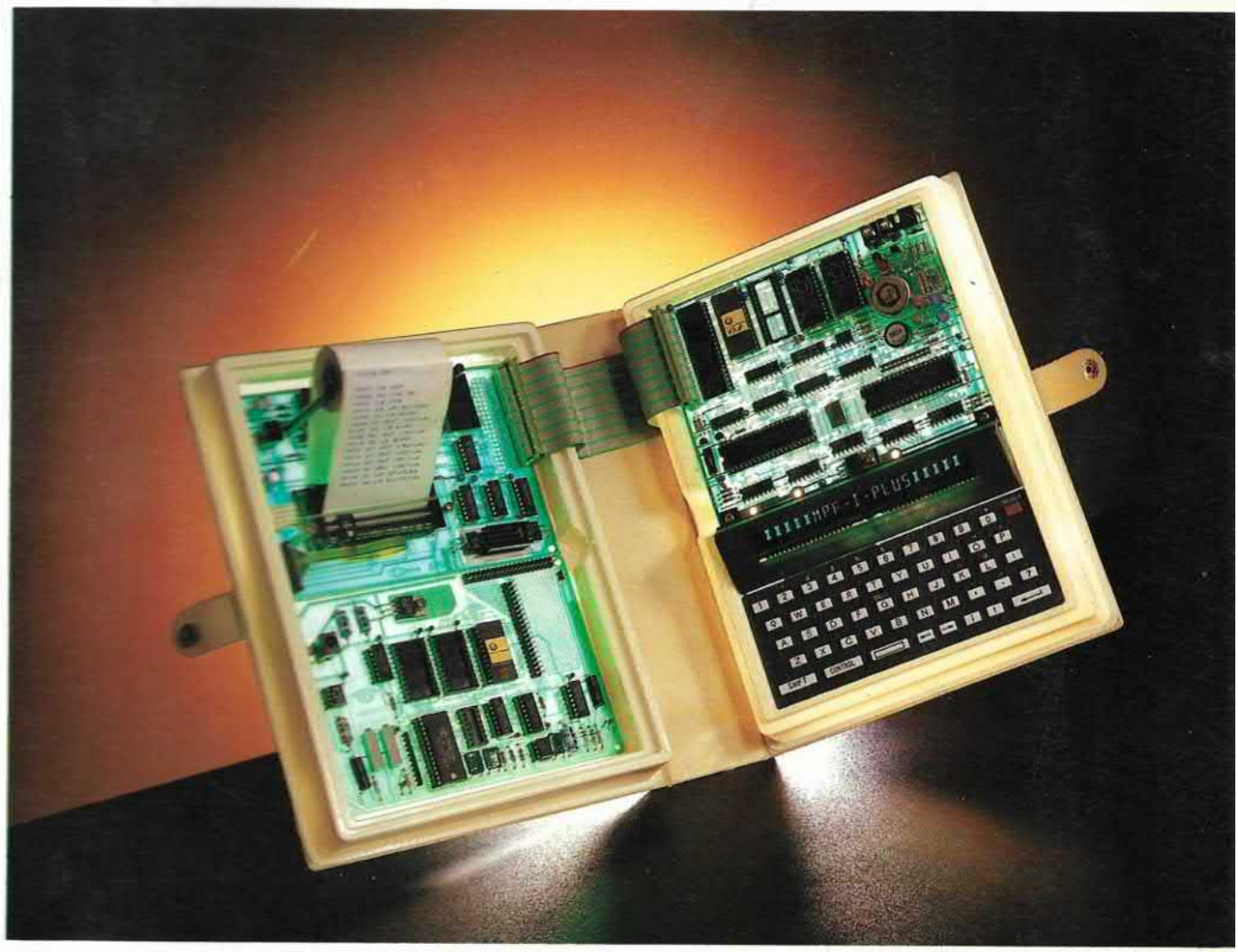
Micro-Professor

MPF-IP MONITOR PROGRAM SOURCE LISTING



MULTITECH INDUSTRIAL CORPORATION

MULTITECH INDUSTRIAL CORPORATION
OFFICE 977 MIN SHEN E ROAD TAIPEI 105.
TAIWAN, R.O.C.
TEL:(02)769-1225(10 LINES)
[illegible] MULTIIC, 19162 MULTIIC.
FACTORY 2 TECHNOLOGY ROAD III
[illegible] SCIENCE, BASED INDUSTRIAL PARK
HSINCHU, TAIWAN, 300, R.O.C.
TEL:(035)770102(3 LINES)

PRICE:US$10.00

Micro-Professor

MPF-IP MONITOR PROGRAM
SOURCE LISTING

```
                    ;*********************************************************
                    ;*                                                       *
                    ;*    COPYRIGHT , MULTITECH INDUSTRIAL CORP. 1983        *
                    ;*    ALL right reserved.                                *
                    ;*    No part of this software may be copied without     *
                    ;*    the express written consent of MULTITECH           *
                    ;*    INDUSTRIAL CORP.                                   *
                    ;*                                                       *
                    ;*********************************************************
                    ;
                    ;
                    ;
                    ;
                    ;
P82551  EQU     83H     ;8255 I control port
DIG1    EQU     80H     ;8255 I port A
DIG2    EQU     81H     ;8255 I port B
DIG3    EQU     82H     ;8255 I port C
P82552  EQU     93H     ;8255 II Control port
SEG1    EQU     90H     ;8255 II port A
SEG2    EQU     91H     ;8255 II port B
KIN     EQU     92H     ;8255 II port C
PWCODE  EQU     0A5H    ;Power up code
ZSUM    EQU     0E0H    ;This will make the sum of all
                        ;monitor codes to be zero.

;The following EQUATEs are used for timing.Their values
;depend on the CPU clock frequency. (In this version, the
;crystal freqency is 1.79 MHz.)

COLDEL  EQU     80      ;Column delay for routine
                        ;SCAN and SCAN1.
F1KHZ   EQU     65      ;Delay count for 1K Hz square wave,
                        ;used by routine TONE1K.
F2KHZ   EQU     31      ;Delay count for 2K Hz square wave,
                        ;used by routine TONE2K.
MPERIOD EQU     32      ;1KHz and 2KHz threshold used by
                        ;tape input routine PERIOD

; The following EQUATEs are used for tape modulation.
; If the quality of tape recoder is good,the user may
; change '4 4 2 8' to '2 2 1 4'. This will double
; the tape data rate.
; If the quality of tape recoder is poor ,the user may
; change '4 4 2 8' to '6 6 3 12'. This will improve
; error performance but slow down the data rate.
; Although the data format is changed, the tape is still
; compatible in each case,because only the ratio is
; detected in the Tape read.

ONE_1K  EQU     4
ONE_2K  EQU     4
ZERO_1K EQU     2
ZERO_2K EQU     8

;*****************************************************************
;I/O port assignment: (8255 I)
```

```
                        ; port A (address 80H): The first eight digits of  display.
                        ;       bit0--digit 1
                        ;       bit1--digit 2
                        ;       bit2--digit 3
                        ;       bit3--digit 4
                        ;       bit4--digit 5
                        ;       bit5--digit 6
                        ;       bit6--digit 7
                        ;       bit7--digit 8
                        ; Port B (Address 81H): The second eight digits of display.
                        ; bit0--digit 9
                        ;       bit1--digit 10
                        ;       bit2--digit 11
                        ;       bit3--digit 12
                        ;       bit4--digit 13
                        ;       bit5--digit 14
                        ;       bit6--digit 15
                        ;       bit7--digit 16
                        ; port C (address 82H): The last four digits of display.
                        ;       bit0--digit 17
                        ;       bit1--digit 18
                        ;       bit2--digit 19
                        ;       bit3--digit 20
                        ;       bit4--SHIFT key
                        ;       bit5--CTRL key
                        ;       bit6,7--unused
                        
                        ;*****************************************************************
                        ;I/O port assignment: (8255 II)
                        
                        ; port A (address 90H):The first eight segments.
                        ;       bit0--segment a
                        ;       bit1--segment b
                        ;       bit2--segment c
                        ;       bit3--segment d
                        ;       bit4--segment e
                        ;       bit5--segment f
                        ;       bit6--segment g
                        ;       bit7--segment h
                        ; port B (address 91H):The second seven segments.
                        ;       bit0--segment i
                        ;       bit1--segment j
                        ;       bit2--segment k
                        ;       bit3--segment l
                        ;       bit4--segment m
                        ;       bit5--segment n
                        ;       bit6--segment dp
                        ;       bit7:unused
                        ; port C (address 92H): Tape I/O ,Break,Keyboard input.
                        ;       bit0--K1 (Keyboard matrix 1st row)
                        ;       bit1--K2 (keyboard matrix 2nd row)
                        ;       bit2--K3 (Keyboard matrix 3rd row)
                        ;       bit3--EAR (Input from tape recoder)
                        ;       bit4--Break signal
                        ;       bit5--MIC (Output to tape recoder)
                        ;       bit6--7:unused
                        
                        ;*****************************************************************
```

```
                                MPF_IP          1983.1.1
LOC   OBJ CODE M STMT SOURCE STATEMENT

                    117  ;  -- reset --
                    118  ; There are two cases that will generate a RESET s.g.
                    119  ;     (i) power-up
                    120  ;    (ii) 'RS' key pressed
                    121  ; In both cases, the following actions will be taken:
                    122  ;   a) disable interrupt, set interrupt mode to 0
                    123  ;      set I register to 00 and start execution
                    124  ;      at address 0000 (by Z80 CPU itself).
                    125  ;   b) set user's SP to FEA0H;
                    126  ;   c) set user's I register to 00 and disable user's
                    127  ;      interrupt flip-flop;
                    128  ; In addition, subroutine INI will be called on power-up
                    129  ; reset, which has the following effects:
                    130  ;   d) turn on printer (PRT_MPF)
                    131  ;   e) disable BREAK POINT;
                    132  ;   f) set the contents of location FF00H  FF01H to 66 and
                    133  ;      and 00 respectively. This will make instruction RST
                    134  ;      38H (opcode FF) have the same effect as BREAK.
                    135  ;   g) set the default value of EDITOR and ASSEMBLER.
                    136  ;   h) set the limit address of INSERT and DELETE.
                    137  ; Memory location POWERUP is used to distinguish power-up
                    138  ; from RS-key.  (POWERUP) contains a random data when
                    139  ; power-up and contains PWCODE (0A5H) thereafter.
                    140
0000   010003        141          LD      BC,300H          ;Power_up delay
                    142  RS_START:
0003   EDA9          143          CPD
0005   EA0300        144          JP      PE,RS_START
                    145
                    146  ; Initial 8255 I to mode 0 with port A and B output ,port C
                    147  ;pc0-pc3:out pc4-pc7:in.The control word is 88H.
                    148
0008   3E88          149          LD      A,10001000B
000A   D383          150          OUT     (P82551),A
                    151
                    152  ; Intial 8255 II to mode 0 with port A and B output ,port C
                    153  ;pc0-pc3:in pc4-pc7:out.The control word is 81H.
                    154
000C   3E81          155          LD      A,10000001B
000E   D393          156          OUT     (P82552),A
                    157
                    158  ; When the control word is sent out to 8255, all output
                    159  ; ports are cleared to 0.  It is necessary to disable
                    160  ; BREAK and deactivate all I/O by sending 0FFH to
                    161  ; 8255 I portA,B,C and 8255 II portC.
                    162
0010   3EFF          163          LD      A,0FFH
0012   D392          164          OUT     (KIN),A          ;Disable the BREAK signal.
0014   D380          165          OUT     (DIG1),A         ;Disable all digits.
0016   D381          166          OUT     (DIG2),A
0018   D382          167          OUT     (DIG3),A
001A   31D0FE        168          LD      SP,SYSSTK        ;Initial system stack.
                    169
                    170  ;If the content of location POWERUP is not equal to
                    171  ;PWCODE, call subroutine INI. Continue otherwise.
                    172
001D   3AF0FE        173          LD      A,(POWERUP)
0020   FEA5          174          CP      PWCODE
```

```
0022  C4B001      175          CALL    NZ,INI          ;Cold start
                  176
                  177  ; Address 28H and 30H are reserved for BREAK (RST 28H)
                  178  ; and software BREAK (RST 30H). Skip these area, monitor
                  179  ; program resumes at RESET1.
                  180
0025  180B        181          JR      RESET1
                  182  ;
                  183  ;*****************************************************************
0028              184          ORG     28H
                  185  ; Address 28H is the entry point of BREAK trap.
                  186  ; If a location is set as a BREAK point, the monitor
                  187  ; will change the content of this location to C7 (RST 28H)
                  188  ; before transfering control to user's program.
                  189  ; In execution of user's program, a trap will occur if
                  190  ; user's PC passes this location.  The monitor then takes
                  191  ; over control and the content of BREAK address
                  192  ; will be restored.  Monitor takes care of everything
                  193  ; and makes the whole mechanism transparent to the user.
                  194  ; The return address pushed onto stack is the PC after
                  195  ; executing RST 28H.  The original break address should
                  196  ; be one less than that.  The following 3 instructions
                  197  ; decrease the content of (SP) by one without changing
                  198  ; HL.
                  199
0028  E3          200          EX      (SP),HL
0029  2B          201          DEC     HL
002A  E3          202          EX      (SP),HL
002B  22FFFE      203          LD      (HLTEMP),HL
002E  180E        204          JR      CONT28
                  205  ;
                  206  ;*****************************************************************
0030              207          ORG     30H
                  208
                  209  ; Instruction RST 30H (opcode F7) is usually used as:
                  210  ;   i) Software break;
                  211  ;  ii) Terminator of user's program.
                  212  ; The effect of this instruction is to save all user's
                  213  ; registers and return to monitor.
                  214
0030  1834        215          JR      NMI
                  216
                  217  ;*****************************************************************
                  218  ; This is a part of reset routine.  Address 0028 and
                  219  ; 0030 are reserved for break point.  Reset routine
                  220  ; skips this area and resumes here.
                  221  ;
                  222  RESET1:
0032  210000      223          LD      HL,0
0035  181D        224          JR      RESET2
                  225
                  226  ;*****************************************************************
                  227
                  228  ;The following byte makes the sum of the monitor
                  229  ;code in ROM zero.
                  230
0037  E0          231          DEFB    ZSUM
                  232
```

```
                    233  ;*************************************************************
                    234
0038                235          ORG     38H
                    236
                    237  ; Entry point of RST 38H (opcode FF) or mode 1 interrupt.
                    238  ; Fetch the address stored in location FF00 and FF01,
                    239  ; then jump to this address.  Initially, FF00 and FF01
                    240  ; are set to 0066.  So RST 38 will have the same effect
                    241  ; as software break.  By changing the content of FF00
                    242  ; and FF01, the user can define his or her own service
                    243  ; routine.
                    244  ; The next three instructions push the contents of FF00
                    245  ; and FF01 to stack without changing any registers.
                    246
0038  E5            247          PUSH    HL
0039  2A01FF        248          LD      HL,(IM1AD)      ;Initially stored 0066H.
003C  E3            249          EX      (SP),HL
                    250
                    251  ; The top of the stack is now the address of user
                    252  ; defined service routine.  Pop out this address then
                    253  ; branch to it.
                    254
003D  C9            255          RET
                    256  ;
                    257  ;*************************************************************
                    258  CONT28:
                    259  ; This is a part of break service routine.  It continues
                    260  ; the program at RST28.
                    261
003E  32FEFE        262          LD      (ATEMP),A
                    263
                    264  ; The monitor has changed the content of user's
                    265  ; program at break address.  The next 3 instructions
                    266  ; restored the destroyed content.  BRAD contains the
                    267  ; break address, BRDA contains the original data at
                    268  ; break address.
                    269
0041  2AEDFE        270          LD      HL,(BRAD)
0044  3AEFFE        271          LD      A,(BRDA)
0047  77            272          LD      (HL),A
                    273  ; Send break enable signal to hardware counter.
                    274  ; A nonmaskable interrupt will be issued at the 5th M1's.
                    275
0048  3EEF          276          LD      A,11101111B
004A  D392          277          OUT     (KIN),A
004C  3AFEFE        278          LD      A,(ATEMP)       ;1st M1
004F  2AFFFE        279          LD      HL,(HLTEMP)     ;2nd M1
0052  00            280          NOP                     ;3rd M1
0053  C9            281          RET                     ;4th M1
                    282
                    283  ; Return to user's program.  Execute the instruction
                    284  ; at break address.  After finishing one instruction,
                    285  ; a nonmaskable interrupt happens and then returns
                    286  ; to the monitor program again.
                    287  ;
                    288  RESET2:
0054  22A0FF        289          LD      (USERIF),HL     ;Set user's I register and
                    290                                  ;interupt flip flop to 0
```

```
0057   22F1FE      291          LD      (TEST),HL       ;Set the contents of TEST and
                   292                                  ;STEPBF to be zero.
                   293
                   294  ; TEST is a flag for the use of monitor itself. Illegal key-in
                   295  ; blanking (bit 7 of TEST) and automatic leading zero
                   296  ; (bit 0) use this flag. Clear it here.
                   297
005A   21A0FE      298          LD      HL,USERSTK
005D   229CFF      299          LD      (USERSP),HL
0060   CD1E02      300          CALL    INI7
0063   37          301          SCF
                   302
                   303  ; Address 66H is the address for nonmaskable interrupt.
                   304  ; Skip this area,monitor resumes at SETST0
                   305
0064   1858        306          JR      SETST0
                   307  ;
                   308  ;*****************************************************************
0066               309  NMI:    ORG     66H
                   310
                   311  ; Entry point of nonmaskable interrupt.  NMI will occur
                   312  ; when user's program is breaked.
                   313  ; The service routine which starts here saves all
                   314  ; user's registers and status.  It also check the validity
                   315  ; of user's SP.
                   316
0066   32FEFE      317          LD      (ATEMP),A       ;Save A register
0069   3EFF        318          LD      A,0FFH  ;Disable BREAK signal and all digits.
006B   D380        319          OUT     (DIG1),A
006D   D381        320          OUT     (DIG2),A
006F   D382        321          OUT     (DIG3),A
0071   D392        322          OUT     (KIN),A
0073   3AFEFE      323          LD      A,(ATEMP)       ;Restore A register
0076   22FFFE      324  RGSAVE: LD      (HLTEMP),HL     ;Save register HL
0079   E1          325          POP     HL              ;Get return address from stack
007A   229EFF      326          LD      (USERPC),HL     ;Set user's PC to return
                   327                                  ;address
007D   2AFFFE      328          LD      HL,(HLTEMP)     ;Restore HL register
0080   ED739CFF    329          LD      (USERSP),SP     ;Set user's SP to current SP
0084   319CFF      330          LD      SP,USERIY+2     ;Save other registers by
0087   FDE5        331          PUSH    IY              ;continuously pushing them
0089   DDE5        332          PUSH    IX              ;onto stack
008B   D9          333          EXX
008C   E5          334          PUSH    HL
008D   D5          335          PUSH    DE
008E   C5          336          PUSH    BC
008F   D9          337          EXX
0090   08          338          EX      AF,AF'
0091   F5          339          PUSH    AF
0092   08          340          EX      AF,AF'
0093   E5          341          PUSH    HL
0094   D5          342          PUSH    DE
0095   C5          343          PUSH    BC
0096   F5          344          PUSH    AF
                   345
                   346  ; The next two instructions save I register.
                   347  ; The interrupt flip-flop (IFF2) is copied into
                   348  ; parity flag (P/V) by instruction LD  A,I.
```

```
                    349  ; The interrupt status (enabled or disabled)
                    350  ; can be determined by testing parity flag.
                    351
0097    ED57        352          LD      A,I
0099    32A1FF      353          LD      (USERIF+1),A
                    354
                    355  ; The next four instructions save IFF2 into
                    356  ; user's IFF.
                    357
009C    3E00        358          LD      A,0
009E    E2A300      359          JP      PO,SETIF        ;PO--P/V=0
00A1    3E01        360          LD      A,1
00A3    32A0FF      361  SETIF:  LD      (USERIF),A
                    362  ;
00A6    31D0FE      363          LD      SP,SYSSTK       ;Set SP to system stack.
                    364
                    365  ; The next 7 instructions check user's SP.
                    366  ; If the user's SP points to a location not
                    367  ; in RAM, display ERR-SP.
                    368
00A9    2A9CFF      369          LD      HL,(USERSP)
00AC    2B          370          DEC     HL
00AD    CD1908      371          CALL    RAMCHK
00B0    2026        372          JR      NZ,SETST2
00B2    2B          373          DEC     HL
00B3    CD1908      374          CALL    RAMCHK
00B6    2020        375          JR      NZ,SETST2
                    376
                    377  ; If the user's stack and system stack are
                    378  ; overlayed, then display SYS-SP.  This checking
                    379  ; is done by the following  instructions.
                    380
00B8    116101      381          LD      DE,-USERSTK+1
00BB    19          382          ADD     HL,DE
00BC    381F        383          JR      C,SETST3
                    384  SETST0:
00BE    3AEFFE      385          LD      A,(BRDA)
00C1    2AEDFE      386          LD      HL,(BRAD)       ;Restore the data at breakpoint
                    387                                  ;address.
00C4    77          388          LD      (HL),A
                    389
                    390  ; In execution of STEP or GO command ,if the
                    391  ; user's SP is legal (carry flag is zero)then
                    392  ; display user's PC and the first four
                    393  ; register contents.
                    394  ; User can use the UP or DOWN keys to check
                    395  ; the register contents.
                    396  ; Otherwise, display fixed message (ERR-SP
                    397  ; or SYS-SP )
                    398
00C5    D42806      399          CALL    NC,MEMDP2
                    400  ;
                    401  ;
                    402  ;*****************************************************************
                    403  ; Scan the display and keyboard. When a key is
                    404  ; detected, take proper action  according to the
                    405  ; key pressed.
                    406
```

```
                    407  MAIN:
00C8   31D0FE       408          LD      SP,SYSSTK       ;Initial system stack.
00CB   CD4602       409          CALL    SCAN            ;Scan display and input keys.
                    410                                  ;Routine SCAN will not return
                    411                                  ;until any key is pressed.
                    412                                  ;After a key is detected,there
                    413                                  ;will be accompanied with a
                    414                                  ;beep sound.
00CE   F5           415          PUSH    AF
00CF   CDF607       416          CALL    CLRBF
00D2   F1           417          POP     AF
00D3   CD0B01       418          CALL    KEYEXEC         ;Input key dispatch routine.
00D6   18F0         419          JR      MAIN            ;Back to MAIN, get more keys
                    420                                  ;and execute them.
                    421  ;
                    422  ;*****************************************************************
                    423
                    424  SETST2:
00D8   21720C       425          LD      HL,ERR_SP       ;Display ERR_SP
00DB   1803         426          JR      SETST4
                    427  SETST3:
00DD   217B0C       428          LD      HL,SYS_SP       ;Display SYS_SP
                    429  SETST4:
00E0   CD8608       430          CALL    PRTMES          ;Print message
00E3   37           431          SCF
00E4   18D8         432          JR      SETST0
                    433
                    434  ;*****************************************************************
                    435
                    436  ;SOFTWARE ESC command -- Reenter montior.
                    437  ;Executed by depressing the Q and CTRL keys together.
                    438  ;The ESC command escapes from the existing command
                    439  ;and returns to monitor.
                    440  ;ESC is operative only in the commands that sample
                    441  ;the kepboard.
                    442  ;MPF_IP will responds to ESC by displaying the MPF_IP
                    443  ;montior prompt <.
                    444
                    445  ESCAPE:
00E6   31D0FE       446          LD      SP,SYSSTK
00E9   CDF607       447          CALL    CLRBF
00EC   CD8509       448          CALL    CR3
00EF   18D7         449          JR      MAIN
                    450
                    451  ;*****************************************************************
                    452
                    453  ;Executed whe UP arrow or DOWN arrow key is pressed.
                    454
                    455  FOR:
00F1   3A80FF       456          LD      A,(TYPEFG)
00F4   FE10         457          CP      10H             ;M
00F6   CAA203       458          JP      Z,MFOR          ;Display next four memory
                    459                                  ;contents.
00F9   D2B905       460          JP      NC,RFOR         ;Display next four register
                    461                                  ;contents.
00FC   1879         462          JR      IGNORE
                    463  BACK:
00FE   3A80FF       464          LD      A,(TYPEFG)
```

```
0101   FE10          465          CP      10H             ;M
0103   CAB703        466          JP      Z,MBACK         ;Display last four memory
                     467                                  ;contents.
