

```

%-----
% COMMAND_TERMINAL-3
% VER 1.1.0
%
% SERIAL TERMINAL PROGRAM FOR PERSEUS-9 (CPU:R6502A)
%
% HAND ASSEMBLED
%
% MITSURU YAMADA 24/JUN/2022
%
%-----
% COPYRIGHT (C) 2022 MITSURU YAMADA. ALL RIGHTS RESERVED.
%
%-----
% ADDRESS MAPPING:
%
% 40 X 7 CHARACTER DISPLAY BUFFER:      (ADDRESS $0200 - $0317)
%
% LEFT                                     RIGHT
% $0200 $0201, , , , , , , , , , , , $0226 $0227     TOP
% $0228 $0229, , , , , , , , , , , , $024E $024F
% $0250 $0251, , , , , , , , , , , , $0276 $0277
% $0278 $0279, , , , , , , , , , , , $029E $029F
% $02A0 $02A1, , , , , , , , , , , , $02C6 $02C7
% $02C8 $02C9, , , , , , , , , , , , $02EE $02EF
% $02F0 $02F1, , , , , , , , , , , , $0316 $0317     BOTTOM
%
% CHARACTER FONT TABLE:                  (ADDRESS $FD00 - $FEFF)
%
% KEY SCAN ASCII CODE TABLE:            (ADDRESS $FF00 - $FF4F)
%-----
% VARIABLES
%
% SYMBOL                                     DATA
% BP_0                                     $0000     BUFFER INPUT POINTER (16 BIT)
% BP_1                                     $0002     BUFFER DISPLAY POINTER (16 BIT)
% BP_2                                     $0004     BUFFER END POINTER (16BIT)
% BP_3                                     $0006     BUFFER LEFT END POINTER (16BIT)
% BP_4                                     $0008     BUFFER HEAD POINTER (16BIT)
% FP                                       $000A     CHARACTER FONT POINTER (16 BIT)
% TP_0                                     $000C     SCAN TABLE POINTER (16 BIT)
% SCAN_PAT                                 $000E     SCAN PATTERN
% SCAN_PAT_2                               $000F     SCAN PATTERN NUMBER
% PUSH_FLAG_1                              $0010     KEY PRESSED FLAG
% PUSH_FLAG_2                              $0011     KEY PRESSED LAST TIME FLAG
% SHIFT_FLAG                               $0012     SHIFT KEY PRESSED FLAG
% REG_0                                     $0013     FONT DATA SAVE MEMORY
% REG_1                                     $0014     CONTROL DATA SAVE MEMORY
% REG_3                                     $0015     LOOP COUNTER OF INIT DISP, SAVE MEMORY
% TEMP_0                                    $0016     INPUT PARAMETER OUT FONT, SAVE MEMORY
% TEMP_1                                    $0017     SCAN PATTERN NUMBER SAVE MEMORY
% TEMP_2                                    $0018     LOOP COUNTER OF OUT FONT, SAVE MEMORY
% B_COUNT                                  $0019     CURSOR BLINK COUNTER
%
%-----
%
% PORT                                     $4000     PARALLEL INTERFACE ADDRESS (R/W)
% ACIA_STATUS                             $8000     ACIA STATUS REGISTER
% ACIA_DATA                                $8001     ACIA DATA REGISTER
%-----
%
%                                     ADDRESS(HEX) DATA(HEX)
%-----
% PROGRAM START ADDRESS   CT_3_MAIN_1   $F800
%
% RESET VECTOR
% .ORIGIN $FFFC
%                                     FFFC  00 F8
%-----
% SERIAL TERMINAL MAIN PROGRAM
%-----
%
% .ORIGIN $FC00
%
% CT_3_MAIN   LDX #$FF           F800  A2 FF

```

```

          TXS          F802  9A          INITIALIZE STACK POINTER.
          CLD          F803  D8          CLEAR DECIMAL MODE.
          JSR INIT_REG F804  20 46 F8  INITIALIZE REGISTERS.
          JSR CLEAR_BUFFER F807  20 D0 F8  CLEAR BUFFER.
          JSR INIT_DISP_3 F80A  20 A0 F8  INITIALIZE LED MODULES.
          JSR INIT_ACIA F80D  20 20 F8  INITIALIZE SERIAL INTERFACE.
L01      JSR DISP_BUFFER_2 F810  20 70 F9  DISPLAY BUFFER & SCAN KEY.
          CLC          F813  18
          BCC L01      F814  90 FA

