



Intel[®] StrongARM^{*} SA-1110 Development Platform - Windows^{*} CE Software

User's Guide

July 2000

Release 1.0



Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

These procedures may contain errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

Copyright © Intel Corporation, 2000

*Third-party brands and names are the property of their respective owners.

Contents

1.0	Overview	5
1.1	Description	5
1.2	Requirements	5
1.2.1	Prerequisites	5
1.2.2	Related Documentation	5
1.2.3	Software Requirements	6
1.2.4	Host System Requirements	6
1.2.5	Target Requirements:	6
1.2.6	Network Requirements	6
2.0	Building the Microsoft Windows* CE Operating System	7
2.1	Installing Microsoft Windows* CE Platform Builder	7
2.1.1	Configuring the Windows* CE Build Environment	8
3.0	Validating the Microsoft Windows* CE Platform Builder Installation... 8	
3.1	Validating Proper SA-1110 Target Installation	9
4.0	Installing the SA-1110 Reference Design Windows* CE Driver Kit	9
5.0	Building an Intel® StrongARM* Development Board Windows* CE Image under Platform Builder	10
6.0	Modifying the Build	11
6.1	Environment Variables	11
6.2	Building for the SA-1110 Development Platform	13
6.3	Building for the SA-1110 and SA-1111 Development Boards	13
6.4	Running Remote Programs with SA-1110 and SA-1111 Development Boards Windows* CE MAXALL Builds	14
7.0	Loading the Windows* CE Image via the Ethernet Boot Loader	15
7.1	Load and Launching Files	17
7.2	Adding an Application to the your Image	17
7.3	Debugging an Application using a Kernel Debugger	18
7.3.1	Debugging an Application Using Ethernet Messaging	18
7.3.2	Debugging a Driver with Ethernet Messaging	19

Figures

1	Network Configuration Alternatives	7
---	--	---

1.0 Overview

This document describes how to use the Microsoft Windows* CE Platform Builder with the Intel® StrongARM* SA-1110 Development Platform (SA-1110 Development Platform) to build the Windows CE operating system for the SA-1110 Development Platform. This document also describes how to install and validate the Windows CE Platform Builder installation, and how to build an image and use the E-boot loader.

Note: This document and the Intel® StrongARM* SA-1110 Development Platform Windows* CE software kit are for the Release 1.0 build of this product. For the latest information and updates, see the hardware release notes that are provided in hardcopy format, and the software readme.txt files that are provided in the software kits.

1.1 Description

The SA-1110 Reference Design Windows CE Driver Kit contains example drivers for the Intel® StrongARM* SA-1110 Development Board (SA-1110 Development Board) and the Intel® StrongARM* SA-1111 Development Module (SA-1111 Development Module) for use with Windows CE Platform Builder. Although these drivers have had initial testing, they are only *examples* and cannot be used for production systems. For warranty and support information, see the license agreement when you install the driver kit.

1.2 Requirements

This section describes the requirements for installing Windows CE Platform Builder and building the Windows CE operating system for the SA-1110 Development Platform.

Note: The SA-1100 selection in the Windows CE Platform Builder supports the SA-1110 Development Board and the SA-1111 Development Module.

1.2.1 Prerequisites

The following list describes the prerequisites for users:

- Users of this software should be limited to programmers who have experience with Windows CE driver development and porting.
- Users must review the readme file(s) and release notes that are provided with the SA-1110 Development Board and the SA-1111 Development Module before proceeding with the installation.
- Users must review the *Intel® StrongARM* SA-1110 Development Board User's Guide*, order number 279278, and the Windows CE Platform Builder documentation.

1.2.2 Related Documentation

For related documentation, see the documents listed in the *Intel® StrongARM* SA-1110 Development Board User's Guide*, order number 279278.

1.2.3 Software Requirements

The SA-1110 Reference Design Windows* CE Driver Kit¹ is required to build the Windows CE operating system for the SA-1110 Development Platform.

1.2.4 Host System Requirements

Intel requires at least 1 GB of free hard drive space for unzipping files. For the Windows CE operating system build requirements, see the Windows CE Platform Builder documentation.

