Chapter 19 Application Notes

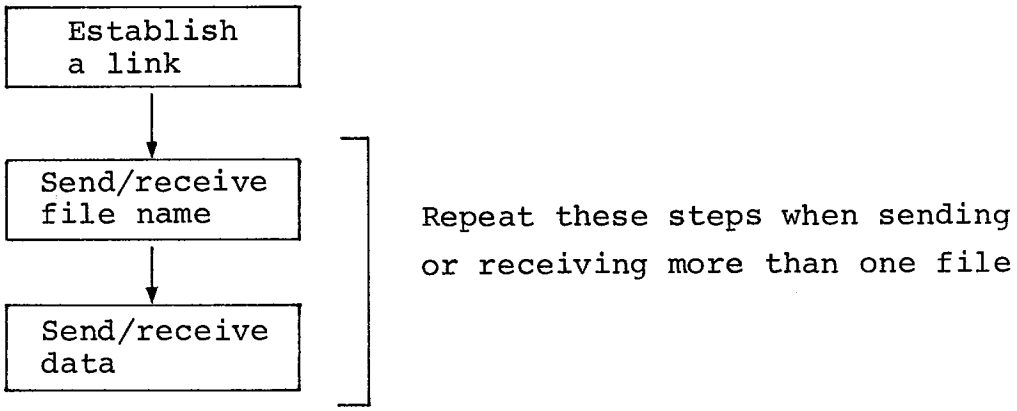
This chapter gives various information which will aid the user in developing application programs.

Interface

19.12 Procedure for Sending and Detecting the RS-232C Break Signal

19.1 FILINK Communications Protocol

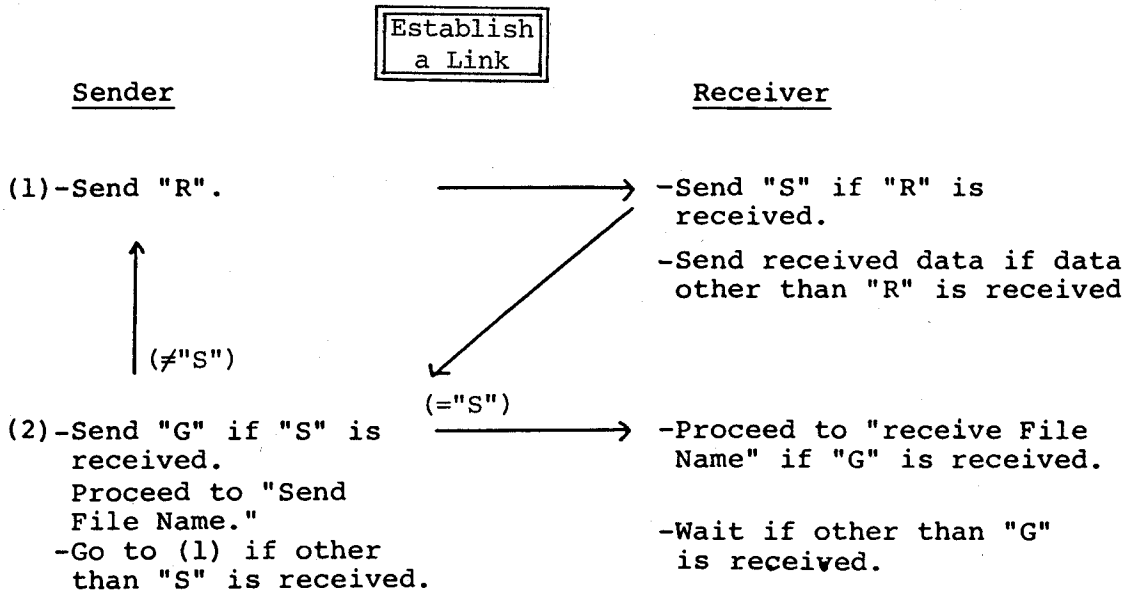
FILINK transmits and receives files via the RS-232C interface using the protocol illustrated below.

