

EPSON

CARD-PC

ROM Adaptation Kit for DBIOS

Manual



1. Introduction..... 1

2. What is the RAK?..... 2

2.1 RAK Functions 2

 2.1.1 Function for Adjusting and Setting System Parameters in the BIOS..... 2

 2.1.2 Function for Installing Other Software or Data in the CARD-PC 2

2.2 Notes on Using the RAK..... 2

2.3 Operating Configuration 3

3. RAK Installation..... 4

3.1 Contents of the RAK Disk..... 4

 3.1.1 BIOS Directory 4

 3.1.2 RAKDA Directory 5

 3.1.3 Naming the BIOS Carton File 6

3.2 RAK Installation..... 6

4. CARD-PC BIOS Overview..... 7

4.1 What is the CARD-PC BIOS?..... 7

4.2 CARD-PC BIOS Configuration 7

4.3 Modifying the CARD-PC BIOS..... 8

4.4 What Is a Carton File?..... 9

 4.4.1 Carton File Configuration 9

 4.4.2 Changing a Carton File and Displaying its Contents 12

 4.4.3 What Is an Option ROM Program?..... 12

4.5 From Carton File to ROM Image 14

4.6 BIOS Operation at CARD-PC Startup 15

5. Overview of Operation..... 17

5.1 Preparing Carton Files..... 17

5.2 Changing the Carton File 17

 5.2.1 Listing the Contents of a Carton File: BMLIST..... 18

5.3 Modifying the BIOS Parameters 18

5.4 Adding and Deleting Packets	19
5.4.1 Adding a Packet: BMADD	19
5.4.2 Deleting a Packet: BMDEL.....	19
5.4.3 Changing a Packet Mapping Address: BMPAR	20
5.5 Setting a Data Block Start Address	21
5.6 Creating a BIOS ROM Image	22
5.6.1 Error Messages.....	22
5.7 Installing in the CARD-PC.....	23
5.7.1 Using WFLASH.EXE	23
5.7.2 Using ROM WRITER.....	25
6. About RAKDA.....	27
6.1 Operation.....	27
6.1.1 Keyboard Commands.....	27
6.1.2 Mouse Operation	27
6.1.3 BIOS Creation Procedure.....	28
6.1.4 Using Help.....	28
6.1.5 Other Functions	29
6.2 Parameter Setting Procedure	30
6.2.1 Blue Items.....	30
6.2.2 Selecting a Menu Choice.....	30
6.2.3 Numeric Input	30
6.2.4 Character Input.....	30
6.3 Using BPX Files	31
6.3.1 What Are BPX Files.....	31
6.3.2 Displaying a BPX file.....	31
6.3.3 Loading a BPX File	31
6.4 RAKDA Functions	32
6.4.1 File I/O Functions.....	32
6.4.2 Undo Function.....	32
6.4.3 View Function	32
6.4.4 Back Function.....	32
6.4.5 Help Function	33

6.5 RAKDA Specifications	33
6.5.1 Operating Environment	33
6.5.2 Command Format	33
6.5.3 File Name Argument	33
6.5.4 Parameter Table Set Version (TSV)	34
6.5.5 Script Files	34
6.5.6 BPX Files	34
6.5.7 Error Messages	34
7. Command Descriptions	37
7.1 BMCARTON	37
7.2 BMADD	38
7.3 BMDEL	41
7.4 BMLIST	41
7.5 BMAKE	43
7.6 BMPAR	43
7.7 PCOPY	44
7.8 Error Messages	45
7.9 WFLASH	51
7.10 RAKDA	51
8. Implementation Example	52
8.1 Adding an OPTION ROM image	52
8.1.1 Adding an option ROM program in the flash ROM built into the CARD-PC	52
8.1.2 Adding an option ROM program on the ISA bus	54
8.1.3 Adding a VGA BIOS from another manufacturer	55
8.1.4 Increasing the Flash ROM Free Area	56
8.1.5 Changing the method of compression	56
8.1.6 Using rare data packets	57
8.1.7 Setting an optimum data block start address	58
8.1.8 Using CMOS-RAM for user applications	59

9. Packet Installation Utility (ROMADD.EXE)	60
9.1 ROMADD.EXE	60
9.2 ROMADD.EXE Script Files	60
9.3 ROMADD.EXE Operation	64
10. LFLASH.EXE	65
10.1 Introduction	65
10.2 Using LFLASH.EXE.....	65
10.3 LFLASH.EXE Functions	65
10.4 How to Use LFLASH.EXE.....	68
10.4.1 Writing to Flash ROM.....	68
10.4.2 Verifying the Data in Flash ROM.....	68
10.4.3 Reading Data from Flash ROM	68
10.4.4 Erasing Data in Flash ROM	68
10.5 LFLASH.EXE Error Messages.....	69
11. Glossary	71

1. Introduction

The ROM Adaptation Kit (referred to hereafter as "RAK") is an extremely convenient and powerful system development tool for modifying the CARD-PC BIOS functions, and installing system programs and data in the built-in flash ROM. The RAK consists of the ROM Adaptation Kit for DBIOS Adjustment (RAKDA) program and the necessary BIOS data.

The CARD-PC provides an IBM-PC/AT compatible computer at a credit-card size, and is a product ideal for embedded systems, with a range of expansion functions. By installing a ROM version of the operating system, customer software, and so forth in the CARD-PC, it is easy to assemble a compact, low power consumption system. The RAK is a CARD-PC system development tool provided to allow customers to use these functions.

This manual describes the various functions of the RAK, and how to use them.

Some versions of the CARD-PC (CARD-486D4 etc.) incorporate a local flash ROM in addition to the 256KB of flash ROM. The RAK includes a utility (LFLASH.EXE) for managing this local flash ROM. It also includes a utility (ROMADD.EXE) which allows easy installation of binary files (option ROM or MS-DOS® ROM version etc.) into the BIOS carton file. ROMADD.EXE uses script files to add or delete binary files, automatically set the data block start address, and to perform other functions.

This document also describes how to use these utilities.

2. What is the RAK?

2.1 RAK Functions

The BIOS stored in the flash ROM of the CARD-PC includes parameters for controlling the system configuration, and parameters for a range of system function settings. To adjust these parameters, or to install other software or data in the flash ROM together with the BIOS requires the RAK. The RAK offers the following functions.

2.1.1 Function for Adjusting and Setting System Parameters in the BIOS

The RAK includes what is termed a "carton file", which is a file based on the original BIOS ROM image. By changing values in parameter tables within this file, various BIOS functions can be selected or deselected, and it is also possible to change the default settings of the BIOS itself.

2.1.2 Function for Installing Other Software or Data in the CARD-PC

The CARD-PC has 256 KB of built-in flash ROM. Of this, the area used by the BIOS depends on the BIOS version and the functions provided in the system, but at least half is empty, and available to the user. By using the RAK, it is possible to install system programs and data in this empty area. Further by deleting unwanted BIOS function modules, this free area can be enlarged.

2.2 Notes on Using the RAK

Before using the RAK, be sure to check the RAK software license agreement enclosed in the package.

The RAK is supplied on a disk (marked "CARD-xxx ROM Adaptation Kit with RAKDA Tools"). The disk has a BIOS directory and RAKDA directory. The BIOS directory contains the BIOS data files for the CARD-PC and the various parameter files for the RAKDA program. The RAKDA directory contains programs for BIOS modification and installation. The RAKDA program may not be able to handle data from a different BIOS version. If the RAK version has been updated, be sure to install the latest RAKDA program as described in "3. RAK Installation".

2.3 Operating Configuration

To use a program included in the RAK requires the following system:

- IBM PC/AT compatible computer
- Version 3.1 or later of MS-DOS, or compatible operating system
- Keyboard
- CRT or LCD monitor
- Hard disk
- 3.5-inch 1.44 MB floppy disk drive

Further to use the RAK to write an updated BIOS to the CARD-PC requires a ROM-writer, evaluation kit, development kit, or other hardware that supports a function of writing data to the flash ROM in the CARD-PC.

3. RAK Installation

3.1 Contents of the RAK Disk

The RAK is supplied on a disk (marked "CARD-xxx ROM Adaptation Kit with RAKDA Tools"). The BIOS directory on this disk contains data files required for making a new BIOS, as well as various parameter files. The RAKDA directory contains programs for making modifications to the BIOS, and for adding programs and creating a new BIOS binary image.

3.1.1 BIOS Directory

The BIOS directory contains the following files.

BIOS carton file

A data file which serves for creating a new BIOS is called a carton file. Carton files have the extension CTN. Store the original carton file carefully, because it is required to create a standard BIOS. For information on naming the BIOS carton file, please refer to section "3.1.3 Naming the BIOS Carton File".

BPX files

BPX files serve to set parameters such as LCD parameters, ATA Boot parameters and other parameters controlled by RAKDA. Values to be set in the BIOS together and comments are stored in text file format. BPX files have the extension BPX.

A BPX file for the standard BIOS named ###Vnnn.BPX is also provided. The "####" section uses the same principle as used for naming BIOS carton files. For information, please refer to section "3.1.3 Naming the BIOS Carton File".

Setup utility (SETUP###.COM)

This is a binary file that serves for ROM setup. It is required if the ROM setup has been deleted while creating the CARD-PC BIOS. This binary file can be executed from DOS to perform setup on systems where the ROM setup routine has been deleted. The "####" section uses the first three characters according to the principle for naming BIOS carton files (see section "3.1.3 Naming the BIOS Carton File").

For example, for the CARD-486HB, the file is called SETUP4HB.COM.

SCX files (BPARM###.SCX)

SCX files are binary files which contain various menus and selection choices for the BIOS to be used with RAKDA. SCX files have the extension SCX.

The "####" section uses the first three characters according to the principle for naming BIOS carton files (see section "3.1.3 Naming the BIOS Carton File").

For example, for the CARD-486HB, the file is called BPARM4HB.SCX.

3.1.2 RAKDA Directory

RAKDA program

The program is called RAKDA.EXE.

RAKDA script files

These are binary files which contain various menus and selection choices. RAKDA script files have the extension SCX.

Carton file management utilities

The following utilities for managing carton files are provided.

- BMCARTON.EXE for changing carton file attributes
- BMPAR.EXE for changing carton file registration information
- BMADD.EXE, BMDEL.EXE for changing the carton file configuration
- BMLIST.EXE for checking the contents of carton files
- BMMAKE.EXE for converting a carton file into a binary image

BIOS flash ROM write utility (WFLASH.EXE)

This utility serves for writing the BIOS binary image created with RAKDA to the BIOS flash ROM (256KB) in the CARD-PC.

Local flash ROM write utility (LFLASH.EXE)

This utility serves for writing a binary image to the local flash ROM for CARD-PCs equipped with a local flash ROM.

Packet addition utility (ROMADD.EXE)

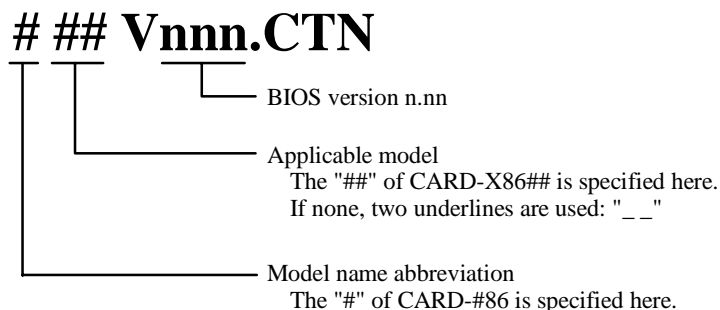
This utility serves for adding a binary file to the BIOS carton file. ROMADD.EXE operates according to a script file and can add or delete a binary file and automatically set the data block start address etc. For information on how to use ROMADD.EXE, please refer to section "9.1 ROMADD.EXE".

Packet addition utility script file (ROMADD.SCR)

This is a sample script file for use with the ROMADD.EXE utility. For information on script file syntax, please refer to section "9.2 ROMADD.EXE Script File".

3.1.3 Naming the BIOS Carton File

The BIOS carton file is named according to the following rules.



Example

Carton file for CARD-486HB BIOS Ver. 4.00: 4HBV400.CTN

Carton file for CARD-586HB BIOS Ver. 1.00: 5_ _V100.CTN

Carton file for CARD-686 BIOS Ver. 1.00: 6_ V100.CTN

3.2 RAK Installation

First make sure that a development system complying with Section 2.3, "Operating Configuration" is available. Next, use the following procedure to install the RAK in the development system. The following description assumes that the installation is carried out in directory C:\RAK.

