Windows CE 6.0
Guide For MV-V210

Micro Vision

Microvision Co., Ltd.

Document Information

Version	1.2
File Name	MV-V210_WindowsCE_Guide.doc
Date	2010. 9. 13
Satus	Working

Revision History

Date	Version	Update Descriptions	Editor
2010. 6. 16.	V1.0	First Edition	Jongill Wee
2010. 6. 24.	V1.1	Second Edition Add 5.3 Application Build 42/51	Jongill Wee
2010. 9. 13	V1.2	Third Edition Additional Description to each picture	Jongill Wee

Published by MicroVision Co.,Ltd.

(☎) +82-2-3283-0101, (✉) sale@microvision.co.kr
http://www.microvision.co.kr, http://www.mvtool.co.kr
Room #610, Hanshin IT Tower 235, Guro3-dong, Guro-gu, Seoul, Korea.

Contents

1. Windows CE 6.0 Installation

The procedure for installing Microsoft Windows CE 6.0 must be done correctly so as to avoid any related malfunctions.

MV-V210 BSP (Board Support Package) requires the QFE version, dated 7/31/09, with R3 for the compilation. The installation steps are listed below with the title for Windows CE 6.0, including a QFE list.

- Step1 Installing Visual Studio 2005
- Step2 Updating Visual Studio 2005 Service Pack 1
- Step3 Installing Windows CE 6.0
- Step4 Updating .NET Compact framework 2.0 Service Pack 1
- Step5 Updating Windows CE 6.0 Service Pack 1
- Step6 Updating Windows CE qfe 2006

  WinCEPB60-061231-Product-Update-Rollup-Armv4I.msi
- Step7 Installing Virtual Machine Network Driver
- Step8 Updating Windows CE 6.0 R2
- Step9 Updating Windows CE qfe 2007, 2008

  WinCEPB60-070824-KB940982-COMMON.msi

  WinCEPB60-071031-KBHELPDOCS-COMMON_ENU.msi

  WinCEPB60-071231-Product-Update-Rollup-Armv4I.msi

  WinCEPB60-081231-Product-Update-Rollup-Armv4I.msi

  WinCEPB60-090131-2009M01-Armv4I.msi

  WinCEPB60-090228-2009M02-Armv4I.msi

  WinCEPB60-090331-2009M03-Armv4I.msi

  WinCEPB60-090430-2009M04-Armv4I.msi

  WinCEPB60-090531-2009M05-Armv4I.msi

  WinCEPB60-090630-2009M06-Armv4I.msi

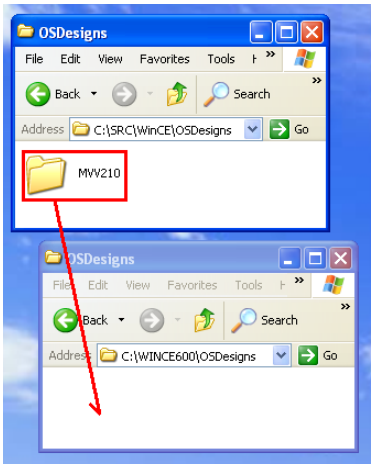
  WinCEPB60-090731-2009M07-Armv4I.msi
- Step8 Updating Windows CE 6.0 R3

If Step2 and Step3 are switched during the installation, the background icon may not be generated while the NK.bin, which is generated with sysgen, is being programmed into the board. If this occurs, common. reg in C:\WINCE600\PUBLIC\COMMON\OAK\FILES will need to be modified, as shown below, and sysgen will need to be performed again.

```
; @CESYSGEN IF GWES_MGTT

; @CESYSGEN IF CE_MODULES_MGTT_O || CE_MODULES_DECOMPDLL
[HKEY_LOCAL_MACHINE\System\GDI\FontFiles\TrueType]
; @CESYSGEN IF CE_MODULES_MGTT_O
	"EXT0"=multi_sz:".ttf",".ttc"
; @CESYSGEN ENDIF
; @CESYSGEN IF CE_MODULES_DECOMPDLL
	"EXT1"=multi_sz:".ac3"
; @CESYSGEN ENDIF
; @CESYSGEN ENDIF

; @CESYSGEN IF CE_MODULES_MGTT_O
[HKEY_LOCAL_MACHINE\System\GDI\FontDrivers\Microsoft TrueType]
	"File"="\\windows\\mgtt_o.dll"
	"Order"=dword:10
; @CESYSGEN ENDIF

; @CESYSGEN IF CE_MODULES_DECOMPDLL
[HKEY_LOCAL_MACHINE\System\GDI\FontDrivers\Microsoft AC3]
	"File"="\\windows\\decompdll.dll"
	"Order"=dword:10
; @CESYSGEN ENDIF

; @CESYSGEN ENDIF GWES_MGTT
```

2. Preparing to Compile Windows CE 6.0

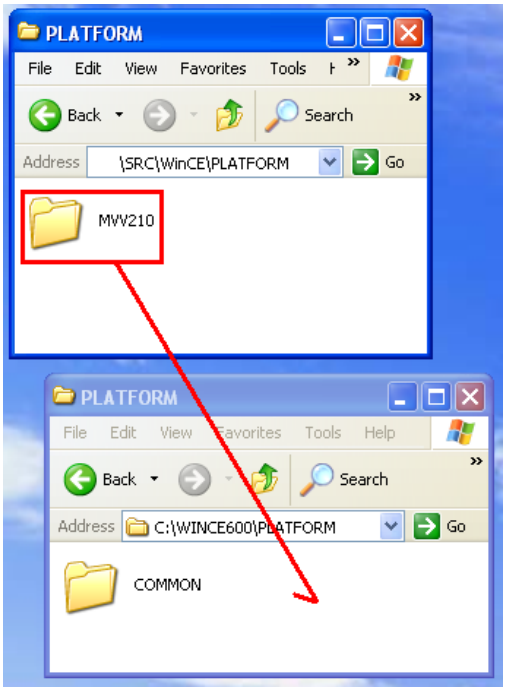
Once the Windows CE 6.0 installation is done, copy the folder MVV210 in the \SRC\WinCE\OSDesigns to the C:\WINCE600\OSDesigns. If OSDesigns doesn't exist, create the folder.