```

```

%-----
% INITIALIZE ACIA
%-----

```

```

          .ORIGIN $F820
%
INIT_ACIA LDA #$03          F820  A9 03
          STA ACIA_STATUS F822  8D 00 80
          LDA #$15          F825  A9 15          4800 BIT/S 8BIT NO PARITY.
          STA ACIA_STATUS F827  8D 00 80
          RTS          F82A  60

```

```

%-----
% SET BUFFER INPUT POINTER TO TOP LEFT
%-----

```

```

SET_BP0_START LDA #$00          F82B  A9 00          SET BP_0 TO $0200, TOP LEFT.
          STA BP_0          F82D  85 00
          LDA #$02          F82F  A9 02
          STA BP_0+1        F831  85 01
          RTS          F833  60

```

```

%-----
% SET BUFFER INPUT POINTER TO BOTTOM LEFT
%-----

```

```

SET_BP0_LEFT LDA #$F0          F834  A9 F0          SET BP_0 TO $02F0, BOTTOM LEFT.
          STA BP_0          F836  85 00
          LDA #$02          F838  A9 02
          STA BP_0+1        F83A  85 01
          RTS          F83C  60

```

```

%-----
% SET BUFFER DISPLAY POINTER TO TOP LEFT
%-----

```

```

SET_BP1_START LDA #$00          F83D  A9 00          SET BP_1 TO $0200, TOP LEFT.
          STA BP_1          F83F  85 02
          LDA #$02          F841  A9 02
          STA BP_1+1        F843  85 03
          RTS          F845  60

```

```

%-----
% INITIALIZE REGISTERS
%-----

```

```

INIT_REG LDA #$18          F846  A9 18          SET BP_2 TO $0318, END+1.
          STA BP_2          F848  85 04
          LDA #$03          F84A  A9 03
          STA BP_2+1        F84C  85 05
          LDA #$F0          F84E  A9 F0          SET BP_3 TO $02F0, BOTTOM LEFT.
          STA BP_3          F850  85 06
          LDA #$02          F852  A9 02
          STA BP_3+1        F854  85 07
          LDA #$FF          F856  A9 FF          SET BP_4 TO $01FF, HEAD-1.
          STA BP_4          F858  85 08
          LDA #$01          F85A  A9 01
          STA BP_4+1        F85C  85 09
          STA B_COUNT       F85E  85 19          INITIALIZE B_COUNT.
          RTS          F860  60

```

```

%-----
% INCREMENT 16BIT POINTER

```

PARAMETER: IX (POINTER ADDRESS)

```
%-----  
      .ORIGIN $F870  
%  
INC_POINTER_2 CLC          F870  18          CLEAR CARRY.  
              LDA $00,X    F871  B5 00      LOAD LOWER 8BIT.  
              ADC #$01     F873  69 01      LOWER 8BIT +1.  
              STA $00,X    F875  95 00  
              BCS L01      F877  B0 01      IF CARRY IS SET, UPPER 8BIT +1.  
              RTS          F879  60  
L01          INC $01,X     F87A  F6 01  
              RTS          F87C  60  
%  
%-----
```

% DECREMENT 16BIT POINTER  
PARAMETER: IX (POINTER ADDRESS)

```
%-----  
%  
DEC_POINTER_2 SEC          F87D  38          CLEAR BORROW.  
              LDA $00,X    F87E  B5 00      LOAD LOWER 8BIT.  
              SBC #$01     F880  E9 01      LOWER 8BIT -1.  
              STA $00,X    F882  95 00  
              BCC L01      F884  90 01      IF BORROW IS SET, UPPER 8BIT +1.  
              RTS          F886  60  
L01          DEC $01,X     F887  D6 01  
              RTS          F889  60  
%  
%-----
```

