

CHAPTER 7 MISCELLANEOUS TOPICS CONTENTS

7.1	Versions	II-679
7.1.1	General	II-679
7.1.2	Version Check	II-679
7.1.3	Version Differences	II-680
7.2	DIP Switches	II-682
7.2.1	General	II-682
7.2.2	DIP Switch Bit Definitions	II-683
7.2.3	System Processing	II-683
7.3	Guide for Programming ROMs	II-686

CHAPTER 7 MISCELLANEOUS TOPICS

7.1 Versions

7.1.1 General

PINE OS is available in the following three versions:

Kana V1.0	} (Japan only)
Kana V2.0	
Export version 1.0	

For application programs which have to run in different modes depending on the OS version, PINE OS provides a facility which allows them to refer to the version of the OS under which they are currently running.

7.1.2 Version Check

The version can be examined using one of the following methods:

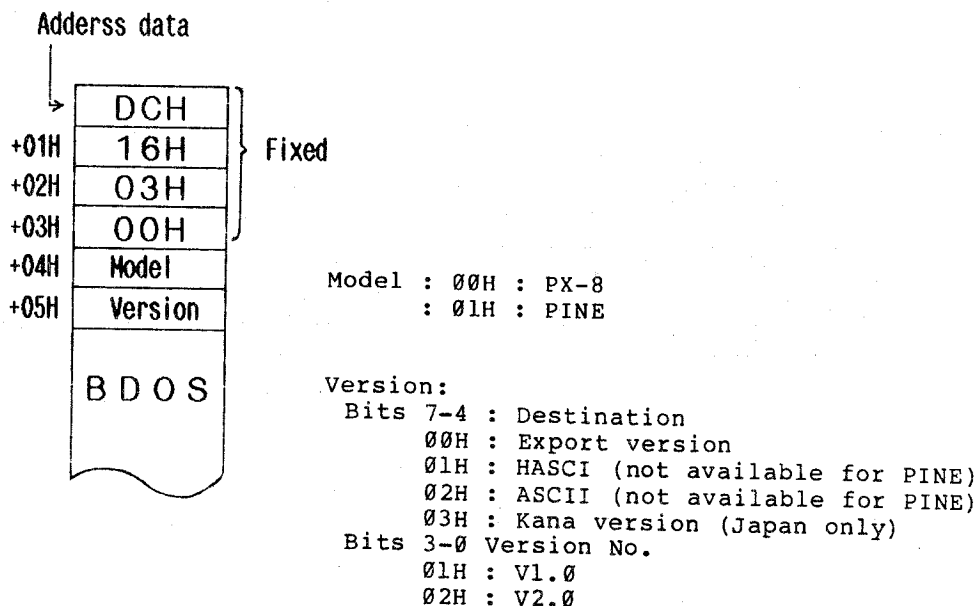
1. Checking the CP/M serial number.
2. Calling ROMID on OS ROM.

This subsection shows how to test the version using the above methods.

7.1.2.1 Checking the CP/M serial number

The model type and OS version number, and other information are stored in 6-byte CP/M BDOS fields, labeled Serial No., which are located at the beginning of RBDOS1 and RBDOS2.

To refer to Serial No. in RBDOS1, check the 6 bytes before the BDOS entry address which is stored in addresses 0006H and 0007H. To refer to Serial No. in RBDOS2, check the 6 bytes starting at 0EA00H. These two fields contain the same information.



7.1.2.2 ROMID

ROMID identifies the OS ROM installed in the PINE. It is located in an 8-byte area on OS ROM at 7FF8H. ROMID can be referenced using the LDIRX BIOS function.

The format of the ROMID field is shown below. Data in this field is all in ASCII code.

7FF8H	
9	Model code
A	
B	Country
C	Program ver.
D	Data ver.
E	Area ver.
F	Reserved

Model code : Set to 'PIN'.

Country — : 'J' : Kana version(Japan only)

'E' : Export version

'U' : USA version

(not available for PINE)

Program ver. : Identifies the OS version.

'1' : Ver.1.0

'2' : Ver.2.0

Data ver. : Identifies the version of the initial system area data.