For the file **MVV210.pbxml**, the creation of a new project includes information from the catalog. Shown below is an example of the file MVV210.pbxml.

```
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_audio" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_auth" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_auth_ntlm" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_auth_schannel" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_autoras" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_aygshell" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_battery" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_certs" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_connmc" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_crypto" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_cursor" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_eap" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_fsreplbit" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_help" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_iabase" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_ie_jscript" />
<Feature Name="Item" CatalogItemId="Item:MS:sysgen_menu_overlap" />
```

The next step is to copy the MVV210 folder at \SRC\WinCE\PLATFORM from within the CD to C:\WINCE600\PLATFORM.



This MVV210 contains the Boot Loader for board operations, memory information, the timing for drivers, and all the device drivers for Kernel(NK.bin).

The source tree is written as shown below.

/ CATALOG :
Location for Catalog (MVV210.pbcxml)

/ CESYSGEN :
Compile PUBLIC or COMMON in CATALOG. The makefile at C:\WINCE600\PLATFORM\MVV210\CESYSGEN has the directory for compilation. !INCLUDE$(_WINCEROOT)\public\common\cesysgen\CeSysgen Platform.mak

/ DOC :
Document (Includes Samsung basic document)

/ ETC :
JTAG Emulator Script for board and USB device drivers

/ FILES :
Folder for application developers – includes applications, registry information, and the definition of the size of information for the boot and kernel

/ lib:
Library for ARM core

/ SAMPLES :
Sample code by silicon vendors

/ SRC :
Device drivers for CPU and board

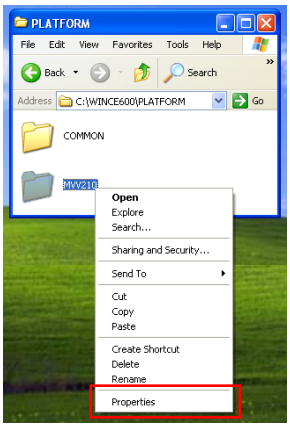
/ target :
Library for Target processor

The MVV210.bat shows the ON/OFF signals for the device drivers. A sample is shown below.

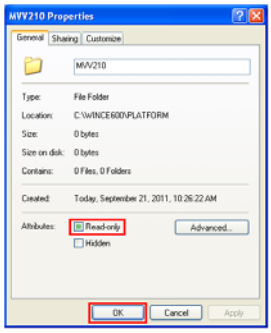
```
@REM ----- SD/MMC Driver Configuration    -------------------------
set BSP_NOSDMMC_CH0=
set BSP_USE_SDMMC_CH0_8BIT=
set BSP_NOSDMMC_CH1=1
set BSP_NOSDMMC_CH2=1
set BSP_USE_SDMMC_CH2_8BIT=
set BSP_NOSDMMC_CH3=
if /i "%BSP_NOSDMMC_BOOT%"==""" set BSP_NOSDMMC_CH0=1

@REM ----- SERIAL/SIR Device Driver Configuration    -----------------
@REM SERIAL/SIR Device Driver
set BSP_NOSERIAL=
set BSP_NOUART0=
set BSP_NOUART1=
set BSP_NOUART2=1
```

“1” indicates OFF (Not used).

If folders or files are copied to HDD from the CD, they will be “Read Only” and could cause compilation errors. The “Read Only” mode can be cleared as shown below by copying the MVV210 folder to the C:\WINCE600\PLATFORM.



Right-mouse click “Properties”

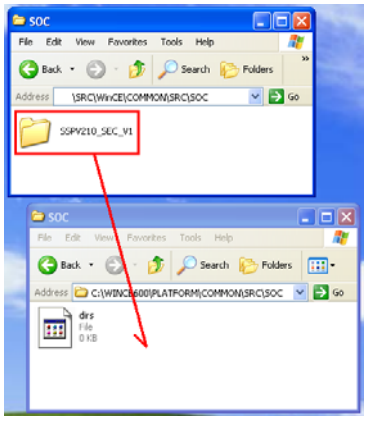


Uncheck “Read-only” and click “OK”.

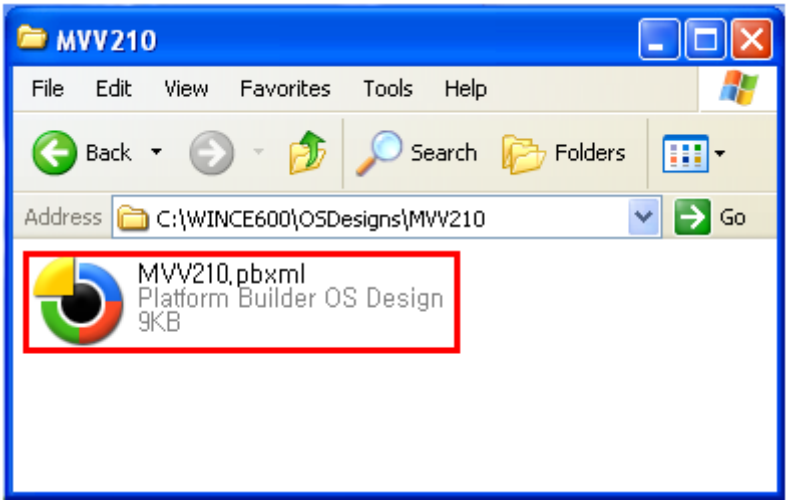
Copy the folder “S5PV210_SEC_V1” at \SRC\WinCE\COMMON\SRC\SOC in the CD to C:\WINCE600\PLATFORM\COMMON\SRC\SOC.

