PM9263

Getting Started with Windows CE 6.0

RONETIX

embedded development tools

February, 2009
Ronetix has made every attempt to ensure that the information in this document is accurate and complete. However, Ronetix assumes no responsibility for any errors, omissions, or for any consequences resulting from the use of the information included herein or the equipment it accompanies. Ronetix reserves the right to make changes in its products and specifications at any time without notice.
Any software described in this document is furnished under a license or non-disclosure agreement. It is against the law to copy this software on magnetic tape, disk, or other medium for any purpose other than the licensee's personal use.

Ronetix Development Tools GmbH
Waidhausenstrasse 13/5
1140 Vienna
Austria

Tel:	+43-720-500315
	+43-1 956 3138
Fax:	+43-1- 8174 955 3464
Internet:	www.ronetix.at
E-Mail	info@ronetix.at

Acknowledgments:

ARM, ARM7, ARM9, ARM11 are trademarks of ARM Ltd.
Windows, Win32, Windows CE are trademarks of Microsoft Corporation.
Ethernet is a trademark of XEROX.
All other trademarks are trademarks of their respective companies.

1. Installation of development tools

- Install Microsoft Visual Studio 2005
- Install Windows CE 6.0 with ARM4I support
- Install Microsoft Visual Studio 2005 Team Suite Service Pack 1
- Install Windows Embedded CE 6.0 Platform Builder Service Pack 1
- Install Windows Embedded CE 6.0 R2

There are two possibilities to prepare the BSP. The first is to download and install AT91SAM9263EK BSP in source from:

http://www.at91.com/windows4sam

The second way is to download the BSP in binary form from:

http://download.ronetix.info/sk-eb926x/wince/PM9263_binary_BSP.zip

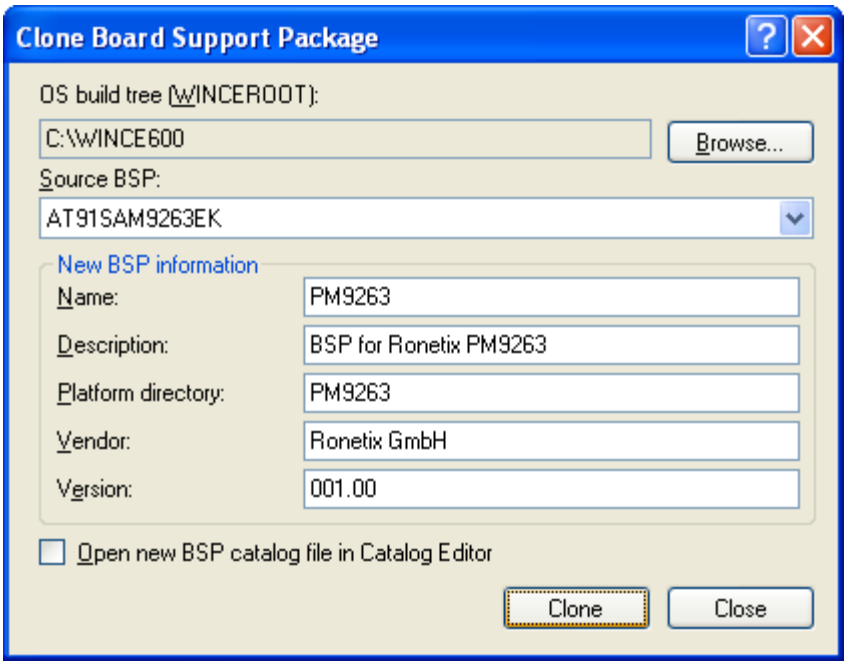
If you use the binary BSP, then skip **"2. Clone a BSP"**.

Note: *With BSP in source form you must accept vendor license. Binary BSP is a compiled source BSP. There are no differences which of them you are using when creating applications.*

2. Clone a BSP

- Run Visual Studio 2005
- Clone AT91SAM9263EK BSP. (For source form BSP)

- Click on `Tools/Platform Builder for CE 6.0/Clone BSP`



o For `Source BSP`: from the drop down menu select `AT91SAM9263EK`

For `New BSP information:`

Name: **PM9263**

Description: **BSP for Ronetix PM9263**

Platform directory: **PM9263**

Vendor: **Ronetix GmbH**

Version: **001.00**

o Press `Clone` button.

- Close Visual Studio 2005.
- In C:\WINCE600*PLATFORM\PM9263* directory rename the name of these files as follow:

`at91sam9263ek.bat`	→ `pm9263.bat`,
`AT91SAM9263EK_BIN.bat`	→ `pm9263_BIN.bat`,
`AT91SAM9263EK_BIN.pbcxml.bin`	→ `pm9263_bin.pbcxml.bin`,
`Generate_AT91SAM9263EK_BIN.bat`	→
`Generate_pm9263_BIN.bat`	

- In *`Generate_PM9263_BIN.bat`* file change

  SET BSP_NAME=AT91SAM9263EK

to

SET BSP_NAME=PM9263

- Edit `PM9263_BIN.pbcxml.bin` file with WordPad, and replace **AT91SAM9263EK** string in tags (not in file names) with **PM9263**.

- Close Visual Studio 2005.

- Download the PM9263 patch and the `patch.exe` utility form Ronetix web page. (Necessary for the Source BSP) and save it into the C:\WINCE600\PLATFORM directory. The `patch.exe` utility can be also downloaded from http://gnuwin32.sourceforge.net
  http://download.ronetix.info/sk-eb926x/wince/ronetix-pm9263-wince.patch

- Apply the patch:
  Open a MS-DOS prompt: on your computer press `Start\Run` and type cmd
  On the open command prompt, type:

```
cd c:\wince600\platform\pm9263
..\patch.exe -p1 < ..\ronetix-pm9263-wince.patch
```

the result is:

```
C:\WINCE600\PLATFORM\PM9263>..\patch -p1 < ..\ronetix-pm9263-wince.patch
patching file SRC/DRIVERS/Emacb/dp83848.c
patching file SRC/DRIVERS/Emacb/Emac.c
patching file SRC/DRIVERS/Emacb/PhyInterface.h
patching file SRC/DRIVERS/Emacb/sources
patching file SRC/DRIVERS/NandFlash/NandFlash.c
patching file SRC/DRIVERS/SDHC/at91sam9263ek_slot.cpp
patching file SRC/DRIVERS/SDMEMORY/driver/sdmemory.c
patching file SRC/DRIVERS/SDMEMORY/Loader/sdmem_loader.c
patching file SRC/DRIVERS/SPI/SPI.c
patching file SRC/DRIVERS/usbhdc/at91sam9263ek_ohcd.c
patching file SRC/OAL/OALLIB/init.c
```

- Run Visual Studio 2005.

3. Building the OS with PM9263 BSP

There are two possibilities to build the OS:

- using a prepared by Ronetix project
- creating a new project from scratch

The results of the building are two binary files which should be programmed into the NAND Flash:

`C:\WINCE600\OSDesigns\pm9263\pm9263\RelDir\PM9263_ARMV4I_Release\EBOOT.nb0`

and

`C:\WINCE600\OSDesigns\pm9263\pm9263\RelDir\PM9263_ARMV4I_Release\NK.nb0`

The EBOOT.nb0 is a bootloader; the NK.nb0 is the WinCE kernel.

3.1. Using a prepared project

Extract one of the ZIP files in C:\WINCE600\OSDesigns directory.

