



Zoom™ Development Kit Windows® CE Evaluation Guide

Software Documentation

Logic Product Development
Published: January 2005
Last Revised: June 2007

Abstract

This document describes how to load a sample Windows CE binary image (NK.bin) from a CompactFlash card and perform simple debugging operations through ActiveSync in Microsoft Embedded Visual C++ on Logic's Zoom™ Development Kits.

This file contains source code, ideas, techniques, and information (the Information) which are Proprietary and Confidential Information of Logic Product Development, Inc. This information may not be used by or disclosed to any third party except under written license, and shall be subject to the limitations prescribed under license.

No warranties of any nature are extended by this document. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipments. The only warranties made by Logic Product Development, if any, with respect to the products described in this document are set forth in such license or agreement. Logic Product Development cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

Logic Product Development may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. Except as expressly provided in any written agreement from Logic Product Development, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

© Copyright 2005–2007, Logic Product Development, Inc. All Rights Reserved.

REVISION HISTORY

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	Aaron Stewart	Release	HAR	01/03/05
B	Aaron Stewart	Document Edit: changed serial port references to terminal emulator	JAW	01/06/05
C	James Wicks	Added captions to figures/ doc edit	JAW	01/28/05
D	Aaron Stewart	Added Sample NK.bin information, installation instructions for ActiveSync, and Logic's USBF Workstation Driver	JAW	07/21/05
E	Aaron Stewart	Updated Section 2.2.1 instructions and images for ActiveSync; Removed WinCE USBF Driver as a Required Component because it is automatically included with the latest ActiveSync; Generalized document to include usage for other Zoom Development Kits and SOMs; General grammatical and formatting edits	JCA	11/20/06
F	Jed Anderson	Section 2.2: Added note about touch calibration parameter in the boot script	MAA	06/21/07

1 Introduction

This document describes how to load a Windows CE Sample Development Image on a Zoom Development Kit, and then use the Sample Development Image and eVC++ to debug and auto-launch an application. By following the steps in this document, you will be able to evaluate the capabilities of Windows CE 5.0 with Logic's Zoom Development Kit.

1.1 Required Components

1. Microsoft eMbedded Visual C++—this is a free download from Microsoft, available online at: <http://www.microsoft.com/downloads/>.
2. Tera Term terminal emulator (or equivalent).
3. WinCE 5.0 Software Development Kit (SDK) for the applicable System on Module (SOM)—this can be found under the 'Software Development Tools' category of the 'All Downloads' page for your SOM.
4. WinCE 5.0 Sample Development Image for the applicable SOM—this can be found under the 'Windows CE Sample Images' category of the 'All Downloads' page for your SOM.
5. A CompactFlash card with a capacity of at least 32 MB—this card must be large enough to store the sample Windows CE OS image binary file.
6. The latest version of Microsoft's ActiveSync synchronization software—this software is free from Microsoft on the Windows Mobile downloads page at: <http://www.microsoft.com/windowsmobile/downloads>.

1.2 Logic's Sample NK.bins

Logic's sample NK.bins allow users to evaluate Logic's LoCE Binary BSP on the Zoom Development Kit. Logic's sample NK.bins are also useful for application development because each image includes helpful libraries and a variety of drivers.

1.2.1 Custom NK.bin Options

Logic's sample NK.bin images are not meant for production and should not be regarded as production-ready.

Logic offers these solutions to customers requiring a custom Windows CE image for a device:

- Contract Logic to provide a production-ready NK.bin that is custom suited to your project's needs.
- Use Logic's LoCE Binary BSP with Microsoft Platform Builder to create a custom image for your development.

1.2.2 Sample Development Image for Windows CE 5.0

The purpose of Logic's 'Sample Development Image' is to aid application development with the Windows CE Software Development Kit (available for download on Logic's site).

Sample Development Image .zip files, located on the downloads page under the 'Windows CE Sample Images' heading, are comprised of the following:

1. A binary image,
2. A clean Windows CE 5.0 workspace that can be used to build the image,
3. Example boot scripts for Logic's Display Kits,
4. Release notes which detail revision information and component descriptions.

1.2.3 Sample Show Image for Windows CE 5.0

The purpose of the 'Sample Show Image' is to demonstrate the capabilities of Logic's LoCE BSP driver and the OAL, and show the ability of the SOM to perform various operations in Windows CE.

Sample Show Image .zip files, located on the downloads page under the 'Windows CE Sample Images' heading, are comprised of the following:

1. A binary image,
2. Example boot scripts for Logic Display Kits.

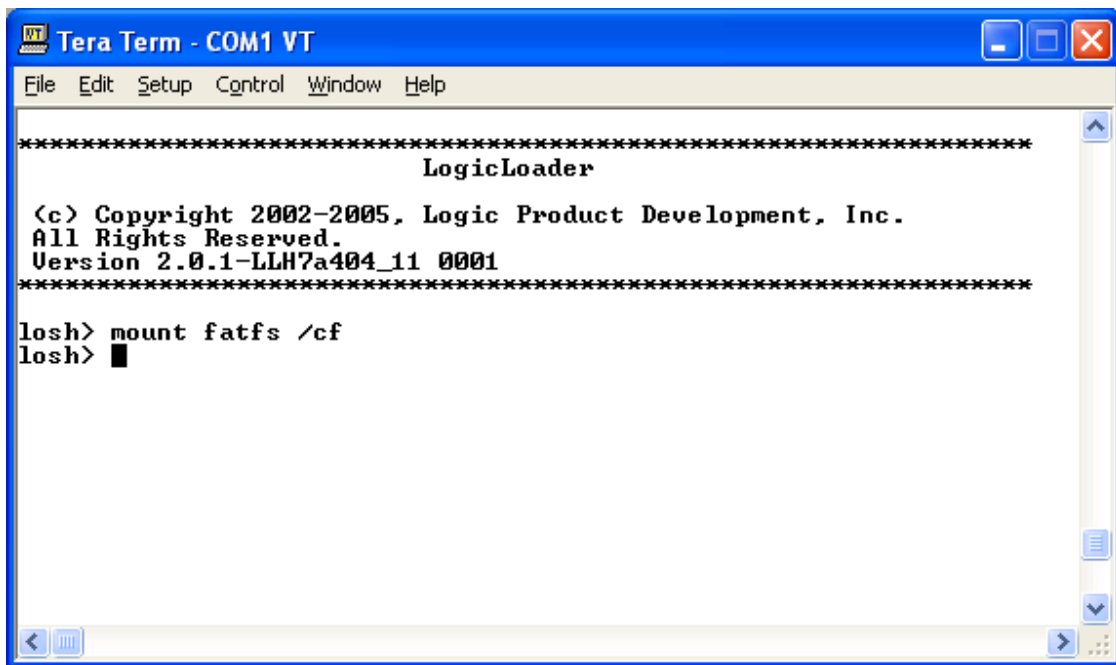
2 Loading the Windows CE 5.0 Sample Development Image

2.1 Downloading the OS Image Binary

This section describes how to download the OS image from a CompactFlash card in LogicLoader™ (LoLo).