0106   D2ED05        468          JP      NC,RBACK        ;Display last four register
                     469                                  ;contents.
0109   186C          470          JR      IGNORE
                     471
                     472  ;*************************************************************
                     473
                     474  ;Input key dispatch routine.
                     475  ;This routine uses the key codes returned by subroutine
                     476  ;SCAN, which is one byte (ASC II code) stored in A.
                     477
                     478  KEYEXEC:
010B   FE69          479          CP      69H
010D   28E2          480          JR      Z,FOR           ;DOWN ARROW.
010F   FE5E          481          CP      5EH
0111   28EB          482          JR      Z,BACK          ;UP ARROW.
0113   FE4D          483          CP      'M'
0115   CAF302        484          JP      Z,MEMEXC        ;MEMORY DISPLAY AND MODIFY.
0118   FE52          485          CP      'R'
011A   CA2505        486          JP      Z,REGEXC        ;REGISTER DISPLAY AND MODIFY.
011D   FE4C          487          CP      'L'
011F   CACA06        488          JP      Z,LOAD          ;TAPE READ.
0122   FE57          489          CP      'W'
0124   CA7906        490          JP      Z,DUMP          ;TAPE WRITE.
0127   FE47          491          CP      'G'
0129   CAC304        492          JP      Z,GOEXEC        ;EXECUTION
012C   FE53          493          CP      'S'
012E   CAB104        494          JP      Z,STEP          ;SINGLE STEP.
0131   FE42          495          CP      'B'
0133   CA7D04        496          JP      Z,BREAK         ;BREAK AT SPECIFIED ADDRESS.
0136   FE46          497          CP      'F'
0138   CAC003        498          JP      Z,FILLDA        ;FILL DATA.
013B   FE49          499          CP      'I'
013D   CAE603        500          JP      Z,INSET         ;INSERT A BLOCK OF DATAS.
0140   FE44          501          CP      'D'
0142   CA4804        502          JP      Z,DELETE        ;DELETE ONE BYTE OF DATA.
0145   FE4A          503          CP      'J'
0147   CA6204        504          JP      Z,JUMP          ;JUMP RELATIVE.
014A   FE01          505          CP      1
014C   CA0000    X   506          JP      Z,ASM           ;ASSEMBLER (CONTROL A).
014F   FE0C          507          CP      0CH
0151   CA0000    X   508          JP      Z,LASM          ;LINE ASSEMBLER (CONTROL L).
0154   FE02          509          CP      2
0156   282D          510          JR      Z,BASIC3        ;ENTER BASIC (CONTROL B).
0158   FE03          511          CP      3
015A   2829          512          JR      Z,BASIC3        ;REENTER BASIC (CONTROL C).
015C   FE04          513          CP      4
015E   281D          514          JR      Z,DEASM3        ;DISASSEMBLER (CONTROL D).
0160   FE05          515          CP      5
0162   CA0000    X   516          JP      Z,EDIT          ;EDITOR (CONTROL E).
0165   FE07          517          CP      7
0167   2840          518          JR      Z,BEEP_CONTROL  ;BEEP SOUND CONTROL (CONTROL G).
0169   FE12          519          CP      12H
016B   CA0000    X   520          JP      Z,REEDIT        ;REEDIT (CONTROL R).
016E   FE10          521          CP      10H
0170   2823          522          JR      Z,PRT_CONTROL   ;PRINTER CONTROL (CONTROL P).
```

```
 LOC   OBJ CODE M STMT SOURCE STATEMENT                          ASM 5.9

0172   FE0D        523          CP      0DH
0174   CA8509      524          JP      Z,CR3           ;LINE FEED.
                   525
                   526  ;*****************************************************************
                   527
                   528  IGNORE:
0177   21F1FE      529          LD      HL,TEST
017A   CBFE        530          SET     7,(HL)          ;Routine SCAN will check
                   531                                  ;bit 7 of test. If it is set
                   532                                  ;all DIGITS will be disabled.
                   533                                  ;This is a warning message to
                   534                                  ;the user when a illegal key
                   535                                  ;is entered.
017C   C9          536          RET
                   537
                   538  ;*****************************************************************
                   539
                   540  ;Executed by depressing the D and CTRL keys together.
                   541  ;Since the disassembler is located on the montior
                   542  ;of printer,so that MPF_IP will ignore command D
                   543  ;unless printer (PRT_MPF) is exists.
                   544
                   545  DEASM3:
017D   CDA808      546          CALL    PTESTT  ;Ret if printer is not exists
                   547                          ;or the toggle printer switch is off.
0180   C0          548          RET     NZ
0181   CD0000   X  549          CALL    DEASM   ;Call disassembler.
0184   C9          550          RET
                   551
                   552  ;*****************************************************************
                   553
                   554  ;Executed by depressing B and CTRL keys together.
                   555  ;The optional MPF_IP BASIC INTERPRETER is a 8K ROM
                   556  ;resident .It is supplied as one 2764 ROM that plugs
                   557  ;into socket U3.
                   558  ;The starting address of BASIC INTERPRETER is 2000H
                   559  ;MPF_IP will check the content of the memory location
                   560  ;2000H is 0CDH or not.If yes enter BASIC ,otherwise
                   561  ;ignore this command and return to monitor.
                   562  ;Avoid to changing the contents in RAM ,we used the
                   563  ;command C to reenter BASIC.
                   564
                   565  BASIC3:
0185   47          566          LD      B,A
0186   3A0020      567          LD      A,(2000H)
0189   FECD        568          CP      0CDH
018B   78          569          LD      A,B
018C   C0          570          RET     NZ
018D   FE02        571          CP      2
018F   CA0000   X  572          JP      Z,BASICZ
0192   C32020      573          JP      BASICC
                   574
                   575  ;*****************************************************************
                   576
                   577  ; Control print command -- Toggle printer on/off
                   578  ; The CTRL PRINT command turns the printer on
                   579  ; if it is off and off if it is on.
                   580  ; The command is entered by depressing the P and CTRL
```

```
                    581  ; keys together.
                    582
                    583  PRT_CONTROL:
0195  3AF3FE        584          LD      A,(PRTFLG)
0198  2F            585          CPL
0199  32F3FE        586          LD      (PRTFLG),A
019C  A7            587          AND     A
019D  21890C        588          LD      HL,PRTOFF
01A0  2003          589          JR      NZ,PRTF
01A2  21820C        590          LD      HL,PRTON
                    591  PRTF:
01A5  CD8608        592          CALL    PRTMES
01A8  C9            593          RET
                    594
                    595  ;
                    596  ;***************************************************
                    597
                    598  ; Control sound command -- Toggle switch on/off
                    599
                    600  BEEP_CONTROL:
01A9  21F4FE        601          LD      HL,BEEPSET
01AC  7E            602          LD      A,(HL)
01AD  2F            603          CPL
01AE  77            604          LD      (HL),A
01AF  C9            605          RET
                    606
                    607  ;***************************************************************
                    608
                    609  ;Power_up intialization.
                    610
                    611  INI:
01B0  210000        612          LD      HL,0
01B3  22F3FE        613          LD      (PRTFLG),HL             ;Set toggle printer
                    614                                          ;switch on.
                    615                                          ;Set toggle sound beep
                    616                                          ;switch on.
                    617
                    618  ;The next 7 instructions check IC on U4 is RAM or not.
                    619
01B6  21FFF7        620          LD      HL,0F7FFH
01B9  010008        621  RAMT1   LD      BC,800H
01BC  CD1908        622  RAMT2   CALL    RAMCHK
01BF  2802          623          JR      Z,TNEXT
01C1  187E          624          JR      INI8
01C3  EDA9          625  TNEXT   CPD
01C5  EABC01        626          JP      PE,RAMT2
                    627
                    628  ;The next four instructions set the default values according
                    629  ;to EDITOR and ASSEMBLER respectively.
                    630
01C8  219D0C        631          LD      HL,RAM4K_VALUE_SET
                    632  INI6:
01CB  11DFFE        633          LD      DE,RAM_START_ADDR
01CE  010C00        634          LD      BC,12
01D1  EDB0          635          LDIR
                    636
01D3  CD1E02        637          CALL    INI7    ;Get reset display pattern.
01D6  DD212CFF      638          LD      IX,DISPBF
```

```
                    639                         ;DiSplay the following
                    640                         ;patterns sequence, each 0.157
                    641                         ;seconds:
                    642                         ;       '                   *'
                    643                         ;       '                  **'
                    644                         ;       '                 ***'
                    645                         ;       '                ****'
                    646                         ;       '               *****'
                    647                         ;       '              *****M'
                    648                         ;       '             *****MP'
                    649                         ;       '            *****MPF'
                    650                         ;       '           *****MPF-'
                    651                         ;       '          *****MPF-I'
                    652                         ;       '         *****MPF-I-'
                    653                         ;       '        *****MPF-I-P'
                    654                         ;       '       *****MPF-I-PL'
                    655                         ;       '      *****MPF-I-PLU'
                    656                         ;       '     *****MPF-I-PLUS'
                    657                         ;       '    *****MPF-I-PLUS*'
                    658                         ;       '   *****MPF-I-PLUS**'
                    659                         ;       '  *****MPF-I-PLUS***'
                    660                         ;       ' *****MPF-I-PLUS****'
                    661                         ;       '*****MPF-I-PLUS*****'
01DA  0E14          662           LD    C,20    ;Pattern counts.
01DC  060A          663  INI1:    LD    B,10            ;Display 0.157 sec.
                    664  INI2:
01DE  CD9B02        665           CALL  SCAN1
01E1  10FB          666           DJNZ  INI2
01E3  DD23          667           INC   IX
01E5  DD23          668           INC   IX
01E7  0D            669           DEC   C
01E8  20F2          670           JR    NZ,INI1
01EA  CDA808        671           CALL  PTESTT
01ED  2003          672           JR    NZ,INI5
01EF  CD0000        673           CALL  PRT_MPF         ;Print '*****MPF-I-PLUS*****'
01F2  3EA5          674  INI5     LD    A,PWCODE
01F4  32F0FE        675  INI3:    LD    (POWERUP),A     ;Load power_up code into
                    676                                ;(POWERUP). The monitor
                    677                                ;uses the location to decide
                    678                                ;whether a reset signal is
                    679                                ;on power_up.
01F7  21F5FE        680           LD    HL,FBEEP
01FA  3644          681           LD    (HL),44H        ;Freqency of BEEP.
01FC  23            682           INC   HL
01FD  362F          683           LD    (HL),2FH        ;Time duration of BEEP.
01FF  23            684           INC   HL
0200  3600          685           LD    (HL),0
0202  216600        686  INI4:    LD    HL,NMI
0205  2201FF        687           LD    (IM1AD),HL      ;Set the service routine
                    688                                ;of RST 38H to NMI ,which is the
                    689                                ;nonmaskable interrupt service
                    690                                ;routine for break point and
                    691                                ;single step.
0208  2EE6          692           LD    L,0E6H          ;Set SOFTWARE ESCAPE address
                    693                                ;to be 00E6H.
                    694                                ;(i.e.,User's program return
                    695                                ;address.)
020A  22A0FE        696           LD    (USERSTK),HL
```

```
020D   CD1702       697          CALL    CLRI
                    698  CLRB:
                    699  ; Clear break point by setting the breakpoint address
                    700  ;to 1FFFH. This address is the last address of monitor,
                    701  ; so,break can never happen.
                    702
0210   21FF1F       703          LD      HL,1FFFH
0213   22EDFE       704          LD      (BRAD),HL
0216   C9           705          RET
                    706  CLRI:
                    707  ; Clear limit address of INSERT and DELETE command.
                    708  ; Avoid to changing the contents of SYSTEM RAM ,we must
                    709  ; set limit address.
                    710  ; The default value of limit address is 0FE00H.
                    711
0217   2100FE       712          LD      HL,0FE00H
021A   22EBFE       713          LD      (END_ADDR),HL
021D   C9           714          RET
                    715  INI7:
                    716  ; Get pattern '                    *****MPF-PLUS-*****'
                    717
021E   CD4008       718          CALL    CLRDSP
0221   2118FF       719          LD      HL,INPBF+20
0224   2282FF       720          LD      (OUTPTR),HL
0227   2154FF       721          LD      HL,DISPBF+40
022A   2284FF       722          LD      (DISP),HL
022D   215D0C       723          LD      HL,MPFII
0230   CDCA09       724          CALL    MSG
0233   DD2154FF     725          LD      IX,DISPBF+40
0237   210000       726          LD      HL,0
023A   2280FF       727          LD      (TYPEFG),HL     ;Set the contents of TYPEFG and
                    728                                  ;CRSET to be zero.
023D   CD7109       729          CALL    CR0
0240   C9           730          RET
                    731  INI8:
0241   21910C       732          LD      HL,RAM2K_VALUE_SET
0244   1885         733          JR      INI6
                    734  ;
                    735  ;*****************************************************************
                    736  ; Function: Same as SCAN2 including BEEP effect.
                    737  ; Input: Same as SCAN2
                    738  ; Output: Same as SCAN2
                    739  ; Reg effected: AF BC DE HL AF' BC' DE' HL'.
                    740  ; Call: SCAN2 BEEP.
                    741
                    742  SCAN:
0246   CD4D02       743          CALL    SCAN2
0249   CD0308       744          CALL    BEEP
024C   C9           745          RET
                    746
                    747  ;*****************************************************************
                    748  ; Function: Scan the keyboard and display.  Loop until
                    749  ;           a key is detected.  If the some key is already
                    750  ;           pressed when this routine starts execution,
                    751  ;           return when next key is entered.
                    752  ; Input: IX points to the buffer contains display patterns.
                    753  ;        20 digits require 40 bytes of data.  (IX) contains
                    754  ;        the pattern for the leftmost digit, (IX+39) contains
```

```
                    755  ;        the pattern for the rightmost digit.
                    756  ; Output: internal code of the key pressed.
                    757  ; Destroyed reg. : AF, BC, HL, AF', BC', DE',HL'.
                    758  ;                  All other registers except IY are also
                    759  ;                  changed during execution, but they are
                    760  ;                  restored before return.
                    761  ; Call: SCAN1
                    762
                    763  SCAN2:
024D   DDE5         764          PUSH    IX
024F   21F1FE       765          LD      HL,TEST
0252   CB7E         766          BIT     7,(HL)   ;This bit is set if user
                    767                           ;has entered illegal key.  The
                    768                           ;display will be disabled as
                    769                           ;a warning to the user.  This
                    770                           ;is done by replacing the display
                    771                           ;buffer pointer IX by BLANK.
0254   2804         772          JR      Z,SCPRE
0256   DD21D06F     773          LD      IX,BLANK
                    774
                    775  ; Wait until all keys are released for 47 ms.
                    776  ; (The execution time of SCAN1 is 15.7 ms,
                    777  ; 47 = 15.7 * 3).
                    778
025A   0603         779  SCPRE:  LD      B,3
025C   CD9B02       780  SCNX:   CALL    SCAN1            ;Get position code.
025F   30F9         781          JR      NC,SCPRE         ;If any key is pressed,
                    782                                   ;reload the debounce counter
                    783                                   ;B by 3.
0261   10F9         784          DJNZ    SCNX
0263   CBBE         785          RES     7,(HL)           ;Clear error flag.
0265   DDE1         786          POP     IX               ;Restore original IX.
                    787
                    788  ; Loop until any key is pressed.
                    789
0267   CD9B02       790  SCLOOP: CALL    SCAN1
026A   38FB         791          JR      C,SCLOOP
                    792
                    793  ; Convert the key-position-code returned by SCAN1 to
                    794  ; ASC II code.  This is done by table-lookup.
                    795  ; The table used is KEYTAB.
                    796
                    797  KEYMAP:
026C   21520B       798          LD      HL,KEYTAB
026F   4F           799          LD      C,A
0270   0600         800          LD      B,0
0272   09           801          ADD     HL,BC
0273   7E           802          LD      A,(HL)
0274   F5           803          PUSH    AF
0275   DB82         804          IN      A,(DIG3)
0277   CB67         805          BIT     4,A
0279   280A         806          JR      Z,KSHIFT         ;SHIFT KEY ?
027B   CB6F         807          BIT     5,A
027D   2802         808          JR      Z,KCTRL          ;CONTROL KEY?
027F   F1           809          POP     AF
0280   C9           810          RET
                    811
                    812  ;Executed by depressing any key with CTRL key together.
```

```
                    813  ;The key code is one byte stored in A register.
                    814
                    815  KCTRL:
0281   F1           816          POP     AF
0282   CBB7         817          RES     6,A
0284   C9           818          RET
                    819
                    820  ;Executed by depressing any key with SHIFT key together.
                    821  ;The key code is one byte stored in A register.
                    822
                    823  KSHIFT:
0285   F1           824          POP     AF
0286   D62C         825          SUB     2CH
0288   38DD         826          JR      C,SCLOOP
028A   FE25         827          CP      25H
028C   30D9         828          JR      NC,SCLOOP
028E   21380C       829          LD      HL,SHIFTT
0291   4F           830          LD      C,A
0292   0600         831          LD      B,0
0294   09           832          ADD     HL,BC
0295   7E           833          LD      A,(HL)
0296   FEFF         834          CP      0FFH
0298   28CD         835          JR      Z,SCLOOP        ;Zero, if illegal key in.
029A   C9           836          RET
                    837  ;
                    838  ;*****************************************************************
                    839  ; Function: Scan keyboard and display one cycle.
                    840  ;           Total execution time is about 16 ms (exactly
                    841  ;           15.7 ms, 28040 clock states @ 1.79 MHz).
                    842  ; Input: Same as SCAN.
                    843  ; Output:  i) no key during one scan
                    844  ;                  Carry flay -- 1
                    845  ;         ii) key pressed during one scan
                    846  ;                  Carry flag -- 0,
                    847  ;                  A -- position code of the key pressed.
                    848  ;                       If more than one key is pressed, A
                    849  ;                       contains the largest position-code.
                    850  ;                       (This key is the last key scanned.)
                    851  ; Destroyed reg: AF, AF', BC', DE', HL'. (see comments on SCAN)
                    852  ; Call: none.
                    853
                    854  SCAN1:
                    855  ;In hardware, the display and keyboard are
                    856  ;arranged as a 20 by 3 matrix.  Each cloumn
                    857  ;corresponds to one digit and three key buttons.
                    858  ;In normal operation, at most one column is
                    859  ;active.  The pattern of the active digit is the
                    860  ;data output on port A,B,C of 8255 I.  The data input
                    861  ;from bit 0-2~5 on port C of 8255 II are the status of key
                    862  ;buttons in the active column.  All signals on
                    863  ;I/O port are active low.
                    864
029B   37           865          SCF
029C   08           866          EX      AF,AF'
029D   D9           867          EXX
                    868
                    869  ;Carry flag of F' is used to return the status of
                    870  ;the keyboard.  If any key is pressed during one
```

```
                        871  ;scan, the flag is reset; otherwise, it is set.
                        872  ;Initially, this flag is set.  A' register is used
                        873  ;to store the position-code of the key pressed.
                        874  ;In this routine, 60 key positions are checked one
                        875  ;by one.  C register contains the code of the key
                        876  ;being checked.  The value of C is 0 at the beginning,
                        877  ;and is increased by 1 after each check.  So the code
                        878  ;ranges from 0 to 3BH (total 60 positions).  On each
                        879  ;check, if the input bit is 0 (key pressed), C register
                        880  ;is copied into A'.  The carry flag of F' is set also.
                        881  ;When some key is detected, the key positions after
                        882  ;this key will still be checked.  So if more than
                        883  ;one key are pressed during one scan, the code of the
                        884  ;last one will be returned.
029E   0E00             885          LD      C,0
02A0   11FEFF           886          LD      DE,0FFFEH       ;Activate the first digit.
02A3   6A               887          LD      L,D
02A4   2614             888          LD      H,20            ;20 digits.
02A6   DD7E00           889  KCOL:   LD      A,(IX)
02A9   D390             890          OUT     (SEG1),A        ;First byte pattern.
02AB   DD23             891          INC     IX
02AD   DD7E00           892          LD      A,(IX)
02B0   D391             893          OUT     (SEG2),A        ;2nd byte pattern.
02B2   7B               894          LD      A,E
02B3   D380             895          OUT     (DIG1),A        ;1-8 digitS
02B5   7A               896          LD      A,D
02B6   D381             897          OUT     (DIG2),A        ;9-16 digitS
02B8   7D               898          LD      A,L
02B9   D382             899          OUT     (DIG3),A        ;17-20 digitS
02BB   0650             900          LD      B,COLDEL
02BD   10FE             901          DJNZ    $               ;Delay 1.5 ms per digit.
02BF   D5               902          PUSH    DE
02C0   0603             903          LD      B,3             ;Each colomn has three keys.
02C2   DB92             904          IN      A,(KIN)         ; Now,bit 0-2 of A contains the
                        905                                  ;status of the three keys in
                        906                                  ;the activae colomn.
02C4   57               907          LD      D,A
02C5   CB1A             908  KROW:   RR      D               ; Rotate D 1 bit right
                        909                                  ; bit 0 of D will be rotate
                        910                                  ; into carry flag.
02C7   3802             911          JR      C,NOKEY         ;Skip next 2 instruction
                        912                                  ;if the key is not pressed.
                        913                                  ;The next 2 instructions
                        914                                  ;store the current position-code
                        915                                  ;into A' and reset carry flag
                        916                                  ;of F' register.
02C9   79               917          LD      A,C
02CA   08               918          EX      AF,AF'
02CB   0C               919  NOKEY:  INC     C               ;Increase current key code by 1.
02CC   10F7             920          DJNZ    KROW            ;Loop until 3 keys in the active
                        921                                  ;colomns are all checked.
02CE   3EFF             922          LD      A,0FFH          ;Disable all the digits.
02D0   D380             923          OUT     (DIG1),A
02D2   D381             924          OUT     (DIG2),A
02D4   D382             925          OUT     (DIG3),A
02D6   DD23             926          INC     IX
02D8   D1               927          POP     DE
02D9   A7               928          AND     A
```

```
02DA   CB03        929          RLC     E
02DC   3802        930          JR      C,RL1
02DE   CBC3        931          SET     0,E
02E0   CB12        932  RL1:    RL      D
02E2   3802        933          JR      C,RL2
02E4   CBC2        934          SET     0,D
02E6   CB15        935  RL2:    RL      L
02E8   25          936          DEC     H
02E9   20BB        937          JR      NZ,KCOL
02EB   11D8FF      938          LD      DE,-40
02EE   DD19        939          ADD     IX,DE           ;Get original IX.
02F0   D9          940          EXX
02F1   08          941          EX      AF,AF'
02F2   C9          942          RET
                   943
                   944  ;*****************************************************************
                   945
                   946  ;Exeuted when 'M' key is pressed.
                   947  ;  Enter the hexadecimal address of the first of the
                   948  ;four memory locations to be displayed.
                   949  ; (1) Type <CR> -- Display specified memory contents.
                   950  ; (2) Type :    -- Alter memory contents.
                   951  ; (3) Type .    -- Memory dump.
                   952  ; (4) Type /    -- Move data block from one area to another.
                   953
                   954  MEMEXC:
02F3   CD1603      955          CALL    MEMEX2
02F6   3E10        956          LD      A,10H
02F8   3280FF      957          LD      (TYPEFG),A      ;Set memory type.
02FB   1A          958          LD      A,(DE)
02FC   FE3A        959          CP      3AH             ;:
02FE   CA3703      960          JP      Z,MMODFY
0301   FE2E        961          CP      2EH             ;.
0303   283E        962          JR      Z,MDUMP1
0305   FE2F        963          CP      2FH             ;/
0307   2847        964          JR      Z,MMOVE
0309   CD2803      965          CALL    MEMEX3          ;Display specified memory
                   966                                  ;contents.
030C   CD890A      967          CALL    HEXX
030F   CD8603      968  P102    CALL    MEM3
0312   CD8109      969          CALL    CR2
0315   C9          970          RET
                   971
                   972  ;*****************************************************************
                   973
                   974  ;The input data must be hexadecimal values. MPF_IP will
                   975  ;ignore this command if there exists at least one digit
                   976  ;which is not a hexadecimal value. The user can use the
                   977  ;BACKSPACE key to correct the data.
                   978
                   979  MEMEX2:
0316   CD6008      980          CALL    ECHO_CH         ;Echo the input character and
                   981                                  ;prompt.
