CHAPTER 4 CP/M II (EXTENDED CP/M) CONTENTS

CHAPTER 4 CP/M II (Extended CP/M)

This chapter describes the extended facilities of the PINE CP/M. The topics covered in this chapter include the following:

- User BIOS
- Jump tables
- Hook processing
- Bank management
- Resident facility
- ROM program execution
- Interrupt handling

For the basic BDOS and BIOS functions, see Chapter 3, "CP/M I (BDOS, BIOS).

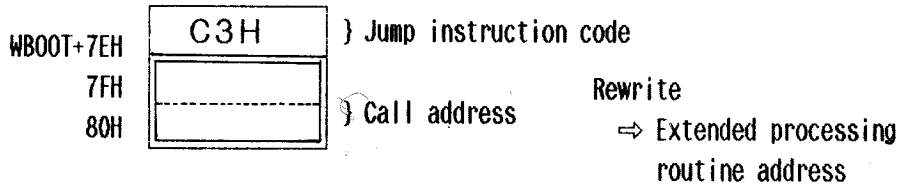
4.1 User BIOS

4.1.1 General

PINE OS provides a BIOS entry, named User BIOS, which allows the user to expand or add user-supplied BIOS routines to BIOS. It also reserves a user BIOS area for use by the user BIOS routines. The user BIOS area may be used by not only the user BIOS routines but also by common machine-language routines (e.g., barcode reader programs) and extended hook processing routines (e.g., alarm time updating routines). This section explains how to use the user BIOS functions and the user BIOS area.

4.1.2 User BIOS

4.1.2.1 Location of the entry to the user BIOS
The PINE has two BIOS entries: RBIOS1 and RBIOS2. Application programs which are to expand the user BIOS capability must rewrite the USERBIOS call address in RBIOS1.



The location of WBOOT is stored in addresses 0001H and 0002H.)

Application programs must observe the following procedure when calling USERBIOS.

(1) Calling USERBIOS from a load-and-go program
Find the entry address of WBOOT in addresses 0001H and 0002H and call WBOOT + 7EH as when calling other BIOS functions.
(2) Calling USERBIOS from a ROM-based program
Since the RBIOS1 area is likely to be used as a background bank when a ROM-based program is running, it is necessary to switch to bank 0 with a CALLXX (0FFAEH) before calling USERBIOS. The address of USERBIOS can be obtained in the same way as in paragraph (1). See Section 4.2, "Jump Tables" for the use of the CALLXX instruction.

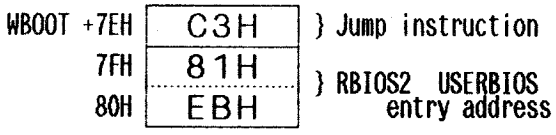
4.1.2.2 Programming notes

Care must be exercised with the following when using user BIOS:
(1) Extending BIOS functions using the user BIOS area
Observe the precautions about the user BIOS area given in 4.1.3 when inserting data pertaining to the extended user BIOS function into the user BIOS area.
(2) Location of user BIOS
The user BIOS is located on bank Ø (RAM). When user BIOS is called, however, the active bank remains to be the one on which the calling program resides. This means that the active bank is either bank 1 or 2 when user BIOS is called from a ROM-based program and bank Ø when it is called from a load-and-go program. When calling extended user BIOS from a ROM-based program, switch to bank Ø before calling user BIOS.

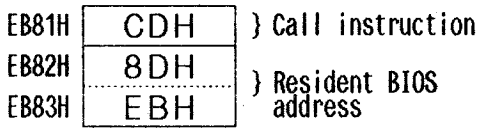
4.1.2.3 Extended user BIOS processing

Note the following when constructing extended user BIOS processing routines:
(1) Every user BIOS routine must end with a RET instruction.
(2) The user stack remains active when user BIOS is called. If necessary, the extended user BIOS processing routine must reserve and use a stack for itself.
(3) For user BIOS to remain in memory after termination of a program, the extended portion of user BIOS must be placed in the user BIOS area. If it is unnecessary to retain user BIOS, reset the user BIOS call address to the old value.

