MF1234-01

EPSON

CARD-E09A

Windows CE 2.11 Development Kit Instruction Manual (English)

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

CARD-E09A

Windows CE 2.11 Development Kit Instruction Manual (English)

NOTICE:

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Control Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from anther government agency.

Contents

1. Introduction

This kit is designed to provide Windows CE development environments for systems using CARD-E09A (SH-card).

In addition to this kit, the following is required for developing Windows CE for CARD-E09A:

- CARD-E09A Evaluation Kit (hereinafter referred to as "Evaluation Kit")

The following are required to use the Windows CE Platform Builder included in this kit:

- PC to which Windows NT 4.0 has been installed
- PC compatible with a PCMCIA card and a CD-ROM

This kit includes the following:

- "CARD-E09A Windows CE 2.11 DevKit" Agreement
- "CARD-E09A Windows CE 2.11 DevKit" Instruction Manual (this manual)
- Windows CE Platform Builder
- CARD-E09A CE 2.11 DevKit CD
- Serial cable
- PPSH cable

"SH-CARD Windows CE 2.11 DevKit" Agreement

Agreement for software packaged together with the SH-CARD CE 2.11 DevKit CD. We recommend you to read it before using the software.

CARD-E09A CE 2.11 DevKit CD

This DevKit provides the required environment for operating Windows CE on CARD-E09A. To be precise, this includes a set of Windows CE OALs and drivers customized for CARD-E09A, NKLOAD required for activating Windows CE, various tools, etc.

Windows CE Platform Builder

Windows CE development kit for builders provided by Microsoft.

PPSH cable

Used to connect a Windows NT-running PC and a CARD-E09A extension board to transfer CE image files directly from the PC.

Serial cable

Used for debugging Windows CE.

2. Before Using DevKit

Be sure to perform the following operation before using this DevKit. The descriptions below are also explained in the Evaluation Kit. The following operation is not required if it has already been done and you have an ATA card for backup. The backup ATA card is required to write a loader for booting the system (Windows CE and simplified debugger) into a new ATA card SBR (second sector within the card). Pay sufficient attention so as not to destroy the contents.

2.1. Backing up ATA Card

An ATA card (master), which a simplified debugger has been installed, is included in the Evaluation Kit. Be sure to back it up before usage. The backup procedure is as follows:

1. Prepare a formatted backup ATA card (4MB min.).
2. Copy all files stored in the master ATA card to the backup ATA card. For copying, use a PC or the like having two PCMCIA slots. The Explorer of Windows NT or the like and the copy command at the command prompt are also available for this purpose. If no PC is available, the files can still be copied by entering the following command after starting Windows CE using the Evaluation Kit from the master ATA card and executing CMD.EXE.

```
CD  STORAGE CARD  <Enter>
COPY  *.*  \STORAGE CARD2  <Enter>
```

3. Insert the master ATA card into PCMCIA slot (CN11) A of the SH card extension board and the backup ATA card into slot B. For the positions of slots A and B, refer to the figure below.

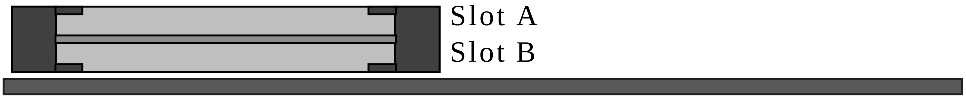


Figure 1 – PCMCIA Slot

4. Connect the AC plug of the power unit to the power connector and turn the power on.
5. After the simplified debugger screen appears on the CRT, enter the following command using the keyboard.

```
-SBR  SBR.BIN  B  <Enter>
```

6. After the above execution is completed and prompt "-" appears, turn the power off.
7. This completes backup.

2.2. If the Contents of Backup ATA Card Are Destroyed

The backup ATA card is always required to write an SBR onto a new ATA card. So, if the contents of the backup ATA card are destroyed, the simplified debugger and Windows CE cannot be started. In this case, contact the nearest EPSON sales office since another backup ATA card needs to be prepared.

3. Installing DevKit

Please follow this procedure to install correctly the Windows CE DevKit.

3.1. Installing Platform Builder

Install Windows CE Platform Builder 2.11 onto a PC used for developing Windows CE. Executing SETUP in Disc 1 of the Platform Builder starts installation.

Note

☞ For the installing method, “Custom install” for “Install option” of each stage is strongly recommended. Since “Custom install” eliminates not necessary platform installation, the number of files to be copied into the hard disk decreases, significantly reducing the hard disk capacity to be occupied.

3.2. Installing CARD-E09A CE2.11 DevKit CD

It is possible to install two version of the DevKit :

- MAXALL permits to generate a complete kernel of Windows CE
- MINSHELL permits to generate a reduction of Windows CE kernel

3.2.1. Install DevKit for the Full Version (MAXALL)

After installing the Platform Builder, execute INSTALL1.EXE in the CARD-E09A CE2.11 DevKit CD. This generates CARD-E09A Windows CE development environments.

INSTALL1 .EXE

☞ Building CE images before executing INSTALL1.EXE after installing the Platform Builder may prevent CARD-E09A Windows CE development environments from being properly installed.

☞ Executing INSTALL1.EXE displays the installation state on the screen. Make sure that no error message appears after installation completed. If any error occurs, check the contents of the INSTALL.LOG text file generated in the WINCE211 folder.

☞ Before reinstalling the DevKit, be sure to execute UNINST1.BAT.

3.2.2. Install DevKit for the Minimum Version (MINSHELL)

This DevKit is provided with an installer for the above two types of projects. If you want to use other project than the above, follow the procedure below.

1. Copy all files in the DevKit CD as they are to the HDD. Using the Explorer attached to Windows NT or the like simplifies copying.
2. Open the INSTALL2.BAT file in the copy-to root folder using the editor.
3. Change MINSHELL written below to a desired project name. The following shows an example in the case of MINKERN.

   Before change

   ```
   SET NEW_PLAT=SH-CARD
   SET NEW_PROJ=MINSHELL
   SET SRC_PLAT=ODO
   SET SRC_PROJ=MINSHELL
   SET SHORTCUTNAME=CARDMIN
   ```

   After change

   ```
   SET NEW_PLAT=SH-CARD
   SET NEW_PROJ=MINKERN
   SET SRC_PLAT=ODO
   SET SRC_PROJ=MINKERN
   SET SHORTCUTNAME=CARDMIN
   ```
4. Save the contents of the file to update them after change.
5. Execute INSTALL2.EXE in the same folder as the updated file. No operation is changed after installation.

After installing the Platform Builder, execute INSTALL2.EXE in the CARD-E09A CE2.11 DevKit CD. This generates CARD-E09A Windows CE development environments.

INSTALL2.EXE

☞ Building CE images before executing INSTALL2.EXE after installing the Platform Builder may prevent CARD-E09A Windows CE development environments from being properly installed.

☞ Executing INSTALL2.EXE displays the installation state on the screen. Make sure that no error message appears after installation completed. If any error occurs, check the contents of the INSTALL.LOG text file generated in the WINCE211 folder.

☞ Before reinstalling the DevKit, be sure to execute UNINST2.BAT.

4. Creating and Executing Windows CE Image

This chapter explains how to create a standard CE image and execute it on CARD-E09A.

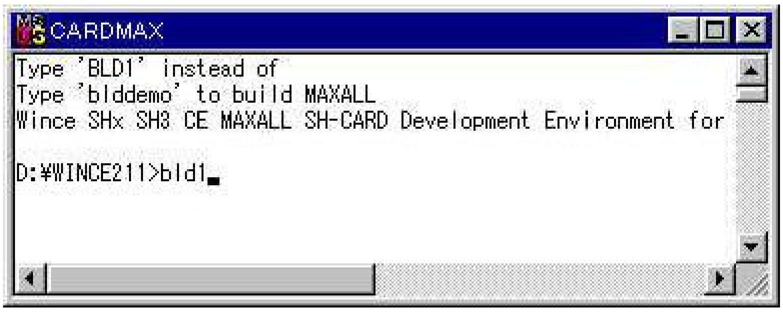
4.1. Create CE Image File

4.1.1. CARDMAX: Create CE Image file

1. Execute CARDMAX in the WINCE211\PUBLIC\MAXALL folder.

CardMax.EXE

1. Execute BLD1 at the command prompt of CARDMAX. The command name to be typed is displayed as shown below. Proceed by following the instruction.

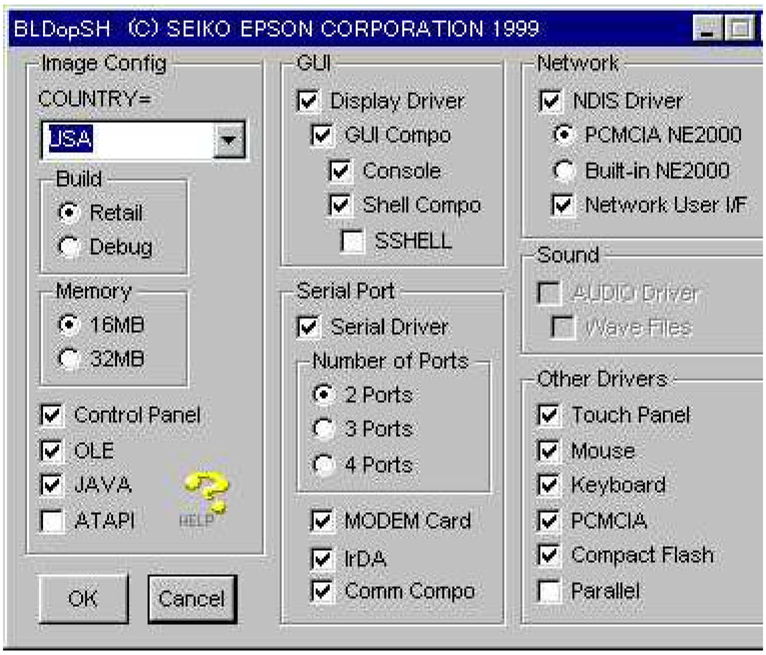


CARDMAX

```
Type 'BLD1' instead of
Type 'blddemo' to build MAXALL
Wince SHx SH3 CE MAXALL SH-CARD Development Environment for

D:¥WINCE211>bld1_
```

2. BLDopSH is executed to display the following screen. Set functions to install on this screen. By default, standard SH-CARD functions of the Platform Builder are set.



BLDopSH (C) SEIKO EPSON CORPORATION 1999

Image Config
COUNTRY= USA
Build: ● Retail ○ Debug
Memory: ● 16MB ○ 32MB
☑ Control Panel
☑ OLE
☑ JAVA
□ ATAPI

GUI
☑ Display Driver
☑ GUI Compo
☑ Console
☑ Shell Compo
□ SSHELL

Serial Port
☑ Serial Driver
Number of Ports: ● 2 Ports ○ 3 Ports ○ 4 Ports
☑ MODEM Card
☑ IrDA
☑ Comm Compo

Network
☑ NDIS Driver
● PCMCIA NE2000
○ Built-in NE2000
☑ Network User I/F

Sound
□ AUDIO Driver
□ Wave Files

Other Drivers
☑ Touch Panel
☑ Mouse
☑ Keyboard
☑ PCMCIA
☑ Compact Flash
□ Parallel

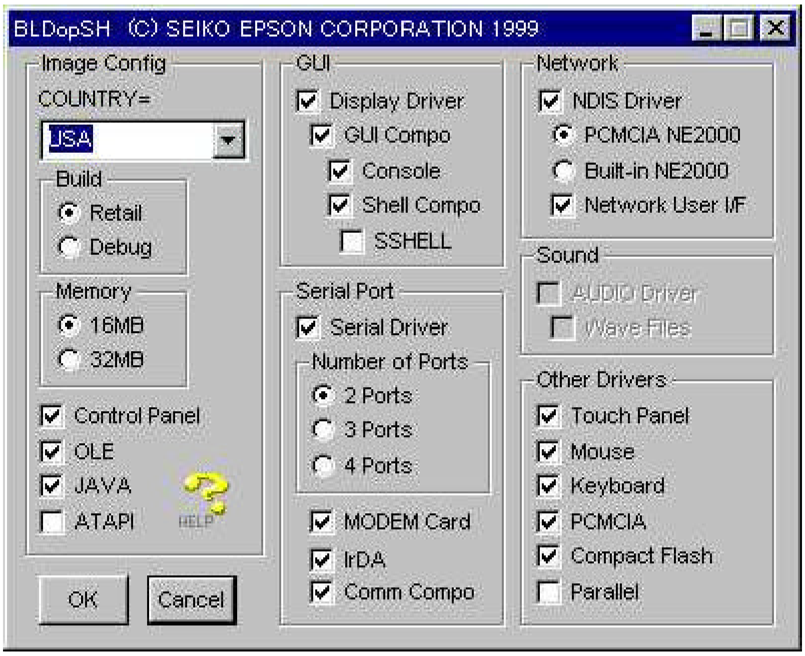
3. Then, BLDDEMO is automatically executed to generate a CE image file (NK.BIN) in the WINCE211\RELEASE folder. This NK.BIN file is used for starting Windows CE.

