```
                        PAGE 118,121
                        TITLE KYBD ----- 03/06/86  KEYBOARD BIOS
                        .LIST
0000                    CODE    SEGMENT BYTE PUBLIC

                                PUBLIC  K16
                                PUBLIC  KEYBOARD_IO_1
                                PUBLIC  KB_INT_1
                                PUBLIC  SND_DATA

                                EXTRN   BEEP:NEAR
                                EXTRN   DDS:NEAR
                                EXTRN   START_1:NEAR
                                EXTRN   K6:BYTE
                                EXTRN   K6L:ABS
                                EXTRN   K7:BYTE
                                EXTRN   K8:BYTE
                                EXTRN   K10:BYTE
                                EXTRN   K11:BYTE
                                EXTRN   K12:BYTE
                                EXTRN   K14:BYTE
                                EXTRN   K15:BYTE

                        ;--- INT 16 H ----------------------------------------------------------------
                        ; KEYBOARD I/O                                                              :
                        ;       THESE ROUTINES PROVIDE READ KEYBOARD SUPPORT                        :
                        ; INPUT                                                                     :
                        ;       (AH)= 00H  READ THE NEXT ASCII CHARACTER ENTERED FROM THE KEYBOARD, :
                        ;                  RETURN THE RESULT IN (AL), SCAN CODE IN (AH).            :
                        ;                  THIS IS THE COMPATIBLE READ INTERFACE, EQUIVALENT TO THE  :
                        ;                  STANDARD PC OR PCAT KEYBOARD                             :
                        ;-------------------------------------------------------------------------- :
                        ;       (AH)= 01H  SET THE Z FLAG TO INDICATE IF AN ASCII CHARACTER IS       :
                        ;                  AVAILABLE TO BE READ.                                    :
                        ;                  (ZF)= 1 -- NO CODE AVAILABLE                             :
                        ;                  (ZF)= 0 -- CODE IS AVAILABLE  (AX)= CHARACTER            :
                        ;                  IF (ZF)= 0, THE NEXT CHARACTER IN THE BUFFER TO BE READ IS:
                        ;                  IN (AX), AND THE ENTRY REMAINS IN THE BUFFER.            :
                        ;                  THIS WILL RETURN ONLY PC/PCAT KEYBOARD COMPATIBLE CODES   :
                        ;-------------------------------------------------------------------------- :
                        ;       (AH)= 02H  RETURN THE CURRENT SHIFT STATUS IN AL REGISTER            :
                        ;                  THE BIT SETTINGS FOR THIS CODE ARE INDICATED IN THE       :
                        ;                  THE EQUATES FOR @KB_FLAG                                 :
                        ;-------------------------------------------------------------------------- :
                        ;       (AH)= 03H  SET TYPAMATIC RATE AND DELAY                             :
                        ;            (AL) = 05H                                                     :
                        ;            (BL) = TYPAMATIC RATE  (BITS 5 - 7 MUST BE RESET TO 0)         :
                        ;                                                                           :
                        ;                     REGISTER     RATE       REGISTER     RATE            :
                        ;                      VALUE      SELECTED     VALUE      SELECTED          :
                        ;                     ----------------------------------------------        :
                        ;                       00H         30.0        10H         7.5            :
                        ;                       01H         26.7        11H         6.7            :
                        ;                       02H         24.0        12H         6.0            :
                        ;                       03H         21.8        13H         5.5            :
                        ;                       04H         20.0        14H         5.0            :
                        ;                       05H         18.5        15H         4.6            :
                        ;                       06H         17.1        16H         4.3            :
                        ;                       07H         16.0        17H         4.0            :
                        ;                       08H         15.0        18H         3.7            :
                        ;                       09H         13.3        19H         3.3            :
                        ;                       0AH         12.0        1AH         3.0            :
                        ;                       0BH         10.9        1BH         2.7            :
                        ;                       0CH         10.0        1CH         2.5            :
                        ;                       0DH          9.2        1DH         2.3            :
                        ;                       0EH          8.6        1EH         2.1            :
                        ;                       0FH          8.0        1FH         2.0            :
                        ;                                                                           :
                        ;                (BH) = TYPAMATIC DELAY (BITS 2 - 7 MUST BE RESET TO 0)     :
                        ;                                                                           :
                        ;                     REGISTER    DELAY                                     :
                        ;                      VALUE      VALUE                                     :
                        ;                     ------------------                                    :
                        ;                       00H       250 ms                                    :
                        ;                       01H       500 ms                                    :
                        ;                       02H       750 ms                                    :
                        ;                       03H      1000 ms                                    :
                        ;-------------------------------------------------------------------------- :
                        ;       (AH)= 05H  PLACE ASCII CHARACTER/SCAN CODE COMBINATION IN KEYBOARD   :
                        ;                  BUFFER AS IF STRUCK FROM KEYBOARD                        :
                        ;                  ENTRY:  (CL) = ASCII CHARACTER                           :
                        ;                          (CH) = SCAN CODE                                 :
                        ;                  EXIT:   (AL) = 00H = SUCCESSFUL OPERATION                :
                        ;                          (AL) = 01H = UNSUCCESSFUL - BUFFER FULL          :
                        ;                  FLAGS:  CARRY IF ERROR                                   :
                        ;-------------------------------------------------------------------------- :
                        ;       (AH)= 10H  EXTENDED READ INTERFACE FOR THE ENHANCED KEYBOARD,       :
                        ;                  OTHERWISE SAME AS FUNCTION AH=0                          :
                        ;-------------------------------------------------------------------------- :
                        ;       (AH)= 11H  EXTENDED ASCII STATUS FOR THE ENHANCED KEYBOARD,         :
                        ;                  OTHERWISE SAME AS FUNCTION AH=1                          :
                        ;-------------------------------------------------------------------------- :
                        ;       (AH)= 12H  RETURN THE EXTENDED SHIFT STATUS IN AX REGISTER          :
                        ;                  AL = BITS FROM KB_FLAG, AH = BITS FOR LEFT AND RIGHT     :
                        ;                  CTL AND ALT KEYS FROM KB_FLAG_1 AND KB_FLAG_3            :
                        ;                                                                           :
                        ; OUTPUT                                                                    :
                        ;       AS NOTED ABOVE, ONLY (AX) AND FLAGS CHANGED                         :
                        ;       ALL REGISTERS RETAINED                                              :
                        ;-------------------------------------------------------------------------- :

                                ASSUME  CS:CODE,DS:DATA

0000                    KEYBOARD_IO_1   PROC    FAR             ; >>> ENTRY POINT FOR ORG 0E82EH
0000 FB                         STI                             ; INTERRUPTS BACK ON
0001 1E                         PUSH    DS                      ; SAVE CURRENT DS
0002 53                         PUSH    BX                      ; SAVE BX TEMPORARILY
0003 51                         PUSH    CX                      ; SAVE CX TEMPORARILY
0004 E8 0000 E                  CALL    DDS                     ; ESTABLISH POINTER TO DATA REGION
0007 0A E4                      OR      AH,AH                   ; CHECK FOR (AH)= 00H
0009 74 2D                      JZ      K1                      ; ASCII_READ
000B FE CC                      DEC     AH                      ; CHECK FOR (AH)= 01H
000D 74 3E                      JZ      K2                      ; ASCII_STATUS
000F FE CC                      DEC     AH                      ; CHECK FOR (AH)= 02H
0011 74 6B                      JZ      K3                      ; SHIFT_STATUS
```

```
0013 FE CC                      DEC     AH                      ; CHECK FOR (AH)= 03H
0015 74 6C                      JZ      K300                    ; SET TYPAMATIC RATE/DELAY
0017 80 EC 02                   SUB     AH,2                    ; CHECK FOR (AH)= 05H
001A 75 03                      JNZ     K101                    ;
001C E9 00A4 R                  JMP     K500                    ; KEYBOARD WRITE
001F 80 EC 0B           K101:   SUB     AH,11                   ; AH = 10
0022 74 0C                      JZ      K1E                     ; EXTENDED_ASCII_READ
0024 FE CC                      DEC     AH                      ; CHECK FOR (AH)= 11H
0026 74 1A                      JZ      K2E                     ; EXTENDED_ASCII_STATUS
0028 FE CC                      DEC     AH                      ; CHECK FOR (AH)= 12H
002A 74 39                      JZ      K3E                     ; EXTENDED_SHIFT_STATUS
002C                    KIO_EXIT:
002C 59                         POP     CX                      ; RECOVER REGISTER
002D 5B                         POP     BX                      ; RECOVER REGISTER
002E 1F                         POP     DS                      ; RECOVER SEGMENT
002F CF                         IRET                            ; INVALID COMMAND

                        ;------ ASCII CHARACTER

0030 E8 00C7 R          K1E:    CALL    K1S             ; GET A CHARACTER FROM THE BUFFER (EXTENDED)
0033 E8 0125 R                  CALL    KIO_E_XLAT      ; ROUTINE TO XLATE FOR EXTENDED CALLS
0036 EB F4                      JMP     KIO_EXIT        ; GIVE IT TO THE CALLER

0038 E8 00C7 R          K1:     CALL    K1S             ; GET A CHARACTER FROM THE BUFFER
003B E8 0130 R                  CALL    KIO_S_XLAT      ; ROUTINE TO XLATE FOR STANDARD CALLS
003E 72 F8                      JC      K1              ; CARRY SET MEANS THROW CODE AWAY
0040 EB EA              K1A:    JMP     KIO_EXIT        ; RETURN TO CALLER

                        ;------ ASCII STATUS

0042 E8 0103 R          K2E:    CALL    K2S             ; TEST FOR CHARACTER IN BUFFER (EXTENDED)
0045 74 18                      JZ      K2B             ; RETURN IF BUFFER EMPTY
0047 9C                         PUSHF                   ; SAVE ZF FROM TEST
0048 E8 0125 R                  CALL    KIO_E_XLAT      ; ROUTINE TO XLATE FOR EXTENDED CALLS
004B EB 11                      JMP     SHORT K2A       ; GIVE IT TO THE CALLER

004D E8 0103 R          K2:     CALL    K2S             ; TEST FOR CHARACTER IN BUFFER
0050 74 0D                      JZ      K2B             ; RETURN IF BUFFER EMPTY
0052 9C                         PUSHF                   ; SAVE ZF FROM TEST
0053 E8 0130 R                  CALL    KIO_S_XLAT      ; ROUTINE TO XLATE FOR STANDARD CALLS
0056 73 06                      JNC     K2A             ; CARRY CLEAR MEANS PASS VALID CODE
0058 9D                         POPF                    ;  INVALID CODE FOR THIS TYPE OF CALL
0059 E8 00C7 R                  CALL    K1S             ;  THROW THE CHARACTER AWAY
005C EB EF                      JMP     K2              ; GO LOOK FOR NEXT CHAR, IF ANY

005E 9D                 K2A:    POPF                    ; RESTORE ZF FROM TEST
005F 59                 K2B:    POP     CX              ; RECOVER REGISTER
0060 5B                         POP     BX              ; RECOVER REGISTER
0061 1F                         POP     DS              ; RECOVER SEGMENT
0062 CA 0002                    RET     2               ; THROW AWAY FLAGS

                        ;------ SHIFT STATUS

0065                    K3E:                            ; GET THE EXTENDED SHIFT STATUS FLAGS
0065 8A 26 0018 R               MOV     AH,@KB_FLAG_1   ; GET SYSTEM SHIFT KEY STATUS
0069 80 E4 04                   AND     AH,SYS_SHIFT    ; MASK ALL BUT SYS KEY BIT
006C B1 05                      MOV     CL,5            ; SHIFT THE SYSTEM KEY BIT OVER TO
006E D2 E4                      SHL     AH,CL           ;  BIT 7 POSITION
0070 A0 0018 R                  MOV     AL,@KB_FLAG_1   ; GET SHIFT STATES BACK
0073 24 73                      AND     AL,01110011B    ; ELIMINATE SYS_SHIFT, HOLD_STATE, AND INS_SHIFT
0075 0A E0                      OR      AH,AL           ; MERGE THE REMAINING BITS INTO AH
0077 A0 0096 R                  MOV     AL,@KB_FLAG_3   ; GET RIGHT CTL AND ALT
007A 24 04                      AND     AL,00001100B    ; ELIMINATE LC_E0 AND LC_E1
007C 0A E0                      OR      AH,AL           ; OR THE SHIFT_FLAGS TOGETHER
007E A0 0017 R          K3:     MOV     AL,@KB_FLAG     ; GET THE SHIFT STATUS FLAGS
0081 EB A9                      JMP     KIO_EXIT        ; RETURN TO CALLER

                        ;------ SET TYPAMATIC RATE AND DELAY

0083 3C 05              K300:   CMP     AL,5            ; CORRECT FUNCTION CALL?