1.2.5 Target Requirements:

In addition to the host system requirements listed in [Section 1.2.2](#), the following requirements are necessary to build the Windows CE operating system for the SA-1110 target platform:

- SA-1110 Reference Design Windows CE Driver Kit
- An SA-1110 Development Board
- An SA-1111 Development Module (optional)
- A Socket Communications Compact Flash LP-E CF+ card (provided with the SA-1110 Evaluation Board kit)

Note: If using the SA-1111 Development Module, a Compact Flash to PCMCIA adapter is also required.

1.2.6 Network Requirements

To use the Ethernet boot application to download the Windows CE image or to do Ethernet debugging, the host and target systems must choose **one** of the network configurations listed below:

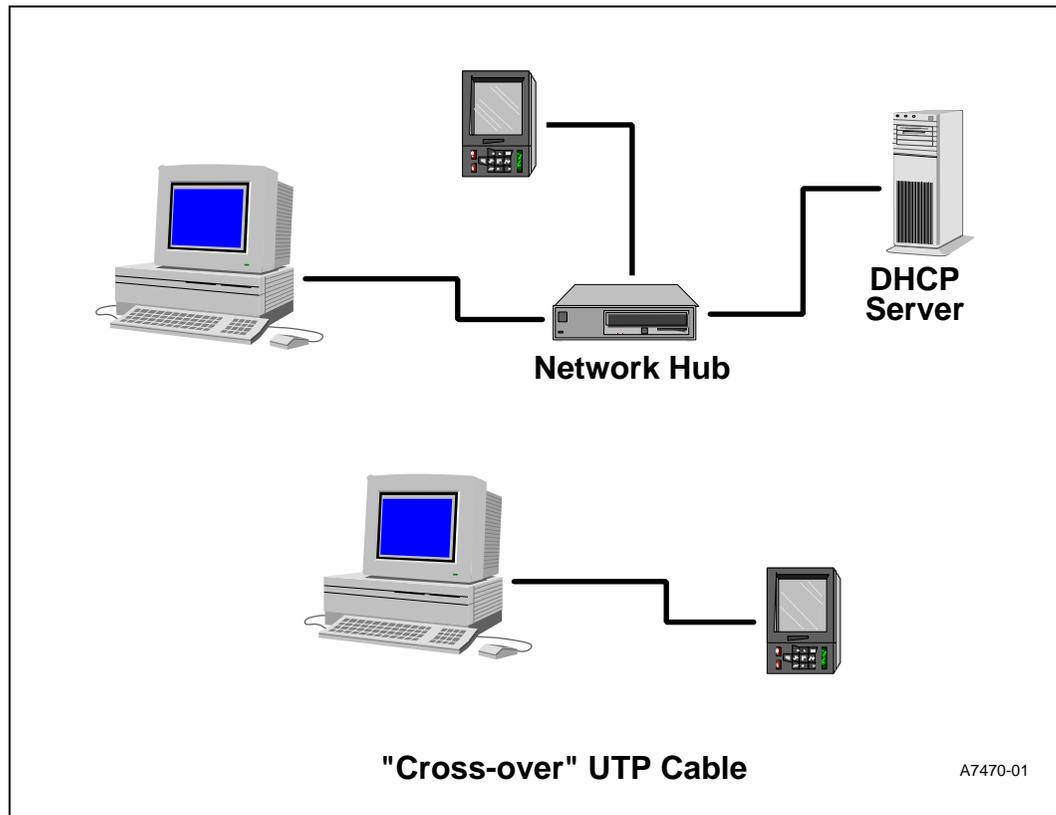
- A network HUB with a DHCP server
- A 10BASE-T unshielded twisted pair (UTP) “cross-over” Ethernet cable to go between the host and SA-1110 Development Board (provided with the SA-1110 Development Board kit)

The amount of network traffic often present on a corporate network HUB can impede the downloading efforts. To avoid network traffic, use of the above mentioned 10BASE-T UTP “cross-over” Ethernet cable is recommended.

For locating ports and connections, see the *Intel[®] StrongARM* SA-1110 Development Board User's Guide* and the *Intel[®] StrongARM* SA-1111 Development Module User's Guide*.

1. To obtain the SA-1110 Reference Design Windows* CE Driver Kit, contact your local Intel sales office.

Figure 1. Network Configuration Alternatives



2.0 Building the Microsoft Windows* CE Operating System

This section describes how to use the Windows CE Platform Builder and the SA-1110 Windows CE Platform Builder Kit to build the Windows CE operating system for the SA-1110 Development Platform.