The first step is to load the Windows CE 5.0 Sample Dev Image from the CompactFlash card in LogicLoader and successfully boot the Image.

1. Download the Windows CE 5.0 Sample Dev Image binary to a development workstation and store the NK.bin file on a CompactFlash card.
2. Insert the CompactFlash card into the Zoom Development Kit's CompactFlash slot.
3. Start a terminal emulator program on the development workstation and connect the serial cable, provided with the Development Kit, from the workstation to the Development Kit.
4. Power up the Development Kit and verify the LogicLoader prompt appears in the terminal emulator window.
5. Use the 'mount' command to mount the CompactFlash card in LogicLoader, the specific usage is presented in the image below: 'mount fatfs /cf'.

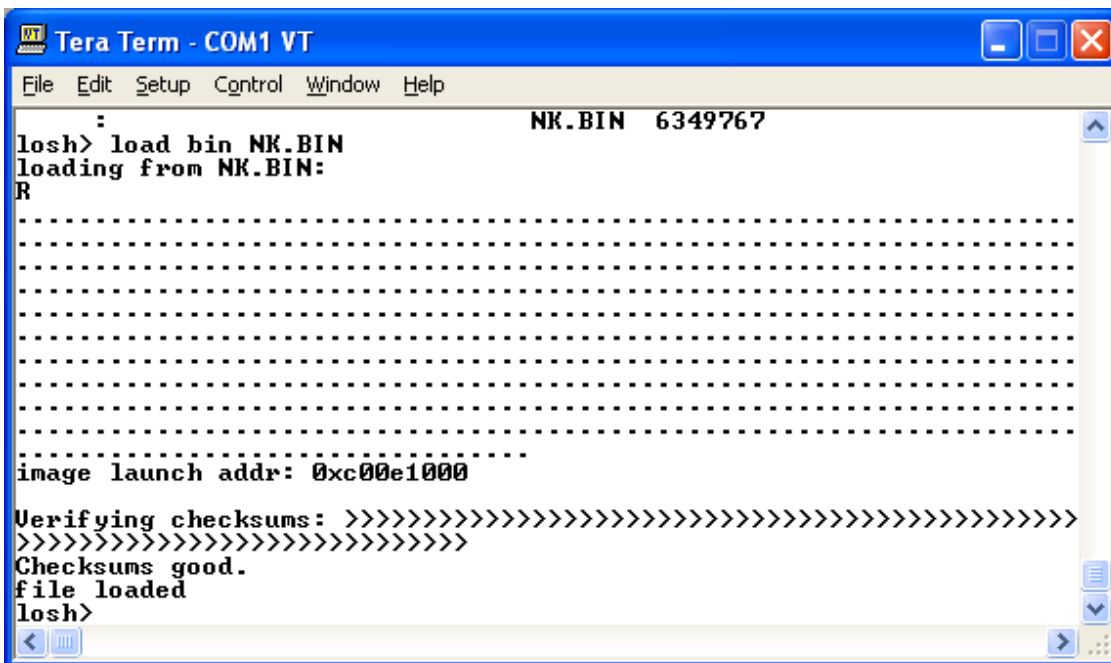


```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2005, Logic Product Development, Inc.
All Rights Reserved.
Version 2.0.1-LLH7a404_11 0001
*****
losh> mount fatfs /cf
losh> █
```

Figure 2.1: Use the 'mount' Command to Mount CompactFlash Card

6. Use the 'cd' command to change directories to the CompactFlash card, and then type 'ls' to confirm that your card has been mounted and the NK.bin, contained in the downloaded .zip file, is on the card.

Figure 2.2: Using the 'cd' and 'ls' Commands



When the 'losh>' prompt returns, your NK.bin has been successfully loaded into system RAM and will be ready to run.

Please note that the sample NK.bin images provided on Logic's website are linked to run from system RAM and cannot be "burned" to system flash. Since the sample NK.bin images are volatile, they need to be loaded from a download mechanism (like CompactFlash) after every power cycle.

2.2 Sending LogicLoader a boot script to Boot the Image

Now send LogicLoader a boot script to boot the image. A boot script is a series of arguments that tells LogicLoader how to boot the Windows CE 5.0 Sample Dev Image. These arguments can be stored in a text file and then sent to LogicLoader using the 'Send File' functionality within TeraTerm.

1. The first step is to store the boot script in a text file. Example boot scripts for Logic's Display Kits are included in the same folder as the Sample Development Image. For a more thorough description of the arguments contained in this boot script please refer to Logic's *LoCE Windows Embedded CE BSP User's Guide* for values that are specific to your SOM.

In the example figure below, a .txt file with the name "LH7A404_RAM_3.5-10.txt" was created and stored in the development workstation's file system for later use.

Some boot scripts include a parameter that will auto-launch the touch calibration utility (**touchcal:1**). The touch calibration utility will not run if this parameter is removed from the boot script. (The touch calibration parameter is currently not supported on all sample images. Please review the example boot scripts included with the sample image to determine if the parameter is supported by that sample image.)