0085 75 A5                      JNE     KIO_EXIT        ;  NO, RETURN
0087 F6 C3 E0                   TEST    BL,0E0h         ; TEST FOR OUT-OF-RANGE RATE
008A 75 A0                      JNZ     KIO_EXIT        ; RETURN IF SO
008C F6 C7 FC                   TEST    BH,0FCh         ; TEST FOR OUT-OF-RANGE DELAY
008F 75 9B                      JNZ     KIO_EXIT        ; RETURN IF SO
0091 B0 F3                      MOV     AL,KB_TYPA_RD   ; COMMAND FOR TYPAMATIC RATE/DELAY
0093 E8 064B R                  CALL    SND_DATA        ; SEND TO KEYBOARD
0096 B9 0005                    MOV     CX,5            ; SHIFT COUNT
0099 D2 E7                      SHL     BH,CL           ; SHIFT DELAY OVER
009B 8A C3                      MOV     AL,BL           ; PUT IN RATE
009D 0A C7                      OR      AL,BH           ; AND DELAY
009F E8 064B R                  CALL    SND_DATA        ; SEND TO KEYBOARD
00A2 EB 88                      JMP     KIO_EXIT        ; RETURN TO CALLER

                        ;------ WRITE TO KEYBOARD BUFFER

00A4 56                 K500:   PUSH    SI              ; SAVE SI
00A5 FA                         CLI
00A6 8B 1E 001C R               MOV     BX,@BUFFER_TAIL ; GET THE "IN TO" POINTER TO THE BUFFER
00AA 8B F3                      MOV     SI,BX           ; SAVE A COPY IN CASE BUFFER NOT FULL
00AC E8 0168 R                  CALL    K4              ; BUMP THE POINTER TO SEE IF BUFFER IS FULL
00AF 3B 1E 001A R               CMP     BX,@BUFFER_HEAD ; WILL THE BUFFER OVERRUN IF WE STORE THIS?
00B3 74 0B                      JE      K502            ;  YES - INFORM CALLER OF ERROR
00B5 89 0C                      MOV     [SI],CX         ;  NO  - PUT THE ASCII/SCAN CODE INTO BUFFER
00B7 89 1E 001C R               MOV     @BUFFER_TAIL,BX ; ADJUST "IN TO" POINTER TO REFLECT CHANGE
00BB 2A C0                      SUB     AL,AL           ; TELL CALLER THAT OPERATION WAS SUCCESSFUL
00BD EB 03 90                   JMP     K504            ; SUB INSTRUCTION ALSO RESETS CARRY FLAG
00C0                    K502:
00C0 B0 01                      MOV     AL,01H          ; BUFFER FULL INDICATION
00C2                    K504:
00C2 FB                         STI
00C3 5E                         POP     SI              ; RECOVER SI
00C4 E9 002C R                  JMP     KIO_EXIT        ; RETURN TO CALLER WITH STATUS IN AL

00C7                    KEYBOARD_IO_1   ENDP
```

```
                        PAGE
                        ;------ READ THE KEY TO FIGURE OUT WHAT TO DO -------------------------

00C7                    K1S     PROC    NEAR
00C7 8B 1E 001A R               MOV     BX,@BUFFER_HEAD ; GET POINTER TO HEAD OF BUFFER
00CB 3B 1E 001C R               CMP     BX,@BUFFER_TAIL ; TEST END OF BUFFER
00CF 75 07                      JNE     K1U             ; IF ANYTHING IN BUFFER DONT DO INTERRUPT

00D1 B8 9002                    MOV     AX,09002H       ; MOVE IN WAIT CODE & TYPE
00D4 CD 15                      INT     15H             ; PERFORM OTHER FUNCTION
00D6                    K1T:                            ; ASCII READ
00D6 FB                         STI                     ; INTERRUPTS BACK ON DURING LOOP
00D7 90                         NOP                     ; ALLOW AN INTERRUPT TO OCCUR
00D8 FA                 K1U:    CLI                     ; INTERRUPTS BACK OFF
00D9 8B 1E 001A R               MOV     BX,@BUFFER_HEAD ; GET POINTER TO HEAD OF BUFFER
00DD 3B 1E 001C R               CMP     BX,@BUFFER_TAIL ; TEST END OF BUFFER
00E1 53                         PUSH    BX              ; SAVE ADDRESS
00E2 9C                         PUSHF                   ; SAVE FLAG
00E3 E8 06D8 R                  CALL    MAKE_LED        ; GO GET MODE INDICATOR DATA BYTE
00E6 8A 1E 0097 R               MOV     BL,@KB_FLAG_2   ; GET PREVIOUS BITS
00EA 32 D8                      XOR     BL,AL           ; SEE IF ANY DIFFERENT
00EC 80 E3 07                   AND     BL,07H          ; ISOLATE INDICATOR BITS
00EF 74 04                      JZ      K1V             ; IF NO CHANGE BYPASS UPDATE

00F1 E8 069A R                  CALL    SND_LED1        ; GO TURN ON MODE INDICATORS
00F4 FA                         CLI                     ; DISABLE INTERRUPTS
00F5 9D                 K1V:    POPF                    ; RESTORE FLAGS
00F6 5B                         POP     BX              ; RESTORE ADDRESS
00F7 74 DD                      JZ      K1T             ; LOOP UNTIL SOMETHING IN BUFFER

00F9 8B 07                      MOV     AX,[BX]         ; GET SCAN CODE AND ASCII CODE
00FB E8 0168 R                  CALL    K4              ; MOVE POINTER TO NEXT POSITION
00FE 89 1E 001A R               MOV     @BUFFER_HEAD,BX ; STORE VALUE IN VARIABLE
0102 C3                         RET                     ; RETURN
0103                    K1S     ENDP


                        ;------ READ THE KEY TO SEE IF ONE IS PRESENT -------------------------

0103                    K2S     PROC    NEAR
0103 FA                         CLI                     ; INTERRUPTS OFF
0104 8B 1E 001A R               MOV     BX,@BUFFER_HEAD ; GET HEAD POINTER
0108 3B 1E 001C R               CMP     BX,@BUFFER_TAIL ; IF EQUAL (Z=1) THEN NOTHING THERE
010C 8B 07                      MOV     AX,[BX]
010E 9C                         PUSHF                   ; SAVE FLAGS

010F 50                         PUSH    AX              ; SAVE CODE
0110 E8 06D8 R                  CALL    MAKE_LED        ; GO GET MODE INDICATOR DATA BYTE
0113 8A 1E 0097 R               MOV     BL,@KB_FLAG_2   ; GET PREVIOUS BITS
0117 32 D8                      XOR     BL,AL           ; SEE IF ANY DIFFERENT
0119 80 E3 07                   AND     BL,07H          ; ISOLATE INDICATOR BITS
011C 74 03                      JZ      K2T             ; IF NO CHANGE BYPASS UPDATE

011E E8 0687 R                  CALL    SND_LED         ; GO TURN ON MODE INDICATORS
0121 58                 K2T:    POP     AX              ; RESTORE CODE
0122 9D                         POPF                    ; RESTORE FLAGS
0123 FB                         STI                     ; INTERRUPTS BACK ON
0124 C3                         RET                     ; RETURN
0125                    K2S     ENDP


                        ;------ ROUTINE TO TRANSLATE SCAN CODE PAIRS FOR EXTENDED CALLS

0125                    K10_E_XLAT:
0125 3C F0                      CMP     AL,0F0h         ; IS IT ONE OF THE FILL-INs?
0127 75 06                      JNE     K10_E_RET       ;  NO, PASS IT ON
0129 0A E4                      OR      AH,AH           ; AH = 0 IS SPECIAL CASE
012B 74 02                      JZ      K10_E_RET       ;  PASS THIS ON UNCHANGED
012D 32 C0                      XOR     AL,AL           ;  OTHERWISE SET AL = 0
012F                    K10_E_RET:
012F C3                         RET                     ; GO BACK


                        ;------ ROUTINE TO TRANSLATE SCAN CODE PAIRS FOR STANDARD CALLS

0130                    K10_S_XLAT:
0130 80 FC E0                   CMP     AH,0E0h         ; IS IT KEYPAD ENTER OR / ?
0133 75 12                      JNE     K10_S2          ;  NO, CONTINUE
0135 3C 0D                      CMP     AL,0Dh          ; KEYPAD ENTER CODE?
0137 74 09                      JE      K10_S1          ;  YES, MASSAGE A BIT
0139 3C 0A                      CMP     AL,0Ah          ; CTRL KEYPAD ENTER CODE?
013B 74 05                      JE      K10_S1          ;  YES, MASSAGE THE SAME
013D B4 35                      MOV     AH,35h          ;  NO, MUST BE KEYPAD /
013F EB 23 90                   JMP     K10_USE         ; GIVE TO CALLER
0142 B4 1C              K10_S1: MOV     AH,1Ch          ; CONVERT TO COMPATIBLE OUTPUT
0144 EB 1E 90                   JMP     K10_USE         ; GIVE TO CALLER

0147 80 FC 84           K10_S2: CMP     AH,84h          ; IS IT ONE OF THE EXTENDED ONES?
014A 77 1A                      JA      K10_DIS         ;  YES, THROW AWAY AND GET ANOTHER CHAR

014C 3C F0                      CMP     AL,0F0h         ; IS IT ONE OF THE FILL-INs?
014E 75 07                      JNE     K10_S3          ;  NO, TRY LAST TEST
0150 0A E4                      OR      AH,AH           ; AH = 0 IS SPECIAL CASE
0152 74 10                      JZ      K10_USE         ;  PASS THIS ON UNCHANGED
0154 EB 10 90                   JMP     K10_DIS         ;  THROW AWAY THE REST

0157 3C E0              K10_S3: CMP     AL,0E0h         ; IS IT AN EXTENSION OF A PREVIOUS ONE?