'1' : Ver.1.0

'2' : Ver.2.0

Area ver. : Identifies the version of the added system area.

'1' : Ver.1.0

'2' : Ver.2.0

'3' : Ver.3.0

7.1.3 Version Differences

The table on the next page summarizes the functional differences among OS Kana version V1.0, and V2.0, and Export version 1.0. The table below lists the OS versions and their identifications placed on the OS ROM (mask ROM).

OS version	Version ID on ROM
Kana V1.0 (Japan only)	EPSON POSK1
Kana V2.0 (Japan only)	EPSON POSK2
Export Version 1.0	EPSON POS01

Difference	Kana V1.0	Export version 1.0	Kana V2.0	Remarks
*CRS state after cartridge installation.	Depends on the cartridge type.	Programmable by rewriting CRSTPTN(OEFC1H).	Same as in Export version 1.0.	OS cartridge check routine(CHKMOD) is made available by rewriting CRSTPTN when adding a cartridge.
Supported cartridge I/F mode	HS mode DB mode	HS mode DB mode IO mode	HS mode DB mode IO mode	IO mode is available for cartridges with device codes of OEH and OFH.
Emergency IF write at power-off time	No	Yes	Yes	
Provisions for modifying hook or adding hook entries	Control can be transferred to any location in the RAM area from the hook.	Control can be transferred to locations at 8000H and higher from the hook. A hook entry to BIOS is added to Kana V1.0.	Control can be transferred to locations at 8000H and higher from the hook. A hook entry to ART-interrupt processing routine is added to Export version 1.0.	Control can be transferred to any locations in the RAM area through hook entries for ICF, OVf, and EXT interrupts.
System utility jump table.	A jump table is available for 33 utility routines.	An entry to cartridge check routine(CHKMOD) is added to the jump table for kana V1.0.	Same as in Export version 1.0.	
Power-off or alarm processing during RSIOX processing.	RSIOX is terminated before power-off or alarm processing is performed.	Power-off or alarm processing is performed during RSIOX processing(no data is lost).	Same as in Export version 1.0.	

Difference	Kana V1.0	Export version 1.0	Kana V2.0	Remarks
Alarm screen display time.	Fixed at 50 seconds.	Programmable. Defaults to 50 seconds. Display time is loaded in ALRMTIME (0F308H).	Same as in Export version 1.0.	
Alarm or wake processing after power failure.	Not inhibited.	Inhibited. Original state is restored by powering on the machine after power is restored.	Controlled by flag BIRVALM (0EFC3H). =00H : Inhibited. =Nonzero : Not inhibited.	In Export version 1.0 and kana V2.0, alarm or wake processing routine is exited if power is shut down by a power failure when it has been inhibited.
Cartridge printer support.	Not supported.	Supported. Assigned to CRT(01) in the LST : field.	Same as in Export version 1.0.	Supported in cartridge IIS mode.
Initialization of printer version data.	After rewriting I/O byte LST : field. By sending ESC+'R'+n immediately after a warm boot.	After rewriting I/O byte LST : field. After a warm boot. After rewriting character generator. By sending ESC+'R'+n immediately after power-on	Not initialized.	In Export version 1.0, PRTFLG is cleared at power-on time. In Export version 1.0, PRTFLG contains information pertaining to LST : and display character set.
I/O device serial parameter (when RS-232C is used)	Input and output devices have different serial parameters.	Only LST: devices have different serial parameters.	Same as in Export version 1.0.	

Difference	Kana V1.0	Export version 1.0	Kana V2.0	Remarks
Character set specifications for display functions	ASCII, England, and Japan	ASCII, France, German, England, Sweden, Denmark, Italy, Spain and Norway	Japan only	
Pointer to character generator table	Fixed.	Fixed.	Fonts data can be extracted from the user-supplied character generator by rewriting the area containing the starting address of the character generator table to be used. Codes 20H-7FH : RLCGENN(0F95EH) Codes 80H-9FH : RLCGENG(0F360H) Codes A0H-DFH : RLCGENK(0F362H)	

7.2 DIP Switches

7.2.1 General

PINE OS uses DIP switches to define the following:

1. Keyboard type
2. BIOS LIST device interface
3. International character set for the keyboard and display functions

The keyboard type can be identified automatically by the keyboard hardware that is actually installed, irrespective of whether the corresponding DIP switch is in the ON or OFF state.

