> LOGIC PD™

Enabling Windows® Embedded CE 6.0 Application Debug in Visual Studio® 2005 on Zoom™ OMAP35x Development Kits
Application Note 402

Abstract

This application note provides instructions for establishing a communication link between a Zoom™ OMAP35x Development Kit and a host PC running Visual Studio 2005. This communication link provides the ability to test and debug hardware and application software.

Revision History

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	BSB	Initial Release	BSB	04/09/09
B	SMC, BSB	-Revised references to generalize for both development kits; -Added Known Issues section; -Updated the Additional Reading list in Section 9.1; -Verified instructions using BSP 2.1.0 release; -General formatting changes	BSB, JCA	08/05/11

Table of Contents

1   Introduction

A new connectivity framework known as CoreCon can establish a connection between your Zoom™ OMAP35x Development Kit running Windows Embedded CE 6.0 (CE 6.0) and your Visual Studio 2005 (VS2005) development system. Establishing this connection allows you to download custom VS2005 applications to the OMAP35x Development Kit and test and debug your application under development.

2   Development Requirements

Your development environment must have the following items setup before proceeding with this application note:

- OMAP35x Development Kit or OMAP35x Torpedo Development Kit connected to your host PC through an Ethernet cable.

- Host PC running VS2005 with all the required service packs and QFEs. For detailed information on installing the necessary components, please see the _OMAP35x Windows Embedded CE 6.0 BSP User Manual_[1] from Logic PD's website.

- OMAP35x Windows Embedded CE 6.0 Source BSP release version 0.9.0 or later, available for download from Logic PD's website[2]. Please see the _OMAP35x Windows Embedded CE 6.0 BSP User Manual_ for important information about the BSP.

The following required items will be discussed within this application note:

- CoreCon framework files (see Section 3).

- Software Development Kit (SDK) built from the OMAP35x Windows Embedded CE 6.0 Source BSP (see Section 4).

- Your custom CE 6.0 application (see Section 5).

3   Add Required CoreCon Files

To establish a connection using the CoreCon framework, several component files will need to be copied to the OMAP35x Development Kit or built into the BSP image. The required CoreCon component files are:

- _Clientshutdown.exe_

- _ConmanClient2.exe_

- _CMAccept.exe_

- _DeviceDMA.dll_

- eDbgTL.dll

- _TcpConnectionA.dll_

These files are installed on your development system during installation of the VS2005 development tools.

You only need to follow the directions in either Section 3.1 or Section 3.2; please choose the method that best fits your situation.

---

[1] http://support.logicpd.com/downloads/1347/
[2] http://support.logicpd.com/downloads/1104/

3.1 Copy CoreCon Files to OMAP35x Development Kit

To copy the CoreCon files directly to your OMAP35x Development Kit, perform the following steps:

1. Navigate to the following directory:

   .\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i

2. Copy the required files to the *\Windows* directory on your OMAP35x Development Kit.

3. Proceed to Section 4.

3.2 Build CoreCon Files into CE 6.0 Image

To build these files directly into your CE 6.0 image, perform the following steps:

1. Copy the required CoreCon files from the following directory:

   .\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i

2. Copy these files to the BSP files folder on your development system located here:

   .\wince600\PLATFORM\LPD_OMAP35X_SOM\files

3. In this BSP files folder on your host PC, locate the *platform.bib* file and open the file in a text editor.

4. In the *platform.bib* file, find the FILES section and add the following lines of text below the FILES header:

```
clientshutdown.exe      $(_FLATRELEASEDIR)\clientshutdown.exe      NK SH
CMAccept.exe            $(_FLATRELEASEDIR)\CMAccept.exe            NK
ConmanClient2.exe       $(_FLATRELEASEDIR)\ConmanClient2.exe       NK
DeviceDMA.dll           $(_FLATRELEASEDIR)\DeviceDMA.dll           NK SH
eDbgTL.dll              $(_FLATRELEASEDIR)\eDbgTL.dll              NK SH
TcpConnectionA.dll      $(_FLATRELEASEDIR)\TcpConnectionA.dll      NK SH
```

This snapshot shows what the file will look like after adding the lines of text.

```
406
407  ;=====================================================================
408
409  FILES
410
411  clientshutdown.exe      $(_FLATRELEASEDIR)\clientshutdown.exe      NK SH
412  CMAccept.exe            $(_FLATRELEASEDIR)\CMAccept.exe            NK
413  ConmanClient2.exe       $(_FLATRELEASEDIR)\ConmanClient2.exe       NK
414  DeviceDMA.dll           $(_FLATRELEASEDIR)\DeviceDMA.dll           NK SH
415  eDbgTL.dll              $(_FLATRELEASEDIR)\eDbgTL.dll              NK SH
416  TcpConnectionA.dll      $(_FLATRELEASEDIR)\TcpConnectionA.dll      NK SH
417
418  ; @CESYSGEN IF CE_MODULES_DEVICE
```

5. Close the text editor, saving your changes to the *platform.bib* file.

6. Build your BSP by following the directions provided in the *OMAP35x Windows Embedded CE 6.0 BSP User Manual*.

   NOTE: If your BSP has been successfully built prior to the changes listed above, using the *Build Current BSP and Subprojects* option will be sufficient. In order to do this, select Build > Advance Build Commands > Build Current BSP and Subprojects.

4 Create an SDK in VS2005

This section describes how to create an SDK on your VS2005 development system. If you already have an SDK installed on your system, proceed to Section 5.

1. Load your OMAP35x CE 6.0 BSP into VS2005. (For detailed information on how to do this, please see the *OMAP35x Windows Embedded CE 6.0 BSP User Manual*.