After inserting the RAK disk into drive A, perform the following procedure.

```

C:\>md rak
C:\>cd rak
C:\RAK> copy a:\rakda\*.*
C:\RAK> copy a:\bios\*.*
  
```

Next, add the new directory to the path.

```

C:\RAK> path = c:\rak;%path%
  
```

When wishing to permanently use this path, add the statement to C:\AUTOEXEC.BAT.

This completes the installation.

4. CARD-PC BIOS Overview

This section describes the basic configuration of the CARD-PC BIOS, and provides information about the basic configuration of the carton files which will be useful when carrying out modifications with the RAK.

4.1 What is the CARD-PC BIOS?

BIOS is an acronym which stands for "Basic I/O Subsystem". The BIOS is software which carries out basic hardware operations such as disk access and screen display. The CARD-PC BIOS is software which enables the CARD-PC to function as an IBM PC/AT compatible computer. Seiko Epson calls the BIOS based on the Falconer chip set and enhanced with installation functions the DBIOS. All CARD-PCs are shipped with the standard CARD-PC BIOS installed in the internal flash ROM.

The CARD-PC BIOS is an original from Seiko Epson. In this documentation, BIOS always refers to the CARD-PC BIOS (DBIOS).

4.2 CARD-PC BIOS Configuration

The BIOS is contained within the flash ROM in the CARD-PC, and comprises a total of 256K bytes of data and programs. The CARD-PC BIOS is divided into modules, each for a different function, and their names are as follows.

Module name	Function
BootBlock	At startup, positions other binary images correctly.
CBIOS	PC/AT compatible BIOS
VGA-BIOS	Enables the VGA built into the CARD-PC.
Minimizer	Carries out power management/keyboard emulation functions.
SETUP	Enables hardware setting changes at startup.

BootBlock

This is the first module executed at startup. It expands the compressed binary image files in flash ROM, and installs them in the appropriate locations in RAM (shadow RAM).

CBIOS

This module provides compatibility with an IBM PC/AT. This is the main BIOS. At startup it initializes hardware, and carries out various BIOS services.

VGA-BIOS

This module enables the VGA built into the CARD-PC. If the built-in VGA will not be used, this can be deleted using the RAK.

Minimizer

This module is an SMI handler for power management functions. SMI stands for "System Management Interrupt". The module also includes keyboard emulation functions and cannot be deleted.

SETUP

This module is accessed by pressing the F10 key during system startup. It carries out an automatic check of the hardware, and changes the CMOS-RAM to correct values. It also allows the user to change hardware settings. If this function is not required, it can be deleted using the RAK.

The SETUP module is different from the other modules in that the mapping destination address can be changed. SETUP must be placed on a 2 KB boundary in the address range FC0000h to FF0000h (rare data). In practice, however, the upper address limit which can be specified is lower than this, to avoid conflicts with other modules.

4.3 Modifying the CARD-PC BIOS

The BIOS is hardware-dependent software, and therefore if the hardware is changed a change to the BIOS is normally required. It is possible to carry out simple changes at startup time, using SETUP, but depending on the details of the changes, a change to the BIOS may be necessary. However, it is not possible to change the binary BIOS data (BIOS ROM image) held in the flash ROM. Therefore, the RAK provides "Carton Files." Using these carton files, it is possible to change the BIOS.

4.4 What Is a Carton File?

A carton file is a file which forms the basis of a BIOS ROM image. The BIOS comprises a number of binary images, and this is a container which holds these together. The user cannot directly modify the BIOS ROM image, but by using carton files, it is possible to change the BIOS.

4.4.1 Carton File Configuration

A carton file comprises a number of packets (Fig. 1). Each packet consists of binary image and packet attributes. The packet attributes include the following information:

- Packet type
- Packet name
- Deletion from carton file allowed/prohibited
- Update allowed/prohibited
- Binary image size
- Binary image mapping destination address
- Other information depending on the packet type

There are the following packet types.

Packet type	Characteristics
Shadow RAM data packet	Copied to shadow RAM at startup. Any number can be added. "CBIOS" cannot be deleted.
Rare data packet	flash ROM data used without change. Any number can be added.
BootBlock data packet	Has a BootBlock as a binary image. One only. Cannot be deleted.
Minimizer data packet	Has an SMI handler as a binary image. One only. It cannot be deleted.
Carton control data packet	Holds carton file attribute data. One only. Cannot be deleted.

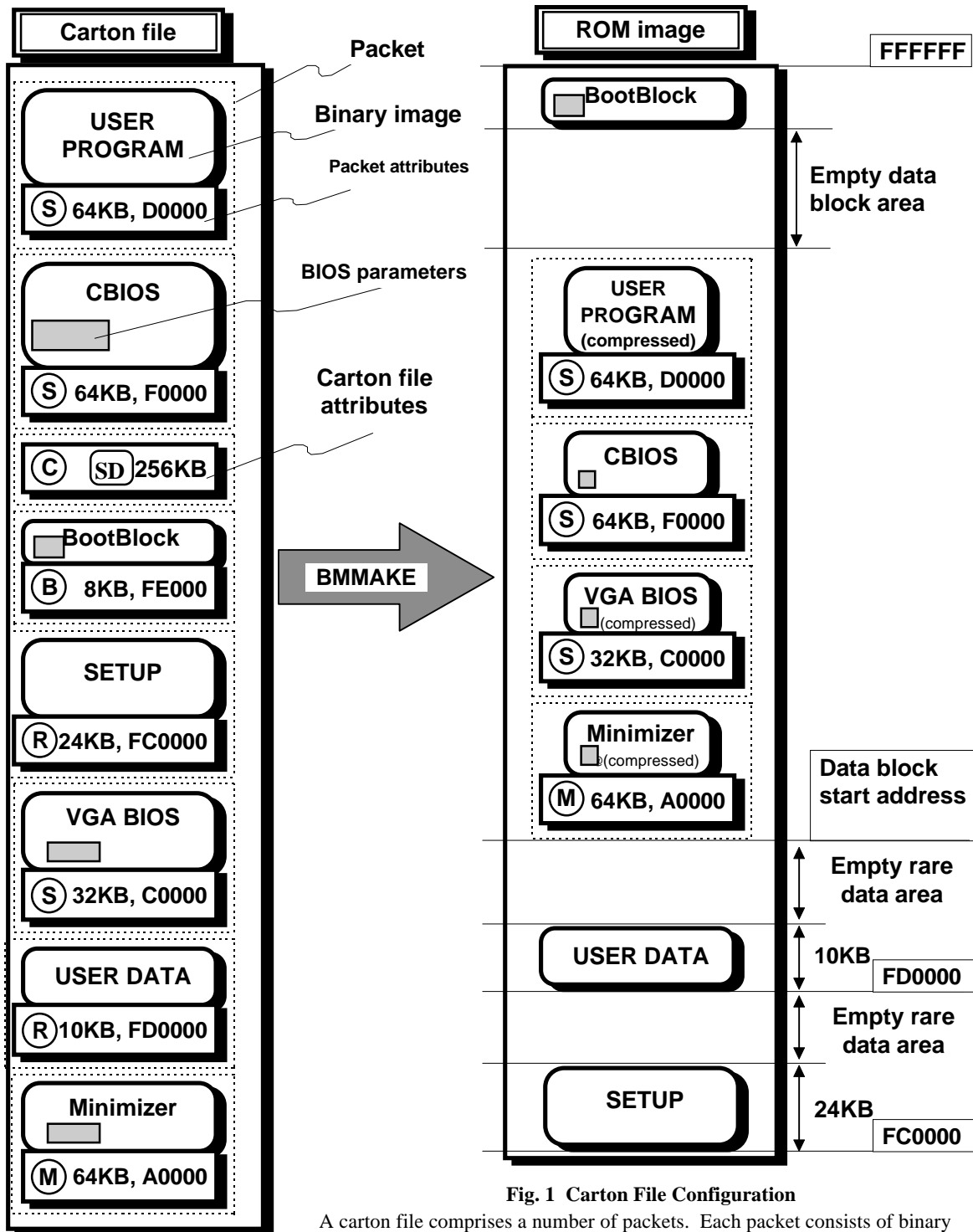


Fig. 1 Carton File Configuration

A carton file comprises a number of packets. Each packet consists of binary image and packet attributes. The packet attributes include information such as the binary image size, the mapping destination address, and packet type (S, R, B, M, or C). The carton control data packet contains information such as the compression method, and ROM image size.

The ROM image of the carton file can be changed with BMMAKE. For a rare data or BootBlock data packet the binary image only is moved to the specified address. On the other hand, shadow RAM data and Minimizer data packets are compressed, and placed in the data block area.

Note that addresses and sizes shown in this figure are only given as examples.

- (S) Shadow RAM data
- (R) Rare data
- (B) BootBlock data
- (M) Minimizer data
- (C) Carton control data

Shadow RAM data packet

This is copied to shadow RAM at startup. It is compressed within the ROM image, and is expanded in RAM (shadow RAM) at the correct address at startup by BootBlock.

- Any number of shadow RAM data packets can be added.
- In general shadow RAM data packets can be deleted, but the "CBIOS" packet already present in the carton file cannot be deleted.

Rare data packet

This packet is used as flash ROM data without change. As a result, when converted to a ROM image, this is loaded at the appropriate address without compression.

- Any number of rare RAM data packets can be added.
- Rare RAM data packets can be deleted.

BootBlock data packet

This packet has a BootBlock as a binary image. The BootBlock is a module which at system startup expands binary images compressed in the flash ROM to the appropriate address in RAM (shadow RAM).

- Only one BootBlock data packet can be inserted in a carton file.
- The BootBlock data packet cannot be deleted from the carton file.

Minimizer data packet

This packet includes a "Minimizer," required for using the power management functions of the CARD-PC. The Minimizer is an SMI (System Management Interrupt) handler. If the user creates a special SMI handler, this too can be added to the carton file as a Minimizer data packet. Like a shadow RAM data packet, a Minimizer data packet is compressed within the ROM image. At startup, this expanded by BootBlock to a special RAM called SM-RAM. SM-RAM is normally invisible, but can be accessed by the CPU when an SMI is issued.

- Only one Minimizer data packet can be inserted in a carton file.
- The Minimizer data packet can never be deleted.

Carton control data packet

A carton control data packet contains no binary image, and is a special packet type which holds carton file attributes within the packet attributes. There are the following types of carton file attributes.

Attribute	Details
ROM size	256KB (fixed)
Data block compression method	Selected from Self-dictionary/Run-Length, or none
Data block start address	Start address of a packet compressed in the ROM image.

4.4.2 Changing a Carton File and Displaying its Contents

By modifying a carton file, the user can change the BIOS.

The following operations are possible on a carton file.

Operation	Utility used
Adding a packet	BMADD
Deleting a packet	BMDEL
Changing the address of a packet	BMPAR
Changing the carton file attributes	BMCARTON
Changing the BIOS parameters	RAKDA
Displaying the contents of a carton file	BMLIST

4.4.3 What Is an Option ROM Program?

This section describes an "Option ROM Program," which is a typical set of shadow RAM data used by a user.

An option ROM program is located in the space between addresses C0000h and EFFFFh and is executed at system startup. This is basically an additional ROM provided at this address on the ISA bus for expanding the BIOS functions, but if placed in shadow RAM at the same address, this is still termed an option ROM program.

Example option ROM programs:

- VGA BIOS
- SCSI BIOS
- ROM version OS (MS-DOS® ROM version etc)

At system startup, CBIOS checks the data in the range C0000h to EFFFFh, and if data following the conventions for an option ROM is present, recognizes the option ROM program, and calls its initialization routine. The conditions for being recognized as an option ROM are as follows:

- Must be located on a 2 KB boundary in the address range C0000h to EFFFFh
- Must have an option ROM header
- Size must be less than 64 KB, and an integral multiple of 512 bytes.
- Checksum must match ($\sum \text{data (byte)} = 0$)

Because the CBIOS uses a far call to the option ROM routine, design it in such a way that it returns with "RETF".

Particularly when using BMADD and BMPAR described below, care must be taken over the address conditions.

The flash ROM built into the CARD-PC is mapped to addresses FC0000h to FFFFFFFh. As a result, when adding an option ROM program to a carton file, it must be made into a shadow RAM data packet.