7.2.2 DIP Switch Bit Definitions

Figure 7.2.1 shows the definition of the DIP switches. The state of bit 8 is determined by hardware depending on the keyboard actually installed, irrespective of the position of the DIP switch.

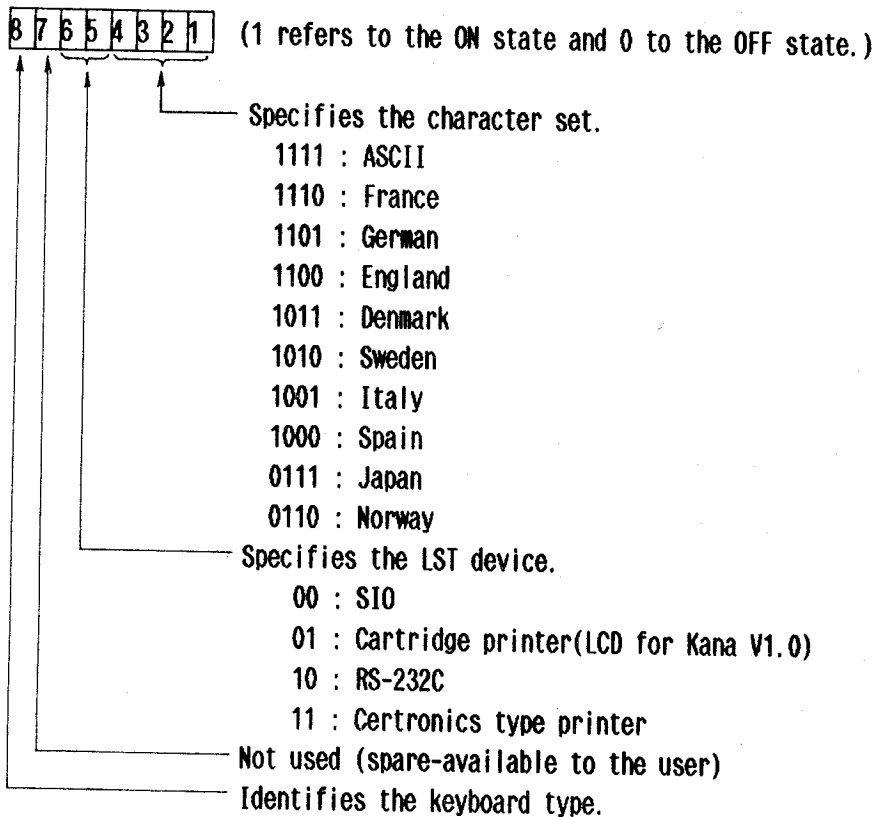


Fig. 7.2.1 DIP Switch Bit Definitions

7.2.3 System Processing

PINE OS reads in the DIP switch status during boot processing initiated by a system initialize or reset and stores it in YKCOUNTRY (0F775H) for subsequent reference during key entry, display, and list processing.

7.2.3.1 Key entry processing

YKCOUNTRY bit 7 is referenced during key entry processing to identify the type of the keyboard installed. When the standard keyboard is used, YKCOUNTRY bits 3-0 are used to indicate the character set employed.

7.2.3.2 Display processing

Flag areas for display processing includes YLDFLTC (0F776H) and YLCOUNTRY (0F777H). Bits 3-0 of YKCOUNTRY are copied into YLDFLTC during boot processing and the contents of YLDFLTC are copied into YLCOUNTRY during a processing routine common to BOOT and WBOOT. The display function references YLCOUNTRY when converting character codes to the correct fonts of the specified character set.

7.2.3.3 List processing

The IO byte at RAM address 0003H is referenced during list processing. When BOOT is started, it copies bits 5 and 4 of YKCOUNTRY into bits 7 and 6 of the IO byte. When the BDOS or BIOS list function is called, it reads in the IO byte into RIOBYTE (0F529H). The actual list processing routine references RIOBYTE to determine the actual list driver routine.

During list processing, YKCOUNTRY is also referenced to output the international character set specification to the list device.