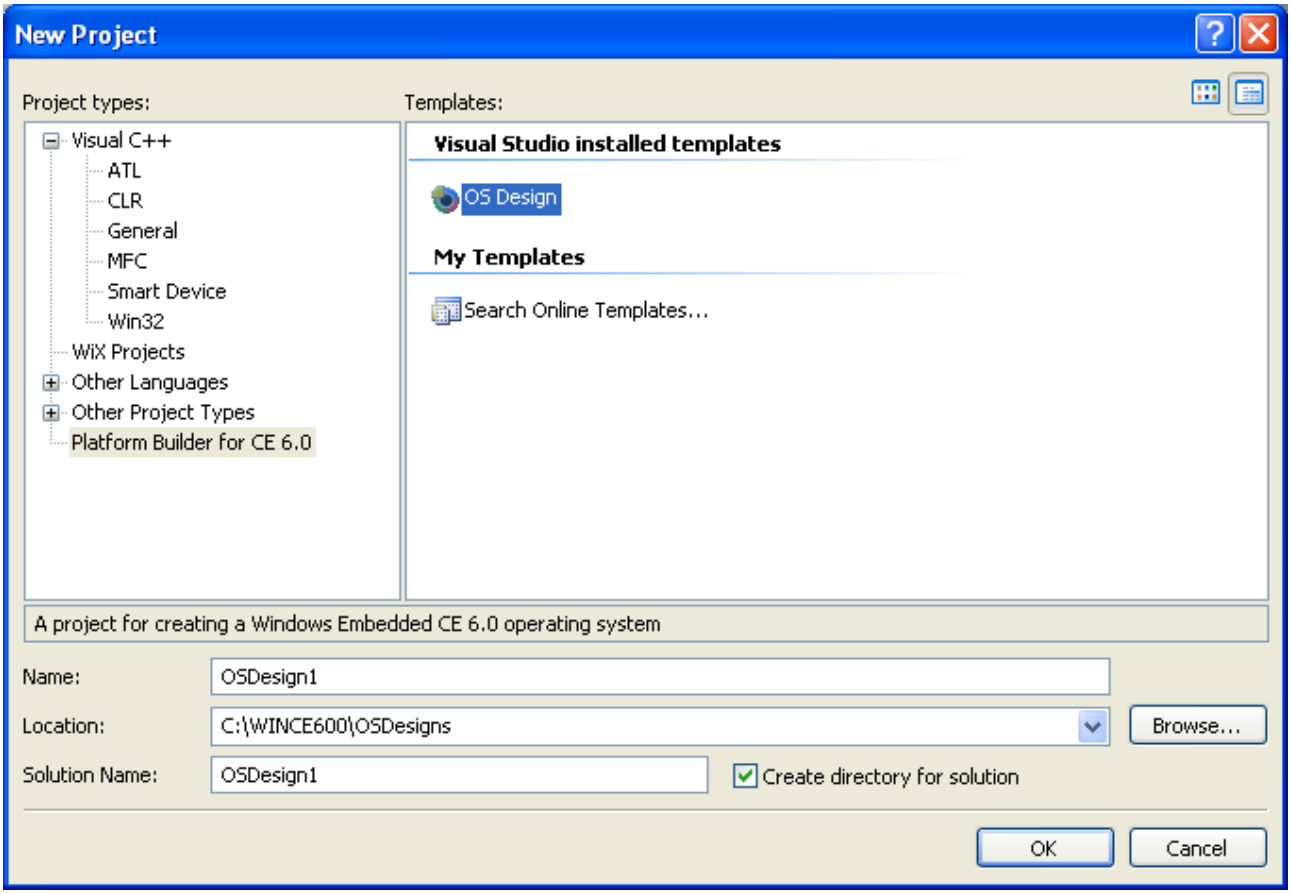
http://download.ronetix.info/sk-eb926x/wince/pm9263_example_binBSP.zip - if binary BSP is used

http://download.ronetix.info/sk-eb926x/wince/pm9263_example_softBSP.zip - if source BSP is used

- Click on `File/Open Project/Solution`
- Click on `PM9263_example`
- Click on `Build/Build Solution`

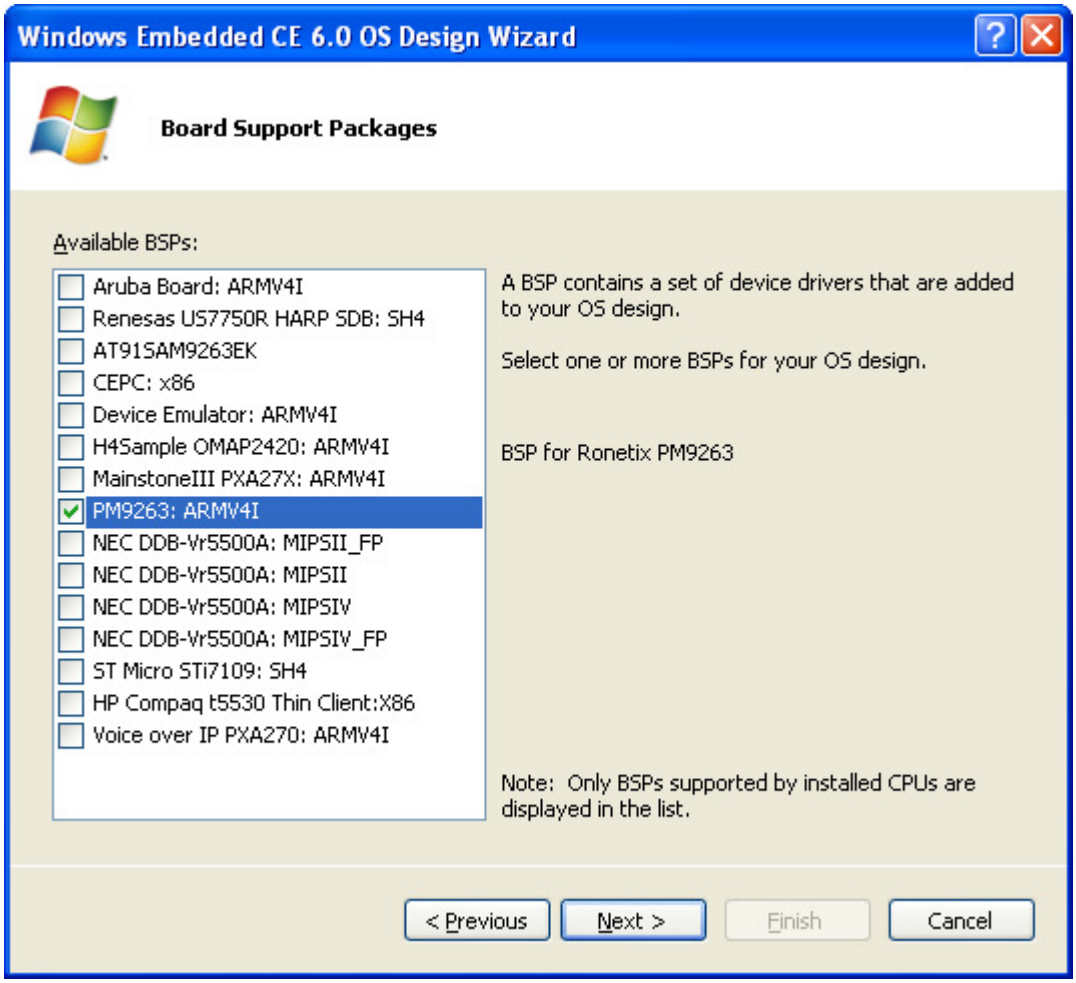
3.2. Creating a new project

- Click on `File/New/Project`, choose `Platform Builder for CE6.0` and then press *OK*.



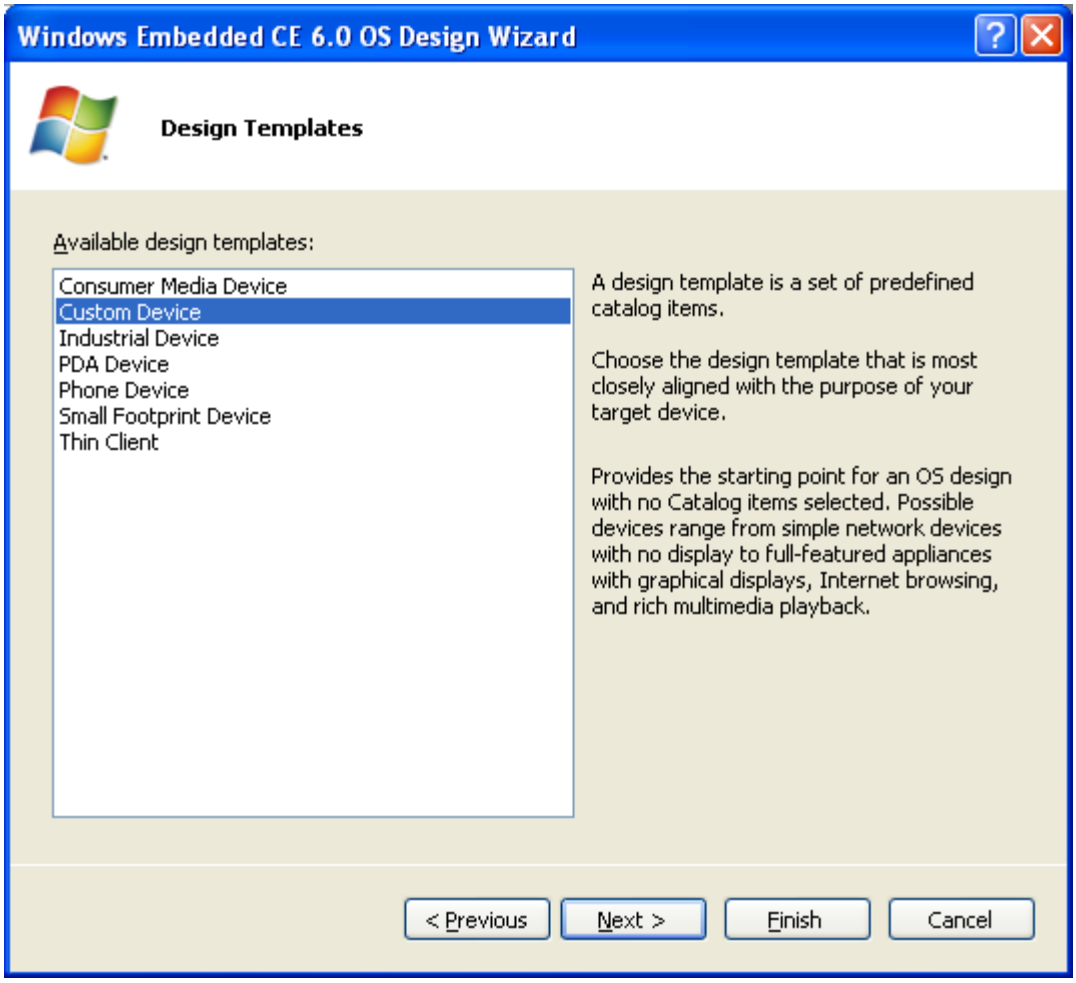
A window appears with available BSP packages, you choose `PM9263: ARMV4I` (This is Source Form BSP) or `PM9263_BIN: ARMV4I` (This is Binary Form BSP).