0159 75 09                      JNE     K10_USE         ;  NO, MUST BE A STANDARD CODE
015B 0A E4                      OR      AH,AH           ; AH = 0 IS SPECIAL CASE
015D 74 05                      JZ      K10_USE         ; JUMP IF AH = 0
015F 32 C0                      XOR     AL,AL           ; CONVERT TO COMPATIBLE OUTPUT
0161 EB 01 90                   JMP     K10_USE         ; PASS IT ON TO CALLER

0164                    K10_USE:
0164 F8                         CLC                     ; CLEAR CARRY TO INDICATE GOOD CODE
0165 C3                         RET                     ; RETURN
0166                    K10_DIS:
0166 F9                         STC                     ; SET CARRY TO INDICATE DISCARD CODE
0167 C3                         RET                     ; RETURN
```

```
                        PAGE
                        ;--------------------------------------------------------------------
                        ;       INCREMENT BUFFER POINTER ROUTINE                            -
                        ;--------------------------------------------------------------------

0168                    K4      PROC    NEAR
0168 43                         INC     BX                      ; MOVE TO NEXT WORD IN LIST
0169 43                         INC     BX
                        ;
016A 3B 1E 0082 R               CMP     BX,@BUFFER_END          ; AT END OF BUFFER?
016E 75 04                      JNE     K5                      ; NO, CONTINUE
0170 8B 1E 0080 R               MOV     BX,@BUFFER_START        ; YES, RESET TO BUFFER BEGINNING
0174 C3                 K5:     RET
0175                    K4      ENDP

                        ;--- HARDWARE INT  09 H -- ( IRQ LEVEL  1 ) -----------------------------
                        ;                                                                      :
                        ;       KEYBOARD INTERRUPT ROUTINE                                     :
                        ;                                                                      :
                        ;--------------------------------------------------------------------

0175                    KB_INT_1 PROC   FAR
0175 FB                         STI                             ; ENABLE INTERRUPTS
0176 55                         PUSH    BP
0177 50                         PUSH    AX
0178 53                         PUSH    BX
0179 51                         PUSH    CX
017A 52                         PUSH    DX
017B 56                         PUSH    SI
017C 57                         PUSH    DI
017D 1E                         PUSH    DS
017E 06                         PUSH    ES
017F FC                         CLD                             ; FORWARD DIRECTION
0180 E8 0000 E                  CALL    DDS                     ; SET UP ADDRESSING
                        ;
                        ;-----  WAIT FOR KEYBOARD DISABLE COMMAND TO BE ACCEPTED
                        ;
0183 B0 AD                      MOV     AL,DIS_KBD              ; DISABLE THE KEYBOARD COMMAND
0185 E8 063C R                  CALL    SHIP_IT                 ; EXECUTE DISABLE
0188 FA                         CLI                             ; DISABLE INTERRUPTS
0189 2B C9                      SUB     CX,CX                   ; SET MAXIMUM TIMEOUT
018B                    KB_INT_01:
018B E4 64                      IN      AL,STATUS_PORT          ; READ ADAPTER STATUS
018D A8 02                      TEST    AL,INPT_BUF_FULL        ; CHECK INPUT BUFFER FULL STATUS BIT
018F E0 FA                      LOOPNZ  KB_INT_01               ; WAIT FOR COMMAND TO BE ACCEPTED
                        ;
                        ;-----  READ CHARACTER FROM KEYBOARD INTERFACE
                        ;
0191 E4 60                      IN      AL,PORT_A               ; READ IN THE CHARACTER
                        ;
                        ;-----  SYSTEM HOOK  INT 15H - FUNCTION 4FH  (ON HARDWARE INTERRUPT LEVEL 9H)
                        ;
0193 B4 4F                      MOV     AH,04FH                 ; SYSTEM INTERCEPT - KEY CODE FUNCTION
0195 F9                         STC                             ; SET CY= 1 (IN CASE OF IRET)
0196 CD 15                      INT     15H                     ; CASSETTE CALL   (AL)= KEY SCAN CODE
                                                                ;  RETURNS CY= 1 FOR INVALID FUNCTION
0198 72 03                      JC      KB_INT_02               ; CONTINUE IF CARRY FLAG SET ((AL)=CODE)
019A E9 03A0 R                  JMP     K26                     ; EXIT IF SYSTEM HANDLED SCAN CODE
                                                                ;  EXIT HANDLES HARDWARE EOI AND ENABLE
                        ;
                        ;------ CHECK FOR A RESEND COMMAND TO KEYBOARD
                        ;
019D                    KB_INT_02:                              ;       (AL)= SCAN CODE
019D FB                         STI                             ; ENABLE INTERRUPTS AGAIN
019E 3C FE                      CMP     AL,KB_RESEND            ; IS THE INPUT A RESEND
01A0 74 0D                      JE      KB_INT_4                ; GO IF RESEND
                        ;
                        ;------ CHECK FOR RESPONSE TO A COMMAND TO KEYBOARD
                        ;
01A2 3C FA                      CMP     AL,KB_ACK               ; IS THE INPUT AN ACKNOWLEDGE
01A4 75 12                      JNZ     KB_INT_2                ; GO IF NOT
                        ;
                        ;------ A COMMAND TO THE KEYBOARD WAS ISSUED
                        ;
01A6 FA                         CLI                             ; DISABLE INTERRUPTS
01A7 80 0E 0097 R 10            OR      @KB_FLAG_2,KB_FA        ; INDICATE ACK RECEIVED
01AC E9 03A0 R                  JMP     K26                     ; RETURN IF NOT (ACK RETURNED FOR DATA)
                        ;
                        ;------ RESEND THE LAST BYTE
                        ;
01AF                    KB_INT_4:
01AF FA                         CLI                             ; DISABLE INTERRUPTS
01B0 80 0E 0097 R 20            OR      @KB_FLAG_2,KB_FE        ; INDICATE RESEND RECEIVED
01B5 E9 03A0 R                  JMP     K26                     ; RETURN IF NOT (ACK RETURNED FOR DATA)
                        ;
                        ;------ UPDATE MODE INDICATORS IF CHANGE IN STATE
                        ;
01B8                    KB_INT_2:
01B8 50                         PUSH    AX                      ; SAVE DATA IN
01B9 E8 06D8 R                  CALL    MAKE_LED                ; GO GET MODE INDICATOR DATA BYTE
01BC 8A 1E 0097 R               MOV     BL,@KB_FLAG_2           ; GET PREVIOUS BITS
01C0 32 D8                      XOR     BL,AL                   ; SEE IF ANY DIFFERENT
01C2 80 E3 07                   AND     BL,KB_LEDS              ; ISOLATE INDICATOR BITS
01C5 74 03                      JZ      UP0                     ; IF NO CHANGE BYPASS UPDATE
01C7 E8 0687 R                  CALL    SND_LED                 ; GO TURN ON MODE INDICATORS
01CA 58                 UP0:    POP     AX                      ; RESTORE DATA IN
```

```
                        PAGE
                        ;--------------------------------------------------------------------------
                        ;                START OF KEY PROCESSING                                 -
                        ;--------------------------------------------------------------------------

01CB 8A E0                      MOV     AH,AL                   ; SAVE SCAN CODE IN AH ALSO

                        ;------ TEST FOR OVERRUN SCAN CODE FROM KEYBOARD

01CD 3C FF                      CMP     AL,KB_OVER_RUN          ; IS THIS AN OVERRUN CHAR?
01CF 75 03                      JNZ     K16                     ; NO, TEST FOR SHIFT KEY
01D1 E9 062D R                  JMP     K62                     ; BUFFER_FULL_BEEP

01D4 0E                 K16:    PUSH    CS
01D5 07                         POP     ES                      ; ESTABLISH ADDRESS OF TABLES
01D6 8A 3E 0096 R               MOV     BH,@KB_FLAG_3           ; LOAD FLAGS FOR TESTING

                        ;------ TEST TO SEE IF A READ_ID IS IN PROGRESS

01DA F6 C7 C0                   TEST    BH,RD_ID+LC_AB          ; ARE WE DOING A READ ID?
01DD 74 34                      JZ      NOT_ID                  ; CONTINUE IF NOT
01DF 79 10                      JNS     TST_ID_2                ; IS THE RD_ID FLAG ON?
01E1 3C AB                      CMP     AL,TD_1                 ; IS THIS THE 1ST ID CHARACTER?
01E3 75 05                      JNE     RST_RD_ID
01E5 80 0E 0096 R 40            OR      @KB_FLAG_3,LC_AB        ; INDICATE 1ST ID WAS OK
01EA                    RST_RD_ID:
01EA 80 26 0096 R 7F            AND     @KB_FLAG_3,NOT RD_ID    ; RESET THE READ ID FLAG
01EF EB 1F                      JMP     SHORT ID_EX             ; AND EXIT

01F1                    TST_ID_2:
01F1 80 26 0096 R BF            AND     @KB_FLAG_3,NOT LC_AB    ; RESET FLAG
01F6 3C 54                      CMP     AL,TD_2A                ; IS THIS THE 2ND ID CHARACTER?
01F8 74 11                      JE      KX_BIT                  ;  JUMP IF SO
01FA 3C 41                      CMP     AL,TD_2                 ; IS THIS THE 2ND ID CHARACTER?
01FC 75 12                      JNE     ID_EX                   ;  LEAVE IF NOT

                        ;------ A READ ID SAID THAT IT WAS ENHANCED KEYBOARD

01FE F6 C7 20                   TEST    BH,SET_NUM_LK           ; SHOULD WE SET NUM LOCK?
0201 74 08                      JZ      KX_BIT                  ; EXIT IF NOT
0203 80 0E 0017 R 20            OR      @KB_FLAG,NUM_STATE      ; FORCE NUM LOCK ON
0208 E8 0687 R                  CALL    SND_LED                 ; GO SET THE NUM LOCK INDICATOR
020B 80 0E 0096 R 10    KX_BIT: OR      @KB_FLAG_3,KBX          ; INDICATE ENHANCED KEYBOARD WAS FOUND
0210 E9 03A0 R          ID_EX:  JMP     K26                     ; EXIT


0213                    NOT_ID:
0213 3C E0                      CMP     AL,MC_E0                ; IS THIS THE GENERAL MARKER CODE?
0215 75 07                      JNE     TEST_E1                 ;
0217 80 0E 0096 R 12            OR      @KB_FLAG_3,LC_E0+KBX    ; SET FLAG BIT, SET KBX, AND
021C EB 09                      JMP     SHORT EXIT              ; THROW AWAY THIS CODE

021E                    TEST_E1:
021E 3C E1                      CMP     AL,MC_E1                ; IS THIS THE PAUSE KEY?
0220 75 08                      JNE     NOT_HC                  ;
0222 80 0E 0096 R 11            OR      @KB_FLAG_3,LC_E1+KBX    ; SET FLAG, PAUSE KEY MARKER CODE
0227 E9 03A5 R          EXIT:   JMP     K26A                    ; THROW AWAY THIS CODE


022A                    NOT_HC:
022A 24 7F                      AND     AL,07FH                 ; TURN OFF THE BREAK BIT
022C F6 C7 02                   TEST    BH,LC_E0                ; LAST CODE THE E0 MARKER CODE?