0319   CD6F06      982  MEMEX1  CALL    GET             ;Get a string of characters
                   983                                  ;and end the input with <CR>.
031C   CD5408      984          CALL    CHKINP          ;Check hexadecimal values.
031F   38F8        985          JR      C,MEMEX1        ;Jump to MEMEX1 if the input
                   986                                  ;data is illegal.
```

```
0321  CDDF08        987          CALL    CHKHEX          ;Get the hexadecimal address
                    988                                  ;of the first of four memory
                    989                                  ;locations to be displayed.
0324  22F8FE        990          LD      (MADDR),HL
0327  C9            991          RET
                    992
                    993
                    994  MEMEX3:                         ;Reset the counters of INPUT
                    995                                  ; BUFFER and DISPBF BUFFER.
0328  E5            996          PUSH    HL
0329  2108FF        997          LD      HL,INPBF+4
032C  2282FF        998          LD      (OUTPTR),HL
032F  2134FF        999          LD      HL,DISPBF+8
0332  2284FF       1000          LD      (DISP),HL
0335  E1           1001          POP     HL
0336  C9           1002          RET
                   1003
                   1004  ;*****************************************************************
                   1005
                   1006  ;If you want to change the values in these location,
                   1007  ;just type a colon and the values separated by spaces.
                   1008  ;The final command look like this:
                   1009  ;<M>=<start>:<data1> <data2> <data3><CR>
                   1010
                   1011  MMODFY:
                   1012  SET:
0337  E5           1013          PUSH    HL
0338  CDE508       1014          CALL    GETHL           ;Get data.
033B  E1           1015          POP     HL
033C  77           1016          LD      (HL),A
033D  CA8509       1017          JP      Z,CR3
0340  23           1018          INC     HL
0341  18F4         1019          JR      SET
                   1020
                   1021  ;*****************************************************************
                   1022
                   1023  ;Type the first address,followed by a period and the second
                   1024  ;address .This two-address-separated-by-a-period form is
                   1025  ;called a memory range.
                   1026  ;If PRT_MPF exists ,then it will print out the data,
                   1027  ;otherwise MPF_IP will ignore this command
                   1028  ;The final command look like this:
                   1029  ;<M>=<start>.<end><CR>    or
                   1030  ;<M>=<start>.<end> <linking address><CR>
                   1031
                   1032  MDUMP1:
0343  CDA308       1033          CALL    PTEST
0346  C0           1034          RET     NZ
0347  3E30         1035          LD      A,30H           ;Set memory dump type.
0349  320000     X 1036          LD      (TEST5),A
034C  CD0000     X 1037          CALL    MDUMP
034F  C9           1038          RET
                   1039
                   1040  ;*****************************************************************
                   1041
                   1042  ;You can treat a range of memory (specifed by two address
                   1043  ;separated by a slash),move it from one place to another
                   1044  ;in memory by using the MOVE command.
```

```
                    1045  ;The final command look like this:
                    1046  ;<M>=<start>/<end> <destination><CR>
                    1047
                    1048  MMOVE:
0350   22D0FE       1049          LD      (STEPBF),HL     ;The starting address in HL.
0353   CDE508       1050          CALL    GETHL           ;Get the ending address.
0356   22D2FE       1051          LD      (STEPBF+2),HL
0359   CDE508       1052          CALL    GETHL           ;Get the destination address.
035C   22D4FE       1053          LD      (STEPBF+4),HL
035F   CD6503       1054          CALL    GMV
0362   C38509       1055          JP      CR3
                    1056
                    1057  ;*****************************************************************
                    1058
                    1059  GMV:
0365   21D0FE       1060          LD      HL,STEPBF
0368   CDAF07       1061          CALL    GETP            ;Load parameters from
                    1062                                  ;step buffer into registers.
                    1063                                  ;Also check if the parameters
                    1064                                  ;are legal. After GETP,
                    1065                                  ;HL = start address of source
                    1066                                  ;BC = length to MOVE.
036B   DAC406       1067          JP      C,ERROR         ;Jump to ERROR if the
                    1068                                  ;parameters are illegal.(i.e.,
                    1069                                  ;ending address < starting address.)
036E   ED5BD4FE     1070          LD      DE,(STEPBF+4)   ;Load destination
                    1071                                  ;address into DE.
0372   ED52         1072          SBC     HL,DE           ;Compare HL and DE to
                    1073                                  ;determine to move up or down.
0374   300B         1074          JR      NC,MVUP
                    1075                                  ;Move down
0376   EB           1076          EX      DE,HL           ;HL = destination address.
0377   09           1077          ADD     HL,BC           ;HL = dest.address+length
0378   2B           1078          DEC     HL              ;HL = end address of dest.
0379   EB           1079          EX      DE,HL           ;DE = end address of dest.
037A   2AD2FE       1080          LD      HL,(STEPBF+2)   ;HL = end address of source.
037D   EDB8         1081          LDDR                    ;Block transter instruction.
037F   13           1082          INC     DE              ;DE = last address moved.
0380   C9           1083          RET
                    1084  MVUP:                           ;Move up
0381   19           1085          ADD     HL,DE           ;HL is changed by
                    1086                                  ;SBC HL,DE. Restore HL.
0382   EDB0         1087          LDIR                    ;Block transter
0384   1B           1088          DEC     DE              ;DE = last address moved.
0385   C9           1089          RET
                    1090
                    1091  ;*****************************************************************
                    1092
                    1093  ;To display four consecutive memory contents.
                    1094
                    1095  MEM3:
0386   2AF8FE       1096          LD      HL,(MADDR)
0389   0604         1097          LD      B,4
                    1098  MEM5:
038B   CD950A       1099          CALL    SPACE1          ;Insert a space.
038E   7E           1100          LD      A,(HL)
038F   CD9A0A       1101          CALL    HEX2
0392   23           1102          INC     HL
```

```
0393   10F6        1103          DJNZ    MEM5
                   1104  DECDSP:                          ;Clear the rightmost display
                   1105                                   ;pattern.
                   1106                                   ;in DISPLAY BUFFER.The display
                   1107                                   ;pattern is usually a cursor.
0395   DD212CFF    1108          LD      IX,DISPBF
0399   3EFF        1109  DEC_SP  LD      A,0FFH
039B   2A84FF      1110          LD      HL,(DISP)
039E   77          1111          LD      (HL),A
039F   23          1112          INC     HL
03A0   77          1113          LD      (HL),A
03A1   C9          1114          RET
                   1115
                   1116  ;*************************************************************
                   1117
                   1118  ;Executed when UP or down arrow key is pressed.
                   1119  ;Executed in memory mode only.
                   1120
                   1121  MFOR:                            ;Display next four memory
                   1122                                   ;contents.
03A2   2AF8FE      1123          LD      HL,(MADDR)
03A5   23          1124          INC     HL
03A6   23          1125          INC     HL
03A7   23          1126          INC     HL
03A8   23          1127          INC     HL
03A9   22F8FE      1128  P101    LD      (MADDR),HL
03AC   3E4D        1129          LD      A,'M'
03AE   CD6008      1130          CALL    ECHO_CH          ;Get pattern '<M>='
03B1   CD890A      1131          CALL    HEXX
03B4   C30F03      1132          JP      P102
                   1133  MBACK:                           ;Display last four memory
                   1134                                   ;contents.
03B7   2AF8FE      1135          LD      HL,(MADDR)
03BA   2B          1136          DEC     HL
03BB   2B          1137          DEC     HL
03BC   2B          1138          DEC     HL
03BD   2B          1139          DEC     HL
03BE   18E9        1140          JR      P101
                   1141
                   1142  ;*************************************************************
                   1143
                   1144  ;Executed when 'F' key is pressed.
                   1145  ;Store the data byte into all memory locations from
                   1146  ;addl to addr2.
                   1147  ;The final command look like this:
                   1148  ;<F>=<addr1> <addr2> <data><CR>
                   1149
                   1150  FILLDA:
03C0   CD1603      1151          CALL    MEMEX2           ;Get starting address.
03C3   CD1908      1152          CALL    RAMCHK
03C6   C2C406      1153          JP      NZ,ERROR         ;Jump to ERROR if the
                   1154                                   ;memory location of the
                   1155                                   ;starting address is not RAM.
03C9   E5          1156          PUSH    HL
03CA   CDE508      1157          CALL    GETHL            ;Get ending address.
03CD   E5          1158          PUSH    HL
03CE   CDE508      1159          CALL    GETHL            ;Get data.
03D1   7D          1160          LD      A,L
```

```
03D2  A7          1161          AND     A
03D3  E1          1162          POP     HL
03D4  D1          1163          POP     DE
03D5  12          1164          LD      (DE),A
03D6  ED52        1165          SBC     HL,DE
03D8  C8          1166          RET     Z
03D9  DAC406      1167          JP      C,ERROR         ;Jump to ERROR if starting
                  1168                                  ; address > ending address.
03DC  44          1169          LD      B,H
03DD  4D          1170          LD      C,L
03DE  62          1171          LD      H,D
03DF  6B          1172          LD      L,E
03E0  13          1173          INC     DE
03E1  EDB0        1174          LDIR
03E3  C38509      1175          JP      CR3
                  1176
                  1177  ;*****************************************************************
                  1178
                  1179  ;Exeuted when 'I' key is pressed.
                  1180  ;MPF_IP will display the current limit address.
                  1181  ;SYSTEM RAM data that of course can not be shifted
                  1182  ;so we must set the limit address of INSERT command.
                  1183  ;When one byte is inserted at some address,all
                  1184  ;data below this address will be shifted down one
                  1185  ;position.
                  1186  ;The last location will be shifted out and therefore lost.
                  1187
                  1188  ; (1) Type <CR> -- To see the current limit address
                  1189  ;                   of INSERT command.
                  1190  ; (2) Type C    -- To clear limit address (i.e.,set
                  1191  ;                   limit address to be 0FE00H).
                  1192  ; (3) Enter the hexadecimal address -- To set new high
                  1193  ;                   limit address of INSERT command.
                  1194
                  1195  ;When MPF_IP display <I>=
                  1196  ; You can enter the hexadecimal address and values separated
                  1197  ;                   by spaces -- To insert a block of data.
                  1198  ;The final command look like this:
                  1199  ;<I>=<address> <data1> <data2> <data3> <data4><CR>
                  1200
                  1201
                  1202  INSET:
03E6  CD1504      1203          CALL    INSET4
                  1204  INSET3:
03E9  22D0FE      1205          LD      (STEPBF),HL
03EC  23          1206          INC     HL
03ED  22D4FE      1207          LD      (STEPBF+4),HL
03F0  ED5BEBFE    1208          LD      DE,(END_ADDR)
03F4  1B          1209          DEC     DE
03F5  ED53D2FE    1210          LD      (STEPBF+2),DE
03F9  13          1211          INC     DE
03FA  A7          1212          AND     A
03FB  ED52        1213          SBC     HL,DE
03FD  D2C406      1214          JP      NC,ERROR
0400  CDE508      1215          CALL    GETHL
0403  F5          1216          PUSH    AF
                  1217
                  1218  ;Routine GMV needs 3 parameters which are stored  in
```

```
                    1219  ;step-buffer (STEPBF):
                    1220  ;STEPBF  : starting address (2 bytes).
                    1221  ;STEPBF+2:   ending address (2 bytes).
                    1222  ;STEPBF+4:destination address (2 bytes).
                    1223
0404    CD6503      1224          CALL    GMV
0407    2AD4FE      1225          LD      HL,(STEPBF+4)
040A    77          1226          LD      (HL),A
040B    2AD0FE      1227          LD      HL,(STEPBF)
040E    23          1228          INC     HL
040F    F1          1229          POP     AF
0410    CA8509      1230          JP      Z,CR3
0413    18D4        1231          JR      INSET3
                    1232  INSET4:
0415    F5          1233          PUSH    AF
0416    CD6008      1234          CALL    ECHO_CH         ;Echo the input character with
                    1235                                  ;<?>=
                    1236                                  ;? is I or D according to INSERT
                    1237                                  ;command or DELETE command
                    1238                                  ;respectively.
0419    2AEBFE      1239          LD      HL,(END_ADDR)   ;Get the current limit address.
041C    CD890A      1240          CALL    HEXX
041F    3E2F        1241          LD      A,2FH           ;/
0421    CD2409      1242          CALL    CHRWR   ; '<I>=< current limit address>/^
0424    37          1243          SCF
0425    3F          1244          CCF
                    1245  INSET1:
0426    CD6F06      1246          CALL    GET             ; Get a string of characters
                    1247                                  ; end the input with <CR>.
0429    3A0DFF      1248          LD      A,(INPBF+9)
042C    2812        1249          JR      Z,INSET2        ;For <CR> condition.
042E    FE43        1250          CP      'C'
0430    CC1702      1251          CALL    Z,CLRI          ;For C condition.
0433    280B        1252          JR      Z,INSET2
0435    2108FF      1253          LD      HL,INPBF+4
0438    CDE208      1254          CALL    CHKHE2          ;Get new limit address.
043B    38E9        1255          JR      C,INSET1        ;Jump to INSET1 if the input
                    1256                                  ;datas is illegal.
043D    22EBFE      1257          LD      (END_ADDR),HL
                    1258  INSET2:
0440    CD8509      1259          CALL    CR3             ;Print message.
0443    F1          1260          POP     AF
0444    CD1603      1261          CALL    MEMEX2
0447    C9          1262          RET
                    1263
                    1264  ;*************************************************************
                    1265
                    1266  ; Executed when 'D' key is pressed.
                    1267  ; MPF_IP will display the current limit address of DELETE
                    1268  ; command.
                    1269  ; Avoid to changing the contents in SYSTEM RAM we must
                    1270  ; set the limit address.
                    1271  ; The default value of limit address is 0FE00H.
                    1272
                    1273  ; (1) Type <CR> -- To see the current limit address.
                    1274  ; (2) Type C    -- To clear limit address (i.e.,set the
                    1275  ;                    limit address to be 0FE00H).
                    1276  ; (3) Enter the hexadecimal address -- To set the new
```

```
                    1277                        limit address.
                    1278
                    1279  ; When MPF_IP display <D>= you can enter the hexadecimal
                    1280  ; address which content you want to be deleted.
                    1281  ; You can delete one byte from memory at a time.
                    1282
                    1283  DELETE:
0448  CD1504        1284          CALL    INSET4
044B  22D4FE        1285          LD      (STEPBF+4),HL
044E  ED5BEBFE      1286          LD      DE,(END_ADDR)
0452  ED53D2FE      1287          LD      (STEPBF+2),DE
0456  23            1288          INC     HL
0457  22D0FE        1289          LD      (STEPBF),HL
045A  CD6503        1290          CALL    GMV
045D  AF            1291          XOR     A
045E  12            1292          LD      (DE),A
045F  C38509        1293          JP      CR3
                    1294
                    1295  ;*****************************************************************
                    1296
                    1297  ; Executed when 'J' key is pressed.
                    1298  ; Instruction JR and DJNZ requires relative addresses.
                    1299  ; MPF_IP supports the calculation of relative addresses
                    1300  ; throught the 'J' command.
                    1301
                    1302  JUMP:
0462  CD1603        1303          CALL    MEMEX2          ;Get starting address.
0465  E5            1304          PUSH    HL
0466  CDE508        1305          CALL    GETHL           ;Get destination address.
0469  D1            1306          POP     DE              ;Load starting address
                    1307                                  ;into DE.
046A  13            1308          INC     DE              ;Increase this address by 2.
                    1309                                  ;Relative address is used in
                    1310                                  ;instruction JR and DJNZ.
                    1311                                  ;The codes for them are 2 bytes.
                    1312                                  ;The PC is increased by 2 after
                    1313                                  ;opcode is fetched.
046B  13            1314          INC     DE
046C  B7            1315          OR      A
046D  ED52          1316          SBC     HL,DE           ;Load destination
                    1317                                  ;address into HL.
046F  7D            1318          LD      A,L             ;Check if the offset is between
                    1319                                  ;+127 (007FH) and -128 (FF80H).
                    1320                                  ;If the offset is positive, both
                    1321                                  ;H and bit 7 of L must be zero;
                    1322                                  ;if it is negtive, H and bit 7 of
                    1323                                  ;L must be FF and 1. In both cases
                    1324                                  ;adding H with bit 7 of L results
                    1325                                  ;in 0.
                    1326                                  ;Rotote bit 7 of L into carry
                    1327                                  ;flag.
0470  17            1328          RLA
0471  7C            1329          LD      A,H
0472  CE00          1330          ADC     A,0             ;Add H and bit 7 of L.
0474  C2C406        1331          JP      NZ,ERROR        ;Branch to ERROR if
                    1332                                  ;the results is nonzero.
0477  7D            1333          LD      A,L
0478  1B            1334          DEC     DE
```

```
0479  12          1335          LD      (DE),A          ;Save the offset into
                  1336                                  ;the next byte of opcode.
                  1337                                  ;(JR or DJNZ)
047A  C38509      1338          JP      CR3
                  1339
                  1340  ;*************************************************************
                  1341
                  1342  ;Executed when 'B' key is pressed.
                  1343  ;The MPF_IP will display the current address of breakpoint.
                  1344  ; (1) Type <CR> -- To see the currently assigned breakpoint
                  1345  ;                          address.
                  1346  ; (2) Type C    -- To clear breakpoint.
                  1347  ; (3) Enter the hedadecimal address -- To set new breakpoint.
                  1348
                  1349  BREAK:
047D  CD6008      1350          CALL    ECHO_CH ;Echo the input character with <B>=
0480  2AEDFE      1351  DISBR:  LD      HL,(BRAD)
0483  CD890A      1352          CALL    HEXX    ;Display the current assigned breakpoint.
0486  3E2F        1353          LD      A,2FH   ;/
0488  CD2409      1354          CALL    CHRWR   ; <B>=< current breakpoint address >/^
048B  37          1355          SCF
048C  3F          1356          CCF
                  1357  BREAK1:
048D  CD6F06      1358          CALL    GET
0490  3A0DFF      1359          LD      A,(INPBF+9)
0493  2812        1360          JR      Z,B3            ;For <CR> condition.
0495  FE43        1361          CP      'C'             ; For C condition.
0497  CC1002      1362          CALL    Z,CLRB
049A  280B        1363          JR      Z,B3
049C  2108FF      1364          LD      HL,INPBF+4
049F  CDE208      1365          CALL    CHKHE2          ;Get new breakpoint address
                  1366                                  ;stored into HL.
04A2  38E9        1367          JR      C,BREAK1        ;Jump to BREAK1 if the datas
                  1368                                  ;are not hexadecimal values.
04A4  22EDFE      1369          LD      (BRAD),HL
                  1370  B3:
04A7  2AEDFE      1371          LD      HL,(BRAD)
04AA  7E          1372          LD      A,(HL)
04AB  32EFFE      1373          LD      (BRDA),A
04AE  C38509      1374          JP      CR3
                  1375
                  1376  ;*************************************************************
                  1377
                  1378  ;Executed when 'S' key is pressed.
                  1379  ;Execution at specified address or current address.
                  1380
                  1381  STEP:
04B1  47          1382          LD      B,A
04B2  3AF2FE      1383          LD      A,(STEPFG)
04B5  A7          1384          AND     A
04B6  78          1385          LD      A,B
04B7  2006        1386          JR      NZ,P111         ;If zero,then execute at the
                  1387                                  ;current address.
04B9  CD1603      1388          CALL    MEMEX2          ;Get the specified address.
04BC  229EFF      1389          LD      (USERPC),HL
04BF  3EEF        1390  P111    LD      A,11101111B     ;This data will be output to
                  1391                                  ;port C of 8255 II to enable BREAK.
                  1392                                  ;It is done by routine PREOUT.
```

```
04C1   181C       1393          JR      PREOUT
                  1394
                  1395  ;
                  1396  ;*****************************************************************
                  1397
                  1398  ;Executed when 'G' key is pressed.
                  1399  ;Execution at specified address or current address.
                  1400  ;The following routine is the service routine for
                  1401  ;'GO' key.
                  1402
                  1403  GOEXEC:
04C3   CD6008     1404          CALL    ECHO_CH         ;Echo the input character with <G>=
04C6   CD6F06     1405  GOEXE1  CALL    GET             ;Get a string of characters
                  1406                                  ;end the input with <CR> .
04C9   3A08FF     1407          LD      A,(INPBF+4)
04CC   FE0D       1408          CP      0DH
04CE   2808       1409          JR      Z,EXEC2         ;If zero ,then execute at
                  1410                                  ;the current address.
04D0   CDDF08     1411          CALL    CHKHEX          ;Get specified address.
04D3   38F1       1412          JR      C,GOEXE1        ;Jump to GOEXE1 if one of
                  1413                                  ;the input datum is illegal.
04D5   229EFF     1414          LD      (USERPC),HL
                  1415  EXEC2:
04D8   2AEDFE     1416          LD      HL,(BRAD)       ;Get the address of breakpoint.
04DB   36EF       1417          LD      (HL),0EFH       ;Instruction RST 28H.
                  1418                                  ;The content of break address
                  1419                                  ;is changed to RST 28H before
                  1420                                  ;the control is transfered to
                  1421                                  ;user's program.  This
                  1422                                  ;will cause a trap when user's
                  1423                                  ;PC passes this point.
04DD   3EFF       1424          LD      A,0FFH          ;Save FF into TEMP1. This data
                  1425                                  ;will be output to port C later.
                  1426                                  ;FF is used to disable breakpoint.
04DF   32FAFE     1427  PREOUT: LD      (TEMP1),A
04E2   3EA5       1428          LD      A,0A5H
04E4   32F2FE     1429          LD      (STEPFG),A
04E7   3AA0FF     1430          LD      A,(USERIF)      ;Save two instructions into
                  1431                                  ;TEMP and TEMP+1.  These two
                  1432                                  ;instructions will be executed
                  1433                                  ;later.  If the user's IFF
                  1434                                  ;(interrupt flip-flop) is 1,
                  1435                                  ;the instructions are 'EI RET'.
                  1436                                  ;Otherwise, they are 'DI RET'.
04EA   CB47       1437          BIT     0,A
04EC   21FBC9     1438          LD      HL,0C9FBH       ;'EI','RET'
04EF   2002       1439          JR      NZ,EIDI
04F1   2EF3       1440          LD      L,0F3H          ;'DI'
                  1441  EIDI:
04F3   22FBFE     1442          LD      (TEMP1+1),HL
04F6   3188FF     1443          LD      SP,REGBF        ;Restore user's registers by
                  1444                                  ;setting SP to REGBF (register
                  1445                                  ;buffer) and continuously
                  1446                                  ;popping the stack.
04F9   F1         1447          POP     AF
04FA   C1         1448          POP     BC
04FB   D1         1449          POP     DE
04FC   E1         1450          POP     HL
```

```
04FD  08          1451          EX      AF,AF'
04FE  F1          1452          POP     AF
04FF  08          1453          EX      AF,AF'
0500  D9          1454          EXX
0501  C1          1455          POP     BC
0502  D1          1456          POP     DE
0503  E1          1457          POP     HL
0504  D9          1458          EXX
0505  DDE1        1459          POP     IX
0507  FDE1        1460          POP     IY
0509  ED7B9CFF    1461          LD      SP,(USERSP)     ;Restore user's SP .
050D  3289FF      1462          LD      (USERAF+1),A
0510  3AA1FF      1463          LD      A,(USERIF+1)    ;Restore user's I .
0513  ED47        1464          LD      I,A
0515  E5          1465          PUSH    HL              ;The next 3 instructions
                  1466                                  ;push the address being
                  1467                          ;displayed now (in USERPC)
                  1468                          ;onto stack without changing
                  1469                          ;HL register.  This address will be
                  1470                          ;treated as user's new PC.
0516  2A9EFF      1471          LD      HL,(USERPC)
0519  E3          1472          EX      (SP),HL
051A  3AFAFE      1473          LD      A,(TEMP1);Output the data stored in
                  1474                           ;TEMP1 to port C of 8255 II .
                  1475                           ;This data is prepared by
                  1476                           ;command STEP or GO .
                  1477                           ; In first case, it is
                  1478                           ;11101111 and will enable
                  1479                           ;break point.  In other
                  1480                           ;cases, it is FF and will
                  1481                           ;disable break point.
                  1482                           ;If break is enabled, non-
                  1483                           ;maskable interrupt will occur
                  1484                           ;5 M1's after the OUT instruction.
051D  D392        1485          OUT     (KIN),A
051F  3A89FF      1486          LD      A,(USERAF+1)    ;1st M1 .