Note: *Both of them could be selected. Later you must choose which configuration you would like to build.*



Press `Next`.

A `Design Template` appears. You may choose whatever you like, but for the example Choose `Custom Device`.

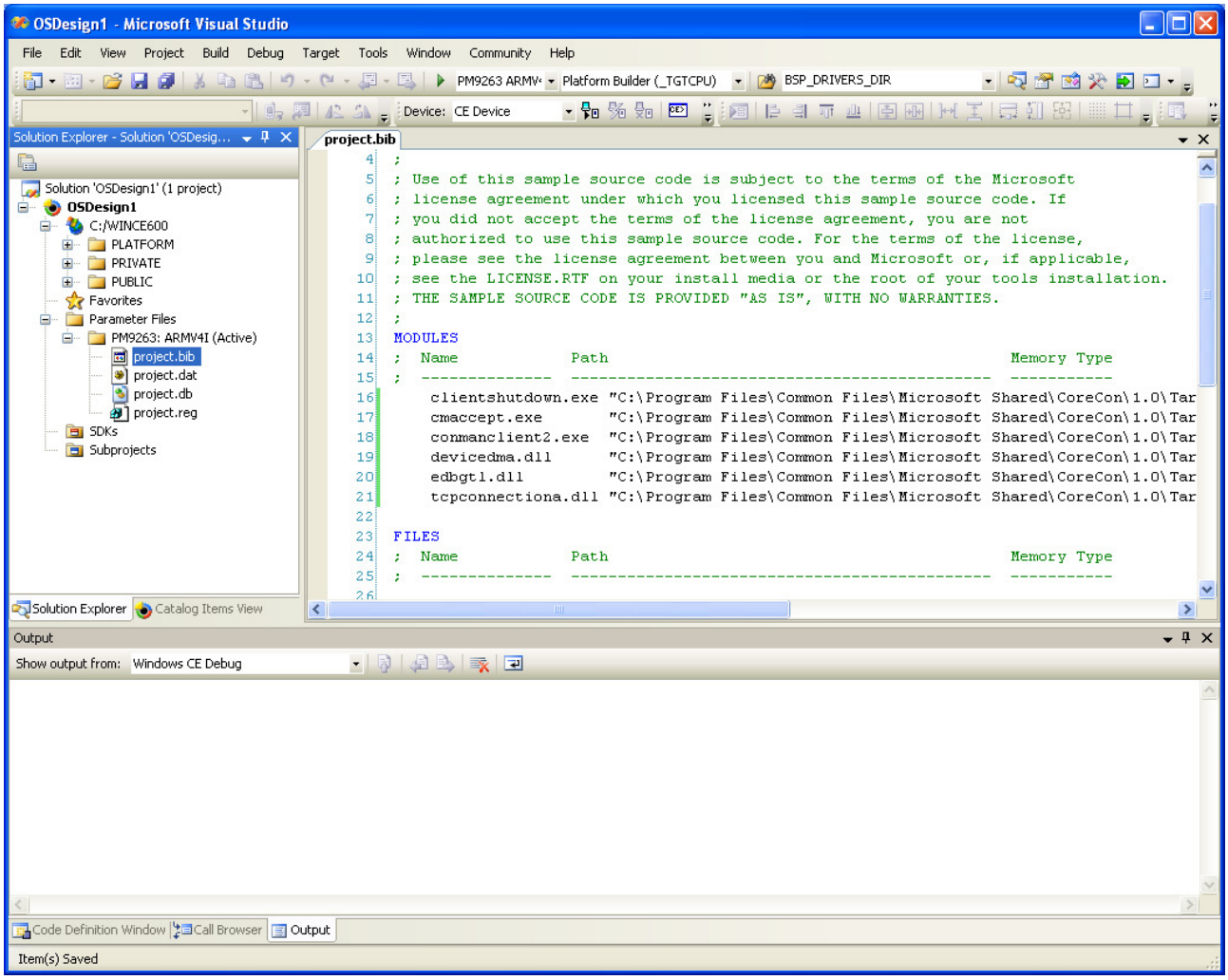


Click on `Next` and again and again …, and `Finish`.

- From `Solution Explorer\<OS Design name>\Parameter Files\ <Active configuration>`

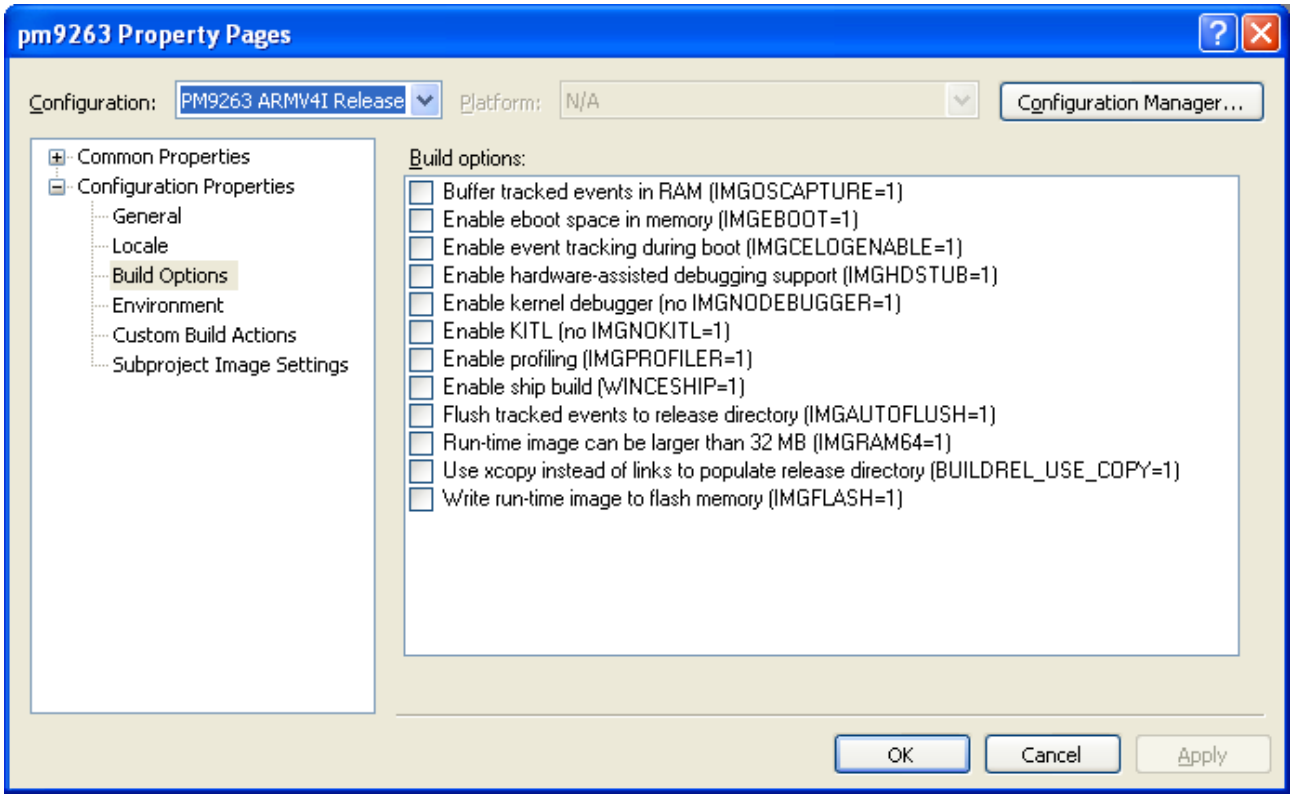
  double click on `project.bib` file. In `MODULES` section of the opened `project.bib` file add the following 6 lines.

```
clientshutdown.exe "C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i\clientshutdown.exe"  NK RS
cmaccept.exe  "C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i\cmaccept.exe"  NK RS
conmanclient2.exe  "C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i\conmanclient2.exe"  NK RS
devicedma.dll  "C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i\devicedma.dll"  NK RS
edbgtl.dll  "C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i\edbgtl.dll"  NK RS
tcpconnectiona.dll "C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i\tcpconnectiona.dll"  NK RS
```