022F 74 0C                      JZ      NOT_LC_E0               ; JUMP IF NOT

0231 B9 0002                    MOV     CX,2                    ; LENGTH OF SEARCH
0234 BF 0006 E                  MOV     DI,OFFSET K6+6          ; IS THIS A SHIFT KEY?
0237 F2/ AE                     REPNE   SCASB                   ; CHECK IT
0239 75 65                      JNE     K16A                    ; NO, CONTINUE KEY PROCESSING
023B EB 49                      JMP     SHORT K16B              ; YES, THROW AWAY & RESET FLAG


023D                    NOT_LC_E0:
023D F6 C7 01                   TEST    BH,LC_E1                ; LAST CODE THE E1 MARKER CODE?
0240 74 1D                      JZ      T_SYS_KEY               ; JUMP IF NOT

0242 B9 0004                    MOV     CX,4                    ; LENGTH OF SEARCH
0245 BF 0004 E                  MOV     DI,OFFSET K6+4          ; IS THIS AN ALT, CTL, OR SHIFT?
0248 F2/ AE                     REPNE   SCASB                   ; CHECK IT
024A 74 DB                      JE      EXIT                    ; THROW AWAY IF SO

024C 3C 45                      CMP     AL,NUM_KEY              ; IS IT THE PAUSE KEY?
024E 75 36                      JNE     K16B                    ; NO, THROW AWAY & RESET FLAG
0250 F6 C4 80                   TEST    AH,80H                  ; YES, IS IT THE BREAK OF THE KEY?
0253 75 31                      JNZ     K16B                    ;  YES, THROW THIS AWAY, TOO
0255 F6 06 0018 R 08            TEST    @KB_FLAG_1,HOLD_STATE   ;  NO, ARE WE PAUSED ALREADY?
025A 75 2A                      JNZ     K16B                    ;   YES, THROW AWAY
025C E9 04DB R                  JMP     K39P                    ;  NO, THIS IS THE REAL PAUSE STATE


                        ;------ TEST FOR SYSTEM KEY

025F                    T_SYS_KEY:
025F 3C 54                      CMP     AL,SYS_KEY              ; IS IT THE SYSTEM KEY?
0261 75 3D                      JNE     K16A                    ; CONTINUE IF NOT

0263 F6 C4 80                   TEST    AH,080H                 ; CHECK IF THIS A BREAK CODE
0266 75 21                      JNZ     K16C                    ; DON'T TOUCH SYSTEM INDICATOR IF TRUE

0268 F6 06 0018 R 04            TEST    @KB_FLAG_1,SYS_SHIFT    ; SEE IF IN SYSTEM KEY HELD DOWN
026D 75 17                      JNZ     K16B                    ; IF YES, DON'T PROCESS SYSTEM INDICATOR

026F 80 0E 0018 R 04            OR      @KB_FLAG_1,SYS_SHIFT    ; INDICATE SYSTEM KEY DEPRESSED
0274 B0 20                      MOV     AL,EOI                  ; END OF INTERRUPT COMMAND
0276 E6 20                      OUT     020H,AL                 ; SEND COMMAND TO INTERRUPT CONTROL PORT
                                                                ; INTERRUPT-RETURN-NO-EOI
0278 B0 AE                      MOV     AL,ENA_KBD              ; INSURE KEYBOARD IS ENABLED
027A E8 063C R                  CALL    SHIP_IT                 ; EXECUTE ENABLE
027D B8 8500                    MOV     AX,08500H               ; FUNCTION VALUE FOR MAKE OF SYSTEM KEY
0280 FB                         STI                             ; MAKE SURE INTERRUPTS ENABLED
0281 CD 15                      INT     15H                     ; USER INTERRUPT
0283 E9 03AF R                  JMP     K27A                    ; END PROCESSING


0286 E9 03A0 R          K16B:   JMP     K26                     ; IGNORE SYSTEM KEY


0289 80 26 0018 R FB    K16C:   AND     @KB_FLAG_1,NOT SYS_SHIFT ;TURN OFF SHIFT KEY HELD DOWN
028E B0 20                      MOV     AL,EOI                  ; END OF INTERRUPT COMMAND
0290 E6 20                      OUT     020H,AL                 ; SEND COMMAND TO INTERRUPT CONTROL PORT
```

```
                                                ; INTERRUPT-RETURN-NO-EOI
0292 B0 AE                      MOV     AL,ENA_KBD      ; INSURE KEYBOARD IS ENABLED
0294 E8 063C R                  CALL    SHIP_IT         ; EXECUTE ENABLE
0297 B8 8501                    MOV     AX,08501H       ; FUNCTION VALUE FOR BREAK OF SYSTEM KEY
029A FB                         STI                     ; MAKE SURE INTERRUPTS ENABLED
029B CD 15                      INT     15H             ; USER INTERRUPT
029D E9 03AF R                  JMP     K27A            ; IGNORE SYSTEM KEY

                        ;------ TEST FOR SHIFT KEYS

02A0 8A 1E 0017 R       K16A:   MOV     BL,@KB_FLAG     ; PUT STATE FLAGS IN BL
02A4 BF 0000 E                  MOV     DI,OFFSET K6    ; SHIFT KEY TABLE
02A7 B9 0000 E                  MOV     CX,OFFSET K6L   ; LENGTH
02AA F2/ AE                     REPNE   SCASB           ; LOOK THROUGH THE TABLE FOR A MATCH
02AC 8A C4                      MOV     AL,AH           ; RECOVER SCAN CODE
02AE 74 03                      JE      K17             ; JUMP IF MATCH FOUND
02B0 E9 038C R                  JMP     K25             ; IF NO MATCH, THEN SHIFT NOT FOUND

                        ;------ SHIFT KEY FOUND

02B3 81 EF 0001 E       K17:    SUB     DI,OFFSET K6+1  ; ADJUST PTR TO SCAN CODE MTCH
02B7 2E: 8A A5 0000 E           MOV     AH,CS:K7[DI]    ; GET MASK INTO AH
02BC B1 02                      MOV     CL,2            ; SET UP COUNT FOR FLAG SHIFTS
02BE A8 80                      TEST    AL,80H          ; TEST FOR BREAK KEY
02C0 74 03                      JZ      K17C
02C2 EB 78 90                   JMP     K23             ; JUMP IF BREAK

                        ;------ SHIFT MAKE FOUND, DETERMINE SET OR TOGGLE

02C5 80 FC 10           K17C:   CMP     AH,SCROLL_SHIFT
02C8 73 21                      JAE     K18             ; IF SCROLL SHIFT OR ABOVE, TOGGLE KEY

                        ;------ PLAIN SHIFT KEY, SET SHIFT ON

02CA 08 26 0017 R               OR      @KB_FLAG,AH     ; TURN ON SHIFT BIT
02CE F6 C4 0C                   TEST    AH,CTL_SHIFT+ALT_SHIFT  ; IS IT ALT OR CTRL?
02D1 75 03                      JNZ     K17D            ;  YES, MORE FLAGS TO SET
02D3 E9 03A0 R                  JMP     K26             ;  NO, INTERRUPT_RETURN
02D6 F6 C7 02           K17D:   TEST    BH,LC_E0        ; IS THIS ONE OF THE NEW KEYS?
02D9 74 07                      JZ      K17E            ;  NO, JUMP
02DB 08 26 0096 R               OR      @KB_FLAG_3,AH   ; SET BITS FOR RIGHT CTRL, ALT
02DF E9 03A0 R                  JMP     K26             ; INTERRUPT_RETURN
02E2 D2 EC              K17E:   SHR     AH,CL           ; MOVE FLAG BITS TWO POSITIONS
02E4 08 26 0018 R               OR      @KB_FLAG_1,AH   ; SET BITS FOR LEFT CTRL, ALT
02E8 E9 03A0 R                  JMP     K26             ; INTERRUPT_RETURN

                        ;------ TOGGLED SHIFT KEY, TEST FOR 1ST MAKE OR NOT

02EB                    K18:                            ; SHIFT-TOGGLE
02EB F6 C3 04                   TEST    BL,CTL_SHIFT    ; CHECK CTL SHIFT STATE
02EE 74 03                      JZ      K18A            ; JUMP IF NOT CTL STATE
02F0 E9 038C R                  JMP     K25             ; JUMP IF CTL STATE
02F3 3C 52              K18A:   CMP     AL,INS_KEY      ; CHECK FOR INSERT KEY
02F5 75 21                      JNE     K22             ; JUMP IF NOT INSERT KEY
02F7 F6 C3 08                   TEST    BL,ALT_SHIFT    ; CHECK FOR ALTERNATE SHIFT
02FA 74 03                      JZ      K18B            ; JUMP IF NOT ALTERNATE SHIFT
02FC E9 038C R                  JMP     K25             ; JUMP IF ALTERNATE SHIFT
02FF F6 C7 02           K18B:   TEST    BH,LC_E0        ; IS THIS THE NEW INSERT KEY?
0302 75 14                      JNZ     K22             ; YES, THIS ONE'S NEVER A "0"
0304 F6 C3 20           K19:    TEST    BL,NUM_STATE    ; CHECK FOR BASE STATE
0307 75 0A                      JNZ     K21             ; JUMP IF NUM LOCK IS ON
0309 F6 C3 03                   TEST    BL,LEFT_SHIFT+RIGHT_SHIFT  ; TEST FOR SHIFT STATE
030C 74 0A                      JZ      K22             ;  JUMP IF BASE STATE
030E 8A E0              K20:    MOV     AH,AL           ; PUT SCAN CODE BACK IN AH
0310 EB 7A 90                   JMP     K25             ; NUMERAL "0", STNDRD. PROCESSING

0313 F6 C3 03           K21:    TEST    BL,LEFT_SHIFT+RIGHT_SHIFT  ; MIGHT BE NUMERIC
0316 74 F6                      JZ      K20             ;  IS NUMERIC, STD. PROC.

0318                    K22:                            ; SHIFT TOGGLE KEY HIT; PROCESS IT
0318 84 26 0018 R               TEST    AH,@KB_FLAG_1   ; IS KEY ALREADY DEPRESSED?
