbols in a size more easily recognizable on the three-inch CRT at a typical viewing distance in a general aviation cockpit.

The 6847 video display driver incorporated in the Ohio University Loran-C receiver has an on-chip character generator to produce standard ASCII characters in the devices character mode [1]. Characters from this generator produce 5x7 dot cells. For purposes of experimentation and especially for ease of execution, a 6x8 dot cell was chosen to be evaluated. A sample character in the 6x8 cell is shown in Figure 1. Several of the characters using this new font aspect ratio were evaluated by storing the six 8-bit bytes that represent the dot pattern of the character in the VDM-1 video memory locations. It was determined that the increase of character size of 37% produced no appreciable readability when viewed from a distance of 3 feet. The next logical step was to produce a character font that was significantly larger than the 6x8 cell font size. To make the characters twice as large in both directions with the current memory write technique, requires 32 load and store instructions and 3K bytes of memory alone to store the font definitions, which is far too awkward. A simple solution was found on examination of the 6847 data sheet. The 6847 video driver chip could be placed in a course graphics mode of 128x96 pixels per screen. In this mode writing one pixel will write an equivalent 2x2 pixel in the 256x192 high resolution graphics mode. In this manner storing a 6x8 cell font will produce a 12x16 dot graphics character. In the configuration that exists on the VDM-1 board, mode initialization switch settings are changed to set up the 128x96 course graphics mode. This character size is a 450% increase in size over the original 5x7 dot font of the on board character generator. The readability of these characters is excellent and should provide a better information display in evaluating the Ohio University-developed Loran-C receiver.



Stored 6x8 cell



Enhanced 12x16 cell size
as displayed.

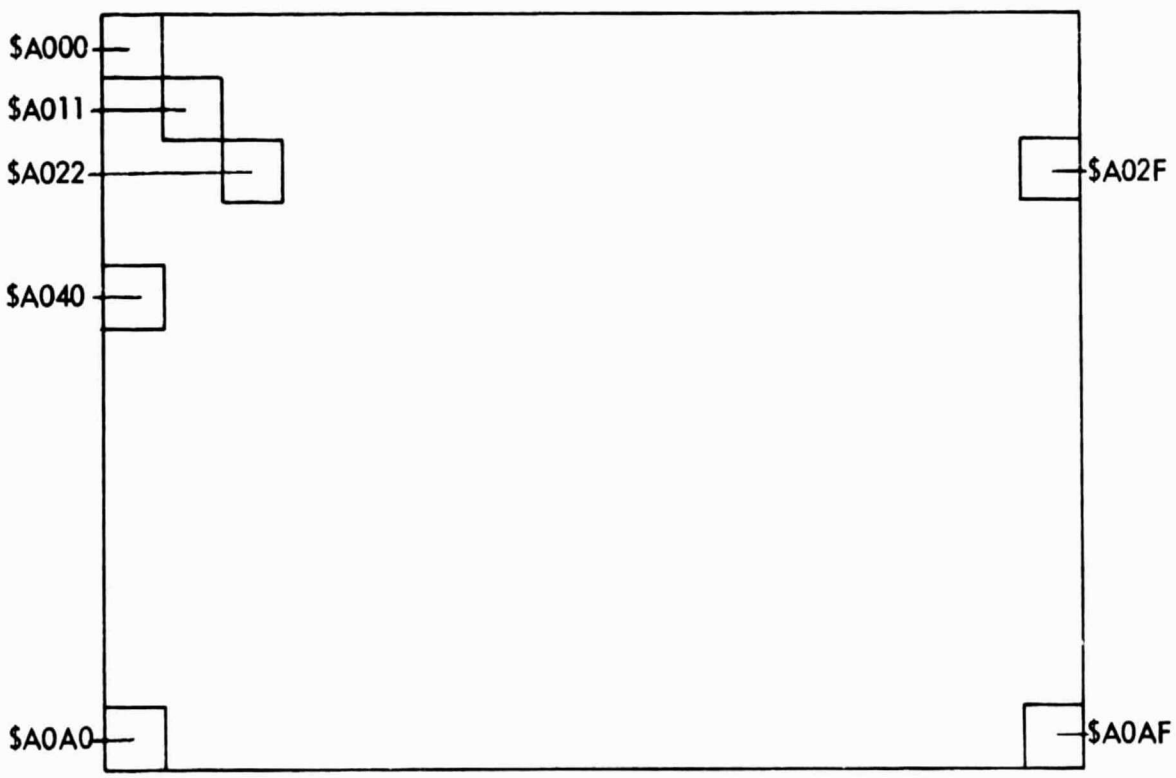


Figure 1. Character Font and Display Memory Map.