7.2.3.4 Programming notes

- (1) The application program can check the state of the DIP switches using the READSW BIOS function.
- (2) Since PINE OS reads the DIP switch status when a reset is performed, changes in the DIP switch status by the user do not take effect immediately. To make a change in the DIP switch status effective, press the RESET button.
- (3) The character sets supported by the keyboard and display functions differ between OS Export and Kana versions.

Country OS	ASCII	France	Germany	England	Denmark	Sweden	Italy	Spain	Japan	Norway
Kana V1.0	○	×	×	×	×	×	×	×	○	×
Kana V2.0	×	×	×	×	×	×	×	×	×	×
Export V1.0	○	○	○	○	○	○	○	○	×	○

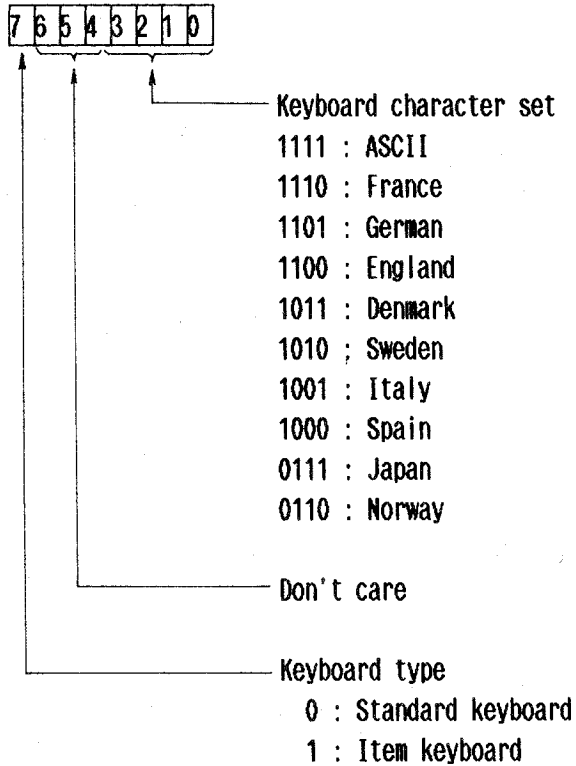
○ : Supported.

× : Not supported.

The system areas related to system processing are listed below.

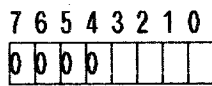
YKCOUNTRY (0F775H) 1 byte

Contains the DIP switch status read during BOOT. This flag is referenced by the key entry processing routine to identify the keyboard installed.



YLDFLTC (0F776H) 1 byte

Contains the default display character set. This flag is loaded with the status of the DIP switches during BOOT processing.



Display character set

- 1111 : ASCII
- 1110 : France
- 1101 : Germany
- 1100 : England
- 1011 : Denmark
- 1010 : Sweden
- 1001 : Italy
- 1000 : Spain
- 0111 : Japan
- 0110 : Norway

YLCOUNTRY (0F777H) 1 byte

Flag indicating the display character set. This area is loaded with a copy of YLDFLTC during BOOT or WBOOT processing. The format of YLDFLTC is identical to that of YDFLTC.

RIOBYTE (0F529H) 1 byte

Loaded with the value of IOBYTE (at 0003H) when BDOS or BIOS is called.

The format of RIOBYTE is identical to that of IOBYTE. See Section 3.9, "I/O Byte" for details.

