



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

User's Guide

October 2000

Version 2.0



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	Installing Microsoft Windows* CE Platform Builder 3.0	7
3.0	Validating the Microsoft Windows* CE Platform Builder Installation	8
4.0	Installing the SA-111x Board Support Package	8
5.0	Modifying the Build	10
5.1	Environment Variables	10
5.2	Building for the SA-1110 Development Platform	12
5.3	Building for the SA-1110 and SA-1111 Development Boards	12
6.0	Tips for Loading the Windows* CE Image via the Ethernet Boot Loader	13
6.1	Ethernet Connection Issues	14
7.0	Adding an Application to the your Image	14
7.1	Debugging an Application using a Kernel Debugger	15
7.1.1	Debugging an Application Using Ethernet Messaging	16
7.1.2	Debugging a Driver with Ethernet Messaging	16

Figures

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

1.0 Overview

This document describes how to use the Microsoft Windows* CE Platform Builder 3.0 with the Intel® StrongARM* SA-1110 Development Platform (SA-1110 Development Platform). This document also describes installation and build procedures, environment variables, and tips for using the Ethernet boot loader.

Note: The term *SA-1110 Development Platform* refers to using the Intel® StrongARM* SA-1110 Development Board in a stand-alone environment, and for using the Intel® StrongARM* SA-1110 Development Board with the Intel® StrongARM* SA-1111 Development Module.

1.1 Description

The Intel® SA-111x Development Board Support Package (SA-111x Board Support Package), Version 2.0, for Microsoft Windows* CE Platform Builder 3.0 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 3.0. Although these drivers have been tested, they are only *examples* and cannot be used for production systems. For warranty and support information, see the license agreement when you install this SA-111x Board Support Package.

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 Platform.

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-111x Board Support Package¹ 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, the following requirements are necessary to build the Windows CE operating system for the SA-1110 target platform:

- An SA-111x Board Support Package
- 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 Development 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:

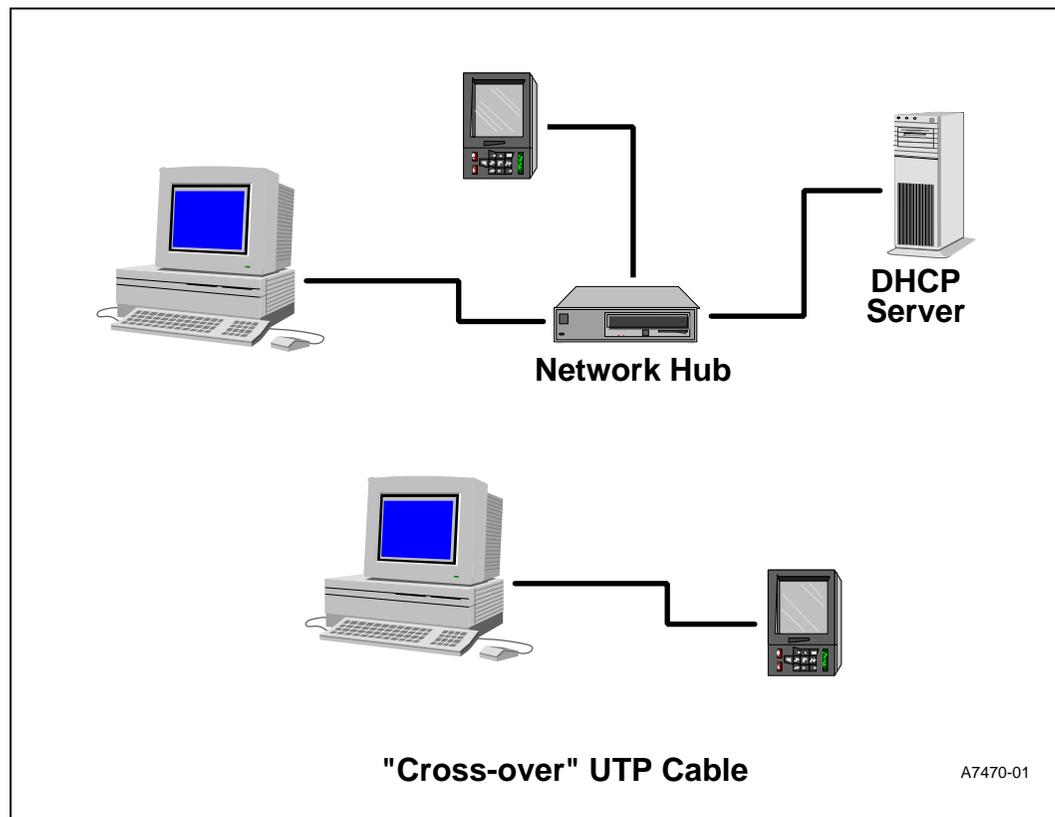
- A network HUB with a DHCP server
- A 10BASE-T unshielded twisted pair (UTP) “cross-over” Ethernet cable to go between the host and SA-1110 Development Board (provided with the SA-1110 Development Board kit)