2.1 Installing Microsoft Windows* CE Platform Builder

Follow the instructions in the Microsoft Windows CE Platform Builder documentation to install the Microsoft Windows CE Platform Builder.

Note: To save disk space, only the StrongARM platform (denoted as SA-1100 in Platform Builder) is required (the MIPS, HITACHI, ARM* 7xx and other platforms are not necessary).

To install both the Platform Builder development tools and the Windows CE Operating System build environment:

1. Choose: Install the Windows CE Operating System and Development Tools
2. Follow the on-screen defaults for the installation directory settings.

2.1.1 Configuring the Windows* CE Build Environment

When selecting the type of installation desired, choose the “Custom” button and configure the Windows CE build environment for the SA-1110 configuration in the following manner:

Note: The Platform Builder SA-1100 configuration supports the SA-1110.

1. Select Windows CE EDK and install all options.
2. Select Windows CE Binaries and install only the SA-1100 binaries.
3. Select Windows CE Sample and Demos and install only the following options:
 - Demonstration Projects
 - SA-1100 Demonstration Project Binaries
 - Hardware Development Platforms
4. Select Windows CE SDK (OEM Version) and install only the following options:
 - Windows CE SDK (OEM Version)
 - ARM SA1100 Support
5. When prompted with the Windows CE Debug Shell Tool (CESH) dialog, choose to install it with the following response:

```
Install CESH
```

Note: If you choose **not** to install CESH (the default), then your installation of Platform Builder will not be set up for Ethernet debugging.

6. Follow the instructions on the screen.

3.0 Validating the Microsoft Windows* CE Platform Builder Installation

Use the following procedure to validate that the Windows CE Platform Builder is properly installed:

1. Open a DOS* command prompt window.
2. Enter the following command to confirm that the environment variable `_WINCEROOT` exists and is properly assigned.

```
set _WINCEROOT
```

This environment variable specifies the fully-qualified path of the directory in which the Platform Builder is installed. If the set command returns `_WINCEROOT=<fully qualified path>`, the Platform Builder is installed correctly (for example `_WINCEROOT= C:\wince212.`)

If the set command returns the error message:

```
Environment variable _WINCEROOT not defined.
```

Then either the development workstation did not restart following setup, or the setup application did not complete successfully. If the environment variable is not defined, uninstall and reinstall Platform Builder.

3.1 Validating Proper SA-1110 Target Installation

To validate the Platform Builder has been properly installed for the SA-1110 target platform, perform the following procedure:

1. Choose the Programs menu from the Start menu button.
2. Select the ARM Tools option from Microsoft Windows CE Platform Builder menu.
3. Invoke the selection Build Minshell for SA-1100. A DOS window appears.
4. Type the following text in the DOS windows and wait for the build to complete:

```
blddemo
```

If the Windows CE Platform Builder for the SA-1110 Target platform was installed properly, the build will complete successfully with the message:

```
MINSHELL build complete.
```

If the selection Build Minshell for SA_1100 does not exist, or the build did not complete successfully, verify that all shortcuts and directory names were named properly and reinstall the Platform Builder ensuring that the Platform Builder Windows CE Operating System is customized for the SA-1110 target platform.

4.0 Installing the SA-1110 Reference Design Windows* CE Driver Kit

The SA-1110 Reference Design Windows CE Driver Kit consists of the following items:

- OEM adaption layer for the SA-1110 Development Board and the SA-1111 Development Module.
- Sample drivers source code for the SA-1110 Development Board and the SA-1111 Development Module.
- Readme.txt file containing in-depth details of which drivers are provided and any known problems that currently exist with the kit.

A set of sample I/O drivers are provided and available for developers using Windows* CE. See your Intel sales representative to obtain a copy of this kit.

Use the following procedure to install the SA-1110 Reference Design Windows CE Driver Kit on your host system:

1. Double-click on the self-contained executable file containing the SA-1110 Reference Design Windows CE Driver Kit and follow the instructions to place the ZIP file in a directory on the host system that you have chosen.
2. Unzip the ZIP file containing the SA-1110 Reference Design Windows CE Driver Kit.
3. Read the readme file for in-depth details concerning the kit contents.
4. Copy the SA11X0BD directory contents into the platform builder directory, *WINCEROOT*\Platform, where *WINCEROOT* is the root directory of the Windows CE installation (for example C:\wince212). This sets up the platform builder with a new target platform, namely the SA-1110 Development Board.