Example 1: Resetting RBIOS1 USERBIOS



Example 2: Resetting RBIOS2 USERBIOS



4.1.2.4 Initializing user BIOS

PINE OS initializes the user BIOS entry (USERBIOS) as well as the system BIOS entries when reset or system initialize processing is carried out. As initialized, USERBIOS executes only a RET instruction.

4.1.3 User BIOS Area

4.1.3.1 Location of the user BIOS area

The user BIOS area is reserved between the internal RAM disk unit and the item key table. Figure 4.1.1 shows the location of the user BIOS area in memory.

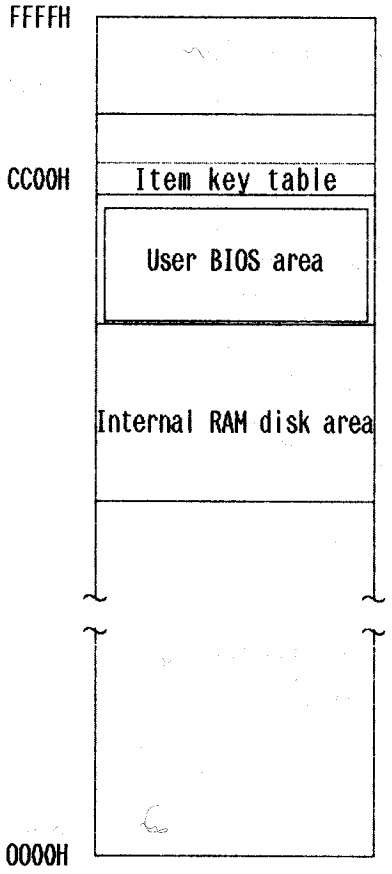


Fig. 4.1.1 User BIOS Area

4.1.3.2 Reserving the user BIOS area

(1) Outline
The user BIOS area can be reserved in one of the following ways:
a) Reserve at system initialize time.
b) Reserve using the CONFIG utility.
c) Reserve using a user-supplied program.

See Section 2.2, "System Initialize" for method a) and the CONFIG manual for method b). Here, the user BIOS reservation procedure using a user-supplied program is introduced.
(2) Reserving the user BIOS area
Follow the flowchart shown in Figure 4.1.2 to reserve space for the user BIOS area. The user BIOS area must be reserved in 256-byte increments.