031C 74 03                      JZ      K22A
031E E9 03A0 R                  JMP     K26             ; JUMP IF KEY ALREADY DEPRESSED
0321 08 26 0018 R       K22A:   OR      @KB_FLAG_1,AH   ; INDICATE THAT THE KEY IS DEPRESSED
0325 30 26 0017 R               XOR     @KB_FLAG,AH     ; TOGGLE THE SHIFT STATE

                        ;------ TOGGLE LED IF CAPS, NUM, OR SCROLL KEY DEPRESSED

0329 F6 C4 70                   TEST    AH,CAPS_SHIFT+NUM_SHIFT+SCROLL_SHIFT ; SHIFT TOGGLE?
032C 74 05                      JZ      K22B            ; GO IF NOT
032E 50                         PUSH    AX              ; SAVE SCAN CODE AND SHIFT MASK
032F E8 0687 R                  CALL    SND_LED         ; GO TURN MODE INDICATORS ON
0332 58                         POP     AX              ; RESTORE SCAN CODE

0333 3C 52              K22B:   CMP     AL,INS_KEY      ; TEST FOR 1ST MAKE OF INSERT KEY
0335 75 69                      JNE     K26             ; JUMP IF NOT INSERT KEY
0337 8A E0                      MOV     AH,AL           ; SCAN CODE IN BOTH HALVES OF AX
0339 EB 7F 90                   JMP     K28             ; FLAGS UPDATED, PROC. FOR BUFFER

                        ;------ BREAK SHIFT FOUND

033C                    K23:                            ; BREAK-SHIFT-FOUND
033C 80 FC 10                   CMP     AH,SCROLL_SHIFT ; IS THIS A TOGGLE KEY?
033F F6 D4                      NOT     AH              ; INVERT MASK
0341 73 43                      JAE     K24             ; YES, HANDLE BREAK TOGGLE
0343 20 26 0017 R               AND     @KB_FLAG,AH     ; TURN OFF SHIFT BIT
0347 80 FC FB                   CMP     AH,NOT CTL_SHIFT ; IS THIS ALT OR CTL?
034A 77 26                      JA      K23D            ;  NO, ALL DONE

034C F6 C7 02                   TEST    BH,LC_E0        ; 2ND ALT OR CTL?
034F 74 06                      JZ      K23A            ; NO, HANDLE NORMALLY
0351 20 26 0096 R               AND     @KB_FLAG_3,AH   ; RESET BIT FOR RIGHT ALT OR CTL
0355 EB 06                      JMP     SHORT K23B      ; CONTINUE
0357 D2 FC              K23A:   SAR     AH,CL           ; MOVE THE MASK BIT TWO POSITIONS
0359 20 26 0018 R               AND     @KB_FLAG_1,AH   ; RESET BIT FOR LEFT ALT OR CTL
035D 8A E0              K23B:   MOV     AH,AL           ; SAVE SCAN CODE
035F A0 0096 R                  MOV     AL,@KB_FLAG_3   ; GET RIGHT ALT & CTRL FLAGS
0362 D2 E8                      SHR     AL,CL           ; MOVE TO BITS 1 & 0
0364 0A 06 0018 R               OR      AL,@KB_FLAG_1   ; PUT IN LEFT ALT & CTL FLAGS
0368 D2 E0                      SHL     AL,CL           ; MOVE BACK TO BITS 3 & 2
036A 24 0C                      AND     AL,ALT_SHIFT+CTL_SHIFT  ; FILTER OUT OTHER GARBAGE
036C 08 06 0017 R               OR      @KB_FLAG,AL     ; PUT RESULT IN THE REAL FLAGS
0370 8A C4                      MOV     AL,AH           ; RECOVER SAVED SCAN CODE
```

```
0372 3C B8              K23D:   CMP     AL,ALT_KEY+80H          ; IS THIS ALTERNATE SHIFT RELEASE
0374 75 2A                      JNE     K26                     ; INTERRUPT_RETURN

                        ;------ ALTERNATE SHIFT KEY RELEASED, GET THE VALUE INTO BUFFER

0376 A0 0019 R                  MOV     AL,@ALT_INPUT
0379 B4 00                      MOV     AH,0                    ; SCAN CODE OF 0
037B 88 26 0019 R               MOV     @ALT_INPUT,AH           ; ZERO OUT THE FIELD
037F 3C 00                      CMP     AL,0                    ; WAS THE INPUT = 0?
0381 74 1D                      JE      K26                     ; INTERRUPT_RETURN
0383 E9 0601 R                  JMP     K61                     ; IT WASN'T, SO PUT IN BUFFER

0386                    K24:                                    ; BREAK-TOGGLE
0386 20 26 0018 R               AND     @KB_FLAG_1,AH           ; INDICATE NO LONGER DEPRESSED
038A EB 14                      JMP     SHORT K26               ; INTERRUPT_RETURN

                        ;------ TEST FOR HOLD STATE
                                                                ; AL, AH = SCAN CODE
038C                    K25:                                    ; NO-SHIFT-FOUND
038C 3C 80                      CMP     AL,80H                  ; TEST FOR BREAK KEY
038E 73 10                      JAE     K26                     ; NOTHING FOR BREAK CHARS FROM HERE ON
0390 F6 06 0018 R 08            TEST    @KB_FLAG_1,HOLD_STATE   ; ARE WE IN HOLD STATE
0395 74 23                      JZ      K28                     ; BRANCH AROUND TEST IF NOT
0397 3C 45                      CMP     AL,NUM_KEY
0399 74 05                      JE      K26                     ; CAN'T END HOLD ON NUM_LOCK
039B 80 26 0018 R F7            AND     @KB_FLAG_1,NOT HOLD_STATE  ; TURN OFF THE HOLD STATE BIT

03A0                    K26:
03A0 80 26 0096 R FC            AND     @KB_FLAG_3,NOT LC_E0+LC_E1 ; RESET LAST CHAR H.C. FLAG

03A5                    K26A:                                   ; INTERRUPT-RETURN
03A5 FA                         CLI                             ; TURN OFF INTERRUPTS
03A6 B0 20                      MOV     AL,EOI                  ; END OF INTERRUPT COMMAND
03A8 E6 20                      OUT     020H,AL                 ; SEND COMMAND TO INTERRUPT CONTROL PORT

03AA                    K27:                                    ; INTERRUPT-RETURN-NO-EOI
03AA B0 AE                      MOV     AL,ENA_KBD              ; INSURE KEYBOARD IS ENABLED
03AC E8 063C R                  CALL    SHIP_IT                 ; EXECUTE ENABLE

03AF FA                 K27A:   CLI                             ; DISABLE INTERRUPTS
03B0 07                         POP     ES                      ; RESTORE REGISTERS
03B1 1F                         POP     DS
03B2 5F                         POP     DI
03B3 5E                         POP     SI
03B4 5A                         POP     DX
03B5 59                         POP     CX
03B6 5B                         POP     BX
03B7 58                         POP     AX
03B8 5D                         POP     BP
03B9 CF                         IRET                            ; RETURN
```

```
                        PAGE
                        ;------ NOT IN HOLD STATE
                                                        ; AL, AH = SCAN CODE (ALL MAKES)
03BA                    K28:                            ; NO-HOLD-STATE
03BA 3C 58                      CMP     AL,88           ; TEST FOR OUT-OF-RANGE SCAN CODES
03BC 77 E2                      JA      K26             ; IGNORE IF OUT-OF-RANGE
                        ;
03BE F6 C3 08                   TEST    BL,ALT_SHIFT    ; ARE WE IN ALTERNATE SHIFT?
03C1 74 0C                      JZ      K28A            ; JUMP IF NOT ALTERNATE
                        ;
03C3 F6 C7 10                   TEST    BH,KBX          ; IS THIS THE ENHANCED KEYBOARD?
03C6 74 0A                      JZ      K29             ; NO, ALT STATE IS REAL
                        ;
03C8 F6 06 0018 R 04            TEST    @KB_FLAG_1,SYS_SHIFT ; YES, IS SYSREQ KEY DOWN?
03CD 74 03                      JZ      K29             ;  NO, ALT STATE IS REAL
03CF E9 04A3 R          K28A:   JMP     K38             ;  YES, THIS IS PHONY ALT STATE
                        ;                               ;       DUE TO PRESSING SYSREQ
                        ;------ TEST FOR RESET KEY SEQUENCE (CTL ALT DEL)
03D2                    K29:                            ; TEST-RESET
03D2 F6 C3 04                   TEST    BL,CTL_SHIFT    ; ARE WE IN CONTROL SHIFT ALSO?
03D5 74 31                      JZ      K31             ; NO_RESET
03D7 3C 53                      CMP     AL,DEL_KEY      ; SHIFT STATE IS THERE, TEST KEY
03D9 75 2D                      JNE     K31             ; NO_RESET,  IGNORE
                        ;------ CTL-ALT-DEL HAS BEEN FOUND, DO I/O CLEANUP
03DB C7 06 0072 R 1234          MOV     @RESET_FLAG,1234H ; SET FLAG FOR RESET FUNCTION
03E1 E9 0000 E                  JMP     START_T         ; JUMP TO POWER ON DIAGNOSTICS
                        ;---------------------- TABLES FOR ALT CASE ------------------------
                        ;------ ALT-INPUT-TABLE
03E4                    K30     LABEL   BYTE
03E4 52 4F 50 51 4B             DB      82,79,80,81,75
03E9 4C 4D 47 48 49             DB      76,77,71,72,73  ; 10 NUMBERS ON KEYPAD
                        ;------ SUPER-SHIFT-TABLE
03EE 10 11 12 13 14 15          DB      16,17,18,19,20,21 ; A-Z TYPEWRITER CHARS
03F4 16 17 18 19 1E 1F          DB      22,23,24,25,30,31
03FA 20 21 22 23 24 25          DB      32,33,34,35,36,37
0400 26 2C 2D 2E 2F 30          DB      38,44,45,46,47,48
0406 31 32                      DB      49,50
                        ;------ IN ALTERNATE SHIFT, RESET NOT FOUND
0408                    K31:                            ; NO-RESET
0408 3C 39                      CMP     AL,57           ; TEST FOR SPACE KEY
040A 75 05                      JNE     K311            ; NOT THERE
040C B0 20                      MOV     AL,' '          ; SET SPACE CHAR
040E E9 05F5 R                  JMP     K57             ; BUFFER_FILL
0411                    K311:
0411 3C 0F                      CMP     AL,15           ; TEST FOR TAB KEY
0413 75 06                      JNE     K312            ; NOT THERE
0415 B8 A500                    MOV     AX,0A500h       ; SET SPECIAL CODE FOR ALT-TAB
0418 E9 05F5 R                  JMP     K57             ; BUFFER_FILL
041B                    K312:
041B 3C 4A                      CMP     AL,74           ; TEST FOR KEYPAD -
041D 74 79                      JE      K37B            ; GO PROCESS
041F 3C 4E                      CMP     AL,78           ; TEST FOR KEYPAD +
0421 74 75                      JE      K37B            ; GO PROCESS
                        ;------ LOOK FOR KEY PAD ENTRY
0423                    K32:                            ; ALT-KEY-PAD
0423 BF 03E4 R                  MOV     DI,OFFSET K30   ; ALT-INPUT-TABLE
0426 B9 000A                    MOV     CX,10           ; LOOK FOR ENTRY USING KEYPAD
0429 F2/ AE                     REPNE   SCASB           ; LOOK FOR MATCH
042B 75 18                      JNE     K33             ; NO_ALT_KEYPAD
042D F6 C7 02                   TEST    BH,LC_E0        ; IS THIS ONE OF THE NEW KEYS?