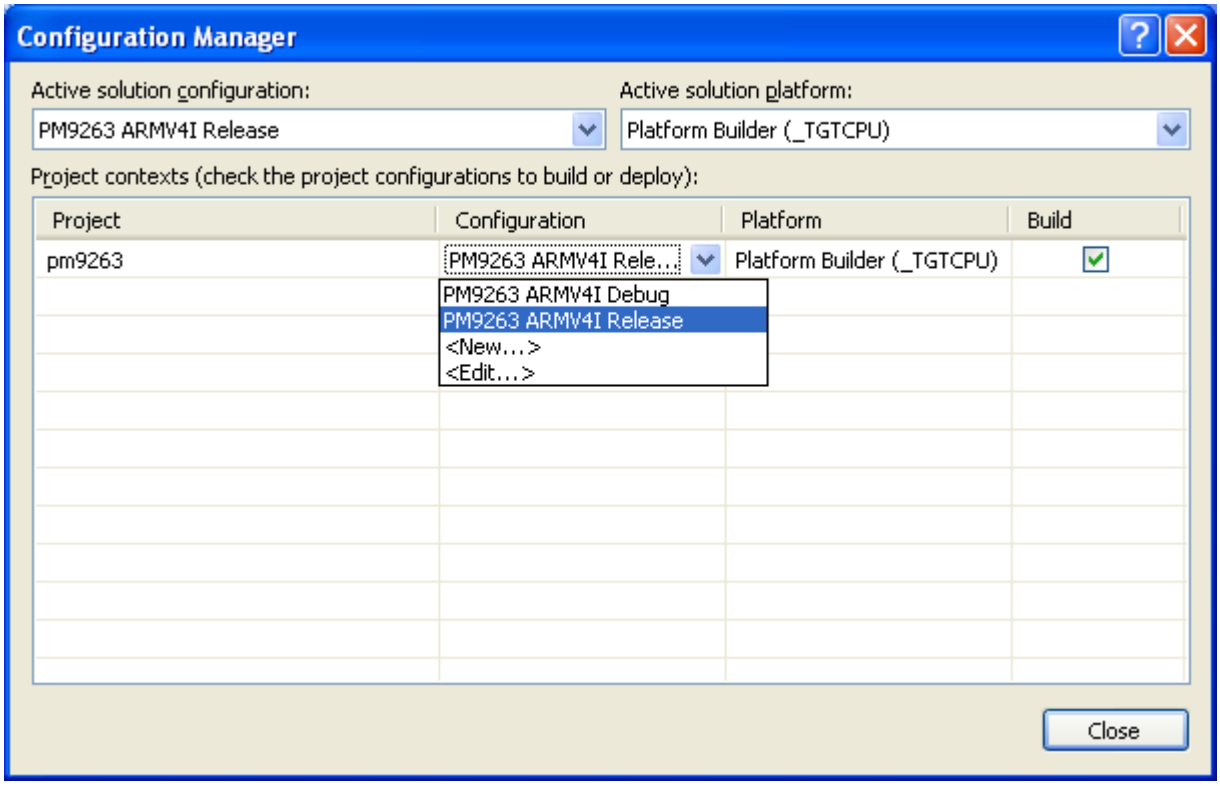


- Save All Files. (`File\Save All`)

- In `Catalog Items View`, You may choose OS features, drivers, BSP features that you need.



- Disable Debug options of kernel.
- Click on `Project\pm9263` properties
- Expand `Configuration properties.`
- From `Configuration` choose `PM9263 ARMV4I Release`
- Make sure no option is checked.
- Click on `Configuration Manager` Button.
- Click on `Configuration` and choose `PM9263 ARMV4I Release.`
- Then `Close` button.
- And then `OK` button.


Configuration Manager
Active solution configuration:
PM9263 ARMV4I Release
Active solution platform:
Platform Builder (_TGTCPU)
Project contexts (check the project configurations to build or deploy):
Project
Configuration
Platform
Build
pm9263
PM9263 ARMV4I Rele...
Platform Builder (_TGTCPU)
PM9263 ARMV4I Debug
PM9263 ARMV4I Release
<New...>
<Edit...>
Close

4. Programming of all images

Three binary images are necessary for a working Windows CE system:

- **first stage bootloader** for the NOR Flash which should be programmed at address 0x00000000
  The NOR bootloader can be downloaded here:

  http://download.ronetix.info/sk-eb926x/wince/nor_loader.bin

- **second stage bootloader** for the NAND Flash – EBOOT.nb0 which should be programmed into the NAND Flash at address 0x00020000
  The EBOOT.nb0 can be found on the Ronetix's website or

  C:\WINCE600\OSDesigns\pm9263\pm9263\RelDir\PM9263_ARMV4I_Release

- **Windows CE kernel** for the NAND Flash – NK.nb0 which should be programmed into the NAND Flash at address 0x00080000
  The NK.nb0 can be found on the Ronetix's website or

  C:\WINCE600\OSDesigns\pm9263\pm9263\RelDir\PM9263_ARMV4I_Release

There are two possibilities to program the binary images into the PM9263 board:

- Using PEEDI JTAG Emulator and Flash Programmer (http://www.ronetix.at/peedi.html)
- Using SAM-BA package from Atmel's website

4.1. Programming using PEEDI

Make sure the paths in the PEEDI's configuration file are correct and open a serial or telnet console:

```
pm9263> run $wince
```

4.2. Programming using SAM-BA

- Install the SAM-BA package from Atmel's website and copy the SAM-BA applet isp-nandflash-at91sam9263.bin in the folder `\lib\AT91SAM9263-EK`.

  http://download.ronetix.info/sk-eb926x/wince/isp-nandflash-at91sam9263.bin

- Open BMS (boot mode select) jumper (J21 on the base board) and connect the Evaluation Board to the PC via a USB cable.
- Power on the board, start the SAM-BA and select the AT91SAM9263-EK board in the SAM-BA menu
- In the NOR tab, execute `enableNOR` and then send the NOR bootloader at address 0x0

  http://download.ronetix.info/sk-eb926x/wince/nor_loader.bin

- In the NAND tab, execute `enableNAND`, select `EBOOT.nb0` file and flash it at address 0x20000.
- Select `NK.nb0` file and flash it at address 0x80000
- close the BMS (J21) jumper
- close SAM-BA
- remove the USB cable
- Power off the board

4.3. Setting EBOOT

- Connect the RS232-DEBUG port from the EB9263 to the PC using a serial cable and start a terminal program (115200, 8, N, 1, no flow control).
- Power on the board and enter in the EBOOT menu by pressing [SPACE]:

```
NOR WinCE BOOTLOADER v1.0 - EBOOT on NF(0x20000)
 WARNING : LoadEBootCFG: No valid Eboot configuration found.
 INFO : Loading default bootloader settings

 Press [ENTER] to download now or [SPACE] to cancel.
```

Select option 9 and change to:

```
    9) Launch existing flash resident image at startup
```

Select option n ("Nand Flash Menu") and then sub-option 2

("Enter manually the image parameters"):

```
            Physical Start Address : (0x8006c000) : 0x8006c000
            Starting ip : (0x8006d000) : 0x8006d000
            Total ROM size : (0x16174f0) : 0x192b710
```

Note: *`Physical Start Address`, `Starting ip`, `Total ROM size` can be viewed from build output windows or from `C:\WINCE600\Build.log` after successful building. The values above are an example*

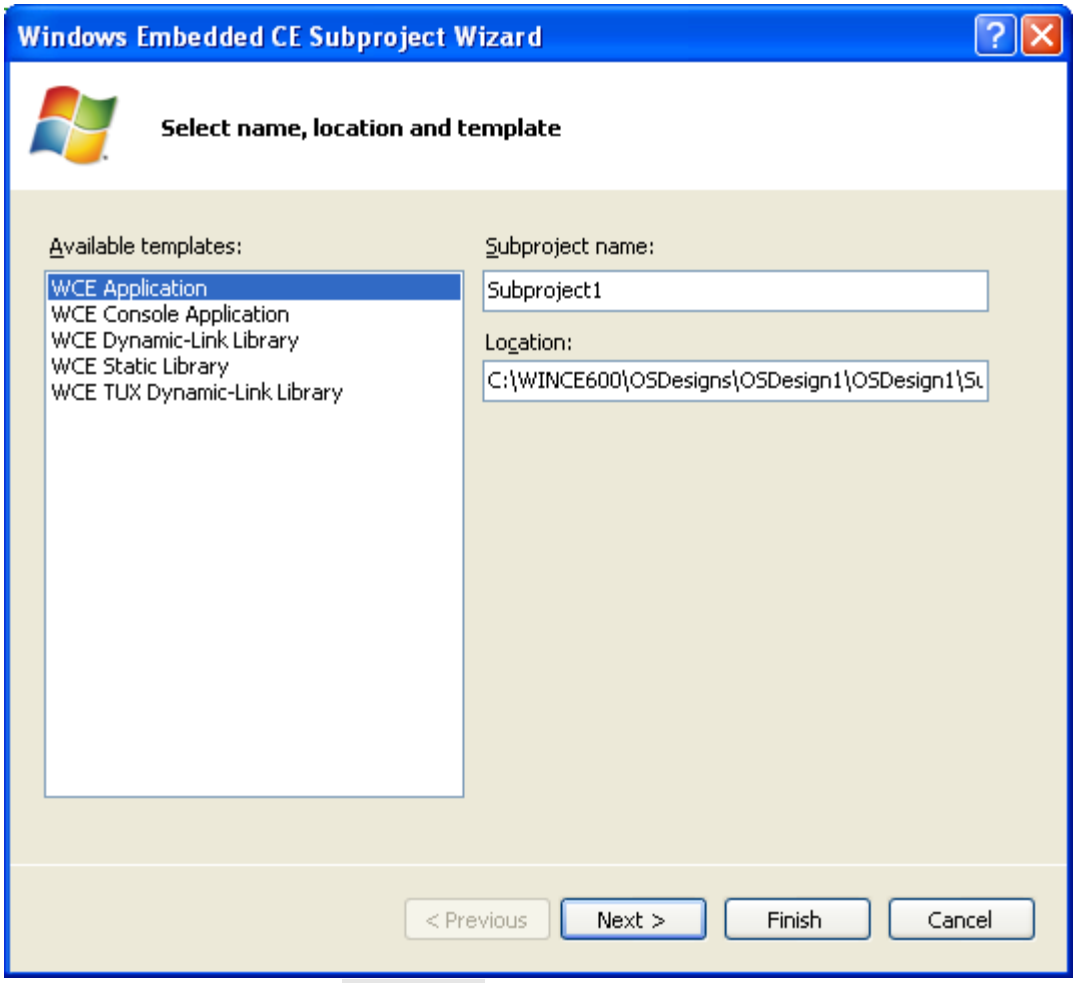
- `Select sub-option 3 ("Quit") and then s ("Save configuration now")`
- `Reset the board.`

5. Creating user application

There are two types of application: managed and native. Managed applications could be made only on separate project (Managed project that uses ATL, dotNET or MFC), SDK is also needed. Native application could be build in separate project or in OSDesign subproject. When in separate project a SDK is required, when as subproject in OSDesign no SDK is needed.