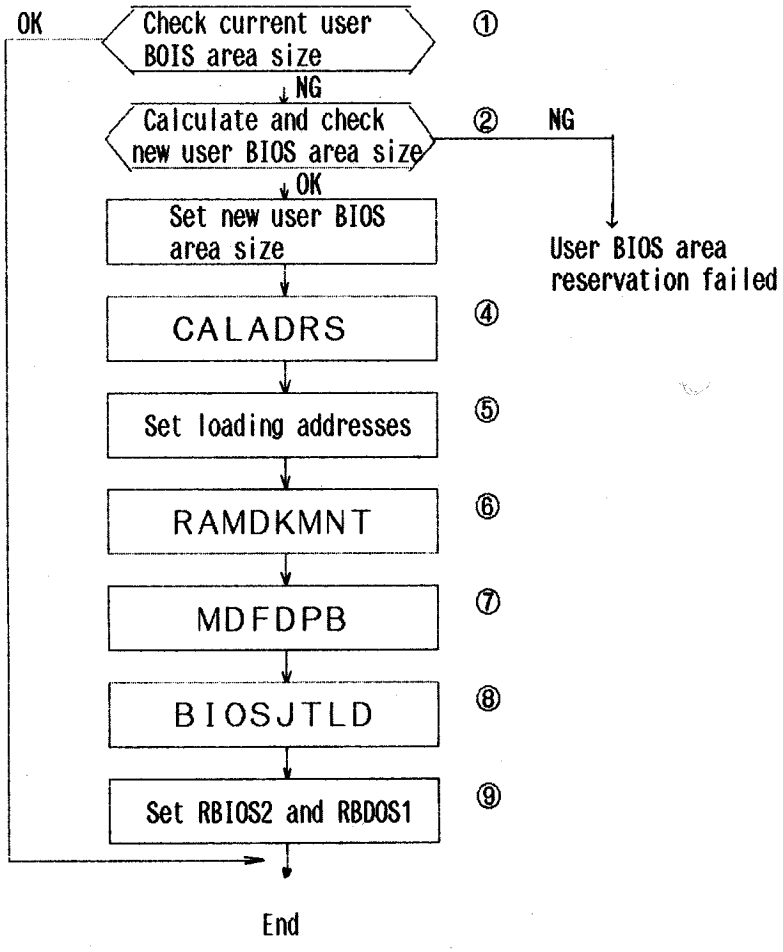


Fig. 4.1.2 Reserving the User BIOS Area

Step 1: Check the current user BIOS area size.
Check the size of the current user BIOS area for validity.
The current BIOS area size is stored in USERBIOS (ØEF2DH).

Step 2:Calculate the size of the new user BIOS area.
Check the size of the required user BIOS area and the internal RAM disk to see whether their sum is less than 35.5K bytes.
The user BIOS area cannot be reserved if the sum is greater than 35.5K bytes. The size of the current internal RAM disk is given in 1K bytes in SIZRAM (ØEF2CH).

Step 3: Set the size of the new user BIOS area.
Load the size in 256-byte units of the new user BIOS area into USERBIOS (ØEF2DH).

Step 4: Call CALADRS.
CALADRS calculates the size of CP/M. CALADRS is cataloged in the jump table on OS ROM and its address is ØØ18H.
Use the BIOS CALLX function to call CALADRS. See Section 4.2, "Jump Tables" for CALADRS and Section 3.4, "BIOS Details" for CALLX.

Step 5: Set the loading addresses.
Loads the CP/M loading addresses that are returned by CALADRS into the system area.

(BC) → TOPRAM (ØEF94H)
(DE) → BIlLAD (ØEF26H)
(IX) → BDSLAD (ØEF24H)
(IY) → CCPLAD (ØEF22H)

Step 6: Call RAMDKMNT.
RAMDKMNT checks for the presence of a RAM disk. RAMDKMNT is cataloged in the jump table on OS ROM and its address is ØØlEH. Use the BIOS CALLX function to call RAMDKMNT.

Step 7: Call MDFDDB.
Modify the contents of the disk parameter block. MDFDPB is cataloged in the jump table on OS ROM and its address is ØØ21H. Use the CALLX function to call MDFDPB.

Step 8: Call BIOSJTLD.
BIOSJTLD loads the contents of the BIOS jump table. BIOSJTLD is cataloged in the jump table on OS ROM and its address is ØØlBH. Use the BIOS CALLX function to call BIOSJTLD.

Step 9: Set RBIOSl and RBDOSl.
Set the entry address of RBIOSl and RBDOSl.

BIlLAD (ØEF26H) + 3 → (ØØØlH)
BDSLAD (ØEF24H) + 6 → (ØØØ6H)

(3) Programming notes

1. BIOS in RBIOS2 must be used when reserving the user BIOS area. This is because RBIOSl is relocated as the user BIOS area expands or shrinks.
2. Once the user BIOS area is reserved, do not use RBDOSl (at address ØØØ5H) until WBOOT is invoked. This is because once the user BIOS area expands or shrinks, RBDOSl is not loaded until WBOOT is invoked. Use RBDOS2 when using BDOS.
3. The sum of the sizes of the user BIOS area and internal RAM disk must not exceed 35.5K bytes. Otherwise, a system initialize will occur when the RESET switch is pressed (this does not hold true at power-on time because in such a case no size check is performed).

(4) Reference
The table below lists the work areas that are affected when the user BIOS area size is updated.

SIZRAM (ØEF2CH) 1 byte
Contains the size in 1K-byte units of the internal RAM disk.
Ø ≤ SIZRAM ≤ 35, SIZRAM ≠ 1

USERBIOS (ØEF2DH) 1 byte
Contains the size in 256-byte units of the user BIOS area.
Ø ≤ SIZRAM ≤ 142,
(SIZRAM x 4) + USERBIOS ≤ 142.

CCPLAD (ØEF22H) 2 bytes
Contains the CCP loading address.

BDSLAD (ØEF24H) 2 bytes
Contains the RBDOS1 loading address.