2. Select the necessary Applications and Services Development components in the BSP *Catalog Items View* window (see those items with checkmarks in the following image).
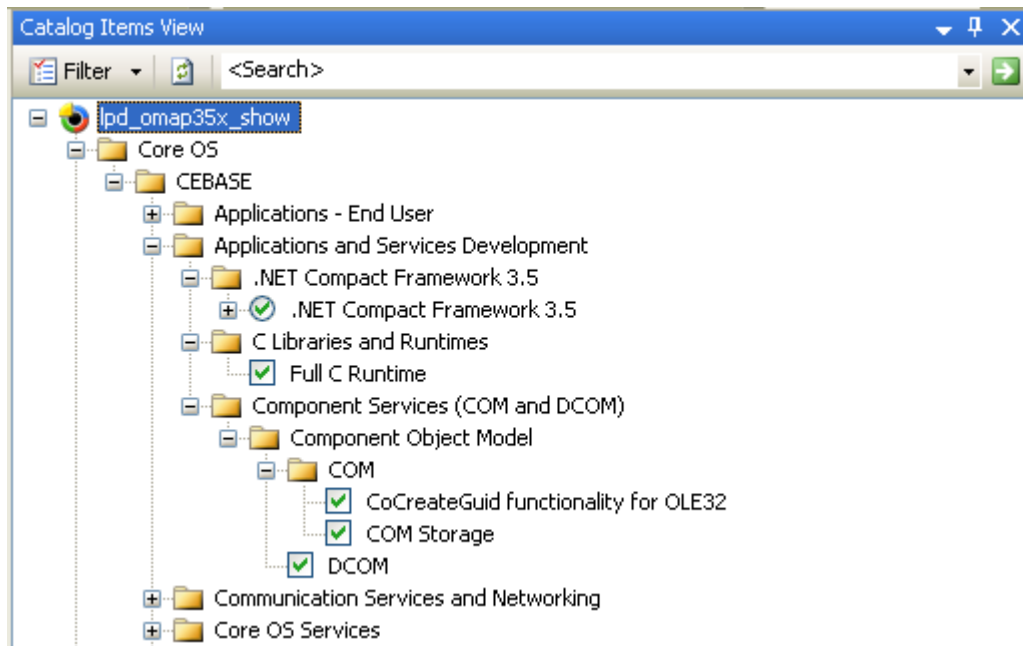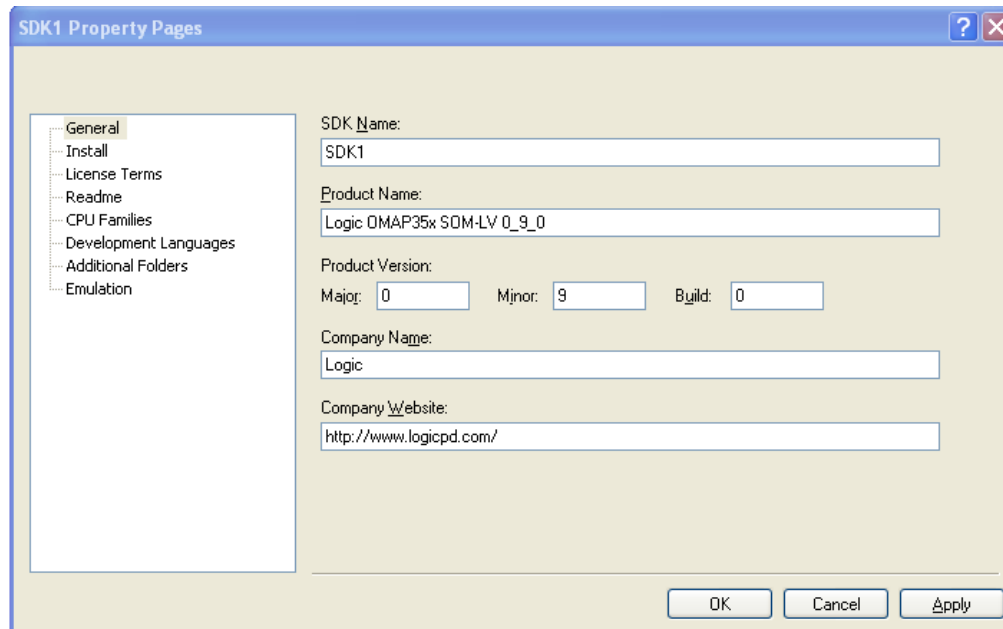

3. In your VS2005 project window, right-click on **SDK** and select **Add New SDK…**
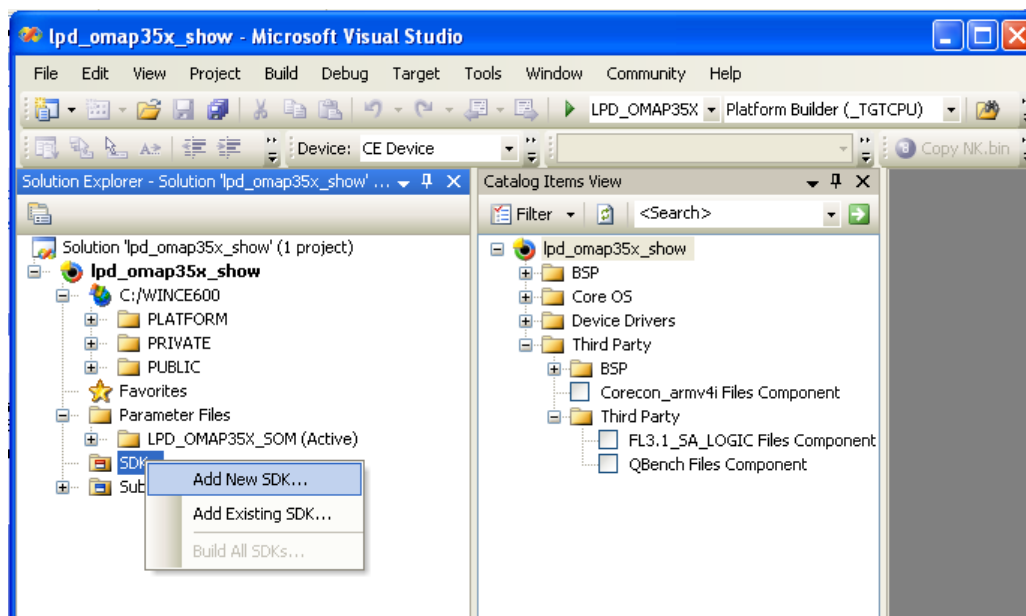
4. In the window that appears, fill in the fields and click **OK**.

   NOTE: For the purpose of these instructions, we use the name SDK1. As you encounter the SDK1 name within this document, please substitute it with the SDK Name you choose in this step.


5. To build the SDK, make sure you are in the VS2005 OSDesign window. Expand the *SDKs* folder, then right-click on **SDK1** and select **Build**.
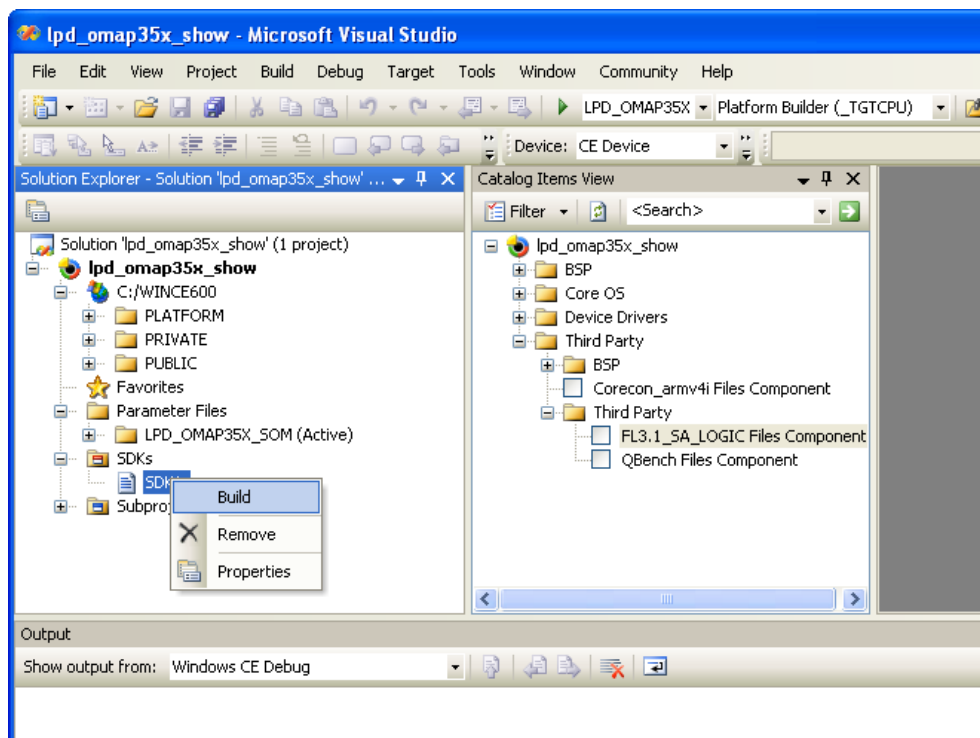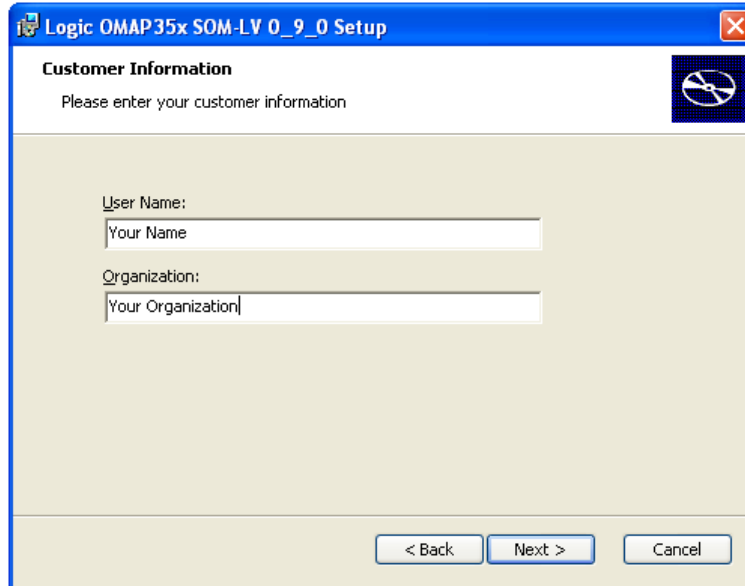
6. After your SDK build has succeeded, you will need to install the MSI file created by the build. The *SDK1.msi* file location is shown in the last line of the compile. The following snapshot shows the *SDK1.msi* file is located in:

   C:\WINCE600\OSDesigns\LPD_OMAP35x_0.9.0\lpd_omap35x_0.9.0_base\SDKs\SDK1\MSI\SDK1.msi
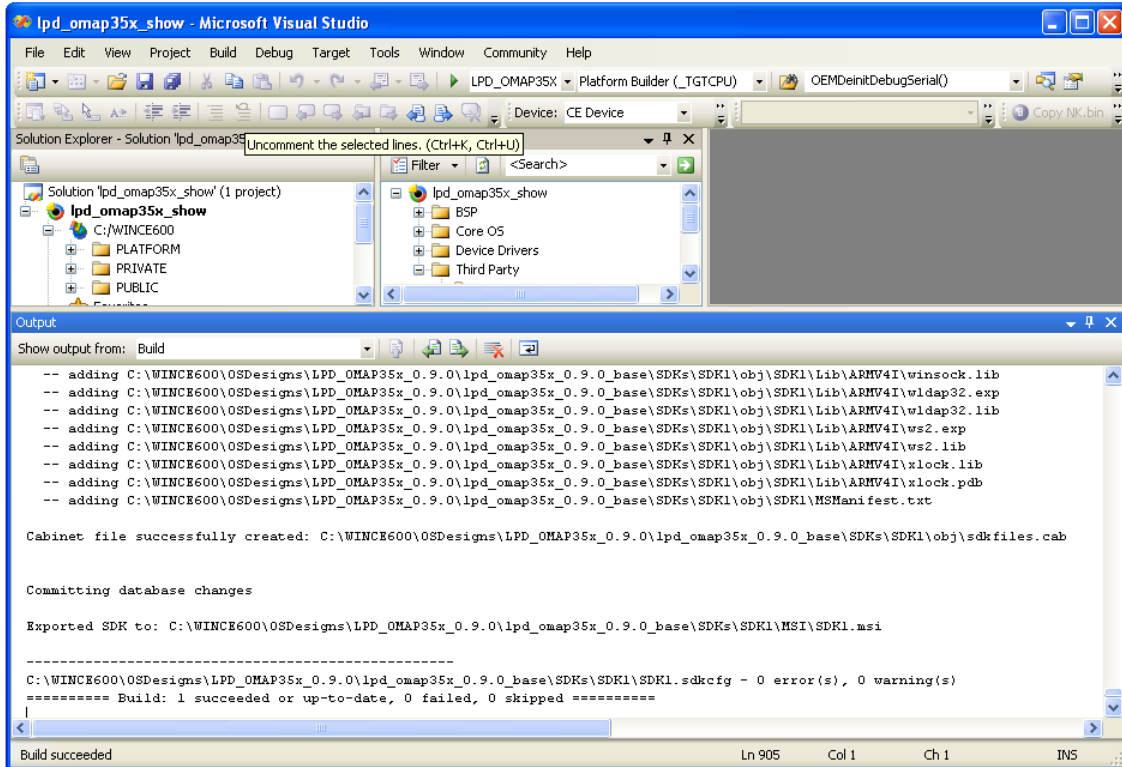
7. Navigate to the folder where the *SDK1.msi* file is located and double-click **SDK1.msi** to start the installation process.

8. When the *Setup Wizard* window opens, click **Next** to continue.

9. In the *License Agreement* window, select **Accept** and then click **Next** to continue the installation.

10. In the *Customer Information* window, enter your name and your organization. Then click **Next**.

11. In the *Choose Setup Type* window, click the **button next to Complete**. Then click **Next**.

12. Select your desired installation path and click **Next**.
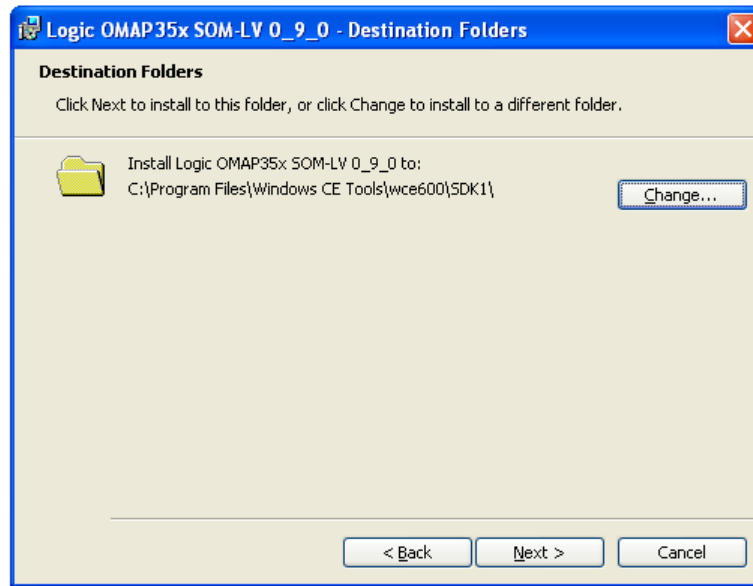
13. In the *Ready to Install* window, click **Install**.

14. After your SDK installation has completed, click **Finish** to exit the Setup Wizard.

5   Create an Application to Debug

This section describes how to use the VS2005 Application Wizard to create an application for your OMAP35x Development Kit. If you already have an application to debug, proceed to Section 6.