III. SOFTWARE

In the 128x96 dot graphics mode the video screen consists of 1536 memory locations. Each row consists of 16 8-bit wide slices. A "1" or "0" turns each bit in a character cell on or off respectively. Each row of bits begins with a hexidecimal address on a 16-byte boundary, i.e., the first row might be $A000, the next $A010, the next $A020 and so on. To draw a character, the software routines use a two-byte address on page zero to point at the top row of the next printing location; this is conventionally called the cursor. A blank line is stored at the address contained in cursor. The cursor is incremented by $10 to point at the row beneath the one most recently written. The first byte of the character is retrieved from the font bit table and stored at the second row in the character. The cursor is advanced by $10 to point at the next row of the character font, the byte from the font table is fetched and stored and then the process repeats until all 8 of the bytes that describe the character are written at the location desired on the VDM-1 display board. In this manner, any of the printing characters in the font table can be placed in the video display field.

The font table contains all 96 of the ASCII printable characters, the remaining 32 nonprinting characters are trapped out so that nothing is printed on the screen. The non-printing characters have hexidecimal values $00-1F. By subtracting $20 from the ASCII value, the nonprinting characters can be trapped out and not printed. The remaining printable characters range from $20-$7F. Once a valid ASCII printable character is found in this range it is adjusted to the range $00-$5F which corresponds to the values in the offset address table with the same index value. This then provides the offset value to find the 8 bytes in the font table to draw the character.

The font table and offset tables for this version of the software only support the first 64 characters of the 96 printable characters. This includes the numbers symbols and upper case characters in the ASCII character set. The lower case letters are mapped into the upper case reference addresses for offset and font table look up. The addition of 256 more bytes of font storage would allow lower case letters to be displayed.

IV. IMPLEMENTATION

The software implementation of this display format is accomplished using the software as listed in the appendix. Essentially, the software consists of three subroutines, two tables and a main program. The entire program as listed is intended as a subroutine with input conditions of (x) = column position on display, (y) = row position on display and (A) = ASCII character value.

The main segment of the program initializes variables and pointers and calls the subroutines to locate the character memory cell and write the character.
There are three subroutines that do the following:

Subroutine CLEAR will clear the entire screen of all characters.

Subroutine SETCSR will, based on the values in the X and Y registers, determine the cursor position in the VDM-1 display memory space to write the character. The value of X can range from $00-$0F, and the value of Y from $00-$0C.

Subroutine DRWCHR will, based on the value in the A accumulator, fetch the proper offset address value from the offset address table and then fetch and write to the screen memory the image of the values stored in the font table.

The two tables consist of the offset address table which provides the indexing based on the ASCII character value into the font table. Table two is the font table which contains the bit pattern in byte-wise form for output to the display screen.

For future implementation, the offset address table can be eliminated by a fixed multiply routine such that the index into the font table can be seen to be an integermultiple of 8. For example, if the 6th printable ASCII character is required, the offset necessary to fetch the character from the font table is 6x8, which is 48 or $30. This is the correct entry as shown in the offset address table in the appendix. This is expected to be implemented in the next revision of this software.

V. SUMMARY

Presented here is a relatively simple method of implementing a more readable display for output of Loran-C navigation information to the pilot of the aircraft. The software takes advantage of hardware bit replication for producing the enhanced character size required in this application without the need for excessive memory allocation for font data storage.