7.3 Guide for Programming ROMs

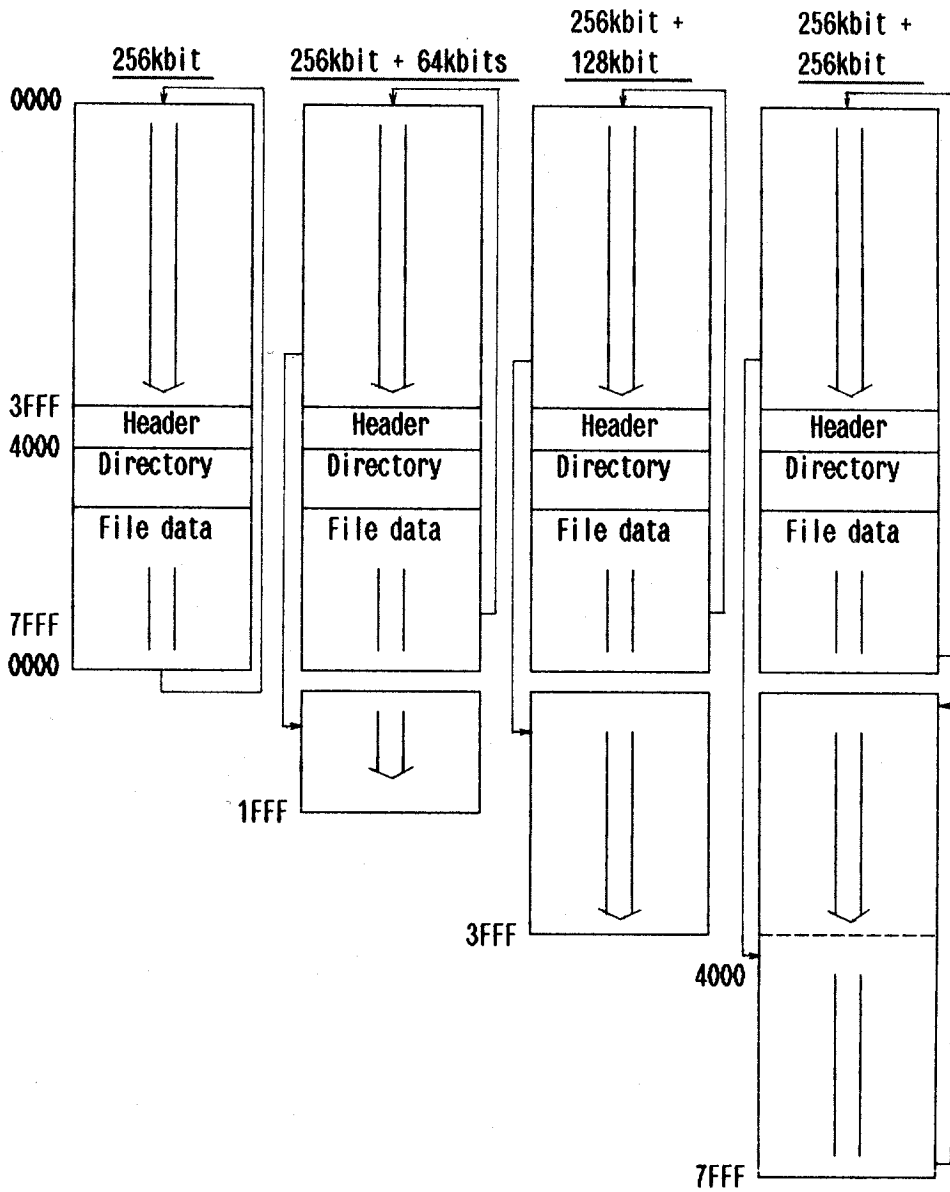
This section gives a guide for programming ROM devices for ROM capsules and cartridges.

7.3.1 Formats

In PINE CP/M, ROMs can be programmed in two formats: M- and P-formats. The M-format is used for MAPLE and PINE ROM capsules and PINE ROM cartridges. Load-and-go programs and data files are created in this format. The P-format is dedicated to PINE ROM capsules and used to construct ROM-based programs.