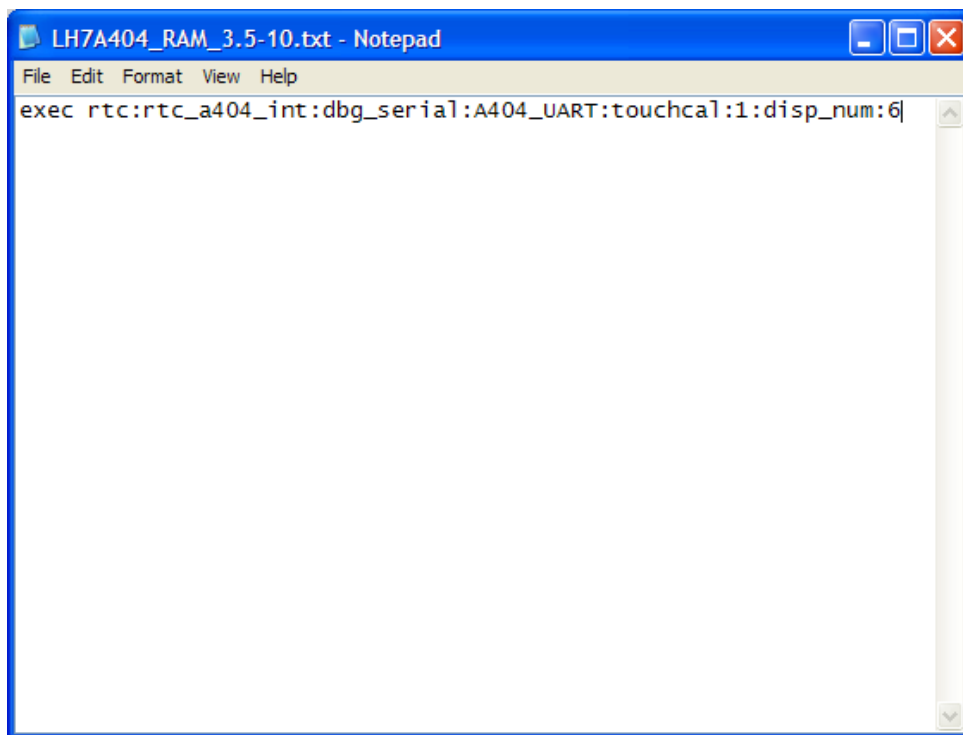


Figure 2.4: Storing the Boot Script in a Text File

2. Go to 'File | Send File' and select the boot script that you want to send LogicLoader to boot the image, and then click 'Open'.