4.1.2. CARDMIN: Create CE Image file

1. Execute CARDMIN in the WINCE211\PUBLIC\project name folder.

CardMin.EXE

2. BLDopSH is executed to display the following screen. Set functions to install on this screen. By default, standard SH-CARD functions of the Platform Builder are set. After OK button is pressed, some environment valuables are set automatically and command prompt screen is displayed.



BLDopSH (C) SEIKO EPSON CORPORATION 1999

Image Config

COUNTRY= USA

Build: ● Retail ○ Debug

Memory: ● 16MB ○ 32MB

☑ Control Panel
☑ OLE
☑ JAVA
□ ATAPI

GUI

☑ Display Driver
☑ GUI Compo
☑ Console
☑ Shell Compo
□ SSHELL

Serial Port

☑ Serial Driver

Number of Ports: ● 2 Ports ○ 3 Ports ○ 4 Ports

☑ MODEM Card
☑ IrDA
☑ Comm Compo

Network

☑ NDIS Driver
● PCMCIA NE2000
○ Built-in NE2000
☑ Network User I/F

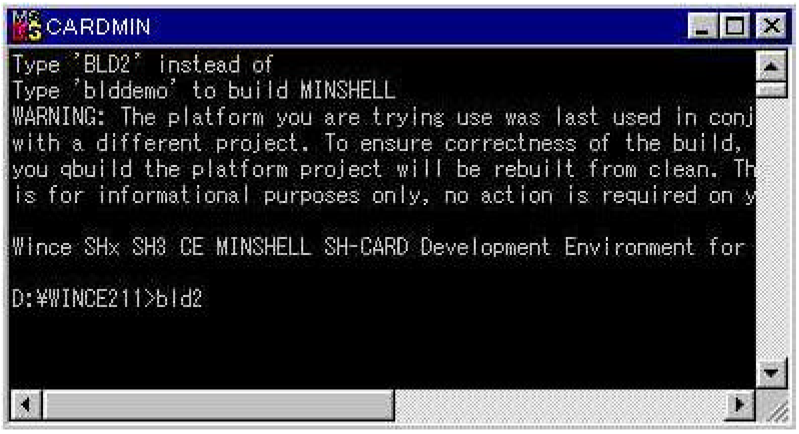
Sound

□ AUDIO Driver
□ Wave Files

Other Drivers

☑ Touch Panel
☑ Mouse
☑ Keyboard
☑ PCMCIA
☑ Compact Flash
□ Parallel

3. Execute BLD2 at the command prompt of CARDMIN. Or, execute BLD2 if INSTALL2.EXE is executed at the time of installation. The command name to be typed is displayed as shown below. Proceed by following the instruction.



```
Type 'BLD2' instead of
Type 'blddemo' to build MINSHELL
WARNING: The platform you are trying use was last used in conj
with a different project. To ensure correctness of the build,
you qbuild the platform project will be rebuilt from clean. Th
is for informational purposes only, no action is required on y

Wince SHx SH3 CE MINSHELL SH-CARD Development Environment for

D:¥WINCE211>bld2
```

4. Then, BLDDEMO is automatically executed to generate a CE image file (NK.BIN) in the WINCE211\RELEASE folder. This NK.BIN file is used for starting Windows CE.

4.2. Transferring and Starting CE Image File

This section explains how to transfer a created CE image file (NK.BIN) to a target platform.
Basically, you could transfer the CE image file via the PCMCIA socket or via the parallel cable .

4.2.1. Transfer and Startup via PPSH Cable

The following conditions must be satisfied to use a PPSH cable for transfer.

- Windows NT is running on the PC
- The ERP (Enhanced Parallel Port) function is supported

The transfer method is as follows:

1. Connect the parallel ports of the PC and the CARD-E09A extension board using the PPSH cable.
2. Generate the CE image (see 4.1)
3. Type as follows and execute CESH.
   cesh -p CEPC <Enter>
4. Insert an ATA card including a loader to PCMCIA slot A of the CARD-E09A extension board and turn the SH extension board on.
5. Then, transfer automatically starts. It takes approximately more than one minute to complete the transfer, although it depends on the file size.
6. After transfer is completed, Windows CE is automatically started.

NK.BIN written in the CompactFlash is started in the following cases:

- No PPSH cable is connected.
- CESH is not started.
- No NK.BIN file exists in the Wince211\release folder.
- Transfer fails.
- PPSH is disabled in the [DEBUG] key in startup data.

Note that Windows CE operation may slow down if the PPSH cable is connected and if CESH is not executed.

4.2.2. Starting Windows CE2.11

This section explains how to start Windows CE using an ATA card (including the CompactFlash). Before performing the following operation, make sure that the ATA card has been backed up.

1. Copy all files under folder Tools in the CARD-E09A CE 2.11 DevKit CD attached to this DevKit without changing the images using a PC compatible with a PCMCIA card and a CD-ROM into the ATA card (CompactFlash).
   Note the following during copying.
   ☞ When an ATA card created by following the procedure written in “2.1 Backing up ATA Card” is used, the files need not to be copied.
   ☞ Once the files have been copied to an ATA card (CompactFlash), they need not be copied again except the case they have been erased or destroyed.
2. Copy the NK.BIN file created on a Windows NT-running PC to the ATA card (CompactFlash).
3. Mount the ATA card (CompactFlash) onto CARD-E09A or the SH card extension board and turn the power on.

This starts Windows CE.

- Since no Explorer is attached to Windows CE 2.11, specify the files using the task bar (Ctrl + ESC) for startup.

Windows CE 2.11 can be started with the following two methods:

- Mount the CompactFlash card connected to the ATA card adapter onto slot A or B of the PCMCIA slot (CN11) of the SH card extension board and turn the system power on.
- Remove the card from the ATA card adapter, mount it to the CARD E09A CompactFlash card connector and turn the system power on. This method is not recommended for system evaluation since CARD-E09A need to be removed from the SH card extension board every time of evaluation.

4.2.3. Priorities in Windows CE (Simplified Debugger) Startup

Windows CE (and the simplified debugger) is started according to the predefined priorities. This section explains these priorities in startup.

- Windows CE (and the simplified debugger) is started according to the following priorities. In order to activate both cards, however, it is required that an SBR has been written and the files required for startup exist.

1. ATA card in PCMCIA slot A on SH card extension board
2. ATA card in PCMCIA slot B on SH card extension board
3. CompactFlash card built in SH card

- The file reading order at startup is as follows: If the SBR fails to be read, the next card is read. Even if no STARTUP.DAT exists, NK.BIN, if any, is started. As startup data in that case, the default values in CE are used.

1. SBR
2. STARTUP.DAT
   The simplified debugger is executed according to the description in STARTUP.DAT.

The debugger is not started if no STARTUP.DAT exists.

3. NKLOADS.BIN NK.BIN

- If reading of all files written above fails, the simplified debugger written inside IPL is started.

4.2.4. Before Using FrzDri

The CARD-E09A Windows CE 2.11 DevKit is equipped with a function to save Windows CE data as files and restore it at rebooting. This is called the FrzDri function. In order to use this function, a file for saving Windows CE data requires to be created when a new NK.BIN is built.
Be sure to create this file when the project name for building NK.BIN or an environment variable setting has been changed (in other words, the size of NK.BIN has been changed). The file creating method is as follows:

How to create FrzDri file

1. Mount an ATA card including the NK.BIN file on a PC compatible with a PCMCIA card.
2. Open the command prompt assigned to the ATA card.
3. Make sure that NK.BIN exists in the root of the opened command prompt.
4. Execute the following command at the command prompt where NK.BIN exists.

```
WinNT\i386\FzdPrep  <Enter>
```

This creates an NK.FZD file for saving data in the ATA card.

5. Loader

The EPSON CARD-E09A loader is provided with additional functions to start Windows CE and supplement its functions.

5.1. Major Functions

Load Windows CE

Activates Windows CE.

Display Logo

A desired bitmap file can be displayed when activating Windows CE.

FrzDri function

Saves setting information or a file on the RAM of Windows CE and restores it at startup.

LCD Compatible

Compatible with various LCDs simply by changing startup data without changing the Windows CE video driver.

Adjust Resolution

The resolution can be adjusted simply by changing startup data without changing the Windows CE video driver.

Activating simplified debugger

The simplified debugger can be started instead of Windows CE.

Transfer OS system image to target platform

The NK.BIN file can be transferred to CARD-E09A from a Windows NT-running PC using the PPSH cable.

5.2. Activating Windows CE

Different from OSs for PCs like Windows 98 or NT, Windows CE is designed to always hold memory data. This results in completely different methods for activation and the like from other OSs. Such difference may limit Windows CE specifications. Extended functions prepared for CARD-E09A are intended to eliminate such limitation and enable easier operation. This section explains how CE operates.

NK.BIN

A file named NK.BIN appears after building Windows CE. Putting this file onto the memory space and jumping to the entry point activates Windows CE. On the H/PC and the like, NK.BIN is positioned on the ROM mapped to the memory space.

RAM section

Since Windows CE is designed to operate without writable storage besides the RAM, it saves items whose contents change like registries and files in the RAM. This area is called “RAM section”.

Cold boot

It refers to the state in which the DRAM is initialized to activate Windows CE (NK.BIN) after testing DRAM read/write. This state is assumed after turning the power on or resetting hardware (or pressing the reset button). Since the RAM section is initialized, no previous state is saved. Cold boot can also be performed from sample shell SSHELL.

Warm boot

Initializes Windows CE while data in the RAM section is held. This is required to reboot the system in a way to reflect registry change after, for example, a device has been added. This functions in the same way as when the H/PC reset switch is pressed. Since CARD-E09A Windows CE can be warm-booted using manual reset, preparing a circuit for activating manual reset materializes a reset switch for Windows CE. Warm boot can also be performed from sample shell SSHELL.

Turn Windows CE On/Off

On the H/PC, a typical Windows CE built-in product, the function of the ON/OFF key is not to turn the power on and off but to suspend and resume the system. So, when turned on again after turned off, it returns to the state immediately before turned off. On CARD-E09A, the previous state is not held with normal turning on/off operation. So, the CARD-E09A extension board is equipped with an SRB button for suspend/resume operation. In addition, the FrzDri function is equipped for CARD-E09A to enable setting information to be held. However, the FrzDri function is different from the suspend/resume function and not designed to hold all states before turned off. For example, the program running state is not saved.

5.3. Configuration of Loader

The CARD-E09A loader consists of 3 programs, namely IPL (ipl.bin) to be written onto the Flash ROM on CARD-E09A, SBR (sbr.bin) and NKLOADS (nkloads.bin) to be stored on the CompactFlash and a data file called “startup data” (startup.dat).

IPL (Initial Program Loader)

IPL is executed first after CARD-E09A is started. After simple initialization, it reads the SBR on the CompactFlash and passes control to the SBR.

SBR (Sub Boot Record) (or Second Boot Record)

SBR is a sub loader to be stored in an area not to be formatted. It is written in an area from the sub boot record (sector #2) of the ATA card (CompactFlash). It analyzes the ATA card (CompactFlash) file system and controls reading by NKLOADS, a DOS file.