0522  C3FBFE      1487          JP      TEMP1+1 ;2nd M1,
                  1488                          ;Execute the two instructions
                  1489                          ;stored in RAM.  They are:
                  1490                          ;    EI (or DI)     ;3rd M1
                  1491                          ;    RET            ;4th M1
                  1492                          ;The starting address of user's
                  1493                          ;program has been pushed onto
                  1494                          ;the top of the stack.  RET pops
                  1495                          ;out this address and transfers
                  1496                          ;control to it.  The first M1
                  1497                          ;of user's program will be the
                  1498                          ;5th M1 after OUT. If break point
                  1499                          ;is enabled, NMI will occur after
                  1500                          ;this instruction is completed.
                  1501                          ;This is the mechanism of single
                  1502                          ;step.
                  1503
                  1504  ;*************************************************************
                  1505
                  1506  ;The monitor reserves 26 locations in memory for the
                  1507  ;twenty-one registers as follows:
                  1508  ; AF BC DE HL AF' BC' DE' HL' IX IY SP PC I
```

```
                    1509
                    1510  ; Type <CR> -- Display registers (two pairs of registers).
                    1511  ; Type <register name><CR> -- Display registers (pair of
                    1512  ;                              registers)
                    1513  ; Type :    -- Alter register contents.
                    1514
                    1515  REGEXC:
0525  CD6008        1516          CALL    ECHO_CH ;Echo the input character with <R>=
0528  CD6F06        1517  REGEX2  CALL    GET             ;Get a string of characters
                    1518                                  ;end the input with <CR> .
052B  2107FF        1519          LD      HL,INPBF+3
052E  E5            1520          PUSH    HL
052F  23            1521          INC     HL
0530  CDE208        1522          CALL    CHKHE2
0533  E1            1523          POP     HL
0534  38F2          1524          JR      C,REGEX2        ;Jump to REGEX2 if one
                    1525                                  ;of the input datas is illegal.
0536  CDB108        1526          CALL    LDA
0539  110000        1527          LD      DE,0
053C  4A            1528          LD      C,D             ;Set C=0
053D  3E0D          1529          LD      A,0DH
053F  56            1530          LD      D,(HL)
0540  BA            1531          CP      D
0541  2825          1532          JR      Z,REGALL        ;Display the first four register
                    1533                                  ;contents (AF BC).
0543  23            1534          INC     HL
0544  7E            1535          LD      A,(HL)
0545  FE27          1536          CP      27H             ;Check ' condition.
                    1537                                  ;(i.e.,A' F' B' C' D' E' H' L')
0547  2003          1538          JR      NZ,P105
0549  23            1539          INC     HL
054A  4F            1540          LD      C,A
054B  7E            1541          LD      A,(HL)
054C  FE3A          1542  P105    CP      3AH             ;Check : condition.
                    1543                                  ;If zero ,then change the content
                    1544                                  ;of a single byte register
                    1545                                  ;(i.e.,AFBCDEHLA'F'B'C'D'E'H'L'I)
054E  CAFC05        1546          JP      Z,RMODFY
0551  FE0D          1547          CP      0DH
0553  281A          1548          JR      Z,RDSPL3        ;If zero ,then display two single
                    1549                                  ;byte register (A F A' F' I).
                    1550                                  ;or a register pair (BC DE HL
                    1551                                  ;BC' DE' HL').
                    1552                                  ;or a two byte registers.
                    1553                                  ;(IX IY SP PC)
0555  5E            1554          LD      E,(HL)
0556  23            1555          INC     HL
0557  7E            1556          LD      A,(HL)
0558  FE27          1557          CP      27H             ;Check ' condition.
                    1558                                  ;(i.e., AF' BC' DE' HL')
055A  2003          1559          JR      NZ,P106
055C  23            1560          INC     HL
055D  4F            1561          LD      C,A
055E  7E            1562          LD      A,(HL)
055F  FE3A          1563  P106    CP      3AH
0561  CA1206        1564          JP      Z,RMODF1        ;Change the contents of two
                    1565                                  ;single byte register.
                    1566                                  ;(AF AF' IF ).
```

```
                    1567                                    ;or a register pair (BC DE
                    1568                                    ;HL BC' DE' HL').
                    1569                                    ;or a two byte register (
                    1570                                    ;IX IY SP PC).
0564    FE0D        1571            CP      0DH
0566    280E        1572            JR      Z,RDSPLY        ;Display two single byte
                    1573                                    ;register or a register pair
                    1574                                    ;or a two byte register.
                    1575
                    1576    ;***************************************************************
                    1577
                    1578    REGALL:                         ;Display 'AF BC'
0568    110041      1579            LD      DE,4100H
                    1580
                    1581    RDSPL4:                         ;Display four bytes of register
                    1582                                    ;contents.
056B    0602        1583            LD      B,2
056D    1809        1584            JR      RDSPL0
                    1585    RDSPL3:
056F    7A          1586            LD      A,D
0570    FE49        1587            CP      'I'             ;Check I register.
0572    2002        1588            JR      NZ,RDSPLY
0574    1E46        1589            LD      E,46H
                    1590
                    1591    ;***************************************************************
                    1592
                    1593    ;Display two single byte register (AF AF' I) or
                    1594    ;a register pair ( BC DE HL BC' DE' HL' ) or
                    1595    ;a two byte register ( IX IY SP PC ).
                    1596
                    1597    RDSPLY:
0576    0601        1598            LD      B,1
                    1599    RDSPL0:
0578    C5          1600            PUSH    BC
0579    CD3406      1601            CALL    SEARC_REG
057C    C1          1602            POP     BC
057D    28E9        1603            JR      Z,REGALL        ;Jump to REGALL if the input
                    1604                                    ;register name is illegal.
057F    CD2803      1605            CALL    MEMEX3
0582    3A03FF      1606    RDISP6  LD      A,(RCOUNT)
0585    CB47        1607            BIT     0,A
0587    280A        1608            JR      Z,RDSPL1
0589    2B          1609            DEC     HL
058A    1B          1610            DEC     DE
058B    3A03FF      1611            LD      A,(RCOUNT)
058E    CB87        1612            RES     0,A             ;Registers are displayed by
                    1613                                    ;pair. Find the count of
                    1614                                    ;pair leader. (count of
                    1615                                    ;the lower one)
0590    3203FF      1616            LD      (RCOUNT),A
0593    CD950A      1617    RDSPL1  CALL    SPACE1          ;Insert a space.
0596    7E          1618            LD      A,(HL)          ;Get the first register name.
0597    CD2409      1619            CALL    CHRWR
059A    23          1620            INC     HL
059B    7E          1621            LD      A,(HL)          ;Get the second register name.
059C    CD2409      1622            CALL    CHRWR
059F    CD950A      1623            CALL    SPACE1          ;Insert a space.
05A2    13          1624            INC     DE
```

```
05A3   1A          1625          LD      A,(DE)          ;Get the first register content.
05A4   CD9A0A      1626          CALL    HEX2
05A7   1B          1627          DEC     DE
05A8   1A          1628          LD      A,(DE)          ;Get the second register content.
05A9   CD9A0A      1629          CALL    HEX2
05AC   23          1630          INC     HL
05AD   13          1631          INC     DE
05AE   13          1632          INC     DE
05AF   10E2        1633          DJNZ    RDSPL1
05B1   3E20        1634          LD      A,20H           ;Set register mode.
05B3   3280FF      1635          LD      (TYPEFG),A
05B6   C38C08      1636          JP      REG2
                   1637
                   1638  ;*****************************************************************
                   1639
                   1640  ;Executed when UP or DOWN arrow is pressed.
                   1641  ;Executed in register mode only.
                   1642
                   1643  RFOR:                   ;Display next four register contents.
05B9   3A03FF      1644          LD      A,(RCOUNT)
05BC   3C          1645          INC     A
05BD   3C          1646          INC     A
05BE   3C          1647          INC     A
05BF   3C          1648          INC     A
05C0   FE18        1649          CP      24
05C2   2834        1650          JR      Z,RBACK1
05C4   3030        1651          JR      NC,RBACK2
05C6   3203FF      1652  RFOR1   LD      (RCOUNT),A
05C9   3AF2FE      1653          LD      A,(STEPFG)      ;If the content of STEPBF is
                   1654                                  ;zero.
                   1655                                  ;it means MPF_IP executes STEP
                   1656                                  ; or GO command.
05CC   A7          1657          AND     A
05CD   280B        1658          JR      Z,RFOR2
05CF   CDB909      1659          CALL    CLEAR
05D2   2A9EFF      1660          LD      HL,(USERPC)
05D5   CD920A      1661          CALL    HEX4
05D8   1805        1662          JR      RFOR3
05DA   3E52        1663  RFOR2   LD      A,52H           ;Get pattern '<R>='
05DC   CD6008      1664          CALL    ECHO_CH
05DF   218E0B      1665  RFOR3   LD      HL,RTABLE
05E2   3A03FF      1666          LD      A,(RCOUNT)
05E5   85          1667          ADD     A,L
05E6   6F          1668          LD      L,A
05E7   56          1669          LD      D,(HL)
05E8   23          1670          INC     HL
05E9   5E          1671          LD      E,(HL)
05EA   C36B05      1672          JP      RDSPL4
                   1673  RBACK:                  ;Display last four register
                   1674                          ;contents.
05ED   3A03FF      1675          LD      A,(RCOUNT)
05F0   FE02        1676          CP      2
05F2   2804        1677          JR      Z,RBACK1
05F4   38D0        1678          JR      C,RFOR1
05F6   3D          1679  RBACK2  DEC     A
05F7   3D          1680          DEC     A
05F8   3D          1681  RBACK1  DEC     A
05F9   3D          1682          DEC     A
```

```
05FA   18CA        1683          JR      RFOR1
                   1684
                   1685  ;*****************************************************************
                   1686
                   1687  ;There are four kinds of register modify mode as folloes:
                   1688
                   1689  ; (1) <R>=H:< one byte data ><CR>
                   1690  ; (2) <R>=H':< one byte data ><CR>
                   1691  ; (3) <R>=HL:< two byte data ><CR>
                   1692  ; (4) <R>=HL':< two byte data ><CR>
                   1693
                   1694  RMODFY:
05FC   CD3406      1695          CALL    SEARC_REG
05FF   2830        1696          JR      Z,MEMDP3        ; Illegal register name.
0601   D5          1697          PUSH    DE
0602   CDE508      1698          CALL    GETHL
0605   D1          1699          POP     DE
0606   CB41        1700          BIT     0,C
0608   2805        1701          JR      Z,RODD
060A   1B          1702          DEC     DE
060B   12          1703  RMODF2  LD      (DE),A
060C   C38509      1704          JP      CR3
060F   13          1705  RODD    INC     DE
0610   18F9        1706          JR      RMODF2
0612   CD3406      1707  RMODF1  CALL    SEARC_REG
0615   281A        1708          JR      Z,MEMDP3        ; Illegal register name.
0617   CB41        1709          BIT     0,C
0619   2801        1710          JR      Z,RMODF3
061B   1B          1711          DEC     DE
061C   D5          1712  RMODF3  PUSH    DE
061D   CDE508      1713          CALL    GETHL
0620   D1          1714          POP     DE
0621   12          1715          LD      (DE),A
0622   13          1716          INC     DE
0623   7C          1717          LD      A,H
0624   12          1718          LD      (DE),A
0625   C38509      1719          JP      CR3
                   1720
                   1721  ;*****************************************************************
                   1722
                   1723  ; You can examine these registers when you STEP or GO
                   1724  ; a machine language.
                   1725
                   1726  MEMDP2:
0628   CDB909      1727          CALL    CLEAR
062B   2A9EFF      1728          LD      HL,(USERPC)
062E   CD920A      1729          CALL    HEX4
0631   C36805      1730  MEMDP3  JP      REGALL
                   1731
                   1732  ;*****************************************************************
                   1733
                   1734  ; Find bases of the register name and contents.
                   1735  ; Input :Register name (ASC II code) stored in DE.
                   1736  ; Output: HL -- Base of RTABLE (i.e.,point to register
                   1737  ;                               name beginning).
                   1738  ;         DE -- Base of REGBF  (i.e.,point to register
                   1739  ;                               buffer begining).
                   1740  ;          C -- Counts of register in RATBLE.
```

```
                        SEARC_REG:
2634  218E0B                      LD      HL,RTABLE
2637  C5                          PUSH    BC
2638  AF                          XOR     A
2639  4F                          LD      C,A
263A  7A                          LD      A,D
263B  0619                        LD      B,25
263D  BE                SERCH     CP      (HL)          ;Compare with the first
                                                        ;register name.
263E  2027                        JR      NZ,SERCH1
2640  7B                          LD      A,E
2641  A7                          AND     A
2642  2808                        JR      Z,SERCH2      ;Zero,if it is a single
                                                        ;byte register.
2644  23                          INC     HL
2645  0C                          INC     C
2646  BE                          CP      (HL)          ;Compare with the second
                                                        ;register name.
2647  2803                        JR      Z,SERCH2
2649  7A                          LD      A,D
264A  181D                        JR      SERCH4
                        SERCH2:
264C  79                          LD      A,C
264D  C1                          POP     BC
264E  47                          LD      B,A
264F  79                          LD      A,C
2650  FE27                        CP      27H           ;Check ' (A' F' B' C' D' E' H' L')
2652  48                          LD      C,B
2653  2008                        JR      NZ,SERCH3
2655  7D                          LD      A,L
2656  C608                        ADD     A,8
2658  6F                          LD      L,A
2659  79                          LD      A,C
265A  C608                        ADD     A,8
265C  4F                          LD      C,A
                        SERCH3:
265D  79                          LD      A,C
265E  3203FF            SERCH5    LD      (RCOUNT) ,A
2661  1188FF                      LD      DE,REGBF
2664  83                          ADD     A,E
2665  5F                          LD      E,A
2666  C9                          RET
2667  23                SERCH1    INC     HL
2668  0C                          INC     C
2669  10D2              SERCH4    DJNZ    SERCH
266B  AF                          XOR     A
266C  A7                          AND     A
266D  C1                          POP     BC
266E  C9                          RET           ;Zero if illegal register name exists.
                        ;
                        ;****************************************************************
                        ; Function: Refer to READLN.
                        ;               C -- Get a string of characters.
                        ;               NC-- Reset the content of INPTR.
                        ;
                        GET:
266F  DADA09                      JP      C,RDLOOP
```

```
0672   CDD409      1799  GETT    CALL    READLN
0675   CD9503      1800          CALL    DECDSP
0678   C9          1801          RET
                   1802  ;
                   1803  ;*****************************************************************
                   1804  ; TAPE WRITE:
                   1805
                   1806  DUMP:
0679   CD6008      1807          CALL    ECHO_CH         ;Echo the input character
                   1808                                  ;with <W>=
067C   CD6F06      1809  DUMP1   CALL    GET             ;Get a string of characters
                   1810                                  ;and end the input with <CR>
067F   CDDF08      1811          CALL    CHKHEX          ;Get starting address.
0682   38F8        1812          JR      C,DUMP1         ;Jump to DUMP1 if the input
                   1813                                  ;datum are illegal.
0684   22D4FE      1814          LD      (STEPBF+4),HL
0687   CDE508      1815          CALL    GETHL           ;Get ending address.
068A   38F0        1816          JR      C,DUMP1         ;Jump to DUMP1 if the input
                   1817                                  ;datas are illegal.
068C   22D6FE      1818          LD      (STEPBF+6),HL
068F   CDAE08      1819  DUMP2   CALL    GETCHR          ;Get tape filename.
0692   11D0FE      1820          LD      DE,STEPBF
0695   010400      1821          LD      BC,4
0698   EDB0        1822          LDIR
069A   CD9F07      1823          CALL    SUM1            ;Load parameters from
                   1824                                  ;step buffer into registers
                   1825                                  ;Check if the parameters
                   1826                                  ;are legal. If legal,calculate
                   1827                                  ;the sum of all data to be
                   1828                                  ;output to tape.
069D   3825        1829          JR      C,ERROR         ;Branch to ERROR if the
                   1830                                  ;parameters are illegal.(length
                   1831                                  ;is negative)
069F   32D8FE      1832          LD      (STEPBF+8),A    ;Store the ckecksum into
                   1833                                  ;STEPBF+8.
06A2   21A00F      1834          LD      HL,4000         ;Output 1K Hz square
                   1835                                  ;wave for 4000 cycles.
                   1836                                  ;Leading sync signal.
06A5   CD6E08      1837          CALL    TONE1K
06A8   21D0FE      1838          LD      HL,STEPBF       ;Output 27 bytes starting
                   1839                                  ;at STEPBF.(Include:
                   1840                                  ;filename ,starting,ending
                   1841                                  ;address and checksum and all
                   1842                                  ;the parameters of EDITOR and
                   1843                                  ;ASSEMBLER.
06AB   011B00      1844          LD      BC,27
06AE   CDBF07      1845          CALL    TAPEOUT
06B1   21A00F      1846          LD      HL,4000         ;Output 2K Hz square
                   1847                                  ;waves for 4000 cycles.
                   1848                                  ;Middle sync. The filename of
                   1849                                  ;the file being read will be
                   1850                                  ;displayed in the interval.
06B4   CD7208      1851          CALL    TONE2K
06B7   CDAC07      1852          CALL    GETPTR          ;Load parameters into
                   1853                                  ;registers(Starting,ending and
                   1854                                  ;length).
06BA   CDBF07      1855          CALL    TAPEOUT         ;Output user's data
06BD   21A00F      1856          LD      HL,4000         ;Output 4000 cycles
```

```
                    1857                                    ;of 2K Hz square wave.
                    1858                                    ;(Tail sync.)
06C0    CD7208      1859            CALL    TONE2K
06C3    C9          1860            RET
                    1861    ;
                    1862    ;*************************************************************
                    1863    ; Function: Print ERROR message.
                    1864    ; Input: None
                    1865    ; Output: Display patterns ' ERRORS' in display buffer.
                    1866    ;         (OUTPTR) <- INPBF+8
                    1867    ;         (DISP)   <- DISPBF+16
                    1868    ; Reg affected: AF HL  .
                    1869    ; Call: PRTMES
                    1870
                    1871    ERROR:
06C4    21AF0C      1872            LD      HL,ERRSMSG
06C7    C38608      1873            JP      PRTMES
                    1874    ;
                    1875    ;*************************************************************
                    1876    ; Function: TAPE READ .
                    1877
                    1878    LOAD:
06CA    CD6008      1879            CALL    ECHO_CH         ;Echo the input character
                    1880                                    ;with <L>=
06CD    CD6F06      1881            CALL    GET             ;Get a string of characters
                    1882                                    ;and end the input with <CR> .
06D0    3EBF        1883    LEAD:   LD      A,10111111B     ;Decimal point.
06D2    D391        1884            OUT     (SEG2),A        ;When secrching for filename
                    1885                                    ;the display is blank intially.
                    1886                                    ;If the data read from MIC is
                    1887                                    ;acceptable 0 or 1,the display
                    1888                                    ;becomes '              '
06D4    3EFF        1889            LD      A,0FFH
06D6    D390        1890            OUT     (SEG1),A
06D8    21E803      1891            LD      HL,1000
06DB    CD7A07      1892    LEAD1:  CALL    PERIOD          ;The return of PERIOD
                    1893                                    ;is in flag:
                    1894                                    ; NC -- tape input is 1K Hz
                    1895                                    ;  C -- otherwise
06DE    38F0        1896            JR      C,LEAD          ;Load until leading sync.
                    1897                                    ;is detected.
06E0    2B          1898            DEC     HL              ;Decrease HL by one when
                    1899                                    ;one period is detected.
06E1    7C          1900            LD      A,H
06E2    B5          1901            OR      L               ;Check if both H and L are zero.
06E3    20F6        1902            JR      NZ,LEAD1        ;Wait for 1000 periods.
                    1903                                    ;The leading sync  is accepted
                    1904                                    ;if it is longer than 1000
                    1905                                    ;cycles (1 second).
06E5    CD7A07      1906    LEAD2:  CALL    PERIOD
06E8    30FB        1907            JR      NC,LEAD2        ;Wait all leading sync to
                    1908                                    ;pass over.
06EA    21D0FE      1909            LD      HL,STEPBF       ;Load 27 bytes from
                    1910                                    ;tape into STEPBF.
06ED    011B00      1911            LD      BC,27
06F0    CD3B07      1912            CALL    TAPEIN
06F3    38DB        1913            JR      C,LEAD          ;Jump to LEAD if input
                    1914                                    ;is not sucessful.
```

```
06F5   0604        1915          LD      B,4             ;Get filename from DISPLAY BUFFER.
                   1916                                  ;The filename is consisted of 4
                   1917                                  ;alphanumeric characters.
06F7   213EFF      1918          LD      HL,DISPBF+18
06FA   2284FF      1919          LD      (DISP),HL
06FD   21D0FE      1920          LD      HL,STEPBF
0700   7E          1921  LOOP3   LD      A,(HL)
0701   CD2108      1922          CALL    CONVER
0704   23          1923          INC     HL
0705   10F9        1924          DJNZ    LOOP3
0707   0664        1925          LD      B,100           ;Display it for 1.57 sec.
0709   CD9B02      1926  FILEDP: CALL    SCAN1
070C   10FB        1927          DJNZ    FILEDP
070E   0604        1928          LD      B,4             ;Check if the input
                   1929                                  ;filename equals to the
                   1930                                  ;specified filenames.
0710   2B          1931          DEC     HL
0711   ED5B82FF    1932          LD      DE,(OUTPTR)
0715   1B          1933          DEC     DE
0716   1A          1934  LOOP4   LD      A,(DE)
0717   BE          1935          CP      (HL)
0718   2B          1936          DEC     HL
0719   1B          1937          DEC     DE
071A   20B4        1938          JR      NZ,LEAD         ;If not,find the leading
                   1939                                  ;sync of next filename.
071C   10F8        1940          DJNZ    LOOP4
071E   3E3F        1941          LD      A,3FH           ;If filename is found
                   1942                                  ;then display '
0720   D390        1943          OUT     (SEG1),A
0722   3EFF        1944          LD      A,0FFH
0724   D391        1945          OUT     (SEG2),A
0726   CDAC07      1946          CALL    GETPTR          ;The parameters (starting
                   1947                                  ;ending address and checksum)
                   1948                                  ;have been load into STEPBF.
                   1949                                  ;Load them into registers,
                   1950                                  ;calculate the block length
                   1951                                  ;and check if they are legal.
0729   3899        1952          JR      C,ERROR         ;Jump to ERROR if input
                   1953                                  ;is not successful.
072B   CD3B07      1954          CALL    TAPEIN          ;Input user's data.
072E   3894        1955          JR      C,ERROR
0730   CD9F07      1956          CALL    SUM1            ;Calculate the sum of all
0733   21D8FE      1957          LD      HL,STEPBF+8
0736   BE          1958          CP      (HL)            ;Compare it with the
                   1959                                  ;checksum calculated by and
                   1960                                  ;stored by 'W' FUNCTION.
0737   C2C406      1961          JP      NZ,ERROR        ;Jump to ERROR if not
                   1962                                  ;matched.
073A   C9          1963          RET
                   1964
                   1965  ;*****************************************************************
                   1966  TAPEIN:
                   1967  ; Load a memory block from tape.
                   1968  ; Input: HL -- starting address of the block
                   1969  ;        BC -- length of the block
                   1970  ; Output: Carry flag,1 -- reading error
                   1971  ;                  0 -- no error
                   1972  ; Destroyed reg. -- AF,BC,DE,HL,AF',BC',DE',HL'
```

```
                    1973
073B    AF          1974            XOR     A               ;Clear carry flag.
073C    08          1975            EX      AF,AF'
073D    CD4807      1976  TLOOP:    CALL    GETBYTE         ;Get one byte from TAPE.
0740    73          1977            LD      (HL),E          ;Store it into memory.
0741    EDA1        1978            CPI
0743    EA3D07      1979            JP      PE,TLOOP        ;Loop until length is zero.
0746    08          1980            EX      AF,AF'
0747    C9          1981            RET
                    1982  GETBYTE:
                    1983  ; Read one byte from tape.
                    1984  ; Output: E -- data read
                    1985  ;         Carry of F',1 -- reading error
                    1986  ;                       0 -- no error
                    1987  ; Destroy reg. -- AF,DE,AF',BC',DE',HL'
                    1988  ; Byte format:
                    1989
                    1990  ; start bit bit bit bit bit bit bit bit stop
                    1991  ;  bit   0   1   2   3   4   5   6   7   bit
                    1992
0748    CD5907      1993            CALL    GETBIT          ;Get start bit.
074B    1608        1994            LD      D,8             ;Loop 8 times.