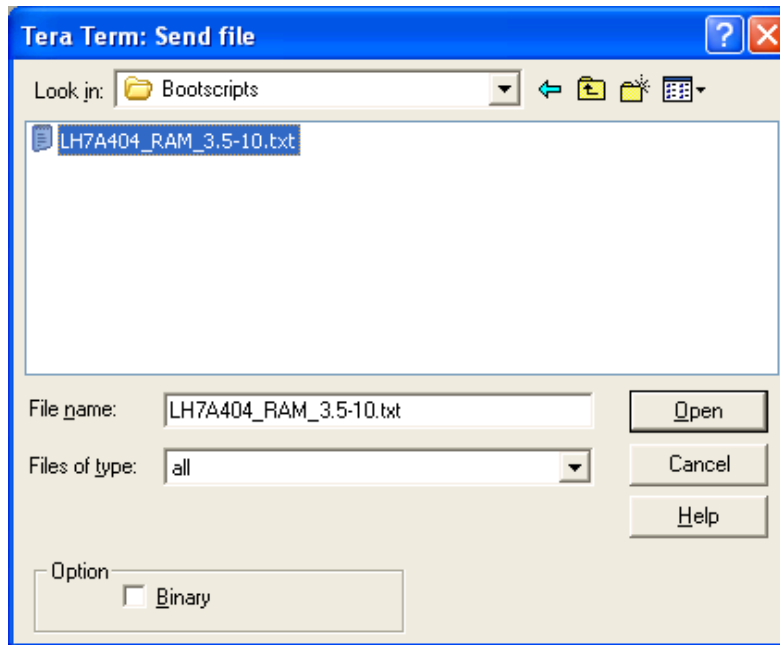


Figure 2.5: Select the Boot Script to Send to LogicLoader

3. You may need to use the 'Enter' key to send the script to LogicLoader.

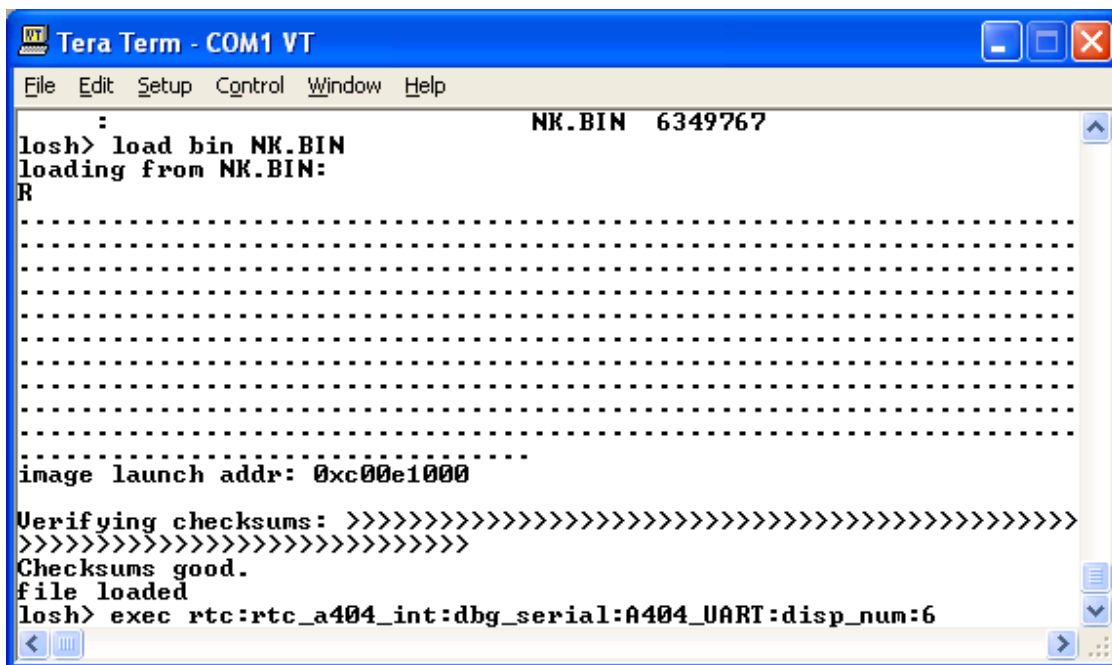
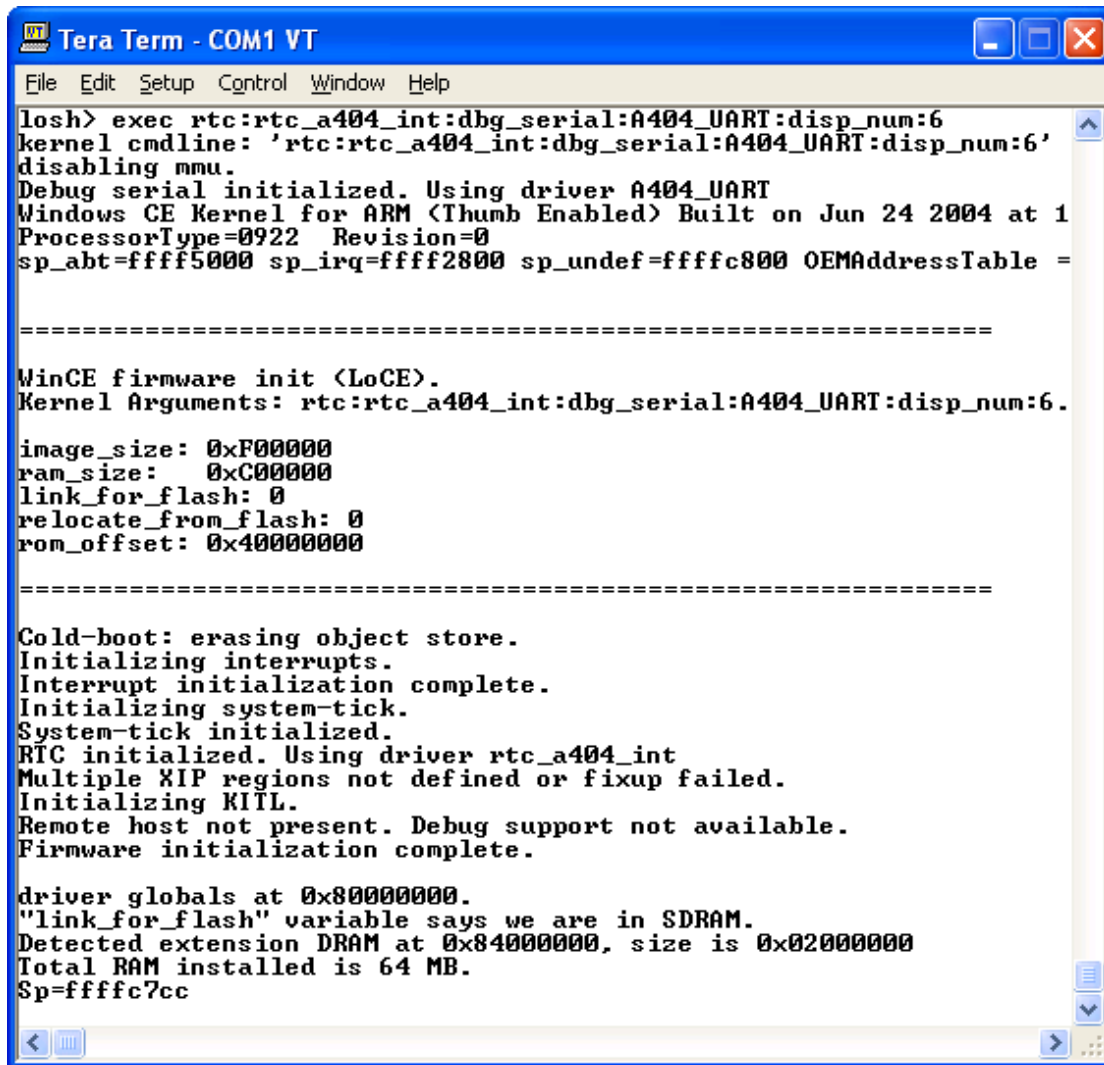


Figure 2.6: Pressing the ‘Enter’ Key to Send the Script to LogicLoader

4. Output from the debug serial port in LogicLoader will look similar to the image below. After a moment, the Sample Dev Image will load.



```
losh> exec rtc:rtc_a404_int:dbg_serial:A404_UART:disp_num:6
kernel cmdline: 'rtc:rtc_a404_int:dbg_serial:A404_UART:disp_num:6'
disabling mmu.
Debug serial initialized. Using driver A404_UART
Windows CE Kernel for ARM <Thumb Enabled> Built on Jun 24 2004 at 1
ProcessorType=0922 Revision=0
sp_abt=ffff5000 sp_irq=ffff2800 sp_undef=ffffc800 OEMAddressTable =

=====

WinCE firmware init <LoCE>.
Kernel Arguments: rtc:rtc_a404_int:dbg_serial:A404_UART:disp_num:6.

image_size: 0xF00000
ram_size: 0xC00000
link_for_flash: 0
relocate_from_flash: 0
rom_offset: 0x40000000

=====

Cold-boot: erasing object store.
Initializing interrupts.
Interrupt initialization complete.
Initializing system-tick.
System-tick initialized.
RTC initialized. Using driver rtc_a404_int
Multiple XIP regions not defined or fixup failed.
Initializing KITL.
Remote host not present. Debug support not available.
Firmware initialization complete.

driver globals at 0x80000000.
"link_for_flash" variable says we are in SDRAM.
Detected extension DRAM at 0x84000000, size is 0x02000000
Total RAM installed is 64 MB.
Sp=ffffc7cc
```

Figure 2.7: Output from the Debug Serial Port

2.2.1 Connecting to the Image with Microsoft ActiveSync

After the Windows CE image has booted, connect a USB AB device cable from the USB function connector on the Development Kit to a USB port on the development workstation. **Important Note:** You will need to download and install the latest version of Microsoft ActiveSync before you proceed with the following steps. ActiveSync can be found at: <http://www.microsoft.com/windowsmobile/downloads>.

1. If this is the first time that the workstation is connecting to a Zoom Development Kit, a typical Windows 'New Hardware Wizard' will appear. If you are running the latest version of Microsoft ActiveSync, the USB driver for your workstation will be automatically installed as shown in Figure 2.8.