NKLOADS

In addition to the function for loading a CE image file (NK.BIN) stored in the CompactFlash, major extended functions of the loader are materialized by NKLOADS. NKLOADS is also used to initialize the PCMCIA controller, video controller or serial with the built-in CPU, bus controller and companion chip. Put NKLOADS as a file in the root folder of the DOS-formatted CompactFlash. File systems to be supported are FAT12 and FAT16 only. Other FAT files cannot be recognized properly.

Startup data (Startup.dat)

Various setting information for starting NKLOADS. The startup data is a text file of a pre-defined format and must be put in the root folder of the CompactFlash.

5.4. Error Detection Using Loader

IPL and NKLOADS output error detection information with the following method if any error is detected at system startup. With this information, the error type can be identified and the action to be taken judged.

1. The step ID is output to the LED (80h) of the ISH board.
2. Loader operation is stopped.

The following describes the contents of an error detected and the action to be taken for each step ID. While the step ID performs LED display only during IPL execution, an error message is also displayed on the CRT during NKLOADS execution. Note that nothing is displayed on the LCD. The error message appears on the upper left corner on the screen.

5.4.1. Error Detection During IPL Execution

- Step ID: 0xEF

  Cause

  - No startup ATA card is recognized.

  ① No startup ATA card or CF is mounted.

  ② SBR is not recorded in the startup ATA card.

  Action

  ① Make sure that the startup ATA card exists.

  ② Rewrite SBR.BIN in the startup ATA card using the simplified debugger.

- Step IDs: F1, F2

  Cause

  - An error is detected in the data or address line.

  Action

  - Make sure that no foreign matter or flaw exists on the SH-CARD.

  - Use another card.

5.4.2. Error Detection During NKLOADS Execution

- Step ID: 0xC0

  Error message

  "RAM section save error..."

  Cause

  - Saving of data in the RAM section of Windows CE has failed.

  ① An error occurs on NK.BIN of the startup ATA card. (NK.BIN signature error)

  ② No NK.FZD file exists in the startup ATA card.

  ③ NK.FZD does not correspond to NK.BIN.

  Action

  ① Make sure that the signature is correct.

  ② Recreate NK.FZD.

- Step ID: C1

  Error message

  "RAM section restore error..."

  Cause

  - Loading data in the of RAM section of Windows CE has failed.

  ① No NK.BIN exists in the startup ATA card. (NK.BIN is required to load the RAM section.)

  ② An error occurs on NK.BIN of the startup ATA card.

  ③ An error occurs on NK.FZD of the startup ATA card.

  Action

  ① Make sure that the signature is correct.

  ② Recreate NK.BIN.

- Step ID: C2

  Error message

  "ATA-CARD error..."

Cause

- An error occurs on the file management system ??? of the ATA card.

Action

- Reformat the startup ATA card.

- Step ID: C4

Error message

"NK.BIN error..."

Cause

- The NK.BIN entry address is wrong.

Action

- Check the address.

- Step ID: D?

Error message

"RAM error..."

Cause

- Memory element has broken.

Action

- Use another element.

6. Startup Data

Startup data is a collection of parameters specifying operation methods of various devices and programs. Changing these parameters enable the following functions.Video initialize parameter

- Logo image file name and display position
- Selection of data file name and restoration method for FrzDri
- Selection of function of manual reset button
- Execution file name of simplified debugger
- Selection of serial port used as Debug Serial
- Disabling PPSH
- Setting data inherent to touch panel/hardware
- Setting level of memory test performed by NKLOADS

The startup data is a text file of a pre-defined format and must be put in the root folder of the CompactFlash. Startup data can easily be edited by mounting the ATA card onto which the data has been written to a PC compatible with a PCMCIA card.

Note the following when handling the startup data.

☞ The file name used for the startup data is fixed at startup.dat. It is not case-sensitive.

☞ The parameter value to be defined in the startup data is an unsigned hexadecimal value of up to 8 digits. Alphabets are not case-sensitive.

☞ The part from “;” to the end of the line is assumed to be a comment and neglected by NKLOADS. So, write “;” at the head to temporarily disable the function.

☞ For details of the startup data specifications, refer to the attached “Startup Data Specifications”.

7. Descriptions of Each Function

7.1. Loading Function of Windows CE

Building Windows CE using the Windows CE Platform Builder generates a file named NK.BIN. This is a Windows CE image. Putting this image onto the memory space (loading) and jumping to the NK.BIN entry point activates Windows CE. For this operation, which is enabled by the loader, note the following:

Precautions on loading Windows CE

- The loader activates Windows CE according to the following priorities:

1. NK.BIN in PCMCIA slot A (further slot from the board) built in the CARD-E09A extension board.
2. NK.BIN in PCMCIA slot B (nearer slot to the board) built in the CARD-E09A extension board.
3. NK.BIN in the CompactFlash built in the CARD-SH.

If no NK.BIN file exists in both ATA cards, nothing is displayed and the system is hung up.

- Since Windows CE is normally started from the CompactFlash built in the CARD-SH, the ATA card inserted in each PCMCIA slot is available with the following folder name. For PCMCIA slots A and B, the folder name to be assigned differs depending on the order with which the card has been inserted. The following lists the assigned folder names for each case.

When the built-in CF card is available and ATA cards are inserted to both PCMCIA slots before startup

CARD-SH-built-in slot	Storage Card
PCMCIA slot A	Storage Card2
PCMCIA slot B	Storage Card3

When the built-in CF card is available but nothing is inserted to either PCMCIA slot before startup and then an ATA card is inserted to PCMCIA slot A first

CARD-SH-built-in slot	Storage Card
PCMCIA slot A	Storage Card2
PCMCIA slot B	Storage Card3

When the built-in CF card is available but nothing is inserted to either PCMCIA slot before startup and then an ATA card is inserted to PCMCIA slot B first

CARD-SH-built-in slot	Storage Card
PCMCIA slot A	Storage Card3
PCMCIA slot B	Storage Card2

- When no built-in CF card is available before startup and ATA cards are inserted to both PCMCIA slots
  Since the existence of the CompactFlash built in the CARD-SH cannot be checked by software, the following message appears when Windows CE is started if no CompactFlash actually exists.

  Unidentified PCCard Adapter
  Enter the name of the driver for this PCCard

Drive Name: Unknown card in Socket 1

In this case, Windows CE can properly be started if NK.BIN exists in the ATA card inserted to slot A after pressing the <Enter> key.
After startup, the ATA card inserted in each PCMCIA slot is respectively available in the following folders:

PCMCIA slot A	Storage Card
PCMCIA slot B	Storage Card2

In this case, the folder name of slot A is changed to “Storage Card” used for CF. This is because the ATA card is assumed to be used primarily for developing applications. By developing an application referring to this folder name on the ATA card and copying it to CF at the time of commercialization, it can be executed as it is.

7.1.1. How Not to Display Message at CE Startup When No CF Exists

Even if no CompactFlash actually exists, it is possible not to display the error message at Windows CE startup. This is enabled by setting BLDopSH to be executed at the time of building NK.BIN. Turn the following checkbox off when executing BLDopSH.

- Other Drivers – CompactFlash

7.2. Changing Video Initialize Parameter

The display color and resolution of Windows CE can be changed. For this purpose, change the startup data parameter values. The following explains the parameter changing method in details.

7.2.1. How to Change Video Initialize Parameter

1. Search the [1355] key in the startup data file.
2. Select appropriate items like the LCD and resolution and change various parameters for 1355. For example, the following table is generated for an LCD of 640 x 480 dots and resolution of 8bpp.

   In the table below, parameters in each line respectively mean as follows:
   SED1355 register index address SED1355 register setting (1-byte data)

3. For details of various registers in SED1355, refer to the attached “SED1355F Dot Matrix Graphics LCD Controller” Technical Manual.

```
[1355]
; Index, data
B4000001    30          ;/* CLKI/512 (65KHz refresh at 33MHz), 2CAS, EDO  */
B4000002    2E          ;/* 8bit Passive, Color */
B4000003    00          ;/* LCD */
B4000004    4F          ;/* H size = (79+1)*8 = 640 pixel */
B4000005    1D          ;/* H non-display = (30+1)*8 = 248 pixel */
B4000006    01          ;/* HRTC start = (1+1)*8-2 = 14 pixel */
B4000007    01          ;/* HRTC pulse = (7+1)*8 = 64 pixel */
B4000008    DF          ;/* V size = 479+1 = 480 */
B4000009    01          ;/* V size */
B400000a    09          ;/* V non-display = 19+1 = 20 lines */
B400000b    00          ;/* VRTC start = 0+1 = 1 line */
B400000c    00          ;/*`VRTC pulse = 2+1 = 3 lines */
B400000d    0F          ;/* CRT & LCD power should off while init the chip */ /* 8bpp */
B400000e    FF          ;/* Split line (Word) */
B400000f    03          ;
B4000010    00          ;/* Screen 1 start [7:0](Word) = 0h */
B4000011    00          ;/* [15:8] */
B4000012    00          ;/* [19:16] */
B4000013    00          ;/* Screen 2 start (Word) = 28000*2 = 50000h */
B4000014    00          ;/* [15:8] */
B4000015    00          ;/* [19:16] */
B4000016    40          ;/* Memory Address Offset */
B4000017    01          ;/* 640 = 280h then 140h (word) */
B4000018    00          ;
B4000019    00          ;/* Clock Configuration Resister */
B400001a    02
B400001b    01
B400001e    00
B400001f    00
B4000020    00
B4000021    00
B4000022    24
B4000023    00
FFFFFFFF    FFFFFFFF            ;END Mark
```

7.3. Displaying Logo

A desired logo image can be displayed when activating Windows CE. The file format available for displaying a logo image is limited to BMP only.
The following shows the procedure for displaying such images.

1. Prepare an 8bpp (256-color) BMP file to be used as a logo image. Be sure to set the resolution of the BMP file equal to or under the resolution of the screen.
2. Copy the BMP file to the root folder of the CompactFlash.
3. Within the startup data file, specify the file name and display position of the BMB file. For the

display position, the center display on the screen or specification by dot can be selected.

7.3.1. How to Specify Logo Image File Name

1. Search the [LOGOFILE] key in the startup data file.
2. In order to specify logofile.bmp as the BMP file name, for example, describe as follows: (the file name is not case-sensitive)

```
[LOGOFILE]
 logofile.bmp
```

3. If no logo image is specified, Windows CE is started without any display.

7.3.2. How to Adjust Display Position of Logo Image File

1. Search the [LOGOXY] key in the startup data file.
2. In order to position the logo image at the center on the screen, for example, describe as follows: (it is not case-sensitive)

```
[LOGOXY]
FFFF  FFFF
```

The first parameter indicates the horizontal position of the left end of the logo image and the second parameter indicates the vertical position of the upper end in pixel units. Only when FFFF is set, the image is positioned at the center on the screen.

7.4. FrzDri Function

Windows CE is an OS designed to always hold memory data. Since the RAM section is used as the storage for registries and files, no setting information or file saved onto the RAM is stored after Windows CE is turned off. The CARD-E09A Windows CE 2.11 DevKit is equipped with a function to save this data as files and restore it at rebooting. This is called the FrzDri (pronounced as “Freeze Dry”) function. This function enables setting information and files on the RAM to be saved.

Using this function easily enables the following:

- When settings require to be changed frequently, setting information is always held after the power is turned off and the system is rebooted next time with the latest settings.
- Although it is desirable that all systems operate as set on shipment, each of them actually requires to be set respectively. (This usually requires NK.BIN for each unit to be built.
- Several startup states can be switched optimally.

FrzDri has the following two major functions:

- Backs up data in the RAM area as files when Windows CE is finished.
- Restores RAM area data saved as files in the RAM at rebooting.

7.4.1. Backup (S2D)

Creat backup file

The contents of RAM data is saved in the NK.FZD file. Since NK.FZD is not automatically created, it must be positioned in the root folder of the startup disk in the same way as for NK.BIN. Prepare NK.FZD using the FZDPREP utility.