VI. REFERENCE

[1] VDM-1 Video Display Board Instruction Manual, Microcomputer Products Inc.

```
 END PASS 1:   0 ERRORS

    1
*****************************************************************************
    2                   *
    3                   *  THIS SOFTWARE WILL PRODUCE ENHANCED ALPHA-NUMERICS FOR
THE
    4                   *  VDM-1 VIDEO DISPLAY CARD INCLUDED IN THE LORAN C
RECEIVER
    5                   *  DESIGNED BY THE AVIONICS ENGINEERING CENTER AT OHIO
    6                   *  UNIVERSITY.  THE DISPLAY CONSISTS OF 12 ROWS OF 16
CHARACTERS.
    7                   *  THE DISPLAY CELL SIZE IS 12 X 16 BITS ASSUMING THE
HIGH RES
    8                   *  DISPLAY MODE OF 256 X 192.  THE VDM-1 IS ASSUMED TO BE
IN THE
    9                   *  128 X 96 LOW RESOLUTION MODE WHEN EXECUTING THIS
SOFTWARE.
   10                   *
   11
*****************************************************************************
   12                   *
   13                   *
   14  0000                     ORG 0
   15  0000 00 00       BITPTR BSS 2
   16  0002 00 00       CURSOR BSS 2
   17  0004 00          FONTST BSS 1
   18  0005             BITS   EQU $1200       ;OFFSET ADDRESS TABLE START
   19                   *
   20                   *    MAIN PROGRAM START
   21                   *
   22  1000                     ORG $1000       ;START PROGRAM ADDRESS
   23  1000 20 11 10            JSR CLEAR       ;CLEAR SCREEN
   24  1003 A9 13               LDA =$13
   25  1005 85 04               STA FONTST      ;SET FONT PAGE #
   26  1007 00          INCHR   BRK             ;SET CHAR TO BE PLOTTED AND (X,Y)
   27  1008 20 45 10            JSR SETCSR      ;READY INDIRECT ADDRESSES
   28  1008 20 2A 10            JSR DRWCHR      ;DRAW THE CHARACTER!!!
   29  100E 4C 07 10            JMP INCHR       ;DO IT AGAIN..
   30                   *
   31                   *
   32
*****************************************************************************
   33                   *
   34                   *    CLEAR SCREEN SUBROUTINE
   35                   *
   36                   *    ENTRY     NONE
   37                   *
   38                   *    EXIT NONE
   39                   *
   40                   *    USES A,Y
   41                   *
   42
*****************************************************************************
   43                   *
   44                   *
   45  1011             CLEAR  EQU *
   46  1011 A9 A0               LDA =$A0
   47  1013 85 03               STA CURSOR+1    ;HIGH BYTE CURSOR POSITION
   48  1015 A9 00       LOOP1   LDA =0
   49  1017 85 02               STA CURSOR      ;LOW BYTE OF CURSOR POSITION
   50  1019 A8                  TAY             ;SET INDEX ADDRESS
   51  101A A9 FF               LDA =$FF        ;BLANKING FILL CHR..
   52  101C 91 02       LOOP2   STA (CURSOR),Y
   53  101E C8                  INY             BUMP ;INDEX ADDRESS
   54  101F D0 FB               BNE LOOP2       ;LOOP TILL Y=0
   55  1021 E6 03               INC CURSOR+1    ;INCREMENT CURSOR OVER PAGE BOUND
   56  1023 A5 03               LDA CURSOR+1
   57  1025 C9 A8               CMP =$A8        ;TEST IF AT BOT OF DISP SCREEN
   58  1027 D0 EC               BNE LOOP1
   59  1029 60                  RTS
   60                   *
```

```
                        *
                        *
*****************************************************************************
                        *
                        *    DRAW CHARACTER ROUTINE
                        *
                        *    ENTRY      A = ASCII VALUE OF CHARACTER
                        *
                        *          (CURSOR) = 16 BIT ADDRESS OF CURSOR POSITION
                        *
                        *    EXIT       NONE
                        *
                        *    USES       X,Y,A
                        *
*****************************************************************************
                        *
                        *
102A                    DRWCHR EQU *
102A A2 08                     LDX =8          ;READY FONT BYTE COUNTER
102C A0 00                     LDY =0          ;READY ZERO-OFFSET
                        *
102E B1 00              PLOT   LDA (BITPTR),Y  ;GET BIT PATTERN
1030 49 FF                     EOR =$FF        ;LET'S GO INVERSE VIDEO
1032 91 02                     STA (CURSOR),Y  ;DRAW IT AT (X,Y) ON SCREEN
1034 A5 02                     LDA CURSOR
1036 18                        CLC
1037 69 10                     ADC =$10        ;FORM NEXT DRAWING ADDRESS
1039 85 02                     STA CURSOR
103B 90 02                     BCC NOFIX
103D E6 03                     INC CURSOR+1    ;ADJUST PAGE-NUMBER IF NEEDED
103F                    NOFIX  EQU *