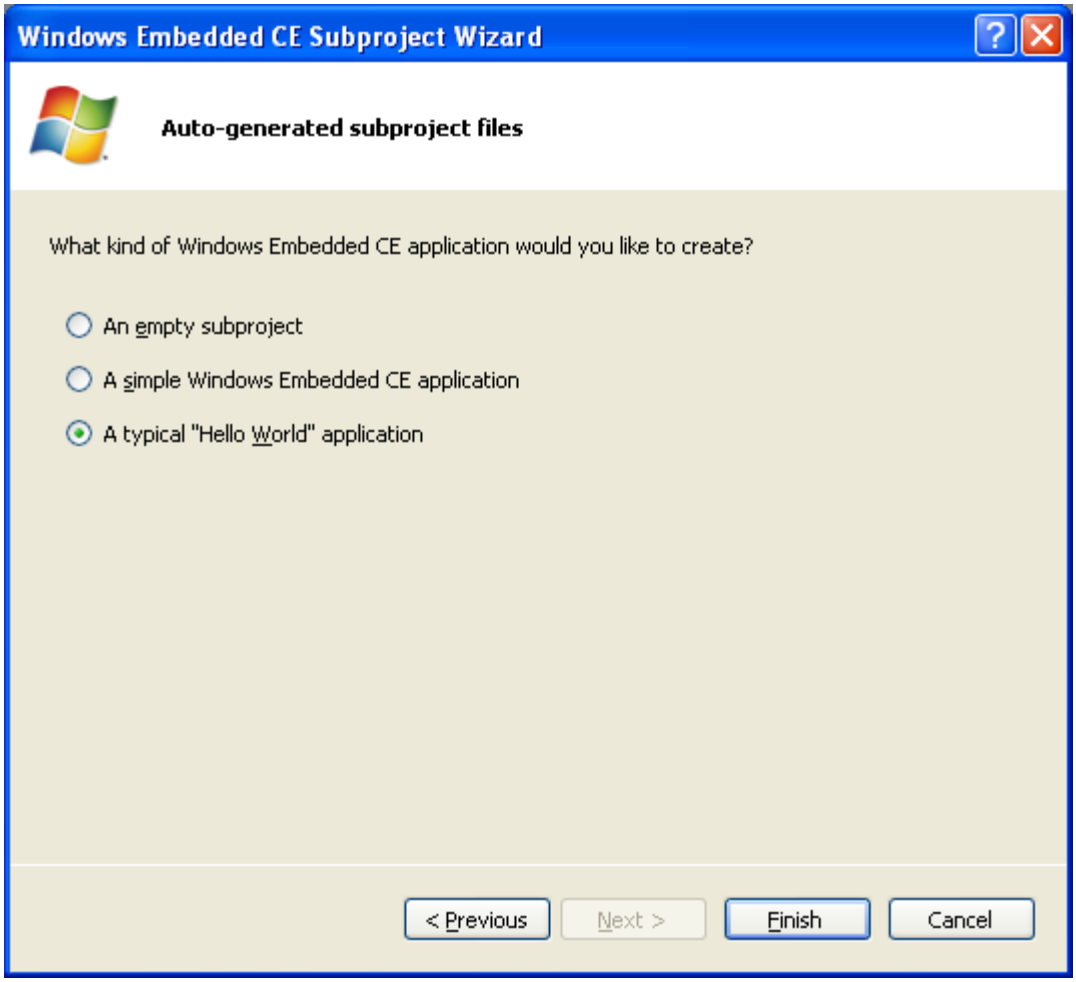
5.1. Design application in a subproject of the OS design project.

- Add new subproject form `Project\Add new subproject`



Enter `Subproject name:` Subproject1

Press `Next`.



- Choose `A typical "Hello World" application`, then press `Finish`.
- Press `F7` to build the solution.
- Program the device - see Programming of all images
-

Note: *When you build application as a subproject, it will be included in the ROM image file NK.nb0*

In the device after loading Windows CE, press `Start\Run...`

Then type cmd and press `Enter`.

On the open command prompt, type subproject1 and press `Enter`.

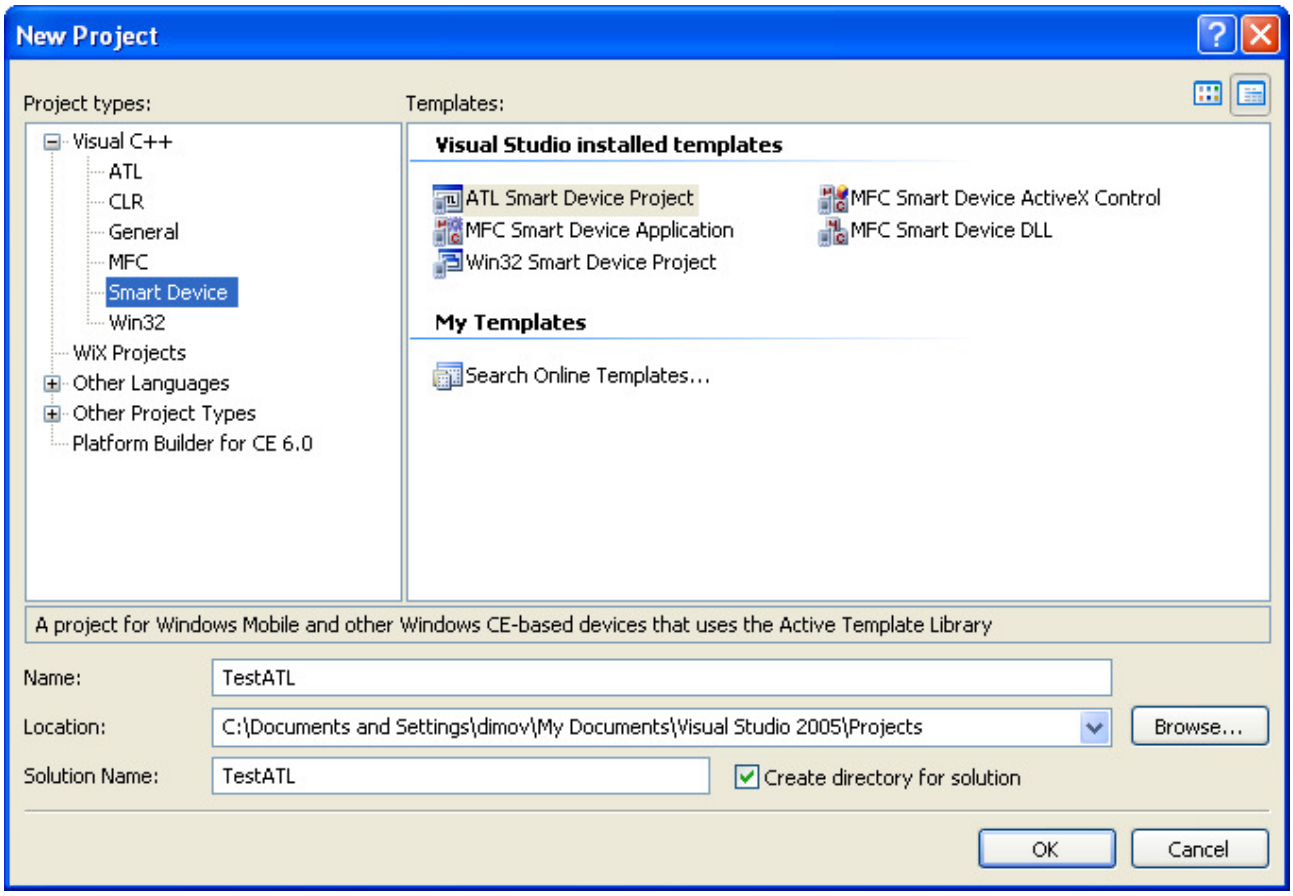
Now you should see the result.