074D    CD5907      1995  BLOOP:    CALL    GETBIT          ;Get one data bit
                    1996                                    ;result in carry flag.
0750    CB1B        1997            RR      E               ;Rotate it into E .
0752    15          1998            DEC     D
0753    20F8        1999            JR      NZ,BLOOP
0755    CD5907      2000            CALL    GETBIT          ;Get stop bit .
0758    C9          2001            RET
                    2002  GETBIT:
                    2003  ; Read one bit from tape.
                    2004  ; Output: Carry of F,0 -- this bit is 0
                    2005  ;                      1 -- this bit is 1
                    2006  ;         Carry of F',1 -- reading error
                    2007  ;                      0 -- no error
                    2008  ; Destroyed reg. -- AF,AF',BC',DE',HL'
                    2009  ; Bit format:
                    2010
                    2011  ;   0 -- 2K Hz 8 cycles + 1K Hz 2 cycles.
                    2012  ;   1 -- 2K Hz 4 cycles + 1K Hz 4 cycles.
0759    D9          2013            EXX
                    2014
                    2015  ; The tape-bit format of both 0 and 1 are
                    2016  ; of the same form:  high freq part + low freq part.
                    2017  : The difference between 0 and 1 is the
                    2018  ; number high freq cycles and low freq
                    2019  ; cycles.  Thus, a high freq period may has
                    2020  ; two meanings:
                    2021  ;  i) It is used to count the number of high
                    2022  ;     freq cycles of the current tape-bit;
                    2023  ; ii) If a high freq period is detected
                    2024  ;     immediately after a low freq period, then
                    2025  ;     this period is the first cycle of next
                    2026  ;     tape-bit and is used as a terminator of the
                    2027  ;     last tape-bit.
                    2028
                    2029  ; Bit 0 of H register is used to indicate the usage
                    2030  ; of a high freq period.  If this bit is zero, high
```

```
                    2031  ; freq period causes counter increment for the current
                    2032  ; tape-bit.  If the high freq part has passed, bit 0
                    2033  ; of H is set and the next high freq period will be used
                    2034  ; as a terminator.
                    2035  ; L register is used to up/down count the number of periods.
                    2036  ; when a high freq period is read, L is increased by
                    2037  ; 1; when a low freq period is read, L is decreased
                    2038  ; by 2. (The time duration for each count is 0.5 ms.)
                    2039  ; At the end of a tape-bit, positive and negative L
                    2040  ; stand for 0 and 1 respectively.
                    2041
075A  210000        2042          LD      HL,00
075D  CD7A07        2043  COUNT1: CALL    PERIOD          ;Read one period.
0760  14            2044          INC     D               ;The next two instructions
                    2045                                  ;check if D is zero. Carry flag
                    2046                                  ;is not affected.
0761  15            2047          DEC     D
0762  2011          2048          JR      NZ,TERR         ;If D is not zero,jump
                    2049                                  ;to error routine TERR .
                    2050                                  ;(Because the period is too
                    2051                                  ;much longer than that of 1K Hz).
0764  3806          2052          JR      C,SHORTP        ;If the period is short
                    2053                                  ;(2K Hz),jump to SHORTP.
0766  2D            2054          DEC     L               ;The period is 1K Hz,
                    2055                                  ;decrease L by 2 . And set
                    2056                                  ;bit 0 of H to indicate this
                    2057                                  ;tape-bit has passed high freq
                    2058                                  ;part and reaches its low freq
                    2059                                  ;part.
0767  2D            2060          DEC     L
0768  CBC4          2061          SET     0,H
076A  18F1          2062          JR      COUNT1
076C  2C            2063  SHORTP: INC     L               ;The period is 2K Hz ,
                    2064                                  ;increase L by 1.
076D  CB44          2065          BIT     0,H             ;If the tape bit has passed
                    2066                                  ;its high freq part, high
                    2067                                  ;frequency means this bit is all
                    2068                                  ;over and next bit has started.
076F  28EC          2069          JR      Z,COUNT1
                    2070                                  ;L= (# of 2K period) - 2*(# of 1K period)
0771  CB15          2071          RL      L
                    2072                          ; 0 --- NCarry (L positive)
                    2073                          ; 1 ---  Carry (L negative)
                    2074                          ;The positive or negative sign of
                    2075                          ;L corresponds to the tape-bit data.
                    2076                          ;'RL  L' will shift the sign bit of
                    2077                          ;L into carry flag.  After this
                    2078                          ;instruction, the carry flag
                    2079                          ;contains the tape-bit.
0773  D9            2080          EXX             ;Restore BC' DE' HL'
0774  C9            2081          RET
0775  08            2082  TERR:   EX      AF,AF'
0776  37            2083          SCF             ;Set carry flag of F' to indicate error.
0777  08            2084          EX      AF,AF'
0778  D9            2085          EXX
0779  C9            2086          RET
                    2087  PERIOD:
                    2088  ; Wait the tape to pass one period.
```

```
                    2089  ; The time duration is stored in DE.  The
                    2090  ; unit is loop count.  We use 32 as the
                    2091  ; threshold for 2K Hz and 1K Hz.
                    2092  ; result is in carry flag. (1K -- NC, 2K -- C)
                    2093  ; Register AF and DE are destroyed.
                    2094
077A   110000       2095          LD      DE,00
077D   DB92         2096  LOOPH:  IN      A,(KIN)         ;Bit 3 of port C is Tapein.
077F   13           2097          INC     DE
0780   CB5F         2098          BIT     3,A
0782   28F9         2099          JR      Z,LOOPH         ;Loop until input goes low.
0784   3EDF         2100          LD      A,11011111B     ;Echo the tape input to
                    2101                                  ;speaker on MPF_IP.
0786   D392         2102          OUT     (KIN),A
0788   3EF0         2103          LD      A,0F0H
078A   D381         2104          OUT     (DIG2),A
078C   3EFF         2105          LD      A,0FFH
078E   D380         2106          OUT     (DIG1),A
0790   DB92         2107  LOOPL:  IN      A,(KIN)
0792   13           2108          INC     DE
0793   CB5F         2109          BIT     3,A
0795   20F9         2110          JR      NZ,LOOPL        ;Loop until input goes high.
0797   3EFF         2111          LD      A,11111111B     ;Echo the tape input to
                    2112                                  ;speaker on MPF_IP.
0799   D392         2113          OUT     (KIN),A
079B   7B           2114          LD      A,E
079C   FE20         2115          CP      MPERIOD         ;Compare the result with
                    2116                                  ;the threshold.
079E   C9           2117          RET
                    2118  ;
                    2119  ;*****************************************************************
                    2120  SUM1:
                    2121  ; Calculate the sum of the data in a memory
                    2122  ; block.  The starting and ending address
                    2123  ; of this block are stored in STEPBF+2 ~ STEPBF+4.
                    2124  ;   Registers AF,BC,DE,HL are destroyed.
                    2125
079F   CDAC07       2126          CALL    GETPTR          ;Get parameters from
                    2127                                  ;step buffer.
07A2   D8           2128          RET     C               ;Return if the parameters
                    2129                                  ;are illegal.
                    2130  SUM:
                    2131  ; Calculate the sum of a memory block.
                    2132  ; HL contains the starting address of
                    2133  ; this block, BC contains the length.
                    2134  ; The result is stored in A.  Registers
                    2135  ; AF,BC,HL are destroyed.
                    2136
07A3   AF           2137          XOR     A               ;Clear A
07A4   86           2138  SUMCAL: ADD     A,(HL)
07A5   EDA1         2139          CPI
07A7   EAA407       2140          JP      PE,SUMCAL
07AA   B7           2141          OR      A               ;Clear flag.
07AB   C9           2142          RET
                    2143  GETPTR:
                    2144  ; Get parameters from step buffer.
                    2145  ; Input: (STEPBF+4) and (STEPBF+5) contain
                    2146  ;        starting address.
```

```
                    2147  ;         (STEPBF+6) and (STEPBF+7) contain
                    2148  ;         ending address.
                    2149  ; Output: HL register contains the starting
                    2150  ;         address.
                    2151  ;          BC register contains the length.
                    2152  ;          Carry flay  0 -- BC positive
                    2153  ;                      1 -- BC negative
                    2154  ; Destroyed reg.: AF,BC,DE,HL.
                    2155
07AC    21D4FE      2156            LD      HL,STEPBF+4
07AF    5E          2157  GETP      LD      E,(HL)          ;Load the starting address
                    2158                                    ;into DE .
07B0    23          2159            INC     HL
07B1    56          2160            LD      D,(HL)
07B2    23          2161            INC     HL
07B3    4E          2162            LD      C,(HL)
07B4    23          2163            INC     HL
07B5    66          2164            LD      H,(HL)          ;Load ending address
                    2165                                    ;into HL.
07B6    69          2166            LD      L,C
07B7    B7          2167            OR      A               ;Clear carry flag.
07B8    ED52        2168            SBC     HL,DE           ;Find difference.
                    2169                                    ;Carry flag is changed here.
07BA    4D          2170            LD      C,L
07BB    44          2171            LD      B,H
07BC    03          2172            INC     BC              ;Now BC contains the
                    2173                                    ;length.
07BD    EB          2174            EX      DE,HL           ;Now HL contains the
                    2175                                    ;starting address.
07BE    C9          2176            RET
                    2177
                    2178  ;*************************************************************
                    2179  TAPEOUT:
                    2180  ; Output a memory block to tape.
                    2181  ; Input: HL -- starting address of the block
                    2182  ;        BC -- length of the block
                    2183  ; Destroyed reg. -- AF,BC,DE,HL,BC',DE',HL'
                    2184
07BF    5E          2185            LD      E,(HL)          ;Get the data.
07C0    CDC907      2186            CALL    OUTBYTE         ;Output to tape.
07C3    EDA1        2187            CPI
07C5    EABF07      2188            JP      PE,TAPEOUT      ;Loop until finished.
07C8    C9          2189            RET
                    2190  OUTBYTE:
                    2191  ; Output one byte to tape.  For tape-byte
                    2192  ; format, see comments on GETBYTE.
                    2193  ; Input: E -- data
                    2194  ; Destroyed reg. -- AF,DE,BC',DE',HL'
                    2195
07C9    1608        2196            LD      D,8             ;Loop 8 times.
07CB    B7          2197            OR      A               ;Clear carry flag.
07CC    CDDC07      2198            CALL    OUTBIT          ;Output start bit.
07CF    CB1B        2199  OLOOP:    RR      E               ;Rotate data into carry.
07D1    CDDC07      2200            CALL    OUTBIT          ;Output the carry.
07D4    15          2201            DEC     D
07D5    20F8        2202            JR      NZ,OLOOP
07D7    37          2203            SCF                     ;Set carry flag.
07D8    CDDC07      2204            CALL    OUTBIT          ;Output stop bit.
```

```
07DB   C9          2205          RET
                   2206  OUTBIT:
                   2207  ; Output one bit to tape.
                   2208  ; Input: data in carry flag.
                   2209  ; Destroyed reg. -- AF,BC',DE',HL'
07DC   D9          2210          EXX
07DD   2600        2211          LD      H,0
07DF   3809        2212          JR      C,OUT1          ;If data = 1 ,output 1 .
07E1   2E08        2213  OUT0:   LD      L,ZERO_2K
07E3   CD7208      2214          CALL    TONE2K
07E6   2E02        2215          LD      L,ZERO_1K
07E8   1807        2216          JR      BITEND
                   2217  OUT1:                           ;2K 4 cycles ,1K 4 cycles
07EA   2E04        2218          LD      L,ONE_2K
07EC   CD7208      2219          CALL    TONE2K
07EF   2E04        2220          LD      L,ONE_1K
07F1   CD6E08      2221  BITEND: CALL    TONE1K
07F4   D9          2222          EXX                     ;Restore registers.
07F5   C9          2223          RET
                   2224  ;
                   2225  ;*****************************************************************
                   2226  ; Function: Clear display buffer and display prompt.
                   2227  ; Input: None
                   2228  ; Output: (OUTPTR) <- INPBF
                   2229  ;         (DISP)   <- DISPBF
                   2230  ;                     IX <- DISPBF
                   2231  ;         Set all the contents of display buffer to be FF .
                   2232  ; Reg affected: AF IX .
                   2233  ; Call: CLEAR CHRWR .
                   2234
                   2235  CLRBF:
07F6   CDB909      2236          CALL    CLEAR
07F9   3E3C        2237          LD      A,3CH
07FB   CD2409      2238          CALL    CHRWR
07FE   DD212CFF    2239          LD      IX,DISPBF
0802   C9          2240          RET
                   2241  ;
                   2242  ;*****************************************************************
                   2243  ; Function: Generate a sound.
                   2244  ; Input:None
                   2245  ; Output: None
                   2246  ; Reg affected: AF BC DE HL .
                   2247  ; Call: TONE
                   2248
0803   F5          2249  BEEP:   PUSH    AF
0804   21F4FE      2250          LD      HL,BEEPSET
0807   7E          2251          LD      A,(HL)
0808   A7          2252          AND     A
0809   200C        2253          JR      NZ,NOTONE
080B   23          2254          INC     HL
080C   4E          2255          LD      C,(HL)
080D   2AF6FE      2256          LD      HL,(TBEEP)
0810   CD7408      2257          CALL    TONE
0813   F620        2258          OR      20H             ;LED off.
0815   D392        2259          OUT     (KIN),A
                   2260  NOTONE:
0817   F1          2261          POP     AF
0818   C9          2262          RET
```

```
                    2263  ;
                    2264  ;*****************************************************************
                    2265  ; Function: check if a memory address is in RAM.
                    2266  ; Input: HL -- address to be check.
                    2267  ; Output: Zero flag -- 0, ROM or nonexistant;
                    2268  ;                      1, RAM.
                    2269  ; Destroyed reg.: AF.
                    2270  ; Call: none
                    2271
                    2272  RAMCHK:
0819  7E            2273          LD      A,(HL)
081A  2F            2274          CPL
081B  77            2275          LD      (HL),A
081C  7E            2276          LD      A,(HL)
081D  2F            2277          CPL
081E  77            2278          LD      (HL),A
081F  BE            2279          CP      (HL)
0820  C9            2280          RET
                    2281  ;
                    2282  ;*****************************************************************
                    2283  ; Function: Convert a byte (ASC II code) in A register
                    2284  ;           to display pattern.
                    2285  ; Input: A -- ASC II code.
                    2286  ;        (DISP) -- Point to the result address in display buffer.
                    2287  ; Output: Pattern for two bytes. The first byte in (DISP) and
                    2288  ;         the second byte in (DISP)+1 .
                    2289  ;        (DISP) <- (DISP)+2
                    2290  ; Reg affected: AF
                    2291  ; Call: None
                    2292
                    2293  CONVER:
0821  C5            2294          PUSH    BC
0822  D5            2295          PUSH    DE
0823  E5            2296          PUSH    HL
0824  21A80B        2297          LD      HL,SEGTAB
0827  0620          2298          LD      B,20H
0829  90            2299          SUB     B
082A  87            2300          ADD     A,A
082B  4F            2301          LD      C,A
082C  0600          2302          LD      B,0
082E  09            2303          ADD     HL,BC
082F  5E            2304          LD      E,(HL)
0830  23            2305          INC     HL
0831  56            2306          LD      D,(HL)
0832  2A84FF        2307          LD      HL,(DISP)
0835  73            2308          LD      (HL),E
0836  23            2309          INC     HL
0837  72            2310          LD      (HL),D
0838  23            2311          INC     HL
0839  2284FF        2312          LD      (DISP),HL
083C  E1            2313          POP     HL
083D  D1            2314          POP     DE
083E  C1            2315          POP     BC
083F  C9            2316          RET
                    2317  ;
                    2318  ;*****************************************************************
                    2319  ; Function: Clear the display buffer.
                    2320  ; Input: None
```

```
                    2321  ; Output: Set all contents in display buffer to be FF.
                    2322  ; Reg effected: None
                    2323  ; Call: None
                    2324
                    2325  CLRDSP:
0840  E5            2326          PUSH    HL
0841  D5            2327          PUSH    DE
0842  C5            2328          PUSH    BC
0843  212CFF        2329          LD      HL,DISPBF
0846  112DFF        2330          LD      DE,DISPBF+1
0849  015000        2331          LD      BC,80
084C  36FF          2332          LD      (HL),0FFH
084E  EDB0          2333          LDIR
0850  C1            2334          POP     BC
0851  D1            2335          POP     DE
0852  E1            2336          POP     HL
0853  C9            2337          RET
                    2338  CHKINP:                 ;Check  all the  datum in input
                    2339                          ;buffer are hexadecimal values or not
                    2340                          ;until <CR> met.
                    2341                          ;Carry flag is set if there exists
                    2342                          ;at least one non hexadecimal value.
0854  CDDF08        2343          CALL    CHKHEX
0857  D8            2344          RET     C
0858  C8            2345          RET     Z
0859  CDE508        2346  CHKIN1  CALL    GETHL
085C  D8            2347          RET     C
085D  C8            2348          RET     Z
085E  18F9          2349          JR      CHKIN1
                    2350  ECHO_CH:                        ;Echo the input character
                    2351                                  ;with <?>=
                    2352                                  ;? is the input character.
0860  CD2409        2353          CALL    CHRWR           ;?
0863  3E3E          2354          LD      A,3EH           ;>
0865  CD2409        2355          CALL    CHRWR
0868  3E3D          2356          LD      A,3DH           ;=
086A  CD2409        2357          CALL    CHRWR
086D  C9            2358          RET
                    2359          ;
                    2360  ;*****************************************************************
                    2361  ; Function: Generate square wave to the MIC & speaker
                    2362  ;           on MPF_IP.
                    2363  ; Input :  C -- period = 2*(44+13*C) clock states.
                    2364  ;         HL -- number of periods.
                    2365  ; Output: none.
                    2366  ; Destroyed reg.: AF, B(C), DE, HL.
                    2367  ; Call: none.
                    2368
                    2369  TONE1K:
086E  0E41          2370          LD      C,F1KHZ
0870  1802          2371          JR      TONE
                    2372  TONE2K:
0872  0E1F          2373          LD      C,F2KHZ
                    2374  TONE:
0874  29            2375          ADD     HL,HL
0875  110100        2376          LD      DE,1
0878  3EFF          2377          LD      A,0FFH
087A  D392          2378  SQWAVE: OUT     (KIN),A
```

```
                    2321  ; Output: Set all contents in display buffer to be FF.
                    2322  ; Reg effected: None
                    2323  ; Call: None
                    2324
                    2325  CLRDSP:
0840  E5            2326          PUSH    HL
0841  D5            2327          PUSH    DE
0842  C5            2328          PUSH    BC
0843  212CFF        2329          LD      HL,DISPBF
0846  112DFF        2330          LD      DE,DISPBF+1
0849  015000        2331          LD      BC,80
084C  36FF          2332          LD      (HL),0FFH
084E  EDB0          2333          LDIR
0850  C1            2334          POP     BC
0851  D1            2335          POP     DE
0852  E1            2336          POP     HL
0853  C9            2337          RET
                    2338  CHKINP:                 ;Check  all the  datum in input
                    2339                          ;buffer are hexadecimal values or not
                    2340                          ;until <CR> met.
                    2341                          ;Carry flag is set if there exists
                    2342                          ;at least one non hexadecimal value.
0854  CDDF08        2343          CALL    CHKHEX
0857  D8            2344          RET     C
0858  C8            2345          RET     Z
0859  CDE508        2346  CHKIN1  CALL    GETHL
085C  D8            2347          RET     C
085D  C8            2348          RET     Z
085E  18F9          2349          JR      CHKIN1
                    2350  ECHO_CH:                        ;Echo the input character
                    2351                                  ;with <?>=
                    2352                                  ;? is the input character.
0860  CD2409        2353          CALL    CHRWR           ;?
0863  3E3E          2354          LD      A,3EH           ;>
0865  CD2409        2355          CALL    CHRWR
0868  3E3D          2356          LD      A,3DH           ;=
086A  CD2409        2357          CALL    CHRWR
086D  C9            2358          RET
                    2359          ;
                    2360  ;*****************************************************************
                    2361  ; Function: Generate square wave to the MIC & speaker
                    2362  ;           on MPF_IP.
                    2363  ; Input :  C -- period = 2*(44+13*C) clock states.
                    2364  ;         HL -- number of periods.
                    2365  ; Output: none.
                    2366  ; Destroyed reg.: AF, B(C), DE, HL.
                    2367  ; Call: none.
                    2368
                    2369  TONE1K:
086E  0E41          2370          LD      C,F1KHZ
0870  1802          2371          JR      TONE
                    2372  TONE2K:
0872  0E1F          2373          LD      C,F2KHZ
                    2374  TONE:
0874  29            2375          ADD     HL,HL
0875  110100        2376          LD      DE,1
0878  3EFF          2377          LD      A,0FFH
087A  D392          2378  SQWAVE: OUT     (KIN),A
```

```
087C   41          2379          LD      B,C
087D   10FE        2380          DJNZ    $
087F   EE20        2381          XOR     20H             ;TOGGLE OUTPUT
0881   ED52        2382          SBC     HL,DE
0883   20F5        2383          JR      NZ,SQWAVE
0885   C9          2384          RET
                   2385  ;
                   2386  ;*****************************************************************
                   2387  ; Function: Print message until <CR> met.
                   2388  ; Input: HL --  Starting address of characters.
                   2389  ; Output: (OUTPTR) <- (OUTPTR)+?
                   2390  ;         (DISP)   <- (DISP)+2*?
                   2391  ;         ? is the number of charcters to be printed.
                   2392  ;         2*? is fails if there exists TAB key in input buffer.
                   2393  ; Reg affected: AF HL .
                   2394  ; Call: CLEAR MSG DECDSP CR2 .
                   2395
                   2396  PRTMES:
0886   CDB909      2397          CALL    CLEAR
0889   CDCA09      2398          CALL    MSG
088C   CD9503      2399  REG2    CALL    DECDSP
088F   CD8109      2400          CALL    CR2
0892   C9          2401          RET
                   2402  ;
                   2403  ;*****************************************************************
                   2404  ; Function: Print out all the contents in display buffer.
                   2405  ; Input: None
                   2406  ; Output: None
                   2407  ; Reg affected: AF
                   2408  ; Call: PTEST MTPPRT .
                   2409
0893   CDA308      2410  PRINTT  CALL    PTEST
0896   C0          2411          RET     NZ
0897   DDE5        2412          PUSH    IX
0899   DD2104FF    2413          LD      IX,INPBF
089D   CD0000    X 2414          CALL    MTPPRT  ; Refer to printer manual.
08A0   DDE1        2415          POP     IX
08A2   C9          2416          RET
                   2417  ;
                   2418  ;*****************************************************************
                   2419  ; Function: Check the toggle printer switch and
                   2420  ;           the condition of printer interface.
                   2421  ; Input: None
                   2422  ; Output: Zero flag = 1     (1) Printer exists and toggle
                   2423  ;                            switch is on.
                   2424
                   2425  ;         Zero flag = 0     (2) Printer exists but the
                   2426  ;                            toggle switch is off.
                   2427  ;                           (3) Printer not exists.
                   2428  ; Reg affected: AF
                   2429  ; Call:None
                   2430
                   2431  PTEST:
08A3   3AF3FE      2432          LD      A,(PRTFLG)
08A6   A7          2433          AND     A
08A7   C0          2434          RET     NZ
                   2435  PTESTT:                 ; Check printer interface.
                   2436                          ; Carry flag = 1 if printer exists.
```

```
08A8  3A0060      2437          LD      A,(6000H)
08AB  FECD        2438          CP      0CDH
08AD  C9          2439          RET
                  2440  ;
                  2441  ;*****************************************************************
                  2442  ; Function: Use (GETPT) as a pointer increase HL until
                  2443  ;           (HL-1) is one of the following delimeters:
                  2444  ;           SPACE TAB . : / =  and (HL+1) is not SPACE
                  2445  ;            or TAB.
                  2446  ; Input: HL = (GETPT) -- Starting address.
                  2447  ; Output: HL <- HL+?
                  2448  ;         (GETPT) <- (GETPT)+?
                  2449  ; Reg affected: AF HL .