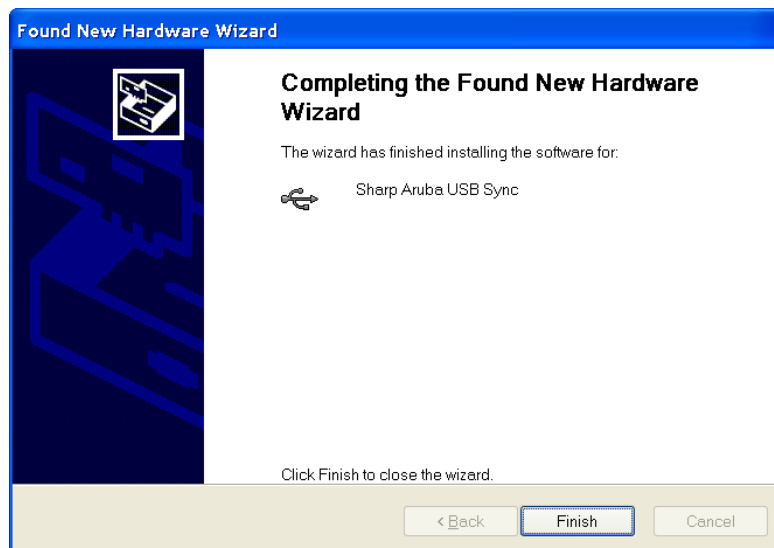


Figure 2.8: Windows XP will Automatically Install the USB Driver

2. Once the USB driver is installed on the development workstation, the PC will be able to connect to the Development Kit over ActiveSync. At this point you may choose to set up a new partnership with the Zoom Development Kit. For the purposes of this sample evaluation, it is only necessary to connect as a guest. Select 'No' and then click 'Next'.



Figure 2.9: Connect as a Guest

3. The PC is now connected to the Development Kit via ActiveSync.

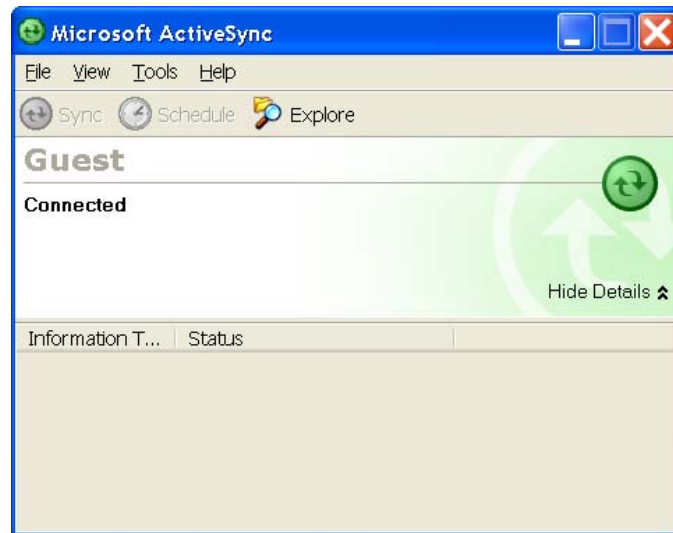


Figure 2.10: Microsoft ActiveSync: Connected

3 Using the Sample Dev Image and eVC++ to Debug an Application

Once you have the Development Kit running Windows CE and connected via ActiveSync, it is possible to connect the eMbedded Visual tools to the kit for application development.

The following steps give an overview of how to build a sample application in Microsoft eMbedded Visual C++ and how to use the remote tools and debugger over the ActiveSync connection.

1. First, create a new eVC++ project:
 - ☐ Open eVC++ from the Start Menu and click 'File | New'.
 - ☐ Under the 'Projects' tab select 'WCE Application'.
 - ☐ For 'Project name', type 'boot_app', select the processor that you want to build this application for, and then click 'OK.'