[Reference] Option ROM format

Byte	Contents
0	55h (header)
1	AAh (header)
2	ROM size (512-byte units)
3	Entry point of initialization routine
4 to last	Program code

4.5 From Carton File to ROM Image

The carton file is converted by BMMAKE into a ROM image for writing to flash ROM. At this point, the ROM image size and method of compression are determined from the attributes of the carton file. Depending on the packet type, the position of storage within the ROM image varies (Fig. 1).

a) Objects which will be located at the appropriate address at system startup, but which can be anywhere in the ROM image:

- Shadow RAM data packets
- Minimizer data packets

b) Objects which must be at the correct address in the ROM image:

- Rare data packets
- BootBlock data packets

c) Objects which have no substantive existence, and cannot be placed in the ROM image:

- Carton control data packet

The packets in a) are actually located in a place known as the data block area. Of the packets in b), the location of the rare data packets is called the rare data packet area.

Data blocks and data block area

A "data block" is a collection of packets compressed within the ROM image. At system startup, the data blocks are expanded by BootBlock, and located at the correct addresses.

The "data block area" is the area of the ROM image from the data block start address, which is one of the carton file attributes, to immediately before BootBlock. A data block must always be located within the data block area. When compressed with BMMAKE, if a data block cannot be fitted into the data block area, an error results. At this point, it is necessary either to reduce the data block start address to increase the data block area, or to make the data block packet smaller.

4.6 BIOS Operation at CARD-PC Startup

This section describes simply the BIOS operation at CARD-PC startup.

BootBlock startup

When the system is started, the CPU executes the code at a particular address. In the CARD-PC BIOS, the BootBlock is this code. The BootBlock first looks for a data block in flash ROM. When a data block is found, it creates shadow RAM in the area to which mapping will take place. The data block is then expanded at the correct location. In this way, the memory map is changed as shown in Fig. 2. After the BootBlock has carried out the expansion, the data block is a collection of data with no further significance.

For a rare data area, however, BootBlock carries out no processing. The data in the rare data area remains in flash ROM, and can also be used after system startup.

CMOS-RAM check

After the BIOS main CBIOS has been expanded, control is transferred to it. The CBIOS first checks the contents of CMOS-RAM, and if the contents are corrupted copies the default CMOS-RAM settings from flash ROM. (This operation can be changed with the RAK.)

POD execution

Next the CBIOS carries out a hardware check and initialization. This is known as "POD" (Power On Diagnostics). At this point, if the hardware configuration in CMOS-RAM and the actual hardware are different, an error is displayed. Additionally if any device is found to be faulty, an error is displayed.

F10 key acceptance

From a point some way through the POD, an F10 key press is accepted to carry out SETUP. Again, if an error occurs during the POD, the system waits for an F10 key input for SETUP and an F1 key input to continue. If F10 is pressed, the CBIOS looks for SETUP in the flash ROM. If SETUP is found, control is passed to SETUP. After completion of SETUP execution, control returns to "BootBlock startup."

Int 19h execution

Once all of the initializations are completed, INT 19h is called, to pass control to the operating system. The CBIOS INT 19h handler searches for a system disk, searching floppy disks then hard disks. If an option ROM program such as a ROM version of the operating system has changed the INT 19h handler, then control is passed accordingly.

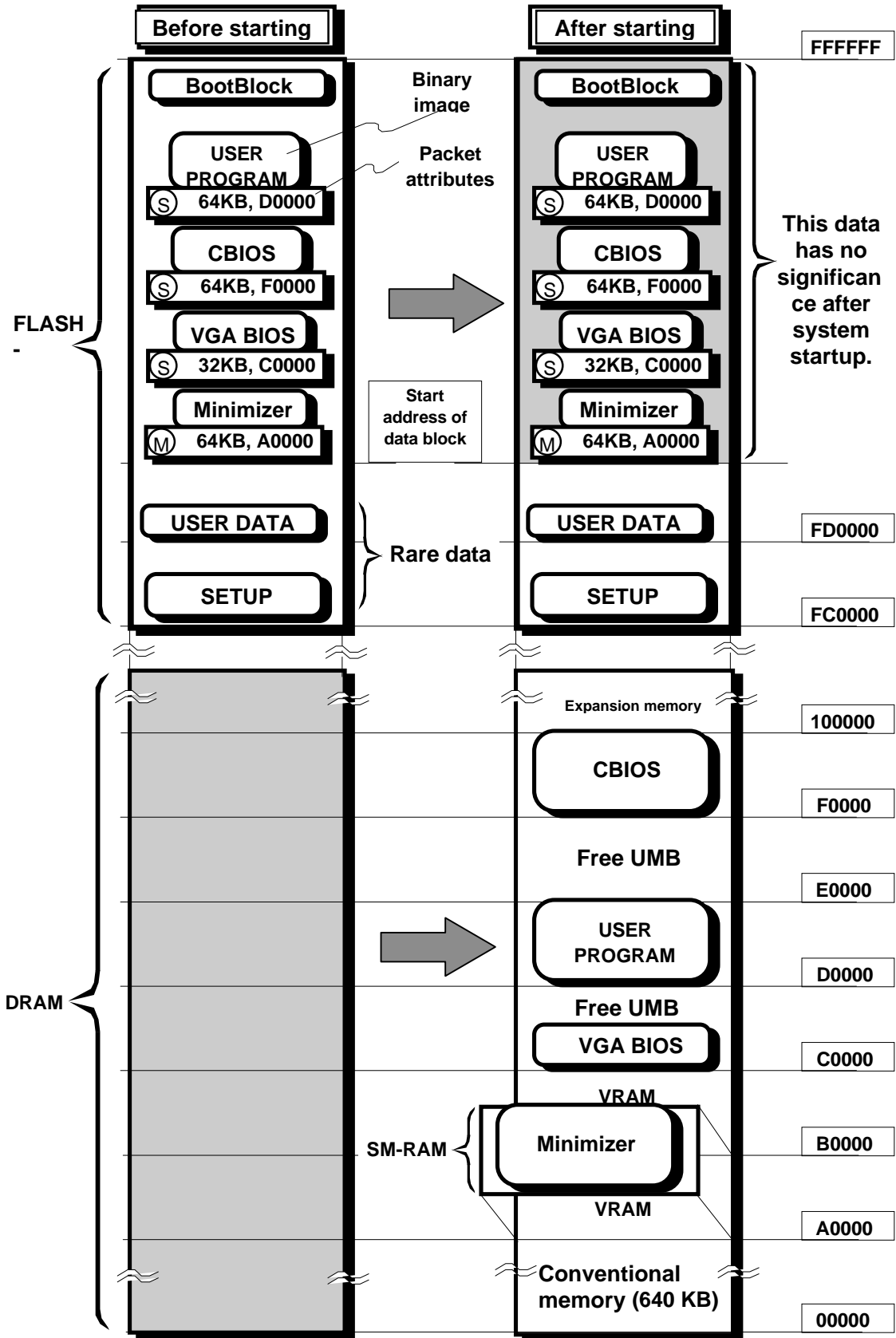


Fig. 2 Memory maps before and after startup

A ROM image created with BMMAKE is held in the flash ROM built into the CARD-PC. At startup time, the data block which was compressed and located in the flash ROM is expanded to the correct address in RAM by BootBlock. This is called "Shadowing." After system startup, the data in the data block area has no significance. Rare data, on the other hand, is left in flash ROM and used there.

5. Overview of Operation

This chapter describes the sequence of operations from using the RAK to modify the BIOS to writing to the CARD-PC. It assumes that the RAK has been installed in the directory C:\RAK, using the procedure in 3. RAK Installation.

5.1 Preparing Carton Files

First the carton files must be prepared. The carton files are the files in the directory C:\RAK with the extension .CTN.

Since it is convenient to preserve the original files, copy these files with different names in a different directory. The copy will be made to the file name TEST.CTN in the directory C:\RAK.

```
C:\>cd rak
```

```
C:\RAK> copy ###Vnnn.ctn test.ctn
```

###Vnnn.ctn: carton file name

For information on BIOS carton file naming, see section "3.1.3 Naming the BIOS Carton File".

5.2 Changing the Carton File

By changing the carton file, the user can modify the BIOS.

The following modifications are possible on a carton file.

Operation	Utility used
Adding a packet	BMADD
Deleting a packet	BMDEL
Changing the address of a packet	BMPAR
Changing the carton file attributes	BMCARTON
Changing the BIOS parameters	RAKDA

5.2.1 Listing the Contents of a Carton File: BMLIST

It is possible to check the contents of a carton file by using BMLIST.

```
C:\RAK>bmlist test.ctn | more
```

This displays the carton file attributes and the attributes of each packet. If there are problems in the file such as overlapping packet addresses, an error message is displayed.

5.3 Modifying the BIOS Parameters

The BIOS parameters are modified using the carton file.

```
C:\RAK>rakda bparm###.scx test.ctn  
bparm###.scx: script file name
```

The "###" part of the script file name uses the first three characters according to the principle for naming BIOS carton files (see section "3.1.3 Naming the BIOS Carton File").

Alternatively, for opening the script file and carton file within RAKDA:

```
C:\RAK\rakda
```

For information on operation of the RAKDA utility, please refer to section "6. About RAKDA". In RAKDA, parameters can be set in two ways, either "individually" or "by purpose".

5.4 Adding and Deleting Packets

In this section, the packets in a carton file are changed.

5.4.1 Adding a Packet: BMADD

To add a packet, use BMADD. As an example, the binary data in the file USER.BIN will be added to TEST.CTN. The user program in the file USER.BIN must already be present in the directory C:\RAK.

Adding as shadow RAM data

```
C:\RAK>bmadd test.ctn user.bin "UserPro" /a:d0000
```

In this way, a shadow RAM data packet with the packet name "UserPro" is added. When the system starts up, this data will be loaded at address D0000h in shadow RAM.

Adding as rare data

To add rare data, use the /R flag in BMADD.

```
C:\RAK>bmadd test.ctn user.bin "UserPro" /a:d0000 /r
```

In this way, a rare RAM data packet with the packet name "UserPro" is added. This data is located at address FD0000h in flash ROM.

Note

The flash ROM built into the CARD-PC is mapped to the 256 KB at addresses FC0000h to FFFFFFFh. Specifying the /R flag adds F00000h to the address. Therefore, the address in the /A: parameter does not require an F in the sixth digit (from the right).

5.4.2 Deleting a Packet: BMDEL

If SETUP or Minimizer is not required, or in order to remove a user program added with BMADD from the carton file, use BMDEL. As an example, the following deletes Minimizer:

```
C:\RAK>bmdel test.ctn setup
```

Undoing a deletion

It is not possible to return the carton file to the state before the deletion, but a backup file is automatically created. Using this, you can retrieve the file from before the deletion.

```
C:\RAK>copy test.bak test.ctn
```


5.4.3 Changing a Packet Mapping Address: BMPAR

To change the mapping address of a packet already present in the carton file, use BMPAR.

It is not possible to change the mapping address of BootBlock, CBIOS, VGA BIOS, or Minimizer, but SETUP can be moved. When the SETUP address conflicts as a result of adding rare data, for example, use BMPAR to move it.

```
C:\RAK>bmpar test.ctn setup /a:c8000
```

This moves SETUP to address FC8000 (original value is FC0000).

Note 1

BMPAR only changes the address, and cannot change the type of a packet. Therefore it is not possible to change a packet of rare data at the address FC0000 to shadow RAM data at address C8000. To change the type of a packet, first delete it with BMDEL, then add the new version with BMADD.

Note 2

If you move SETUP, ensure that it is located on a 2 KB boundary. SETUP must always be installed as rare data.

Note 3

Using ROMADD, SETUP can be moved automatically.

Notes on installing an option ROM program

An option ROM program is a program installed in the address range C0000h to EFFFFh at system startup. This is basically an additional ROM provided at this address on the ISA bus for expanding the BIOS functions, but if placed in shadow RAM at the same address, this is still termed an option ROM program.

Example option ROM programs:

- VGA BIOS
- SCSI BIOS
- ROM version OS (MS-DOS® ROM version etc.)

At system startup, CBIOS checks the data in the range C0000h to EFFFFh, and if data following the conventions for an option ROM is present, recognizes the option ROM program, and calls the initialization routine. The conditions for being recognized as an option ROM are as follows:

- Must be located on a 2 KB boundary in the address range C0000h to EFFFFh
- Must have an option ROM header
- Size must be less than 64 KB, and an integral multiple of 512 bytes.
- Checksum must match (\sum data (byte) = 0)

Because the CBIOS uses a far call to the option ROM routine, design it in such a way that it returns with "RETF".

Particularly when using BMADD and BMPAR described below, care must be taken over the address conditions.

The flash ROM in the CARD-PC is mapped to the address range FC0000h - FFFFFFFh. When adding the option ROM program to the carton file, make it a shadow RAM data packet.