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 Intel[®] SA-111x Development Board Support Package, Version 2.0, for Microsoft Windows* CE Platform Builder 3.0, see Intel's website for developer's at: <http://developer.intel.com/design/strong/swsup/>

Figure 1. Network Configuration Alternatives



2.0 Installing Microsoft Windows* CE Platform Builder 3.0

Use the following procedure to install Microsoft Windows* CE Platform Builder 3.0.

1. Follow the instructions in the Microsoft Windows CE Platform Builder documentation to install the Microsoft Windows CE Platform Builder. To save disk space, only the StrongARM platform (denoted as SA-1100 in Platform Builder) is required. Installing the MIPS, HITACHI, ARM* 7xx and other platforms are not necessary.
2. Choose your source and destination locations for installing Microsoft Windows CE Platform Builder. After the destination location has been chosen, the CESH Update Screen appears.
3. Determine if this is a new install or update of Microsoft Windows CE Platform Builder:
 - For new installs, choose: Use Ethernet.
 - For updates, choose: Do not change Current CESH settings.
4. Follow the instructions on the screen and verify that the option Register Environment Variables is checked on the last menu.

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:\wince300`.)

If the set command returns the error message:

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

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

4.0 Installing the SA-111x Board Support Package

The SA-111x Board Support Package consists of the following items:

- OEM adaptation 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 SA-111x Board Support Package.

Use the following procedure to install the SA-111x Board Support Package on your host system:

Note: For more information on these directions, see the Microsoft Windows CE Platform Builder 3.0 Getting Started Guide, document number X05-69830. If you need to repeat this process, be sure to delete the SA11X0BD board support package file, `sa11x0bd.bsp`, found in `\Windows CE Platform Builder\3.00\Cepb\Bin\Ide`. Failure to do will cause the creation of the SA11X0BD platform to fail.

1. Double-click on the self-contained executable file containing the SA-111x Board Support Package 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 driver sources for the Version 2.0 Microsoft Windows* CE PBK.
3. 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:\wince300). This sets up the platform builder with a new target platform, namely the SA-1110 Development Board.

4. On the development workstation, choose Microsoft Windows CE Platform Builder in the program group Microsoft Windows CE Platform Builder 3.00 to open the Platform Builder application.
5. Choose New from the File menu. The New dialog box appears.
6. Select the Platform tab. Enter SA11X0BD for platform name.
7. Enter a new path for the platform workspace in the Location box (optional).
8. Select WCE SA-1100 for supported microprocessors in the Processor box.

Note: The selection SA-1100 supports SA-1100 and SA-1110 microprocessors.