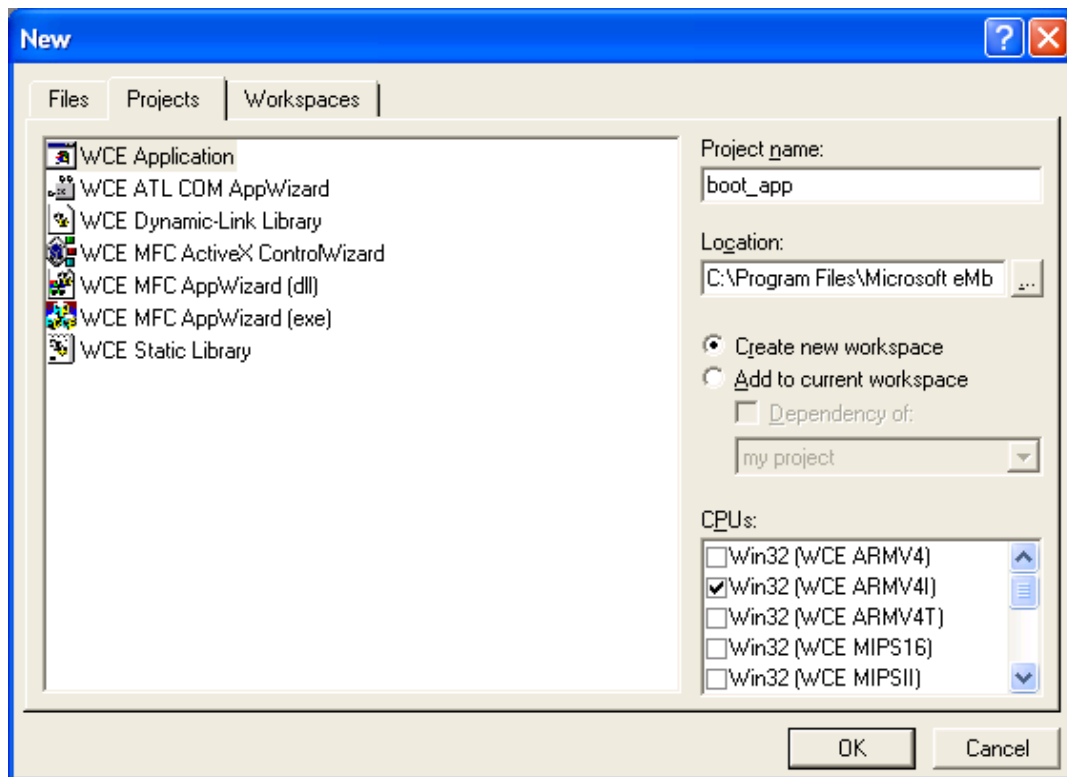


Figure 3.1: Create a new eVC++ Project

2. Next, select 'A Typical Hello World Application' and click 'Finish.' See Figure 3.2 below.

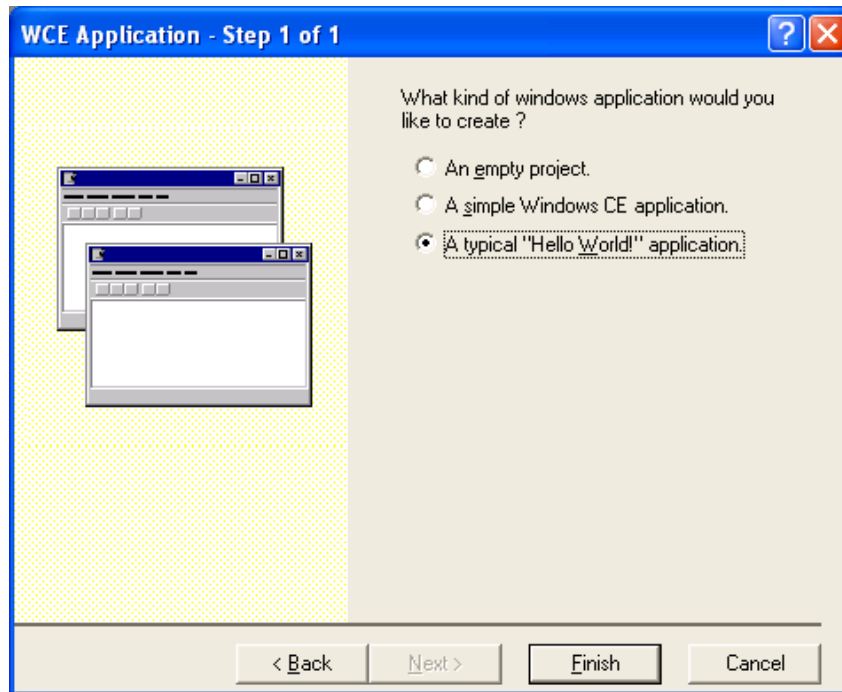


Figure 3.2: Select 'A typical Hello World application' and click 'Finish'

3. A 'New Project Information' window will appear. Click 'OK' to continue:

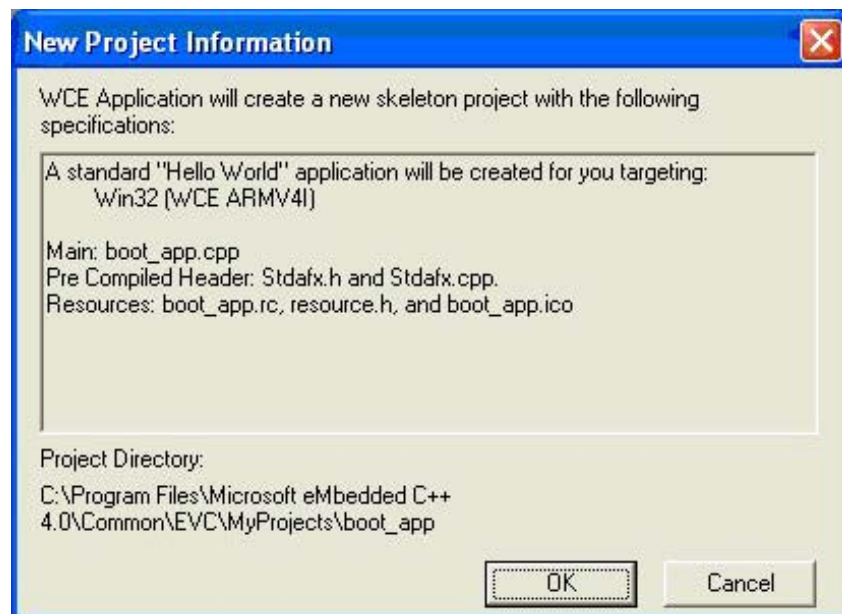


Figure 3.3: New Project Information Window

4. Configure Platform Manager to connect to your image over ActiveSync. To do this, click on 'Tools | Configure Platform Manager'. From here you'll need to find the platform that has 'Dev Image' in the title. In the example below the target platform is A404_DEV_IMAGE_REV_C. Click the '+' box to expand your list of devices for that platform.

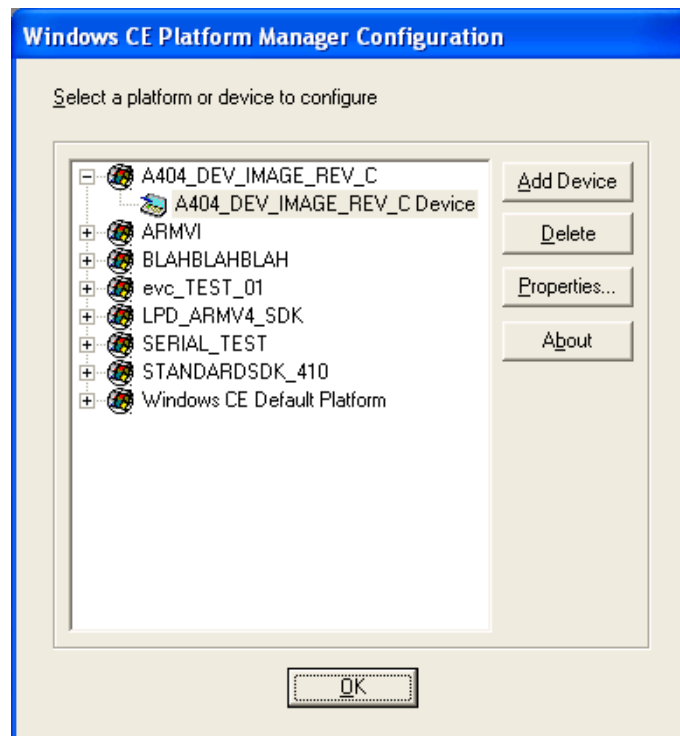


Figure 3.4: Select a Platform or Device to Configure

5. Once your platform is highlighted, click the 'Properties' button. In the window that opens, select 'Microsoft ActiveSync' for both the 'Transport' and 'Startup Server' options. Then click 'Test'. See Figure 3.5 below.