103F E6 00                     INC BITPTR      ;POINT TO NEXT SET OF BITS
1041 CA                        DEX             ;COUNT BITS JUST PLOTTED
1042 D0 EA                     BNE PLOT        ;IF NOT DONE, PLOT REST OF CHAR
1044 60                 DONE   RTS
                        *
                        *
*****************************************************************************
                        *
                        *    SET CURSOR ROUTINE
                        *
                        *     ENTRY      X = COLUMN FOR CURSOR POSITION
                        *                Y = ROW FOR CURSOR POSITION
                        *                A = ASCII VALUE OF CHAR TO BE WRITTEN
                        *
                        *
                        *     EXIT       (CURSOR) = ADDRESS OF CURSOR LOCATION
                        *
                        *    USES        NONE
                        *
*****************************************************************************
                        *
                        *
                        *
1045                    SETCSR EQU *
1045 48                        PHA             ;SAVE CHR ON STACK
1046 A9 A0                     LDA =$A0
1048 85 03                     STA CURSOR+1
104A A9 00                     LDA =0
104C 85 02                     STA CURSOR      ;INIT CURSOR TO (0,0) ON SCREEN
104E C0 00                     CPY =0
1050 F0 0A                     BEQ ROWSET      ;NO NEED TO ADJUST CURSOR FOR
ROW..
1052 18                 LOCROW CLC             ;COMPUTE NEW ADDRESS FROM X,Y
COORDS
1053 69 80                     ADC =$80        ;MOVE TO NEXT ROW
1055 90 02                     BCC DECNT       ;NO NEED TO FIX PAGE OF CURSOR
1057 E6 03                     INC CURSOR+1    ;DO SO OTHERWISE
1059 88                 DECNT  DEY
```

```
   127  105A D0 F6           BNE LOCROW      ;LOOP UNTIL AT ROW (Y)
   128  105C 85 02    ROWSET STA CURSOR
   129  105E 8A              TXA             ;GET X-COORD TO ADD TO CURSOR
   130  105F 29 0F           AND =$0F        ;NO COL NUMBERS > 15!
   131  1061 18              CLC
   132  1062 65 02           ADC CURSOR      ;ADD IN COLUMN NUMBER
   133  1064 85 02           STA CURSOR      ;FORM ABSOULUTE CURSOR ADDRESS
   134                  *
   135                  *    CURSOR READY FOR PLOTTING CHARACTER..
   136                  *
   137  1066 68              PLA             ;RECOVER CHARACTER TO COMPUTE BIT
TABLE ADD.
   138  1067 48              PHA             ;SAVE IT FOR PLOTTING
   139  1068 18              CLC
   140  1069 E9 20           SBC =$20        ;DISALLOW CTRL CHRS; ALSO SAVES
SPACE!!
   141  106B AA              TAX             ;USE CHAR AS OFFSET INTO AD RESS
TABLE
   142  106C BD 00 12        LDA BITS,X      ;GET LOW-ORDER BYTE OF FONT ENTRY
   143  106F 85 00           STA BITPTR
   144  1071 68              PLA             ;RECOVER TO COMPUTE WHICH BIT
TABLE
   145  1072 29 40           AND =$40        ;SAVE BIT 6 FOR TABLE
DETERMINATION
   146  1074 18              CLC
   147  1075 2A              ROL A
   148  1076 2A              ROL A
   149  1077 2A              ROL A
   150  1078 65 04           ADC FONTST      ;COMPUTE TABLE PAGE #, CARRY CLEAR
FROM ROL'S
   151  107A 85 01           STA BITPTR+1    ;FORM ABSOLUTE FONT ENTRY POINTER
   152  107C 60              RTS
   153                  *
   154                  *
   155
*****************************************************************************
   156                  *
   157                  *    OFFSET ADDRESS TABLE FOR NUMBERS AND SYMBOLS
   158                  *
   159
*****************************************************************************
   160                  *
   161                  *
   162  1200                 ORG $1200
   163  1200 00              HEX 00
   164  1201 08              HEX 08
   165  1202 10              HEX 10
   166  1203 18              HEX 18
   167  1204 20              HEX 20
   168  1205 28              HEX 28
   169  1206 30              HEX 30
   170  1207 38              HEX 38
   171  1208 40              HEX 40
   172  1209 48              HEX 48
   173  120A 50              HEX 50
   174  120B 58              HEX 58
   175  120C 60              HEX 60
   176  120D 68              HEX 68
   177  120E 70              HEX 70
   178  120F 78              HEX 78
   179  1210 80              HEX 80
   180  1211 88              HEX 88
   181  1212 90              HEX 90
   182  1213 98              HEX 98
   183  1214 A0              HEX A0
   184  1215 A8              HEX A8
   185  1216 B0              HEX B0
   186  1217 B8              HEX B8
   187  1218 C0              HEX C0
   188  1219 C8              HEX C8
   189  121A D0              HEX D0