5. Set up a Command Prompt Build window for the SA-1110 target by creating a shortcut for building a MAXALL image for the SA-1110 development platform:
 - a. From the Start menu, choose the Setting menu and then choose Task bar.
 - b. Select the Start Menu Programs tab.
 - c. Select the Advanced... tab and navigate to:
WINNT\Profiles\All Users\Start Menu\Programs\Microsoft Windows CE Platform Builder\ARM Tools.
 - d. Copy the Build Minshell for SA-1110 shortcut and paste it into the ARM Tools directory.
 - e. Change the name of the shortcut to identify the new platform to be built:
Build Maxall for Intel SA-1110 Dev Platform.
 - f. Right click on the newly named shortcut and select Properties.
 - g. Select the Shortcut tab and click in the Target field.
 - h. Move your cursor to the end of the field and replace MINSHELL with MAXALL and replace ODO with SA11X0BD. Assuming that the platform builder was installed to drive C, the edited target test string should appear as:

```
C:\WINNT\system32\CMD.EXE /k C:\WINCEROOT\public\common\oak\misc\wince.bat
ARM SA1100 CE MAXALL SA11X0BD
```

where *WINCEROOT* is the root directory of the Windows CE installation.

- i. Apply your changes and close the windows.

You are now ready to build the Windows CE MAXALL image for the SA-1110 Development Platform.

5.0 Building an Intel® StrongARM* Development Board Windows* CE Image under Platform Builder

Use the following procedure to build the Windows CE drivers under Platform Builder for the SA-1110 Development Platform:

1. 1) Unpack the WindowsCE* Driver kit using the directions in [Section 4.0](#) and use

```
[drive]:\Wince212 as winceroot.
```

2. Set environment variables for the configuration you need. Be sure to set the following environment variable in the SA11X0BD.BAT file:

```
IMGPB212=1
```

3. Copy the file \wince212\platform\SA11X0BD\SA11X0BD.bsp to

```
C:\Program Files\Windows CE Platform Builder\2.12\CEPB212\bin\IDE
```

4. Start Platform Builder 2.12.
5. Before creating a new project, select "Manage Platform Builder Components" using the "File" menu.
6. In the "Manage PB Components" dialog box, select "Import New".

7. In the "Import Platform Builder Component" dialog box, navigate to the `\wince212\platform\SA11X0BD` directory and select `SA11X0BD.ccc`. Click "Done".
8. Use the Wizard to create a new platform using "File"->"New" and selecting the "Platform" tab.
9. Type SA11X0BD in the platform name entry and click "Ok".
10. Select SA11X0BD from the list of Board Support packages and click "next". (If SA11X0BD does not appear as a BSP go back to step 3 and confirm that the bsp file is in the proper place).
11. Select MAXALL as the project and finish the wizard.
12. You can now build your image. Before building the image, be sure the environment variables set in the IDE do not conflict with those set in the SA-1110 batch file, `SA11X0BD.bat`. The IDE settings will take precedence.

6.0 Modifying the Build

The following section describes environment variables that can be modified to adapt the SA-1110 Development Board to your design needs.

6.1 Environment Variables

The specific environment variables used to control the SA-1110 Development Platform build are contained in the `SA11X0BD.bat` file in the target platform area in the file:

```
WINCEROOT\Platform\SA11X0BD\SA11X0BD.bat
```

where *WINCEROOT* is the root directory of the Windows CE installation.

For more information about environment variables, see the Platform Builder documentation. The following is a description of the variables in that file and their use:

- `PLAT_SA1110` – Defaults to 1. When set to 1, configures the build for the SA-1110 microprocessor.
- `PLAT_SA1111` – Defaults to 0. When set to 1, configures the build for the SA-1111 Companion Chip.
- `PLAT_SA1110DB` – Defaults to 1. When set to 1, configures the build for the SA-1110 Development Board.
- `PLAT_SA1111DB` – Defaults to 0. When set to 1, configures the build for the SA-1111 Development Module.
- `PLAT_SA1110EB` – Defaults to 0. When set to 1, configures the build for the SA-1110 Evaluation Board.
- `PLAT_SA1100MM` – Defaults to 0. When set to 1, configures the build for the SA-1100 Multimedia Module.
- `PLAT_SA1101CB` – Defaults to 0. When set to 1, configures the build for the SA-1101 Companion Board.
- `PLAT_MEMSIZE` – Defaults to 32. Configures the build for 32Mb of system DRAM.
- `PLAT_NOSERIAL` – Defaults to 0. When set to 1, configures the build to exclude the serial driver.