1. Launch VS2005 on your host PC.

2. Create a project by using the Project Design Wizard. Select **File > New > Project…** to launch the wizard.
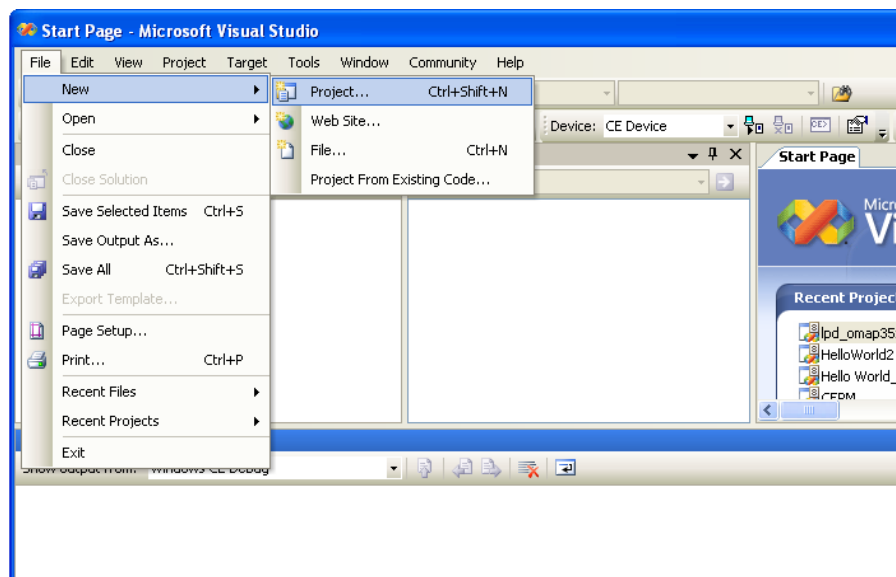
3. Select the type of project you wish to create. In our example, we will select a Visual C++, MFC Smart Device Application.

4. Enter the name of your project in the *Name:* field and click **OK**.
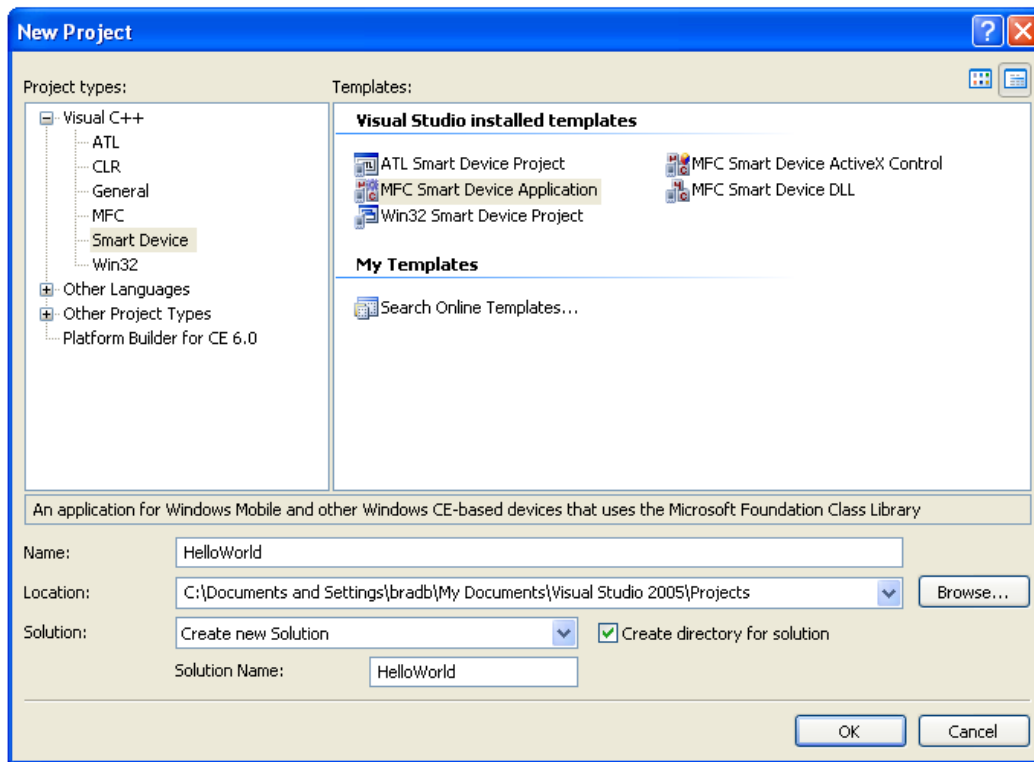
5. A *Welcome to the MFC Smart Device Application Wizard* window will open. There are a few project settings to change, so click **Next**.
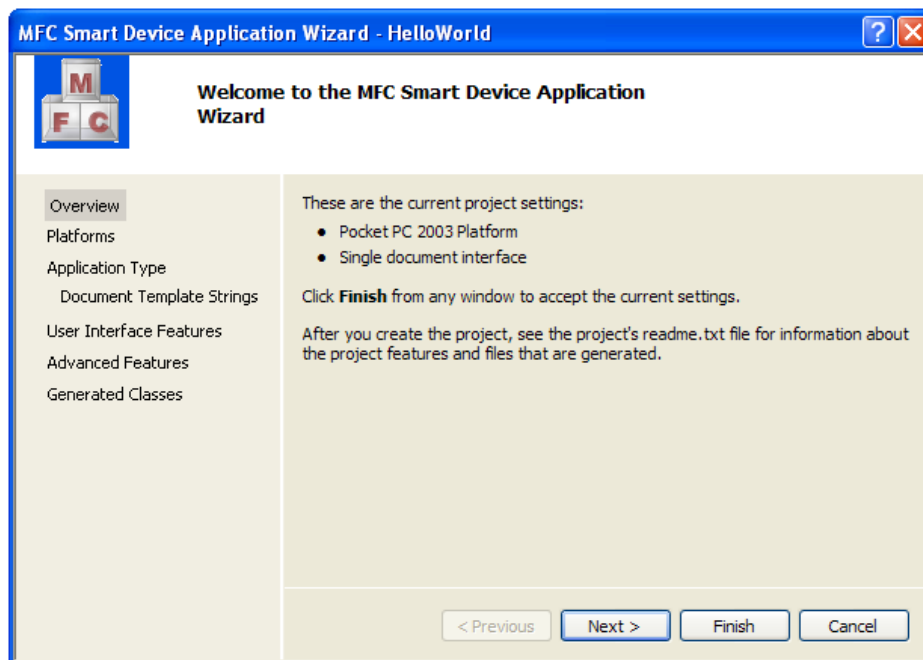
6. In the *Platforms* window, locate the *Selected SDKs:* section. Remove **Pocket PC 2003** and add **SDK1** to the *Selected SDKs:* section. Click **Finish** to complete the creation of your HelloWorld application.
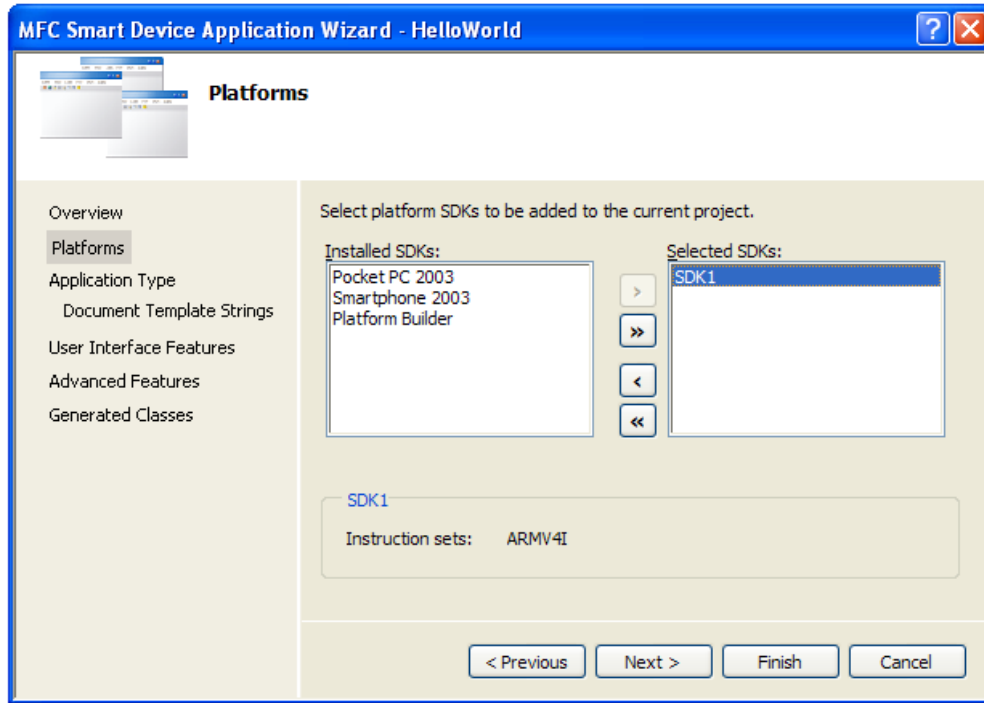
# 6 Prepare the OMAP35x Development Kit