% COMPARE 16BIT POINTER  
PARAMETER: IX (POINTER NO.1 ADDRESS), IY (POINTER NO.2 ADDRESS),

```
%-----  
%  
CMP_POINTER_2 LDA $01,X    F88A  B5 01  
              CMP $0001,Y  F88C  D9 01 00      COMPARE UPPER 8BIT.  
              BNE L01      F88F  D0 05      IF IT IS NOT THE SAME, EXIT.  
              LDA $00,X    F891  B5 00  
              CMP $0000,Y  F893  D9 00 00      COMPARE LOWER 8BIT.  
L01          RTS          F896  60  
%  
%-----
```

% INITIALIZE LED MODULE

```
%-----  
      .ORIGIN $F8A0  
%  
INIT_DISP_3 LDA #$07      F8A0  A9 07  
              STA PORT    F8A2  8D 00 40      RESET LED MODULE (*RE=L).  
              LDA #$00    F8A5  A9 00  
              STA PORT    F8A7  8D 00 40      REREASE RESET LED MODULE (*RE=H).  
              LDA #$30    F8AA  A9 30  
              STA PORT    F8AC  8D 00 40      (CLK=L,RS=H,*CE=H).  
              LDA #$20    F8AF  A9 20  
              STA PORT    F8B1  8D 00 40      (CLK=L,RS=H,*CE=L).  
              STA REG_1   F8B4  85 14      SAVE CONTROL PATTERN.  
              LDX #$46    F8B6  A2 46      SET LOOP COUNTER TO # OF 35 MODULES.  
L01          STX REG_3    F8B8  86 15      SAVE LOOP COUNTER.  
              LDA #$4A    F8BA  A9 4A      SET CONTROL WORD 0.  
              JSR OUTFNT_SERL2 F8BC  20 00 F9      SEND SERIAL DATA TO LED MODULES.  
              LDX REG_3   F8BF  A6 15      LOAD LOOP COUNTER.  
              DEX         F8C1  CA        LOOP COUNTER -1.  
              BNE L01     F8C2  D0 F4  
              JMP PATCH_1 F8C4  4C C0 FA      JUMP TO PATCH.  
%  
%-----
```