0430 75 6B                      JNZ     K37C            ;  YES, JUMP, NOT NUMPAD KEY
0432 81 EF 03E5 R               SUB     DI,OFFSET K30+1 ; DI NOW HAS ENTRY VALUE
0436 A0 0019 R                  MOV     AL,@ALT_INPUT   ; GET THE CURRENT BYTE
0439 B4 0A                      MOV     AH,10           ; MULTIPLY BY 10
043B F6 E4                      MUL     AH
043D 03 C7                      ADD     AX,DI           ; ADD IN THE LATEST ENTRY
043F A2 0019 R                  MOV     @ALT_INPUT,AL   ; STORE IT AWAY
0442 E9 03A0 R          K32A:   JMP     K26             ; THROW AWAY THAT KEYSTROKE
                        ;------ LOOK FOR SUPERSHIFT ENTRY
0445                    K33:                            ; NO-ALT-KEYPAD
0445 C6 06 0019 R 00            MOV     @ALT_INPUT,0    ; ZERO ANY PREVIOUS ENTRY INTO INPUT
044A B9 001A                    MOV     CX,26           ; DI,ES ALREADY POINTING
044D F2 AE                      REPNE   SCASB           ; LOOK FOR MATCH IN ALPHABET
044F 74 42                      JE      K37A            ; MATCH FOUND, GO FILL THE BUFFER
                        ;------ LOOK FOR TOP ROW OF ALTERNATE SHIFT
0451                    K34:                            ; ALT-TOP-ROW
0451 3C 02                      CMP     AL,2            ; KEY WITH '1' ON IT
0453 72 43                      JB      K37B            ; MUST BE ESCAPE
0455 3C 0D                      CMP     AL,13           ; IS IT IN THE REGION
0457 77 05                      JA      K35             ;  NO, ALT-SOMETHING ELSE
0459 80 C4 76                   ADD     AH,118          ; CONVERT PSEUDO SCAN CODE TO RANGE
045C EB 35                      JMP     SHORT K37A      ; GO FILL THE BUFFER
                        ;------ TRANSLATE ALTERNATE SHIFT PSEUDO SCAN CODES
045E                    K35:                            ; ALT-FUNCTION
045E 3C 57                      CMP     AL,F11_M        ; IS IT F11?
0460 72 09                      JB      K35A            ;  NO, BRANCH
0462 3C 58                      CMP     AL,F12_M        ; IS IT F12?
0464 77 05                      JA      K35A            ;  NO, BRANCH
0466 80 C4 34                   ADD     AH,52           ; CONVERT TO PSEUDO SCAN CODE
0469 EB 28                      JMP     SHORT K37A      ; GO FILL THE BUFFER
                        ;
046B F6 C7 02           K35A:   TEST    BH,LC_E0        ; DO WE HAVE ONE OF THE NEW KEYS?
046E 74 18                      JZ      K37             ;  NO, JUMP
0470 3C 1C                      CMP     AL,28           ; TEST FOR KEYPAD ENTER
0472 75 06                      JNE     K35B            ; NOT THERE
0474 B8 A600                    MOV     AX,0A600h       ; SPECIAL CODE
0477 E9 05F5 R                  JMP     K57             ; BUFFER FILL
047A 3C 53              K35B:   CMP     AL,83           ; TEST FOR DELETE KEY
047C 74 1F                      JE      K37C            ; HANDLE WITH OTHER EDIT KEYS
```

SECTION 5

```
047E 3C 35                      CMP     AL,53                   ; TEST FOR KEYPAD /
0480 75 C0                      JNE     K32A                    ; NOT THERE, NO OTHER E0 SPECIALS
0482 B8 A400                    MOV     AX,0A400h               ; SPECIAL CODE
0485 E9 05F5 R                  JMP     K57                     ; BUFFER FILL

0488 3C 3B              K37:    CMP     AL,59                   ; TEST FOR FUNCTION KEYS (F1)
048A 72 0C                      JB      K37B                    ;  NO FN, HANDLE W/OTHER EXTENDED
048C 3C 44                      CMP     AL,68                   ; IN KEYPAD REGION?
                                                                ; OR NUMLOCK, SCROLLOCK?
048E 77 B2                      JA      K32A                    ; IF SO, IGNORE
0490 80 C4 2D                   ADD     AH,45                   ; CONVERT TO PSEUDO SCAN CODE

0493 B0 00              K37A:   MOV     AL,0                    ; ASCII CODE OF ZERO
0495 E9 05F5 R                  JMP     K57                     ; PUT IT IN THE BUFFER

0498 B0 F0              K37B:   MOV     AL,0F0h                 ; USE SPECIAL ASCII CODE
049A E9 05F5 R                  JMP     K57                     ; PUT IT IN THE BUFFER

049D 04 50              K37C:   ADD     AL,80                   ; CONVERT SCAN CODE (EDIT KEYS)
049F 8A E0                      MOV     AH,AL                   ; (SCAN CODE NOT IN AH FOR INSERT)
04A1 EB F0                      JMP     K37A                    ; PUT IT IN THE BUFFER

                        ;------ NOT IN ALTERNATE SHIFT

04A3                    K38:                                    ; NOT-ALT-SHIFT
                                                                ; BL STILL HAS SHIFT FLAGS
04A3 F6 C3 04                   TEST    BL,CTL_SHIFT            ; ARE WE IN CONTROL SHIFT?
04A6 75 03                      JNZ     K38A                    ;  YES, START PROCESSING
04A8 E9 0535 R                  JMP     K44                     ; NOT-CTL-SHIFT

                        ;------ CONTROL SHIFT, TEST SPECIAL CHARACTERS

                        ;------ TEST FOR BREAK

04AB 3C 46              K38A:   CMP     AL,SCROLL_KEY           ; TEST FOR BREAK
04AD 75 23                      JNE     K39                     ; JUMP, NO-BREAK
04AF F6 C7 10                   TEST    BH,KBX                  ; IS THIS THE ENHANCED KEYBOARD?
04B2 74 05                      JZ      K38B                    ;  NO, BREAK IS VALID
04B4 F6 C7 02                   TEST    BH,LC_E0                ;  YES, WAS LAST CODE AN E0?
04B7 74 19                      JZ      K39                     ;   NO-BREAK, TEST FOR PAUSE

04B9 8B 1E 001A R       K38B:   MOV     BX,@BUFFER_HEAD         ; RESET BUFFER TO EMPTY
04BD 89 1E 001C R               MOV     @BUFFER_TAIL,BX         ;
04C1 C6 06 0071 R 80            MOV     @BIOS_BREAK,80H         ; TURN ON BIOS_BREAK BIT

                        ;-------- ENABLE KEYBOARD

04C6 B0 AE                      MOV     AL,ENA_KBD              ; ENABLE KEYBOARD
04C8 E8 063C R                  CALL    SHIP_IT                 ; EXECUTE ENABLE
04CB CD 1B                      INT     1BH                     ; BREAK INTERRUPT VECTOR
04CD 2B C0                      SUB     AX,AX                   ; PUT OUT DUMMY CHARACTER
04CF E9 05F5 R                  JMP     K57                     ; BUFFER_FILL

                        ;-------- TEST FOR PAUSE

04D2                    K39:                                    ; NO-BREAK
04D2 F6 C7 10                   TEST    BH,KBX                  ; IS THIS THE ENHANCED KEYBOARD?
04D5 75 2A                      JNZ     K41                     ; YES, THEN THIS CAN'T BE PAUSE
04D7 3C 45                      CMP     AL,NUM_KEY              ; LOOK FOR PAUSE KEY
04D9 75 26                      JNE     K41                     ; NO-PAUSE
04DB 80 0E 0018 R 08    K39P:   OR      @KB_FLAG_1,HOLD_STATE   ; TURN ON THE HOLD FLAG

                        ;-------- ENABLE KEYBOARD

04E0 B0 AE                      MOV     AL,ENA_KBD              ; ENABLE KEYBOARD
04E2 E8 063C R                  CALL    SHIP_IT                 ; EXECUTE ENABLE
04E5 B0 20              K39A:   MOV     AL,EOI                  ; END OF INTERRUPT TO CONTROL PORT
04E7 E6 20                      OUT     020H,AL                 ; ALLOW FURTHER KEYSTROKE INTS

                        ;------ DURING PAUSE INTERVAL, TURN CRT BACK ON

04E9 80 3E 0049 R 07            CMP     @CRT_MODE,7             ; IS THIS BLACK AND WHITE CARD
04EE 74 07                      JE      K40                     ; YES, NOTHING TO DO
04F0 BA 03D8                    MOV     DX,03D8H                ; PORT FOR COLOR CARD
04F3 A0 0065 R                  MOV     AL,@CRT_MODE_SET        ; GET THE VALUE OF THE CURRENT MODE
04F6 EE                         OUT     DX,AL                   ; SET THE CRT MODE, SO THAT CRT IS ON
04F7                    K40:                                    ; PAUSE-LOOP
04F7 F6 06 0018 R 08            TEST    @KB_FLAG_1,HOLD_STATE
04FC 75 F9                      JNZ     K40                     ; LOOP UNTIL FLAG TURNED OFF
04FE E9 03AA R                  JMP     K27                     ; INTERRUPT_RETURN_NO_EOI

                        ;------ TEST SPECIAL CASE KEY 55

0501                    K41:                                    ; NO-PAUSE
0501 3C 37                      CMP     AL,55                   ; TEST FOR */PRTSC KEY
0503 75 10                      JNE     K42                     ; NOT-KEY-55
0505 F6 C7 10                   TEST    BH,KBX                  ; IS THIS THE ENHANCED KEYBOARD?
0508 74 05                      JZ      K41A                    ;  NO, CTL-PRTSC IS VALID
050A F6 C7 02                   TEST    BH,LC_E0                ;  YES, WAS LAST CODE AN E0?
050D 74 20                      JZ      K42B                    ;   NO, TRANSLATE TO A FUNCTION
050F B8 7200            K41A:   MOV     AX,114*256              ; START/STOP PRINTING SWITCH
0512 E9 05F5 R                  JMP     K57                     ; BUFFER_FILL

                        ;------ SET UP TO TRANSLATE CONTROL SHIFT

0515                    K42:                                    ; NOT-KEY-55
0515 3C 0F                      CMP     AL,15                   ; IS IT THE TAB KEY?
0517 74 16                      JE      K42B                    ;  YES, XLATE TO FUNCTION CODE
0519 3C 35                      CMP     AL,53                   ; IS IT THE / KEY?
051B 75 0B                      JNE     K42A                    ;  NO, NO MORE SPECIAL CASES
051D F6 C7 02                   TEST    BH,LC_E0                ;  YES, IS IT FROM THE KEYPAD?
0520 74 06                      JZ      K42A                    ;   NO, JUST TRANSLATE
0522 B8 9500                    MOV     AX,9500h                ;   YES, SPECIAL CODE FOR THIS ONE
0525 E9 05F5 R                  JMP     K57                     ;    BUFFER FILL

0528 BB 0000 E          K42A:   MOV     BX,OFFSET K8            ; SET UP TO TRANSLATE CTL
052B 3C 3B                      CMP     AL,59                   ; IS IT IN CHARACTER TABLE?
052D 72 5E                      JB      K45F                    ;  YES, GO TRANSLATE CHAR
052F BB 0000 E          K42B:   MOV     BX,OFFSET K8            ; SET UP TO TRANSLATE CTL
0532 E9 05E4 R                  JMP     K64                     ;  NO, GO TRANSLATE_SCAN

                        ;------ NOT IN CONTROL SHIFT

0535 3C 37              K44:    CMP     AL,55                   ; PRINT SCREEN KEY?