This section describes the steps you will need to perform on the OMAP35x Development Kit before establishing a connection with your VS2005 development system.
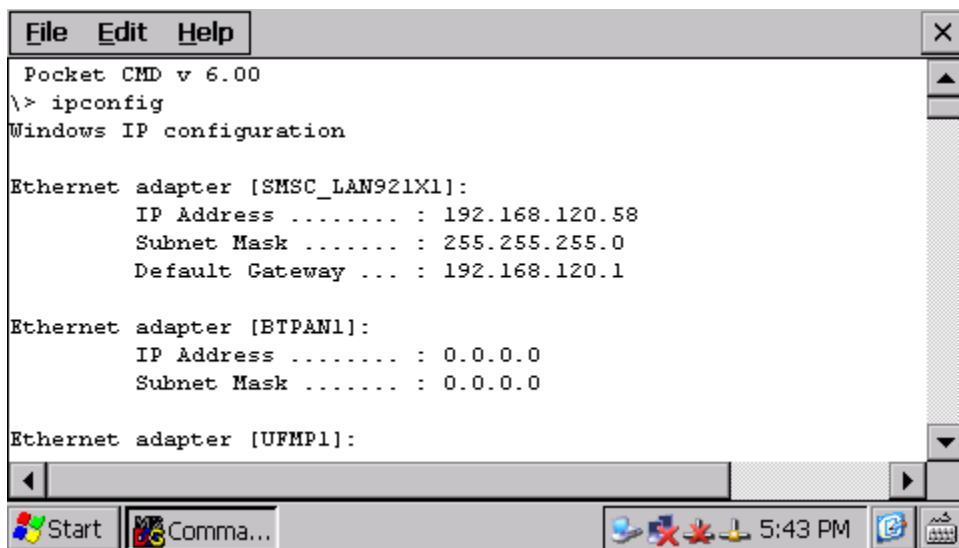
6.1 Determine IP Address

To setup your VS2005 development system environment, you will need to know the IP address of your OMAP35x Development Kit.

1. After you have built your CE 6.0 image, boot the image on your OMAP35x Development Kit.

2. Select **Start > Run** to open the *Run* window.

3. In the *Open:* field, type `cmd` and click **OK**.
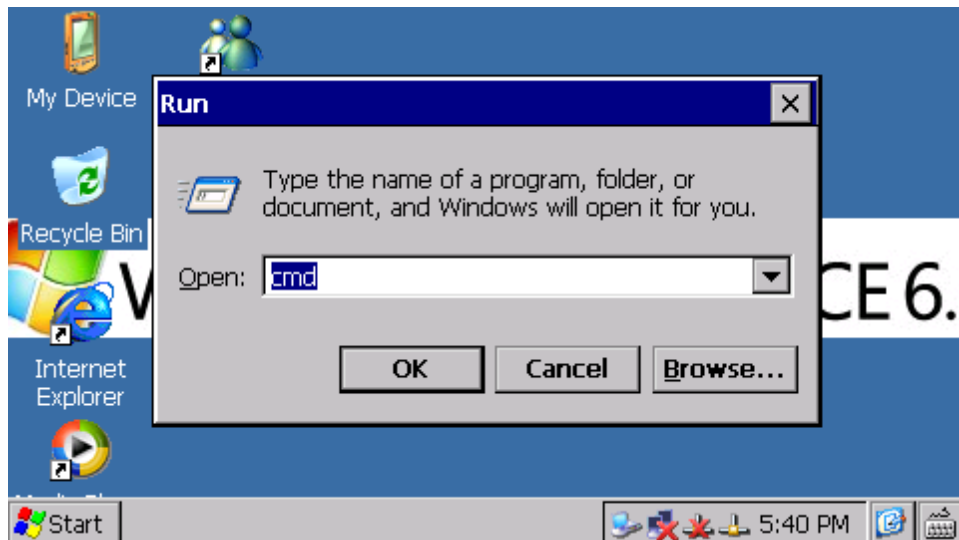
4. Type `ipconfig` at the `\>` prompt and press **Enter**.

5. Scroll to the top of the window until you can see your IP address (in this example the IP address is 192.168.120.58). Write down your IP address, as you will need this information when setting up the connection to your VS2005 development system.
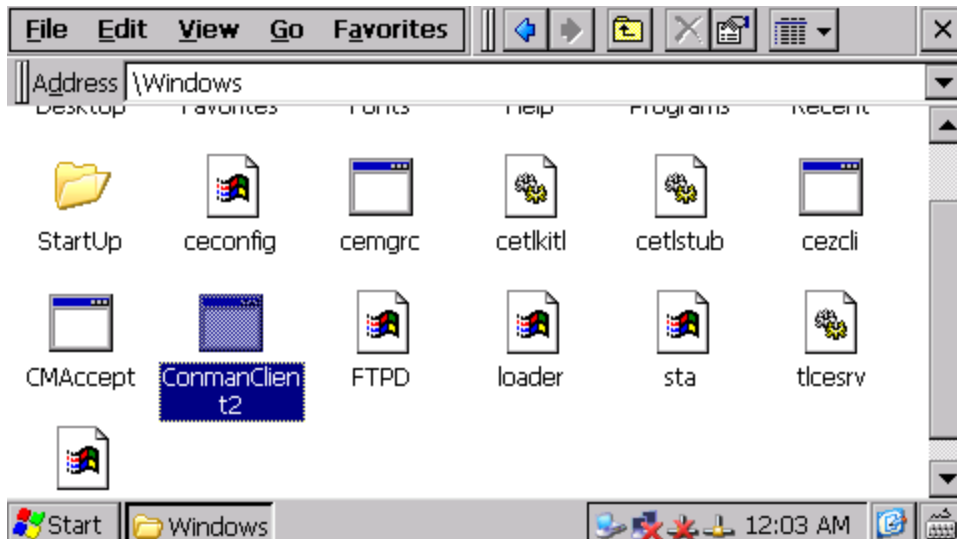
6.2 Execute Necessary Applications