% CLEAR BUFFER

```
%-----  
      .ORIGIN $F8D0  
%  
CLEAR_BUFFER JSR SET_BP0_START F8D0  20 2B F8      SET BP_0 TO START ADDRESS, TOP LEFT.  
L01          LDA #$20      F8D3  A9 20      ASCII CODE "SPACE".  
              LDY #$00     F8D5  A0 00  
              STA (BP_0),Y F8D7  91 00      CLEAR ONE CHARACTER.  
              LDX #BP_0    F8D9  A2 00
```

```

        JSR INC_POINTER_2  F8DB  20 70 F8      INCREMENT BUFFER POINTER.
        LDX #BP_2          F8DE  A2 04
        JSR CMP_POINTER_2  F8E0  20 8A F8      COMPARE BUFFER POINTER AND END ADDRESS.
        BNE L01            F8E3  D0 EE
        LDA #5F             F8E5  A9 5F
        STA $02F0           F8E7  8D F0 02      SET UNDERSCORE CODE TO BOTTOM LEFT.
        JSR SET_BP0_LEFT    F8EA  20 34 F8      SET POINTER TO BOTTOM LEFT.
        RTS                 F8ED  60

%-----
% OUTPUT 8 BIT FONT PATTERN TO LED DISPLAY MODULE BY SERIAL
% INPUT PARAMETER ACC: 8BIT FONT DATA OR CONTROL REGISTER VALUE
% REG_1: CURRENT CONTROL LINE LEVEL
%-----
        .ORIGIN $F900

%
OUTFONT_SERL2 LDX #08          F900  A2 08      SET LOOP COUNTER (8BIT).
              STA REG_0       F902  85 13      SAVE FONT DATA.
              LDA REG_1       F904  A5 14      RETURN CONTROL DATA.
              AND #3F         F906  29 3F      (DATA=L,CLK=L,RS=,*CE=).
              STA REG_1       F908  85 14
L01          LDA REG_0       F90A  A5 13      RETURN FONT DATA.
              AND #80         F90C  29 80      EXTRACT D7.
              ORA REG_1       F90E  05 14      SYNTHESIZE D7 AND CONTROL DATA.
              STA PORT        F910  8D 00 40    OUTPUT TO LED MODULE.
              ORA #40         F913  09 40      (DATA=L,CLK=H,RS=,*CE=).
              STA PORT        F915  8D 00 40    OUTPUT TO LED MODULE.
              ASL REG_0       F918  06 13      SHIFT FONT DATA TO LEFT.
              DEX             F91A  CA        LOOP COUNTER -1.
              BNE L01         F91B  D0 ED
              LDA ACIA_STATUS F91D  AD 00 80    RECEIVED FROM SERIAL INTERFACE?.
              LSR             F920  4A
              BCC L02         F921  90 06      IF NOT RECEIVED, EXIT.
              LDA ACIA_DATA   F923  AD 01 80
              JSR REC_1_CHA   F926  20 B0 F9    IF RECEIVED, GO TO 1 CHARACTER PROCESS.
L02          RTS                 F929  60

%-----
% OUTPUT 5 BYTE OF FONT PATTERN TO LED DISPLAY
% INPUT PARAMETER ACC:ASCII CODE
% FONT POINTER: (ASCII CODE - $20) THEN SHIFT LEFT 3BIT = (D4 TO D8) IS 16BIT POINTER
%-----
        .ORIGIN $F940

%
OUT_FONT_2   STA TEMP_0       F940  85 16
              LDX #FONT_DATA_H F942  A2 FD      SET FONT DATA ADDRESS UPPER 8 BIT.
              STX FP+1        F944  86 0B
              SEC             F946  38
              SBC #20         F947  E9 20      ASCII CODE -$20.
              AND #3F         F949  29 3F      CLEAR UPPER 2 BIT.
              ASL             F94B  0A        SHIFT LEFT 3 BIT.
              ASL             F94C  0A
              ASL             F94D  0A
              BCC L01         F94E  90 02
              INC FP+1        F950  E6 0B      IF CARRY=1, UPPER 8BIT + 1.
L01          STA FP           F952  85 0A      SET CODE TO FONT POINTER LOWER 9 BIT.
              LDY #05         F954  A0 05      IY: LOOP COUNTER FOR 5 BYTE FONT.
L02          LDX #00         F956  A2 00
              LDA (FP,X)     F958  A1 0A      EXTRACT FONT DATA.
              STY TEMP_2     F95A  84 18      SAVE LOOP COUNTER.
              JSR OUTFONT_SERL2 F95C  20 00 F9    OUTPUT FONT BY SERIAL.
              LDX #FP        F95F  A2 0A
              JSR INC_POINTER_2 F961  20 70 F8    FONT POINTER +1.
              LDY TEMP_2     F964  A4 18
              DEY            F966  88        LOOP COUNTER -1.
              BNE L02         F967  D0 ED      REPEAT UNTIL OUTPUT 5BYTE IS COMPLETED.
              RTS                 F969  60

%-----
% UPDATE DISPLAY FOR ENTIRE BUFFER
%-----
        .ORIGIN $F970