9. Choose OK. The WCE Platform -Step 1 of 2 dialog box appears.
10. Select MyBSP from the SA-111x Board Support Package list. Enter SA11X0BD for the path for the BSP at the subdirectory prompt.
11. Choose Next. The WCE Platform -Step 2 of 2 dialog box appears.
12. Select Maximum OS and choose Finish. The New Platform Information dialog box appears that displays information about your configuration.
13. Choose OK to close the New Platform Information dialog box and to finish creating the platform.
14. Import the SA11X0BD catalog file, sa11x0bd.cec, to populate the board support package with the SA1110 Development Board driver components. Choose Manage Platform Builder Components from the File menu.
15. Choose the Import New... button. Select the sa11x0bd.cec file in the WinCe300\Platform\sa11x0bd directory. Press the Refresh button to update the Catalog view with the sa11x0bd driver components.
16. Go to the Catalog View on the right side of the IDE. If the Catalog is not already displayed, choose Catalog on the View menu to display the Catalog. Browse to the SA11X0BD OAL.
17. Add the SA11X0BD OAL component to the Sa11X0BD Board Support Package by Right-clicking the component and select Add to Platform on the shortcut menu.
18. Browse to the SA11X0BD drivers to view the SA11X0BD driver components.
19. Add the SA11X0BD driver components to the Sa11X0BD Board Support Package by Right-clicking the component and select Add to Platform on the shortcut menu. Repeat until all SA11X0BD components are added to the platform.

The new components appear in your ComponentView window.

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

Note: Intel recommends using the 3.0 Integrated Development Environment (IDE) for building your image. For instructions on the 3.0 Integrated Development Environment (IDE), refer to the

Platform Builder 3.0 on-line documentation for Building a Basic Platform - Building an OS Image - IDE Procedures.

5.0 Modifying the Build

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

Note: For more information on customizing the Windows CE operating system, refer to the section about Customizing a Platform in the Platform Builder Library documentation.

5.1 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_NO SERIAL – Defaults to 0. When set to 1, configures the build to exclude the serial driver.
- PLAT_NO SERMODEM – Defaults to 0. When set to 1, configures the build to exclude the serial modem driver.
- PLAT_NO AUDIO – 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 audio driver on the SA-1110 Development Board.

- PLAT_UDA1341_AUDIO – Defaults to 1 when SA1110DB_CFG or SA111XDB_CFG is set. When set to 1, configures the build to use the UDA-1341 audio driver on the SA-1110 Development Board.
- PLAT_AC97_AUDIO – Defaults to 0. When set to 1, configures the build to use the AC97 audio driver on the SA-1111 Development Module.
- PLAT_NOUSB – Defaults to 1. When set to 1, configures the build to exclude the USB host controller.
- PLAT_NOUSBCLIENT – Defaults to 0. When set to 1, configures the build to exclude the USB host function drivers.
- PLAT_NOKEYBD – Defaults to 1. When set to 1, configures the build to exclude keyboard driver. When not set will include the PS/2 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_NOIRDA – Defaults to 0. When set to 1, configures the build to exclude the IrDA 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 PS/2 mouse driver
- PLAT_DBGSER_UART1 – Defaults to 1. 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.
- IMGPB212 – Defaults to 1. When set to 1, configures the build to support builds under Platform Builder 2.12 or later.
- IMGVOICE – Defaults to 1.
- IMGNOFONTS – Defaults to 0.
- IMGNOETHER – Defaults to 0.
- IMGCOMMDEMOS – Defaults to 0.
- IMGACMSAMPLES – Defaults to 0.
- IMGEBOOT – Defaults to 0 unless doing a debug build. When set to 1, includes Windows CE Ethernet debugging services.
- IMGUSB – Defaults to 0. When set to 1, the USB host driver is included in the image.

- IMGNODEBUGGER – Defaults to 1 for builds without kernel debugging. For more information, see the Platform Builder Documentation.
- IMGFLASH – Defaults to 0. When set to 1, builds a flashable image. For more information, see the Platform Builder Documentation.