Reference Option ROM format

Byte	Content
0	55h (header)
1	AAh (header)
2	ROM size (512-byte units)
3	Initialization routine entry point
4 - last	Program code

5.5 Setting a Data Block Start Address

It is not necessary to change the data block start address when simply changing the BIOS parameters, but if a packet has been added, a change may be required. To change the data block start address, use the following setting:

C:\RAK>bmcarton test.ctn /s:d8000

The following points must be considered when deciding the data block start address:

- The value must be greater than the last address of rare data.
- It must be a paragraph boundary (last hexadecimal digit in address must be 0).

In practice specifying an address immediately after the rare data allows the maximum number of option ROM programs to be included and increases efficiency.

Note 1

Using ROMADD, the data block start address can be optimized automatically.

5.6 Creating a BIOS ROM Image

When the BIOS modifications are complete, a BIOS ROM image is created.

```
C:\RAK>bmmake test.ctn test.bio
```

This creates a ROM image in the file test.bio. If an error occurs, and the image file is not created, take appropriate steps, referring to the section "5.6.1 Error Messages" below.

5.6.1 Error Messages

- Address of data in execution time is conflicted.

Explanation

There is a conflict in packet addresses.

Countermeasure

Use BMLIST to check and use BMPAR to change the address.

- The Rare data block "XXX" overruns into data block area.

Explanation

The rare overruns into the data block area.

Countermeasure

Use BMLIST to check the data block start address. Specify a higher address with BMCARTON /S:. This can be performed automatically with ROMADD.

- Too small data block area

- Data block conflict is found. "XXX" and "BootBlock"

Explanation

The data block overruns into the boot block area.

Countermeasure

Because the data block is compressed, actual size cannot be predicted. Use BMMAKE and check the "xxxx bytes free in Data Block area" message. If there is not enough room, delete a packet from the data block area.

Make the start address of the data block as low as possible. This can be performed automatically with ROMADD.

5.7 Installing in the CARD-PC

Next the created ROM image is written to the flash ROM built into the CARD-PC. There are two methods: using WFLASH or using ROM WRITER. The following table compares the advantages and disadvantages of these two methods.

Tool	Advantages	Disadvantages
WFLASH	Requires only hardware with Vpp control. It is not necessary to unplug the CARD-PC to rewrite the flash ROM.	Rewriting takes several minutes. If the rewriting fails, it may not be possible to restart the system without using ROM WRITER.
ROM WRITER	Rewriting is fast. Even if the CARD-PC cannot be started, rewriting is possible.	The CARD-PC must be unplugged in order to carry out the rewriting.

5.7.1 Using WFLASH.EXE

WFLASH is a program which uses the CARD-PC to write the CMOS-RAM content and the flash ROM content in the CARD-PC. Even when the flash ROM is overwritten, the BIOS itself is present in shadow RAM, so that until a reset is carried out, or the system powered off, the BIOS continues operating as before. If, however, the BIOS ROM image includes a ROM version OS or ROMDISK as rare data, then this may not continue to operate normally after the flash ROM is rewritten. Even in this case, a hardware reset will restart the BIOS. When WFLASH terminates, the following message appears:

REBOOT now, to enable the new BIOS? [Y: Yes/N: No]

Be sure to press the Y key to reset the system.

If the BIOS data are invalid or if an error occurs during the write process, the PC may become unable to boot until the ROM writer described in the next section is used.

To use WFLASH, the hardware must be compatible with Vpp control. The following boards supplied by Seiko Epson are compatible:

- Evaluation kit (DIO5300)
- CardPresso (SEK0630)

Evaluation kit (DIO5300) is not available for CARD-686.

Hardware preparations

Hardware which supports WFLASH is required. A disk drive (FDD, HDD, etc.) which can store the BIOS ROM image and WFLASH itself is also required.

Software preparations

WFLASH is an application which runs under MS-DOS®. The above disk must be preformatted as a startup disk. Copy WFLASH and the BIOS ROM image to it.

CONFIG.SYS and AUTOEXEC.BAT settings

If there are CONFIG.SYS and AUTOEXEC.BAT files in the root directory of the startup disk, rename them.

Example

```
C:\>ren config.sys config.sy_
```

```
C:\>ren autoexec.bat autoexec.ba_
```

This is because if there are protected mode or virtual 86 mode drivers installed, WFLASH will not operate correctly.

System restarting

Restart the system with no CONFIG.SYS.

WFLASH execution

If WFLASH and the BIOS ROM image are in C:\, proceed as follows:

```
C:\>wflash test.bio
```

WFLASH carries out the following operations:

1. Reading the file(s)
2. Checking the flash ROM built into the CARD-PC
3. Erasing the contents of flash ROM
4. Writing to flash ROM
5. Comparing the contents of flash ROM with the data read from the file(s)

Note

It takes several minutes for WFLASH to complete execution. No messages are displayed during this interval. When WFLASH ends, it returns to the DOS prompt. Until that happens, under no circumstances unplug the CARD-PC or power off or reset. If WFLASH execution is aborted the CARD-PC may not restart.

System reset

When a hardware reset is carried out, the system will start up with the new BIOS. Note that a software reset will use the old BIOS present in shadow RAM.

The reset carried out when WFLASH terminates has the same effect as a hardware reset, i.e. the system will start up with the new BIOS.

5.7.2 Using ROM WRITER

For bulk rewriting of the CARD-PC BIOS, or for installing a ROM version OS, use of the optional ROM WRITER (DIO5302) is recommended.

This section describes in general terms how to carry out rewriting with ROM WRITER. For details of individual commands, refer to the User's Manual supplied with ROM WRITER.

Note

The utilities for ROM WRITER run on an IBM PC/AT compatible computer. They require MS-DOS®.

Installing the ROM WRITER utilities

Copy the ROM WRITER utilities to an appropriate directory. If the ROM WRITER utility disk is in drive A:

```
C:\>md rw
```

```
C:\>cd rw
```

```
C:\RW>copy a:\*.*
```

Preparing the ROM WRITER disk

Insert a new 2HD floppy disk in drive A, and create a startup disk which can be used with ROM WRITER.

```
C:\RW>rwformat a: /s
```

Next, insert another floppy disk, and create a data disk which can be used with ROM WRITER.

```
C:\RW>rwformat a:
```

Copying the BIOS ROM image

Insert the ROM WRITER data disk in drive A, and copy the BIOS ROM image.

```
C:\RW>rwcop y c:\rak\work\test.bio a:0 "This is a test BIOS."
```

The string in double quotes is a comment.

Starting ROM WRITER

Insert the ROM WRITER startup disk in the FDD of the ROM WRITER, and power on the ROM WRITER.

Loading data

Insert the ROM WRITER data disk in the FDD of the ROM WRITER, and press the LOAD button.

Writing the BIOS

When the load operation is completed, insert the CARD-PC in the CARD-PC slot of the ROM WRITER, and press the PROGRAM button. When the ACTIVE indicator goes off, the writing is completed.

Note

Under no circumstances unplug the CARD-PC while the ACTIVE indicator is on. Unplugging the CARD-PC while the ACTIVE indicator is on may damage the CARD-PC.

6. About RAKDA

The ROM Adaptation Kit for DBIOS Adjustment (RAKDA) program is a tool for editing Seiko Epson BIOS carton files. RAKDA uses Windows® conventions for mouse and keyboard commands and allows adjusting the various BIOS parameters.

6.1 Operation

6.1.1 Keyboard Commands

Ctrl-Tab is used to switch back and forth between the help screen and the main screen. The cursor up/down keys and the PgUp and PgDn keys can be used for scrolling. Pressing Alt-H once more turns the help screen off.

Key	Function
↑ ↓ ← →	Move up/down/left/right
Tab	Move to next item
Shift-Tab	Move to previous item
PgUp/PgDn	Page up/page down
Home	Move to top field
End	Move to bottom field
Ctrl-Tab	Switch between help screen and main screen
F1	Display context-sensitive help
Alt	Make menu active
Esc	Make menu inactive
Enter	Execute command/jump to linked page
Space	Select/deselect item

6.1.2 Mouse Operation

Left button: Execute command or jump to linked page. Select or delete item.
 Right button: Display context-sensitive help.

6.1.3 BIOS Creation Procedure

Using the steps described below, the BIOS parameters of the BIOS carton file can be changed.

(1) Open the script file (*.SCX)

Open the script file necessary for BIOS adjustment.

1. Use "Script list" from the File menu (Alt-F + L) to call up a list of script files.
2. Select the desired script file (extension SCX).

(2) Open the carton file (*.CTN)

Open the carton file to be edited.

1. Use "Open Carton File" from the File menu (Alt-F + C) to call up a list of carton files.
2. Select the desired carton file (extension CTN).

(3) Adjust the parameters

Adjust the parameters. For details, please refer to section "6.2 Parameter Setting Procedure".

(4) Write the carton file

Store the carton file with the new content.

Use "Write parameters" from the File menu (Alt-F + W) to write the carton file to disk.

(5) Terminate the program

Use "Exit" from the File menu (Alt-F + E) to terminate the program.

(6) Create the BIOS binary file

Convert the carton file into a BIOS binary file.

(7) Write the BIOS

Write the BIOS binary file to the BIOS ROM of the CARD-PC.

6.1.4 Using Help

Pressing Alt-H or selecting the Help menu calls up a help screen to the lower half of the screen.

Pressing Alt-H or selecting the Help menu once more turns the help screen off.

Ctrl-Tab allows switching between the help screen and the main screen. Within the help screen, the cursor up/down keys and the PgUp and PgDn keys can be used for scrolling.

By dragging the border between the help screen and the main screen with the mouse, the size of the help screen can be adjusted.

6.1.5 Other Functions

Mouse operation

- The mouse can be used within the program. Normally, the left mouse button is clicked once to make a selection.
- In the file open dialog box, double-clicking the left mouse button on a file or directory can be used to select it.
- Click the up and down arrows at the right edge of the screen to scroll the screen.

Keyboard operation

- Use Tab and Shift-Tab to move between items on a page. Use the up/down/left/right cursor keys to move within a field.
- Use Home to move to the first item on a page.
- Use End to move to the last item on a page. This is also useful to return to the previous page.

Cut and paste

- When used in a DOS box under Windows®, cut and paste of character strings can be carried out. Use the "Paste" command from the "Edit" menu.

"More" indicator

- If a page does not fit on a single screen, the "More" indicator is shown. Use the cursor keys etc. to scroll.

Title bar

- The topmost line of the screen shows the name of the currently loaded carton file and script file.
- If no file is open, "Carton file not specified", "Script file not specified" is shown.
- If the directory name and file name is very long, the end will be truncated.

Grayed-out items

- Items that cannot be set are shown in light gray. This applies for example when no carton file is open, or when the BIOS does not support certain functions.

6.2 Parameter Setting Procedure

6.2.1 Blue Items

By moving the cursor to items shown in shown in blue and pressing the Enter key, settings for that item will be displayed. The cursor can be moved with the cursors keys and the Tab key. Pressing Shift-Tab moves the cursor backwards. A single click of the left mouse button also calls up the item settings.

6.2.2 Selecting a Menu Choice

When a selection is shown, use the cursor keys to move the cursor and press the Space bar make the selection. A single click of the left mouse button also makes the selection. Selections shown in round brackets are single- choice selections (selected item marked by an asterisk). Selections shown in square brackets are multiple-choice selections (selected items marked by x).

6.2.3 Numeric Input

When blue arrows (← →) appear when the cursor is moved to an item, a numeric input can be made. (A setting such as text color corresponding to a number may also be shown.) Use the + and - keys to change the value. The numeric keys can also be used to directly input a value. If an out-of-range value is input, it will be adjusted automatically. Single-clicking the ← → blue arrows also changes the value.

6.2.4 Character Input

Some item fields allow character input. These fields cannot be set with the mouse. Use the keyboard to make the required input. The allowable character range can be shown by calling up context-sensitive help (with F1 or the right mouse button). Normally, the box is filled with spaces. After input, the remainder of the field will be padded with spaces.

6.3 Using BPX Files

6.3.1 What Are BPX Files

BPX files serve for setting special parameters such as LCD parameters or ATA Boot parameters. BPX files are text files which contain the values to be set in the BIOS, as well as comments. You can open such a file and check its contents.

Note

If a BPX file contains an improper value, the BIOS will not operate normally. Use BPX files as supplied by Seiko Epson.

6.3.2 Displaying a BPX file

Use "Script list" from the File menu (Alt-F + L) to call up a list of script files. Files with extension BPX in the current directory as well as in directories in the path will also be shown. Valid BPX files will be shown with a comment on the right side.

Detail information is shown by F1 key or right-clicking of the mouse.

Note

When left-clicking on a BPX file with the mouse, the file will be loaded.