```

```

%
DISP_BUFFER_2 LDA REG_1          F970  A5 14
                AND #$D0         F972  29 D0      (DATA=,CLK=,RS=L,*CE=).
                STA PORT         F974  8D 00 40
                AND #$C0         F977  29 C0      (DATA=,CLK=,RS=L,*CE=L).
                STA PORT         F979  8D 00 40
                STA REG_1        F97C  85 14      SAVE CONTROL PATTERN.
L01            JSR SET_BP1_START  F97E  20 3D F8  SET BP_1 TO BUFFER HEAD ADDRESS.
                LDY #$00         F981  A0 00
                LDA (BP_1),Y     F983  B1 02      GET 1 CHARACTER FROM BUFFER.
                JSR OUT_FONT_2    F985  20 40 F9  SEND 1 CHARACTER FONT TO LED.
                JSR KEY_SCAN_2    F988  20 E0 FA  KEY SCAN.
                LDX #BP_1        F98B  A2 02
                JSR INC_POINTER_2 F98D  20 70 F8  DISPLAY POINTER +1.
                LDY #BP_2        F990  A0 04
                JSR CMP_POINTER_2 F992  20 8A F8  COMPARE DISPLAY POINTER AND BUFFER END.
                BNE L01          F995  D0 EA
                LDA REG_1        F997  A5 14
                ORA #$10         F999  09 10      (DATA=,CLK=,RS=,*CE=H).
                STA PORT         F99B  8D 00 40
                AND #$B0         F99E  29 B0      (DATA=,CLK=L,RS=,*CE=).
                STA PORT         F9A0  8D 00 40
                STA REG_1        F9A3  85 14
                JSR BLINK_CUSOR   F9A5  20 F0 FA
                RTS              F9A8  60

%
%-----
% RECEIVE 1 CHARACTER
%-----
                .ORIGIN $F9B0

%
REC_1_CHA     CMP #$61          F9B0  C9 61      ASCII CODE > $60?
                BCC L01         F9B2  90 03
                SEC              F9B4  38
L01            SBC #$20         F9B5  E9 20      CONVERT LOWERCASE TO UPPERCASE.
                CMP #$0A        F9B7  C9 0A      LF CODE?
                BEQ SCROLL      F9B9  F0 2E      IF LF CODE THEN GO TO SCROLL PROCESS.
                CMP #$0D        F9BB  C9 0D      CR CODE?
                BEQ LEFT_CUSOR_1 F9BD  F0 68      IF CR CODE THEN MOVE CURSOR TO LEFT END
                CMP #$08        F9BF  C9 08      BS CODE?
                BNE L03         F9C1  D0 18
BACK_SPACE    LDA #$20         F9C3  A9 20      BACK SPACE PROCESS.
                LDY #$00         F9C5  A0 00
                STA (BP_0),Y     F9C7  91 00      CLEAR CURRENT POINTER POSITION.
                LDX #BP_0        F9C9  A2 00
                JSR DEC_POINTER_2 F9CB  20 7D F8  POINTER -1.
                LDY #BP_4        F9CE  A0 08
                JSR CMP_POINTER_2 F9D0  20 8A F8  KEEP POINTER OVER THAN TOP.
                BNE L04         F9D3  D0 0D
                JSR INC_POINTER_2 F9D5  20 70 F8
                CLC              F9D8  18
                BCC L04         F9D9  90 07
L03            LDY #$00         F9DB  A0 00
                STA (BP_0),Y     F9DD  91 00      WRITE 1 CHARACTER TO BUFFER.
                JSR INC_POINTER_2 F9DF  20 70 F8  POINTER +1.
L04            LDY #BP_2        F9E2  A0 04
                JSR CMP_POINTER_2 F9E4  20 8A F8  BUFFER RIGHT END?
                BNE L05         F9E7  D0 37
SCROLL        LDY #BP_3        F9E9  A0 06
                JSR CMP_POINTER_2 F9EB  20 8A F8  BUFFER LEFT END?
                BEQ SCROLL_2     F9EE  F0 06
                LDA #$20         F9F0  A9 20
                LDY #$00         F9F2  A0 00
                STA (BP_0),Y     F9F4  91 00      IF POINTER IS NOT LEFT END,
                LDX #$10         F9F6  A2 10      CLEAR POINTER POSITION.
SCROLL_2     LDA $0218,X        F9F8  BD 18 02  SET BUFFER POINTER OFFSET.
L01            STA $01F0,X        F9FB  9D F0 01  LOAD UNDER LINE.
                INX              F9FE  E8      STORE UPPER LINE.
                BNE L01          F9FF  D0 F7  POINTER +1.
                LDA #$20         FA01  A9 20      REPEAT UNTIL POINTER=0.
                LDX #$28         FA03  A2 28
L02            STA $02EF,X        FA05  9D EF 02  SET BUFFER POINTER OFFSET.
                CLEAR BOTTOM LINE.