FZDPREP.EXE is included in the \TOOLS directory of the DevKit CD. FZDPREP.EXE has two versions, namely for Windows CE and Win 32.

Windows CE version

Copy FZDPREP.EXE and FZDPREPR.EXE into the same directory as NK.BIN and execute FZDPREP.EXE. On success, empty NK.FZD is created in the same directory.

Win 32 version

Copy FZDPREP.EXE and FZDPREPR.EXE into the same directory as NK.BIN and execute FZDPREP.EXE. On success, empty NK.FZD is created in the same directory. Copy created NK.FZD to the root folder of the ATA card for activating CE together with NK.BIN.

On either version, an error occurs to finish the system if NK.FZD has already existed in the same directory. Both FZDPREP.EXE and FZDPREPR.EXE are required since the former calls the latter. FZDPREPR.EXE prepares a file of an appropriate size based on map information in NK.BIN.

Back up RAM data

RAM data can be backed up with the following methods:

1. Use sample shell SSHELL included in the DevKit. Selecting “Power-off after Backup” for “Exit Windows” in sample shell SSHELL starts operation necessary for finishing CE and saves the RAM data in NK.FZD.
2. Press the manual reset button. After the button is pressed, the RAM data is automatically saved in NK.FZD. Before using this method, make sure that “Specify manual reset button function” in key [FRZDRIOPTION] in the startup data is set at FrzDri.

With both methods, the display disappears after saving has succeeded. This states allow the power to be turned off. If saving has failed, error information is output and processing halted. If saving has properly been completed, PWOFF# turns “LOW”. For end users using this function, it is strongly recommended to configure the hardware so as to materialize a soft switch of PWOFF# for turning the power off outside CARD-E09A to automatically turn the power off.

Shrink backup file

FZDPREP.EXE prepares a backup file of the same size as the RAM available by CE. Since data seldom exists in all RAM area, however, NK.FZD can be shrunk if NK.FZD data is fixed (e.g. the case in which data is always restored by default). The FZDCUT utility is designed to cut an unnecessary area to shrink a file. By reducing the size, the disk area can be saved and the size to restore decreased, resulting in a shorter startup time.

Before shrinking a backup file, back up the default data in NK.FZD. Shrinking NK.FZD using FZDCUT.EXE generates a shrunk file named NK0.FZD. FZDCUT.EXE is included in the \TOOLS directory of the DevKit CD. FZDCUT.EXE has two versions, namely for Windows CE and Win 32.

Windows CE version

Put FZDCUT.EXE and FZDCUTTR.EXE in the same directory as NK.FZD and execute FZDCUT.EXE. On success, NK0.FZD is created in the same directory. NK.FZD must has backed up the RAM data (FzdDri) once at least.

Win 32 version

Prepare NK.FZD which has backed up the RAM data on Windows CE. Copy FZDCUT.EXE and FZDCUTTR.EXE into the same directory and execute FZDCUT.EXE. On success, NK0.FZD is created in the same directory. Copy created NK0.FZD to the root directory of the HDD for Windows CE.

Both FZDCUT.EXE and FZDCUTTR.EXE are required since the former calls the latter. FZDCUTTR.EXE deletes an unnecessary part of a file based on FrzDri information in NK.FZD.

7.4.2. Restore-booting RAM Data

RAM data is restored by the loader before starting Windows CE. Whether to boot after initializing the DRAM or restore-boot is determined by the existence of the NK.FZD and NK0.FZD files and the setting of the startup data. For restore-boot, the file to restore can be selected..

- NK.FZD fixed.
- NK0.FZD fixed.
- Started by NK.FZD. Started by NK0.FZD, however, if NK.FZD is not found or an error occurs.
- Started by NK0.FZD. Started by NK.FZD, however, if NK0.FZD is not found or an error occurs.

If restoration has failed due to the fact that no backup file exists or the file is not proper, it can be selected whether to perform cold boot or halt the system. This can be selected using the startup data setting. For details of the startup data specifications, refer to the attached “Startup Data Specifications”.

When using the restore function, note the following:

☞ If the NK.FZD file exists in the root folder, the file is automatically restored even if no description of the FrzDri file exists in the startup data. This initializes the contents of the registry. When you want to disable the restore function, change the file name or delete the file.

7.4.3. How to Change Manual Reset Button Function

1. Search the [FRZDRIOPTION] key in the startup data file.
2. Describe as follows to use the manual reset button for FrzDri: (by default, it is available for warm boot)

```
[FRZDRIOPTION]
 8    ;  Bit3 - 1: Functions as FrzDri.
```

7.4.4. How to Change FrzDri Logo Image

1. Search the [FZDBMP] key in the startup data file.
2. In order to specify sample.bmp as the logo image file name, for example, describe as follows:

```
[FZDBMP]
 sample.bmp
```

3. If no logo image file is specified or the specified file does not exist, nothing appears on the screen during FrzDri.

7.4.5. Chang File to Restore

1. Search the [FRZDRIOPTION] key in the startup data file.
2. Describe as follows to change the file to restore to NK0.FZD: (by default, NK.FZD is used)

```
[FRZDRIOPTION]
  1   ;  Bit0 - 1: NK0.FZD is used.
```

3. Even if restoration has failed with the specified file, restoration with another file is possible or the method for handling failed restoration can be selected. The following explains various setting methods.

   e.g.1: NK.FZD is used as the file to restore and NK0.FZD used if restoration has failed. If restoration has failed, the memory is cleared and the system is rebooted.

```
[FRZDRIOPTION]
   2  ;  Bit0 - 0: NK.FZD is used.
      ;  Bit1 – 1: NK0.FZD file is used if restoration has failed.
      ;  Bit2 – 0: The memory is cleared and the system is rebooted if restoration has failed.
```

   e.g.2: NK0.FZD is used as the file to restore and NK.FZD used if restoration has failed. If restoration has failed, the system is halted.

```
[FRZDRIOPTION]
   7   ;  Bit0 - 1: NK0.FZD is used.
       ;  Bit1 – 1: NK.FZD file is used if restoration has failed.
       ;  Bit2 – 1: The system is halted if restoration has failed.
```

7.5. LCD Compatible

This DevKit is compatible with an LCD simply by changing parameters in the startup data although the Windows CE video driver usually requires to be changed to enable LCD display with Windows CE. Standard compatible LCDs are 640 x 480 and 800 x 600. This function is available only when the CARD-E09A video driver of this DevKit is used. Although operation has been confirmed on the LCDs listed below, the setting parameters differ for each LCD. For details, contact EPSON sales personnel.

- TFT LCD 640X480 256/65536 color
- TFT LCD 800X600 256/65536 color
- STN LCD 640X480 256/65536 color

7.6. Adjust Resolution

The resolution can be changed simply by changing data in the startup data although the Windows CE video driver usually requires to be changed to enable the resolution to be adjusted on Windows CE. Standard available resolutions are 640 x 480 and 800 x 600. This function is available only when the CARD-E09A video driver of this DevKit is used. Consult EPSON sales personnel if you want to use other resolution.

7.7. Start Simplified Debugger

Specifying debugger startup in the startup data starts the simplified debugger instead of starting NK.BIN. The loader loads a debugger to the memory to executed it only when the debugger name is specified.

7.7.1. How to Start Simplified Debugger

1. Search the [DEBUGGER] key in the startup data file.
2. Describe as follows to specify DEBUGS.BIN as a debugger:

```
[DEBUGGER]
DEBUGS.BIN
```

3. If no debugger is specified, no debugger is started.

7.8. Set When No Mouse Is Set

Starting CE with a mouse driver installed and with no mouse connected takes a longer time. In this case, the startup time can be shortened by setting BLDopSH appropriately. Note, however, that the mouse is disabled if the following setting is turned on. So, turn the following checkbox off when executing BLDopSH.

- Other Drivers – Mouse

8. Set Windows CE-related Debug Functions

Windows CE is equipped with debug functions called “Debug Serial” and “PPSH”. These debug functions are designed to output debug information to respective connectors using serial and parallel respectively. So, it these connectors are used for other purposes, debugging is not performed properly. The following explains how to change the settings relating to the Debug Serial and the PPSH function.

8.1. Debug Serial

8.1.1. Communication Setting

The Debug Serial is set as shown in the table below. Set a debug information receiver accordingly.

Item	Setting
Data length	8 bit
Stop bit	1 bit
Communication rate	19200bps
Parity	None

8.1.2. Selecting Serial Channel

For the Debug Serial, select one of the following three channels to use it.

- Serial 0
- Serial 2
- Serial 3

The debug serial function can also be disabled.
The following explains the characteristics of each channel and how to use them.

8.1.3. How to Change Debug Serial Channel Number

1. Search the [DEBUG] key in the startup data file.
2. Describe as follows to use Serial 2 as the Debug Serial:

```
[DEBUG]
```

2

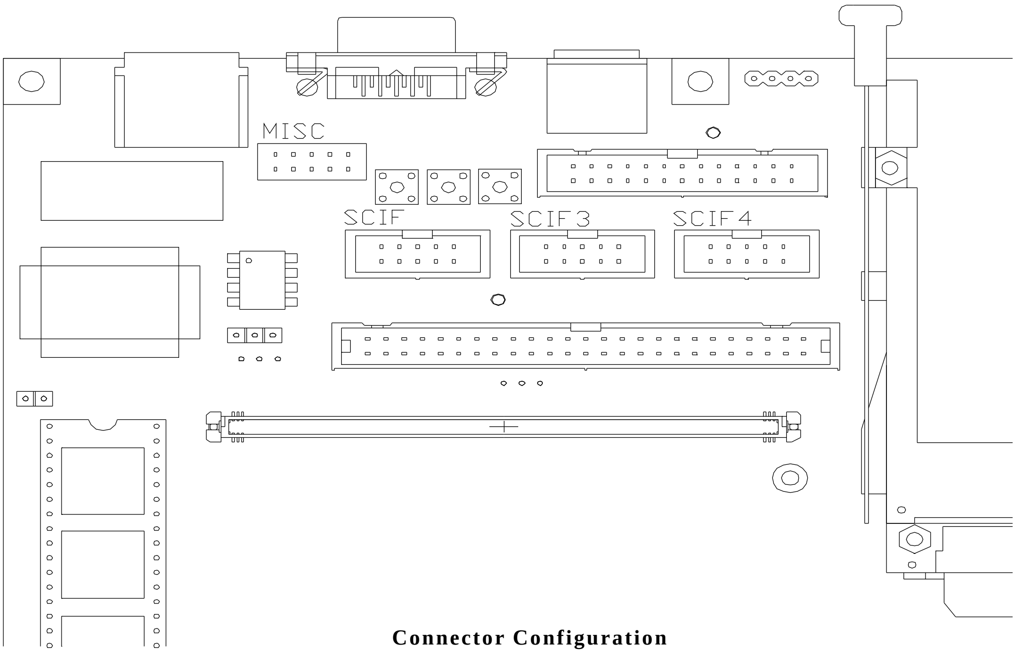
3. Selectable channel numbers are limited to Serial 0, Serial 2 and Serial 3. Describe the following numbers for each. By default, Serial 3 is used.