- PLAT_NOSERMODEM – Defaults to 0. When set to 1, configures the build to exclude the serial modem driver.
- PLAT_NOAUDIO – Defaults to 0. When set to 1, configures the build to exclude the audio driver.
- PLAT_UCB1300_AUDIO – Defaults to 0. When set to 1, configures the build to use the UCB-1300 for the audio driver.
- PLAT_UDA1341_AUDIO – Defaults to 1. When set to 1, configures the build to use the UDA-1341 for the audio driver.
- PLAT_NOUSB – Defaults to 1. When set to 1, configures the build to exclude the USB host controller.
- PLAT_NOKEYBD – Defaults to 1. When set to 1, configures the build to exclude keyboard driver. When not set will include the SA-11X1 PS2 Keyboard driver
- PLAT_NOBATT – Defaults to 0. When set to 1, configures the build to exclude the battery driver.
- PLAT_NONLED – Defaults to 0. When set to 1, configures the build to exclude the GWES LED driver.
- PLAT_NODISPLAY – Defaults to 0. When set to 1, configures the build to exclude the display driver.
- PLAT_NOTOUCH – Defaults to 0. When set to 1, configures the build to exclude the touch screen driver.
- PLAT_NOPCMCIA – Defaults to 0. When set to 1, configures the build to exclude the PCMCIA driver.
- PLAT_NOGWES – Defaults to 0. When set to 1, configures the build to exclude the GWES components.
- PLAT_NOMOUSE – Defaults to 1. When set to 1, configures the build to exclude the mouse driver. When not set will include the SA-11X1 PS2 mouse driver
- PLAT_DBGSER_UART1 – Defaults to 0. When set to 1, configures the build to use UART 1 as the debug serial port. When not set, the build will use UART 3 as the debug serial port.
- PLAT_CLICKS – Defaults to 0. When set to 1, configures the build to include waveclick.dll.
- PLAT_DISPLAY_SA_LCD – Defaults to 1.
- PPSH_PROTOCOL_NOTIMEOUT – Defaults to 1.
- IMGPB212 – Defaults to 1. When set to 1, configures the build to support builds under Platform Builder 2.12.
- IMGVOICE – Defaults to 1. For more information, see the Platform Builder Documentation.
- IMGNOFONTS – Defaults to 0. For more information, see the Platform Builder Documentation.
- IMGNOETHER – Defaults to 0. For more information, see the Platform Builder Documentation.
- IMGNOSIR – Defaults to 1.
- IMGCOMMDEMOS – Defaults to 0. For more information, see the Platform Builder Documentation.

- IMGACMSAMPLES – Defaults to 0. For more information, see the Platform Builder Documentation.
- IMGEBOOT – Defaults to 1. For more information, see the Platform Builder Documentation.
- IMGUSB – Defaults to 0. For more information, see the Platform Builder Documentation.
- IMGNODEBUGGER – Defaults to 1. For more information, see the Platform Builder Documentation.
- IMGFLASH – Defaults to 0. For more information, see the Platform Builder Documentation.
- IMGMOUSE – Defaults to 0. Required to be set for Jupiter builds. For more information, see the Jupiter Application Kit Documentation.
- IMGNOWELCOME – Defaults to 1. For more information, see the Jupiter Application Kit Documentation.
- IMGUSEPROXY – Defaults to 1. For more information, see the Jupiter Application Kit Documentation.