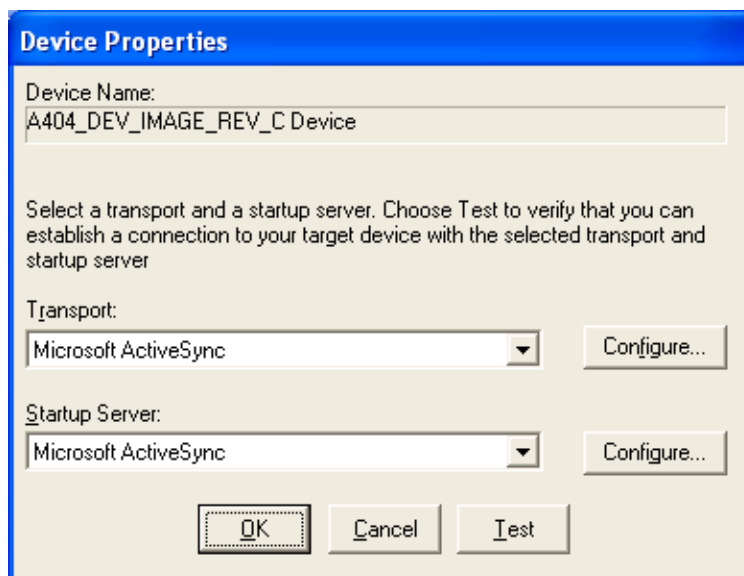


Figure 3.5: Select Microsoft ActiveSync

6. The 'Test' button in the previous step verifies a connection to your device. Once a connection is established, click 'OK.' Continue clicking 'OK' on the remaining windows until you return to the eVC++ workspace.



Figure 3.6: Once a Connection is Established, Click 'OK'

7. Make sure the following settings are appropriate for your eVC++ workspace: Active Project, Active WCE Configuration, Active Configuration, and Default Device.

See Figure 3.7 below for an example of a session using the LH7A404-11 Starter Development Kit with a Rev C sample image and SDK would be as follows:

- Active Project = boot_app
- Active WCE Configuration = A404_DEV_IMAGE_REV_C
- Active Configuration = Win32 (WCE ARMV4I) Debug
- Default Device = A404_DEV_IMAGE_REV_C Device

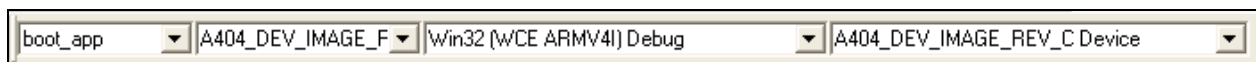


Figure 3.7: Example Session using the LH7A404-11

- Click the 'FileView' tab, located at the bottom of the workspace window. Expand the 'boot_app files' by clicking on the '+' icon, and do the same for 'Source Files'. Then open 'boot_app.cpp' by double-clicking on it. See Figure 3.8 below.

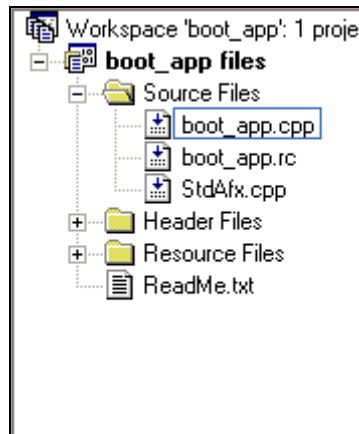


Figure 3.8: Open 'boot_app.cpp'

- In order to step into code and debug the application, a break point must be inserted into the source. Right-click where you want to insert the break point and select 'Insert/Remove Breakpoint'.

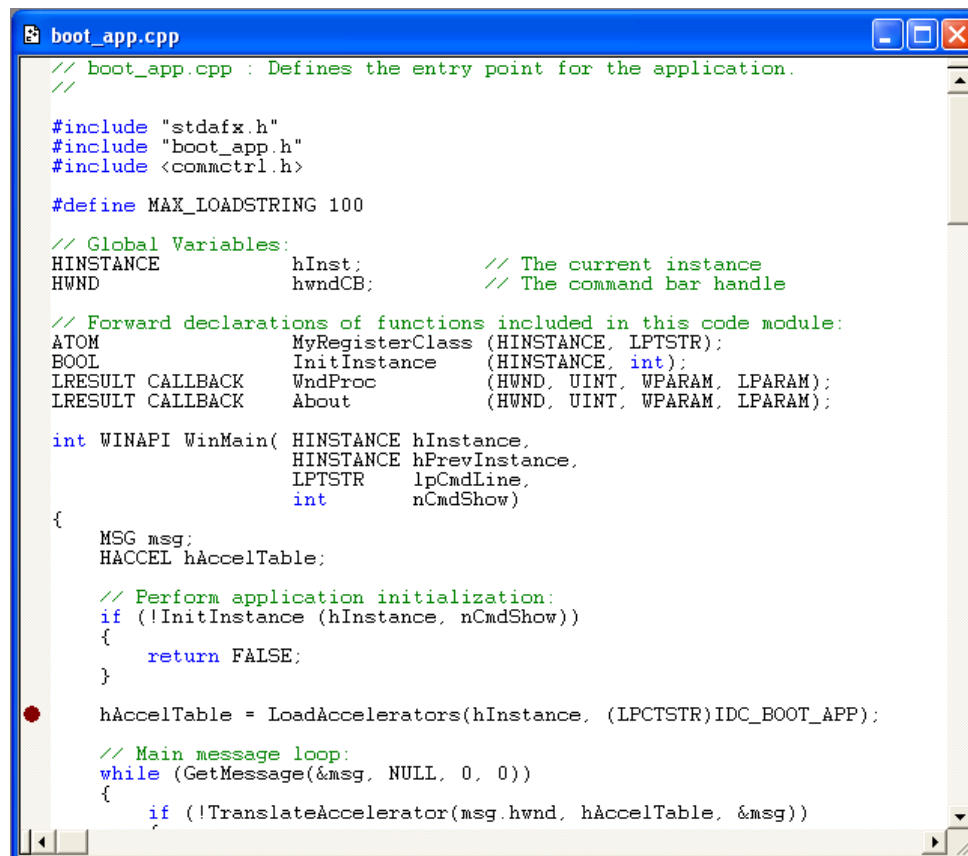


Figure 3.9: Click 'Insert/Remove Breakpoint'

10. Press the 'F7' key to build the application. The application will download to the device and will be ready to be run.

```
-----Configuration: boot_app - Win32 (WCE ARMV4I) Debug-----
boot_app.exe - 0 error(s), 0 warning(s)
Downloading files
Downloading file c:\program files\...\boot_app\armv4idbg\boot_app.exe.
Finished downloading.
```

Figure 3.10: Build the Application

11. The application is now ready to run. Press 'CTRL + F5' or click the '!' icon to begin executing the application. See the image below for an example of what the tool bar looks like:



Figure 3.11: '!' Icon in the Tool Bar

12. Execution of the application will stop at the break point that was inserted and the Debugger will be running. At this point, the interface is typical of Microsoft development environments, providing context, watches, etc.