    0 - Serial 0
    3 - Serial 3

8.1.4. Connector Position

The figure below shows the connector position on the SH-CARD extension board. Each corresponds to as follows:

Serial 0/1 – MISC connector
Serial 2 – SCIF connector
Serial 3 – SCIF3
Serial 4 – SCIF4



Connector Configuration

8.2 Serial 0

8.2.1. Characteristics

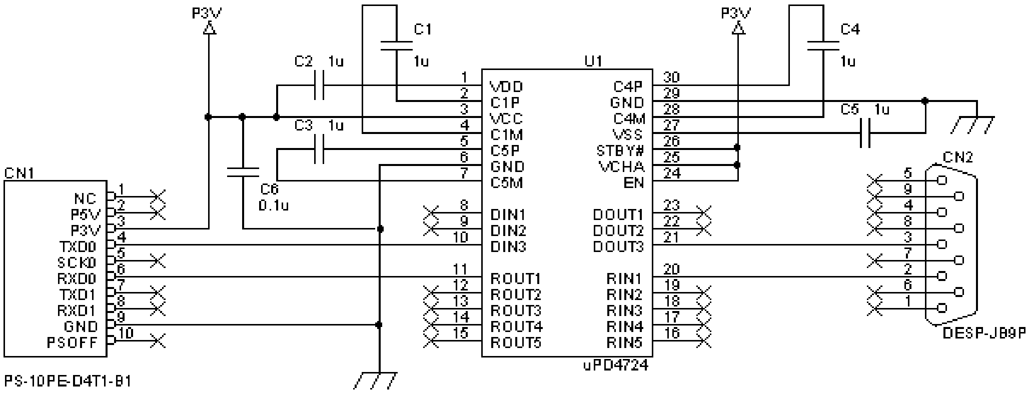
On the SH-CARD extension board, no FIFO for communication data or driver for making signals to satisfy the RS232C standard exists. In order to use this channel as the Debug Serial, an RS232C standard-based driver requires to be installed. When Serial 0 is used as the Debug Serial, all of Serial 1 to Serial 4 are available on Windows CE. When 4 channels are set to be available by the DevKit, each is assigned to as follows:

- Serial 1 – COM1
- Serial 2 – COM2
- Serial 3 – COM3
- Serial 4 – COM4

8.2.2. Prepare Hardware

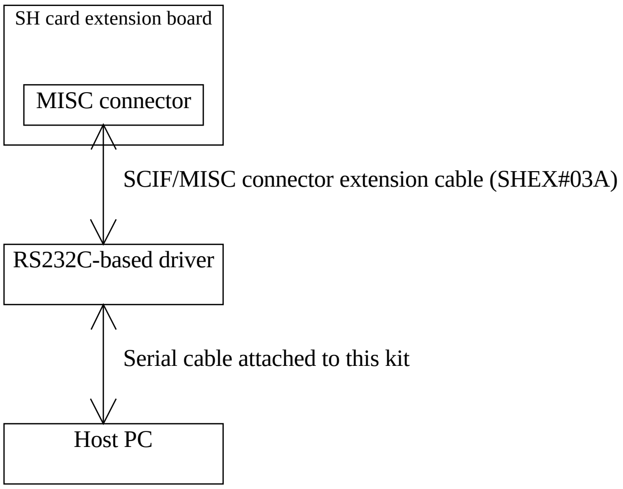
Signals of Serial 0 are connected to the MISC connector on the extension board.

Serial 0 has no RS232C-based driver on the SH card extension board. In order to use Serial 0 as the Debug Serial, prepare a cable with an RS232C-based driver installed using an SCIF/MISC connector extension cable (SHEX#03A). The figure below shows a reference circuit.



Reference Circuit of RS232C-based Driver

Connect the serial cable attached to this kit to the SCIF/MISC connector extension cable (SHEX#03A) with the above reference circuit added and then connect it further to the host PC.


SH card extension board
MISC connector
SCIF/MISC connector extension cable (SHEX#03A)
RS232C-based driver
Serial cable attached to this kit
Host PC

Typical Connection Using Serial 0 as Debug Serial

8.3. Serial 2/Serial 3

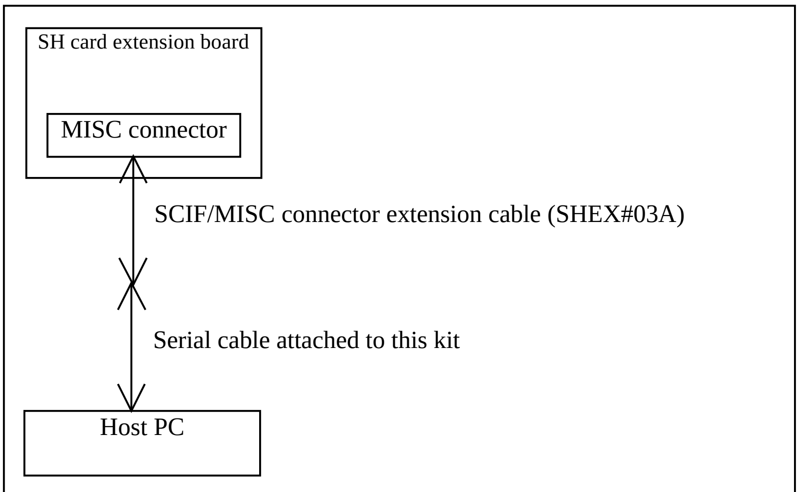
8.3.1. Characteristics

They have FIFO for communication data and the SH card extension board is equipped with an RS232C standard-based driver. They can be used as the Debug Serial by using the serial conversion cable (SHEX#01A) attached to the Evaluation Kit and the serial cable attached to this kit. When Serial 2 is used as the Debug Serial, Serial 3 and Serial 4 are available on Windows CE by this kit. When 2 channels are set to be available by the DevKit, each is assigned to as follows:

- Serial 3 – COM1
- Serial 4 – COM2

8.3.2. Preparing Hardware

Signals of Serial 2 are connected to the SCIF connector on the SH card extension board.
Signals of Serial 3 are connected to the SCIF3 connector on the SH card extension board.
Connect the serial conversion cable (SHEX#01A) to the SCIF or SCIF3 connector on the extension board and then connect it further to the host PC using the serial cable.


SH card extension board
MISC connector
SCIF/MISC connector extension cable (SHEX#03A)
Serial cable attached to this kit
Host PC

Typical Connection Using Serial 2 as Debug Serial

8.3.3. How to Disable Debug Serial and PPSH Function

1. Search the [DEBUG] key in the startup data file.
2. In order to enable the Debug Serial but disable the PPSH function, describe the second parameter as shown below. (Be sure to describe the first parameter as well.) By default, both Debug Serial and PPSH are enabled. (namely, setting 3)

```
[DEBUG]
  X   ;  1st parameter for Debug Serial
  2   ;  Bit0 - 0: Debug Serial enabled
          ;  Bit1 - 1: PPSH disabled
```

3. In order to disable both Debug Serial and the PPSH function, describe the second parameter as shown below.

```
[DEBUG]
  X   ;  1st parameter for Debug Serial
```

```
3    ;  Bit0 -1: Debug Serial disabled
        ;  Bit1 - 1: PPSH disabled
```

9. Power Management

The Power Management has two functions, namely “CPU standby” and “suspend/resume”.

9.1. CPU Standby

The CPU standby function is called when the Windows CE system is judged to be in the idle state (when no thread to be executed by the kernel exists).

With the CPU standby function, the CPU transits to the sleep mode. At this time, the CPU stops but the peripheral modules built in the CPU continue to operate and clocks are continuously output from the CKI0 terminal.

The system returns from the CPU standby state when one of the following events turns active:

- NMI
- IRQ0 (Suspend/Resume button, RTC Alarm, Timer)
- IRQ1 (LAN)
- IRQ2
- IRQ3
- IRQ4
- Reset

9.2. Suspend/resume

The suspend/resume function is called when the SRBTN# signal turns active (when the suspend/resume button is pressed).

When the suspend/resume button is pressed, Windows CE detects the request to suspend, calls power-off operation for each driver and calls power-off operation for HAL in the end to turn the system into the suspend state. In this state, the CPU goes to the standby mode, the peripheral modules built in the CPU stops at the same time as the CPU stops and clock output from the CKI0 also stops.

The state of CARD-E09A during suspension is as follows:

SH7709A

• CPU	Standby mode (peripheral modules built in CPU and clock output from CKI0 stop)
• SDRAM	Self-refresh

SED1355

• SED1355	Hardware suspend
• DAC	IREF off
• VRAM	Self-refresh
• LCD	Off

Companion Chip

• Companion Chip	Standby mode
• Timer Ch0 clock	Stop
• Timer Ch1 clock	Stop
• Timer Ch2 clock	Stop
• Serial 3 clock	Stop
• Serial 4 clock	Stop
• Keyboard interface clock	Stop
• Mouse interface clock	Stop
• PCMCIA A/B/CompactFlash interface clock	Stop

The system returns from the suspend state (resume) when one of the following events turns active:

- NMI
- IRQ0 (Suspend/Resume button, RTC Alarm, Timer)
- IRQ1 (LAN)
- IRQ2
- IRQ3
- IRQ4
- Reset

When the LCD is turned on and off with the suspend/resume function, wait time is provided for the time set by the startup data in the CompactFlash until the LCD becomes stabled.
This wait time parameter exists in “Video mode property data” indicated by the [1355PROPERTY] key in the startup data, for which the fourth parameter is used.

10. Details of Image Creation

This chapter explains how to create CE image files described in Chapter 4.

10.1. CardMax and CardMin

Tool for opening a command prompt for building a CARD-PC CE binary. Sets a 2500-line scroll buffer and environment variables required for building a CARD-PC image.

10.2. BLDopSH

For the Platform Builder, environment variables are set before building in order to change Windows CE image settings. BLDopSH is a tool for enabling this setting through GUI. BLDopSH creates BLDenv.BAT in the same directory to save the settings.
If BLDenv.BAT exists in the same directory as BLDopSH, BLDopSH checks the setting of BLDenv.BAT and displays the contents. Deleting BLDenv.BAT before executing BLDopSH displays the default settings held by BLDopSH.

For BLDopSH, clicking the right button of the mouse on each item or pressing the F1 key displays a help screen. This screen lists environment variables to set. These variables are roughly classified into the following two types.

Standard variables of Platform Builder

ODO_x x x
CEPC_xxx
IMG x x x
For details of the settings, refer to the Platform Builder documentation.

Variables inherent to EPSON SH-CARD

SHC_x x x
The list below shows the environment variables inherent to the SH-CARD. In addition to those set by BLDopSH, some are set by OtherEnv.bat and those not set by either of them are set manually.

Environment variables	Setting	Value	Operation	Setter	Remarks
SHC_SSHELL	*N		SSHELL.EXE is not to be included.	BLDopSH	Note 1
	Y	1	SSHELL.EXE is to be included.		
SHC_32MB_RAM	*N		Sets RAM at 16MB.	BLDopSH	
	Y	1	Sets RAM at 32MB.		Note 2
SHC_BUILTIN_NE2000	*N		Network function built in extension board is disabled.	BLDopSH	Note 3
	Y	1	Network function built in extension board is enabled.		
SHC_PCMCIA_NE2000	N		PCMCIA network card is disabled.	BLDopSH	Note 3
	*Y	1	PCMCIA network card is enabled.		
SHC_NUMCOM	N		No COM port available by CE.	BLDopSH	
	Y	2, 3,4	Indicates the number of COM ports available by CE.		
SHC_NOMOUSE	N		No mouse is connected.	BLDopSH	Note 4
	*Y	1	A mouse is connected.		
SHC_NOCF_ALWAYS	*N		No error message appears at startup even if the CompactFlash does not exist on the SH-CARD.	BLDopSH	Note 5
	Y	1	No error message appears at startup even if the CompactFlash does not exist on the SH-CARD(?? An error message appears at startup if the CompactFlash does not exist on the SH-CARD.).		
SHC_PARALLEL	*N		The parallel function built in SH-CARD is disabled.	BLDopSH	
	Y	1	The parallel function built in SH-CARD is enabled.		
MAXALL2_NOCURSOR	*N		Mouse cursor is displayed.	OtherEnv	
	Y	1	Mouse cursor is hidden.		Note 4

- "*" in the "Setting" column indicates the default settings of the DevKit.

Note 1. SSHELL and FrzDri.DLL are not to be installed. The FrzDri function cannot be executed from an application. To use it, prepare FrzDri.DLL separately.

Note 2. NK.BIN built by assuming 16MB DRAM is compatible also with 32MB DRAM.