The folder has the following sources:

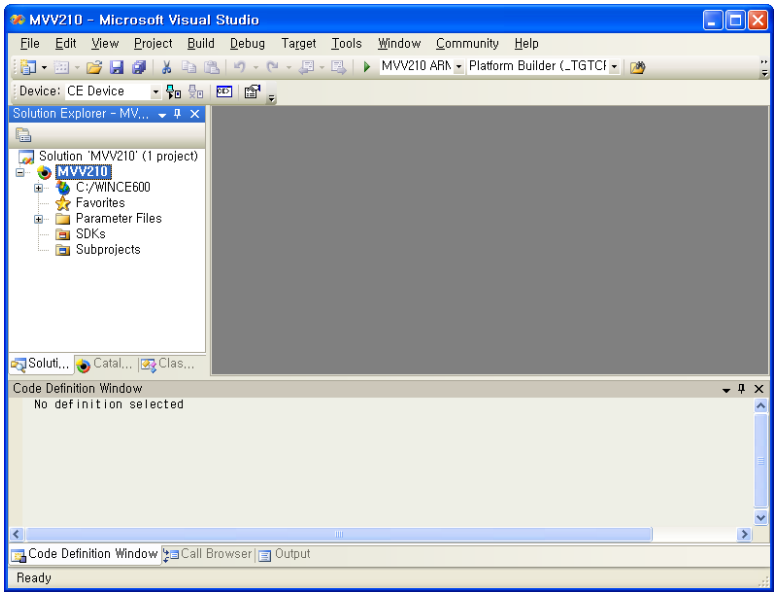
CACHE, INTR, IOCTL, MISC, POWER, PROFILER, RTC, SYSTEM, TIMER



Double click “MVV210.pbxml” in the folder C:\WINCE600\OSDesigns\MVV210 and execute Windows CE 6.0.

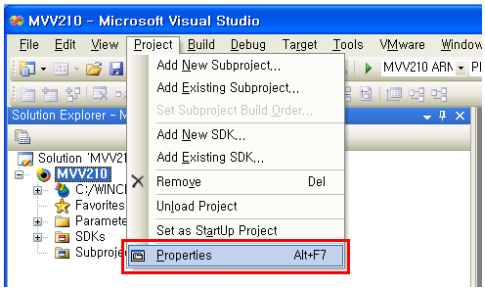


Display of the Window CE 6.0 Platform Builder:

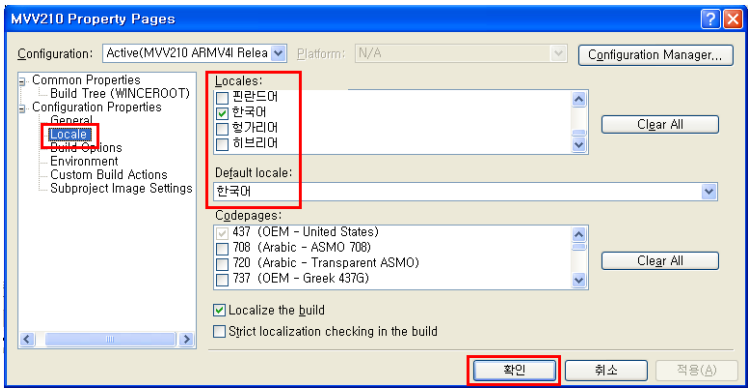


The following steps outline the set-up for sysgen:

Select Project → Properties to display the options for language and region.

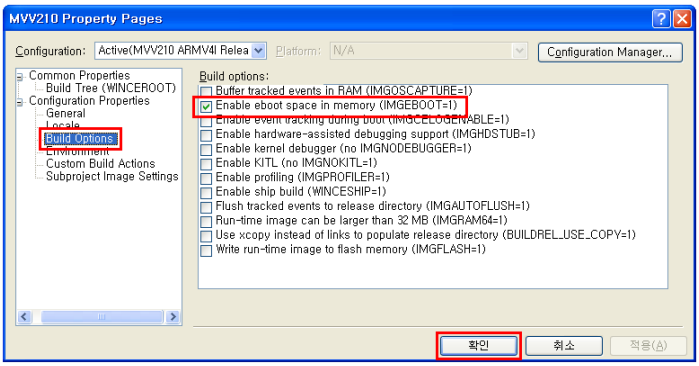


Choose the language under Locale.