   190  121B D8              HEX D8
```

```
191  121C E0                HEX E0
192  121D E8                HEX E8
193  121E F0                HEX F0
194  121F F8                HEX F8
195                  *
196                  *
197
*****************************************************************************
198                  *
199                  *     OFFSET ADDRESS TABLE FOR UPPER CASE CHARACTERS
200                  *
201
*****************************************************************************
202                  *
203                  *
204  1220 00                HEX 00
205  1221 08                HEX 08
206  1222 10                HEX 10
207  1223 18                HEX 18
208  1224 20                HEX 20
209  1225 28                HEX 28
210  1226 30                HEX 30
211  1227 38                HEX 38
212  1228 40                HEX 40
213  1229 48                HEX 48
214  122A 50                HEX 50
215  122B 58                HEX 58
216  122C 60                HEX 60
217  122D 68                HEX 68
218  122E 70                HEX 70
219  122F 78                HEX 78
220  1230 80                HEX 80
221  1231 88                HEX 88
222  1232 90                HEX 90
223  1233 98                HEX 98
224  1234 A0                HEX A0
225  1235 A8                HEX A8
226  1236 B0                HEX B0
227  1237 B8                HEX B8
228  1238 C0                HEX C0
229  1239 C8                HEX C8
230  123A D0                HEX D0
231  123B D8                HEX D8
232  123C E0                HEX E0
233  123D E8                HEX E8
234  123E F0                HEX F0
235  123F F8                HEX F8
236                  *
237                  *
238
*****************************************************************************
239                  *
240                  *     CHARACTER FONT TABLE FOR NUMBERS AND SYMBOLS
241                  *
242
*****************************************************************************
243                  *
244                  *
245  1300                   ORG $1300
246  1300 00 00 00  SP      HEX 00,00,00,00,00,00,00,00
          00 00 00
          00 00
247  1308 20 20 20  XCL     HEX 20,20,20,20,00,20,20,00
          20 00 20
          20 00
248  1310 50 50 50  DQOU    HEX 50,50,50,00,00,00,00,00
          00 00 00
          00 00
249  1318 50 50 F8  LBS     HEX 50,50,F8,50,F8,50,50,00
          50 F8 50
          50 00
250  1320 20 78 A0  DOLR    HEX 20,78,A0,70,28,F0,20,00
```

```
                70 28 F0
                20 00
251   1326  C0 C8 10  PRONT   HEX C0,C8,10,20,40,98,18,00
                20 40 98
                18 00
252   1330  40 A0 A0  ANDSN   HEX 40,A0,A0,40,A8,90,68,00
                40 A8 90
                68 00
253   1338  20 20 20  APOS    HEX 20,20,20,00,00,00,00,00
                00 00 00
                00 00
254   1340  20 40 80  LPAREN  HEX 20,40,80,80,80,40,20,00
                80 80 40
                20 00
255   1348  20 10 08  RPAREN  HEX 20,10,08,08,08,10,20,00
                08 08 10
                20 00
256   1350  20 A8 70  ASTRSK  HEX 20,A8,70,20,70,A8,20,00
                20 70 A8
                20 00
257   1358  00 20 20  PLUSSN  HEX 00,20,20,78,20,20,00,00
                78 20 20
                00 00
258   1360  00 00 00  COMMA   HEX 00,00,00,00,20,20,40,00
                00 20 20
                40 00
259   1368  00 00 00  PERIOD  HEX 00,00,00,00,20,20,00,00
                00 20 20
                00 00
260   1370  00 08 10  OSLASH  HEX 00,08,10,20,40,80,00,00
                20 40 80
                00 00
261   1378  70 88 98  ZERO    HEX 70,88,98,A8,C8,88,70,00
                A8 C8 88
                70 00
262   1380  20 60 20  ONE     HEX 20,60,20,20,20,20,70,00
                20 20 20
                70 00
263   1388  70 88 08  TWO     HEX 70,88,08,70,80,80,F8,00
                70 80 80
                F8 00
264   1390  F8 08 10  THREE   HEX F8,08,10,30,08,88,70,00
                30 08 88
                70 00
265   1398  10 30 50  FOUR    HEX 10,30,50,F8,10,10,10,00
                F8 10 10
                10 00
266   13A0  F8 80 F0  FIVE    HEX F8,80,F0,08,08,88,70,00
                08 08 88
                70 00
267   13A8  38 40 80  SIX     HEX 38,40,80,F0,88,88,70,00
                F0 88 88
                70 00
268   13B0  F8 08 08  SEVEN   HEX F8,08,08,10,20,40,80,00
                10 20 40
                80 00
269   13B8  70 88 88  EIGHT   HEX 70,88,88,70,88,88,70,00
                70 88 88
                70 00
270   13C0  70 88 88  NINE    HEX 70,88,88,78,08,10,E0,00
                78 08 10
                E0 00
271   13C8  00 00 20  COLON   HEX 00,00,20,00,20,00,00,00
                00 20 00
                00 00