                  2450  ; Call: None
                  2451
                  2452  GETCHR:
08AE  2A7EFF      2453          LD      HL,(GETPT)
                  2454  LDA:
08B1  7E          2455          LD      A,(HL)
08B2  FE20        2456          CP      ' '             ;SPACE..
08B4  2804        2457          JR      Z,SKIP_
08B6  FE09        2458          CP      '       '       ;TAB.
08B8  200E        2459          JR      NZ,EOS?
                  2460  SKIP_:
08BA  23          2461          INC     HL
08BB  7E          2462          LD      A,(HL)
08BC  FE20        2463          CP      ' '             ;SPACE.
08BE  28FA        2464          JR      Z,SKIP_
08C0  FE09        2465          CP      '       '       ;TAB.
08C2  28F6        2466          JR      Z,SKIP_
                  2467  STPTR:
08C4  227EFF      2468          LD      (GETPT),HL
08C7  C9          2469          RET
                  2470  EOS?:
08C8  FE0D        2471          CP      0DH             ;End of string?
08CA  28F8        2472          JR      Z,STPTR         ;Yes
08CC  FE3A        2473          CP      3AH             ;:
08CE  28EA        2474          JR      Z,SKIP_
08D0  FE2E        2475          CP      2EH             ;.
08D2  28E6        2476          JR      Z,SKIP_
08D4  FE3D        2477          CP      3DH             ;=
08D6  28E2        2478          JR      Z,SKIP_
08D8  FE2F        2479          CP      2FH             ;/
08DA  28DE        2480          JR      Z,SKIP_
08DC  23          2481          INC     HL
08DD  18D2        2482          JR      LDA
                  2483  CHKHEX:
08DF  2104FF      2484          LD      HL,INPBF
                  2485  CHKHE2:
08E2  227EFF      2486          LD      (GETPT),HL
                  2487  ;
                  2488  ;*****************************************************************
                  2489  ; Function: Call GETCHR and convert ASC II codes to hexadecimal
                  2490  ;           values and store them into HL .
                  2491  ; Input: (GETPT)
                  2492  ; Output: (GETPT) <- (GETPT)+?
                  2493  ;         A <- L
                  2494  ;         H=0   If there is only one hexadecimal digit.
```

```
                    2495  ;         Carry flag =1  If the data is not hexadecimal digits.
                    2496  ;         Zero flag = 1  If the last ASC II code is <CR> .
                    2497  ; Reg affected: AF DE HL .
                    2498  ; Call: GETCHR ONE .
                    2499
                    2500  GETHL:                                ;Get 4 digit number to HL &L=A
                    2501                                       ;C (Non hexadecimal values)
                    2502                                       ;Z (ØDH)
08E5  210000        2503          LD      HL,Ø                 ;Assume input ØØØØ
08E8  E5            2504          PUSH    HL                   ;Temporary store in (SP),(SP+1)
08E9  39            2505          ADD     HL,SP                ;HL=SP
08EA  EB            2506          EX      DE,HL                ;Borrow SP for tempory buffer.
08EB  CDAE08        2507          CALL    GETCHR
08EE  EB            2508          EX      DE,HL
08EF  FE30          2509  CV3     CP      '0'
08F1  300A          2510          JR      NC,CVT
08F3  FE0D          2511          CP      ØDH
08F5  2003          2512          JR      NZ,CV2
08F7  E1            2513  CV1     POP     HL
08F8  7D            2514          LD      A,L                  ;String end.
08F9  C9            2515          RET
                    2516  CV2:
08FA  A7            2517          AND     A
08FB  18FA          2518          JR      CV1
                    2519  CVT:
08FD  FE3A          2520          CP      3AH                  ;:
08FF  28F9          2521          JR      Z,CV2
                    2522  CVTHEX:
0901  CD140B        2523          CALL    ONE                  ;ASCII to HEX
0904  380A          2524          JR      C,NOTHEX
0906  ED6F          2525          RLD                          ;Rotate into (HL) i.e. (SP)
0908  23            2526          INC     HL                   ;SP+1
0909  ED6F          2527          RLD
090B  2B            2528          DEC     HL
090C  13            2529          INC     DE
090D  1A            2530          LD      A,(DE)
090E  18DF          2531          JR      CV3
                    2532  NOTHEX:                              ;Error
0910  E1            2533          POP     HL
0911  C9            2534          RET
                    2535  ;
                    2536  ;*************************************************************
                    2537  ; Function: Check the numbers of content in display buffer,
                    2538  ;           if it excess 40 the change the IX pointer.
                    2539  ; Input: (DISP)
                    2540  ; Output: IX <- IX          (If the number of contents are less
                    2541  ;                           than 40).
                    2542  ;         IX <- (DISP)-38 (If the  numbers of contents are lager
                    2543  ;                           than 40).
                    2544  ;         Carry flag = 1  If (DISP) < (DISP)+38
                    2545  ; Reg affected: AF DE HL IX .
                    2546  ; Call: None
                    2547
                    2548  CHK40:
0912  2A84FF        2549          LD      HL,(DISP)
0915  1152FF        2550          LD      DE,DISPBF+38
0918  A7            2551          AND     A
0919  ED52          2552          SBC     HL,DE
```

```
091B   DD212CFF   2553          LD      IX,DISPBF
091F   D8         2554          RET     C
0920   EB         2555          EX      DE,HL
0921   DD19       2556          ADD     IX,DE
0923   C9         2557          RET
                  2558  ;
                  2559  ;*****************************************************************
                  2560  ; Function: Convert a byte (ASC II code) in A register
                  2561  ;           to display patterns and store them into
                  2562  ;           input buffer and display buffer respectively.
                  2563  ; Input: A -- a byte of ASC II code.
                  2564  ;        (OUTPTR) -- Point to the result address in input buffer.
                  2565  ;        (DISP) -- Point to the result address in display buffer.
                  2566  ; Output: Store the ASC II code into (OUTPTR)
                  2567  ;         Pattern for two bytes. The first byte in (DISP)
                  2568  ;         and the second byte in (DISP)+1 .
                  2569  ;         (OUTPTR) <- (OUTPTR)+1
                  2570  ;         (DISP)   <- (DISP)+2
                  2571  ; Reg affected: AF
                  2572  ; Call: CONVER CURSOR .
                  2573
                  2574  CHRWR:
0924   E5         2575          PUSH    HL
0925   D5         2576          PUSH    DE
0926   2A82FF     2577          LD      HL,(OUTPTR)
0929   77         2578          LD      (HL),A
092A   23         2579          INC     HL
092B   2282FF     2580          LD      (OUTPTR),HL
092E   FE09       2581          CP      9
0930   2857       2582          JR      Z,TABOUT
0932   CD2108     2583          CALL    CONVER
                  2584  TAB_RET:
0935   CD790A     2585          CALL    CURSOR
0938   D1         2586          POP     DE
0939   E1         2587          POP     HL
093A   C9         2588          RET
                  2589  ;
                  2590  ;*****************************************************************
                  2591  ;Function:  Print out all the contents in input buffer
                  2592  ;           Check the TV interface ,if TV interface
                  2593  ;           board exists then jump to TV interface
                  2594  ;           service routine.
                  2595  ;           There are four kinds of CRX as follows:
                  2596  ; Input: (OUTPTR) -- Point to the result address in input buffer.
                  2597  ; Output: (OUTPTR) <- INPBF
                  2598  ;         (DISP)   <- DISPBF
                  2599  ; Reg affected: AF .
                  2600  ; Call: CR0 PTEST PRINTT CLEAR CURSOR .
                  2601
                  2602  CR:
093B   3E05       2603          LD      A,5
                  2604  CR4:
093D   3281FF     2605          LD      (CRSET),A
0940   E5         2606          PUSH    HL
0941   2A82FF     2607          LD      HL,(OUTPTR)
0944   360D       2608          LD      (HL),0DH
0946   CD7109     2609          CALL    CR0             ;Check TV interface.
0949   CDA308     2610          CALL    PTEST           ;Check printer interface.
```

```
094C   280D       2611          JR      Z,CR5
094E   3A81FF     2612          LD      A,(CRSET)
0951   FE40       2613          CP      40H
0953   2006       2614          JR      NZ,CR5
0955   47         2615          LD      B,A
0956   CD9B02     2616  DELAY   CALL    SCAN1
0959   10FB       2617          DJNZ    DELAY
095B   CD9308     2618  CR5     CALL    PRINTT          ;Print message.
095E   E1         2619          POP     HL
095F   3A81FF     2620          LD      A,(CRSET)
0962   FE20       2621          CP      20H
0964   C8         2622          RET     Z
0965   FE30       2623          CP      30H
0967   CAF607     2624          JP      Z,CLRBF
096A   CDB909     2625          CALL    CLEAR
096D   CD790A     2626          CALL    CURSOR
0970   C9         2627          RET
                  2628  CR0:                            ;Routine for TV interface.
0971   3A00A0     2629          LD      A,(TVSET)
0974   FEA5       2630          CP      0A5H
0976   CA01A0     2631          JP      Z,TV
0979   C9         2632          RET
                  2633  ;
                  2634  ;*****************************************************************
                  2635  ; Function: Same as CR but the display timing is about 1 sec.
                  2636  ; Input: (OUTPTR) -- Point to the result address in input buffer.
                  2637  ; Output: (OUTPTR) <- INPBF
                  2638  ;         (DISP)   <- DISPBF
                  2639  ; Reg affected: AF AF' BC' DE' HL' ,HL.
                  2640  ; Call: CR0 PTEST SCAN1 PRINTT CLEAR CURSOR .
                  2641
                  2642  CR1:
097A   CD9503     2643          CALL    DECDSP
097D   3E40       2644          LD      A,40H
097F   18BC       2645          JR      CR4
                  2646  ;
                  2647  ;*****************************************************************
                  2648  ; Function: Same as CR but CR2 do not call CLEAR and CURSOR.
                  2649  ; Input: (OUTPTR) -- Point to the result address in input buffer.
                  2650  ; Output: None
                  2651  ; Reg affected: AF
                  2652  ; Call: CR0 PTEST PRINTT .
                  2653
                  2654  CR2:
0981   3E20       2655          LD      A,20H
0983   18B8       2656          JR      CR4
                  2657  ;
                  2658  ;*****************************************************************
                  2659  ; Function: Same as CR but CR3 call routine CLRBF insted of CLEAR .
                  2660  ; Input: (OUTPTR) -- Point to the result address in input buffer.
                  2661  ; Output: (OUTPTR) <- (OUTPTR)+1
                  2662  ;         (DISP)   <- (DISP)+2
                  2663  ; Reg affected: AF IX .
                  2664  ; Call: CR0 PTEST CLRBF .
                  2665
                  2666  CR3:
0985   3E30       2667          LD      A,30H
0987   18B4       2668          JR      CR4
```

```
                    2669
                    2670  ;*****************************************************************
                    2671  ; Routine for TAB key
                    2672  ;
                    2673  TABOUT:
0989    2A84FF      2674          LD      HL,(DISP)
098C    1174FF      2675          LD      DE,DISPBF+72
098F    A7          2676          AND     A
0990    ED52        2677          SBC     HL,DE
0992    300C        2678          JR      NC,TAB??
0994    3E20        2679          LD      A,' '
0996    CD2108      2680          CALL    CONVER
0999    CDA909      2681          CALL    TAB?
099C    20EB        2682          JR      NZ,TABOUT
099E    1895        2683          JR      TAB_RET
                    2684  TAB??:
09A0    2A82FF      2685          LD      HL,(OUTPTR)
09A3    2B          2686          DEC     HL
09A4    2282FF      2687          LD      (OUTPTR),HL
09A7    188C        2688          JR      TAB_RET
                    2689
                    2690  TAB?:                   ;Check if cursor at TAB position.
                    2691                          ;Zero flag :Set if yes.
                    2692
09A9    112CFF      2693          LD      DE,DISPBF
09AC    2A84FF      2694          LD      HL,(DISP)
09AF    A7          2695          AND     A
09B0    ED52        2696          SBC     HL,DE
09B2    7D          2697          LD      A,L
                    2698  TAB?LP:
09B3    C8          2699          RET     Z
09B4    D60C        2700          SUB     12
09B6    30FB        2701          JR      NC,TAB?LP
09B8    C9          2702          RET
                    2703  ;
                    2704  ;*****************************************************************
                    2705  ; Function: Clear the display buffer and set the contents
                    2706  ;           of (DISP) and (OUTPTR) to the starting address
                    2707  ;           of display buffer and input buffer respectively.
                    2708  ; Input: None
                    2709  ; Output: (OUTPTR) <- INPBF
                    2710  ;         (DISP)   <- DISPBF
                    2711  ;         Set all the contents of display buffer to be FF .
                    2712  ; Reg affected: None
                    2713  ; Call: CLRDSP
                    2714
                    2715  CLEAR:
09B9    E5          2716          PUSH    HL
09BA    2104FF      2717          LD      HL,INPBF
09BD    2282FF      2718          LD      (OUTPTR),HL
09C0    212CFF      2719          LD      HL,DISPBF
09C3    2284FF      2720          LD      (DISP),HL
09C6    E1          2721          POP     HL
09C7    C34008      2722          JP      CLRDSP
                    2723  ;
                    2724  ;*****************************************************************
                    2725  ; Function: Convert ASC II codes to display patterns until
                    2726  ;           <CR> met.
```

```
                    2727  ;           Use HL as a pointer , convert the ASC II codes to
                    2728  ;           display patterns and stored them into display buffer.
                    2729  ; Input: HL -- Starting address of characters.
                    2730  ;        (OUTPTR) -- Point to the result address in input buffer.
                    2731  ;        (DISP) -- Point to the result address in display buffer.
                    2732  ; Output: HL <- HL+?
                    2733  ;         (OUTPTR) <- (OUTPTR)+?
                    2734  ;         (DISP)   <- (DISP)+2*?
                    2735  ;         ? is the number of cheracters to be printed.
                    2736  ; Reg affected: AF HL .
                    2737  ; Call: CHRWR
                    2738
                    2739  MSG:
09CA  7E            2740          LD      A,(HL)
09CB  23            2741          INC     HL
09CC  FE0D          2742          CP      0DH
09CE  C8            2743          RET     Z
09CF  CD2409        2744          CALL    CHRWR
09D2  18F6          2745          JR      MSG
                    2746  ;
                    2747  ;*****************************************************************
                    2748  ; Function: Get a string of characters and end with <CR> .
                    2749  ; Input:
                    2750  ;        (OUTPTR) -- Point to the result address in input buffer.
                    2751  ;        (DISP) -- Point to the result address in display buffer.
                    2752  ; Output: (INPTR) <- (OUTPTR)
                    2753  ;         (OUTPTR) <- (OUTPTR)+?
                    2754  ;         (DISP)   <- (DISP)+2*?
                    2755  ;          ? is the number of input characters. If the input
                    2756  ;         characters contains TAB code,then condition 2*? fails.
                    2757  ;         (COUNT) -- Number of characters including <CR> .
                    2758  ;         Zero flag -- Set if only <CR> is depressed.
                    2759  ; Reg affected: AF BC DE HL AF' BC' DE' HL' .
                    2760  ; Call: CHK40 CURSOR CR0 SACN CHRWR
                    2761
                    2762  READLN:
                    2763
09D4  2A82FF        2764          LD      HL,(OUTPTR)
09D7  2286FF        2765          LD      (INPTR),HL      ; Set input pointer.
                    2766  RDLOOP:
09DA  CD1209        2767          CALL    CHK40   ;Adjust IX pointer.
09DD  CD790A        2768          CALL    CURSOR
09E0  3E50          2769          LD      A,50H
09E2  3281FF        2770          LD      (CRSET),A
09E5  CD7109        2771          CALL    CR0     ;Check TV interface.
09E8  CD4602        2772          CALL    SCAN
09EB  FE11          2773          CP      11H
09ED  CAE600        2774          JP      Z,ESCAPE        ;SOFTWARE ESCAPE (CONTROL Q).
09F0  FE0D          2775          CP      0DH     ; CR
09F2  2822          2776          JR      Z,RD_END
09F4  FE5F          2777          CP      05FH    ; <--
09F6  282E          2778          JR      Z,LEFT
09F8  FE5E          2779          CP      5EH             ;UP arrow.
09FA  28DE          2780          JR      Z,RDLOOP
09FC  FE69          2781          CP      69H             ;DOWN arrow.
09FE  28DA          2782          JR      Z,RDLOOP
0A00  2A84FF        2783          LD      HL,(DISP)
0A03  117CFF        2784          LD      DE,DISPBF+80    ;Check the numbers of character
```

```
                    2785                                    ;in input buffer.
                    2786                                    ;The numbers of input characters
                    2787                                    ;is limited to 40.
0A06    A7          2788            AND     A
0A07    ED52        2789            SBC     HL,DE
0A09    30CF        2790            JR      NC,RDLOOP
0A0B    FE68        2791            CP      K_TAB           ;Check TAB key .
0A0D    2002        2792            JR      NZ,NOTTAB
0A0F    3E09        2793            LD      A,09            ;09 is the ASC II code for
                    2794                                    ;TAB key .
                    2795    NOTTAB:
0A11    CD2409      2796            CALL    CHRWR
0A14    18C4        2797            JR      RDLOOP
                    2798    RD_END:
0A16    2A82FF      2799            LD      HL,(OUTPTR)
0A19    77          2800            LD      (HL),A          ; Store 0DH .
0A1A    ED5B86FF    2801            LD      DE,(INPTR)
0A1E    ED52        2802            SBC     HL,DE           ; zero flag .
0A20    23          2803            INC     HL
0A21    7D          2804            LD      A,L
0A22    320000    X 2805            LD      (COUNT),A       ; Set /COUNT/
0A25    C9          2806            RET
                    2807    LEFT:                           ;Backspace key service routine.
0A26    2A86FF      2808            LD      HL,(INPTR)
0A29    ED5B82FF    2809            LD      DE,(OUTPTR)
0A2D    A7          2810            AND     A
0A2E    ED52        2811            SBC     HL,DE
0A30    30A8        2812            JR      NC,RDLOOP       ; Ignore if exceeding LEFT end.
0A32    EB          2813            EX      HL,DE
0A33    2B          2814            DEC     HL              ;Decrease the pointer of
                    2815                                    ;input buffer by one.
0A34    2282FF      2816            LD      (OUTPTR),HL
0A37    7E          2817            LD      A,(HL)
0A38    FE09        2818            CP      09
0A3A    2805        2819            JR      Z,B_TAB
0A3C    CD6A0A      2820            CALL    B_SP
0A3F    1899        2821            JR      RDLOOP
                    2822    B_TAB:
0A41    CD6A0A      2823            CALL    B_SP
0A44    CDA909      2824            CALL    TAB?            ;Check if cursor at TAB position.
0A47    280A        2825            JR      Z,B_TAB1
0A49    2A84FF      2826            LD      HL,(DISP)
0A4C    2B          2827            DEC     HL
0A4D    7E          2828            LD      A,(HL)
0A4E    2B          2829            DEC     HL
0A4F    A6          2830            AND     (HL)
0A50    3C          2831            INC     A
0A51    28EE        2832            JR      Z,B_TAB
                    2833    B_TAB1:
0A53    2A82FF      2834            LD      HL,(OUTPTR)
                    2835    B_TAB2:
0A56    2B          2836            DEC     HL
0A57    7E          2837            LD      A,(HL)
0A58    FE20        2838            CP      ' '
0A5A    C2DA09      2839            JP      NZ,RDLOOP
0A5D    CDA909      2840            CALL    TAB?
0A60    CADA09      2841            JP      Z,RDLOOP
0A63    3E20        2842            LD      A,' '
```

```
0A65   CD2108      2843          CALL    CONVER
0A68   18EC        2844          JR      B_TAB2
                   2845  B_SP:                          ;Clear the rightmost patterns
                   2846                                 ;in display buffer.
0A6A   E5          2847          PUSH    HL
0A6B   CD9903      2848          CALL    DEC_SP
0A6E   2B          2849          DEC     HL
0A6F   2B          2850          DEC     HL
0A70   2B          2851          DEC     HL
0A71   2284FF      2852          LD      (DISP),HL
0A74   CD790A      2853          CALL    CURSOR
0A77   E1          2854          POP     HL
0A78   C9          2855          RET
                   2856  ;
                   2857  ;*************************************************************
                   2858  ; Function: Get cursor message.
                   2859  ; Input: (DISP) -- Point to the result address in display buffer.
                   2860  ; Output: The first byte of cursor in (DISP) and the second
                   2861  ;         byte of cusor in (DISP)+1
                   2862  ;         (DISP) <- (DISP)   The content of (DISP) is unchanged.
                   2863  ; Reg affected: AF
                   2864  ; Call: CONVER
                   2865
                   2866  CURSOR:
0A79   3E5B        2867          LD      A,05BH           ; PROMPT
                   2868  CCURSOR:                       ; CALL HERE IF CHANGE PROMPT
0A7B   CD2108      2869          CALL    CONVER
0A7E   E5          2870          PUSH    HL
0A7F   2A84FF      2871          LD      HL,(DISP)
0A82   2B          2872          DEC     HL
0A83   2B          2873          DEC     HL
0A84   2284FF      2874          LD      (DISP),HL
0A87   E1          2875          POP     HL
0A88   C9          2876          RET
                   2877  ;
                   2878  ;*************************************************************
                   2879  ; Function: Convert binary data in HL to ASC II code and
                   2880  ;           display patterns.
                   2881  ; Input: HL -- Two bytes of hexadecimal values in HL.
                   2882  ;        (OUTPTR) -- Point to the result address in input buffer.
                   2883  ;        (DISP) -- Point to the result address in display buffer.
                   2884  ; Output: Four ASC II code in (OUTPTR) - (OUTPTR)+3
                   2885  ;         Eight bytes of display pattern in (DISP) - (DISP)+7
                   2886  ;         (OUTPTR) <- (OUTPTR)+4
                   2887  ;         (DISP)   <- (DISP)+8
                   2888  ; Reg affected: AF
                   2889  ; Call: HEX2
                   2890
                   2891  HEXX:
0A8C   7C          2892          LD      A,H
0A8A   CD9A0A      2893          CALL    HEX2
0A8D   7D          2894          LD      A,L
0A8E   CD9A0A      2895          CALL    HEX2
0A91   C9          2896          RET
                   2897  ;
                   2898  ;*************************************************************
                   2899  ; Function: Convert binary datas in HL to ASC II codes and
                   2900  ;           display patterns.
```

```
                    2901  ;           Call routint SPACE1 to insert a space.
                    2902  ; Input: Same as HEXX
                    2903  ; Output: Five ASC II codes in (OUTPTR) - (OUTPTR)+4 .
                    2904  ;         Ten bytes of display pattern in (DISP) - (DISP)+8 .
                    2905  ;         (OUTPTR) <- (OUTPTR)+5
                    2906  ;         (DISP)   <- (DISP)+10
                    2907  ; Reg affected: AF
                    2908  ; Call: HEXX SPACE1 .
                    2909
                    2910  HEX4:
0A92  CD890A        2911          CALL    HEXX
0A95  3E20          2912  SPACE1  LD      A,' '
0A97  C32409        2913          JP      CHRWR
                    2914  ;
                    2915  ;*****************************************************************
                    2916  ; Function: Convert binary data to ASC II code and
                    2917  ;           display patterns.
                    2918  ; Input: A -- a byte in A register.
                    2919  ;        (OUTPTR) -- Point to the result address in input buffer.
                    2920  ;        (DISP) -- Point to the result address in display buffer.
                    2921  ; Output: The first ASC II code in (OUTPTR) and the second
                    2922  ;         ASC II code in (OUTPTR)+1 . Display patterns for
                    2923  ;         four bytes . The first byte in (DISP) and the
                    2924  ;         second byte in (DISP)+1 ,and so on.
                    2925  ;         (OUTPTR) <- (OUTPTR)+2
                    2926  ;         (DISP)   <- (DISP)+4
                    2927  ; Reg affected: AF
                    2928  ; Call: HEX1
                    2929
                    2930  HEX2:
0A9A  E5            2931          PUSH    HL
0A9B  210000      X 2932          LD      HL,TEMP
0A9E  77            2933          LD      (HL),A
0A9F  AF            2934          XOR     A
0AA0  ED6F          2935          RLD
0AA2  CDAD0A        2936          CALL    HEX1
0AA5  AF            2937          XOR     A
0AA6  ED6F          2938          RLD
0AA8  CDAD0A        2939          CALL    HEX1
0AAB  E1            2940          POP     HL
0AAC  C9            2941          RET
                    2942  ;
                    2943  ;*****************************************************************
                    2944  ; Function: Convert binary data to ASC II code and display
                    2945  ;           pattern.