1. Close the *Command* window after you have written down your IP address.
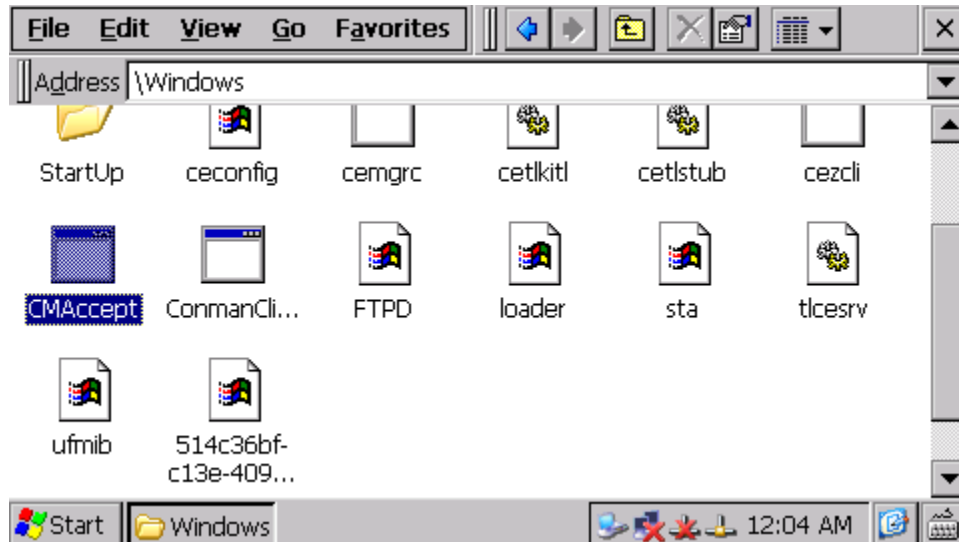
2. Launch Windows Explorer by selecting **Start > Programs > Windows Explorer**.

3. Double-click on the **Windows** folder to open it.

4. In the *Windows* directory, double-click on **ConmanClient2.exe**.

5. Next, double-click on **CMAccept.exe** in the *\Windows* directory.

7 Connect the OMAP35x Development Kit to VS2005 Development System

This section describes how to configure your VS2005 development system and then connect to the OMAP35x Development Kit.

7.1 Configure the VS2005 Development System

1. To load the application you want to run and debug on the OMAP35x Development Kit, locate the *.sln file created for your application and double-click it (this location was set in Section 5).

   In our example, the *HelloWorld.sln* is located in the following directory:
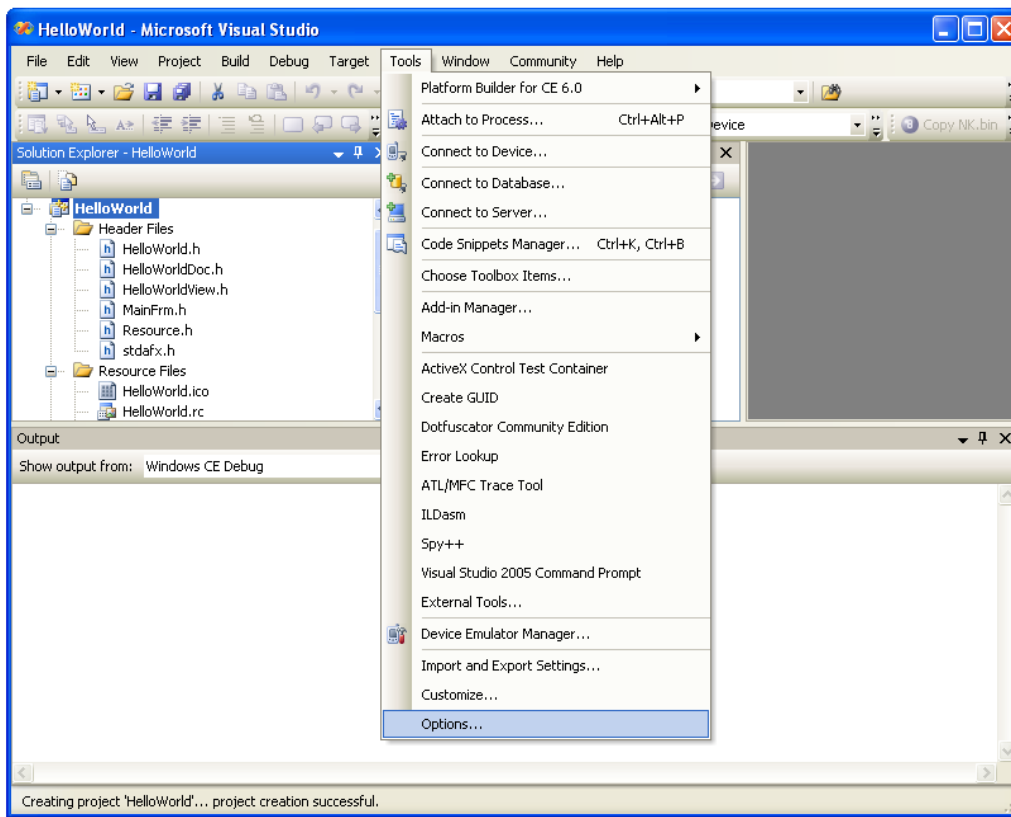
   C:\Documents and Settings\user\My Documents\Visual Studio 2005\Projects\HelloWorld

2. In the VS2005 window that opens, set the *Solution Platform* to be **SDK1 (ARMV4I)** and the *Target Device* to be **SDK1 ARMV4I Device**.

   NOTE: As shown in the following snapshot, the *Solution Platform* is located just below the *Help* menu.
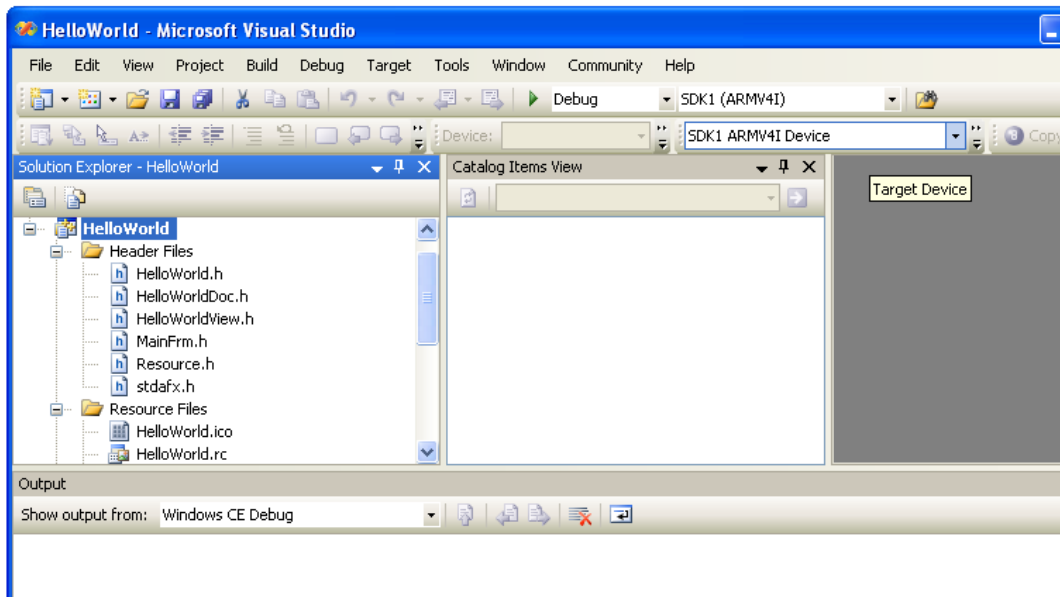