272   13D0  00 00 20  SEM     HEX 00,00,20,00,20,20,40,00
                00 20 20
                40 00
273   13D8  10 20 40  LTHAN   HEX 10,20,40,80,40,20,10,00
                80 40 20
                10 00
274   13E0  00 00 F8  EQALS   HEX 00,00,F8,00,F8,00,00,00
```

```
                00 F8 00
                00 00
  275  13E8 40 20 10  GTHAN  HEX 40,20,10,08,10,20,40,00
                08 10 20
                40 00
  276  13F0 70 88 08  QUESTN HEX 70,88,08,30,20,00,20,00
                30 20 00
                20 00
  277  13F8 70 88 A8  ATSIGN HEX 70,88,A8,B8,B0,80,78,00
                B8 B0 80
                78 00
  278                    *
  279                    *
  280
*********************************************************************************
  281                    *
  282                    *     CHARACTER FONT TABLE FOR UPPER CASE LETTERS
  283                    *
  284
*********************************************************************************
  285                    *
  286                    *
  287  1400 20 50 88  CAPA   HEX 20,50,88,88,F8,88,88,00
                88 F8 88
                88 00
  288  1408 F0 88 88  CAPB   HEX F0,88,88,F0,88,88,F0,00
                F0 88 88
                F0 00
  289  1410 70 88 80  CAPC   HEX 70,88,80,80,80,88,70,00
                80 80 88
                70 00
  290  1418 F0 88 88  CAPD   HEX F0,88,88,88,88,88,F0,00
                88 88 88
                F0 00
  291  1420 F8 80 80  CAPE   HEX F8,80,80,F0,80,80,F8,00
                F0 80 8C
                F8 00
  292  1428 F8 80 80  CAPF   HEX F8,80,80,F0,80,80,80,00
                F0 80 80
                80 00
  293  1430 70 88 80  CAPG   HEX 70,88,80,80,90,88,78,00
                80 90 88
                78 00
  294  1438 88 88 88  CAPH   HEX 88,88,88,F8,88,88,88,00
                F8 88 88
                88 00
  295  1440 70 20 20  CAPI   HEX 70,20,20,20,20,20,70,00
                20 20 20
                70 00
  296  1448 08 08 08  CAPJ   HEX 08,08,08,08,08,88,70,00
                08 08 88
                70 00
  297  1450 88 90 A0  CAPK   HEX 88,90,A0,C0,A0,90,88,00
                C0 A0 90
                88 00
  298  1458 80 80 80  CAPL   HEX 80,80,80,80,80,80,F8,00
                80 80 80
                F8 00
  299  1460 88 B8 A8  CAPM   HEX 88,B8,A8,A8,A8,88,88,00
                A8 A8 88
                88 00
  300  1468 88 80 C8  CAPN   HEX 88,80,C8,A8,98,88,88,00
                A8 98 88
                88 00
  301  1470 70 88 88  CAPO   HEX 70,88,88,88,88,88,70,00
                88 88 88
                70 00
  302  1478 F0 88 F8  CAPP   HEX F0,88,F8,F0,80,80,80,00
                F0 80 80
                80 00
  303  1480 70 88 88  CAPQ   HEX 70,88,88,88,A8,90,68,00
                88 A8 90
```

```
                68 00
  304   1488 F0 88 88  CAPR   HEX F0,88,88,F0,A0,90,88,00
             F0 A0 90
             88 00
  305   1490 70 88 80  CAPS   HEX 70,88,80,70,08,88,70,00
             70 08 88
             70 00
  306   1498 F8 A8 20  CAPT   HEX F8,A8,20,20,20,20,20,00
             20 20 20
             20 00
  307   14A0 88 88 88  CAPU   HEX 88,88,88,88,88,88,70,00
             88 88 88
             70 00
  308   14A8 88 88 88  CAPV   HEX 88,88,88,A0,A0,20,20,00
             A0 A0 20
             20 00
  309   14B0 88 88 88  CAPW   HEX 88,88,88,A8,A8,A8,50,00
             A8 A8 A8
             50 00
  310   14B8 88 88 50  CAPX   HEX 88,88,50,20,50,88,88,00
             20 50 88
             88 00
  311   14C0 88 88 50  CAPY   HEX 88,88,50,20,20,20,20,00
             20 20 20
             20 00
  312   14C8 F8 08 10  CAPZ   HEX F8,08,10,20,40,80,F8,00
             20 40 80
             F8 00
  313   14D0 F8 C0 C0  LBRKT  HEX F8,C0,C0,C0,C0,C0,F8,00
             C0 C0 C0
             F8 00
  314   14D8 00 80 40  LSLSH  HEX 00,80,40,20,10,08,00,00
             20 10 08
             00 00
  315   14E0 F8 18 18  RBRKT  HEX F8,18,18,18,18,18,F8,00
             18 18 18
             F8 00
  316   14E8 20 50 81  CARET  HEX 20,50,81,00,00,00,00,00
             00 00 00
             00 00
  317   14F0 00 00 00  USCORE HEX 00,00,00,00,00,00,F8,00
             00 00 00
             F8 00
  318                          END

END PASS 2:   0 ERRORS

1 SYMB  ADDR  DEF   REFERENCES

ANDSN   1330   252
APOS    1338   253
ASTRSK  1350   256
ATSIGN  13F8   277
BITPTR  0000    15      91   143   151    81
BITS    1200    18     142
CAPA    1400   287
CAPB    1408   288
CAPC    1410   289
CAPD    1418   290
CAPE    1420   291
CAPF    1428   292
CAPG    1430   293
CAPH    1438   294
CAPI    1440   295
CAPJ    1448   296
CAPK    1450   297
CAPL    1458   298
CAPM    1460   299
CAPN    1468   300
CAPO    1470   301
CAPP    1478   302
CAPQ    1480   303
```