BIlLAD (ØEF26H) 2 bytes
Contains the RBIOS1 loading address.

TOPRAM (ØEF94H) 2 bytes
Contains the user BIOS area starting address. Set to CCØØH if no user BIOS area is reserved.

```
                        ;       ************************************************
                        ;               CHANGE RAM DISK & USER BIOS SIZE PROGRAM
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;               This sample program is changing RAM disk
                        ;               and User BIOS size.
                        ;
                        ;       <> assemble condition <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address <>
                        ;
                                .PHASE  100H
                        ;
                        ;       <> constant values <>
                        ;
                        ;       BIOS entry
                        ;
EB03                    WBOOT           EQU     0EB03H          ; Warm Boot entry
EB0C                    CONOUT          EQU     WBOOT   +09H    ; Console out entry
EB69                    CALLX           EQU     WBOOT   +66H    ; Call extra entry
                        ;
                        ;       System area
                        ;
EF94                    TOPRAM          EQU     0EF94H          ; Top of User BIOS
EF26                    BIlLAD          EQU     0EF26H          ; RBIOSl loading addr
EF24                    BDSLAD          EQU     0EF24H          ; RBDOSl loading addr
EF22                    CCPLAD          EQU     0EF22H          ; CCP loading addr
EF9D                    QTRAMEX         EQU     0EF9DH          ; Quantity of external RAM disk
EF9C                    QTRAMIN         EQU     0EF9CH          ; Quantity of internal RAM disk
EF2D                    USERBIOS        EQU     0EF2DH          ; Size of User BIOS area
F77A                    YSIZERAM        EQU     0F77AH          ; Size of RAM disk
EF2C                    SIZRAM          EQU     0EF2CH          ; Size of internal RAM disk
F52E                    DISBNK          EQU     0F52EH          ; Destination bank for CALLX
                        ;
                        ;       Bank value
                        ;
00FF                    SYSBANK         EQU     0FFH            ; System bank
0000                    BANK0           EQU     000H            ; Bank 0 (RAM)
0001                    BANK1           EQU     001H            ; Bank 1 (ROM capsel 1)
0002                    BANK2           EQU     002H            ; Bank 2 (ROM capsel 2)
                        ;
                        ;       User BIOS area
                        ;
CBF0                    UB_HEAD         EQU     0CBF0H          ; Top addr of User BIOS area's header
CBFB                    UB_OVWRITE      EQU     UB_HEAD +11     ;  Over write flag
CBFC                    UB_RELEASE      EQU     UB_HEAD +12     ;  Release address
                        ;
                        ;
0001                    BIOSENTRY       EQU     00001H          ; CP/M BIOS entry addr
0006                    BDOSENTRY       EQU     00006H          ; CP/M BDOS entry addr
                        ;
                        ;       OS ROM jump table
                        ;
0018                    CALADRS         EQU     00018H          ; Calculate loading addr
001E                    RAMDKMNT        EQU     0001EH          ; RAM disk mount check
0021                    MDFYDPB         EQU     00021H          ; Modify disk parameter block
001B                    BIOSJTLD        EQU     0001BH          ; BIOS jump table load
                        ;
                        ;       New RAM disk size and User BIOSsize
                        ;
001E                    SRAMDISK        EQU     30
0004                    SUSERBIOS       EQU     4
                        ;
008E                    MAXSIZE         EQU     142             ; 35.5 KB * 4
                        ;
                        ;       ************************************************
                        ;               MAIN PROGRAM
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;               This program is changing size as following.
                        ;                       RAM disk  size --> 30 kbytes
                        ;                       User BIOS size --> 4  kbytes
                        ;
0100                    MAIN:
0100    31 1000                 LD      SP,1000H
0103    06 1E                   LD      B,SRAMDISK
0105    0E 10                   LD      C,SUSERBIOS*4
0107    CD 01A4                 CALL    CHNGSZ
                        ;