3. Select **Tools > Options...** to access the options for this project.

4. To change the properties setting for the SDK1 device:

   a. In the left section, expand the *Device Tools* folder and click on **Devices.**

   b. In the top section, select **SDK1** from the *Show devices for platform:* drop-down menu.
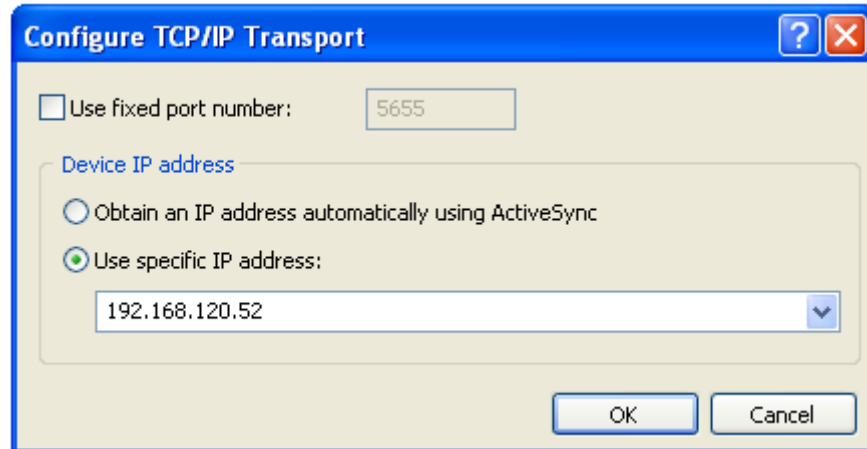
   c. Click **Properties…**
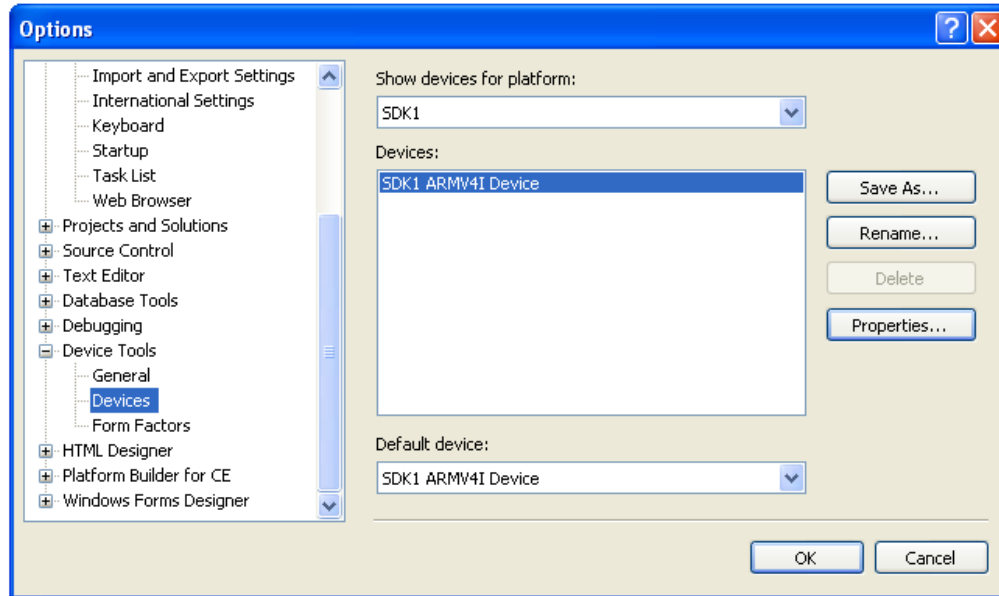
5. In the *Properties* window, select **TCP Connect Transport** from the *Transport:* drop-down menu and then click **Configure…**
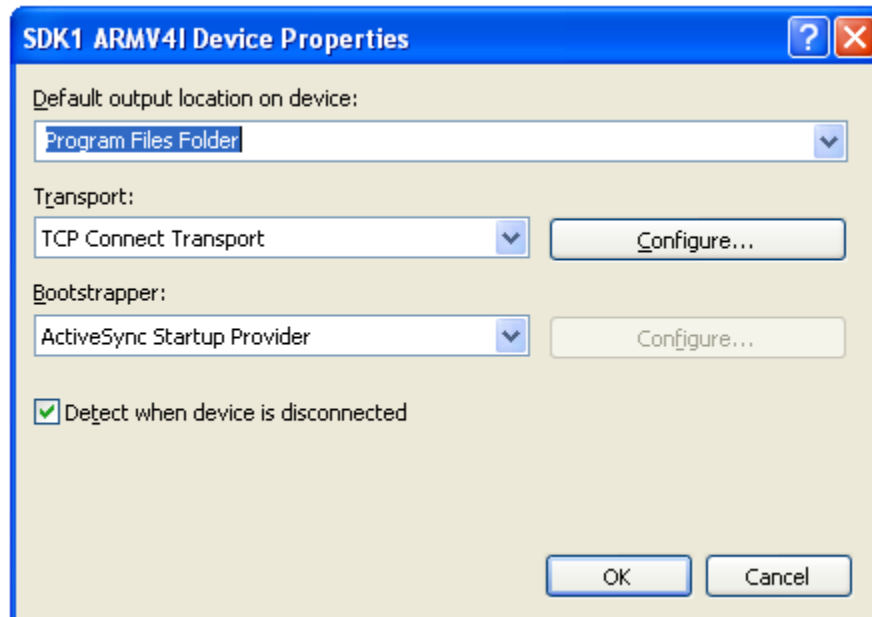
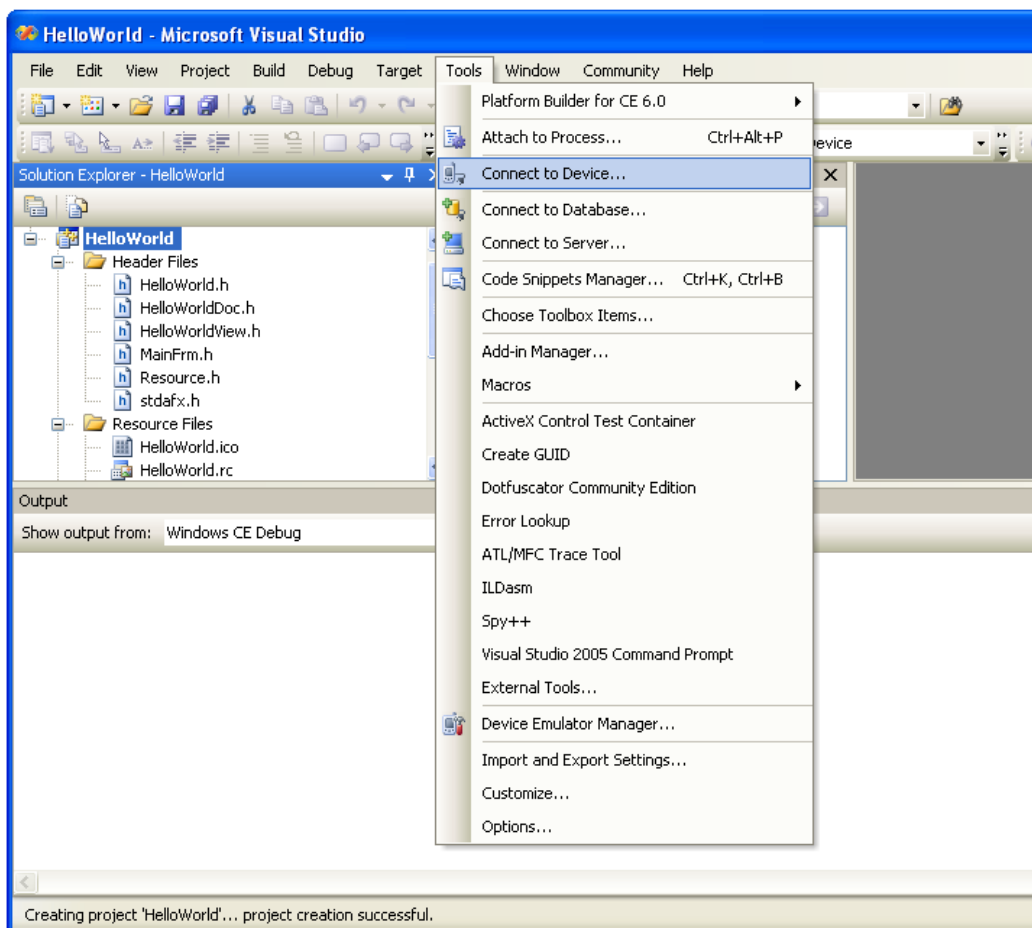
6. Select **Use specific IP address:**.