                    2946  ; Input: A -- LSB 4 bits contains the binary data.
                    2947  ;        (OUTPTR) -- Point to the result address in input buffer.
                    2948  ;        (DISP) -- Point to the result address in display buffer.
                    2949  ; Output: ASC II code in (OUTPTR).
                    2950  ;         Pattern for two bytes. The first byte in (DISP)
                    2951  ;         and the second byte in (DISP)+1 .
                    2952  ;         (OUTPTR) <- (OUTPTR)+1
                    2953  ;         (DISP)   <- (DISP)+2
                    2954  ; Reg affected: AF
                    2955  ; Call: CHRWR
                    2956
                    2957  HEX1:
0AAD  C630          2958          ADD     A,'0'
```

```
0AAF   FE3A       2959          CP      '9'+1
0AB1   3802       2960          JR      C,HHH
0AB3   C607       2961          ADD     A,7
                  2962  HHH:
0AB5   C32409     2963          JP      CHRWR
                  2964  ;
                  2965  ;*************************************************************
                  2966  ; Function: Convert hexadecimal values in HL to corresponding
                  2967  ;           decimal format (in ASC II CODE format).
                  2968  ; Input: HL -- Hexadecimal values to be changed.
                  2969  ;        (OUTPTR) -- Point to the result address in input buffer.
                  2970  ;        (DISP) -- Point to the result address in display buffer.
                  2971  ; Output: (OUTPTR) <- (OUTPTR)+?
                  2972  ;         (DISP)   <- (DISP)+2*?
                  2973  ; Reg affected: AF BC DE HL IY .
                  2974  ; Call: CHRWR
                  2975
                  2976  DECIMAL:
                  2977
0AB8   FD21A90C   2978          LD      IY,TENS         ; Table of ten's powers.
0ABC   0603       2979          LD      B,3             ; Output three digits.
0ABE   0E00       2980          LD      C,0             ; Zero supress flag.
                  2981  CLOOP:
0AC0   FD5E00     2982          LD      E,(IY)
0AC3   FD23       2983          INC     IY
0AC5   FD5600     2984          LD      D,(IY)
0AC8   FD23       2985          INC     IY
0ACA   AF         2986          XOR     A
                  2987  DECLOOP:
0ACB   ED52       2988          SBC     HL,DE
0ACD   3803       2989          JR      C,ADDBACK
0ACF   3C         2990          INC     A
0AD0   18F9       2991          JR      DECLOOP
                  2992  ADDBACK:
0AD2   19         2993          ADD     HL,DE
0AD3   CDD90A     2994          CALL    SUPRESS
0AD6   10E8       2995          DJNZ    CLOOP
0AD8   C9         2996          RET
                  2997  SUPRESS:
0AD9   A7         2998          AND     A
0ADA   2806       2999          JR      Z,YES_0         ; If zero then ckeck zero
                  3000                                  ;supress flag.
0ADC   4F         3001          LD      C,A             ; Else
0ADD   C630       3002          ADD     A,30H           ; Convert to ASC II code format
0ADF   C32409     3003          JP      CHRWR           ; and output.
                  3004  YES_0:
0AE2   79         3005          LD      A,C
0AE3   A7         3006          AND     A
0AE4   2805       3007          JR      Z,BLANK0        ; Supress leading zero .
                  3008  PRINT0:
0AE6   3E30       3009          LD      A,'0'
0AE8   C32409     3010          JP      CHRWR
                  3011  BLANK0:
0AEB   78         3012          LD      A,B             ; Still check for last digit.
0AEC   3D         3013          DEC     A
0AED   28F7       3014          JR      Z,PRINT0        ; If last digit then print '0'
0AEF   3E20       3015          LD      A,' '
0AF1   C32409     3016          JP      CHRWR
```

```
                    3017
                    3018  ;*****************************************************************
                    3019  ; Function: Convert ASC II codes to corresponding hexadecimal
                    3020  ;           values until met none hexadecimal digit.
                    3021  ;           The return value is stored in HL .
                    3022  ; Input: DE -- Point to the first location of ASC II code
                    3023  ;           to be changed.
                    3024  ; Output: HL -- Return values (hexadecimal digits).
                    3025  ;           (HEXFLAG) is set if there exists a digit within
                    3026  ;           ('A'..'F') or the last none hexadecimal character
                    3027  ;           is 'H' .
                    3028  ; Reg affected: AF BC DE HL .
                    3029  ; Call: ONE
                    3030
                    3031  HEXBIN:
                    3032
0AF4  AF            3033          XOR     A
0AF5  320000    X   3034          LD      (HEXFLAG),A
0AF8  47            3035          LD      B,A
0AF9  67            3036          LD      H,A
0AFA  6F            3037          LD      L,A
                    3038  HBLOOP:
0AFB  1A            3039          LD      A,(DE)
0AFC  CD140B        3040          CALL    ONE
0AFF  3809          3041          JR      C,H?
0B01  29            3042          ADD     HL,HL    · [HL]=16*[HL]
0B02  29            3043          ADD     HL,HL
0B03  29            3044          ADD     HL,HL
0B04  29            3045          ADD     HL,HL
0B05  4F            3046          LD      C,A
0B06  09            3047          ADD     HL,BC
0B07  13            3048          INC     DE
0B08  18F1          3049          JR      HBLOOP
                    3050  H?:
0B0A  1A            3051          LD      A,(DE)
0B0B  FE48          3052          CP      'H'
0B0D  C0            3053          RET     NZ
0B0E  13            3054          INC     DE
0B0F  320000    X   3055          LD      (HEXFLAG),A
0B12  1A            3056          LD      A,(DE)
0B13  C9            3057          RET
                    3058
                    3059
                    3060  ;
                    3061  ;*****************************************************************
                    3062  ; Function: Convert a byte (ASC II code) in A register to
                    3063  ;           hexadecimal digit.
                    3064  ; Input: A -- ASC II code.
                    3065  ; Output: A -- Hexadecimal values.
                    3066  ;         Carry flag = 1   If the data is not a hexadecimal digit.
                    3067  ;         (HEXFLAG) is not zero If the content of A  within 'A'
                    3068  ;         and 'F'.
                    3069  ; Reg affected: AF
                    3070  ; Call: None
                    3071
                    3072  ONE:
0B14  FE47          3073          CP      'F'+1
0B16  3F            3074          CCF
```

```
0B17   D8          3075          RET     C
0B18   D630        3076          SUB     '0'
0B1A   D8          3077          RET     C
0B1B   FE0A        3078          CP      10
0B1D   3F          3079          CCF
0B1E   D0          3080          RET     NC
0B1F   D607        3081          SUB     7
0B21   FE0A        3082          CP      10
0B23   D8          3083          RET     C
0B24   320000    X 3084          LD      (HEXFLAG),A
0B27   C9          3085          RET
                   3086  ;
                   3087  ;*****************************************************************
                   3088  ; Function: Convert ASC II codes to corresponding decimal values
                   3089  ;           in binary until met non decimal digits.
                   3090  ; Input: DE -- Point to the first ASC II code (Decimal format)
                   3091  ;              to be changed.
                   3092  ; Output: HL -- Return values (Decimal digits).
                   3093  ; Reg affected: AF BC DE HL .
                   3094  ; Call: None
                   3095
                   3096  DECBIN:
                   3097
0B28   210000      3098          LD      HL,0
0B2B   1A          3099          LD      A,(DE)
                   3100  NDIGIT:
0B2C   D630        3101          SUB     '0'
0B2E   D8          3102          RET     C
0B2F   FE0A        3103          CP      10
0B31   D0          3104          RET     NC
0B32   29          3105          ADD     HL,HL   ; [HL]=10*[HL]
0B33   44          3106          LD      B,H
0B34   4D          3107          LD      C,L
0B35   29          3108          ADD     HL,HL
0B36   29          3109          ADD     HL,HL
0B37   09          3110          ADD     HL,BC
0B38   0600        3111          LD      B,0
0B3A   4F          3112          LD      C,A
0B3B   09          3113          ADD     HL,BC
0B3C   13          3114          INC     DE
0B3D   1A          3115          LD      A,(DE)
0B3E   18EC        3116          JR      NDIGIT
                   3117  ;
                   3118  ;*****************************************************************
                   3119  ; Function: Skip TABs and BLANKs.
                   3120  ; Input: HL -- Address to be check.
                   3121  ; Output: HL <- HL+?  (? is the numbers of TAB and BLANK).
                   3122  ;                      and (HL) is not TAB or BLANK.
                   3123  ;         A  <- (HL)
                   3124  ;         Carry flag = 0   If (HL) is between 'A' and 'Z'.
                   3125  ; Reg affected: AF HL .
                   3126  ; Call: None
                   3127
                   3128  SKIP:
0B40   7E          3129          LD      A,(HL)
0B41   FE20        3130          CP      ' '
0B43   2803        3131          JR      Z,SK1
0B45   FE09        3132          CP      09H     ; TAB
```

```
ØB47   CØ          3133            RET     NZ
                    3134    SK1:
ØB48   23          3135            INC     HL
ØB49   18F5        3136            JR      SKIP
                    3137    A_Z?:
                    3138    ; RETURN C-FLAG IF [A] IS NOT WITHIN {'A'..'Z'}
ØB4B   FE41        3139            CP      'A'
ØB4D   D8          3140            RET     C
ØB4E   FE5B        3141            CP      'Z'+1
ØB5Ø   3F          3142            CCF
ØB51   C9          3143            RET
                    3144
                    3145    ;KEY CODE FOR DEPRESSED KEY
                    3146    KEYTAB:
ØB52   31          3147    KØ      DEFB    31H             ;1
ØB53   41          3148    K1      DEFB    41H             ;A
ØB54   2Ø          3149    K2      DEFB    2ØH             ;SPACE
ØB55   32          3150    K3      DEFB    32H             ;2
ØB56   53          3151    K4      DEFB    53H             ;S
ØB57   5F          3152    K5      DEFB    5FH             ;<--
ØB58   33          3153    K6      DEFB    33H             ;3
ØB59   44          3154    K7      DEFB    44H             ;D
ØB5A   68          3155    K8      DEFB    68H             ;-->
ØB5B   34          3156    K9      DEFB    34H             ;4
ØB5C   46          3157    KA      DEFB    46H             ;F
ØB5D   69          3158    KB      DEFB    69H             ;DOWN ARROW
ØB5E   35          3159    KC      DEFB    35H             ;5
ØB5F   47          3160    KD      DEFB    47H             ;G
ØB6Ø   5E          3161    KE      DEFB    5EH             ;UP ARROW
ØB61   36          3162    KF      DEFB    36H             ;6
ØB62   48          3163    K1Ø     DEFB    48H             ;H
ØB63   ØD          3164    K11     DEFB    ØDH             ;CR
ØB64   37          3165    K12     DEFB    37H             ;7
ØB65   4A          3166    K13     DEFB    4AH             ;J
ØB66   2F          3167    KK2F    DEFB    2FH             ;/
ØB67   38          3168    K15     DEFB    38H             ;8
ØB68   4B          3169    K16     DEFB    4BH             ;K
ØB69   3C          3170    KK3C    DEFB    3CH             ;<
ØB6A   39          3171    K18     DEFB    39H             ;9
ØB6B   4C          3172    K19     DEFB    4CH             ;L
ØB6C   3E          3173    KK3E    DEFB    3EH             ;>
ØB6D   3Ø          3174    K1B     DEFB    3ØH             ;Ø
ØB6E   3A          3175    K1C     DEFB    3AH             ;:
ØB6F   7B          3176    K1D     DEFB    7BH             ;UNUSED
ØB7Ø   51          3177    K1E     DEFB    51H             ;Q
ØB71   5A          3178    K1F     DEFB    5AH             ;Z
ØB72   2D          3179    KK12    DEFB    2DH             ;-
ØB73   57          3180    K21     DEFB    57H             ;W
ØB74   58          3181    K22     DEFB    58H             ;X
ØB75   3B          3182    KK13    DEFB    3BH             ;;
ØB76   45          3183    K24     DEFB    45H             ;E
ØB77   43          3184    K25     DEFB    43H             ;C
ØB78   4Ø          3185    KK19    DEFB    4ØH             ;@
ØB79   52          3186    K27     DEFB    52H             ;R
ØB7A   56          3187    K28     DEFB    56H             ;V
ØB7B   5B          3188    KK14    DEFB    5BH             ;
ØB7C   54          3189    K2A     DEFB    54H             ;T
ØB7D   42          3190    K2B     DEFB    42H             ;B
```

```
0B7E  2B          3191  KK15    DEFB   2BH          ;+
0B7F  59          3192  K2D     DEFB   59H          ;Y
0B80  4E          3193  K2E     DEFB   4EH          ;N
0B81  3D          3194  KK24    DEFB   3DH          ;=
0B82  55          3195  K30     DEFB   55H          ;U
0B83  4D          3196  K31     DEFB   4DH          ;M
0B84  7B          3197  K32     DEFB   7BH          ;UNUSED
0B85  49          3198  K33     DEFB   49H          ;I
0B86  2C          3199  K34     DEFB   2CH          ;,
0B87  7B          3200  K35     DEFB   7BH          ;UNUSED
0B88  4F          3201  K36     DEFB   4FH          ;O
0B89  2E          3202  K37     DEFB   2EH          ;.
0B8A  7B          3203  K38     DEFB   7BH          ;UNUSED
0B8B  50          3204  K39     DEFB   50H          ;P
0B8C  3F          3205  K3A     DEFB   3FH          ;?
0B8D  7B          3206  K7B     DEFB   7BH          ;UNUSED
                  3207  RTABLE:
0B8E  41          3208          DEFB   41H          ;A
0B8F  46          3209          DEFB   46H          ;F
0B90  42          3210          DEFB   42H          ;B
0B91  43          3211          DEFB   43H          ;C
0B92  44          3212          DEFB   44H          ;D
0B93  45          3213          DEFB   45H          ;E
0B94  48          3214          DEFB   48H          ;H
0B95  4C          3215          DEFB   4CH          ;L
0B96  60          3216          DEFB   60H          ;A'
0B97  61          3217          DEFB   61H          ;F'
0B98  62          3218          DEFB   62H          ;B'
0B99  63          3219          DEFB   63H          ;C'
0B9A  64          3220          DEFB   64H          ;D'
0B9B  65          3221          DEFB   65H          ;E'
0B9C  66          3222          DEFB   66H          ;H'
0B9D  67          3223          DEFB   67H          ;L'
0B9E  49          3224          DEFB   49H          ;I
0B9F  58          3225          DEFB   58H          ;X
0BA0  49          3226          DEFB   49H          ;I
0BA1  59          3227          DEFB   59H          ;Y
0BA2  53          3228          DEFB   53H          ;S
0BA3  50          3229          DEFB   50H          ;P
0BA4  50          3230          DEFB   50H          ;P
0BA5  43          3231          DEFB   43H          ;C
0BA6  49          3232          DEFB   49H          ;I
0BA7  46          3233          DEFB   46H          ;F
                  3234  SEGTAB:
0BA8  FFFF        3235          DEFW   0FFFFH       ;SPACE
0BAA  FEF1        3236          DEFW   0F1FEH       ;!
0BAC  DFF7        3237          DEFW   0F7DFH       ;"
0BAE  31FC        3238          DEFW   0FC31H       ;#
0BB0  12FC        3239          DEFW   0FC12H       ;$
0BB2  1BC3        3240          DEFW   0C31BH       ;%
0BB4  24E7        3241          DEFW   0E724H       ;&
0BB6  FFFB        3242          DEFW   0FBFFH       ;'
0BB8  FFEB        3243          DEFW   0EBFFH       ;(
0BBA  FFD7        3244          DEFW   0D7FFH       ;)
0BBC  3FC0        3245          DEFW   0C03FH       ;*
0BBE  3FFC        3246          DEFW   0FC3FH       ;+
0BC0  FFDF        3247          DEFW   0DFFFH       ;,
0BC2  3FFF        3248          DEFW   0FF3FH       ;-
```

```
ØBC4   FFBF        3249          DEFW    ØBFFFH          ;.
ØBC6   FFDB        3250          DEFW    ØDBFFH          ;/
ØBC8   CØDB        3251          DEFW    ØDBC0H          ;Ø
ØBCA   FFFC        3252          DEFW    ØFCFFH          ;1
ØBCC   24FF        3253          DEFW    ØFF24H          ;2
ØBCE   30FF        3254          DEFW    ØFF30H          ;3
ØBDØ   19FF        3255          DEFW    ØFF19H          ;4
ØBD2   72F7        3256          DEFW    ØF772H          ;5
ØBD4   Ø2FF        3257          DEFW    ØFFØ2H          ;6
ØBD6   F8FF        3258          DEFW    ØFFF8H          ;7
ØBD8   ØØFF        3259          DEFW    ØFFØØH          ;8
ØBDA   1ØFF        3260          DEFW    ØFF1ØH          ;9
ØBDC   7FEF        3261          DEFW    ØEF7FH          ;:
ØBDE   BFDF        3262          DEFW    ØDFBFH          ;;
ØBEØ   F7DB        3263          DEFW    ØDBF7H          ;<
ØBE2   37FF        3264          DEFW    ØFF37H          ;=
ØBE4   F7E7        3265          DEFW    ØE7F7H          ;>
ØBE6   7CFD        3266          DEFW    ØFD7CH          ;?
ØBE8   AØFD        3267          DEFW    ØFDAØH          ;@
ØBEA   ØBFF        3268          DEFW    ØFFØBH          ;A
ØBEC   7ØFC        3269          DEFW    ØFC7ØH          ;B
ØBEE   C6FF        3270          DEFW    ØFFC6H          ;C
ØBFØ   FØFC        3271          DEFW    ØFCFØH          ;D
ØBF2   Ø6FF        3272          DEFW    ØFFØ6H          ;E
ØBF4   ØEFF        3273          DEFW    ØFFØEH          ;F
ØBF6   42FF        3274          DEFW    ØFF42H          ;G
ØBF8   Ø9FF        3275          DEFW    ØFFØ9H          ;H
ØBFA   F6FC        3276          DEFW    ØFCF6H          ;I
ØBFC   E1FF        3277          DEFW    ØFFE1H          ;J
ØBFE   8FEB        3278          DEFW    ØEB8FH          ;K
ØCØØ   C7FF        3279          DEFW    ØFFC7H          ;L
ØCØ2   C9F3        3280          DEFW    ØF3C9H          ;M
ØCØ4   C9E7        3281          DEFW    ØE7C9H          ;N
ØCØ6   CØFF        3282          DEFW    ØFFCØH          ;O
ØCØ8   ØCFF        3283          DEFW    ØFFØCH          ;P
ØCØA   CØEF        3284          DEFW    ØEFCØH          ;Q
ØCØC   ØCEF        3285          DEFW    ØEFØCH          ;R
ØCØE   12FF        3286          DEFW    ØFF12H          ;S
ØC1Ø   FEFC        3287          DEFW    ØFCFEH          ;T
ØC12   C1FF        3288          DEFW    ØFFC1H          ;U
ØC14   CFDB        3289          DEFW    ØDBCFH          ;V
ØC16   C9CF        3290          DEFW    ØCFC9H          ;W
ØC18   FFC3        3291          DEFW    ØC3FFH          ;X
ØC1A   FFF1        3292          DEFW    ØF1FFH          ;Y
ØC1C   F6DB        3293          DEFW    ØDBF6H          ;Z
ØC1E   FFCF        3294          DEFW    ØCFFFH          ;^
ØC2Ø   FFE7        3295          DEFW    ØE7FFH          ;/
ØC22   FØFF        3296          DEFW    ØFFF0H          ;]
ØC24   FFCD        3297          DEFW    ØCDFFH          ;^
ØC26   7FEB        3298          DEFW    ØEB7FH          ;<
ØC28   Ø8BF        3299          DEFW    ØBF08H          ;A'
ØC2A   ØEBF        3300          DEFW    ØBF0EH          ;F'
ØC2C   7ØBC        3301          DEFW    ØBC70H          ;B'
ØC2E   C6BF        3302          DEFW    ØBFC6H          ;C'
ØC3Ø   FØBC        3303          DEFW    ØBCF0H          ;D'
ØC32   Ø6BF        3304          DEFW    ØBF06H          ;E'
ØC34   Ø9BF        3305          DEFW    ØBF09H          ;H'
ØC36   C7BF        3306          DEFW    ØBFC7H          ;L'
```

```
                    3307  SHIFTT:
0C38    3C          3308          DEFB    3CH             ;<
0C39    FF          3309          DEFB    0FFH
0C3A    3E          3310          DEFB    3EH             ;>
0C3B    FF          3311          DEFB    0FFH
0C3C    2A          3312          DEFB    2AH             ;*
0C3D    21          3313          DEFB    21H             ;!
0C3E    22          3314          DEFB    22H             ;"
0C3F    23          3315          DEFB    23H             ;#
0C40    24          3316          DEFB    24H             ;$
0C41    25          3317          DEFB    25H             ;%
0C42    26          3318          DEFB    26H             ;&
0C43    27          3319          DEFB    27H             ;'
0C44    28          3320          DEFB    28H             ;(
0C45    29          3321          DEFB    29H             ;)
0C46    3B          3322          DEFB    3BH             ;;
0C47    FF          3323          DEFB    0FFH
0C48    FF          3324          DEFB    0FFH
0C49    FF          3325          DEFB    0FFH
0C4A    FF          3326          DEFB    0FFH
0C4B    2F          3327          DEFB    2FH             ;/
0C4C    FF          3328          DEFB    0FFH
0C4D    FF          3329          DEFB    0FFH
0C4E    FF          3330          DEFB    0FFH
0C4F    FF          3331          DEFB    0FFH
0C50    FF          3332          DEFB    0FFH
0C51    FF          3333          DEFB    0FFH
0C52    FF          3334          DEFB    0FFH
0C53    FF          3335          DEFB    0FFH
0C54    FF          3336          DEFB    0FFH
0C55    2D          3337          DEFB    2DH             ;-
0C56    FF          3338          DEFB    0FFH
0C57    5B          3339          DEFB    5BH             ;^
0C58    40          3340          DEFB    40H             ;@
0C59    FF          3341          DEFB    0FFH
0C5A    FF          3342          DEFB    0FFH
0C5B    3D          3343          DEFB    3DH             ;=
0C5C    2B          3344          DEFB    2BH             ;+
                    3345  MPFII:
0C5D    2A2A2A2A    3346          DEFM    '*****MPF'
0C65    2D          3347          DEFB    2DH             ;-
0C66    49          3348          DEFM    'I'
0C67    2D          3349          DEFB    2DH             ;-
0C68    504C5553    3350          DEFM    'PLUS*****'
0C71    0D          3351          DEFB    0DH
                    3352  ERR_SP:
0C72    4552524F    3353          DEFM    'ERROR'
0C77    2D          3354          DEFB    2DH             ;_
0C78    5350        3355          DEFM    'SP'
0C7A    0D          3356          DEFB    0DH
                    3357  SYS_SP:
0C7B    535953      3358          DEFM    'SYS'
0C7E    2D          3359          DEFB    2DH             ;_
0C7F    5350        3360          DEFM    'SP'
0C81    0D          3361          DEFB    0DH
                    3362  PRTON:
0C82    50525420    3363          DEFM    'PRT ON'
0C88    0D          3364          DEFB    0DH
```

```
                    3365  PRTOFF:
0C89    50525420    3366          DEFM    'PRT OFF'
0C90    0D          3367          DEFB    0DH
                    3368  RAM2K_VALUE_SET:
0C91    00F8        3369          DEFW    0F800H          ;SET EDITOR LIMITS.
0C93    FFFC        3370          DEFW    0FCFFH
0C95    00FE        3371          DEFW    0FE00H          ;SET SYMBOL LIMITS.
0C97    A0FE        3372          DEFW    0FEA0H
0C99    00FD        3373          DEFW    0FD00H          ;SET OBJECT LIMITS.
0C9B    FFFD        3374          DEFW    0FDFFH
                    3375  RAM4K_VALUE_SET:
0C9D    00F0        3376          DEFW    0F000H          ;SET EDITOR LIMITS.
0C9F    FFFA        3377          DEFW    0FAFFH
0CA1    00FD        3378          DEFW    0FD00H          ;SET SYMBOL LIMITS.