```

```

DEX          FA08  CA          POINTER -1.
BNE L02      FA09  D0 FA      REPEAT UNTIL POINTER=0.
BEQ LEFT_CURSOR FA0B  F0 10    GO TO SET CURSOL TO LEFT END.
%
      .ORIGIN $FA1D
%
LEFT_CURSOR JSR SET_BP0_LEFT  FA1D  20 34 F8  SET POINTER TO LEFT END.
L05         LDA #$5F          FA20  A9 5F  UNDERSCORE CODE.
           LDY #$00          FA22  A0 00
           STA (BP_0),Y      FA24  91 00  DISPLAY UNDERSCORE TO POINTED POSITION.
           RTS              FA26  60
%
LEFT_CURSOR_1 LDA #$20       FA27  A9 20
           LDY #$00       FA29  A0 00
           STA (BP_0),Y   FA2B  91 00  CLEAR UNDERSCORE.
           JSR SET_BP0_LEFT FA2D  20 34 F8  SET POINTER TO LEFT END.
           RTS              FA30  60
%
-----
% SCAN KEY
% SCAN_PAT_2 USING ACC: VALUE $00 TO $4F, INCREMENT BY $8.
% SCAN_PAT USING IX: VALUE $0 TO $5
% LOOP COUNTER USING IY
%
-----
      .ORIGIN $FA40
%
KEY_SCAN   LDA #$FF          FA40  A9 FF  SET KEY SCAN TABLE ADDRESS UPPER8 BIT.
           STA TP_0+1       FA42  85 0D
           LDA #$00         FA44  A9 00
           STA SCAN_PAT_2   FA46  85 0F  INITIALIZE SCAN PATTERN NUMBER ($00).
           LDX #$00         FA48  A2 00
           STX SCAN_PAT     FA4A  86 0E  INITIALIZE SCAN PATTERN ($00).
           STX SHIFT_FLAG   FA4C  86 12  CLEAR SHIFT KEY FLAG.
L07        STA PUSH_FLAG_1  FA4E  85 10  CLEAR KEY PUSHED FLAG.
L04        JSR OUT_PAT      FA50  20 AE FA  OUTPUT SCAN PATTERN.
           LDA PORT         FA53  AD 00 40  INPUT SCAN RESULT.
           EOR #$FF         FA56  49 FF  INVERT SCAN RESULT.
           BEQ L01          FA58  F0 33  $00? (NOT PRESSED).
           LDY #$00         FA5A  A0 00  CONVERT SCAN RESULT TO 3BIT DATA.
L03        LSR              FA5C  4A          SHIFT RIGHT 1BIT.
           BCS L02          FA5D  B0 04
           INY              FA5F  C8          LOOP COUNTER +1.
           CLC              FA60  18
           BCC L03          FA61  90 F9
L02        LDA SCAN_PAT_2   FA63  A5 0F
           STY TEMP_1       FA65  84 17
           CLC              FA67  18
           ADC TEMP_1       FA68  65 17  ADD 3BIT DATA AND KEY SCAN NUMBER.
           LDY #$01         FA6A  A0 01
           STY PUSH_FLAG_1  FA6C  84 10  SET PRESSED FLAG TO '1'.
           LDY PUSH_FLAG_2  FA6E  A4 11  LAST PRESSED FLAG = '1' ?
           BNE L06          FA70  D0 16  IGNORE HOLDING DOWN.
           CMP #$00         FA72  C9 00  SHIFT KEY PRESSED?
           BNE L05          FA74  F0 27  IF SHIFT KEY IS NOT PRESSED,
           STA TP_0         FA76  85 0C  STORE TABLE INDEX TO IX.
           LDY #$00         FA78  A0 00
           LDA (TP_0),Y     FA7A  B1 0C  GET ASCII CODE BY USING TABLE.
           PHA              FA7C  48
OUT_1CHA   LDA ACIA_STATUS  FA7D  AD 00 80  SEND 1 CHARACTER THROUGH ACIA.
           LSR              FA80  4A
           LSR              FA81  4A
           BCC OUT_1CHA     FA82  90 F9
           PLA              FA84  68
           STA ACIA_DATA    FA85  8D 01 80
L06        LDA PUSH_FLAG_1  FA88  A5 10  UPDATE KEY PRESSED LAST TIME FLAG.
           STA PUSH_FLAG_2  FA8A  85 11
           RTS              FA8C  60  END OF KEY SCAN SUBROUTINE
%
L01        LDA SCAN_PAT_2   FA8D  A5 0F
           CLC              FA8F  18
           ADC #$08         FA90  69 08  UPDATE SCAN PATTEN NUM.(D3 OR HIGHER+1)
           STA SCAN_PAT_2   FA92  85 0F