A confirmation dialog box appears, and the carton file will be overwritten if OK is selected. It cannot be returned to its former state. To only show the file information, use the right mouse button.

6.3.3 Loading a BPX File

Pressing the Enter key while a BPX file is selected or single-clicking the left mouse button will load the BPX file. A confirmation dialog box appears, and the carton file will be overwritten if OK is selected. Proceed with care, because the carton file cannot be returned to its former state.

Note

If the parameter table of the BPX file and the parameter table of the carton file are different, an error occurs.

6.4 RAKDA Functions

6.4.1 File I/O Functions

These functions serve to open carton files and script files. Selecting "Open Script File" or "Open Carton File" calls up a dialog box with file and directory fields and a file name entry field. Use the Tab key to move among the fields. Within a field, the cursor keys, space key (select) and Enter key (confirm) can be used. Finally, use the Tab key to move to the OK button and press the Enter key to open the selected file. Double-clicking the left mouse button on a directory or file name also is possible.

Note

The length of the directory tree that can be shown is limited. If directories are nested several levels deep, enter the file name directly in the file name field, or specify it as a command line argument when starting RAKDA.EXE.

Simply changing the parameters will not be reflected in the carton file (except for loading BPX files). To write the changes to the carton file, select "Write parameters" from the File menu. If no parameters have been changed, this selection will be grayed out. Once parameters have been written to the carton file, Undo can no longer be used to undo the changes.

When attempting to terminate the program without saving changes, a confirmation dialog box will appear.

6.4.2 Undo Function

The Edit menu offers two kinds of undo functions.

- Cancel changes in the page

All changes made on the current page will be undone. After changing to another page, page undo is no longer possible.

- Cancel all changes

Cancels all changes that have been made since the start of editing. This returns the carton file to the last saved state created with the "Write parameters" function.

6.4.3 View Function

When "Page list" on the View menu is selected, all selections in a script file are shown. Page divisions and comments are excluded, but normal key and mouse operations are possible.

6.4.4 Back Function

Using the Back menu item, you can return to previously displayed pages (history function). Script file hierarchy is disregarded.

6.4.5 Help Function

Selecting the Help menu item calls up a help display. Selecting the item again turns help off. Pressing F1 or the right mouse button when the cursor is located at a certain item calls up context-sensitive help for that item. (Some items may have no help screen.)

6.5 RAKDA Specifications

6.5.1 Operating Environment

- RAKDA.EXE is a DOS program.
- RAKDA.EXE runs under English and Japanese versions of DOS.
- RAKDA.EXE runs in a DOS box under Microsoft® Windows®.
- RAKDA.EXE supports mouse operation.

BPX files (extension BPX) and help files (RAKDAHJ.TXT, RAKDAHE.TXT) must be located in the same directory as RAKDA.EXE or in a directory included in the path (specified with the DOS environment variable PATH).

6.5.2 Command Format

From the DOS prompt, enter

RAKDA [script_file] [carton_file]

to start the program.

[script_file]: Script file name (drive and path can be included)

[carton_file]: Carton file name (drive and path can be included)

The carton file and script file can be omitted. In this case, the file will be selected from the File menu.

When started from the English mode of DOS, messages will be displayed in English and the help file RAKDAHE.TXT will be used.

When started from the Japanese mode of DOS, messages will be displayed in Japanese and the help file RAKDAHJ.TXT will be used.

6.5.3 File Name Argument

The length of a file name that can be specified as a command line argument is limited. If the file is not opened properly, shorten its path name by placing it in a directory that is not nested very deep.

6.5.4 Parameter Table Set Version (TSV)

The parameter Table Set Version (TSV) is defined separately from the BIOS version and the script file version.

Even if BIOS and script file versions differ, normal operation will be possible if the TSV is the same. If the TSV is different, certain items and BIOS parameters cannot be set. Use script files with matching TSV.

6.5.5 Script Files

The extension SCX denotes a binary script file. Such a file contains menu selections for various settings. If the parameter table set version (TSV) that is defined separately from the BIOS version and script file version is identical, a script file can be used for differing BIOS versions and CARD-PC models.

6.5.6 BPX Files

These files contain special settings such as LCD parameters and ATA Boot parameters. A BPX file contains values to be set in the BIOS, as well as comments. You can open such a file and check its contents.

The parameter table version of the BPX file and the carton file must match. Files with different versions cannot be used together.

Note

If a BPX file contains an improper value, the BIOS will not operate normally. Use BPX files supplied by Seiko Epson.

6.5.7 Error Messages

Failed to open the ooo file xxx

Explanation

- ooo is either "carton", "BIOS parameter", or "script".
- xxx is the file name.

Countermeasure

- Verify the file name.
- Place the file in a directory that is not nested too deep.

ooo file error

Explanation

- ooo is either "Carton", "BIOS parameter", or "Script".
- File data are corrupted.

Countermeasure

- Use a valid file.

Invalid ooo file xxx**Explanation**

- ooo is either "carton", "BIOS parameter", or "script".
- File data are corrupted.

Countermeasure

- Use a valid file.

No carton file to write parameter is opened**Countermeasure**

- Open a carton file.

No parameter table is found in carton file**Explanation**

- An attempt was made to set parameters for a non-existent BIOS module.
- Appears for example when attempting to set LCD parameters for a carton file from which the VGA BIOS has been deleted.

Countermeasure

- Use a valid carton file.

BIOS parameter file is not supported**Countermeasure**

- Use a valid BPX file.

BIOS parameter file loading error**Explanation**

- An error has occurred while loading a BPX file.
- This can occur if the network connection went down while loading a file or if the file is corrupted.

Format error in BIOS parameter file**Countermeasure**

- Use a valid BPX file.

Incorrect BIOS parameter file**Countermeasure**

- Use a valid BPX file.

Script error**Countermeasure**

- Use a valid script file.

Script file not appropriate for carton file. Some parameters may be incorrectly set.

Explanation

- The parameter table set version (TSV) of the carton file and script file differ.

Countermeasure

- Open a script file with matching TSV.

Help file not found

Explanation

The help file that is to be called up with Alt-H is not located in the same directory as RAKDA.EXE and is not in the DOS path.

Countermeasure

- Place the files RAKDAHJ.TXT (Japanese) and RAKDAHE.TXT (English) in the same directory as RAKDA.EXE.

Date data corrected

Explanation

An invalid value was entered in the date input field, and was corrected automatically.

Time data corrected

Explanation

An invalid value was entered in the time input field, and was corrected automatically.

No memory to save xxx

Explanation

There is not enough conventional memory.

Countermeasure

- Refer to the documentation of the OS.

7. Command Descriptions

This chapter describes the RAK commands included in the RAKDA directory.

7.1 BMCARTON

This changes the carton file information.

Syntax

```
BMCARTON carton_file [/S:sssss] [/C:cc] [/R:rrr] [/NM /EM]
```

Parameters

carton_file

This is the carton file name.

/S

This changes the data block start address. The data block will be located after this address. After this address there must be just enough space that all binary files to be saved as BootBlock (data is not compressed) and the data block can be saved.

If this parameter is omitted, no change is made to the start address of the data block in the current carton file.

/C

This specifies the compression method to be used for the data block. This can be selected from 'NC' (no compression), 'RL' (run-length), and 'SD' (self-dictionary). When a carton file is shipped, the default is self-dictionary. In general, self-dictionary provides a higher compression ratio than run-length (i.e. the compressed file is smaller).

/R

This specifies the size of the flash ROM built into the CARD-PC. Specify 256. If this parameter is omitted, the flash ROM size is not changed.

/EM, /NM

This specifies whether the carton file can be modified. Specify one of /EM and /NM.

If /EM is specified, the carton file can be modified. This is the default setting when the carton file is shipped.

If /NM is specified, the carton file can no longer be modified.

7.2 BMADD

This adds a binary image to the carton file.

Syntax

```
BMADD carton_file binary_file [packet_name] [/U] [/S /M /R /B] [/A:aaaaa] [/CR] [/NR]
[/NU] [/F:fffff] [/T:ttttt]
```

Parameters

carton_file

This is the carton file name.

binary_file

This is the binary file name. The contents of this file are stored in the carton file.

packet_name

This is the packet name. This name is recorded in the carton file, for use when manipulation with the BMADD, BMDEL, and BMLIST commands. The name must be shorter than 13 characters.

If this parameter is omitted, the part of the file name being "binary_name" is used as the packet name.

/U

The /U flag specifies update mode. When this flag is specified, an existing binary image in the carton file with the same packet name is replaced by the new binary image. If this flag is omitted, then if there is an existing binary image in the carton file with the same packet name, BMADD produces an error message, and does not add the new packet to the carton file.

/S, /M, /R, /B

These flags specify the type of the binary image.

Specify the /S flag when the binary image is to be copied to the system shadow RAM at execution time. A binary image with this flag set is compressed in the ROM, and thus makes efficient use of the available ROM capacity. Since it is copied to shadow RAM, it also executes faster.

Specify the /M flag when the binary image is Minimizer. When this flag is set, the image is copied to system SM-RAM, and is used as an SMI handler.

Specify the /R flag when the binary image is rare data. The binary image remains in flash ROM. The data is not compressed, and the binary image does not make use of shadow RAM.

Specify the /B flag when the binary image is BootBlock. This binary image is located at the end of the ROM.

When no flag is specified, the /S option is selected.

/A

(When the /S option is specified)

This parameter specifies the start address of shadow RAM as an absolute address.

The start address of shadow RAM must be on a 2 KB boundary. The shadow RAM is created at this address, and the binary image is copied to the shadow RAM.

The range of values that can be used for shadow RAM is from 0C0000h to 0FFFFFFh.

The memory address at execution time is also from 0C0000h to 0FFFFFFh. It is not possible to locate shadow RAM at an address above 100000h or below 0BFFFFh.

There is no default value. When the /S parameter is specified, this parameter must always be specified.

(When the /M option is specified)

This parameter specifies the absolute address of DRAM used as SM-RAM.

The default value is 0A0000h.

(When the /B option is specified)

This is ignored.

(When the /R option is specified)

This parameter specifies the absolute address at which the binary image is to be located.

To locate the binary image in the range 0E0000h to 0FFFFFFh, write this address directly. To locate the binary image in the range 0FC0000h to 0FDFFFFh, write the least significant five digits (for example, for 'FC0000' write 'C0000'). It is not possible to write the binary image at an address at which the BIOS-ROM does not exist. For details, refer to memory map in the BIOS reference manual.

In both of the above cases the address is specified in hexadecimal. The final 'h' is not required.

/CR

(When the /S option is specified)

By means of this parameter it is possible to direct BootBlock to check whether there is an option ROM on the ISA bus before creating the shadow RAM. If there is an option ROM at the address of the binary image, the shadow RAM is not created, and the binary image is not copied.

(When the /S option is not specified)

This is ignored.

/NR

This parameter makes the binary image non-removable from the carton file.

/NU

This parameter makes the binary image non-updatable.

/F

Specify the /F parameter when the binary image to be added to the carton file does not start from the start address of the binary file.

The value 'ffff' is in hexadecimal, and has no final 'h'. A value of '0' indicates that the binary image does start from the start address of the binary file.

The default value is zero.

/T

Specify the /T parameter when the binary image to be added to the carton file does not end at the last address of the binary file.

The value 'tttt' is in hexadecimal, and has no final 'h'.

The default value is the last address of the binary file.

7.3 BMDEL

This deletes a packet from the carton file.

Syntax

BMDEL carton_file packet_name

Parameters

carton_file

This is the carton file name.

packet_name

This is the name of the packet to be deleted.

If the packet was added to the carton file with the BMADD parameter /NR specified, before deleting it, BMDEL asks for confirmation.

7.4 BMLIST

This lists all of the packets in the carton file.

Syntax

BMLIST carton_file

Parameters

carton_file

This is the carton file name.

Display

Standard information

The first five lines of information are the same for all packet types.

Example (shadow RAM packet)

Shadow-RAM data packet Packet name = "CBIOS" Remove = permitted Update = permitted Binary image size = 64 Kbytes
--

The first line indicates the data packet type.

The second line indicates the data packet name.

The third line indicates whether or not the data packet can be removed.

The fourth line indicates whether or not the data packet can be updated.

The fifth line indicates the data packet size.