010A    CD 0110                 CALL    MESSAGE
010D    C3 EB03                 JP      WBOOT
                        ;
                        ;
                        ;
                        ;       ************************************************
                        ;               RETURN MESSAGE DISPLAY
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;
                        ;       <> entry parameter <>
                        ;               A  : Message parameter
                        ;       <> return parameter <>
                        ;               NON
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;
0110                    MESSAGE:
0110    21 0128                 LD      HL,MSGTBL       ; Message table top addr.
0113    87                      ADD     A,A             ; Get target message.
0114    4E                      LD      C,A             ;  A*4      --> C
```

```
0115    06 00                   LD      B,0             ;  0        --> B
0117    09                      ADD     HL,BC           ;  HL + BC  --> HL
0118    5E                      LD      E,(HL)          ;  (HL)     --> HL
0119    23                      INC     HL              ; HL is top addr of message,
011A    56                      LD      D,(HL)          ;
011B    EB                      EX      DE,HL           ;
                        ;
011C                    MSGLOOP:
011C    4E                      LD      C,(HL)          ; Get message.
011D    0D                      DEC     C               ; Data is 0?
011E    0C                      INC     C               ;
011F    C8                      RET     Z               ; Yes.
                        ;
0120    E5                      PUSH    HL              ;
0121    CD EB0C                 CALL    CONOUT          ; Display message.
0124    E1                      POP     HL              ;
0125    23                      INC     HL              ; Pointer update.
0126    18 F4                   JR      MSGLOOP         ; Loop until find 0.
                        ;
                        ;       Message table
                        ;
0128                    MSGTBL:
0128    0130                    DW      MSG1            ;
012A    0152                    DW      MSG2            ;
012C    0165                    DW      MSG3            ;
012E    0188                    DW      MSG4            ;
                        ;
                        ;       Message data
                        ;
0130                    MSG1:
0130    43 68 61 6E             DB      'Changing size is normaly ending',0DH,0AH,00H
0134    67 69 6E 67
0138    20 73 69 7A
013C    65 20 69 73
0140    20 6E 6F 72
0144    6D 61 6C 79
0148    20 65 6E 64
014C    69 6E 67 0D
0150    0A 00
0152
0152    50 61 72 61             DB      'Parameter error.',0DH,0AH,00H
0156    6D 65 74 65
015A    72 20 65 72
015E    72 6F 72 2E
0162    0D 0A 00
0165                    MSG3:
0165    55 73 65 72             DB      'User BIOS area cannot destroyed.',0DH,0AH,00H
0169    20 42 49 4F
016D    53 20 61 72
0171    65 61 20 63
0175    61 6E 6E 6F
0179    74 20 64 65
017D    73 74 72 6F
0181    79 65 64 2E
0185    0D 0A 00
0188                    MSG4:
0188    4F 76 65 72             DB      'Overwrite User BIOS area.',0DH,0AH,00H
018C    77 72 69 74
0190    65 20 55 73
0194    65 72 20 42
0198    49 4F 53 20
019C    61 72 65 61
01A0    2E 0D 0A 00
                        ;
                        ;
                        ;       ************************************************
                        ;               SIZE CHANGING UTILITY
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;
                        ;       <> entry parameter <>
                        ;               B   : New RAM disk  size (unit 1  kbytes)
                        ;               C   : New User BIOS size (unit 256 bytes)
                        ;       <> return parameter <>
                        ;               A   : Return information
                        ;                       =00H -- Normal return
                        ;                       =01H -- Entry parameter is size over
                        ;                       =02H -- Cannot getting User Bios area
                        ;                       =03H -- Overwrite User BIOS area
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;               If new User BIOS size doesn't equal
                        ;               to 0, the programor data in old User
                        ;               BIOS area will by destroyed.
                        ;               If old User BIOS area inhibits over-