```

```

        LDX SCAN_PAT      FA94  A6 0E
        INX               FA96  E8
        CPX #$06         FA97  E0 06      FINISH SCAN PATTERN $05?
        BNE L04          FA99  D0 B5      IF NOT FINISH RETURN OUT SCAN PATTERN.
        BEQ L06          FA9B  F0 EB      IF FINISH GO TO UPDATE PRESSED FLAG.
%
L05      LDA #$01         FA9D  A9 01      SHIFT KEY PRESSED PROCESS.
        STA SHIFT_FLAG   FA9F  85 12      SET SHIFT FLAG TO '1'.
        LDX #$03         FAA1  A2 03      SET SCAN PATTREN $03(SHIFT VALID RANGE)
        STX SCAN_PAT     FAA3  86 0E
        LDA #$38         FAA5  A9 38      SET SCAN PATTERN NUMBER TO $38.
        STA SCAN_PAT_2   FAA7  85 0F
        LDA #$00         FAA9  A9 00
        JMP L07          FAAB  4C 4E FA    RETURN TO DETECTING PRESSED.
%
OUT_PAT  STX SCAN_PAT     FAAE  86 0E      SAVE SCAN PATTERN.
        TXA             FAB0  8A
        AND #$0F         FAB1  29 0F      D4-D7 FORCED ZERO.
        STA PORT         FAB3  8D 00 40    OUTPUT SCAN PATTERN.
        RTS             FAB6  60
%
-----
% PATCH FOR INITIALIZE LED MODULE
-----
        .ORIGIN $FAC0
%
PATCH_1  LDA #$70       FAC0  A9 70      (DATA=,CLK=H,RS=H,*CE=H).
        STA PORT         FAC2  8D 00 40
        LDA #$30         FAC5  A9 30      (DATA=,CLK=L,RS=H,*CE=H).
        STA PORT         FAC7  8D 00 40
        STA REG_1        FACA  85 14
        RTS             FACC  60
%
-----
% PATCH FOR SCANNING KEY INTERVAL
%       SET KEY SCAN INTERVAL TO 25 MILLI SECOND
-----
        .ORIGIN $FAE0
%
KEY_SCAN_2 LDA BP_1      FAE0  A5 02      LOAD DISPLAY POINTER LOWER 8BIT.
        AND #$1F         FAE2  29 1F      EXTRACT LOWER 5BIT.
        BEQ L01          FAE4  F0 01      IF IT IS ZERO, SCAN KEY.
        RTS             FAE6  60
L01      JSR KEY_SCAN    FAE7  20 40 FA    SCAN KEY.
        RTS             FAEA  60
%
-----
% BLINK CURSOR
-----
        .ORIGIN $FAE0
%
BLINK_CURSOR LDA B_COUNT FAF0  A5 19      LOAD BLINK COUNTER.
        BNE L01          FAF2  D0 15      IF IT IS NOT ZERO, COUNTER-1 & EXIT.
        LDY #$00         FAF4  A0 00      IT IT IS ZERO, LOAD CURRENT CHARACTER.
        LDA (BP_0),Y     FAF6  B1 00
        CMP #$20         FAF8  C9 20      SPACE CODE?
        BNE L02          FAFB  D0 04
        LDA #$5F         FAFD  A9 5F
        BNE L03          FAFE  D0 02
L02      LDA #$20         FB00  A9 20      IF IT IS NOT SPACE, DISPLAY SPACE,
L03      STA (BP_0),Y     FB02  91 00      ELSE DISPLAY UNDERSCORE.
        LDA #$01         FB04  A9 01
        STA B_COUNT      FB06  85 19      INITIALIZE BLINK COUNTER.
        RTS             FB08  60
%
L01      DEC B_COUNT     FB09  C6 19      BLINK COUNTER -1.
        RTS             FB0B  60
%
-----
% CHARACTER FONT DATA TABLE
% 5 X 7 DOT MATRIX CHARACTER FONT
-----