Note 3. Since the network function built in the extension board cannot be disabled, set as follows:

```
SHC_BUILTIN_NE2000=1
SHC_PCMCIA_NE2000=
or,
SHC_BUILTIN_NE2000=
SHC_PCMCIA_NE2000=1
```

Note 4. Selecting "Mouse is not connected" sets the mouse cursor to be hidden at the same time. This means that the following is automatically set;

```
SHC_NOMOUSE=
MAXALL2_NOCURSOR=
or,
SHC_NOMOUSE=1
MAXALL2_NOCURSOR=1
```

Note 5. When SHC_NOCF_ALWAYS is set at 1, the CompactFlash built in the SH-CARD is disabled.

10.3. OtherEnv.BAT

Change OtherEnv.BAT for setting an environment variable when installing a driver newly built or changed by the user.

10.4. SaveCON

Saves the contents of the console buffer of the command prompt into a file. To be precise, this is used for saving log information of building the console buffer after BLDDEMO is completed and passing memory map information in it to MapInput.

SaveCON *@Log_file* [/N] [/L: value]

Log_file

File name for saving the contents of the console buffer. Be sure to specify it. Note that the file of the same name, if any, is automatically overwritten. The full path must be specified.

/N

After this option is set, no error message box appears.

/L: *value*

Saves the number of lines specified by "value" from the last line. Unless this option is set, all lines of the console buffer is saved.

10.5. UnInst.BAT

Deletes the installed SH-CARD Windows CE file.

Executing UNINST.BAT without an argument deletes the directory and file created when INSTALL.EXE is executed. If the SH-CARD Windows CE file fails to be installed by

INSTALL.EXE, be sure to execute UNINST.BAT before re-executing this.

By default, UNINST.BAT has the same directory and file names to be deleted as the contents of INSTALL.BAT. When rewriting INSTALL.BAT, be sure to modify UNINST.BAT in the same way.

11. Tools Used on Windows CE

11.1. Configuration Tools

Several tools are available for changing settings of Windows for the CARD-PC on Windows CE.

11.1.1. SEdit

Tool to enable the startup data to be edited on Windows CE. It can also be used simply as an editor.

- Compatible with Unicode and half-size ASCII.
- The upper limit for an available file is 65,535 bytes.
- Only system font is available for the display font.
- A file name is displayed on the title bar.

11.1.2. RegED

Software for editing the registry data on the system with built-in NK.BIN created using the DevKit CD. Use this when the registry is required to be edited for newly installing software, etc. The following explains the menus and operating method of RegEd.

Combo box (top of the screen):

Displays the key name (full path) currently displayed. The separation between keys is indicated with the \ mark. Default open keys (root keys) supported are as follows: Root keys are converted into a truncated form. (characters in parentheses refer to the truncated form)

- HKEY_LOCAL_MACHIINE (HKLM)
- HKEY_CURRENT_USER (HKCU)
- HKEY_CLASSES_ROOT (HKCR)
- HKEY_USERS (HKU)

Key display list (lower left of the screen):

Displays the sub key in the current keys. The contents of the selected sub key are displayed in the value display list on the right. Select an item in the list using the arrow key and press the [GO] button or [Enter] key to open the sub key.

Value display list (lower right of the screen):

Displays the contents of the sub key selected in the key display list. If the list is focused, the current key moves to the sub key selected in the key display list.

Select an item in the list using the arrow key and press the [GO] button or [Enter] key to open the sub key. Select a value and press the [GO] button or [Enter] key to open the value edit dialog.

Display format:
n> val : data
where

n	: Data type (1 character). 5 types (D, S, B, U and -).
Val	: Name of the value.
Data	: Data. DWORD is displayed in the hexadecimal notation. Binary is displayed in the hexadecimal notation by being separated with spaces for each byte. Data type symbols are as follows: -> Sub key D> DWORD S> String B> Binary U> Unknown other than above

Go button: (Alt + G)
Opens the key specified in the combo box, key display list or value display list.
Opens the sub key selected in the key display list.

<= B button: (Alt+B)
Goes back to the previous current key.

F => button: (Alt+F)
Goes forward to the next current key.

< Up button: (Alt+U)
Moves the current key to the next upper key.

M+ button: (Alt+M)
Adds it to “Favorites”. Saves the current key to the RegEd.TXT file in the same directory or adds it to the list in the combo box. When RegEd.EXE is started from the RAM section, RegEd.TXT is not saved until FrzDri is executed.

New button: (Alt+N)
Newly creates a key or value. Creates it under the current key.

Del button: (Alt+D)
Deletes the key or value currently selected.

11.1.3. SSHop

Tool for changing the settings of SSHELL.EXE attached to the DevKit.This enables the following:

- Changing the settings of the size and position of a button
- Registering, changing, deleting and sorting applications

11.1.4. CDATE.EXE

Tool for changing the system date. The date can also be changed using the DEBUGS.BIN command.

- The display and entry format of the date is fixed as “mm/dd/yyyy”.
- If the entered “mm” value of the date is larger than 12, it is forcibly changed to 1.

- If the entered "dd" value of the date is larger than 31, it is forcibly changed to 1. Whether the specified month has 31 or less days is not to be checked.
- If the entered "yyyy" value of the date is smaller than 1998 or larger than 2080, it is forcibly changed to 1999.

11.1.5. CTIME.EXE

Tool for changing the system time. The time can also be changed using the DEBUGS.BIN command.

- The display and entry format of the time is fixed as "hh:mm:ss". The hour is displayed or entered in the 24 hour format.
- If the entered "hh" value of the time is larger than 23, it is forcibly changed to 00.
- If the entered "mm" value of the time is larger than 59, it is forcibly changed to 00.
- If the entered "ss" value of the time is larger than 59, it is forcibly changed to 00.

11.1.6. ALARM.EXE

- Tool for changing the RTC alarm. Setting the RTC alarm enables an RTC alarm interrupt to be generated at a specified time or the system to be resumed if it is suspended. The RTC alarm can also be changed using the DEBUGS.BIN REG command.
-
- The display and entry format of the date is fixed as "mm/dd". The year cannot be entered.
- The display and entry format of the time is fixed as "hh:mm:ss". The hour is displayed or entered in the 24 hour format.
- If the entered "mm" value of the date is larger than 12, it is forcibly changed to 1. If the entered "dd" value of the date is larger than 31, it is forcibly changed to 1. Whether the specified month has 31 or less days is not to be checked.
- If the entered "hh" value of the time is larger than 23, it is forcibly changed to 00. If the entered "mm" value of the time is larger than 59, it is forcibly changed to 00. If the entered "ss" value of the time is larger than 59, it is forcibly changed to 00.
- The date or time item displayed with "xx" indicates that it is invalid (Don't care). Accordingly, the RTC neglects the item displayed with "xx" and outputs an alarm when the settings of the other valid items match the RTC date and time. For the item you want to disable (Don't care), enter "xx" as well.

12. Simplified Debugger

12.1. Simplified Debugger "DEBUGS"

The simplified debugger is a tool to enable CARD-E09A and its extension board hardware to be easily tested. This tool is available in the "Simplified Debugger Starting ATA Card" included in the Evaluation Kit.

12.2. Operating Conditions of Simplified Debugger

In order to execute the simplified debugger, a CRT and a keyboard are required in addition to the Evaluation Kit.

12.3. Method for Starting Simplified Debugger

This section explains how to start the simplified debugger using the ATA card included in the Evaluation Kit. For newly creating an ATA card for starting the simplified debugger, refer to “12.4.5 How to Create Simplified Debugger Starting ATA Card” in this manual.

How to start simplified debugger

1. Installing equipment
   - Connect the following to the extension board.
     (a) CRT
     (b) PS2 keyboard
     (c) Power supply
   - Set the “Simplified Debugger Starting ATA Card” to PC card slot A (further slot from the board) of the extension board.
2. Startup
   - Turn the power on.
   - After turned on, the simplified debugger displays a copy write message on the screen and then simplified debugger prompt “-“.
   - After this prompt is displayed, a simplified debugger command can be entered.
3. Simplified debugger command
   - The following commands are available for the simplified debugger:

① RB, RW, RL commands	(Read byte/word/long word)
② RB, RW, RL commands	(Write byte/word/long word)
③ DB, DW, DL commands	(Dump byte/word/long word)
④ E command	(Edit)
⑤ REG command	(Register editor)
⑥ CLS command	(Clear screen)
⑦ H command	(Help)
⑧ SBR command	(Write to SBR)
⑨ LCD command	(LCD test)

- The following explains operation of each command in details:

① RB, RW, RL commands (Read byte/word/long word)

[Format] RB aaaaaaaa
RW aaaaaaaa
RL aaaaaaaa

[Operation] Displays the data in address “aaaaaaa”.
The RB command is processed in 8-bit units,
the RW command in 16-bit units and
the RL command in 32-bit units.

② WB, WW, WL commands (Write byte/word/long word)

[Format] WB aaaaaaaa dd
WW aaaaaaaa dddd
WL aaaaaaaa dddddddd

[Operation] Records data “dd” in address “aaaaaaaa”.
The WB command is processed in 8-bit units,
the WW command in 16-bit units and
the WL command in 32-bit units.

③ D, DW, DL commands (Dump byte/word/long word)

[Format] D aaaaaaaa
DW aaaaaaaa
DL aaaaaaaa

[Operation] Displays the contents of address “aaaaaaaa” of up to 256 bytes.
The D command is processed in 8-bit units,
the DW command in 16-bit units and
the DL command in 32-bit units.

④ E command (Edit)

[Format] E aaaaaaaa

[Operation] Edits the contents of the memory.
(1) Displays the contents of up to 256 bytes from address “aaaaaaaa”.
(2) With this state, the system enters the standby mode for keyboard entry.
(3) Users can edit the displayed value through key operation.

* Edit key operation

- Arrow keys (→←↑↓): Moves the cursor position.
- Values (0 to 9, A to F) : Enters values.
- Enter key : Records the object data.
- Esc key : Finishes this command.

⑤ REG command (Register editor)

[Format] REG

[Operation] Edits the SH-CARD register setting.
(1) Displays the register setting in the SH-CARD.
(2) With this state, the system enters the standby mode for keyboard entry.
(3) Users can edit the displayed value through key operation.
(4) The registers to be edited are as follows:
(a) Companion chip S3S register
(b) CPU:SH-3 register
(c) LCD/CRT controller 1355 register

* Edit key operation

- Arrow keys (→←↑↓): Moves the cursor position.
- Values (0 to 9, A to F): Enters values.
- Enter key : Records the object data.
- PageDown(Up??)/PageDown: Changes the registers to be edited.
- Esc key : Finishes this command.
- F5 key : Rereads each setting and ??

⑥ CLS command (Clear screen)

[Format] CLS

[Operation]① Clears the characters displayed on the screen,
② Moves the cursor position to the most-upper left.

⑦ H command (Help)

[Format] H

[Operation] Lists commands and their formats.

⑧ SBR command

[Format] SBR SBR.BIN drive
drive A: Slot A

B: Slot B
C: CompactFlash

[Operation] Records the _SBR.BIN_ file recorded in the ATA card at startup into sector #2 or latter sectors of drive “drive”.

- For usage of this command, refer to “12.4.5 How to Create Simplified Debugger Starting ATA Card” in this manual.

- This command performs the following, the results of which are displayed on the screen.

(a) Loading files
(b) Writing
(c) Verifying

⑨ LCD command

[Format] LCD

[Operation] •LCD/CRT controller SED1355 display test tool

- With this command, display can be tested using an optional SED1355 parameter.

[Basic operation of LCD command]

- This command is equipped with a parameter buffer for setting SED1355 parameters. In order to set a parameter to be displayed and tested, set the data in this parameter buffer.
- During display test, the parameter buffer value is recorded in SED1355 and then the test pattern is displayed.
- After the test, the display is switched to the CRT single, 640- x 480-pixel display.

[Details of LCD command]

- Executing the LCD command displays the following menu.