Information display for a carton control data packet

```
Carton control data packet
  Packet name = "PacketControl"
  Remove = inhibited
  Update = permitted
  Binary image size = 0 Kbytes
  Data block start address = e8000
  ROM size = 256 Kbytes
  Data compression = Self-Dictionary
```

Information display for a BootBlock data packet

```
BootBlock data packet
  Packet name = "BootBlock"
  Remove =permitted
  Update = permitted
  Binary image size = 8 Kbytes
  Location address = fe000
  Version number =3.0
```

Information display for a shadow RAM data packet

```
Shadow-RAM data packet
  Packet name = "VGA"
  Remove = permitted
  Update = permitted
  Binary image size = 32 Kbytes
  Shadow-RAM place = c0000
  ROM check = Yes
```

Information display for a Minimizer data packet

```
Minimizer data packet
  Packet name = "MINIMIZE.BIN"
  Remove = permitted
  Update = permitted
  Binary image size = 64 Kbytes
  Physical SM-RAM = a0000
```

Information display for a rare data packet

```
Rare data packet
  Packet name = "Raredata"
  Remove = permitted
  Update = permitted
  Binary image size = 32 Kbytes
  Location address = fc8000
```

7.5 BMMAKE

This creates a ROM image from the carton file.

Syntax

```
BMMAKE carton_file ROM_image_file
```

Parameters

carton_file

This is the carton file name.

ROM_image_file

'ROM_image_file' is the name of the ROM image file to be created.

This command creates a BIOS ROM image. The ROM image file contains a BIOS binary image to be written to flash ROM.

7.6 BMPAR

With respect to binary images saved in a carton file by BMADD, BMPAR changes the binary image saving information saved in the carton file.

Syntax

```
BMPAR carton_file packet_name [/A:aaaaa]
```

Parameters

carton_file

This is the carton file name.

packet_name

This is the name of the packet saved in the carton file. It is the name displayed after the legend "Packet name =" when BMLIST is executed.

/A

(When the specified data packet is a shadow RAM or rare data packet)

Specifies the start address of shadow RAM.

The range of values that can be used for shadow RAM is from 0C0000h to 0FFFFFFh.

(When the specified data packet is a Minimizer data packet)

This parameter specifies the absolute address of DRAM used as SM-RAM.

(When the specified data packet is a BootBlock data packet)

This parameter is ignored.

7.7 PCOPY

This copies the contents of parameter tables from one carton file to another. If the BIOS versions of the source and destination files are different, some parameters may not be set, or settings may be lost.

Syntax

```
PCOPY source_file destination_file
```

Parameters

source_file

This is the filename of the source file used for copying the parameters.

destination_file

This is the filename of the destination file used for copying the parameters.

7.8 Error Messages

The error messages are common to the following commands:

- BMCARTON.EXE
- BMADD.EXE
- BMPAR.EXE
- BMDEL.EXE
- BMLIST.EXE
- BMMAKE.EXE

-/F option has invalid value.

The start address of the binary image within the file exceeds the file size. Change the start address of the binary image so that it does not exceed the file size.

-/T option has invalid value.

The end address of the binary image within the file exceeds the file size. Change the end address of the binary image so that it does not exceed the file size.

-"carton_file_name" is not found.

The specified carton file cannot be found. Specify a carton file that exists.

-Address of data in execution time is conflicted.

"packet_name1" and "packet_name2".

The addresses of packets packet_name1 and packet_name2 in shadow RAM conflict, and there is a possibility that the CARD-PC may not operate correctly. Change the location addresses so that they do not overlap.

-Already the packet "packet_name" is registered.

The specified packet is already present in the carton file. To update the packet, specify the /U flag.

-Carton control packet is not found.

The carton control packet does not exist. This carton file is corrupted. Either use a backup, or copy a carton file from the RAK, and repeat the operation.

-Control data is lost.

The carton control packet does not exist. This carton file is corrupted. Either use a backup, or copy a carton file from the RAK, and repeat the operation.

-Data block conflict is found.

"packet_name1" and "packet_name2".

The addresses of packets packet_name1 and packet_name2 in shadow RAM conflict, and there is a possibility that the CARD-PC may not operate correctly. Change the location addresses so that they do not overlap.

-Data file "binary_file_name" is not found.

The specified binary file does not exist. Specify the correct filename.

-Failed to create a new carton file "carton_file_name"

A carton file of the specified name could not be created. The cause may be that the disk is write-protected, that there is not enough space, or that there are no empty directory entries. Eliminate these problems, then repeat the operation.

-Failed to make backup file.

A backup file could not be created. The cause may be that the disk is write-protected, or that there are no empty directory entries. Eliminate these problems, then repeat the operation.

-Failed to open the target file "ROM_image_file".

An attempt to create the ROM image file failed. The cause may be that the disk is write-protected, or that there are no empty directory entries. Eliminate these problems, then repeat the operation.

-Failed to write on the carton file.

An attempt to write to the carton file failed. The cause may be that the disk is write-protected, or that there is not enough space. Eliminate these problems, then repeat the operation.

-Failed to write on temporary file.

An attempt to write to a temporary file failed. The cause may be that the disk is write-protected, or that there is not enough space. Eliminate these problems, then repeat the operation.

-Invalid compression method is specified

The compression method specification is invalid. The valid specifications are 'NC' (no compression), 'RL' (run-length), and 'SD' (self-dictionary).

-Invalid file name (extension cannot be BAK).

The carton file name is invalid. Use a carton file name with an extension other than .BAK.

-Invalid file name (only extension is found).

The carton file name is invalid. The carton file name may not consist of an extension alone.

-Minimizer data packet found while no data block area is taken.

To use a minimizer packet, you need data block area.

This error message appears when there is no data block area, but a Minimizer data packet is found. To use a minimizer packet, there must be a data block area.

-No binary file name specified.

There is no binary file name specified. Specify a binary file name.

-No carton file name specified.

There is no carton file name specified. Specify a carton file name.

-No data block area is taken.

To use a BootBlock, you need data block area.

This error message appears when there is no data block area, but a BootBlock is found in the carton file. To use a BootBlock, there must be a data block area.

-No file name

The carton file name is invalid. The carton file name may not consist of an extension alone.

-No memory for buffer to place binary image

Sufficient memory to create the binary image file could not be obtained. Use some method, such as eliminating TSRs or device drivers, to increase the amount of available memory, then repeat the operation.

-No memory for data compression

Sufficient memory to compress the data block could not be obtained. Use some method, such as eliminating TSRs or device drivers, to increase the amount of available memory, then repeat the operation.

-No memory to remove a packet.

Sufficient memory to delete or update a packet could not be obtained. Use some method, such as eliminating TSRs or device drivers, to increase the amount of available memory, then repeat the operation.

-No memory to make backup file.

Sufficient memory to create a backup file could not be obtained. Use some method, such as eliminating TSRs or device drivers, to increase the amount of available memory, then repeat the operation.

-No modification is allowed on the carton.

This carton file is marked as non-updatable.

Use BMCARTON to permit updates on the carton file, then repeat the operation.

-No packet name specified

No packet name was specified. Specify the packet on which the parameter adjustment is to be carried out.

-Shadow RAM data packet found while no data block area is taken.

To use a shadow RAM data packet, you need data block area.

This error message appears when there is no data block area, but a shadow RAM data packet is found. To use a shadow RAM packet, there must be a data block area.

-Starting address must be set at paragraph boundary.

The start address of the data block must be on a paragraph boundary. Specify an address of which the least significant digit is 0h.

-The carton file "carton_file_name" is not found.

Confirm file name or create new carton file by BMCARTON.EXE.

The specified carton file does not exist. Specify a carton file that exists.

-The packet "packet_name" is not found.

The specified packet does not exist in the carton file. Specify a packet that exists.

-The packet "packet_name" is not removable. Command is canceled.

The specified packet is not removable. This packet cannot be removed.

-The packet is not updatable.

The specified packet is marked as not updatable. The parameters in this packet cannot be updated.

-The rare data block "packet_name" is outside the ROM area.

The rare data block packet_name is located at an address at which the ROM is not present. Change the rare data block size or location address so that it is completely contained within the ROM.

-The rare data block "packet_name" overruns into data block area.

The rare data block packet_name extends into the data block area. Make the rare data block smaller or change the start address, then repeat the operation.

**-The SM-RAM physical address is not at 64 KB boundary.
It is corrected to x0000.**

The SM-RAM physical memory address is not on a 64 KB boundary. In this case the physical memory address is forcibly corrected to be x0000h, and the ROM image file is created.

**-The starting address is not at 2 KB boundary.
It is corrected to xxxxxh.**

The shadow RAM address at execution time is not on a 2 KB boundary. In this case it is forcibly corrected to be on a 2 KB boundary, and the ROM image file is created.

-This compression method is not supported

The specified compression method is not supported. Specify a correct compression method.

-Too small data block area.

The data block area is too small, and part of the data block will not fit in the ROM image file. Make the data block area larger, and repeat the operation.

-Update on the packet is inhibited.

The specified packet is marked as not updatable.

This packet cannot be updated.

-Unknown BootBlock version

The BootBlock version number is invalid.

This carton file is corrupted. Either use a backup, or copy a carton file from the RAK, and repeat the operation.

-You need BootBlock to use shadow-RAM or SM-RAM.

This error message appears only when there is no BootBlock, but a shadow RAM packet or Minimizer packet is present. To use a shadow RAM packet or Minimizer packet, there must be a BootBlock present in the carton file.

7.9 WFLASH

This command rewrites the flash ROM in the CARD-PC. It can be executed on CARD-PC products such as the evaluation kit and CardPresso.

Syntax

WFLASH binary_file [/smout] [/k] [/y|n]

Parameters

binary_file

The name of the ROM image file to be written to the flash ROM in the CARD-PC.

/smout

Specifies the SMOUT number. If omitted, "3" is used. Because WFLASH controls the 12V power supply line when writing to the flash ROM, the SMOUT signal is required. When using the evaluation kit, set jumper 21 to ON to connect SMOUT3 to the PGM signal. Take care that the SMOUT number specified here does not conflict with a SMOUT number used for HDD or serial port power management.

/k

Use this parameter when the CMOS-RAM should not be rewritten. If the parameter is omitted, the values specified in the default CMOS table will be written to the CMOS-RAM. Normally, this parameter should be omitted.

/y, /n

y: Perform a non-conditional reset when WFLASH terminates.

n: Perform a non-conditional return to DOS when WFLASH terminates.

If the parameter is omitted, a message appears and reset or return to DOS is selected from the keyboard.

7.10 RAKDA

Serves to change various CARD-PC BIOS parameters.

Syntax

RAKDA script_file carton_file

Parameters

script_file

The name of the RAKDA script file

carton_file

The name of the carton file whose parameters are changed

8. Implementation Example

8.1 Adding an OPTION ROM image

An option ROM program is located during startup in the interval between addresses C0000h and EFFFFh. For the CARD-PC there are two ways of adding an option ROM program.

- Adding an option ROM program in the flash ROM built into the CARD-PC
- Adding an option ROM program on the ISA bus

Descriptions of each of these follow.

8.1.1 Adding an option ROM program in the flash ROM built into the CARD-PC

When adding an option ROM program in the flash ROM built into the CARD-PC, use BMADD. If the option ROM program name is USER.BIN,

```
C:\RAK>bmadd carton_file_name user.bin /a:d8000 /s
```

maps to address D8000h. Note the following points.

- **Do not specify the /R flag**

If the /R flag is specified, this is added as a rare data packet. Since a rare data packet is mapped to an address of FC0000h or above, it cannot be used as an option ROM. Basically the /S flag should be specified, but the /S flag can be omitted.

- **Specify an address with the /A parameter on a 2 KB boundary**

The address specified with the /A parameter must be on a 2 KB boundary. In other words, the last 3 digits of the address value must be 000 or 800. This is part of the IBM PC/AT compatible BIOS specification.

- **/CR flag for external ROM priority**

Specifying the /CR flag gives priority to an option ROM on the ISA bus. In other words, at startup, before creating the shadow RAM, a check is made on the ISA bus for an option ROM, and if one is found the option ROM, the shadow RAM is not created, and the image is not copied. At this address the ROM can be seen on the ISA bus. Even if the /CR flag is specified, if there is no external option ROM, the program added with BMADD can be used.

Next, the BIOS ROM image is created.

C:\RAK>bmmake carton_file_name binary_file_name

The ROM image called "test.bio" is now completed. If the image cannot be created, one of the following error messages appears. Take countermeasures as indicated.

- Address of data in execution time is conflicted.

Explanation

There is a conflict in packet addresses.

Countermeasure

Use BMLIST to check and BMPAR to change the address.

- Rare data block "XXX" overruns into data block area.

Explanation

The rare data block XXX overruns into the data block area.

Countermeasure

Use BMLIST to check the data block start address. Specify a higher address with BMCARTON /S:. This can be performed automatically with ROMADD.