```

%	.ORIGIN \$FD00	ADDRESS(HEX)	DATA(HEX)	CHARACTER
		FD00	00 00 00 00 00	' '
		FD08	00 5F 00 00 00	'!'
		FD10	00 03 00 03 00	'"'
		FD18	14 7F 14 7F 14	'#'
		FD20	24 2A 7F 2A 12	'\$'
		FD28	23 13 08 64 62	'%'
		FD30	36 49 56 20 50	'&'
		FD38	00 0B 07 00 00	'\''
		FD40	00 00 3E 41 00	'('
		FD48	00 41 3E 00 00	')'
		FD50	08 2A 1C 2A 08	'*'
		FD58	08 08 3E 08 08	'+'
		FD60	00 58 38 00 00	'\''
		FD68	08 08 08 08 08	'-'
		FD70	00 30 30 00 00	'.'
		FD78	20 10 08 04 02	'/'
		FD80	3E 51 49 45 3E	'0'
		FD88	00 42 7F 40 00	'1'
		FD90	62 51 49 49 46	'2'
		FD98	22 41 49 49 36	'3'
		FDA0	18 14 12 7F 10	'4'
		FDA8	27 45 45 45 39	'5'
		FDB0	3C 4A 49 49 30	'6'
		FDB8	01 71 09 05 03	'7'
		FDC0	36 49 49 49 36	'8'
		FDC8	06 49 49 29 1E	'9'
		FDD0	00 36 36 00 00	':'
		FDD8	00 5B 3B 00 00	':'
		FDE0	00 08 14 22 41	'<'
		FDE8	14 14 14 14 14	'='
		PDF0	41 22 14 08 00	'>'
		PDF8	02 01 51 09 06	'?'
		FE00	3E 41 5D 55 1E	'@'
		FE08	7E 09 09 09 7E	'A'
		FE10	7F 49 49 49 36	'B'
		FE18	3E 41 41 41 22	'C'
		FE20	7F 41 41 41 3E	'D'
		FE28	7F 49 49 49 41	'E'
		FE30	7F 09 09 09 01	'F'
		FE38	3E 41 41 51 72	'G'
		FE40	7F 08 08 08 7F	'H'
		FE48	00 41 7F 41 00	'I'
		FE50	20 40 40 40 3F	'J'
		FE58	7F 08 14 22 41	'K'
		FE60	7F 40 40 40 40	'L'
		FE68	7F 02 0C 02 7F	'M'
		FE70	7F 04 08 10 7F	'N'
		FE78	3E 41 41 41 3E	'O'
		FE80	7F 09 09 09 06	'P'
		FE88	3E 41 51 21 5E	'Q'
		FE90	7F 09 19 29 46	'R'
		FE98	26 49 49 49 32	'S'
		FEA0	01 01 7F 01 01	'T'
		FEA8	3F 40 40 40 3F	'U'
		FEB0	07 18 60 18 07	'V'
		FEB8	7F 20 18 20 7F	'W'
		FEC0	63 14 08 14 63	'X'
		FEC8	03 04 78 04 03	'Y'
		FED0	61 51 49 45 43	'Z'
		FED8	00 00 7F 41 41	'['
		FEE0	02 04 08 10 20	'¥'
		FEE8	41 41 7F 00 00	']'
		FEF0	04 02 01 02 04	'^'
		FEF8	40 40 40 40 40	'_'

%  
 %-----  
 % KEY SCAN ASCII CODE TABLE  
 %-----  
 .ORIGIN \$FF00



%

ADDRESS(HEX) DATA(HEX)

FF00	FF	5A	58	43	56	42	4E	4D
FF08	41	53	44	46	47	48	4A	4B
FF10	51	57	45	52	54	59	55	49
FF18	31	32	33	34	35	36	37	38
FF20	2C	2E	2F	20	4F	50	40	0D
FF28	4C	3B	3A	0A	39	30	2D	08
FF30	FF	FF	FF	FF	FF	FF	FF	FF
FF38	21	22	23	24	25	26	27	28
FF40	3C	3E	3F	FF	FF	FF	5E	FF
FF48	FF	2B	2A	FF	29	5C	3D	FF

%

%-----

% END OF PROGRAM