0CA3    A0FE        3379          DEFW    0FEA0H
0CA5    00FB        3380          DEFW    0FB00H          ;SET OBJECT LIMITS.
0CA7    FFFC        3381          DEFW    0FCFFH
                    3382  TENS    ; TABLE USED BY 'TOASCII' TO CONVERT
                    3383          ; BINARY TO DECIMAL DIGITS
0CA9    6400        3384          DEFW    100
0CAB    0A00        3385          DEFW     10
0CAD    0100        3386          DEFW      1
0CAF    20455252    3387  ERRSMSG DEFM    ' ERRORS'
0CB6    0D          3388          DEFB    0DH
```

```
                    3389  *HEADING RAM STORAGE
                    3390  ;************************;
                    3391  ;*                      *;
                    3392  ;*       STORAGE        *;
                    3393  ;*     FOR MONITOR      *;
                    3394  ;*                      *;
                    3395  ;************************;
                    3396  ;
                    3397  ;
FEA0                3398          ORG     0FEA0H
                    3399  USERSTK:
FEA0                3400          DEFS    30H
FED0                3401          ORG     0FED0H
                    3402  SYSSTK:
FED0                3403  STEPBF  DEFS    9
                    3404  TEXT_F                          ;ASSEMBLER SOURCE FROM.
FED9                3405  EDIT_START_ADDR DEFS    2       ;EDITOR BOTTOM.
                    3406  TEXT_T                          ;ASSEMBLER SOURCE TO.
FEDB                3407  END_DATA_ADDR   DEFS    2       ;EDITOR TOP.
FEDD                3408  END_LN_NO       DEFS    2       ;EDITOR LAST LINE NUMBER.
FEDF                3409  RAM_START_ADDR  DEFS    2       ;EDITOR LOW LIMIT.
FEE1                3410  EDIT_END_ADDR   DEFS    2       ;EDITOR HIGH LIMIT.
FEE3                3411  ST_F            DEFS    2       ;ASSEMBLER SYMBOL TABLE FROM.
FEE5                3412  ST_T            DEFS    2       ;ASSEMBLER SYMBOL TABLE TO.
FEE7                3413  OBJ_F           DEFS    2       ;ASSEMBLER OBJECT CODE FROM.
FEE9                3414  OBJ_T           DEFS    2       ;ASSEMBLER OBJECT CODE TO.
FEEB                3415  END_ADDR        DEFS    2       ;Contains the limit address
                    3416                                  ;of ccommand INSERT or DELETE .
FEED                3417  BRAD    DEFS    2       ;Breakpoint address .
FEEF                3418  BRDA    DEFS    1       ;Data of breakpoint address .
FEF0                3419  POWERUP DEFS    1       ;Power_up initialization .
FEF1                3420  TEST    DEFS    1       ;Bit 7 -- set when illegal key
                    3421                          ;         is entered.
FEF2                3422  STEPFG  DEFS    1       ;STEP mode test flag .
FEF3                3423  PRTFLG  DEFS    1       ;Printer toggle switch .
FEF4                3424  BEEPSET DEFS    1       ;Beep sound toggle switch.
FEF5                3425  FBEEP   DEFS    1       ;Freqency of BEEP .
FEF6                3426  TBEEP   DEFS    2       ;Time duration of BEEP .
FEF8                3427  MADDR   DEFS    2       ;Temporary storage .
FEFA                3428  TEMP1   DEFS    4       ;See comments on command STEP .
FEFE                3429  ATEMP   DEFS    1       ;Temporary storage .
FEFF                3430  HLTEMP  DEFS    2       ;Temporary storage .
FF01                3431  IM1AD   DEFS    2       ;Contains the address of Opcode 'FF'
                    3432                          ;service routine.( RST 38H, mode
                    3433                          ;1 interrupt, etc).
FF03                3434  RCOUNT  DEFS    1       ;Register counts in register table.
FF04                3435  INPBF   DEFS    40      ;Input buffer .
FF2C                3436  DISPBF  DEFS    82      ;Display buffer .
FF7E                3437  GETPT   DEFS    2       ;Temporary storage for GETHL .
FF80                3438  TYPEFG  DEFS    1       ;Type test flag .
FF81                3439  CRSET   DEFS    1       ;Display delay time .
FF82                3440  OUTPTR  DEFS    2       ;Input buffer pointer.
FF84                3441  DISP    DEFS    2       ;Display buffer pointer .
FF86                3442  INPTR   DEFS    2       ;Limit of input buffer pointer .
                    3443  REGBF:
FF88                3444  USERAF  DEFS    2
FF8A                3445  USERBC  DEFS    2
FF8C                3446  USERDE  DEFS    2
```

```
FF8E                3447  USERHL  DEFS    2
FF90                3448  UAFP    DEFS    2               ;AF'
FF92                3449  UBCP    DEFS    2               ;BC'
FF94                3450  UDEP    DEFS    2               ;DE'
FF96                3451  UHLP    DEFS    2               ;HL'
FF98                3452  USERIX  DEFS    2
FF9A                3453  USERIY  DEFS    2
FF9C                3454  USERSP  DEFS    2
FF9E                3455  USERPC  DEFS    2
FFA0                3456  USERIF  DEFS    2
                    3457  BLANK   EQU     6FD0H
                    3458  K_TAB           EQU     068H    ;TAB CODE
                    3459  TVSET           EQU     0A000H  ;The first memory location
                    3460                                  ;of TV interface board .
                    3461  TV              EQU     0A001H  ;The starting address of monitor
                    3462                                  ;program on TV interface board .
                    3463  BASICC          EQU     2020H   ;The starting address of
                    3464                                  ;reenter BASIC .
```

```
SYMBOL  VAL M DEFN REFS

ADDBAC ØAD2   2992 2989
ASM    FFFF X 3519  5Ø6
ATEMP  FEFE G 3429  262  278  317  323 3467
A_Z?   ØB4B G 3137 3515
B3     Ø4A7   1370 1360 1363
BACK   ØØFE    463  482
BASIC3 Ø185    565  51Ø  512
BASICC 2Ø2Ø   3463  573
BASICZ FFFF X 3518  572
BEEP   Ø8Ø3   2249  744
BEEPSE FEF4 G 3424  6Ø1 2250 3475
BEEP_C Ø1A9    6ØØ  518
BITEND Ø7F1   2221 2216
BLANK  6FDØ G 3457  773 3469
BLANKØ ØAEB   3Ø11 3ØØ7
BLOOP  Ø74D   1995 1999
BRAD   FEED G 3417  27Ø  386  7Ø4 1351 1369 1371 1416 3466
BRDA   FEEF G 3418  271  385 1373 3466
BREAK  Ø47D   1349  496
BREAK1 Ø48D   1357 1367
B_SP   ØA6A G 2845 2820 2823 3495
B_TAB  ØA41   2822 2819 2832
B_TAB1 ØA53   2833 2825
B_TAB2 ØA56   2835 2844
CCURSO ØA7B   2868
CHK4Ø  Ø912 G 2548 2767 3476
CHKHE2 Ø8E2 G 2485 1254 1365 1522 3482
CHKHEX Ø8DF G 2483  987 1411 1811 2343 3481
CHKIN1 Ø859   2346 2349
CHKINP Ø854   2338  984
CHRWR  Ø924 G 2574 1242 1354 1619 1622 2238 2353 2355 2357 2744 2796
                   2913 2963 3ØØ3 3Ø1Ø 3Ø16 3498
CLEAR  Ø9B9 G 2715 1659 1727 2236 2397 2625 3497
CLOOP  ØACØ   2981 2995
CLRB   Ø21Ø    698 1362
CLRBF  Ø7F6 G 2235  416  447 2624 3488
CLRDSP Ø84Ø   2325  718 2722
CLRI   Ø217    7Ø6  697 1251
COLDEL Ø05Ø     31  9ØØ
CONT28 Ø03E    258  2Ø4
CONVER Ø821   2293 1922 2583 268Ø 2843 2869
COUNT  FFFF X 3523 28Ø5
COUNT1 Ø75D   2Ø43 2Ø62 2Ø69
CR     Ø93B G 2602 3489
CRØ    Ø971   2628  729 2609 2771
CR1    Ø97A G 2642 349Ø
CR2    Ø981 G 2654  969 24ØØ 3491
CR3    Ø985 G 2666  448  524 1Ø17 1Ø55 1175 123Ø 1259 1293 1338 1374
                   1704 1719 3492
CR4    Ø93D G 2604 2645 2656 2668 3493
CR5    Ø95B   2618 2611 2614
CRSE1  FF81 G 3439 26Ø5 2612 262Ø 277Ø 3466
CURSOR ØA79 G 2866 2585 2626 2768 2853 3505
CV1    Ø8F7   2513 2518
CV2    Ø8FA   2516 2512 2521
CV3    Ø8EF   2509 2531
CVT    Ø8FD   2519 251Ø
CVTHEX Ø9Ø1   2522
```

```
SYMBOL  VAL M DEFN REFS

DEASM   FFFF X 3528  549
DEASM3  017D    545  514
DECBIN  0B28 G 3096 3508
DECDSP  0395 G 1104 1800 2399 2643 3511
DECIMA  0AB8 G 2976 3509
DECLOO  0ACB   2987 2991
DEC_SP  0399   1109 2848
DELAY   0956   2616 2617
DELETE  0448   1283  502
DIG1    0080     16  165  319  895  923 2106
DIG2    0081     17  166  320  897  924 2104
DIG3    0082     18  167  321  804  899  925
DISBR   0480   1351
DISP    FF84 G 3441  722 1000 1110 1919 2307 2312 2549 2674 2694 2720
                    2783 2826 2852 2871 2874 3467
DISPBF  FF2C G 3436  638  721  725  999 1108 1918 2239 2329 2330 2550
                    2553 2675 2693 2719 2784 3468
DUMP    0679   1806  490
DUMP1   067C   1809 1812 1816
DUMP2   068F G 1819 3496
ECHO_C  0860 G 2350  980 1130 1234 1350 1404 1516 1664 1807 1879 3479
EDIT    FFFF X 3520  516
EDIT_E  FEE1 G 3410 3470
EDIT_S  FED9 G 3405 3471
EIDI    04F3   1441 1439
END_AD  FEEB G 3415  713 1208 1239 1257 1286 3468
END_DA  FEDB G 3407 3471
END_LN  FEDD G 3408 3472
EOS?    08C8   2470 2459
ERROR   06C4 G 1871 1067 1153 1167 1214 1331 1829 1952 1955 1961 3477
ERRSMS  0CAF G 3387 1872 3516
ERR_SP  0C72   3352  425
ESCAPE  00E6    445 2774
EXEC2   04D8   1415 1409
F1KHZ   0041     33 2370
F2KHZ   001F     35 2373
FBEEP   FEF5 G 3425  680 3467
FILEDP  0709   1926 1927
FILLDA  03C0   1150  498
FOR     00F1    455  480
GET     066F G 1797  982 1246 1358 1405 1517 1809 1881 3480
GETBIT  0759   2002 1993 1995 2000
GETBYT  0748   1982 1976
GETCHR  08AE G 2452 1819 2507 3483
GETHL   08E5 G 2500 1014 1050 1052 1157 1159 1215 1305 1698 1713 1815
                    2346 3484
GETP    07AF   2157 1061
GETPT   FF7E G 3437 2453 2468 2486 3468
GETPTR  07AC   2143 1852 1946 2126
GETT    0672   1799
GMV     0365   1059 1054 1224 1290
GOEXE1  04C6   1405 1412
GOEXEC  04C3   1403  492
H?      0B0A   3050 3041
HBLOOP  0AFB   3038 3049
HEX1    0AAD G 2957 2936 2939 3512
HEX2    0A9A G 2930 1101 1626 1629 2893 2895 3503
HEX4    0A92 G 2910 1661 1729 3502
```

```
SYMBOL  VAL M DEFN REFS

HEXBIN ØAF4 G 3Ø31 3510
HEXFLA FFFF X 3525 3Ø34 3Ø55 3Ø84
HEXX   ØA89 G 2891  967 1131 1240 1352 2911 3499
HHH    ØAB5   2962 2960
HLTEMP FEFF G 3430  2Ø3  279  324  328 3467
IGNORE Ø177    528  462  47Ø
IMlAD  FFØ1 G 3431  248  687 3467
INI    Ø1BØ    611  175
INI1   Ø1DC    663  67Ø
INI2   Ø1DE    664  666
INI3   Ø1F4    675
INI4   Ø2Ø2    686
INI5   Ø1F2    674  672
INI6   Ø1CB    632  733
INI7   Ø21E    715  3ØØ  637
INI8   Ø241    731  624
INPBF  FFØ4 G 3435  719  997 1248 1253 1359 1364 1407 1519 2413 2484
                   2717 3468
INPTR  FF86 G 3442 2765 28Ø1 28Ø8 3469
INSET  Ø3E6   12Ø2  5ØØ
INSET1 Ø426   1245 1255
INSET2 Ø44Ø   1258 1249 1252
INSET3 Ø3E9   12Ø4 1231
INSET4 Ø415   1232 12Ø3 1284
JUMP   Ø462   13Ø2  5Ø4
KØ     ØB52   3147
K1     ØB53   3148
K1Ø    ØB62   3163
K11    ØB63   3164
K12    ØB64   3165
K13    ØB65   3166
K15    ØB67   3168
K16    ØB68   3169
K18    ØB6A   3171
K19    ØB6B   3172
K1B    ØB6D   3174
K1C    ØB6E   3175
K1D    ØB6F   3176
K1E    ØB7Ø   3177
K1F    ØB71   3178
K2     ØB54   3149
K21    ØB73   318Ø
K22    ØB74   3181
K24    ØB76   3183
K25    ØB77   3184
K27    ØB79   3186
K28    ØB7A   3187
K2A    ØB7C   3189
K2B    ØB7D   3190
K2D    ØB7F   3192
K2E    ØB8Ø   3193
K3     ØB55   3150
K3Ø    ØB82   3195
K31    ØB83   3196
K32    ØB84   3197
K33    ØB85   3198
K34    ØB86   3199
K35    ØB87   32ØØ
```

```
SYMBOL  VAL M DEFN REFS

K36     0B88    3201
K37     0B89    3202
K38     0B8A    3203
K39     0B8B    3204
K3A     0B8C    3205
K4      0B56    3151
K5      0B57    3152
K6      0B58    3153
K7      0B59    3154
K7B     0B8D    3206
K8      0B5A    3155
K9      0B5B    3156
KA      0B5C    3157
KB      0B5D    3158
KC      0B5E    3159
KCOL    02A6     889  937
KCTRL   0281     815  808
KD      0B5F    3160
KE      0B60    3161
KEYEXE  010B     478  418
KEYMAP  026C     797
KEYTAB  0B52    3146  798
KF      0B61    3162
KIN     0092      22  164  277  322  904 1485 2096 2102 2107 2113 2259
                   2378
KK12    0B72    3179
KK13    0B75    3182
KK14    0B7B    3188
KK15    0B7E    3191
KK19    0B78    3185
KK24    0B81    3194
KK2F    0B66    3167
KK3C    0B69    3170
KK3E    0B6C    3173
KROW    02C5     908  920
KSHIFT  0285     823  806
K_TAB   0068 G  3458 2791 3469
LASM    FFFF X  3522  508
LDA     08B1    2454 1526 2482
LEAD    06D0 G  1883 1896 1913 1938 3513
LEAD1   06DB    1892 1902
LEAD2   06E5    1906 1907
LEFT    0A26    2807 2778
LOAD    06CA    1878  488
LOOP3   0700    1921 1924
LOOP4   0716    1934 1940
LOOPH   077D    2096 2099
LOOPL   0790    2107 2110
MADDR   FEF8 G  3427  990 1096 1123 1128 1135 3467
MAIN    00C8 G   407  419  449 3517
MBACK   03B7    1133  466
MDUMP   FFFF X  3527 1037
MDUMP1  0343    1032  962
MEM3    0386    1095  968
MEM5    038B    1098 1103
MEMDP2  0628    1726  399
MEMDP3  0631    1730 1696 1708
MEMEX1  0319     982  985
```

```
SYMBOL  VAL M DEFN REFS

MEMEX2 0316    979  955 1151 1261 1303 1388
MEMEX3 0328    994  965 1605
MEMEXC 02F3    954  484
MFOR   03A2   1121  458
MMODFY 0337   1011  960
MMOVE  0350   1048  964
MPERIO 0020     37 2115
MPFII  0C5D G 3345  723 3478
MSG    09CA G 2739  724 2398 2745 3501
MTP201 FFFF X 3533
MTPPRT FFFF X 3529 2414
MVUP   0381   1084 1074
NDIGIT 0B2C   3100 3116
NMI    0066    309  215  686
NOKEY  02CB    919  911
NOTHEX 0910   2532 2524
NOTONE 0817   2260 2253
NOTTAB 0A11   2795 2792
OBJ_F  FEE7 G 3413 3472
OBJ_T  FEE9 G 3414 3472
OLOOP  07CF   2199 2202
ONE    0B14 G 3072 2523 3040 3494
ONE_1K 0004     51 2220
ONE_2K 0004     52 2218
OUT0   07E1   2213
OUT1   07EA   2217 2212
OUTBIT 07DC   2206 2198 2200 2204
OUTBYT 07C9   2190 2186
OUTPTR FF82 G 3440  720  998 1932 2577 2580 2607 2685 2687 2718 2764
                    2799 2809 2816 2834 3469
P101   03A9   1128 1140
P102   030F    968 1132
P105   054C   1542 1538
P106   055F   1563 1559
P111   04BF   1390 1386
P82551 0083     15  150
P82552 0093     19  156
PERIOD 077A   2087 1892 1906 2043
POWERU FEF0 G 3419  173  675 3466
PREOUT 04DF   1427 1393
PRINT0 0AE6   3008 3014
PRINTT 0893 G 2410 2618 3485
PRTF   01A5    591  589
PRTFLG FEF3 G 3423  584  586  613 2432 3466
PRTMES 0886   2396  430  592 1873
PRTOFF 0C89   3365  588
PRTON  0C82   3362  590
PRT_CO 0195 G  583  522 3474
PRT_M2 FFFF X 3532
PRT_MP FFFF X 3531  673
PTEST  08A3 G 2431 1033 2410 2610 3514
PTESTT 08A8   2435  546  671
PWCODE 00A5     23  174  674
RAM2K_ 0C91   3368  732
RAM4K_ 0C9D   3375  631
RAMCHK 0819   2272  371  374  622 1152
RAMT1  01B9    621
RAMT2  01BC    622  626
```

```
SYMBOL  VAL M DEFN REFS

RAM_ST FEDF G 3409  633 3470
RBACK  05ED   1673  468
RBACK1 05F8   1681 1650 1677
RBACK2 05F6   1679 1651
RCOUNT FF03 G 3434 1606 1611 1616 1644 1652 1666 1675 1779 3468
RDISP6 0582   1606
RDLOOP 09DA G 2766 1798 2780 2782 2790 2797 2812 2821 2839 2841 3504
RDSPL0 0578   1599 1584
RDSPL1 0593   1617 1608 1633
RDSPL3 056F   1585 1548
RDSPL4 056B   1581 1672
RDSPLY 0576   1597 1572 1588
RD_END 0A16   2798 2776
READLN 09D4 G 2762 1799 3506
REEDIT FFFF X 3521  520
REG2   088C   2399 1636
REGALL 0568   1578 1532 1603 1730
REGBF  FF88   3443 1443 1780
REGEX2 0528   1517 1524
REGEXC 0525   1515  486
RESET1 0032    222  181
RESET2 0054    288  224
RFOR   05B9   1643  460
RFOR1  05C6   1652 1678 1683
RFOR2  05DA   1663 1658
RFOR3  05DF   1665 1662
RGSAVE 0076    324
RL1    02E0    932  930
RL2    02E6    935  933
RMODF1 0612   1707 1564
RMODF2 060B   1703 1706
RMODF3 061C   1712 1710
RMODFY 05FC   1694 1546
RODD   060F   1705 1701
RS_STA 0003    142  144
RTABLE 0B8E   3207 1665 1743
SCAN   0246 G  742  409 2772 3486
SCAN1  029B G  854  665  780  790 1926 2616 3487
SCAN2  024D    763  743
SCLOOP 0267    790  791  826  828  835
SCNX   025C    780  784
SCPRE  025A    779  772  781
SEARC_ 0634   1742 1601 1695 1707
SEG1   0090     20  890 1890 1943
SEG2   0091     21  893 1884 1945
SEGTAB 0BA8   3234 2297
SERCH  063D   1749 1786
SERCH1 0667   1784 1751
SERCH2 064C   1763 1754 1760
SERCH3 065D   1777 1770
SERCH4 0669   1786 1762
SERCH5 065E   1779
SET    0337   1012 1019
SETIF  00A3    361  359
SETST0 00BE    384  306  432
SETST2 00D8    424  372  375
SETST3 00DD    427  383
SETST4 00E0    429  426
```

```
SYMBOL  VAL M DEFN REFS

SHIFTT 0C38   3307  829
SHORTP 076C   2063 2052
SK1    0B48   3134 3131
SKIP   0B40 G 3128 3136 3507
SKIP_  08BA   2460 2457 2464 2466 2474 2476 2478 2480
SPACE1 0A95 G 2912 1099 1617 1623 3500
SQWAVE 087A   2378 2383
STEP   04B1   1381  494
STEPBF FED0 G 3403 1049 1051 1053 1060 1070 1080 1205 1207 1210 1225
                   1227 1285 1287 1289 1814 1818 1820 1832 1838 1909
                   1920 1957 2156 3466
STEPFG FEF2 G 3422 1383 1429 1653 3466
STPTR  08C4   2467 2472
ST_F   FEE3 G 3411 3472
ST_T   FEE5 G 3412 3472
SUM    07A3   2130
SUM1   079F   2120 1823 1956
SUMCAL 07A4   2138 2140
SUPRES 0AD9   2997 2994
SYSSTK FED0   3402  168  363  408  446
SYS_SP 0C7B   3357  428
TAB?   09A9   2690 2681 2824 2840
TAB??  09A0   2684 2678
TAB?LP 09B3   2698 2701
TABOUT 0989   2673 2582 2682
TAB_RE 0935   2584 2683 2688
TAPEIN 073B   1966 1912 1954
TAPEOU 07BF   2179 1845 1855 2188
TBEEP  FEF6 G 3426 2256 3467
TEMP   FFFF X 3524 2932
TEMP1  FEFA G 3428 1427 1442 1473 1487 3467
TENS   0CA9   3382 2978
TERR   0775   2082 2048
TEST   FEF1 G 3420  291  529  765 3466
TEST5  FFFF X 3530 1036
TEXT_F FED9 G 3404 3473
TEXT_T FEDB G 3406 3473
TLOOP  073D   1976 1979
TNEXT  01C3    625  623
TONE   0874   2374 2257 2371
TONE1K 086E   2369 1837 2221
TONE2K 0872   2372 1851 1859 2214 2219
TV     A001 G 3461 2631 3469
TVSET  A000 G 3459 2629 3469
TYPEFG FF80 G 3438  456  464  727  957 1635 3468
UAFP   FF90   3448
UBCP   FF92   3449
UDEP   FF94   3450
UHLP   FF96   3451
USERAF FF88   3444 1462 1486
USERBC FF8A   3445
USERDE FF8C   3446
USERHL FF8E   3447
USERIF FFA0   3456  289  353  361 1430 1463
USERIX FF98   3452
USERIY FF9A   3453  330
USERPC FF9E   3455  326 1389 1414 1471 1660 1728
USERSP FF9C   3454  299  329  369 1461
```

```
SYMBOL  VAL M DEFN REFS

USERST FEAØ   3399  298  381  696
YES_Ø  ØAE2   3ØØ4 2999
ZERO_1 ØØØ2     53 2215
ZERO_2 ØØØ8     54 2213
ZSUM   ØØ94     24  231
```

MULTITECH INDUSTRIAL CORPORATION

OFFICE/ 977 MIN SHEN E ROAD TAIPEI 105.
TAIWAN. R.O.C.
TEL:(02)769-1225(10 LINES)
TLX:23756 MULTIIC, 19162 MULTIIC.

FACTORY/5, TECHNOLOGY ROAD III 111
HSINCHU SCIENCE. BASED INDUSTRIAL PARK
HSINCHU TAIWAN. 300. R.O.C.
TEL:(035)775102(3 LINES)

DOC. NO.: M1PM06-304A