                        ;               write by other program, this routine
                        ;               will go back by error return code.
                        ;
01A4                    CHNGSZ:
01A4    3A EF9D                 LD      A,(QTRAMEX)     ; Get quantity of external RAM disk.
01A7    B7                      OR      A               ; Size is 0?
01A8    28 02                   JR      Z,IN_RAM        ; Yes,
01AA    06 00                   LD      B,00H           ; Change new RAM disk size to 0.
01AC                    IN_RAM:
01AC    78                      LD      A,B             ; Check area size.
01AD    87                      ADD     A,A             ;  Calculate by unit 256 bytes.
01AE    87                      ADD     A,A             ;
01AF    81                      ADD     A,C             ;  Add RAM disk size and User BIOS size.
01B0    FE 8F                   CP      MAXSIZE+1       ;  New size is OK?
01B2    3E 01                   LD      A,01H           ;   Return parameter.
01B4    D0                      RET     NC              ;  Return if size over,
                        ;
01B5    AF                      XOR     A               ; Set return parameter.
```

```
01B6    32 024F                 LD      (RET_PRM),A     ;  0 --> RET_PRM
01B9    3A EF2D                 LD      A,(USERBIOS)    ; Get User BIOS area size.
01BC    B7                      OR      A               ; No User BIOS area?
01BD    28 16                   JR      Z,CHNG_UB       ;  Yes.
                ;
01BF    CD 021D                 CALL    CHK_HEAD        ; Check User BIOS header.
01C2    20 11                   JR      NZ,CHNG_UB      ;  No User BIOS header.
01C4    3E 03                   LD      A,03H           ; User BIOS already exists.
01C6    32 024F                 LD      (RET_PRM),A     ;  3 --> RET_PRM
01C9    CD 0239                 CALL    RELEAS          ; User BIOS area release.
01CC    30 07                   JR      NC,CHNG_UB      ;  Release OK.
01CE    3E 02                   LD      A,02H           ; Set error return parameter.
01D0    32 024F                 LD      (RET_PRM),A     ;  2 --> RET_PRM
01D3    18 04                   JR      CHNG_RAM        ;
                ;
01D5                    CHNG_UB:
01D5    79                      LD      A,C             ; Change User BIOS area size.
01D6    32 EF2D                 LD      (USERBIOS),A    ;  New User BIOS area size.
                ;
01D9                    CHNG_RAM:
01D9    78                      LD      A,B             ; Change internal RAM disk size.
01DA    32 EF2C                 LD      (SIZRAM),A      ;  New internal RAM disk size.
                ;
01DD    3E FF                   LD      A,SYSBANK       ; Set desination bank.
01DF    32 F52E                 LD      (DISBNK),A      ;  FFH --> DISBNK
01E2    DD 21 0018              LD      IX,CALADRS      ; Calculate loading addr.
01E6    CD EB69                 CALL    CALLX           ;
                ;
01E9    ED 43 EF94              LD      (TOPRAM),BC     ; Set new loading addrs.
01ED    ED 53 EF26              LD      (BI1LAD),DE     ;
01F1    DD 22 EF24              LD      (BDSLAD),IX     ;
01F5    FD 22 EF22              LD      (CCPLAD),IY     ;
01F9    13                      INC     DE              ; Set BIOS entry.
01FA    13                      INC     DE              ;
01FB    13                      INC     DE              ;
01FC    ED 53 0001              LD      (BIOSENTRY),DE  ;
                ;
0200    DD 21 001E              LD      IX,RAMDKMNT     ; RAM disk mount check.
0204    3A EF2C                 LD      A,(SIZRAM)      ;  Entry parameter. (RAM disk size)
0207    B7                      OR      A               ;  No format.
0208    CD EB69                 CALL    CALLX           ;
                ;
020B    DD 21 0021              LD      IX,MDFYDPB      ; Modify disk parameter block.
020F    CD EB69                 CALL    CALLX           ;
0212    DD 21 001B              LD      IX,BIOSJTLD     ; BIOS jump table loading.
0216    CD EB69                 CALL    CALLX           ;
                ;
0219    3A 024F                 LD      A,(RET_PRM)     ; Restore return parameter. (0 or 2)
021C    C9                      RET                     ;
                ;
                ;
                ;       ************************************************
                ;               CHECK USER BIOS HEADER
                ;       ************************************************
                ;
                ;       NOTE :
                ;               Check of User BIOS header