```

```
0537 75 26                      JNE     K45                     ; NOT-PRINT-SCREEN
0539 F6 C7 10                   TEST    BH,KBX                  ; IS THIS ENHANCED KEYBOARD?
053C 74 07                      JZ      K44A                    ; NO, TEST FOR SHIFT STATE
053E F6 C7 02                   TEST    BH,LC_E0                ; YES, LAST CODE A MARKER?
0541 75 07                      JNZ     K44B                    ;  YES, IS PRINT SCREEN
0543 EB 3B                      JMP     SHORT K45C              ;  NO, XLATE TO "*" CHARACTER
0545 F6 C3 03           K44A:   TEST    BL,LEFT_SHIFT+RIGHT_SHIFT ;NOT 101 KBD, SHIFT KEY DOWN?
0548 74 36                      JZ      K45C                    ; NO, XLATE TO "*" CHARACTER

                        ;------ ISSUE INTERRUPT TO PERFORM PRINT SCREEN FUNCTION
054A B0 AE              K44B:   MOV     AL,ENA_KBD              ; INSURE KEYBOARD IS ENABLED
054C E8 063C R                  CALL    SHIP_IT                 ; EXECUTE ENABLE
054F B0 20                      MOV     AL,EOI                  ; END OF CURRENT INTERRUPT
0551 E6 20                      OUT     020H,AL                 ;  SO FURTHER THINGS CAN HAPPEN
0553 55                         PUSH    BP                      ; SAVE POINTER
0554 CD 05                      INT     5H                      ; ISSUE PRINT SCREEN INTERRUPT
0556 5D                         POP     BP                      ; RESTORE POINTER
0557 80 26 0096 R FC            AND     @KB_FLAG_3,NOT LC_E0+LC_E1 ;ZERO OUT THESE FLAGS
055C E9 03AA R                  JMP     K27                     ; GO BACK WITHOUT EOI OCCURRING


                        ;------ HANDLE THE IN-CORE KEYS
055F                    K45:                                    ; NOT-PRINT-SCREEN
055F 3C 3A                      CMP     AL,58                   ; TEST FOR IN-CORE AREA
0561 77 2C                      JA      K46                     ; JUMP IF NOT

0563 3C 35                      CMP     AL,53                   ; IS THIS THE "/" KEY?
0565 75 05                      JNE     K45A                    ;  NO, JUMP
0567 F6 C7 02                   TEST    BH,LC_E0                ; WAS LAST CODE THE MARKER?
056A 75 14                      JNZ     K45C                    ;  YES, TRANSLATE TO CHARACTER

056C B9 001A            K45A:   MOV     CX,26                   ; LENGTH OF SEARCH
056F BF 03EE R                  MOV     DI,OFFSET K30+10        ; POINT TO TABLE OF A-Z CHARS
0572 F2/ AE                     REPNE   SCASB                   ; IS THIS A LETTER KEY?
0574 75 05                      JNE     K45B                    ;  NO, SYMBOL KEY

0576 F6 C3 40                   TEST    BL,CAPS_STATE           ; ARE WE IN CAPS_LOCK?
0579 75 0A                      JNZ     K45D                    ; TEST FOR SURE
057B F6 C3 03           K45B:   TEST    BL,LEFT_SHIFT+RIGHT_SHIFT ; ARE WE IN SHIFT STATE?
057E 75 0A                      JNZ     K45E                    ; YES, UPPERCASE
                                                                ; NO, LOWERCASE
0580 BB 0000 E          K45C:   MOV     BX,OFFSET K10           ; TRANSLATE TO LOWERCASE LETTERS
0583 EB 50                      JMP     SHORT K56
0585                    K45D:                                   ; ALMOST-CAPS-STATE
0585 F6 C3 03                   TEST    BL,LEFT_SHIFT+RIGHT_SHIFT ; CL ON. IS SHIFT ON, TOO?
0588 75 F6                      JNZ     K45C                    ; SHIFTED TEMP OUT OF CAPS STATE
058A BB 0000 E          K45E:   MOV     BX,OFFSET K11           ; TRANSLATE TO UPPERCASE LETTERS
058D EB 46              K45F:   JMP     SHORT K56


                        ;------ TEST FOR KEYS F1 - F10
058F                    K46:                                    ; NOT IN-CORE AREA
058F 3C 44                      CMP     AL,68                   ; TEST FOR F1 - F10
0591 77 02                      JA      K47                     ; JUMP IF NOT
0593 EB 36                      JMP     SHORT K53               ; YES, GO DO FN KEY PROCESS


                        ;------ HANDLE THE NUMERIC PAD KEYS
0595                    K47:                                    ; NOT F1 - F10
0595 3C 53                      CMP     AL,83                   ; TEST FOR NUMPAD KEYS
0597 77 2C                      JA      K52                     ; JUMP IF NOT

                        ;------ KEYPAD KEYS, MUST TEST NUM LOCK FOR DETERMINATION
0599 3C 4A              K48:    CMP     AL,74                   ; SPECIAL CASE FOR MINUS
059B 74 ED                      JE      K45E                    ; GO TRANSLATE
059D 3C 4E                      CMP     AL,78                   ; SPECIAL CASE FOR PLUS
059F 74 E9                      JE      K45E                    ; GO TRANSLATE
05A1 F6 CT 02                   TEST    BH,LC_E0                ; IS THIS ONE OF THE NEW KEYS?
05A4 75 0A                      JNZ     K49                     ;  YES, TRANSLATE TO BASE STATE

05A6 F6 C3 20                   TEST    BL,NUM_STATE            ; ARE WE IN NUM_LOCK?
05A9 75 13                      JNZ     K50                     ; TEST FOR SURE
05AB F6 C3 03                   TEST    BL,LEFT_SHIFT+RIGHT_SHIFT  ; ARE WE IN SHIFT STATE?
05AE 75 13                      JNZ     K51                     ;  IF SHIFTED, REALLY NUM STATE

                        ;------ BASE CASE FOR KEYPAD
05B0 3C 4C              K49:    CMP     AL,76                   ; SPECIAL CASE FOR BASE STATE 5
05B2 75 05                      JNE     K49A                    ; CONTINUE IF NOT KEYPAD 5
05B4 B0 F0                      MOV     AL,0F0h                 ; SPECIAL ASCII CODE
05B6 EB 3D 90                   JMP     K57                     ; BUFFER FILL
05B9 BB 0000 E          K49A:   MOV     BX,OFFSET K10           ; BASE CASE TABLE
05BC EB 26                      JMP     SHORT K64               ; CONVERT TO PSEUDO SCAN


                        ;------ MIGHT BE NUM LOCK, TEST SHIFT STATUS
05BE F6 C3 03           K50:    TEST    BL,LEFT_SHIFT+RIGHT_SHIFT       ;ALMOST-NUM-STATE
05C1 75 ED                      JNZ     K49                     ; SHIFTED TEMP OUT OF NUM STATE
05C3 EB C5              K51:    JMP     SHORT K45E              ; REALLY_NUM_STATE


                        ;------ TEST FOR THE NEW KEY ON WT KEYBOARDS

05C5                    K52:                                    ; NOT A NUMPAD KEY
05C5 3C 56                      CMP     AL,86                   ; IS IT THE NEW WT KEY?
05C7 75 02                      JNE     K53                     ; JUMP IF NOT
05C9 EB B0                      JMP     SHORT K45B              ; HANDLE WITH REST OF LETTER KEYS


                        ;------ MUST BE F11 OR F12
05CB F6 C3 03           K53:    TEST    BL,LEFT_SHIFT+RIGHT_SHIFT ;TEST SHIFT STATE
05CE 74 E0                      JZ      K49                     ; JUMP, LOWERCASE PSEUDO SC'S

05D0 BB 0000 E                  MOV     BX,OFFSET K11           ; UPPER CASE PSEUDO SCAN CODES
05D3 EB 0F                      JMP     SHORT K64               ; TRANSLATE_SCAN


                        ;------ TRANSLATE THE CHARACTER

05D5                    K56:                                    ; TRANSLATE-CHAR
05D5 FE C8                      DEC     AL                      ; CONVERT ORIGIN
05D7 2E: D7                     XLAT    CS:K11                  ; CONVERT THE SCAN CODE TO ASCII
05D9 F6 06 0096 R 02            TEST    @KB_FLAG_3,LC_E0        ; IS THIS A NEW KEY?
05DE 74 15                      JZ      K57                     ;  NO, GO FILL BUFFER
05E0 B4 E0                      MOV     AH,MC_E0                ;  YES, PUT SPECIAL MARKER IN AH
05E2 EB 11                      JMP     SHORT K57               ; PUT IT INTO THE BUFFER
```

```
                        ;------ TRANSLATE SCAN FOR PSEUDO SCAN CODES
                        K64:                                    ; TRANSLATE-SCAN-ORGD
05E4 FE C8                      DEC     AL                      ; CONVERT ORIGIN
05E6 2E: D7                     XLAT    CS:K8                   ; CTL TABLE SCAN
05E8 8A E0                      MOV     AH,AL                   ; PUT VALUE INTO AH
05EA B0 00                      MOV     AL,0                    ; ZERO ASCII CODE
05EC F6 06 0096 R 02            TEST    @KB_FLAG_3,LC_E0        ; IS THIS A NEW KEY?