7. Enter the IP address of your OMAP35x Development Kit that you wrote down in Section 6.1 (in this example the IP address is 192.168.120.52).
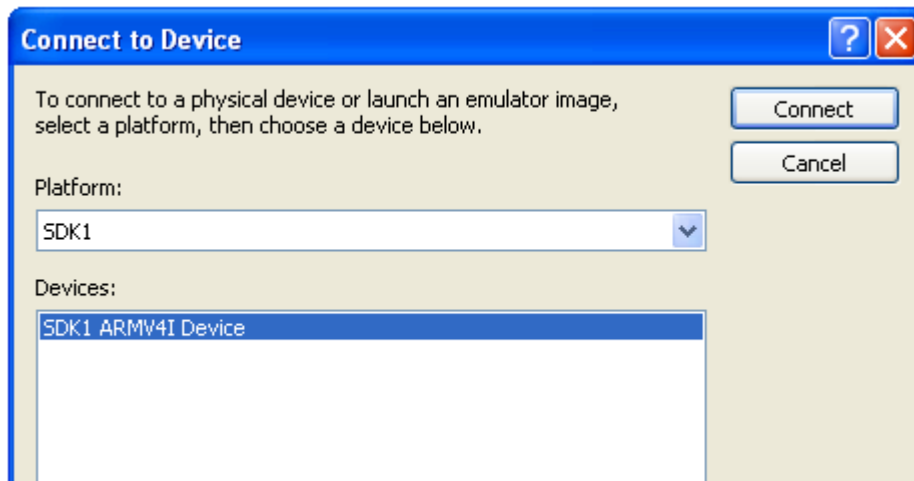

8. Click **OK**.

7.2 Establish the Connection

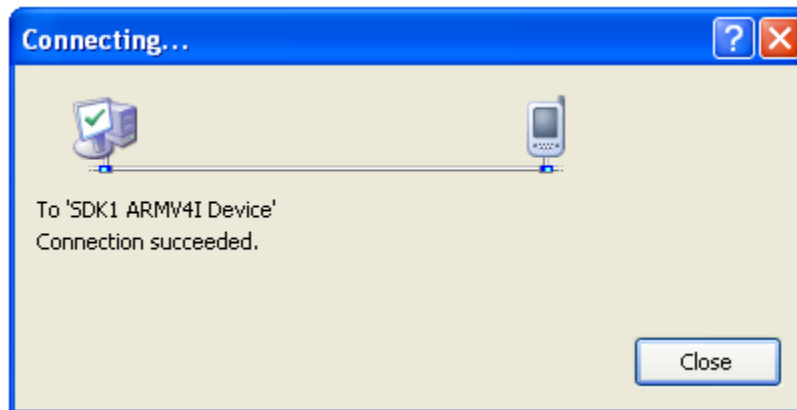1. Back in the VS2005 window, select **Tools > Connect to Device…**

2. Verify SDK1 appears in the *Platform:* drop-down menu and click **Connect**.

3. A *Connecting...* window will open. When `Connection succeeded` appears, click **Close**.

4. At this time, you can set break points, and run and debug your application.

8   Known Issues

8.1   OMAP35x_WinCE600_2_1_0_Source

Issue: When attempting to run an application the following error appears after connecting.

```
1>------ Build started: Project: helloworld2, Configuration: Debug SDK2
(ARMV4I) ------

1>Compiling...

1>stdafx.cpp

1>C:\Program Files\Windows CE
Tools\wce600\SDK2\include\ARMV4I\shlobj.h(120) : fatal error C1083:
Cannot open include file: 'winnetwk.h': No such file or directory

1>Build log was saved at "file://c:\Documents and Settings\<user>\My
Documents\Visual Studio
```

Solution: Copy the *winnetwk.h* file attached to this application note to *C:\Program Files\Windows CE Tools\wce600\SDK1\include\ARMV4I* **NOTE:** Your SDK path may differ from the one provided.

9   Summary

This application note provided instructions to setup a CoreCon communication link between the OMAP35x Development Kit and your Visual Studio 2005 development system. This connection can be used to debug and test hardware and application software. The CE 6.0 BSP and SDK application libraries and header files (*SDK1.msi*) were customized to add CoreCon component files. The *SDK1.msi* file was then used to develop a HelloWorld application. Finally, the OMAP35x Development Kit and the VS2005 were configured for communication and the link was established.

9.1   Additional Reading

  ■ [Windows Embedded Blog](#)[3]

  ■ [Connecting CE 6.0 Device to VS2005 Development Station](#)[4]

---

[3] [blogs.msdn.com/mikehall/archive/2006/01/03/509088.aspx](#)
[4] [http://read.pudn.com/downloads160/ebook/721807/Connect_CE60_to_VS2005.pdf](#)