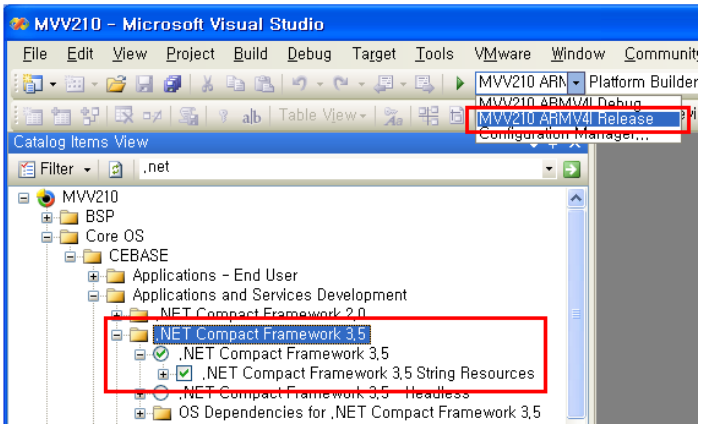
Locale language will be displayed and selected. However, any additional language must be added to the CATALOG. (Selecting more language will increase the size of the NK.bin.)

Select "Build Options" and click OK.

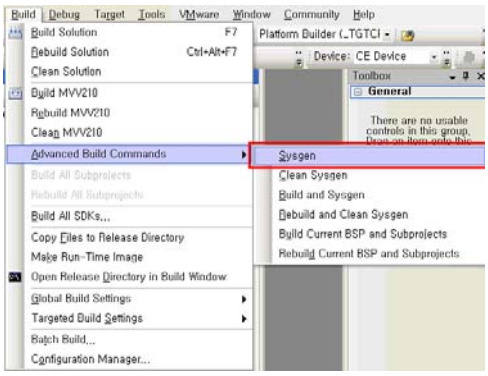


This BSP was created in Release Mode to create Sysgen.

(Reference: This BSP added .NET Compact Framework 3.5 for C#.)



Build → Advanced Build Commands ->Select Sysgen



Listed below are the various compilations.

Sysgen
Compile the sources at PUBLIC and COMMON and the BSP device drivers in CATALOG and then copy each library and boot loader kernel (nk.bin) to the C:\WINCE600\OSDesigns\MVV210\RelDir\MVV210_ARMV4I_Release.

Clean Sysgen
Remove the temporary files in the C:\WINCE600\OSDesigns\MVV210 \RelDir\MVV210_ARMV4I_Release and perform “Sysgen” again.

Build And Sysgen
Compile PUBLIC, COMMON, PRIVATE, and the device drivers from the board BSP and then copy each library and boot loader kernel (nk.bin) to the folder C:\WINCE600\OSDesigns\MVV210\RelDir\MVV210_ARMV4I_Release.

Rebuild And Clean Sysgen
Remove the temporary files in the C:\WINCE600\OSDesigns\MVV210 \RelDir\MVV210_ARMV4I_Release and perform “Build And Sysgen” again.

Build Current BSP and Subprojects
Assemble the current BSP and sub directories of the C:\WINCE600\OSDesigns\MVV210

Rebuild Current BSP and Subprojects
Remove the temporary files from the current BSP and the sub directories at C:\WINCE600\OSDesigns\MVV210 \RelDir\MVV210 and perform “Build Current BSP and Subprojects” again.

Sysgen is required at least once to finish the initial BSP copy process because the built images and libraries in the added CATALOG(/PUBLIC and /COMMON) must be copied to C:\WINCE600\OSDesigns\MVV210\RelDir\MVV210_ARMV4I_Release.

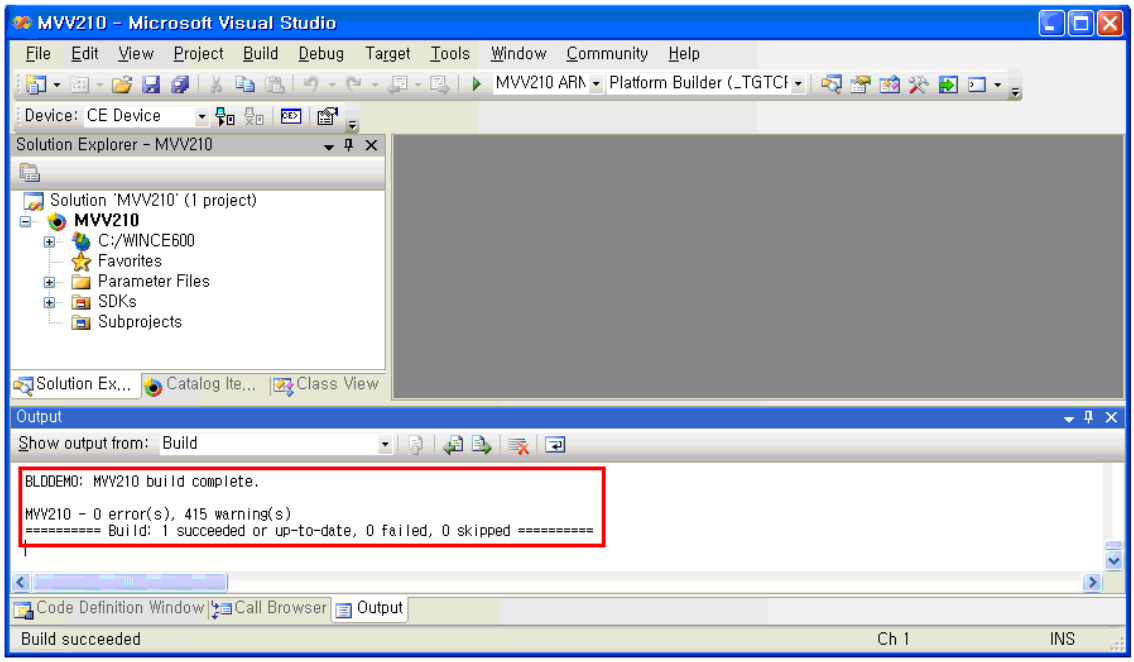
In most cases, “Sysgen” is more commonly used than **Build And Sysgen, Rebuild And Clean Sysgen, Build Current BSP and Subprojects, Rebuild Current BSP and Subprojects** because **Build And Sysgen, Rebuild And Clean Sysgen** can lead to the accumulation of PRIVATE and damage the Windows CE program files.