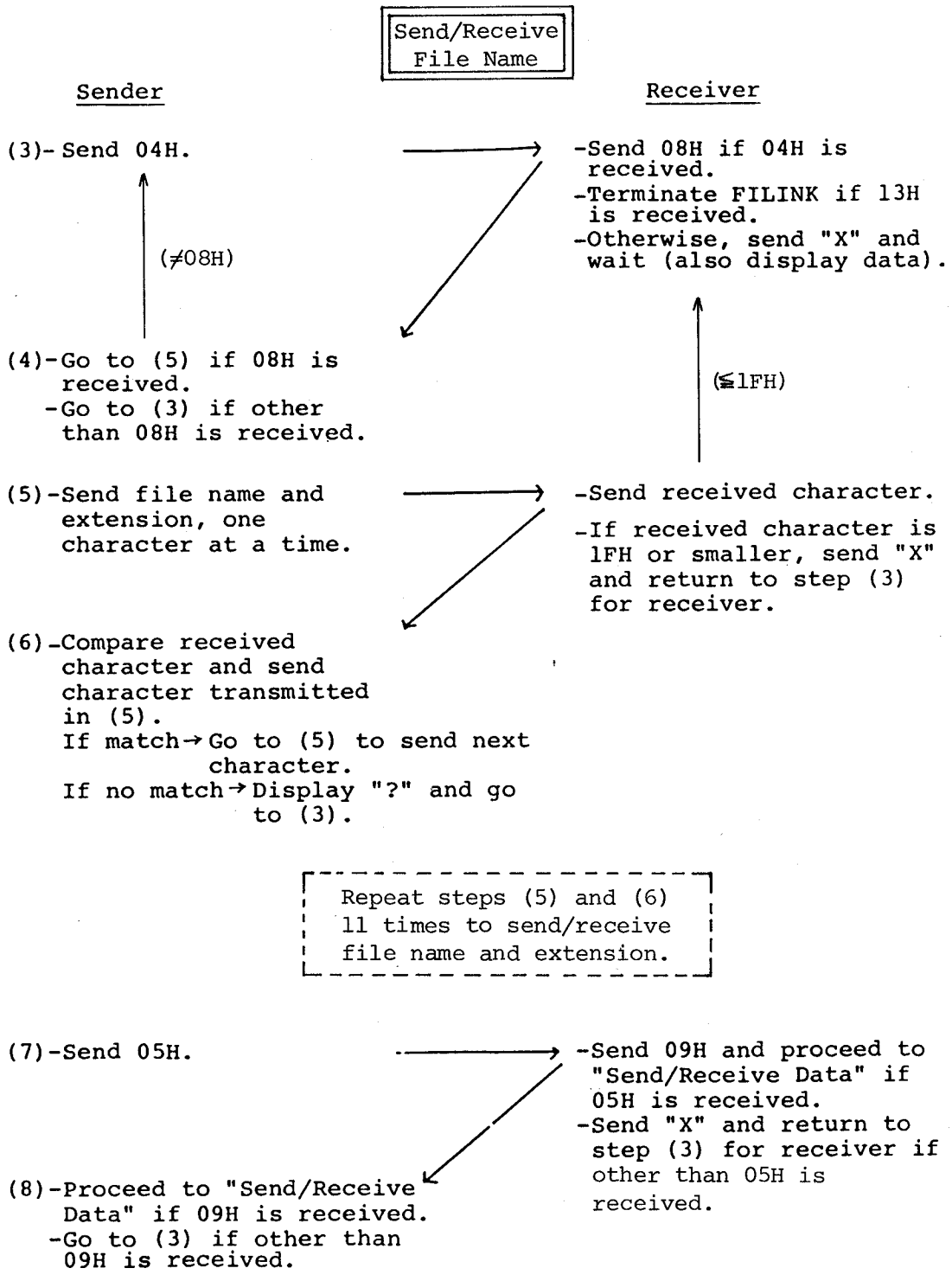


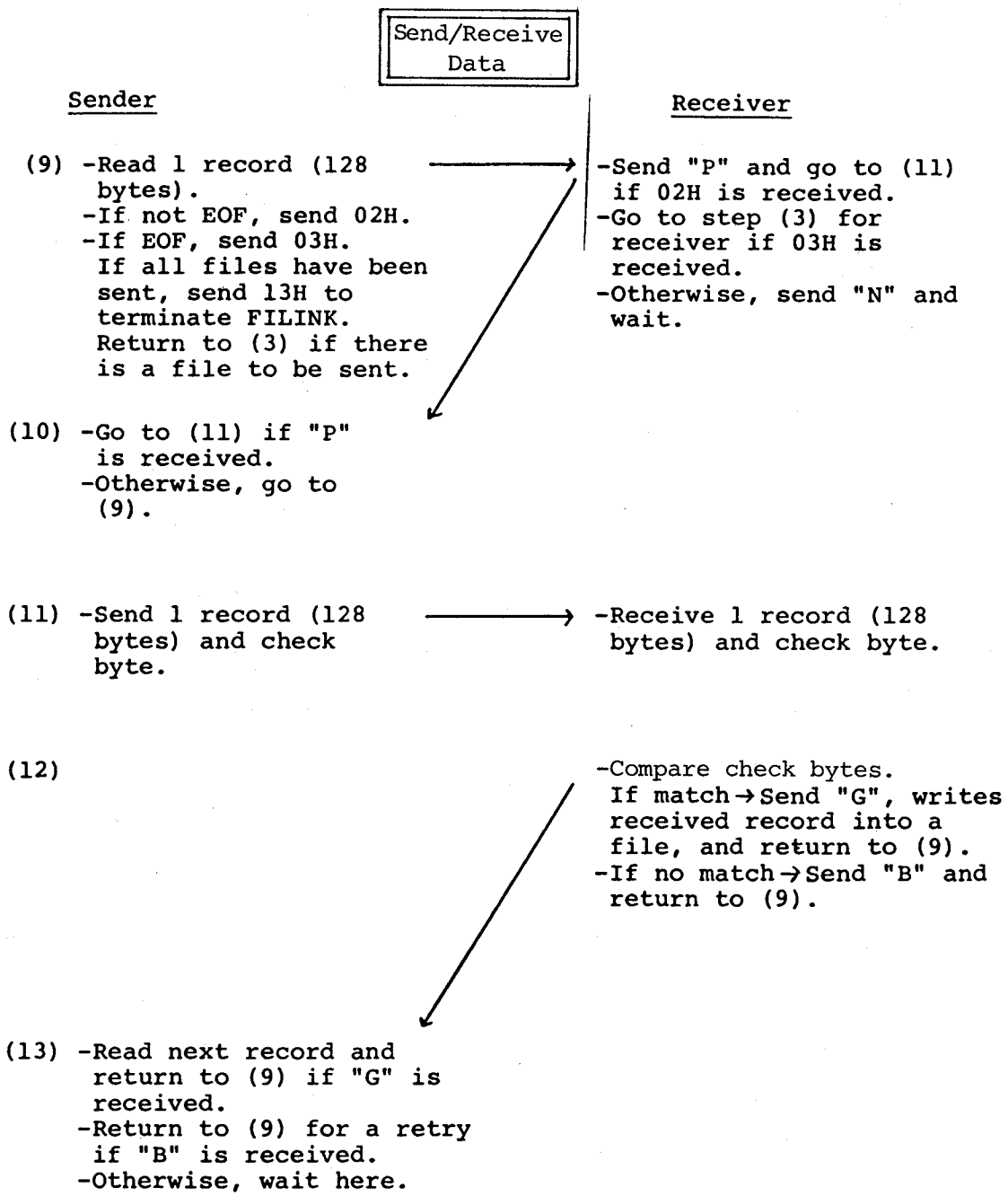
The above communications protocol is supported by the programs/commands listed below.