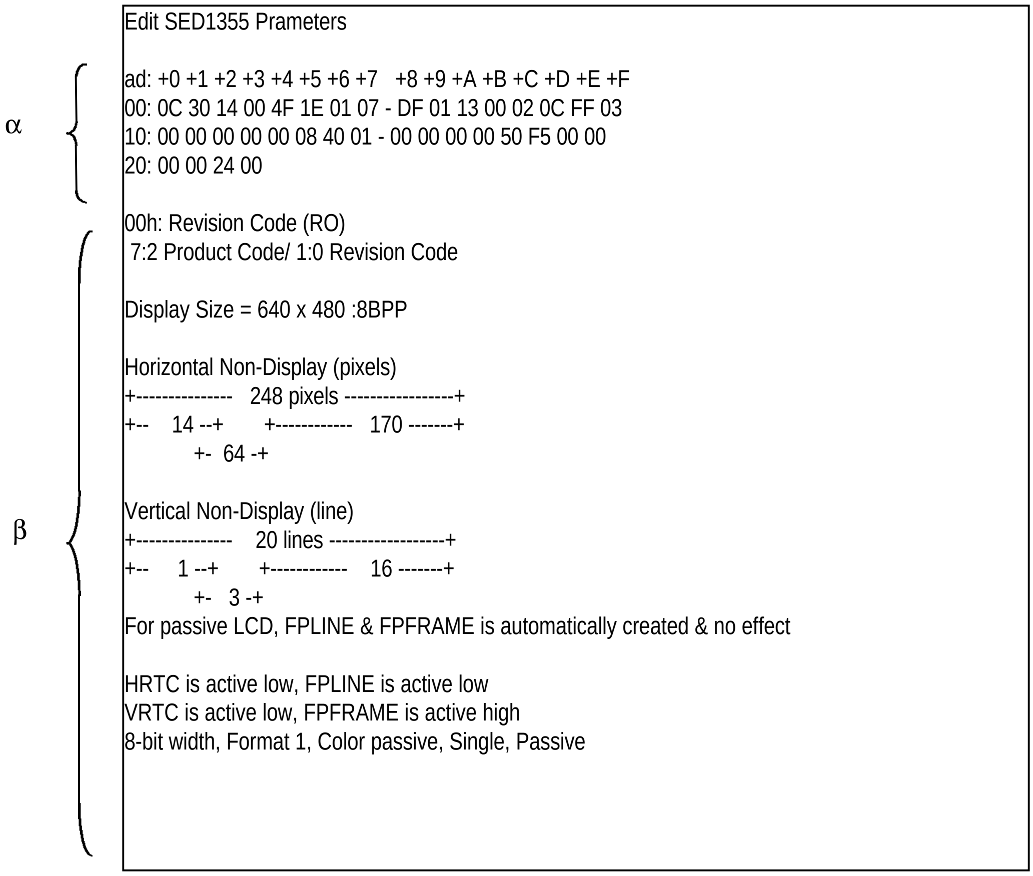
```
LCD Display Test
1. Edit parameters
2. Display
3. Video Test
A. Set default parameter (CRT, 16bpp)
B. Set default parameter (CRT, 15bpp)
C. Set default parameter (CRT, 8bpp)
D. Set default parameter (CRT, 4bpp)
E. Set default parameter (CRT, 2bpp)
F. Set default parameter (CRT, 1bpp)
G. Set default parameter (CRT, 800x600,8bpp)
Q. Quit
H. Show help
```

On this menu, select a desired operation to execute it.
Enter “1”, for example, to execute “Edit parameters”.
Details of each operation are described on the following page.

(1) Edit parameters

- This function is used to edit the parameter buffer.

Executing this function displays the following screen:



```
Edit SED1355 Prameters

ad: +0 +1 +2 +3 +4 +5 +6 +7   +8 +9 +A +B +C +D +E +F
00: 0C 30 14 00 4F 1E 01 07 - DF 01 13 00 02 0C FF 03
10: 00 00 00 00 00 08 40 01 - 00 00 00 00 50 F5 00 00
20: 00 00 24 00

00h: Revision Code (RO)
 7:2 Product Code/ 1:0 Revision Code

Display Size = 640 x 480 :8BPP

Horizontal Non-Display (pixels)
+---------------  248 pixels -----------------+
+--   14 --+      +------------  170 -------+
         +-  64 -+

Vertical Non-Display (line)
+---------------   20 lines ------------------+
+--    1 --+      +------------   16 -------+
         +-  3 -+
For passive LCD, FPLINE & FPFRAME is automatically created & no effect

HRTC is active low, FPLINE is active low
VRTC is active low, FPFRAME is active high
8-bit width, Format 1, Color passive, Single, Passive
```

- The upper section on the screen (the area marked α in the above figure) is a parameter buffer edit screen.
- Set index 0h to 23h of the SED1355 register here.
- Edit it using “1” to “9”, “A” to “F” and arrow keys.
- Press the “Esc” key to finish operation.
- By default, the 640 x 480-pixel, 8BPP, CRT single unit display is set.

- The lower section on the screen (the area marked β in the above figure) shows the meanings of the parameters set in the parameter buffer.
- These contents are updated after the values in the parameter buffer are changed.

* For the meaning of each parameter, refer to the SED1355 Technical Manual.

(2) Display

- This function is used for display test.
- It displays a test pattern after writing the data set by “(1) 1. Edit parameters” explained above into the SED1355 register in the SH-CARD and executing the LCD display sequence.
- Press the “Esc” key to finish operation.

(3) Video Test

- This function is also used for display test just like “(2) 2. Display” explained above.
- Press the “Esc” key to finish operation.

- During this test, different test patterns are displayed on both screens 1 and 2 of SED1355.

(4) A.Set default parameter (CRT, 16bpp) /
B.Set default parameter (CRT, 15bpp) /
C.Set default parameter (CRT, 8bpp) /
D.Set default parameter (CRT, 4bpp) /
E.Set default parameter (CRT, 2bpp) /
F.Set default parameter (CRT, 1bpp) /
G.Set default parameter (CRT, 800x600,8bpp)

- These functions record standard settings in the parameter buffer.
- The amount to edit the parameter buffer can be reduced by executing this function before executing "(1) 1. Edit parameters" explained above.
- All are set to "CRT single display".

(5) Q.Quit

- Quits this command.

(6) H.Show help

- Displays descriptions of key operation on the parameter buffer edit and text display screens.

12.4. How to Create Simplified Debugger Starting ATA Card

The method for creating an ATA card for starting the simplified debugger is the same as the method described in "Backing up ATA Card" in Chapter 2.

13. Appendix

13.1. Overview of Startup Data

Startup data is a collection of parameters specifying operation of various devices and programs. By changing a parameter, the video initialize parameter or a logo file being started can be changed.
For the parameter numbers In this manual, the first value or string is assumed to be "0".

13.1.1. Definition

Word	Definition
Startup data file	Text file generated by collecting startup data.
Startup data	Collection of parameters specifying operation of various devices and functions. Consists of 3 elements, namely, keys, parameters and comments.
Key	String indicating the type of startup data. Enclosed in []. Only one key is included at the head of each startup data.
Parameter	Collection of values or strings actually specifying operation. A parameter is not necessarily required but more than one parameter is allowed to exist at the same time.
Numeric parameter	Unsigned hexadecimal parameter of up to 8 digits. Both uppercase and lowercase alphabets are available.
String parameter	Characters enclosed in “”.
Comment	The part from “;” to the end of the line is assumed to be a comment and neglected by the loader. If “;” appears in a string parameter, however, it is assumed to be a part of the string and not to be a comment.

13.2. Video Initialize Data

Video initialize data is classified into two types and utilized as follows:

- Used by the loader to initialize SED1355.
- Used by the Windows CE video driver to understand the property of the video mode.

13.2.1. SED1355 Initialize Data

SED1355 initialize data is a collection of parameters used by the loader to initialize SED1355.

- **Format**

The key is “1355”. The table below explains the parameters in detail.

Parameter No.	Meaning
0, 1	Initialize register address 1, initialize register data 1.
2n+2, 2n+2	Initialize register address n, initialize register data n.
...	...
2m+2, 2m+2	FFFFFFFF,FFFFFFFF. For both initialize register address and data, “FFFFFFFF” indicates the end of initialize data.

- **Initialize Register Address**

SED1355 register n is located in address 0xb4000000 + n. Accordingly, the initialize register address is “B40000001”, for example, for register 1.

- **Initialize Register Data**

Since the size of the SED1355 register is 1 byte, the initialize register data is a value of 1 byte. If a value of 2 or more bytes is written, the upper byte is neglected.

● Special Control

When the initialize register address is “FFFFxxxx”, special operation can be specified according to the value of the initialize register data.

Initialize register address	Meaning
FFFF0000	Makes the system wait for the time (mS) specified by the initialize register data. While waiting, other initialization (e.g. loading of nk.bin) is executed.
FFFF0001	Clears the screen.
FFFF0002	Switches the SED1355 operation clock to an external clock.
FFFF0003	Loads the logo file to the VRAM.
FFFF0004	Makes the system wait for the frame count specified by the initialize register data. Calculates the counts of the time when the vertical display period shifts to the vertical non-display period based on loops and makes the system exit from the loop when the last vertical non-display period starts. While waiting, other initialization (e.g. loading of nk.bin) is not executed.
FFFF0005	Turns FPVCCON high.
FFFF0000	Makes the system wait for the time (mS) specified by the initialize register data.
FFFFFFFF	Stops initialization.

● Converting SED1355 Register Value to Initialize Parameter

This section explains a general method for creating an initialize parameter from the SED1355 register setting.

Since the initialize parameter describes not only the SED register setting but also the setting order, the screen may be distorted during initializing SED1355 if the register setting is written as it is. For example, setting registers in an ascending order of the register numbers mistakenly allows display signals to be output before a parameter required for display.

Generally, such situation can be avoided by following the rules listed below.

Rule No.	Rule
1	Keep bit 1-0 of register 0Dh (CRT Enable and LCD Enable) at 0 until all other parameters are set.
2	When selecting LCD display, set bit 7 of register 23h (Display FIFO Disable) at 1 at first and then to 0 after waiting for a time of 128 frames.
3	Clear the screen before turning bit 1-0 of register 0Dh (CRT Enable and LCD Enable) to 1.

13.2.2. Video Mode Property Data

This data is used by the Windows CE video driver to understand the property of the video mode.

- **Format**

The key is “1355PROPERTY”. The table below explains the parameters in detail.

Parameter No.	Meaning
0	Dots in the horizontal direction.
1	Dots in the vertical direction.
2	Bits per dot.
3	Frame rate (frame/sec)
4	Time to make video display wait at the time of resuming from suspension (mS).

- **Example**

The following shows an example of the initialize data for 640- x 480-dot, 8bpp video.

```
[1355PROPERTY]
     280 1E0          ;Width and Height in dots.
     8 4B             ;8bpp, 75 frames per a second.
     BB8                   ;Wait video for 3 seconds at resume.
```

13.3. Logo Display Data

Specifies the logo file to be displayed on the screen during loader operation and the position of the display.

13.3.1. Logo File Specifying Data

Specifies the BMP file to be displayed as a logo.

- **Format**

The key is “LOGOFILE”. The table below explains the parameter in detail.

Parameter No.	Meaning
0	Logo file name (string)

- **Example**

The following shows an example of a logo file specifying parameter.

This example specifies a file named LOGOFILE.BMP.

```
[LOGOFILE]
     logofile.bmp
```

13.3.2. Logo Position Specifying Data

Specifies the position for specifying a logo file.

- **Format**

The key is “LOGOXY”. The table below explains the parameters in detail.

Parameter No.	Meaning
0	Horizontal position of the left end of a logo image. The image is positioned at the center if “FFFF” is set.
1	Vertical position of the upper end of a logo image. The image is positioned at the center if “FFFF” is set.

- **Example**

The following shows an example of logo position specifying data. This example specifies the logo image to be positioned at the center on the screen.

```
[LOGOXY]
     FFFF FFFF
```

13.4. Debugger Specifying Data

Specifies a debugger execution file. Only when a debugger is specified, the loader loads the debugger from CF to the memory to execute it.

- **Format**

The key is “DEBUGGER”. The table below explains the parameter in detail.

Parameter No.	Meaning
0	Debugger file name (string)

- **Example**

The following shows an example of debugger specifying data. This example specifies the DEBUGS.BIN file as a debugger.

```
[DEBUGGER]
    “debugs.bin”
```

13.5. Debug Function Control Data

- **Format**

The key is “DEBUG”. These parameters control both NKLOADS and HAL. The table below explains the parameters in detail.