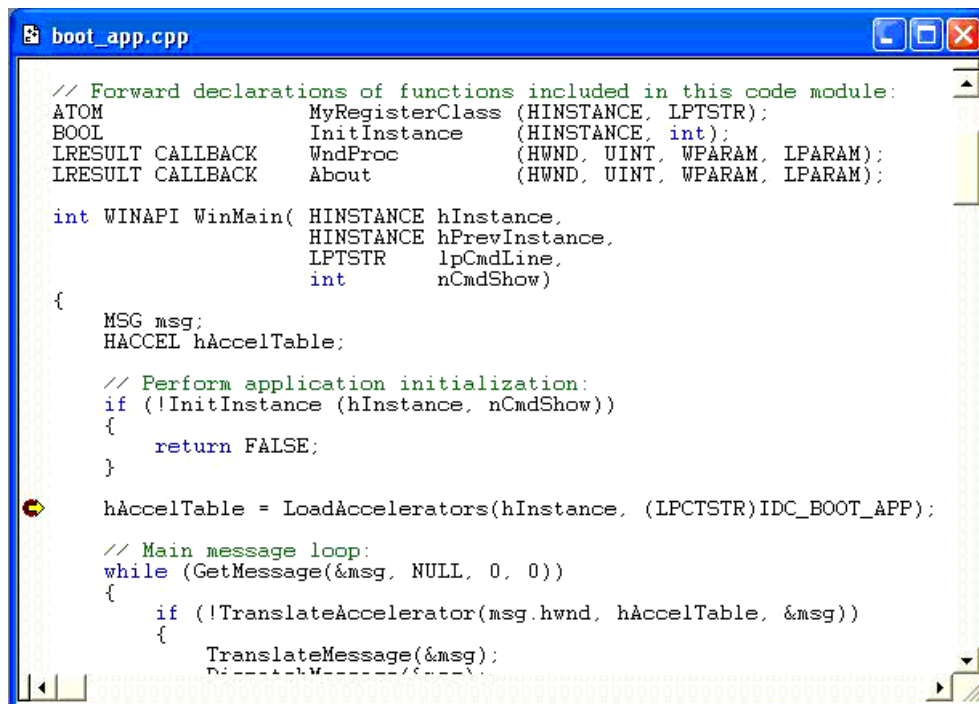


Figure 3.12: Development Interface Screenshot 1

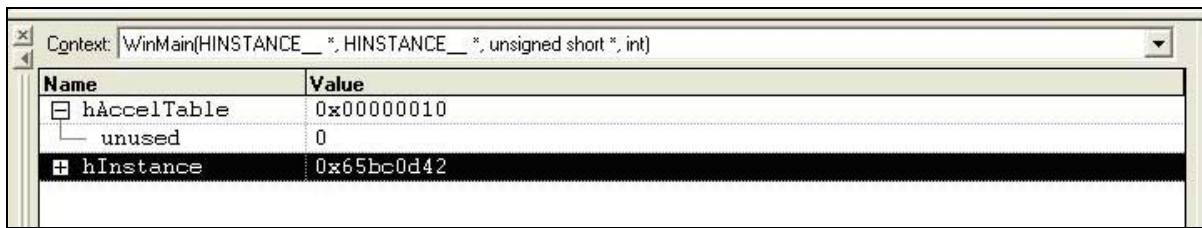


Figure 3.13: Development Interface Screenshot 2

13. You can also use 'Remote Tools' for further analysis of your application. **Note:** 'Remote Tools' must be used during the execution of the application. For more information on using 'Remote Tools', please reference Microsoft eVC++ 'Help.' See Figure 3.14 below for an image of the Remote Tools options.

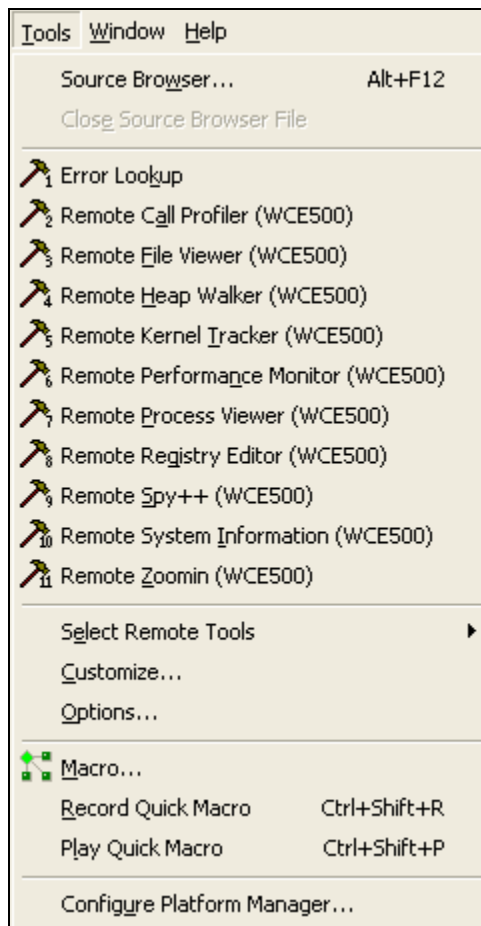


Figure 3.14: Remote Tools Options

4 Auto-Launching an Application

The Windows CE Sample Image automatically launches an application on the Development Kit's file system at startup when an application called "boot_app.exe" is present on a CompactFlash card in the memory mapped CompactFlash slot. Please note that this will only work if the Windows CE image being used has driver and OS support built as part of the image.

The default application name and path can be changed by editing the applicable registry key as shown below.

4.1 Example Auto-Boot Module

```
[HKEY_LOCAL_MACHINE\SOFTWARE\bootapp]
"time"="0"
"path" = "\\Storage Card\\boot_app.exe"
```

"path" description: The auto-boot module will launch an application from a user-defined location as specified by the "path" registry key.

"time" description: The auto-boot module will delay launching the application for the amount of time specified in milliseconds in the "time" registry key.

Since the sample OS image binary is configured to use the hive-based registry, changes to these keys and values can be made permanent by calling "RegFlushKey" or by enacting a Suspend/Resume sequence on the system.