Machine Type	Program/Command
MAPLE	∘ FILINK.COM ∘ WS.COM T and C commands ∘ SC.COM/JSC.COM/Join, Send, and Receive commands
PINE (HC-40, PX-4 / HX-40)	∘ FILINK.COM
QC - 1Ø QX - 1Ø	∘ FILINK.COM
IBM-PC	∘ FILINK.COM (EXE)

The FILINK communications protocol is detailed below.







19.2 Procedure for Calling BDOS and BIOS Directly from BASIC

To call a BDOS or BIOS function directly from BASIC, prepare a machine-language program for interfacing to BDOS or BIOS and run it using the BASIC CALL statement.

19.2.1 Calling BDOS

(Machine-language program)

C5, 4E, E3, 5E, 23, 56, CD, Ø5,
ØØ, C1, Ø2, Ø3, AF, Ø2, C9 (15 bytes)

```
PUSH   BC
LD     C,(HL)
EX     DE,HL
LD     E,(HL)
INC    HL
LD     D,(HL)
CALL   5
POP    BC
LD     (BC),A
INC    BC
XOR    A
LD     (BC),A
RET
```

(BASIC)

```
CALL BDOS(C%, DE%, A%)
```

BDOS specifies the address of the machine-language routine.

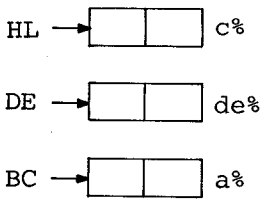
C% specifies the BDOS function number (C% = 255 for dirinit).

DE% specifies the RCB address (optional).

A% holds the return code returned by BDOS (Ø = normal termination; nonzero = error).

(Example)

```
100 CLEAR ,&HA000: BDOS=&HA000
110 FOR I=0 TO 14: READ X: POKE BDOS+I,X
: NEXT
120 C%=255: CALL BDOS(C%,DE%,A%)
130 IF A%<>0 THEN PRINT"Error"
140 DATA &Hc5,&H4e,&Heb,&H5e,&H23,&H56,&
Hcd,&H05
150 DATA &H00,&Hc1,&H02,&H03,&Haf,&H02,&
Hc9
```



19.2.2 Calling BIOS

Change Ø5H, ØØH (BDOS entry address) of the byte sequence CDH, Ø5H, ØØH in the above machine-language program to the required BIOS entry address. To locate the BIOS entry address, read the WBOOT entry address stored in RAM addresses 1 and 2 with the PEEK statement and add to it the offset value of the required BIOS function (see Chapter 4 for details about the procedure for calculating a BIOS entry address).