Microsoft does not recommend the build options as well. If necessary, they should only be used after the build period is over.

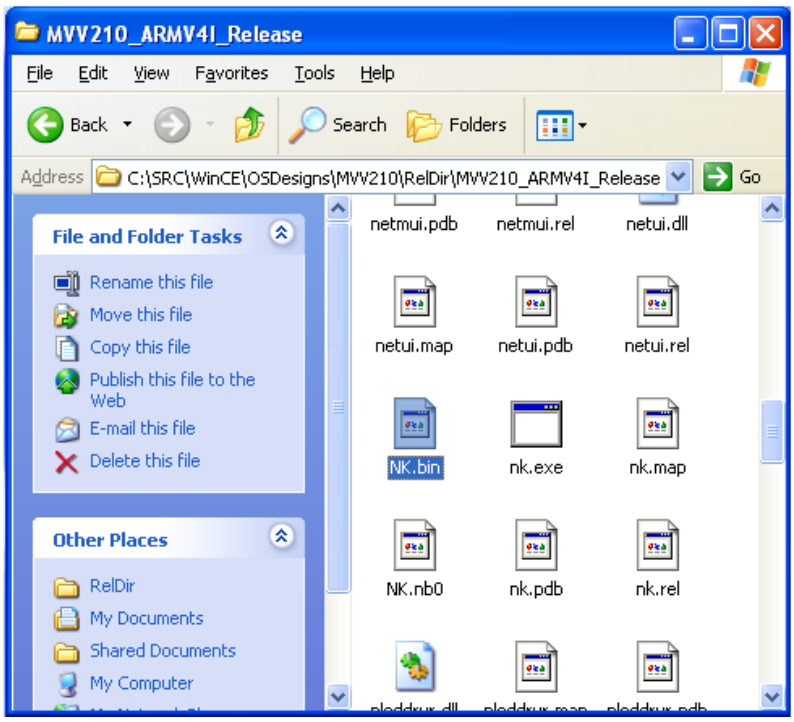
Please do not use the build option **Build And Sysgen, Rebuild And Clean Sysgen. Build Current BSP and Subprojects, Rebuild Current BSP and Subprojects** may remove the device drivers that remain unused.

We recommend using only sysgen.

Display of a successful compilation:



Once the compilation is over, Eboot, BL1_UART, bootimage, NK are generated in the folder C:\WINCE600\OSDesigns\MVV210\RelDir\MVV210_ARMV4I_Release.



Descriptions of the built images:

BL1_UART.nb0
Transmits this image to the DDR2 memory using the UART port to execute the Program Counter (PC)

EBOOT.nb0
Boot Loader and this image are loaded onto the PC address 0x47700000 and executed on DDR2 memory location

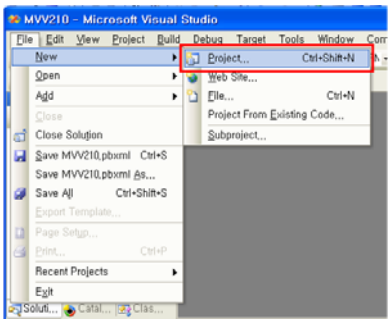
bootimage.nb0
Boot Loader as well as Eboot.nb0, which is executed via DDR2, program this image (bootimage.nb0) to the NAND flash memory.

NK.bin
An image including both Kernel and File System

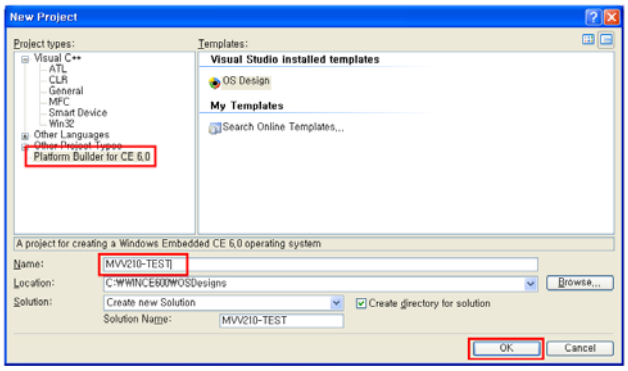
3. Building a new project

Previously, the project was already built into the MVV210.pbxml and the CATALOG was included. This chapter will explain the sysgen procedure of making the new project and adding as well as configuring the CATALOG.

Select File → New → Project



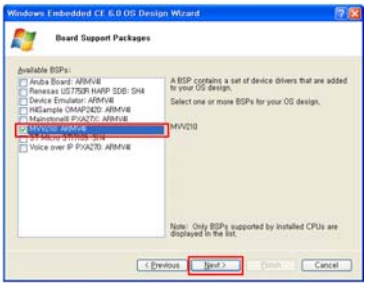
After choosing the Platform Builder for CE 6.0, create a project name and press "OK".



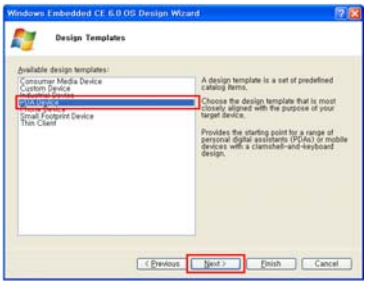
Please follow from left to right.