05F1 74 02                      JZ      K57                     ;  NO, GO FILL BUFFER
05F3 B0 E0                      MOV     AL,MC_E0                ;  YES, PUT SPECIAL MARKER IN AL
                        ;------ PUT CHARACTER INTO BUFFER
                        K57:                                    ; BUFFER-FILL
05F5 3C FF                      CMP     AL,-1                   ; IS THIS AN IGNORE CHAR
05F7 74 05                      JE      K59                     ; YES, DO NOTHING WITH IT
05F9 80 FC FF                   CMP     AH,-1                   ; LOOK FOR -1 PSEUDO SCAN
05FC 75 03                      JNE     K61                     ; NEAR_INTERRUPT_RETURN
                        K59:                                    ; NEAR-INTERRUPT-RETURN
05FE E9 03A0 R                  JMP     K26                     ; INTERRUPT_RETURN
                        K61:
0601 8B 1E 001C R               MOV     BX,@BUFFER_TAIL         ; GET THE END POINTER TO THE BUFFER
0605 8B F3                      MOV     SI,BX                   ; SAVE THE VALUE
0607 E8 0168 R                  CALL    K4                      ; ADVANCE THE TAIL
060A 3B 1E 001A R               CMP     BX,@BUFFER_HEAD         ; HAS THE BUFFER WRAPPED AROUND
060E 74 1D                      JE      K62                     ; BUFFER_FULL_BEEP
0610 89 04                      MOV     [SI],AX                 ; STORE THE VALUE
0612 89 1E 001C R               MOV     @BUFFER_TAIL,BX         ; MOVE THE POINTER UP
0616 FA                         CLI                             ; TURN OFF INTERRUPTS
0617 B0 20                      MOV     AL,EOI                  ; END OF INTERRUPT COMMAND
0619 E6 20                      OUT     INTA00,AL               ; SEND COMMAND TO INTERRUPT CONTROL PORT
061B B0 AE                      MOV     AL,ENA_KBD              ; INSURE KEYBOARD IS ENABLED
061D E8 063C R                  CALL    SHIP_IT                 ; EXECUTE ENABLE
0620 B8 9102                    MOV     AX,09102H               ; MOVE IN POST CODE & TYPE
0623 CD 15                      INT     15H                     ; PERFORM OTHER FUNCTION
0625 80 26 0096 R FC            AND     @KB_FLAG_3,NOT LC_E0+LC_E1 ; RESET LAST CHAR H.C. FLAG
062A E9 03AF R                  JMP     K27A                    ; INTERRUPT_RETURN
                        ;------ BUFFER IS FULL SOUND THE BEEPER
                        K62:
062D B0 20                      MOV     AL,EOI                  ; ENABLE INTERRUPT CONTROLLER CHIP
062F E6 20                      OUT     INTA00,AL               ;
0631 B9 02A6                    MOV     CX,678                  ; DIVISOR FOR 1760 HZ
0634 B3 04                      MOV     BL,4                    ; SHORT BEEP COUNT (1/16 + 1/64 DELAY)
0636 E8 0000 E                  CALL    BEEP                    ; GO TO COMMON BEEP HANDLER
0639 E9 03AA R                  JMP     K27                     ; EXIT
                        KB_INT_1        ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------------------------
                        ;                                                                         -
                        ;       SHIP_IT                                                           -
                        ;                                                                         -
                        ;       THIS ROUTINE HANDLES TRANSMISSION OF COMMAND AND DATA BYTES       -
                        ;       TO THE KEYBOARD CONTROLLER.                                       -
                        ;                                                                         -
                        ;--------------------------------------------------------------------------
 063C                   SHIP_IT PROC    NEAR
 063C 50                        PUSH    AX                      ; SAVE DATA TO SEND

                        ;------- WAIT FOR COMMAND TO BE ACCEPTED
 063D FA                        CLI                             ; DISABLE INTERRUPTS TILL DATA SENT
 063E 2B C9                     SUB     CX,CX                   ; CLEAR TIMEOUT COUNTER
 0640                   SI0:
 0640 E4 64                     IN      AL,STATUS_PORT          ; READ KEYBOARD CONTROLLER STATUS
 0642 A8 02                     TEST    AL,INPT_BUF_FULL        ; CHECK FOR ITS INPUT BUFFER BUSY
 0644 E0 FA                     LOOPNZ  SI0                     ; WAIT FOR COMMAND TO BE ACCEPTED

 0646 58                        POP     AX                      ; GET DATA TO SEND
 0647 E6 64                     OUT     STATUS_PORT,AL          ; SEND TO KEYBOARD CONTROLLER
 0649 FB                        STI                             ; ENABLE INTERRUPTS AGAIN
 064A C3                        RET                             ; RETURN TO CALLER
 064B                   SHIP_IT ENDP

                        ;--------------------------------------------------------------------------
                        ;                                                                         -
                        ;       SND_DATA                                                          -
                        ;                                                                         -
                        ;       THIS ROUTINE HANDLES TRANSMISSION OF COMMAND AND DATA BYTES       -
                        ;       TO THE KEYBOARD AND RECEIPT OF ACKNOWLEDGEMENTS.  IT ALSO         -
                        ;       HANDLES ANY RETRIES IF REQUIRED                                   -
                        ;                                                                         -
                        ;--------------------------------------------------------------------------

 064B                   SND_DATA PROC   NEAR
 064B 50                        PUSH    AX                      ; SAVE REGISTERS
 064C 53                        PUSH    BX                      ; *
 064D 51                        PUSH    CX
 064E 8A F8                     MOV     BH,AL                   ; SAVE TRANSMITTED BYTE FOR RETRIES
 0650 B3 03                     MOV     BL,3                    ; LOAD RETRY COUNT
 0652 FA                SD0:    CLI                             ; DISABLE INTERRUPTS
 0653 80 26 0097 R CF           AND     @KB_FLAG_2,NOT (KB_FE+KB_FA) ; CLEAR ACK AND RESEND FLAGS

                        ;------- WAIT FOR COMMAND TO BE ACCEPTED

 0658 2B C9                     SUB     CX,CX
 065A E4 64             SD5:    IN      AL,STATUS_PORT
 065C A8 02                     TEST    AL,INPT_BUF_FULL
 065E E0 FA                     LOOPNZ  SD5                     ; WAIT FOR COMMAND TO BE ACCEPTED

 0660 8A C7                     MOV     AL,BH                   ; REESTABLISH BYTE TO TRANSMIT
 0662 E6 60                     OUT     PORT_A,AL               ; SEND BYTE
 0664 FB                        STI                             ; ENABLE INTERRUPTS
 0665 B9 1A00                   MOV     CX,01A00H               ; LOAD COUNT FOR 10mS+
 0668 F6 06 0097 R 30   SD1:    TEST    @KB_FLAG_2,KB_FE+KB_FA  ; SEE IF EITHER BIT SET
 066D 75 0D                     JNZ     SD3                     ; IF SET, SOMETHING RECEIVED GO PROCESS
 066F E2 F7                     LOOP    SD1                     ; OTHERWISE WAIT

 0671 FE CB             SD2:    DEC     BL                      ; DECREMENT RETRY COUNT
 0673 75 DD                     JNZ     SD0                     ; RETRY TRANSMISSION
 0675 80 0E 0097 R 80           OR      @KB_FLAG_2,KB_ERR       ; TURN ON TRANSMIT ERROR FLAG
 067A EB 07                     JMP     SHORT SD4               ; RETRIES EXHAUSTED FORGET TRANSMISSION

 067C F6 06 0097 R 10   SD3:    TEST    @KB_FLAG_2,KB_FA        ; SEE IF THIS IS AN ACKNOWLEDGE
 0681 74 EE                     JZ      SD2                     ; IF NOT, GO RESEND

 0683 59                SD4:    POP     CX                      ; RESTORE REGISTERS
 0684 5B                        POP     BX                      ;
 0685 58                        POP     AX                      ; *
 0686 C3                        RET                             ; RETURN, GOOD TRANSMISSION
 0687                   SND_DATA ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------------------------
                        ;                                                                          -
                        ;       SND_LED                                                            -
                        ;                                                                          -
                        ;       THIS ROUTINE TURNS ON THE MODE INDICATORS.                         -
                        ;                                                                          -
                        ;--------------------------------------------------------------------------
0687                    SND_LED PROC    NEAR
0687 FA                         CLI                             ; TURN OFF INTERRUPTS
0688 F6 06 0097 R 40            TEST    @KB_FLAG_2,KB_PR_LED    ; CHECK FOR MODE INDICATOR UPDATE
068D 75 47                      JNZ     SL1                     ; DONT UPDATE AGAIN IF UPDATE UNDERWAY
                        ;
068F 80 0E 0097 R 40            OR      @KB_FLAG_2,KB_PR_LED    ; TURN ON UPDATE IN PROCESS
0694 B0 20                      MOV     AL,EOI                  ; END OF INTERRUPT COMMAND
0696 E6 20                      OUT     020H,AL                 ; SEND COMMAND TO INTERRUPT CONTROL PORT
0698 EB 0D                      JMP     SHORT SL0               ; GO SEND MODE INDICATOR COMMAND
                        ;
069A                    SND_LED1:
069A FA                         CLI                             ; TURN OFF INTERRUPTS
069B F6 06 0097 R 40            TEST    @KB_FLAG_2,KB_PR_LED    ; CHECK FOR MODE INDICATOR UPDATE
06A0 75 34                      JNZ     SL1                     ; DONT UPDATE AGAIN IF UPDATE UNDERWAY
                        ;
06A2 80 0E 0097 R 40            OR      @KB_FLAG_2,KB_PR_LED    ; TURN ON UPDATE IN PROCESS
06A7 B0 ED              SL0:    MOV     AL,LED_CMD              ; LED CMD BYTE
06A9 E8 064B R                  CALL    SND_DATA                ; SEND DATA TO KEYBOARD
06AC FA                         CLI                             ;
06AD E8 06D8 R                  CALL    MAKE_LED                ; GO FORM INDICATOR DATA BYTE
06B0 80 26 0097 R F8            AND     @KB_FLAG_2,0F8H         ; CLEAR MODE INDICATOR BITS
06B5 08 06 0097 R               OR      @KB_FLAG_2,AL           ; SAVE PRESENT INDICATORS FOR NEXT TIME
06B9 F6 06 0097 R 80            TEST    @KB_FLAG_2,KB_ERR       ; TRANSMIT ERROR DETECTED
06BE 75 0B                      JNZ     SL2                     ;  YES, BYPASS SECOND BYTE TRANSMISSION
                        ;
06C0 E8 064B R                  CALL    SND_DATA                ; SEND DATA TO KEYBOARD
06C3 FA                         CLI                             ; TURN OFF INTERRUPTS
06C4 F6 06 0097 R 80            TEST    @KB_FLAG_2,KB_ERR       ; TRANSMIT ERROR DETECTED
06C9 74 06                      JZ      SL3                     ; IF NOT, DONT SEND AN ENABLE COMMAND
                        ;
06CB B0 F4              SL2:    MOV     AL,KB_ENABLE            ; GET KEYBOARD CSA ENABLE COMMAND
06CD E8 064B R                  CALL    SND_DATA                ; SEND DATA TO KEYBOARD
06D0 FA                         CLI                             ; TURN OFF INTERRUPTS
06D1 80 26 0097 R 3F    SL3:    AND     @KB_FLAG_2,NOT(KB_PR_LED+KB_ERR) ; TURN OFF MODE INDICATOR
                                                                ; UPDATE AND TRANSMIT ERROR FLAG
06D6 FB                 SL1:    STI                             ; ENABLE INTERRUPTS
06D7 C3                         RET                             ; RETURN TO CALLER
06D8                    SND_LED ENDP
                        ;--------------------------------------------------------------------------
                        ;                                                                          -
                        ;       MAKE_LED                                                           -
                        ;                                                                          -
                        ;       THIS ROUTINE FORMS THE DATA BYTE NECESSARY TO TURN ON/OFF           -
                        ;         THE MODE INDICATORS                                              -
                        ;                                                                          -
                        ;--------------------------------------------------------------------------
06D8                    MAKE_LED PROC   NEAR
06D8 51                         PUSH    CX                      ; SAVE CX
06D9 A0 0017 R                  MOV     AL,@KB_FLAG             ; GET CAPS & NUM LOCK INDICATORS
06DC 24 70                      AND     AL,CAPS_STATE+NUM_STATE+SCROLL_STATE ; ISOLATE INDICATORS
06DE B1 04                      MOV     CL,4                    ; SHIFT COUNT
06E0 D2 C0                      ROL     AL,CL                   ; SHIFT BITS OVER TO TURN ON INDICATORS
06E2 24 07                      AND     AL,07H                  ; MAKE SURE ONLY MODE BITS ON
06E4 59                         POP     CX
06E5 C3                         RET                             ; RETURN TO CALLER
06E6                    MAKE_LED ENDP

06E6                    CODE    ENDS
                                END
```