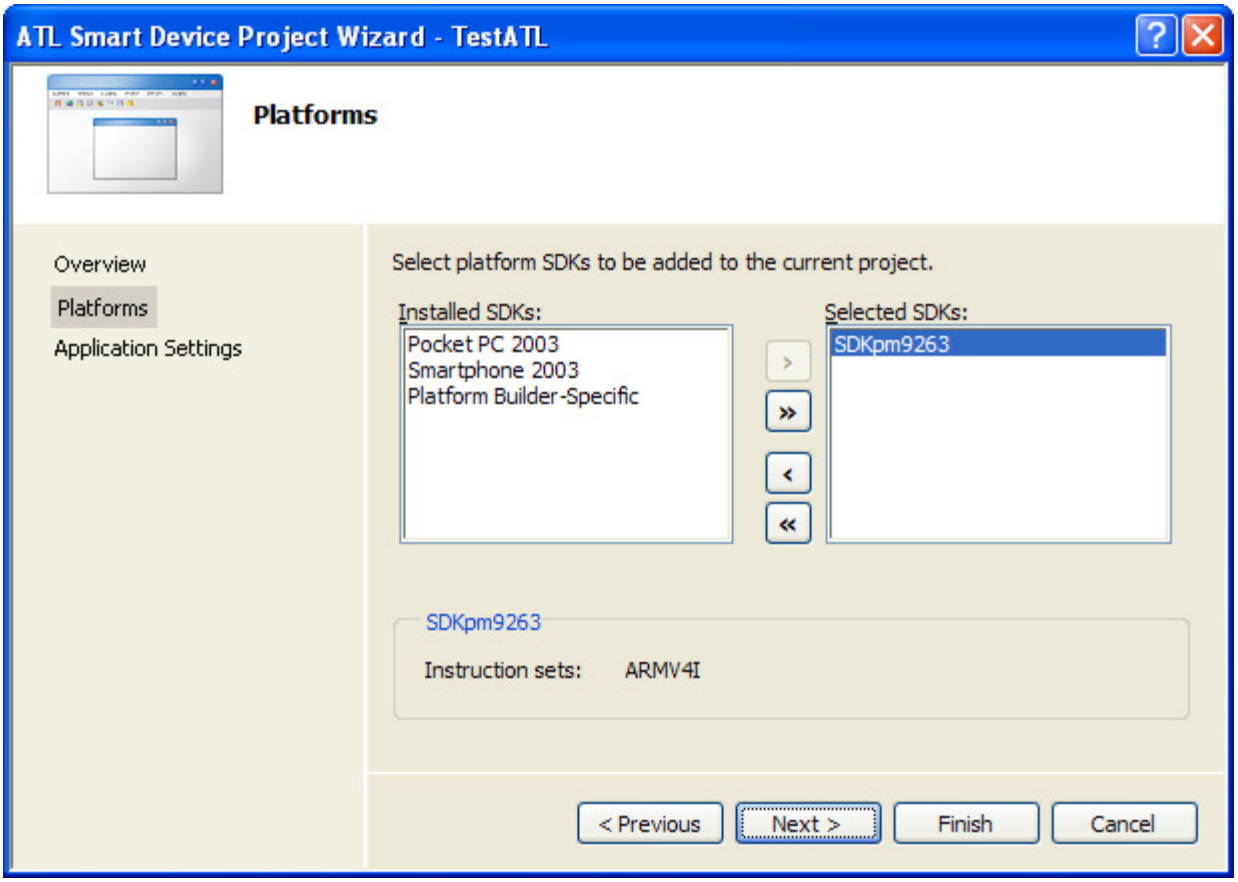
5.2. Application in a managed project

- You have to create a ATL, dotNET or MFC project.

  Click on `File\New\Project...`

  Choose `Smart Device\ATL Smart Device Project`





Then Choose the SDK for the required platform (a device you are writing for).

Note: *In the device should have the corresponding platform framework (ATL, dotNET or others). Currently Windows CE 6.0 comes with dotNET Compact Framework 2.0 and Active Template Library (ATL).*

- After creating the project, open `TestATL.cpp` (in this example that is the file), find this function

```
extern "C" int WINAPI _tWinMain(HINSTANCE /*hInstance*/, HINSTANCE /*hPrevInstance*/,
                                LPTSTR /*lpCmdLine*/, int nShowCmd){
```

and in its body type

```
    printf("Hi there\n");
```

before the `return` statement.

- Connect the device see Uploading and debugging user applications.
- Press `F7` to build the solution.
- Press `F5` on the keyboard to start debugging and deploying application to the device.
- You will see the message on that command prompt where you have started the `conmanclient2.exe`.