Click "NEXT"

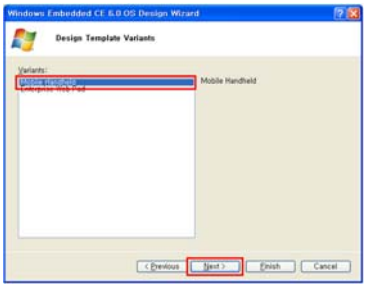
Select "MVV210" in the BSPs list and click "NEXT"



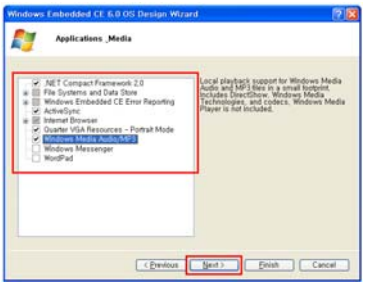
Select "PDA Device" and click "NEXT"



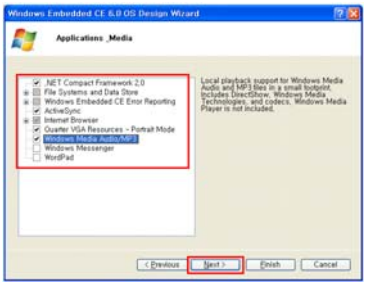
Select “Mobile Handheld” and “NEXT”



Select accordingly and click “NEXT”

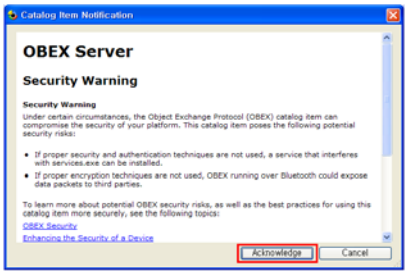


Select accordingly and click “NEXT”

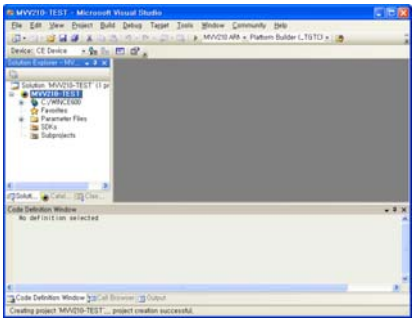


Click “Finish”

Click “Acknowledge”

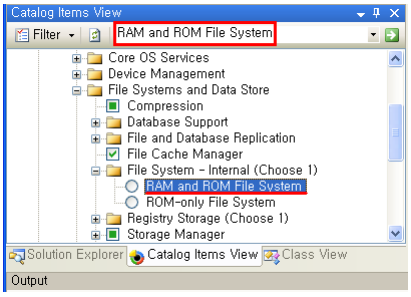


Completed

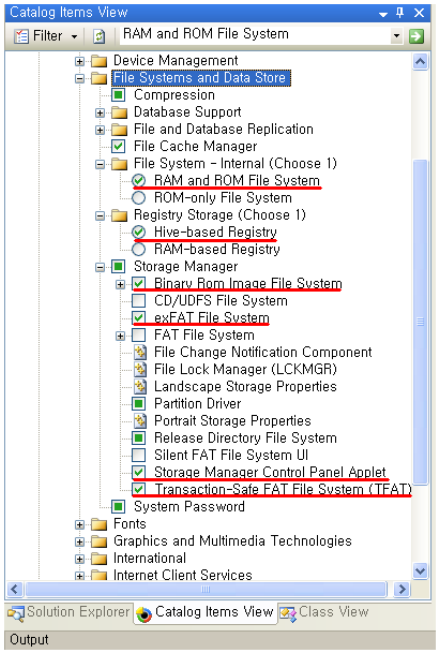


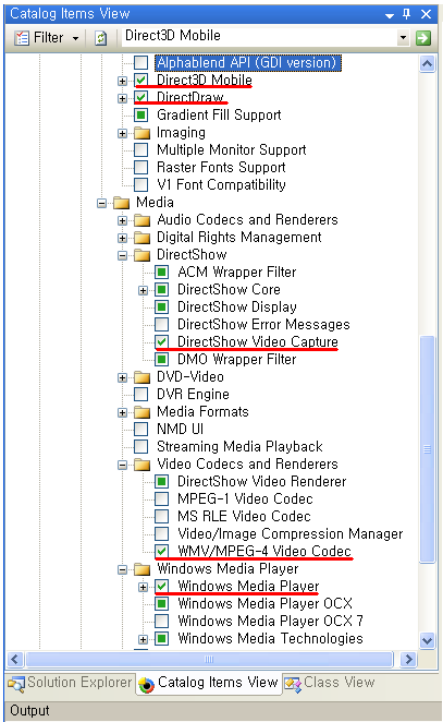
After making the new project, “Catalog Items” should be added before Sysgen (Compile).

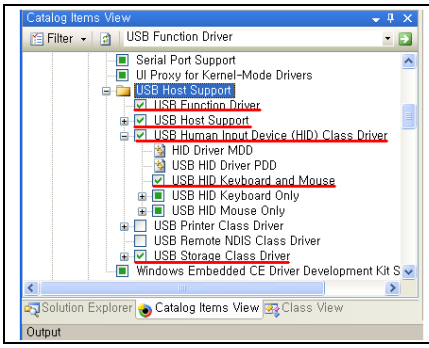
As shown in the figure below, search “RAM and ROM File System” to easily locate the file.