19.3 Procedure for Determining the Type and Size of RAM Disk

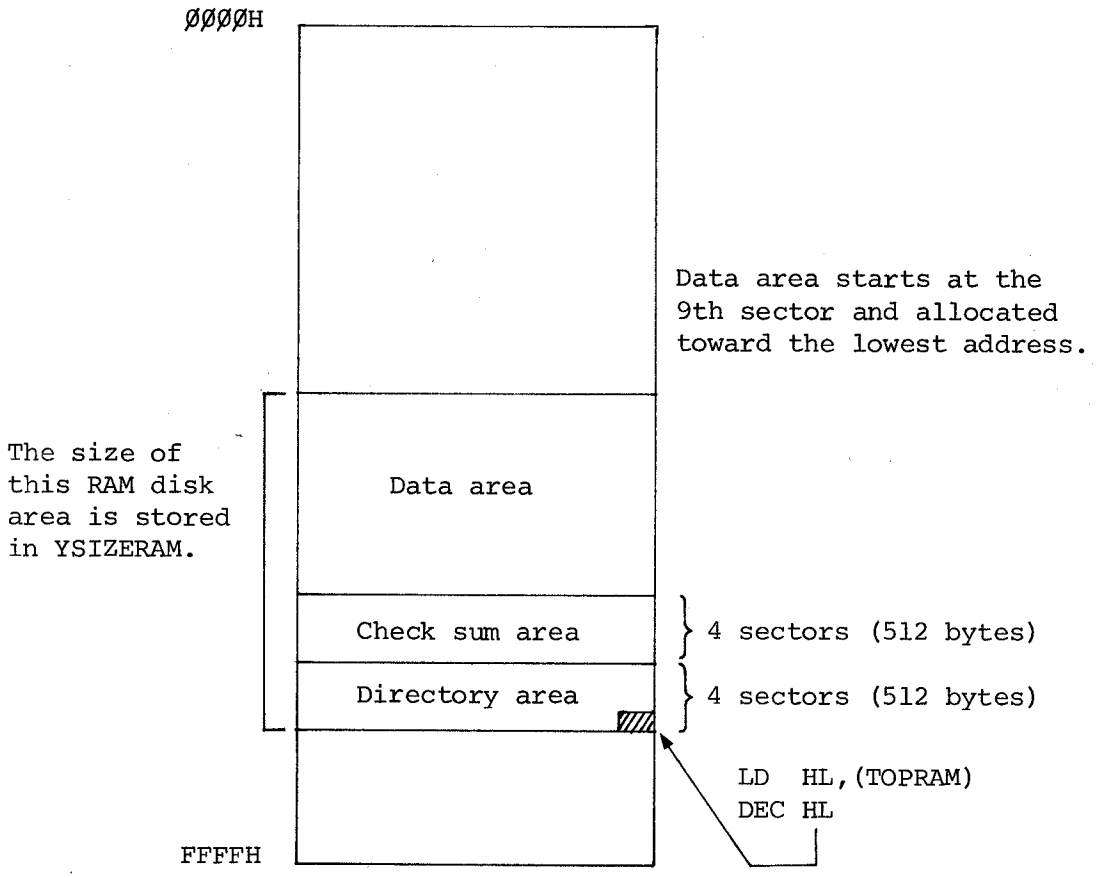
The MAPLE has the following three types of RAM disks:

- Main memory RAM disk
- Intelligent type RAM disk (in expansion unit)
- Unintelligent RAM disk (in expansion unit)

RAM work area YSIZERAM (at location ØF6A8H for overseas version and ØF42BH for Japanese-language version) contains the size in binary form of the active RAM disk in 1K-byte increments. The value of this area also indicates the type of RAM disk.

YSIZERAM contents	RAM disk type
6Ø	6ØK-byte intelligent RAM disk unit
12Ø	12ØK-byte intelligent RAM disk unit
64	64K-byte unintelligent RAM disk unit
128	128K-byte unintelligent RAM disk unit
Other value	Indicates the size of the main memory RAM disk.

See Chapter 16 for the formats of the intelligent and unintelligent RAM disk units. The rest of this section describes the format of the main memory RAM disk.



The lowest address of the RAM disk (marked ///) is calculated as the value of the 2-byte field labeled TOP RAM minus 1. The location of TOPRAM is 0F076H for overseas versions and ØED82H for Japanese-language version.