- Data block area too small

- Data block conflict: "XXX" and "BootBlock"

Explanation

The data block overruns into the boot block area.

Countermeasure

Because the data block is compressed, actual size cannot be predicted.

Use BMMAKE and check the "xxxx bytes free in Data Block area" message.

If there is not enough room, delete a packet from the data block area.

Make the start address of the data block as low as possible. This can be performed automatically with ROMADD.

8.1.2 Adding an option ROM program on the ISA bus

When adding an option ROM program on the ISA bus, no particular settings are required in the BIOS. However, if using BMADD to add the binary to the carton file, note the following points.

- **Assignment of addresses C0000 - DFFFFh to shadow RAM / ISA bus in 32 KB units**

The assignment of addresses in the range C0000 to DFFFFh either to shadow RAM or to the ISA bus is in 32 KB units. For example, executing

```
C:\RAK>bmadd carton file name user.bin "User_Pro" /a:d0010 /u
```

means that even if the size of USER.BIN is 1 byte, addresses D0000 to D7FFFh will be shadow RAM, and will not be available to the ISA bus.

- **/CR flag for option ROM priority on the ISA bus**

Specifying the /CR flag gives priority to an option ROM on the ISA bus. When the /CR flag is specified for all shadow RAM packets mapped to a 32 KB area, then if an option ROM is added in that area, the option ROM is used.

Example: VGA BIOS

The VGA BIOS occupies addresses C0000 to C7FFFh, and to BMADD, /CR is specified. Therefore, inserting a VGA card on the ISA bus means that the VGA BIOS ROM on the VGA card is mapped to addresses C0000 to C7FFFh, and the VGA BIOS built into the CARD-PC is not used.

- **No relation between rare data packets and option ROMs**

When the /R flag is specified, the data is added as a rare data packet. Since the rare data packet is mapped to an address above FC0000h, it can be used without any relation for option ROMs.

Example: SETUP and VGA BIOS

The SETUP and VGA BIOS have been added to the carton file with the following settings.

```
C:\RAK>bmadd carton file name vga8100 "VGA" /a:c0000 /s /cr
```

```
C:\RAK>bmadd carton file name setup "Setup" /a:c0000 /r
```

Note the underlined parameters. At first glance it appears that the addresses coincide, but the VGA is mapped to C0000h, and SETUP is mapped to FC0000h.

8.1.3 Adding a VGA BIOS from another manufacturer

The CARD-PC is shipped with a BIOS including a VGA BIOS produced by Epson. Therefore, when using a VGA controller from another manufacturer on the ISA bus, it is necessary to use the VGA BIOS specific to that controller. Normally a VGA BIOS is included on the VGA board, and using that VGA BIOS enables the VGA display to be used. On the other hand, it is possible to include the specific required VGA BIOS in the flash ROM built into the CARD-PC. Using this function means that it is not necessary to provide a separate VGA BIOS ROM, and thus allows costs to be reduced.

Setting procedure

First remove the VGA BIOS from the carton file as shown below. The method of removing the VGA BIOS is as follows. For the carton file name below, use the name of the file you have to hand.

```
C:\RAK>bmdel carton file name vga8110
```

Next add the other manufacturer's VGA BIOS. The VGA BIOS is located according to the specification at 0C0000h. If the binary filename of the VGA BIOS to be added is VGA.BIN, the following is the method of adding it.

```
C:\RAK>bmadd carton file name vga.bin /a:c0000
```

Next create the ROM image. In the following command line, the filename ROM.BIN is specified for the ROM image file, but any filename may be specified.

```
C:\RAK>bmmake carton file name rom.bin
```

This command creates a ROM image file with the name ROM.BIN. Write this file with WFLASH or ROM Writer to the CARD-PC.

8.1.4 Increasing the Flash ROM Free Area

As ROM version operating systems, option ROM programs, data, and so forth, are added to the flash ROM built into the CARD-PC, as additions are made to the standard carton file, it may become full. In this case, try the following settings.

Eliminating unneeded modules

If there are any unneeded modules in the standard carton file, delete them. Note that BootBlock cannot be deleted.

Deleting SETUP

If ROM SETUP, which is called by pressing the F10 key during powering on, is not required, delete it.

```
C:\RAK>bmdel carton file name setup
```

It is then necessary to set the data block start address to an optimal value.

Deleting VGA-BIOS

If a video card is added on the ISA bus, the built-in VGA BIOS can be deleted.

```
C:\RAK>bmdel carton file name vga8110
```

Deleting CBIOS

If this is deleted the CARD-PC will not function. If for some special application it is required to be deleted, custom BIOS support will be needed.

8.1.5 Changing the method of compression

Shadow RAM data is compressed within the ROM image. Using the /C parameter of the BMCARTON command, it is possible to select the compression method.

- /C:NC ... no compression
- /C:RL ... run-length compression
- /C:SD ... self-dictionary compression

Of these the self-dictionary compression method gives the highest degree of compression. BMLIST can be used to check which compression method is selected.

8.1.6 Using rare data packets

Since shadow RAM data packets are compressed in the ROM image, it appears at first sight that more data can be fitted in the flash ROM. However, since in practice the shadow RAM data packets are mapped into the range of addresses C0000 to EFFFFh, less data can be added as shadow RAM data packets than there is free space in the flash ROM. This waste can be avoided by the use of rare data packets. Since rare data is mapped into the address range FC0000h and higher, it can be used independently of the shadow RAM. For example, SETUP and VGA BIOS can be added to the carton file as follows.

```
C:\RAK>bmadd carton file name vga8100 "VGA" /a:C0000 /s /cr
```

```
C:\RAK>bmadd carton file name setup "Setup" /a:c0000 /r
```

At first glance it appears that the addresses coincide, but the VGA is mapped to C0000h, and SETUP is mapped to FC0000h. It is necessary whenever using a rare data packet to be sure to enter an appropriate data block start address.

Notes on the use of rare data

Since rare data is mapped into the address range FC0000h and higher, the following points should be noted.

[When storing program]

The program must be written as in protected mode.

Since the code is located in ROM on the ISA bus, the execution speed is reduced. CPU caching is also not valid.

[When storing data]

Reading of the data must be carried out, for example:

- By using a program in protected mode
- By using a BIOS service call INT 15h, AH=87h (memory block transfer) in real mode
- By use as a ROM disk in a ROM version operating system

8.1.7 Setting an optimum data block start address

When packets are added or deleted, it is necessary to reset the data block start address to an appropriate value. The data block start address refers to the dividing line between the area holding rare data packets and the area holding shadow RAM data packets.

Determine the optimum data block start address as follows.

- Compute the last address of rare data.
- Subtract F00000h from this address.
- Round up the last digit of this address expressed in hexadecimal.

Example: When the rare data comprises SETUP only

First suppose the SETUP start address is FC0000h.

```
C:\RAK>bmpar carton file name setup /a:c0000
```

Note

When specifying addresses with BMPAR, for rare data the F in the sixth digit (from the right) is not written.

Next use BMLIST to check the SETUP size.

```
C:\RAK>bmlist carton file name | more
```

In this example, since it is 25362 KB in decimal, this is 6312h in hexadecimal. Thus:

$$\text{FC0000h} + 6312\text{h} - \text{F00000h} = \text{C6312}$$

Rounding this up, the optimum value is C6320.

```
C:\RAK>bmcarton carton file name /s:c6320
```

8.1.8 Using CMOS-RAM for user applications

The CARD-PC BIOS provides part of the CMOS-RAM for use by users.

CMOS-RAM user area: addresses 70h to 7Fh

For example, using a ROM version operating system which looks at specific values in CMOS-RAM to select the boot device, the above CMOS addresses can be used. Note that even if these values are changed, it is not necessary to modify the CMOS checksum value. When the system starts up initially, or when it starts up with the CMOS-RAM battery not connected, the values in the Default CMOS table are used. These values are stored in the flash ROM built into the CARD-PC, and can be set using the RAKDA.

9. Packet Installation Utility (ROMADD.EXE)

This utility serves allows easy installation of binary files (option ROM, MS-DOS® ROM version etc.) into the BIOS carton file. ROMADD.EXE uses script files to add or delete binary files, automatically set the data block start address, and to perform other functions.

This section describes how to use ROMADD.EXE and its script files.

9.1 ROMADD.EXE

Functions

The utility operates according to instructions in the specified script file and changes the contents of the carton file. The following changes can be made:

- Add data (address can be specified)
- Delete data
- Change data block start address
- Change ROM setup address

Syntax 1

ROMADD carton file name script file name

Carton file name

Specifies the RAK carton file name including the extension.

Script file name

Specifies the script file name (see below) including the extension.

Syntax 2

ROMADD /h

Displays help regarding the use of ROMADD and script file syntax.

9.2 ROMADD.EXE Script Files

Script files contain instructions for the operation of ROMADD. These script files are regular DOS text files and can be edited with a text editor. The syntax of a script file is shown below.

; (semicolon)

Any string following a semicolon is treated as a comment.

ADD file name address BMADD.EXE option

Adds the data in the specified file to the carton file. (To do this, BMADD.EXE of the RAK is called.)

The address for placing data must be a physical address, specified in hex notation. Allowable ranges are C0000 - FFFFF and FC0000 - FFFFFFF.

When C0000 - FFFFF is specified, the data are compressed and placed in the BIOS ROM in the CARD-PC. At system startup, these data will be uncompressed and placed in shadow RAM.

When FC0000 - FFFFFFF is specified, the data are not compressed and placed in the BIOS ROM in the CARD-PC. This is the same as executing BMADD with the /R option.

The options for BMADD.EXE are the same as for BMADD that is part of RAK.

For example, when wishing to overwrite data already present in the carton file, use the /U option. The name (packet name) of data in the carton file can be specified within double quotes. If the data name is omitted, the file name is used as data name.

DELETE data packet name

Deletes data from a carton file. The data to be deleted are specified by the data packet name. If the specified data do not exist, an error message is displayed and no action is taken.

SET_DBS

When this instruction is included in a script file, the data block start address is set automatically. The address is always chosen so that the rare data area is maximized.

Script file example

```

:-----:
; Script File for ROMADD.EXE          ;
; SEIKO EPSON CORPORATION, 1996      ;
; EMS AVAILABLE                       ;
:-----:
: action      module      address  packet_name  option
:-----:
ADD          E000.BIN     E0000   "MS-DOS5"    /U
ADD          E800.BIN     E8000   "MS-DOS2"    /U
ADD          FC00.BIN     FC0000  "COMMAND"    /U
ADD          FD00.RMD     FD0000  "ROMDISK"    /U
; DELETE     MINIMIZE.BIN
; DELETE     SETUP
SET_DBS

```


9.3 ROMADD.EXE Operation

The following is a simple step-by-step description of the ROMADD execution flow.

1. Data specified in the script file with ADD are added using BMADD.
2. Data specified in the script file with DELETE are deleted using BMDEL.
3. A temporary carton file is created which consists of the file resulting from step 2 minus the rare data.
4. The temporary carton file is run through BMMAKE to determine the data block area size. This results in a temporary binary file.
5. If SET_DBS is specified in the script file, the data block start address is set to the point immediately before the data block area (to maximize the rare data area).
6. If there is a packet name starting with "SETUP" in the carton file, this is moved to an optimized location within the rare data area, with at least the size of setup free from the 2KB boundary address. If there are several possible locations, the one with the lowest address is used.
7. The last address of the rare data is compared to the data block start address and the difference is shown in a message.
8. The temporary carton file and binary file and the backup carton file etc. created by BMADD are deleted.

10. LFLASH.EXE

10.1 Introduction

LFLASH.EXE is a utility for manipulating the contents of local flash ROM in the CARD-486D4. Note that it cannot be used with other models of the CARD-PC or with a CARD-486D4 not equipped with local flash ROM.

10.2 Using LFLASH.EXE

LFLASH.EXE runs under the MS-DOS® (PC-DOS) operating system. The data (program or ROM-disk images) required when carrying out the local flash ROM operations must already be available in the form of MS-DOS® (PC-DOS) files. In this regard, refer to the respective development kits for how to create the DOS® or Windows® data.

When accessing the flash ROM, LFLASH.EXE is used in protected mode. Therefore, when it is running, if device drivers or other programs are running using protected mode, operation will not be correct. In this case, remove the other software, and restart the system.

10.3 LFLASH.EXE Functions

LFLASH.EXE provides functions for the following operations on local flash ROM in the CARD-486D4.

Write function

This function erases a section of the local flash ROM from the specified start address for the specified file size, and updates it. After writing the data, a verify operation is carried out.

Verify function

This function verifies a section of the local flash ROM from the specified start address, comparing the contents with the specified file.