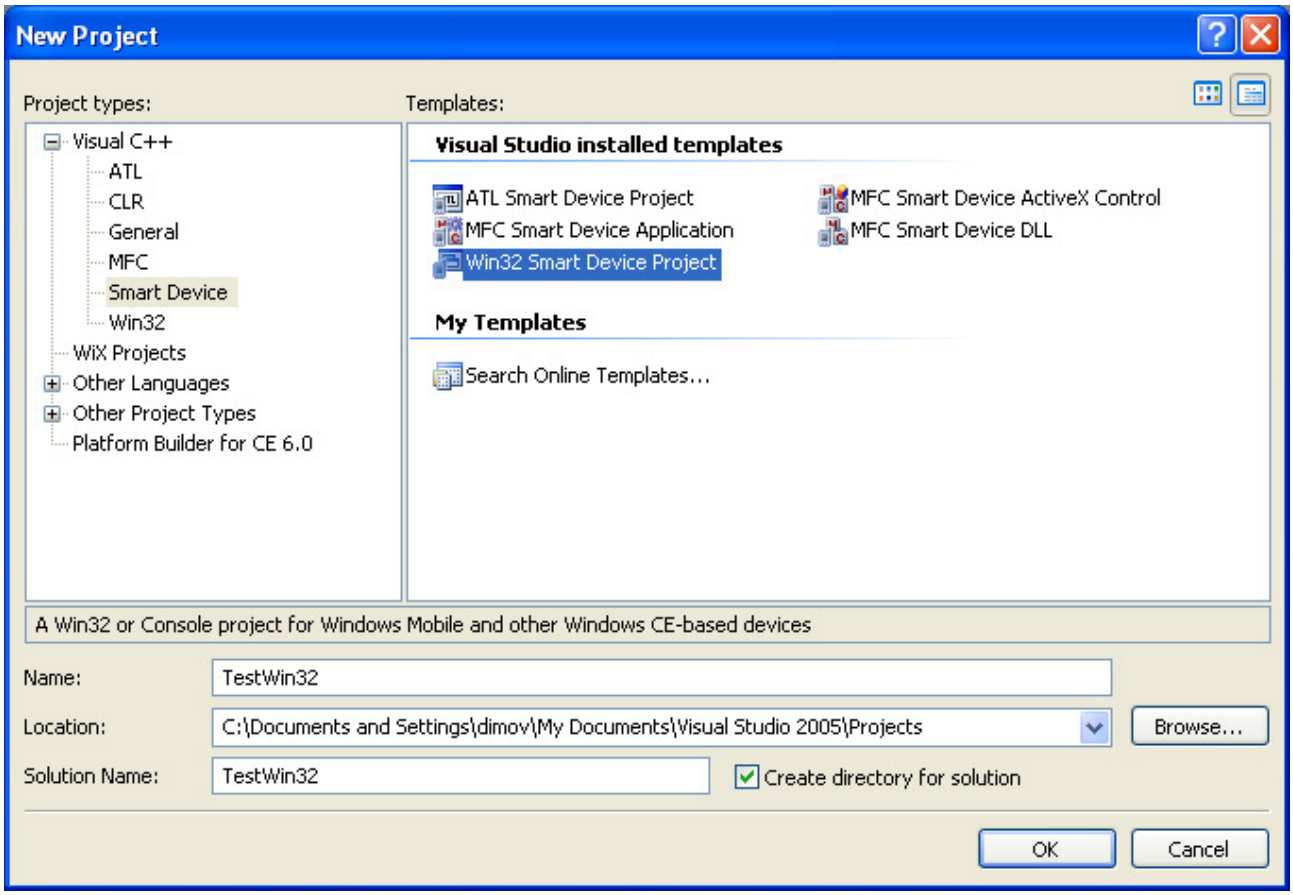
5.3. Native application

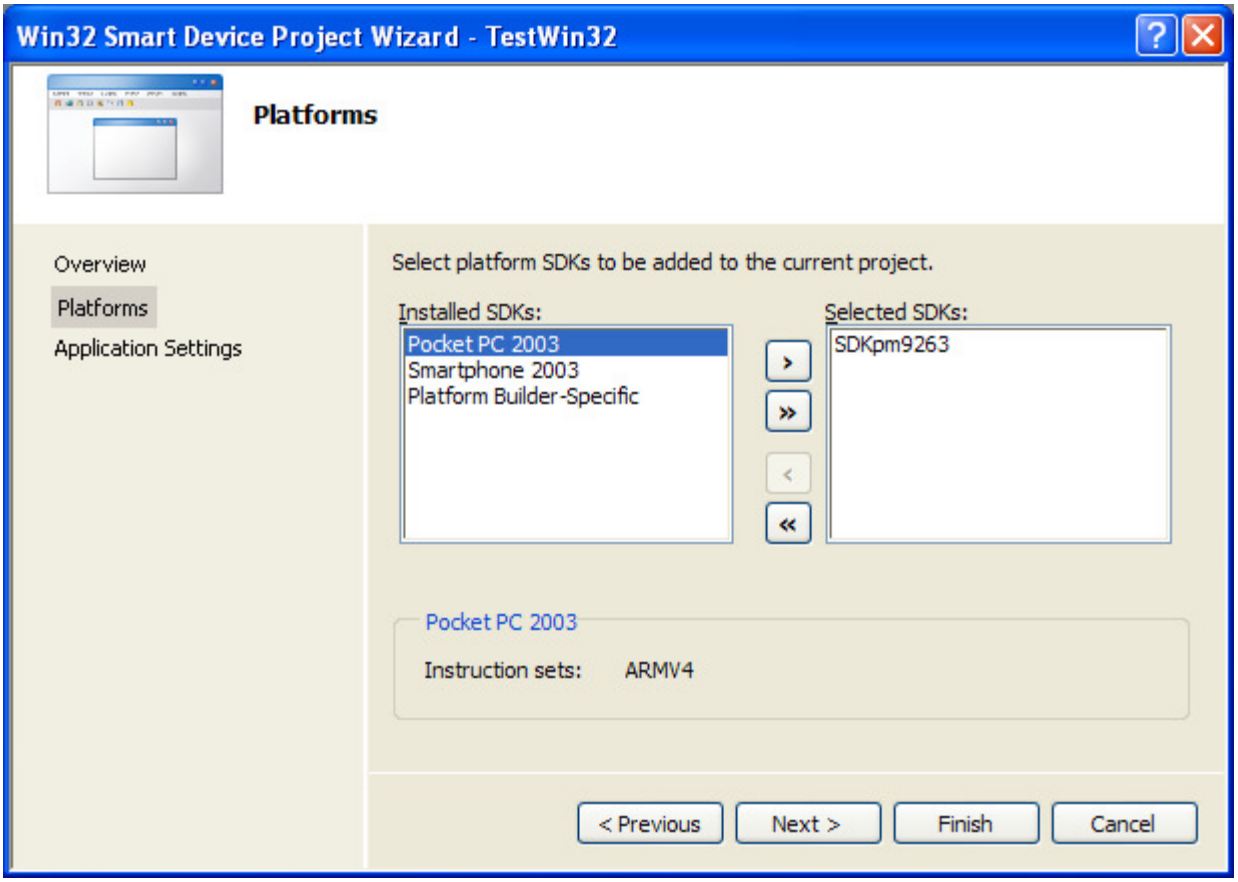
Make a Smart Device project and choose the SDK for the device you want to write for.

Click on `File\New\Project...`

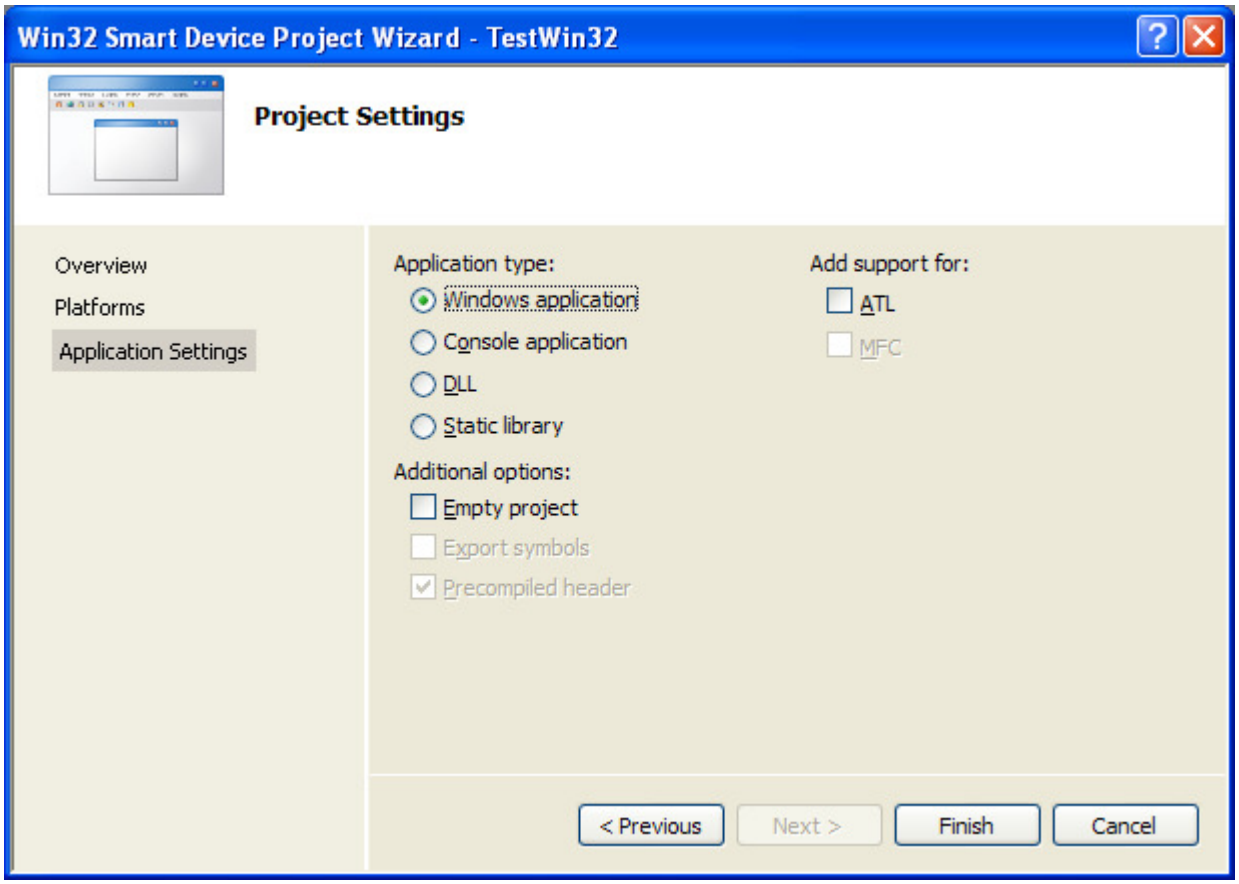
Choose `Smart Device\Win32 Smart Device Project`

Enter name of the project TestWin32





Choose the SDK for the platform you are building.



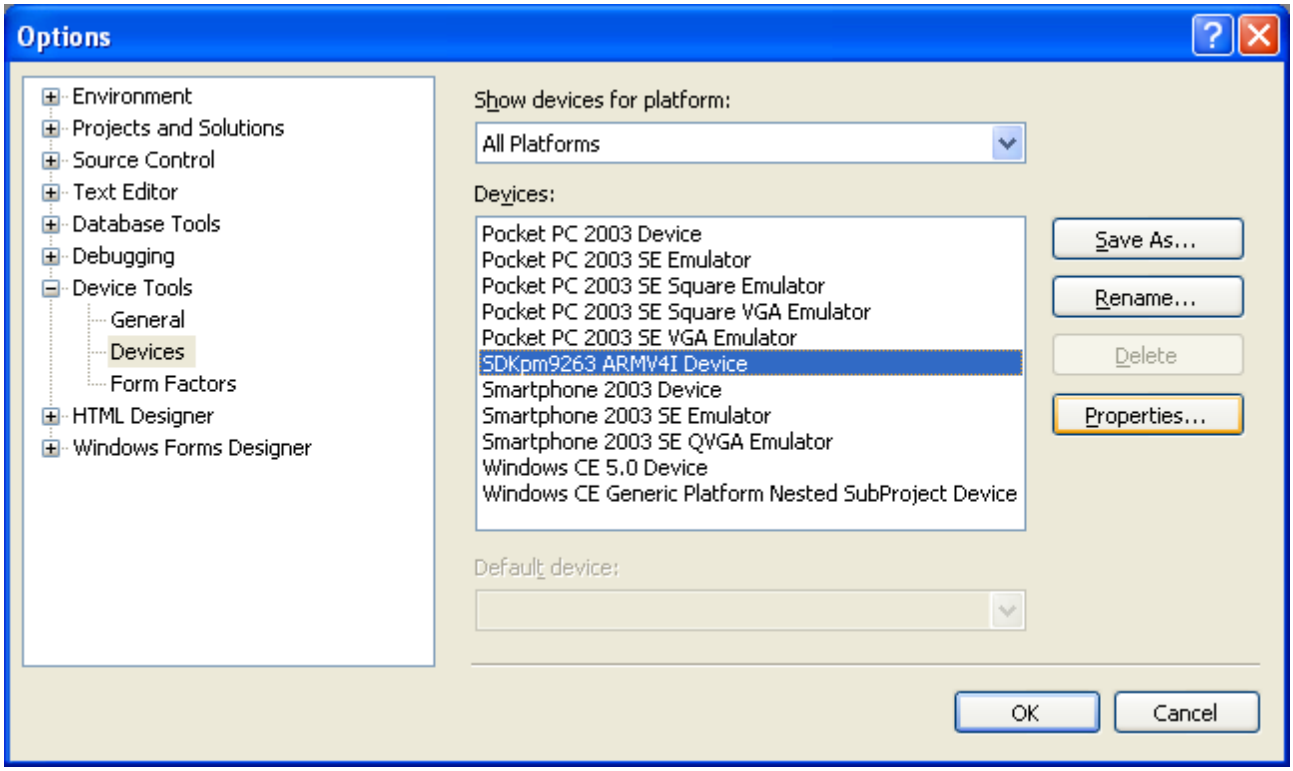
Keep it as it is (see the picture just above), and press `Finish`.