Parameter No.	Meaning
0	Specifies a channel number of the Debug Serial. 0 : Serial 0 2 : Serial 2 3 : Serial 3 (Default) Other numbers are not available.
1	Debug function control parameter

- **Debug Function Control Parameter**

Bit	Meaning
0	Enables or disables the Debug Serial. 0 : Enables (Default) 1 : Disables Unless the Debug Serial is disabled, Windows CE outputs debug information to the Debug Serial. When the Debug Serial is disabled, the debug serial function of Windows CE is disabled.
1	Enables or disables PPSH. 0 : Enables (Default) 1 : Disables Unless PPSH is disabled, the function of NKLOADS to download NK.BIN via the parallel port is enabled and Windows CE uses the port for debugging. When PPSH is disabled, NKLOADS does not come in contact with the parallel port and Windows CE is prohibited from using the port for debugging.
2-31	Reserved.

- **Example**

The following shows an example of a debug function control parameter. This example specifies Serial 3 to be used as the Debug Serial and disables the PPSH function.

```
[DEBUG]
    3
    00000002
```

13.6. Touch Panel Driver Control Data

- **Format**

The key is "TOUCHPANEL". These parameters control the Windows CE touch panel driver.
The table below explains the parameters in detail.

Parameter No.	Meaning
0	Threshold for judging whether the stylus comes in contact with the touch panel or not. Set a large value if outputs from the touch panel to ADC are relatively high or a small value if outputs are relatively low. Values between 8 and 1023 are available. If an invalid value is entered, the default value (512) is set.
1	Time from when ADC input is switched to when the potential is stabled (unit: us). Used for detecting the X and Y coordinates. Values between 100 and 1148 are available. If an invalid value is entered, the default value (500) is set.
2	Time from when ADC input is switched to when the potential becomes stabled (unit: us). Used for detecting stylus contact. Values between 100 and 1148 are available. If an invalid value is entered, the default value (400) is set.

Parameter Nos. 0 and 1 are assumed to exist when there are two parameters or No. 0 when only one parameter exists. If information of each parameter is not available, the default value is set. If the key itself does not exist, all parameters are assumed not to exist. So, operation does not change in the following cases.

- No key exists in the startup data (the startup data is the same as before).
- No parameter available although a key exists.
- Parameters are written, but out of the predefined range (for example, all are 0).

- **Example**

The following shows an example of touch panel driver control data.
This example specifies the default value.

```
[TOUCHPANEL]
    200 1F4 190
```

13.7. FrzDri Option Control Data

- **Format**

The key is "FRZDRIOPTION". This parameter controls the FrzDri function. The table below explains the parameter in detail.

Parameter No.	Meaning
0	FrzDri control parameter

- **FrzDri Control Parameter**

Bit	Meaning
0	Specifies the file to restore. 0 : “NK.FZD” 1 : “NK0.FZD”
1	Specifies whether to use another file to restore when restoration has failed. 0: Not use 1: Use When “1: Use” is selected, restoration starts from NK0.FZD if “0: NK.FZD” is specified by bit 0 or from NK.FZD if “1: NK0.FZD” is specified.
2	Specifies operation after restoration has failed. 0: Clears the memory and reboots the system 1: Halts the system. When “0: Clears the memory and reboots the system” is selected, the contents of the memory is cleared and the system is rebooted after FrzDri has failed to be resumed. This initializes all data located in the RAM like registries and files. When “1: Halts the system” is specified, the system is halted after FrzDri has failed to be resumed.
3	Specifies the function of the manual reset button. 0 : Warm boot 1 : FrzDri When “0: Warm boot” is selected, the system is reset with the contents of the memory held after the button is pressed. When “1: FrzDri” is selected, the FrzDri function is activated when the button is pressed.
4-31	Reserved.

- **Example**

The following shows an example of FrzDri option driver control data. This example specifies restoration from the NK.FZD file and, if fails, the memory to be cleared to reboot the system. It also specifies the FrzDri function to be activated by RESETM# signals.

```
[FRZDRIOPTION]
   08
```

13.8. Test Level Control Data

- **Format**

The key is “TESTLEVEL”. This parameter controls the level of the hardware test to be performed by NKLOADS. The table below explains the parameter in detail.

Parameter No.	Meaning
0	Specifies the level of the memory test. 0 : Easy test (test with only one pattern. It ends in a short period of time.) 2 : No test. The memory is cleared. 3 : No test. The memory is not cleared. 4 : Full test (test with five patterns. This takes a longer time but enables the memory to be tested precisely.)

- **Example**

The following shows an example of test level control data.
This example specifies the full test of the memory.

```
[TESTLEVEL]
  4
```

13.9. FrzDri.DLL

- **What is FrzDri.DLL?**

FrzDri.DLL is a standard function library for utilizing the FrzDri function, an SEIKO EPSON's extended function for Windows CE, from an application. Using this FrzDri.DLL library enables the FrzDri function to be safely called from an optional Windows CE application.

- **Prototype and various definitions**

Frzdri.DLL is described in the C language. Note the prototype declaration when calling it from C++. For the prototype declaration and various definitions, refer to the following header file example.

Header file example

```
#define FZD_MODULE_DLL (1)
#define FZD_MODULE_LOADER (2)

#define FZD_COLDBOOT (1)
#define FZD_WARMBOOT (2)
#define FZD_BACKUP   (3)
#define FZD_SOFTOFF  (5)
#define FZD_SUSPEND  (15)
#define FZD_FNC_NUM  (16)

#if defined __cplusplus
extern "C" {
#endif
__declspec( dllexport ) BOOL FrzDri(UINT);
__declspec( dllexport ) DWORD FzdGetVer(UINT);
__declspec( dllexport ) BOOL FzdCheckFnc(UINT);
#if defined __cplusplus
}
#endif
```

FrzDri function

For each function of FrzDri, refer to "7.4 FrzDri Function".

- **Export function**

The following describes the functions to be exported by FrzDri.

Function: BOOL FrzDri (UINT request)

Usage : Calls the FrzDri function specified by "request".
Returned value: "FALSE" if FrzDri fails or "TRUE" on success.
Constant: The following are defined in FrzDri.h as the values to be set for "request".

FZD_COLDBOOT	: Requests cold boot.
FZD_WARMBOOT	: Requests warm boot.

FZD_BACKUP	: Requests backup.
FZD_SOFTOFF	: Requests soft-off.
FZD_SUSPEND	: Requests suspension.

NOTICE: Depending on the function of FrzDri, the system may not return from this function since it shuts down the system.

Function: DWORD FzdGetVer (UINT module)

Usage : Obtains the version of the module specified by “module”.

Returned value: Major version * 100 + minor version of the specified module.
(e.g. “123” is returned for version 1.23)

Constant: The following are defined in FrzDri.h as the values to be set for “module”.

FZD_MODULE_DLL	: Returns the version of FrzDri.DLL.
FZD_MODULE_LOADER	: Returns the version of FrzDri.DLL.

NOTICE: “100” (version 1.00) is returned if FZD_MODULE_LOADER is specified when Windows CE activated by the loader with no version information.

Function: BOOL FzdCheckFnc (UINT request)

Usage : Checks support availability of the FrzDri function specified by “request”.

Returned value: “TRUE” if supported or “FALSE” if not supported.

Constant : The values to be set for “request” are the same as for the FrzDri (request) function.

● Precautions on CEPC (usage of shut down function)

On CEPC, the shut-down function in HAL of the DevKit used is called by specifying as follows for FrzDri.DLL.

```
[HKEY_CURRENT_USER\Software\Epson\FrzDri]
 "Shutprocess"=dword:1
```

The above registry is automatically set for usage on the DevKit for CEPC.

If the above key does not exist, FrzDri.DLL forcibly resets the system by calling the standard reset code in HAL. Although the FrzDri function is enabled even in this state, forcible reset is not recommended since it may destroy data.

● Typical usage of FrzDri.DLL

It is useful to prepare the following function, for example, in order to call the FrzDri function using FrzDri.DLL from a program written in the C language.

```
#include "FrzDri.h"

BOOL DoFrzDri(UINT request)
{
// This function is a sample code to summon FrzDri function via FrzDri.DLL library.
// The 'request' values are defined on the Frzdri.h header file.

HMODULE hLib;
FARPROC pfnFrzDri;
BOOL (*FrzDri)(UINT);
BOOL ret = FALSE;
```

```
          hLib = LoadLibrary(TEXT("FrzDri.DLL"));
          if (!hLib)
                    DEBUGMSG(1, (TEXT("FrzDri.DLL loading error\r\n")));
          else
          {
                    pfnFrzDri = GetProcAddress(hLib, TEXT("FrzDri"));
                    if (!pfnFrzDri)
                              DEBUGMSG(1, (TEXT("Invalid FrzDri.DLL\r\n")));
                    else
                    {
                              FrzDri = (BOOL (*)(UINT))pfnFrzDri;
                              if (!FrzDri(request))
                                        DEBUGMSG(1, (TEXT("Failed FrzDri request\r\n")));
                              else
                                        ret = TRUE;
                    }
                    FreeLibrary(hLib);
          }
          return ret;
}
```

EPSON International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
1960 E. Grand Avenue
El Segundo, CA 90245, U.S.A.
Phone : +1-310-955-5300 Fax : +1-310-955-5400

SALES OFFICES
West
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone : +1-408-922-0200 Fax : +1-408-922-0238

Central
101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone : +1-815-455-7630 Fax : +1-815-455-7633

Northeast
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone : +1-781-246-3600 Fax : +1-781-246-5443

Southeast
3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone : +1-877-EEA-0020 Fax : +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15
80992 Muenchen, GERMANY
Phone : +49- (0) 89-14005-0 Fax : +49- (0) 89-14005-110

- GERMANY -
SALES OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone : +49- (0) 217-15045-0 Fax : +49- (0) 217-15045-10

- UNITED KINGDOM -
UK BRANCH OFFICE
2.4 Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone : +44- (0) 1344-381700 Fax : +44- (0) 1344-381701

- FRANCE -
FRENCH BRANCH OFFICE
1 Avenue de l' Atlantique, LP 915 Les Conquerants
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE
Phone : +33- (0) 1-64862350 Fax : +33- (0) 1-64862355

ASIA

- CHINA -
EPSON (CHINA)CO.,LTD.
28F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing,CHINA
Phone : 64106655 Fax : 64107320

SHANGHAI BRANCH
4F, Bldg., 27, No. 69, Gui Jing Road
Caohejing, Shanghai, CHINA
Phone : 21-6485-5552 Fax : 21-6485-0775

- HONG KONG, CHINA -
EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, HONG KONG
Phone : +852-2585-4600 Fax : +852-2827-4346
Telex : 65542 EPSCO HX

- TAIWAN, R.O.C. -
EPSON TAIWAN TECHNOLOGY & TRADING LTD.
10F, No. 287,Nanking East Road, Sec. 3
Taipei, TAIWAN, R.O.C.
Phone : 02-2717-7360 Fax : 02-2712-9164
Telex : 24444 EPSONTB

HSINCHU OFFICE
13F-3, No.295, Kuang-Fu Road, Sec. 2
HsinChu 300, TAIWAN, R.O.C.
Phone : 03-573-9900 Fax : 03-573-9169

- SINGAPORE -
EPSON SINGAPORE PTE., LTD.
No. 1 Temasek Avenue, #36-00
Millenia Tower, SINGAPORE 039192
Phone : +65-337-7911 Fax : +65-334-2716

- KOREA -
SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-Dong
Youngdeungpo-Ku, Seoul, 150-010, KOREA
Phone : 02-784-6027 Fax : 02-767-3677

- JAPAN -

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

Electronic Device Marketing Department
Card Products & Marketing Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-7503 Fax: +81-(0)42-587-8423

ED International Marketing Department I
(Europe & U.S.A.)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5812 Fax: +81-(0)42-587-5564

ED International Marketing Department II (Asia)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5110