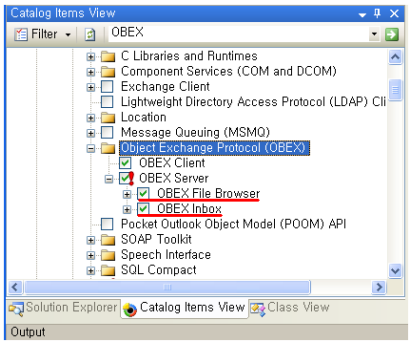


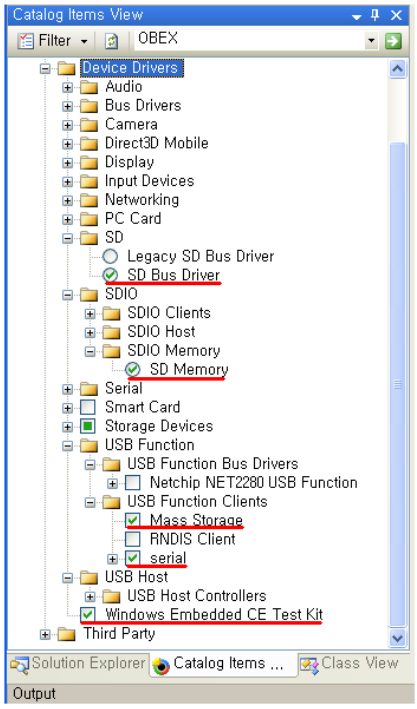
Additional Catalog Items

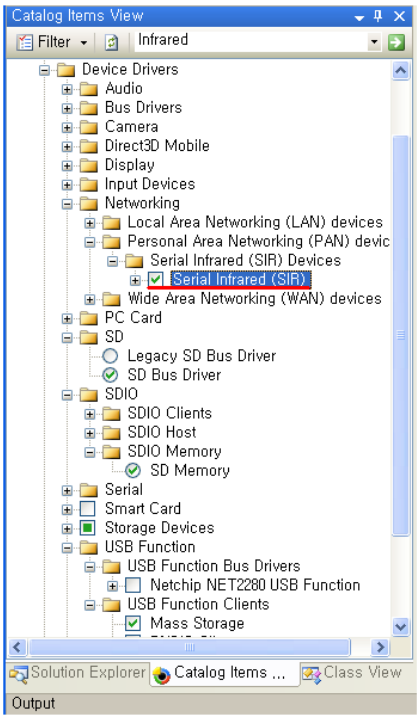
	**RAM and ROM File System** **Hive-based Registry** **Binary Rom Image File System** **exFAT File System** **Storage Manager Control Panel Applet** **Transaction-Safe FAT File System**

	**Direct3D Mobile** **DirectDraw** **DirectShow Video Capture** **WMV/MPEG-4 Video Codec** **Windows Media Player**

	**USB Function Driver** **USB Host Support** **USB Human Input Device(HID) Class Driver** **USB HID Keyboard and Mouse** **USB Storage Class Driver**

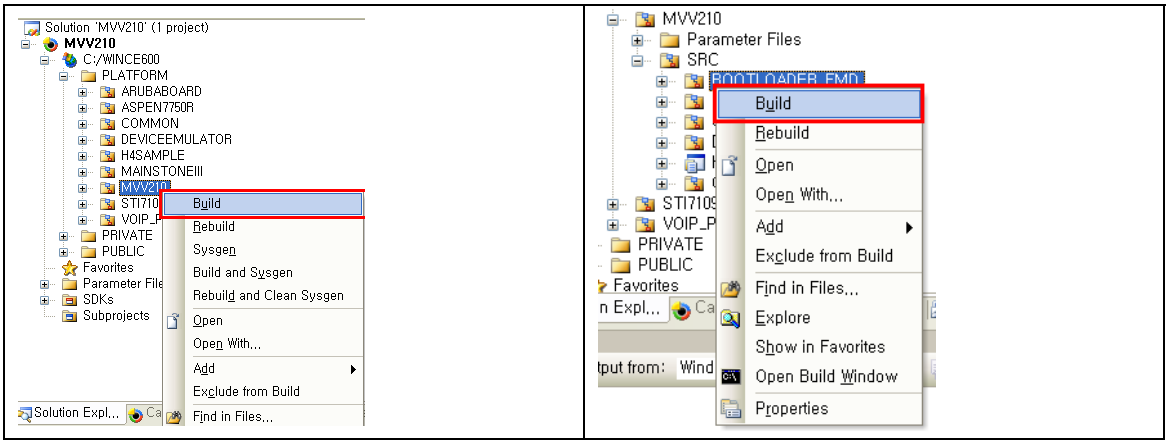
	**OBEX File Brower** **OBEX Inbox**

	**SD Bus Driver** **SD Memory** **Mass Storage** **serial** **Windows Embedded CE Test Kit**

	**Serial Infrared (SIR)**

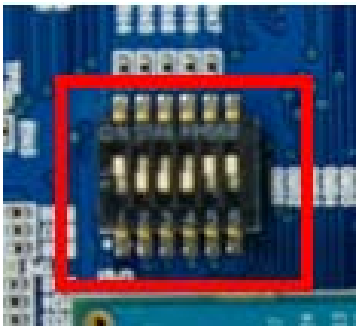
After adding the CATALOG, use sysgen.

However, use sysgen only when the CATALOG (/PUBLIC) added has been provided by Microsoft. When building the source in the board BSP, use the pictures below to choose the specific folders required for building.