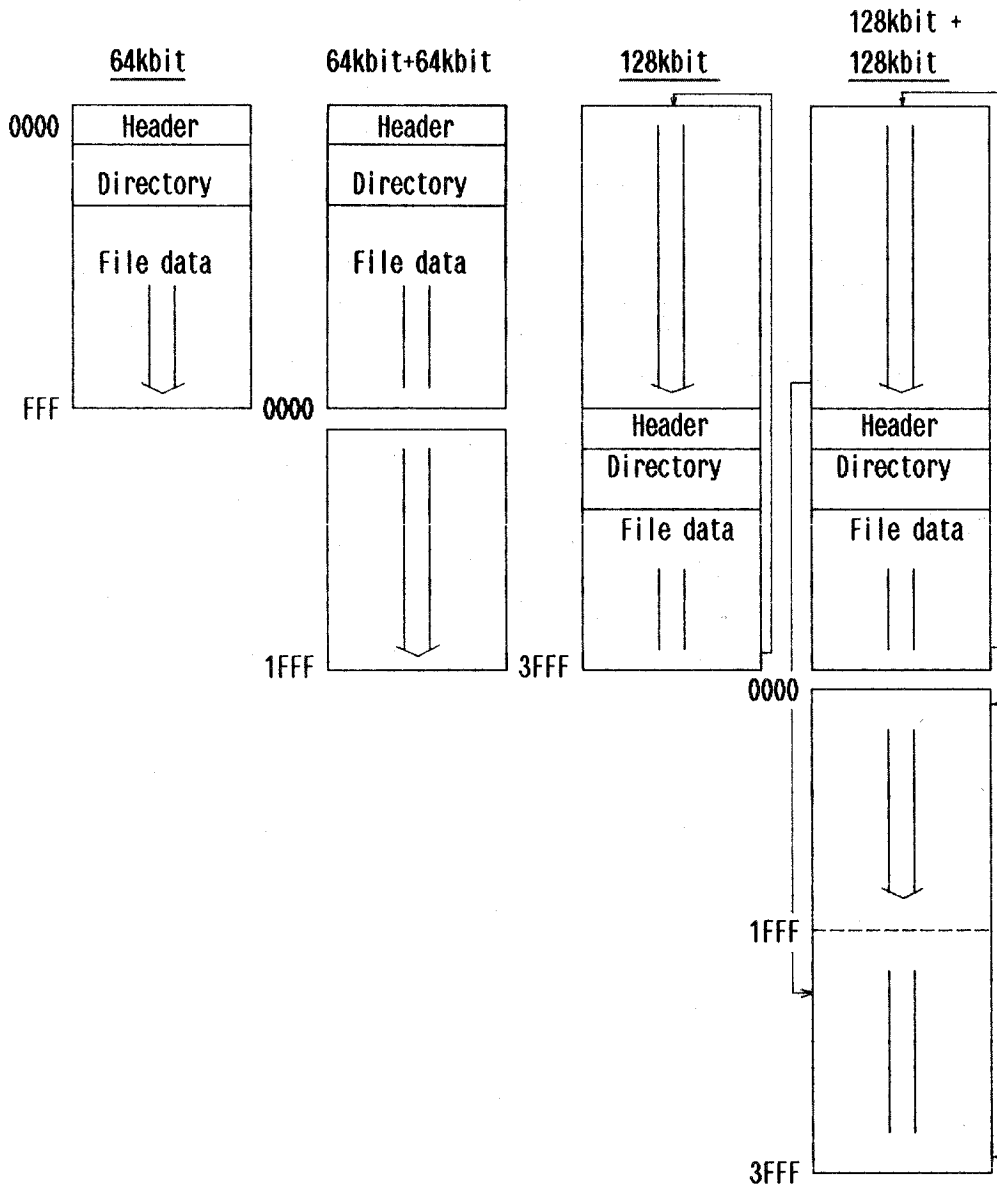
The M- and P-formats differ in the location of the ROM header and directory areas. They also use different addressing schemes.

Supported ROM devices include 64K-, 128K-, and 256K-bit ROMs. Two ROM devices of the same capacity may be combined to form a single ROM cartridge or capsule.