- Connect to the device; see Uploading and debugging user applications.
- Press `F5` to deploy and debug the application.
- And You should see a new window on the screen of the device

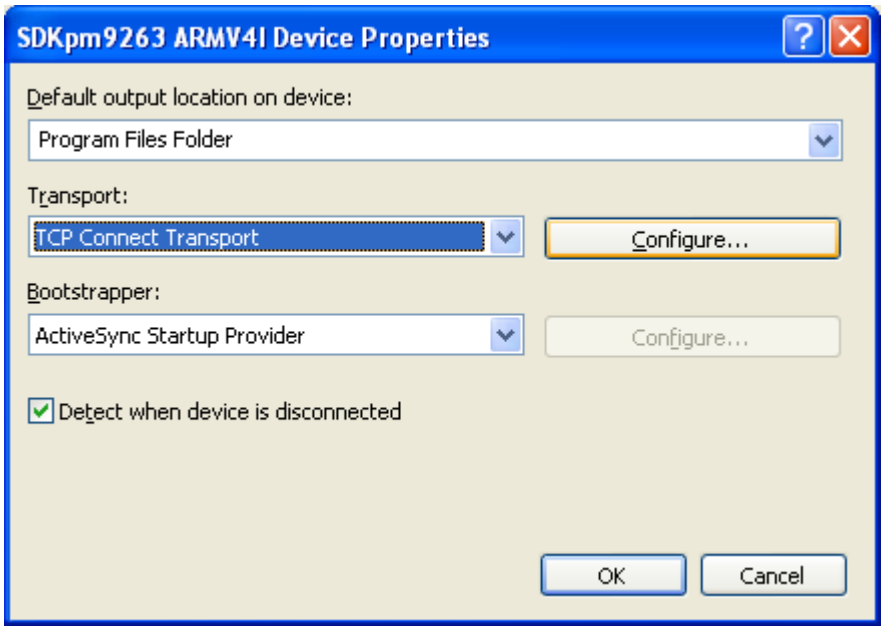
6. Uploading and debugging user applications.

Note: *Applies only to managed and native applications in separate project.*

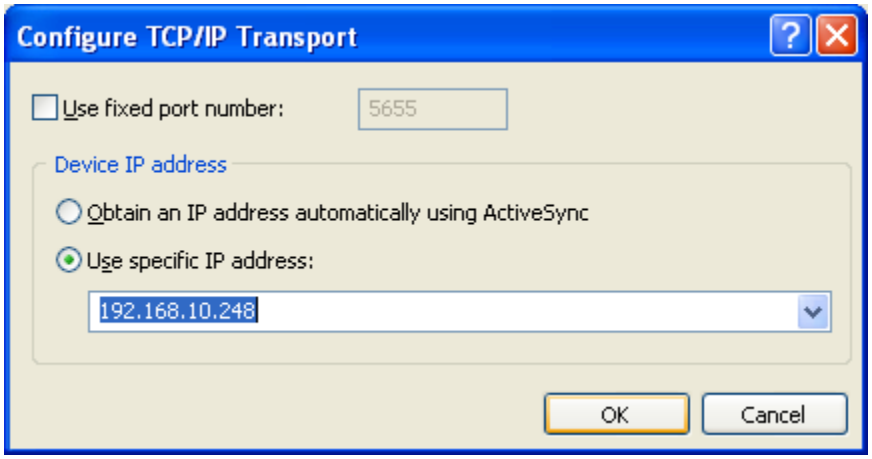
- From VS 2005 menu choose `Tools\Options...`
  and from the opened window click on `Device Tools\Devices`,
  then choose the device for which you have installed the SDK.



Click `Properties`.



Choose `Transport: TCP Connect Transport`, and click `Configure...`

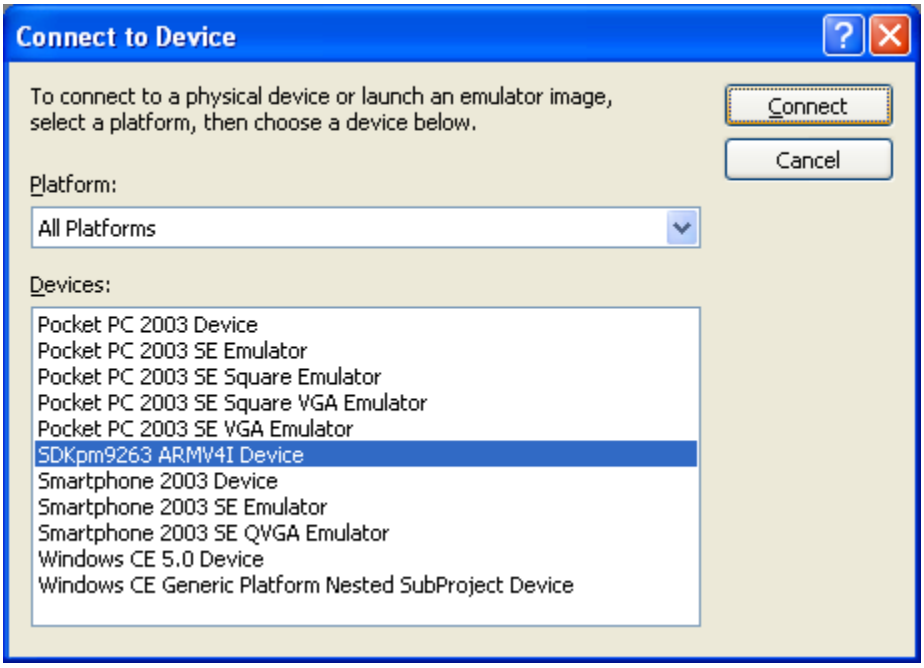


Then you may enter the IP of the device.

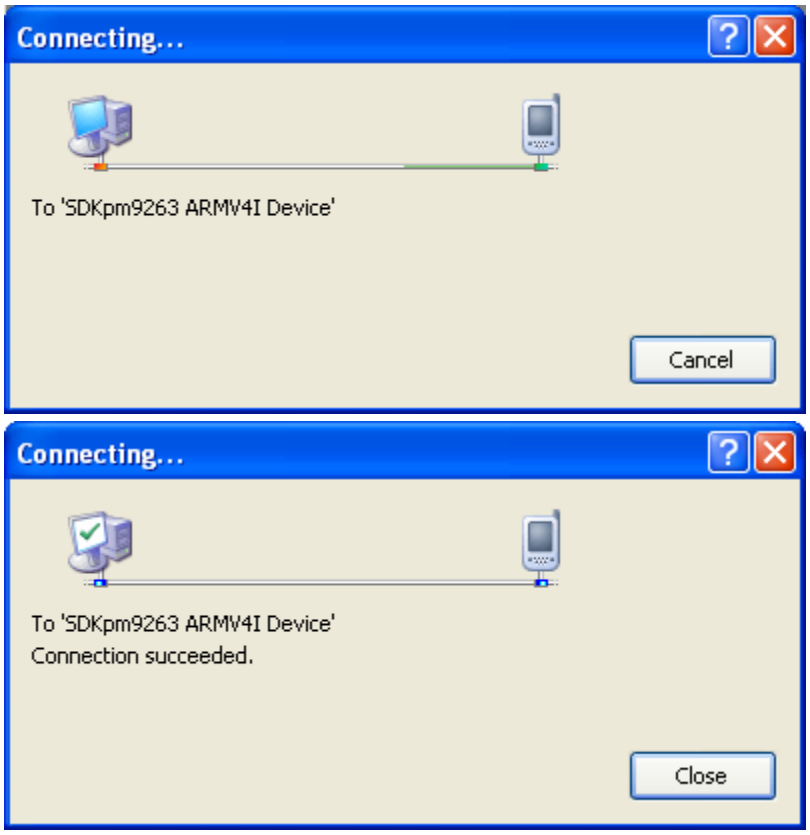
Note: *The IP of the device could be statically set in the Windows CE, or the device could obtain it form DHCP server if available on the network. Entered IP above is just an example(the one I use).After booting Win CE 6.0 you may change it from Start\Settings...\Network and Dial-up Connections then double click on EMACB1, click on Specify an IP address, and fill with appropriate values.*

- The device (the demo board) with Win CE 6.0 will not accept request for debugging connection by default. To enable connection, in the Win CE run command prompt (`Start\Run...` then type cmd and run it). When the command prompt appear execute conmanclient2.exe, the prompt should not appear. Next open another command prompt end execute in it cmaccept.exe, so you just enabled the device to accept connection for around 3 minutes. In this time `cmaccept.exe` will not return to prompt, when `cmaccept.exe` return prompt you can not make connection and have to run only `cmaccet.exe` again to re open the device again. When you make a connection the device stay connected until you close the connection.

- To connect to device choose `Tools\Connect to device`, and from the open window choose your device and click `Connect`



Then window opens, showing the connection status.



Click the `Close` button when connection succeeded, to close the window.

Note: *You may connect and disconnect when you are in a project environment or no project environment.*

You can not connect to device when You are in OS Design project.

- When you write some code and want to test it click `Debug\Start`. Then Visual Studio will send the program to the device and will run it.