5.2 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
```

5.3 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.

6.0 Tips for Loading the Windows* CE Image via the Ethernet Boot Loader

This section documents issues regarding loading of the Windows CE image and details information specific to the SA-1110 Development Platform 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 downloading images using the Windows Platform Builder IDE or the Eshell.exe application, see *Download the OS Image* in the Platform Builder 3.00 Library documentation.

The following information is provided 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 proceeding with the Ethernet boot load, verify that:

- The SA-1110 Development Board and the SA-1111 Development Module are ready to be powered up 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 desired, be sure to build the image with `IMGEBOOT=1` environment variable to include the Windows CE Ethernet debugging services.
- The proper Eboot image is in flash RAM of the SA-1110 Development Board. For this release, Eboot version 4.4 is required.

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 creating a private network.

4. Activate a Hyperterm window and set it to the COM port that is connected to the SA-1110 Development Platform (choose Programs from the Start menu, and then choose Accessories, Hyperterminal, and Hyperterminal again).
5. Enter a name and choose an icon for the connection when prompted.
6. Select the COM port in use when prompted.

7. Set the COM properties as follows:

Bits per second-38400

Data bits-8

Parity-None

Stop bits-1

Flow control-None

8. Use either the Platform Builder IDE or the Eshell.exe applications to download the image. Refer to *Download the OS Image* or *Configuring Eshell.exe for an Ethernet Connection* for information on downloading images uses these tools.

6.1 Ethernet Connection Issues

Please consider the following issues when using Ethernet connections between an SA-1110 Development Platform and an NT Development System:

- If you are *not* connecting to a corporate LAN with a DHCP server, you will need to provide the SA-1110 Development Platform with an IP address and sub-net mask on the same subnet as your development workstation.
- If you are using a UTP “null” Ethernet cable to create a private network, you will need to provide the SA-1110 Development Board with an IP address and sub-net mask on the same subnet as your development workstation.
- Intel recommends that you configure the SA-1110 Development Platform using a static IP address in the form of: 192.168.xxx.xxx. This will prevent packets from leaving the local network. For more information see: <http://www.landfield.com/rfcs/rfc1918.html>
- To set the IP address and sub-net mask, you will need to enter the values in the Hyperterm window when Eboot issues the prompts:

Wait for DHCP, enter new IP address, or CR to use existing IP: (Enter your ip address here)
Enter new subnet mask or CR to use existing mask: (Enter your sub-net mask here)

- The SA-1110 Development Platform will broadcast their name as SA11X0BDXXXXX where xxxxx is a 5 digit id determined by the specific PCMCIA or CF NIC card. The Microsoft documentation illustrates these addresses in the form of *CEPCLSXXXX*.
- If you change the PCMCIA or CF NIC card, the xxxxx component of the device name will also change.

7.0 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 3.00 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. and provide the project name and location.

4. Select an appropriate type to create.
5. Write and build an application using the usual methods.
6. Insert your project into the SA11x0BD platform by selecting Project, Insert, Existing Project...
7. 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.
8. In the Platform View window select: Parameter View tab, and expand the Project Specific Files folder.
9. Double click on the PROJECT.BIB file so that it may be edited.
10. 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
    
```

11. Build the platform as usual.
12. 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.

7.1 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.

7.1.1 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 7.0](#) 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. The kernel debugger will automatically be started.

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 documentation for details on debugging applications.

7.1.2 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
```

This can be done in the following manner in the Platform Builder 3.0 IDE:

- a. Configure the build for debug by choosing the debug configuration from the Build menu.
 - b. Choose Settings from the Platform menu and check the Enable Kernel Debugging box.
 - c. Click on the Environment tab and add the IMGEBOOT variable, setting it to 1.
2. Download nk.bin as you normally do by using HyperTerm and E-shell or the IDE. To expedite the loading process, Intel recommends disconnecting the Hyperterminal window once the image begins to load.
 3. 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 "Insert Breakpoint" icon or type the "F9" key.
The debug output window shows the progress of downloading the Windows CE image. When done downloading, the kernel debugger is automatically started.



Note: See the Microsoft Platform Builder documentation for details on using the debugger.

intel®