19.4 CG Fonts

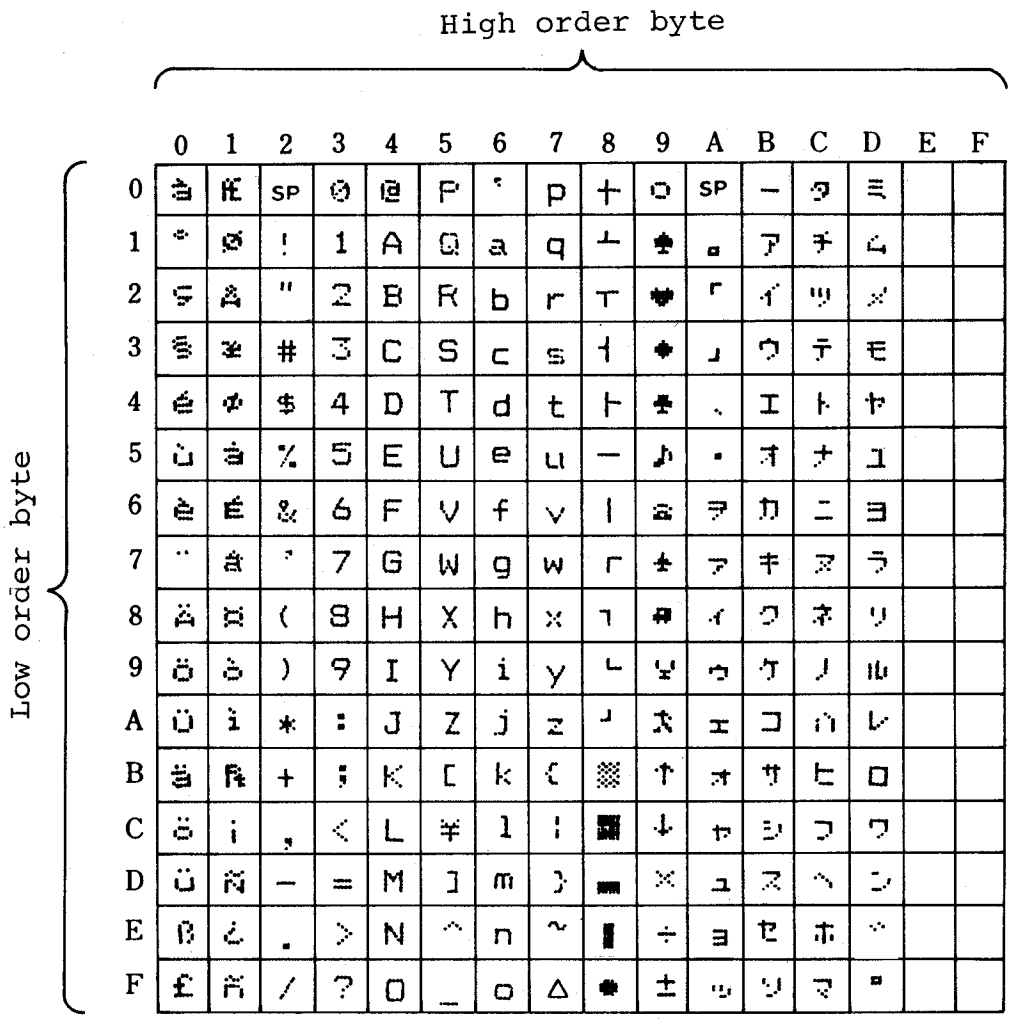
The fonts used for the MAPLE are contained in SED1320 on the LCD controller (except those for Japanese-language kanji, hiragana, and katakana characters). During an OS operation, a character code issued through the BIOS CONOUT routine is converted into the corresponding code in the CG. The converted code is sent to the SED1320 via the slave CPU and used to select the corresponding font, which is then displayed on the LCD screen. A table of CG fonts are given on the next page.

The fonts for the gaiji characters corresponding to character codes 0E0H through 0FFH are stored in the slave CPU. These character codes are converted into fonts within the slave CPU and sent to the SED1320 for display.

The application program can display these CG fonts directly by sending the ESC, "%", (CG code) sequence with the CONOUT BIOS function (see Chapter 6).

To get a CG font pattern from an application program, use the slave CPU command code 1BH (see Chapter 13).