6.2 Building for the SA-1110 Development Platform

When building for the SA-1110 Development Platform without the SA-1111 Development Module, be sure to have the following environment variables in SA11X0BD.bat file set accordingly:

```
PLAT_SA1110=1
PLAT_SA1111=
PLAT_SA1110DB=1
PLAT_SA1111DB=
PLAT_NOKEYBD=1
PLAT_NOMOUSE=1
```

6.3 Building for the SA-1110 and SA-1111 Development Boards

When building for the SA-1110 Development Platform along with the SA-1111 Development Module, be sure to have the following environment variables in SA11X0BD.bat file set accordingly:

```
PLAT_SA1110=1
PLAT_SA1111=1
PLAT_SA1110DB=1
PLAT_SA1111DB=1
```

Additionally, if support for the keyboard and mouse on the SA-1111 Development Module is desired, then clear the following environment variables:

```
PLAT_NOKEYBD=
PLAT_NOMOUSE=
```

Note: If the image is built to expect a keyboard and mouse using the settings above, be sure that the keyboard and mouse are connected to the SA-1111 Development Module before loading the image. Failure to do so will result in the image booting with no mechanism for exiting the touch calibration screen.

6.4 Running Remote Programs with SA-1110 and SA-1111 Development Boards Windows* CE MAXALL Builds

Intel recommends that before you begin you should verify that:

- You can perform a Windows CE MAXALL build as described in the release notes.
- Your SA-1110 and SA-1111 development boards are set up with the PS/2 mouse and keyboard connected as described in the *Intel® StrongARM* SA-1110 Microprocessor Development Board User's Guide*, order number 278278 and the *Intel® StrongARM* SA-1111 Microprocessor Development Module User's Guide*, order number 278281.

Use the following procedure to run remote programs:

1. Edit the file WINCE212\Platform\SA11X0BD\SA11X0BD.bat and make sure that the IMGEBOOT environment variable is cleared. Specifically:

```
IMGEBOOT=
```
2. Build your Windows CE MAXALL image according to the documentation.
3. Apply power to the SA-1110 Development Board and SA-1111 Development Module.
4. Wait for the Windows CE touch panel calibration screen to appear.
5. Calibrate the screen or press the Esc key on your PS/2 keyboard to clear the touch panel calibration screen. A flash card warning dialog box appears but will not have focus.
6. Insert the Socket LAN CF Ethernet card into either slot using the supplied CF to PCMCIA adapter.
7. Press the Ctrl-Alt-Del keys simultaneously to get the Task Manager dialog box to appear.
8. Use the Tab key to select the Run button and then use the Tab key to select the Browse button on the Run dialog box.
9. Select the control program (Control Panel) from the available files.
10. Use the Tab key to select the Communication icon. Give your system a name that can be used in your LAN.
11. Select the Networking icon and then the Identification tab. Enter the user name, password and domain for access to the remote system.
12. Select the Adapter tab and use the Tab key to select the Properties button for the NE2000 compatible card.
13. Press the Enter key to select DHCP or use the Tab key to enter a static address. If you use a static address be sure to fill in the DNS entries on the DNS tab. Press Enter to accept the Adapter properties. A confirmation dialog box will tell you that the next time the Ethernet card is inserted the properties will take effect.
14. Verify that your CF LAN card is connected via twisted pair cable to a hub on your network and then insert the card into one of the PCMCIA slots on the SA-1111 Companion Board.
15. Use the Alt key to close the Control Panel.

16. Press the Ctrl-Alt-Del keys simultaneously to get the Task Manager dialog box.
17. Use the Tab key to select the Run button.
18. In the "Open" field of the Run dialog box, type the network path to the Windows CE executable file you want to run.

7.0 Loading the Windows* CE Image via the Ethernet Boot Loader

This section describes how to load the Windows CE image on to the SA-1110 Development Board using the Ethernet Boot (Eboot) loader. The Ethernet communications necessary for downloading the Windows CE image can be achieved by using a Socket Communications Compact Flash LP-E CF+ card (compact flash Ethernet card) inserted into the Compact Flash slot on the SA-1110 Development Board.

Alternatively, to use an SA-1111 Development Module, a Compact Flash Ethernet card with a PCMCIA adapter inserted into the PCMCIA slot of the SA-1111 Development Module is used.

Note: For more information on customizing the Windows CE Operating System, refer to the *Adapting Windows CE to Your Platform* chapter in the Platform Builder Books On-line documentation.

The following directions are written for a target system consisting of the SA-1110 Development Board and SA-1111 Development Module using the Compact Flash to PCMCIA adapter for Ethernet communication.