                ;                1. First 2 bytes of header is 'UB'?
                ;                2. Check sum of header is OK?
                ;
                ;       <> entry parameter <>
                ;               NON
                ;       <> return parameter <>
                ;               Z-flag : Return information.
                ;                        =1 : Header exists.
                ;                        =0 : Header doesn't exists.
                ;       <> preserved registers <>
                ;               BC,DE,HL
                ;
                ;       CAUTION :
                ;
021D                    CHK_HEAD:
021D    3A CBF0                 LD      A,(UB_HEAD)     ; User BIOS header check.
0220    FE 55                   CP      'U'             ;  Header is 'UB'?
0222    C0                      RET     NZ              ;  No.
0223    3A CBF1                 LD      A,(UB_HEAD+1)   ;
0226    FE 42                   CP      'B'             ;
0228    C0                      RET     NZ              ;  No.
                ;
0229    E5                      PUSH    HL              ; Save registers.
022A    C5                      PUSH    BC              ;
022B    21 CBF0                 LD      HL,UB_HEAD      ; Sum check.
022E    06 10                   LD      B,16            ;  Header size.
0230    AF                      XOR     A               ;
                ;
0231                    CHK_SUM:
0231    86                      ADD     A,(HL)          ;  Add header data.
0232    23                      INC     HL              ;  Next address.
0233    10 FC                   DJNZ    CHK_SUM         ;  Loop 16 times.
                ;                                       ; If result isn't 0,
0235    B7                      OR      A               ;  then non-exist User BIOS.
0236    C1                      POP     BC              ; Restore registers.
0237    E1                      POP     HL              ;
0238    C9                      RET                     ;
                ;
                ;       ************************************************
                ;               USER BIOS AREA OVERWITE CHECK
                ;       ************************************************
                ;
                ;       NOTE :
                ;               Check overwite flag in User BIOS area.
                ;               If overwrite OK, then call release routine
                ;               and clear header.
                ;               if overwrite NG, then return with CY on.
                ;                1. Using SETERR and RSTERR
                ;                2. Replacing BDOS error vector
```

```
                        ;       <> entry parameter <>
                        ;               NON
                        ;       <> return parameter <>
                        ;               CY-flag : Return information
                        ;                         =0 -- Nrmal return
                        ;                         =1 -- Cannot overwrite
                        ;       <> preserved registers <>
                        ;               BC,DE,HL
                        ;
                        ;
0239                    RELEAS:
0239    3A CBFB                 LD      A,(UB_OVWRITE)  ; Check over write flag.
023C    B7                      OR      A               ;  Cannot over write?
023D    37                      SCF                     ;
023E    C8                      RET     Z               ; Yes. then return with carry on.
                        ;
023F    E5                      PUSH    HL              ; Save registers.
0240    D5                      PUSH    DE              ;
0241    C5                      PUSH    BC              ;
0242    21 024A                 LD      HL,REL_RET      ; Set return address.
0245    E5                      PUSH    HL              ;
0246    2A CBFC                 LD      HL,(UB_RELEASE) ; Get release routine addr.
0249    E9                      JP      (HL)            ; Go release routine!
024A                    REL_RET:
024A    B7                      OR      A               ; Carry off.
024B    C1                      POP     BC              ; Restore registers,
024C    D1                      POP     DE              ;
024D    E1                      POP     HL              ;
024E    C9                      RET                     ;
                        ;
                        ;       Work area
                        ;
024F                    RET_PRM:
024F                    DS      1               ; Return parameter area.
                        END
```