CAPR	1488	304									
CAPS	1490	305									
CAPT	1498	306									
CAPU	14A0	307									
CAPV	14A8	308									
CAPW	14B0	309									
CAPX	14B8	310									
CAPY	14C0	311									
CAPZ	14C8	312									
CARET	14E8	316									
CLEAR	1011	45	23								
COLON	13C8	271									
COMMA	1360	258									
CURSOR	0002	16	47	49	55	56	84	87	89	117	119	125
			128	132	133	52	83					
DECNT	1059	126	124									
DOLR	1320	250										
DONE	1044	94										
DQOU	1310	248										
DRWCHR	102A	77	28									
EIGHT	13B8	269										
EQALS	13E0	274										
FIVE	13A0	266										
FONTST	0004	17	25	150								
FOUR	1398	265										
GTHAN	13E8	275										
INCHR	1007	26	29									
LBRKT	14D0	313										
LBS	1318	249										
LOCROW	1052	122	127									
LOOP1	1015	48	58									
LOOP2	101C	52	54									
LPAREN	1340	254										
LSLSH	14D8	314										
LTHAN	13D8	273										
NINE	13C0	270										
NOFIX	103F	90	88									
ONE	1380	262										
OSLASH	1370	260										
PERIOD	1368	259										
PLOT	102E	81	93									
PLUSSN	1358	257										
PRCNT	1328	251										
QUESTN	13F0	276										
RBRKT	14E0	315										
ROWSET	105C	128	121									
RPAREN	1348	255										
SEM	1300	272										
SETCSR	1045	114	27									
SEVEN	13B0	268										
SIX	13A8	267										
SP	1300	246										
THREE	1390	264										
TWO	1388	263										
USCORE	14F0	317										
XCL	1308	247										
ZERO	1378	261										