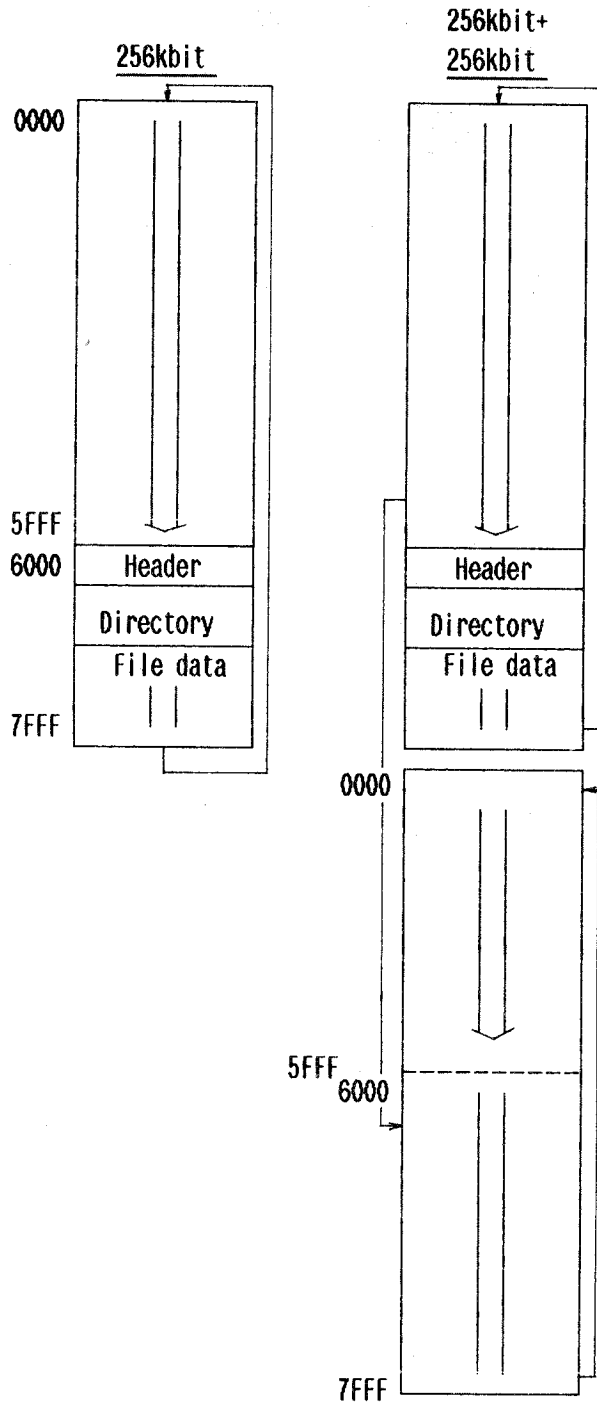


(2) P-format ROM structure

The location of the directory of P-format PROMs differs depending on the type of the PROM device.

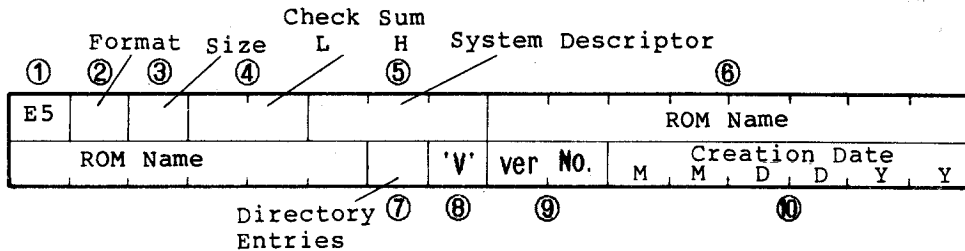


S
A



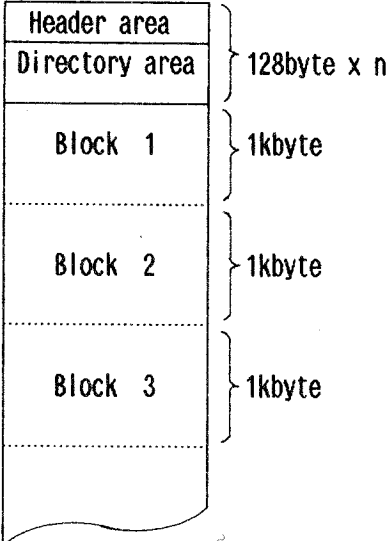
7.3.2 Header Structure

The size of the header for a PROM file is fixed at 32 bytes. The structure of the PROM header is shown below.