4.1.3.3 Using the user BIOS area

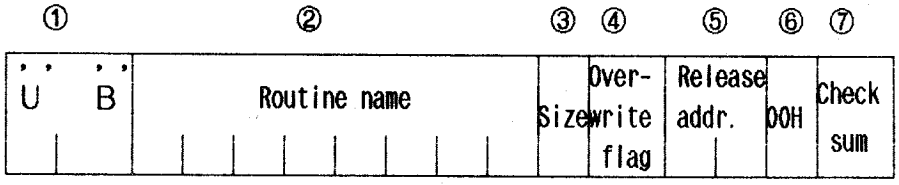
(1) Outline

The user BIOS area cannot be used by more than one program at a time. To identify which program is currently using the user BIOS area, a header is provided at the end of the user BIOS area. Application programs which are to use programs or data in the user BIOS area can identify the program or data that they are going to use by examining this header. Furthermore, when loading a program or data into the user BIOS area, they can check whether the user BIOS area is being used by another program by examining the header.

The contents of the user BIOS area remain unchanged until a system initialize is executed or its size is altered.

(2) Header

The format of the header is shown below.



The header is 16 bytes long and lies between ØCBFØH through ØCBFFH.

(a) Header fields

Header ID (ØCBFØH - ØCBF1H):
Identifies the header area. This field is set to 'UB' (ASCII).

Routine name (ØCBF2H - ØCBF9H):
Contains the name of the routine loaded in the user BIOS area. The routine name may be any string of ASCII codes.

Size (ØCBFAH):
Is a binary number indicating the size in 256-byte units of the routine loaded in the user BIOS area.

Overwrite flag (ØCBFBH):
Indicates whether the current routine can be overwritten by a new routine.
= ØØH: Overwrite is disabled.
= Nonzero: Overwrite is enabled.

Release address (ØCBFCH - ØCBFDH):
Contains the address of the routine which releases the area for the existing routine to load a new routine. The release routine is enabled only when the overwrite flag is nonzero. The release address must be within the user BIOS area. The release processing routine must end with a RET instruction.

ØCBFEH:
Not used. Always set to ØØH.

Checksum (ØCBFFH):
Contains the checksum byte which is calculated by successively subtracting the value of the 15 header bytes from ØØH.

(b) Overwrite flag

The overwrite flag must be set to ØØH for routines which, once loaded, must reside in the the user BIOS area. Such routines can be deleted from the the user BIOS area only by the program that called them or by system initialize processing.

The overwrite flag must be set to a nonzero value for routines which --
Modify the the user BIOS area when loaded,
Can restore the the user BIOS area into the original state, and
Can be overwritten by a new routine.

(c) Release processing

The user BIOS routine in the the user BIOS area must save the contents of the system area into the user BIOS area before making any attempt to alter the system area. The release processing routine restores the contents of the saved user BIOS area into the system area to establish the original system state when the user BIOS routine was loaded, and initializes the header to zeros. The release processing routine clears the header to zeros even if there is no need to restore the system area.

The release processing routine must fit in the 256 byte area at the end of the user BIOS area (ØCBØØH through ØCBFFH). It must end with a RET instruction.

(3) Loading a user BIOS routine

The application program which is to load a user BIOS routine into the user BIOS area must make checks shown in the figure below to verify whether the the user BIOS area is available for the user BIOS routine.

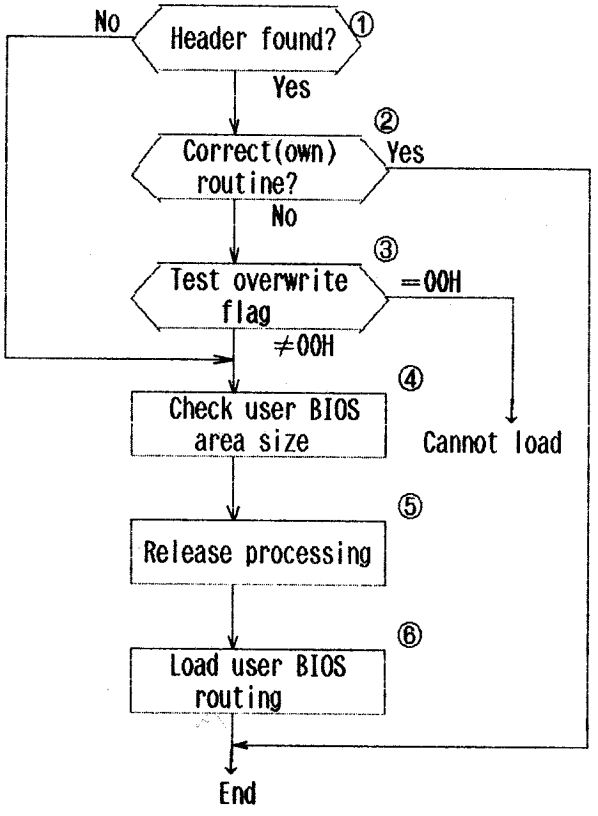


Fig. 4.1.3 Loading a User BIOS Routine