Before preceding with the Ethernet boot load, verify that:

- The SA-1110 Development Board and the SA-1111 Development Module are ready to be powered and are connected to your Windows NT development workstation as outlined in the *Intel® StrongARM* SA-1110 Development Board User's Guide*.
- If Ethernet debugging is enabled, be sure to build the image with WINCEDEBUG=debug, IMGEBOOT=1, and WINCEREL=1 environment variables in your SA11X0BD.bat file in *WINCEROOT\platform\SA11X0BD*, where *WINCEROOT* is the root directory of the Windows CE installation.
- The Eboot image is in flash RAM of the SA-1110 Development Board. For information on changing the Intel® Strataflash™ memory components, see the *Intel® StrongARM* SA-1110 Development Board User's Guide*.

Use the following procedure to load the Windows CE image via the Ethernet boot load software:

1. Plug the Compact Flash-PCMCIA Ethernet card into the outermost PCMCIA slot of the SA-1111 Development Module.
2. Plug a UTP Ethernet cable into the dongle of the Compact Flash Ethernet card and connect it to a hub with your workstation or corporate net with the same sub-net as your development workstation.
3. Plug the RS232 adapter into the SA-1110 Development base station and to the Com port on your workstation.

Note: The amount of network traffic that is often on a corporate network can impede the downloading effort. Alternatively, a UTP “null” Ethernet cable can be used to connect the host system to the SA-1110 Development Board.

4. Create a shortcut for Eshell. For example:
`WINCEROOT\public\common\oak\bin\i386\eshell.exe`
5. Activate a Hyperterm window and set it to the COM port that is connected to the SA-1110 Development Board and SA-1111 Development Module (choose Programs from the Start menu, and then choose Accessories, Hyperterminal, and Hyperterminal again).
6. Enter a name and choose an icon for the connection when prompted.
7. Select the COM port in use when prompted.
8. Set the COM properties as follows:

```
Bits per second-38400
Data bits-8
Parity-None
Stop bits-1
Flow control-None
```

9. Double click on the Eshell.exe icon and apply power to the SA-1110 Development Board and SA-1111 Development Module.
10. Observe the messages in the Hyperterm window as the Eboot device begins to boot.

Note: If you are not connecting to a corporate LAN with DHCP, provide the SA-1110 Development Board with an IP address and sub-net mask on the same subnet as your development workstation.

11. Wait until the message “Sent BOOTME to XXX.XXX.XXX.XXX” appears in the Hyperterm window (where XXXX represents your sub-net).
12. Select the Eshell window, click on the Tools menu and select the option:
 Manage Device List...
13. Double click on the name in the “Non Controlled Devices” list (which places the name in the Controlled Devices list) and write down the name which will be similar to:
`SA11X0BD31092` (which is the *targetname*).
 Note: If you change the socket card, the name will also change.
14. Open a DOS window and type the following command and then press Enter:

```
ceterm -e targetname
```

A Monitor Window for Ethernet debug messages will appear.

15. Click back on the DOS window and type:

```
cesh -e targetname
```

and press Enter. The window will wait for the target system to finish booting.

16. Create a shortcut for windbg (`WINCEROOT\public\common\oak\bin\i386\windbg.exe`).
17. Right click on the shortcut and select Properties.

18. Select the shortcut tab and edit the target field using:

```
WINCEROOT\SDK\BIN\I386\WINDBG.EXE -g -y %_FLATRELEASEDIR% -k arm udp targetname
```

19. Save the changes. If the environment variable `IMGNODEBUGGER` is not set to a 1, double click on the Windbg icon.

20. Click on the Eshell window and select the option Download Only on Command from the Options menu.

21. Click on the File menu and select the option Select Image File.

22. Navigate to the location of your image, which is typically `WINCEROOT\release`, and select the image, which is typically `nk.bin`.

23. Click on the File menu and select the option Download Image.

24. Observe the lower left hand corner of the Eshell window to monitor download progress.

Once your image has booted, Ethernet debug messages will appear in the Monitor window, CESH commands may be entered in the DOS window, and kernel debugging may be conducted from the windbg window.

7.1 Load and Launching Files

CESH commands can be used to load and launch applications, enable Ethernet debugging and the Windows* CE debugging tool. The following CESH command launches an application:

```
s processname
```

where `processname` is the name of the application to be launched. For example, the following command starts the Control Panel:

```
s control
```

7.2 Adding an Application to the your Image

This is one method to add an application to the SA-1110 Development Platform's image.