No.	Item	Size	Description
1	E5H	1	<ul style="list-style-type: none"> * Set to 0E5H * The system locates the beginning of the ROM by checking this and the subsequent format fields.
2	Format	1	<ul style="list-style-type: none"> * Identifies the format. 37H ...M Format 50H ...P Format
3	Capacity	1	<ul style="list-style-type: none"> * Indicates the capacity of the ROM in 1k bytes (binary format). The MSB is set to 1 for a dual-part PROM. 08H ... 64k bits, single-part PROM 88H ... 64k bits, 2-part PROM 10H ...128k bits, single-part PROM 90H ...2-part PROM with one part consisting of 128k bits. 20H ...256k bits, single-part PROM A0H ...2-part PROM with one part consisting of 256k bits
4	Check Sum	2	<ul style="list-style-type: none"> * 2-byte field containing the checksum of the data calculated by adding together the values of the data bytes from beginning to end of the data area and taking the lowest order two bytes. The lower order byte is placed in the first byte position of the field. * This field is not used by the system.

No.	Item	Size	Description																		
1			<p>* Indicates the validity of the directory entry. 00H ... Valid directory entry E5H ... Invalid directory entry</p> <p>* Invalid directory entries are created to ensure that the directory area ends on a 128-byte boundary. If, for example, five valid directory entries are created, counting the header as a single entry, two odd entries (64 bytes) will result. Such entries are referred to as invalid directory entries (see the example below.)</p> <div data-bbox="568 639 1076 1103" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; border: 1px solid black;">Header</td> <td rowspan="2" style="border: none; padding-left: 10px;">}</td> <td rowspan="2" style="border: none; padding-left: 10px;">128byte</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">Directory entry 1</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">Directory entry 2</td> <td rowspan="4" style="border: none; padding-left: 10px;">}</td> <td rowspan="4" style="border: none; padding-left: 10px;">128byte</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">Directory entry 3</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">Directory entry 4</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">Directory entry 5</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">Invalid directory entry</td> <td rowspan="2" style="border: none; padding-left: 10px;">}</td> <td rowspan="2" style="border: none; padding-left: 10px;">128byte</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">Invalid directory entry</td> </tr> <tr> <td colspan="3" style="text-align: center; border: 1px solid black; border-bottom: none;">File data area</td> <td style="border: none;"></td> </tr> </table> </div>	Header	}	128byte	Directory entry 1	Directory entry 2	}	128byte	Directory entry 3	Directory entry 4	Directory entry 5	Invalid directory entry	}	128byte	Invalid directory entry	File data area			
Header	}	128byte																			
Directory entry 1																					
Directory entry 2	}	128byte																			
Directory entry 3																					
Directory entry 4																					
Directory entry 5																					
Invalid directory entry	}	128byte																			
Invalid directory entry																					
File data area																					
2	File name	8	* Contains a file name of up to 8 ASCII characters																		
3	File type	3	* Contains a file type of 3 ASCII characters.																		
4	Logical extent	1	<p>* Contains the logical extent number(00H to 1FH) of this directory entry.</p> <p>* One directory entry can handle a file area of up to 16k bytes. Two or more entries are used for a file larger than 16h bytes. These entries are identified by numbers, called logical extent numbers, assigned to them sequentially starting at 00H.</p>																		

No.	Item	Size	Description
5		1	* Set to 00H.
6		1	* Set to 00H.
7	Total records	1	<p>* Contains the total number of records (00H to 80H) managed in this directory entry.</p> <p>* A record is the unit of data accessed by CP/M. Its size is 128 bytes. Since a single directory entry can handle a maximum of 16k bytes of file data, the greatest total number value for a single entry is 128(80H).</p>
8	Disk allocation map		<p>* Contains the disk allocation map of the ROM as a ROM capsule or cartridge. The disk allocation map is made up of block numbers which identify the blocks allocated for the file (files are managed in 1K-byte units). The block number begins (with 1 which identifies the first 1k byte of the file. Since the disk allocation map is 16 bytes long, it can manage a maximum of 16 blocks (16k bytes) of file data.</p>  <p>The diagram illustrates the structure of the disk allocation map. It consists of a vertical stack of sections:</p> <ul style="list-style-type: none"> Header area: The top section. Directory area: The second section, with a bracket indicating it is $128\text{byte} \times n$. Block 1: A 1kbyte block. Block 2: A 1kbyte block. Block 3: A 1kbyte block. <p>The blocks are separated by dashed lines, and a wavy line at the bottom indicates that the structure continues.</p>