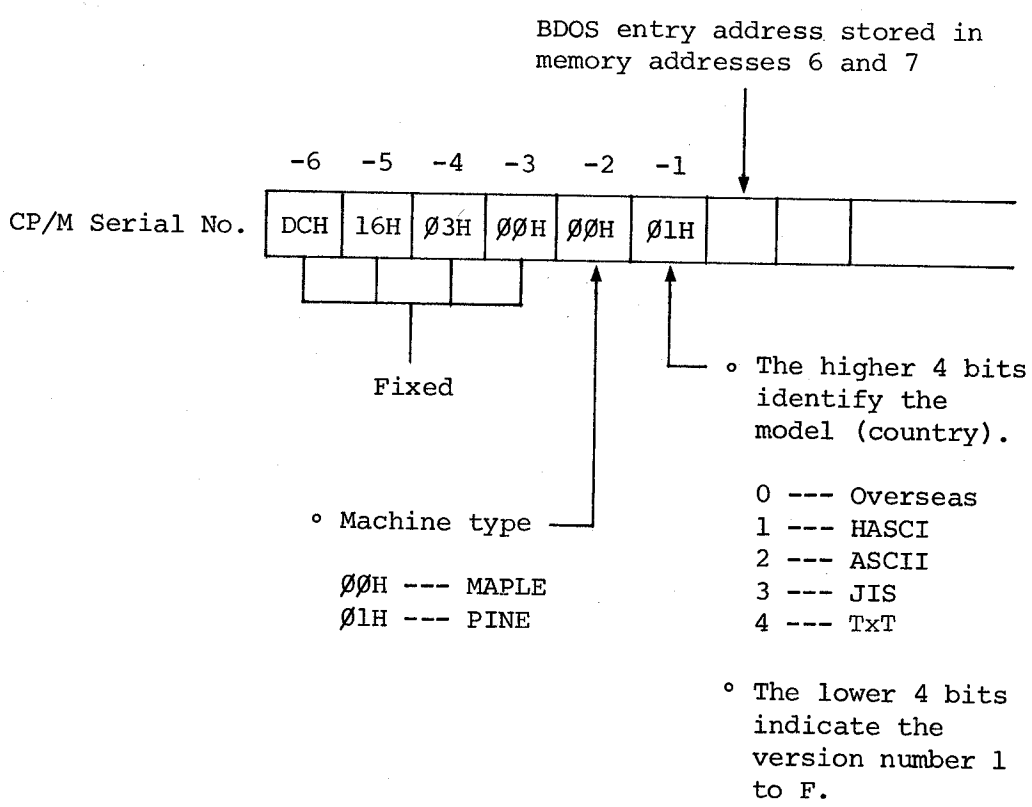
CG ROM Fonts



19.5 Procedure for Identifying the OS Version from an Application Program

MAPLE/PINE (HC-40, PX-4, HX-40) application programs can prevent themselves from causing fatal errors or hanging up when executed under an unintended operating system by checking the version of the running operating system at the beginning of their execution.

To check the OS version, refer to the 6-byte CP/M serial number filed in BDOS that contains the machine type, model (country) name, and version number.



How to refer to the CP/M serial number field

(1) Application programs not dedicated to the MAPLE/PINE need not check the version number. Programs that fall within this category include:

- Programs that reference no system area.
- Programs whose screen handling routines are not intended for MAPLE/PINE.
- Other programs

(2) Since the MAPLE has different memory maps for its overseas and domestic versions, application programs for the MAPLE are also divided into two groups. These programs must use different procedures for identifying the OS version.

1) Application programs for overseas versions

DCH, 16H, Ø3H, ØØH, ØØH, ØXH (X = 1-F)

DCH, 16H, Ø3H, ØØH, ØØH, 1XH (X = 1-F)

DCH, 16H, Ø3H, ØØH, ØØH, 2XH (X = 1-F)

The application programs can run under operating systems that contain one of the OS version field values given above. They should signal an error condition for other values.

2) Application programs for Japanese-language versions:

DCH, 16H, Ø3H, ØØH, ØØH, 3XH --> JIS OS

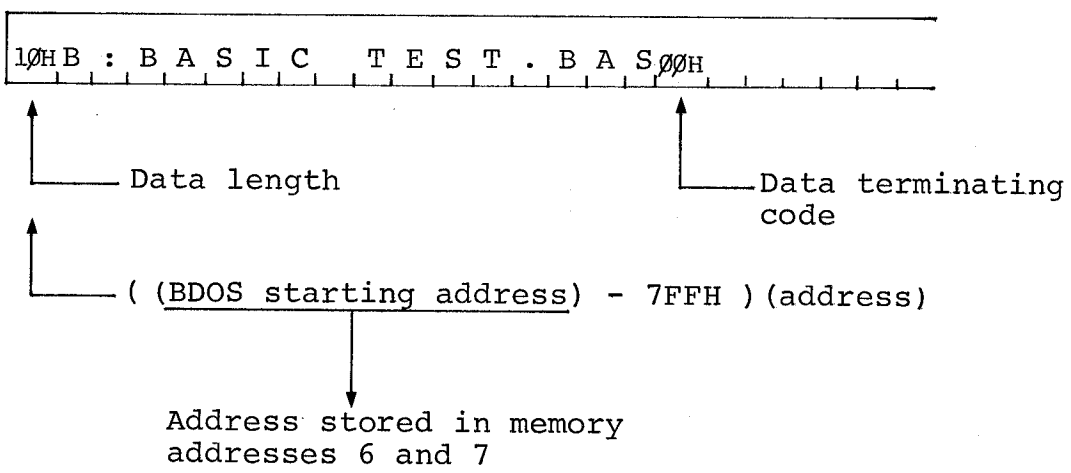
DCH, 16H, Ø3H, ØØH, ØØH, 4XH --> TXT OS (X = 1-F)

The application programs can run under operating systems that contain one of the OS version field values given above. They should signal an error condition for other values.

19.6 Procedure for Checking the Data Received by CCP from an Application Program

When CCP starts an application program, it loads the command line parameters in CP/M system areas at 05CH and 80H. CCP, however, deletes the name of the application program and the drive number of the disk drive from which the application program is started. Consequently, the application program cannot determine from which drive it was loaded into memory. The application program can, however, refer to the parameter data that the user specified via CCP by examining the CCP work area.

Example: A>B:BASIC TEST.BAS/



19.7 Procedure for Detecting the Depression of the CTRL/STOP keys

When the STOP key is pressed, the OS usually clears the keyboard buffer and places a Ø3H (STOP key code) in the keyboard buffer. When the STOP key is pressed simultaneously with the CTRL key, however, the OS terminates the currently executing I/O operation as well as it clears the keyboard buffer and places a Ø3H code in the keyboard buffer. This allows the user to gain control when the program is placed in a loop waiting, for example, for receive data in the RS-232C receive routine. In this case, however, the application program must know that the CTRL and STOP keys have been pressed to terminate itself. This can be accomplished by examining the flag field described below.