1. Start PB 2.12 IDE and open a Workspace.
2. To build the application under the current workspace, create a project by selecting File, New, Project. If you have already created an application in a project outside of the current workspace, proceed to step 5.
3. Select appropriate project: application, dll, etc.
4. Write and build an application using the usual methods.
5. Insert your project into the SA11x0BD platform by selecting Project, Insert Project into Platform...
6. Select the project you wish to insert into the build. If you created the project within the current workspace, a message box will appear indicating that a project with the same name already exists in that workspace; click OK.
7. In the Platform View window select: Parameter View tab, and expand the Project Specific Files folder.

8. Double click on the PROJECT.BIB file so that it may be edited.
9. In the PROJECT.BIB file under MODULES you will need to add all the modules of your application along with the Path, Memory Type. For example:

```

MODULES
; Name          Path                                     Memory Type
; -----
myproject.exe  $(_FLATRELEASEDIR)\myproject.exe                 NK S

```

Note: Environment variables can be used to conditionally add your application to the workspace build. For example:

```

MODULES
; Name          Path                                     Memory Type
; -----
IF ADDMYAPP
myproject.exe  $(_FLATRELEASEDIR)\myproject.exe                 NK S
ENDIF

```

1. Build the platform as usual.
2. At this point you are now ready to debug your application. You may now load your application source code (using the File, Open command) and enter break points, look at local variables, and so on. See the Microsoft Windows CE Platform Builder documentation for details on debugging applications.

7.3 Debugging an Application using a Kernel Debugger

This section describes how to debug an application using the same debugging tools and techniques as a device driver. The note assumes that the reader is working with a standard StrongARM* SA-1110 Development Board.

Download for both of the CF interfaces is supported only for the Socket LP-E CF+ Compact Flash card. Download for the PCMCIA interface is supported only on the Accton EN2216-2, Socket LP-E, or the Socket LP-E CF+ Compact Flash card in the provided PCMCIA adapter.

7.3.1 Debugging an Application Using Ethernet Messaging

Ethernet debugging for the standalone SA-1110 Development Board is only supported on the Socket LP-E CF+ Compact Flash card. Ethernet debugging for the SA-1110 Development Board with the SA-1111 Development Module attached is supported with any of the above mentioned cards but only when used in the outermost slot.

1. Build the application using Windows CE platform build in debug mode so that a PDB (program data base) file is created. (See [Section 7.2](#) for details on adding an application to your image.)
2. Edit your project.bib file to include your application, any required dlls, and all PDB files associated with your application.

3. Before building your image, be sure to set the following environment variables:

```
IMGNODEBUGGER=  
IMGEBOOT=1  
WINCEDEBUG=debug
```

Note: The IMGNODEBUGGER environment variable is set to nothing. That is, no text follows the '=' (equal) sign. This ensures that support for the kernel debugger will be included in the build.

4. Download the image and start the kernel debugger following the directions in [Section 7.3.2](#).

Note: At this point you are now ready to debug your application. You may now load your application source code (using the File, Open command) and enter break points, look at local variables, etc. See the Microsoft Windows CE Platform Builder for details on debugging applications.

7.3.2 Debugging a Driver with Ethernet Messaging

1. Build the SA11X0BD project to include the kernel debugger. Be sure to set the following environment variables:

```
IMGNODEBUGGER=  
IMGEBOOT=1  
WINCEDEBUG=debug
```

2. Open the nk.exe file in the PB 2.12 IDE
(Navigate to the directory of the file and enter nk.exe and then click the Open button.)
3. Choose Debugger Remote Connection from the Build menu
4. Select UDP, and provide the SA-1110 Development Board's Ethernet identification. This is the same number that is used in the E-shell window for identifying your SA-1110 device.
5. Download nk.bin as you normally do by using HyperTerm, E-shell, CTerm, and CESH.
6. In the CTerm window, the booting process will get to KDInit and wait. At this point, start the host debugger component (nk.exe) by pressing the GO icon for the debugger in the nk.exe window. Or go to the Build menu->Start Debug->Go.
7. After the Windows CE image is done booting, breakpoints can be set. To set breakpoints, stop the debugger. Highlight the source code line where the breakpoint should be placed and press the "Set Breakpoint" icon or type the "Alt-F9" keys.

intel®