Read function

This function reads data from the specified start address in local flash ROM for the specified file size. The data read is stored in the specified file.

Erase function

This function erases data from the section of the local flash ROM from the specified start address and of the specified size.

The format for using LFLASH.EXE is as follows.

Syntax

LFLASH filename [/A:address] R[:size]//V//W

or

LFLASH [/A:address] /C[:size]

Parameters

filename

Specify the name of the data file used for the local flash ROM operation.

/A:address

Specify the start address for the operation in flash ROM.

/C:size

Select the function to erase data from the local flash ROM. The size of the data to be erased can also be specified.

/R:size

Select the file save function for local flash ROM data. The size of the data to be read can also be specified.

/V

Select the verify function to compare the contents of local flash ROM with a file.

/W

Select the function to write the contents of the specified file to the local flash ROM.

The parameters should be separated by spaces. If no parameters at all are specified, a "Usage" message appears.

When specifying addresses and sizes, either hexadecimal or decimal can be used. The default is hexadecimal. To use decimal, append a period (.) to the end of the number.

Default values of the parameters are as follows:

When the /A parameter is omitted

The start address of the whole local flash ROM present is used.

When the /C parameter has no size specification

The area from the specified start address to the last address of local flash ROM present is used.

When the /R parameter has no size specification

The area from the specified start address to the last address of local flash ROM present is used.

10.4 How to Use LFLASH.EXE

LFLASH.EXE is a program for the CARD-486D4. It cannot be used with other models. Before using it, check that there are no drivers or programs running which use protected mode.

10.4.1 Writing to Flash ROM

- (1) Have ready the data file to be written to local flash ROM.
- (2) Start the CARD-486D4 system, and start DOS.
- (3) Run LFLASH.EXE with the name of the file to be written, the writing start address, and the /W flag specified.

Format: LFLASH filename /A:writing_start_address /W

10.4.2 Verifying the Data in Flash ROM

- (1) Have ready the file for verification.
- (2) Start the CARD-486D4 system, and start DOS.
- (3) Run LFLASH.EXE with the name of the file to be verified, the verification start address, and the /V flag specified.

Format: LFLASH filename /A:verify_start_address /V

10.4.3 Reading Data from Flash ROM

- (1) Start the CARD-486D4 system, and start DOS.
- (2) Run LFLASH.EXE with the name of the file for saving, the reading start address, and the /R parameter with size specification.

Format: LFLASH filename /A:save_start_address /R:save_size

10.4.4 Erasing Data in Flash ROM

- (1) Start the CARD-486D4 system, and start DOS.
- (2) Run LFLASH.EXE with the erasure start address, and the /C parameter with size specification.

Format: LFLASH /A:erase_start_address /C:erase_size

10.5 LFLASH.EXE Error Messages

When LFLASH.EXE detects an error during operation, it displays an error message on the screen. When such an error is detected, respond as shown in the following.

ERROR: This machine is not supported by this program

LFLASH.EXE can only run on a CARD-486D4. Check the system and configuration being used.

ERROR: No flash ROM or not supported flash ROM

A flash ROM supported by LFLASH.EXE cannot be found. Check that local flash ROM is installed, or check the processor type.

ERROR: XMS driver is running

Please remove it and reboot system

The XMS driver is running. Reboot the system, making sure the XMS driver is not running.

ERROR: Address parameter is invalid

The address value specified by the /A parameter is invalid. Check the value specified.

ERROR: Size parameter is invalid

The size value specified by the /C or /R parameter is invalid. Check the value specified.

ERROR: No operation switch

There is no operation switch (parameter to select function) specified. Specify an operation switch.

ERROR: File name is required

No filename is specified. Specify a filename.

ERROR: File name is not needed

A parameter specification does not require a filename.

ERROR: Invalid parameters are specified

The parameter specification is invalid in some way. Check the parameter specification.

ERROR: Insufficient memory

LFLASH.EXE failed to obtain the memory it requires for a work area. Check the operating configuration, and increase the amount of conventional memory available.

ERROR: File not found or failed to open

When /V or /W was specified, the specified file could not be found, or an error occurred while opening the file. Check the specified filename and drive name.

ERROR: File size is bigger than target flash ROM space

The file size specified with /V or /W is bigger than the target local flash ROM area. Check the access start address and specified file size.

ERROR: Invalid source file

The file specified with /V or /W is of zero size. Check the specified file.

ERROR: Failed to create destination file

When /R was specified, the target file could not be created. Check the target drive and specified filename.

ERROR: File read error found

When /V or /W was specified, an error occurred while reading the file. Check the target drive and contents of the file.

ERROR: File write error found

When /R was specified, an error occurred while writing the file. Check the target drive.

ERROR: Data verify error found

When /V or /W was specified, an error occurred while executing the verify function. Check the configuration settings.

ERROR: Memory release error found

When LFLASH.EXE tried to release an area of memory used for internal working, an error occurred. Check the contents of the AUTOEXEC.BAT and CONFIG.SYS files, and the operating configuration.

If other error messages appear, refer to Seiko Epson Corp. with details.

11. Glossary

BIOS

Abbreviation of Basic I/O Subsystem. Carries out basic hardware operations such as disk access and screen display. The CARD-PC BIOS is a Seiko Epson original BIOS designed to be compatible with an IBM PC/AT.

BIOS binary

See BIOS ROM image.

BIOS flash ROM

256KB of flash ROM designed for holding the CARD-PC BIOS. Access timing is the same as for the 8-bit ISA bus. Incorporated in all CARD-PC models.

See Local flash ROM

BIOS flash ROM write utility

See WFLASH

BIOS parameters

Since the BIOS manipulates the hardware directly, parts of it depend on the hardware. In order to reduce to a minimum the changes required when the hardware changes, the hardware-dependent portions are parameterized. Items such as the method of hardware testing, which also affect the operation of the BIOS, are here also termed BIOS parameters.

BIOS ROM image

Binary data to be written (or already written) to the flash ROM built into the CARD-PC. Created by BMAKE from a carton file. Written to the flash ROM in the CARD-PC using WFLASH or ROM WRITER. Also called a BIOS binary.

BootBlock

The first module executed when the system starts. It creates shadow RAM, and expands the binary images compressed in flash ROM to the correct addresses. Essential for the CARD-PC BIOS.

BPX file

Text file with BPX extension, containing special settings such as LCD parameters and ATA Boot parameters. Can be opened for checking and viewing comments. Cannot be used if parameter table of BPX file and TSV of carton file are different.

See TSV

CardPresso (SEK0630B)

An IBM PC/AT compatible motherboard using the CARD-PC standard. Dimensions conform to half-size PC/AT slot size (177 x 115 mm). Contains two PCMCIA slots and a PC/104 socket. Incorporates flash ROM on an ISA bus and can be used to configure a diskless system.

Carton file

A file forming the basis for a BIOS ROM image. The BIOS comprises a plurality of binary images, and this is a container for handling them as a single unit. This is an intermediate file with the binary images converted to special codes to allow the functions of the RAK to operate.

CBIOS

Module for providing IBM PC/AT compatibility. Also termed the "main" BIOS. Executes POD, option ROM initialization, and provides various BIOS services.

Data block

Collection of data compressed in the flash ROM. Everything except rare data and BootBlock. The area from the data block start address to immediately before BootBlock is termed the data block area.

Data block start address

The start address of the data block. Specified by BMCARTON /S:xxxxx. Must be on a paragraph boundary.

DBIOS

IBM PC/AT compatible BIOS developed for the Seiko Epson Falconer chip set. Characterized by low dependence on the CARD-PC model. See Falconer chip set

Evaluation kit (DIO5300)

Evaluation platform using the CARD-PC standard. Consists of an IDE interface (40-pin/68-pin), floppy drive interface (34-pin), CRT connector (VGA), parallel port, serial port x 2, PS/2 mouse interface, keyboard interface, LCD interface (20-pin, ISA bus slot x 3. Serves to evaluate all basic CARD-PC functions.

Falconer chip set

Chip set developed by Seiko Epson for mobile computing applications using the Intel 486™ SX-SF CPU (designed for the embedded type PC market). Features a CPU bus interface (SPC8210) and PC/AT compatible peripheral interface (SPC8220). Used in the CARD-PC series. Some CARD-PC models use an enhanced version of this chip set.

LFLASH

Utility for managing the flash ROM on CARD-PC models with local flash ROM (CARD-486D4 etc.).

See Local flash ROM

Local flash ROM

Flash ROM connected to local CPU bus, used in some CARD-PC models (CARD-486D4 etc., size 4MB in CARD-486D4). 32-bit bus width and memory cache capability result in higher access speed than expansion ROM connected via ISA bus. Separate from BIOS flash ROM.

See BIOS flash ROM

Local flash ROM write utility

See LFLASH

Minimizer

SMI handler that provides power management functions. SMI (System Management Interrupt) is a function of SL-Enhanced CPUs. The module also includes keyboard emulation functions and cannot be deleted.

Option ROM

A ROM located during startup in the interval between addresses C0000h and EFFFFh. This is basically an additional ROM provided at this address on the ISA bus for expanding the BIOS functions, but RAM (shadow RAM) at the same address is also termed an option ROM. VGA BIOS, SCSI BIOS, and so forth, are typical programs located in option ROM.

Packet

Constituent of the carton file. Comprises the binary image itself and packet attributes which record information about the packet.

Packet installation utility

Utility for adding or deleting binary files to and from the BIOS carton file. Operation is controlled by script files named "*.SCR". Data block start address is also set automatically.

Parameter table version

See TSV

POD

Power On Diagnostics. Hardware checks carried out at power-on. Carried out by CBIOS.

RAK

Acronym for ROM Adaptation Kit. Serves for modifying Seiko Epson BIOS parameters. Consists of BIOS, RAKDA, and other utility and parameter files.
See BIOS, RAKDA

RAKDA

Acronym for ROM Adaptation Kit for DBIOS Addjustment. Allows setting BIOS parameters in a Windows® like GUI environment.

RAKDA script file

See SCX file

Rare data

Data which is not copied to RAM and is used from the flash ROM.

ROMADD

See Packet installation utility

ROM WRITER

In this manual, this indicates ROM WRITER (DIO5302) for the CARD-PC. This is a device for rewriting the flash ROM for the BIOS built into the CARD-PC.

RTC

Abbreviation of Real Time Clock. Built into the CARD-PC, it allows the date and time to be set, then updates them automatically thereafter.

RTC/CMOS-RAM backup

Maintaining the contents of CMOS-RAM by an externally connected battery even when the CARD-PC itself is powered off. In the CARD-PC the backup power supply for the RTC and CMOS-RAM is provided by the same battery.

SCR file

File containing instructions for operation of the packet installation utility (ROMADD.EXE). Regular DOS text file which can be edited with a text editor.
See Packet installation utility

Script file

Script files for the RAK system comprise script files for RAKDA (extension SCX) and script files for RAADD.EXE (extension SCR). SCX files contain menu selections for various BIOS settings under RAKDA. See SCX file
SCR files contain instructions for ROMADD operation. See SCR file

SCX file

Contains menu selections for BIOS settings under RAKDA. Can be used across different BIOS versions and CARD-PC models if TSV (defined separately from BIOS version and script file version) is identical.

SETUP

In this manual, this indicates ROM SETUP for the CARD-PC. In the standard BIOS this is included in the BIOS flash ROM, and when the system starts is accessed by pressing the F10 key. It carries out automatic hardware detection, and sets the correct values in CMOS-RAM. The user can also change the hardware settings. Note that SETUP must be located as rare data on a 2 KB boundary. The module file is a COM program file named SETUP####.COM (###: model name) and can be executed under MS-DOS®.

Setup utility

See SETUP

Shadow RAM data

Data in the flash ROM which at startup time is expanded and copied to addresses in the range A0000h to FFFFFh. In the CARD-PC this expansion and copying is done by BootBlock.

TSV

Acronym for parameter Table Set Version. Specified in the BIOS and in script files. TSV is defined separately from the BIOS version and script file version. If TSV matches, operation with differing BIOS or script file versions is possible. If TSV does not match, certain items and BIOS parameters cannot be set. Use script files with matching TSV.

WFLASH

Utility for rewriting the flash ROM in the CARD-PC. Can be used on hardware that is compatible with flash ROM power supply control, such as the evaluation kit and CardPresso. When a similar circuit is designed, use on a CARD-PC user system is also possible.
See Evaluation kit, CardPresso



CARD-PC

ROM Adaptation Kit for DBIOS

Manual

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

Electronic Devices Information on Epson WWW server

<http://www.epson.co.jp>

First issue January, 1998
Printed in Japan © A