CSTOPFLG (ØF1ØBH for overseas version; ØEE25H for Japanese-language version)

= ØØH: The CTRL and STOP keys are not pressed.

≠ ØØH: The CTRL and STOP keys have been pressed.

After the CTRL and STOP keys are pressed simultaneously, the keyboard buffer contains only Ø3H code and CSTOFLG

is set to a nonzero value. CSTOPFLG is cleared by the OS when the keyboard buffer is emptied.

19.8 Procedure for Assigning Printer Output to RS-232C or Serial Interface

Output to the printer can be directed to either RS-232C or serial port by changing the contents of the I/O byte (at memory address 3).

I/O byte, bits 7 and 6 ... 1, Ø --> RS-232C

I/O byte, bits 7 and 6 ... Ø, Ø --> Serial

Output data will be placed on the specified port when the following list-related commands or routines are executed after the I/O byte is altered as shown above:

(1) BIOS level

LIST routine (WBOOT + ØCH)

(2) BDOS level

Function 5 (list)

(3) BASIC level

LPRINT command

19.9 Procedure for Restoring the Screen into the State Set up by CONFIG

The screen remains in the state set up by an application program when WBOOT is performed at the end of the application program whereas the screen is restored into the state that is defined by CONFIG when BOOT is invoked by pressing the RESET switch. Any application program which reconfigures the screen should restore the screen into the original configuration at the end of its execution.

The OS stores in RAM memory the ESC sequence data related to the screen state defined by CONFIG. The application program can restore the screen configuration into the original state by sending this data to the screen using CONOUT.

The ESC sequence data is stored in the two areas given below. The ESC sequence data in each area is terminated by ØFFH code, so the application program need only send the data bytes until an ØFFH is encountered.

CONSCRN1 (ØFØDDH for overseas version; ØEDBDH for

Japanese-language version): Contains the current screen mode and virtual screen identification.

CONSCRN2 (ØFØF1H for overseas version; ØEDD1H for Japanese-language version): Contains data related to the cursor.

[Sample program]

```
        DI
        LD(   HL,CONSCRN1     ;screen mode, select screen
        CALL  LCDOUT          ;scroll mode, function key display
        LD    HL,conscrn2     ;cursor kind and on/off
        CALL  LCDOUT
        EI
        .
        .
LCDOUT:
        LD    A,(HL)
        INC   HL
        INC   A               ;end of string?
        RET   Z               ;then return
        DEC   A
        LD    C,A
        PUSH  HL
        CALL  CONOUT          ;display character
        POP   HL
        JR    LCDOUT
```

19.10 Procedure for Configuring the System Environment from an Application Program

19.10.1 Auto Power Off (common to both overseas and Japanese-language versions)

See 7.6 "Auto Power Off Feature."

19.10.2 CP/M Function Key (common to both overseas and Japanese-language versions)

See paragraph (6) "Programmable Function keys" in 5.6 "Special Keys."

19.10.3 Cursor & Function Key Display (common to both overseas and Japanese-language versions)

(1) Use the CONOUT ESC sequence function to set up the cursor and function key display modes.

- Cursor tracking: ESC, 95H
- Cursor display: ESC, "2" or ESC, "3"
- Cursor type: ESC, D6H
- Function key display: ESC, D3H

(2) The current settings can be located by checking the following work areas:

- Cursor tracking:

  LSMODE (ØF2D4H for overseas version; ØFØØ4H for Japanese-language version)

  = ØØH: Tracking mode

  ≠ ØØH: Nontracking mode

- Cursor display:

  LUSSTS (ØF2D7H for overseas version; ØFØØ7H for Japanese-language version)

  Bit Ø = Ø: Off

  Bit Ø = 1: On

- Cursor type:

  LUSSTS (ØF2D7H for overseas version; ØFØØ7H for Japanese-language version)

  Bit 1 = Ø: Nonblink

  Bit 1 = 1: Blink

  Bit 2 = Ø: Underline

  Bit 2 = 1: Block

- Function key display

  LUFKDSP (ØF2DØH for overseas version; ØFØØ2H for Japanese-language version)

  = 7: Display

  = 8: No display

The screen may not actually change its configuration when the above work areas are simply changed. This is attributed to reasons associated with the interactions

between the screen and other resources. Nevertheless, the screen configuration must be set up using the CONOUT ESC sequence functions.

19.10.4 Date and Time (common to both overseas and Japanese-language versions)

Use the TIMDAT BIOS function (WBOOT + 4BH) (see Chapter 4).

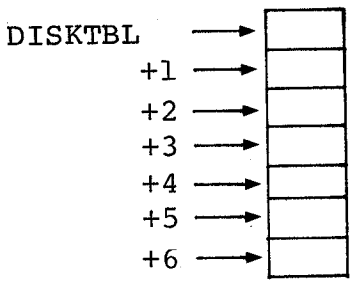
19.10.5 Disk Drives (common to both overseas and Japanese-language versions)

Refer to or change the table in RAM associating the physical and logical drives.

Physical drive codes

00H:	RAM disk
01H:	ROM capsule 1
02H:	ROM capsule 2
03H:	FDD1
04H:	FDD2
05H:	FDD3
06H:	FDD4

The logical drives are indicated in the table shown below.



Drive A (defaults to ØØH)
Drive B (defaults to Ø1H)
Drive C (defaults to Ø2H)
Drive D (defaults to Ø3H)
Drive E (defaults to Ø4H)
Drive F (defaults to Ø5H)
Drive G (defaults to Ø6H)

DISKTBL: (ØF1D2H for overseas version; ØEEEBH for Japanese-language version)

Change the contents of the DISKTBL to redefine the association between the physical and logical drives (e.g., reassigning an FDD to drive A).

Notes:

1. Do not specify values other than ØØH through Ø6H as physical drive codes.
2. Do not specify a physical drive code in duplicate.
3. Specify logical drives between A: and G:
4. The redefined specifications remain valid until the RESET switch is pressed.

19.1Ø.6 Printer (common to both overseas and Japanese-language versions)

Use the I/O byte (see 19.8).

19.10.7 RS-232C (RS-232C (1) for Japanese-language version)

The 5-byte field starting at RDSDAT (0F00FH for overseas version and 0ED0FH for Japanese-language version) contains the parameter values from bit rate through special parameter that are set up by the BIOS RSIOX OPEN function. See Chapter 4.

All conditions pertaining to the RS-232C interface (for BIOS RSOPEN, RSOUT, LIST, PUNCH, etc.) are controlled by the data in this field except when using the RS-232C interface after opening it with the RSIOX OPEN function.

19.10.8 Screen mode (common to both overseas and Japanese-language versions)

(1) Send the ESC 0D0H sequence using the BIOS CONOUT function (WBOOT + 9) to setup the screen mode. See Chapter 6.

(2) The current screen modes can be identified by checking the following work areas:

- Screen mode

  LSCMODE (0F2C9H for overseas version; 0EFFBH for

Japanese-language version)

= ØØH: Screen mode Ø

= Ø1H: Screen mode 1

= Ø2H: Screen mode 2

= Ø3H: Screen mode 3

- Virtual screen 1

  LV1SCT + 4 (ØF71FH for overseas version; ØF56DH for Japanese-language version): Contains the number of virtual screen 1 columns in binary form.

  LV1SCT + 5 (ØF720H for overseas version; ØF56EH for Japanese-language version): Contains the number of virtual screen 1 rows in binary form.

- Virtual screen 2

  LV2SCT + 4 (ØF725H for overseas version; ØF573H for Japanese-language version): Contains the number of virtual screen 2 columns in binary form.

  LV2SCT + 5 (ØF726H for overseas version; ØF574H for Japanese-language version): Contains the number of virtual screen 2 rows in binary form.

- Selected screen:

  LDSPVS (ØF2CAH for overseas version; ØEFFCH for Japanese-language version)

  = ØØH: Displays virtual screen 1.

= Ø1H: Displays virtual screen 2.

- Separation character

  LBOUNDP (ØF2D9H for overseas version; ØFØØ9H for Japanese-language version): Contains the character code proper.

19.10.9 Serial (common to both overseas and Japanese-language versions)

DHSDAT (ØF014H for overseas version; ØED14H for Japanese-language version)

= Ø1H: 4,8ØØ bps

= Ø2H: 6ØØ bps

= Ø3H: 15Ø bps

19.10.1Ø Country (overseas version only)

YLDFLTC (ØF6A1H for overseas version)

= ØFH: ASCII

= ØEH: France

= ØDH: Germany

= ØCH: England

= ØBH: Denmark

= ØAH: Sweden

= Ø9H: Italy

= Ø8H: Spain

= Ø6H: Norway

When this work area is altered, only the display modes are changed and no keyboard mode is changed. After a

WBOOT, the character fonts of the selected country are enabled for display.

19.11 XON/XOFF Control for the Currently Open RS-232C Interface

The following work area is referenced to determine how XON/XOFF control is exercised for the currently open RS-232C interface:

SKXFLG (ØF6C4H for overseas version; ØF447H for Japanese-language version)

Bit 4 = Ø: XON/XOFF control disabled

Bit 4 = 1: XON/XOFF control enabled

Bit 6 = Ø: XON has been sent.

Bit 6 = 1: XOFF has been sent.

Bit 7 = Ø: XON has been received.

Bit 7 = 1: XOFF has been received.

19.12 Procedure for Sending and Detecting the RS-232C Break Signal

19.12.1 Sending the RS-232C Break Signal

Run the following program:

```
LD   A, 3FH
OUT  (ØDH), A

Call software timer to provide a delay
required for sending the code

LD   A, 37H
OUT  (ØDH), A
```

19.12.2 Detecting the RS-232C Break Signal

Use the following program:

```
IN   A, (ØDH)
```

Areg. bit 6 = Ø: No Break signal

Areg. bit 6 = 1: Break signal detected

See an 8251 manual for further information.