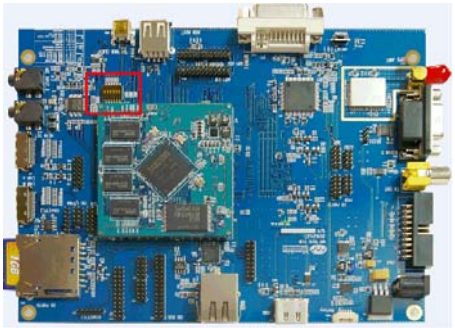


4. Downloading

As indicated in the MVV210_Operating.pdf, BootLoader is viable even without the JTAG Emulator for S5PV210. Currently there is Eboot on the NAND Flash but for the case that it is not present, we will begin with programming Eboot.
First of all, set the DIP switch to iROM booting mode for image transfer. With this mode, Eboot and NK may be downloaded using the USB.



iROM boot mode for image transfer

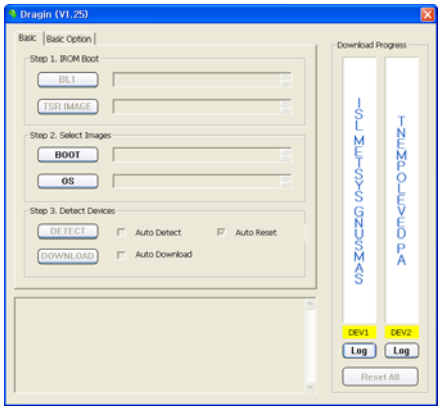


DIP switch location

The images for the programming are as follows: bootimage.nb0, BL1_UART.nb0, EBOOT.nb0, /TOOL/GUYI Downloader For WinCE from NK.bin CD and the Dragin V1.25.exe file from the folder \SRC\WinCE\PLATFORM\MVV210\ETC\Tools\GUI Downloader.

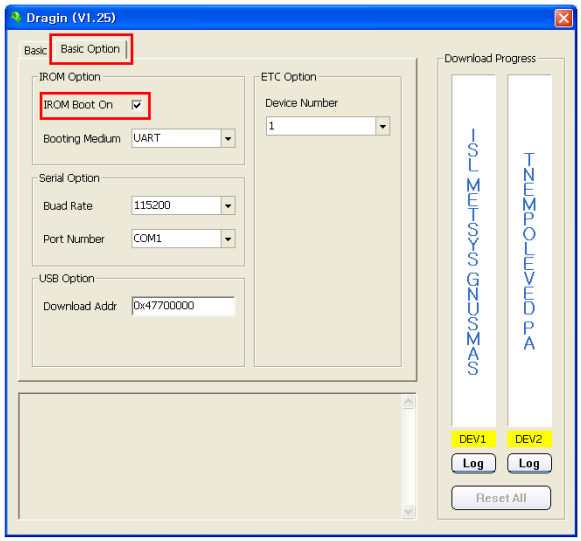
The next step is to program the image by using the execution file.

Execute (double click) Dragin V1.25.exe without applying power to the board. Note that because BL1_UART.nb0 is a the UART transmission, all serial programs will be terminated.



<Dragin V1.25.exe>

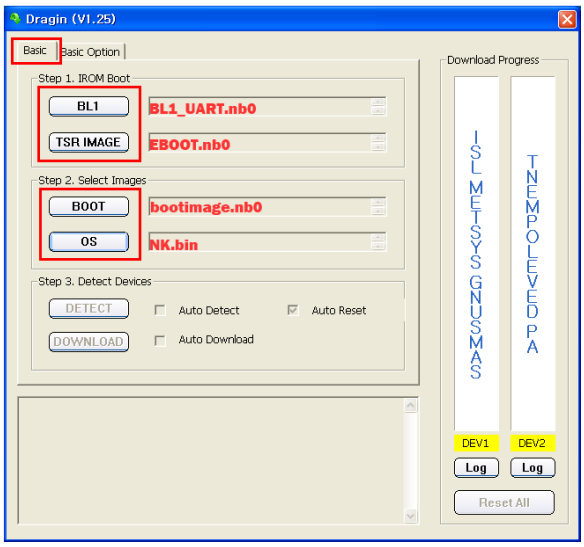
From the Basic Option menu, select IROM Boot On.



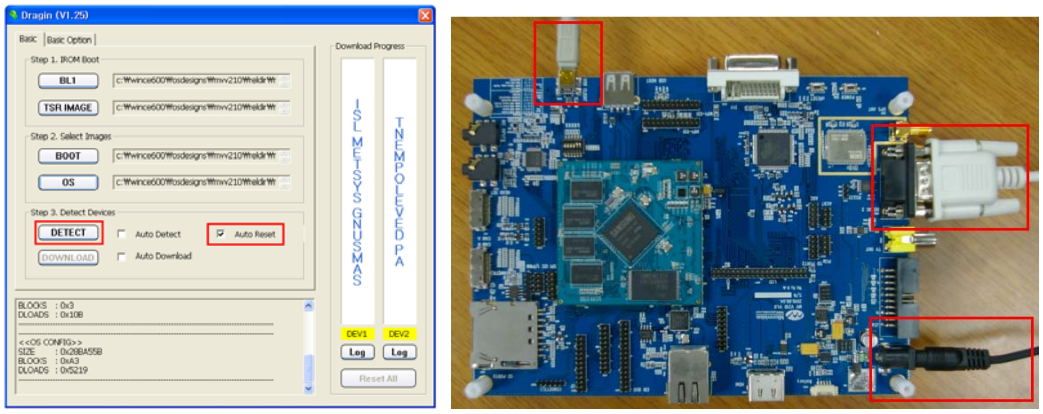
As shown in the picture below, locate the files from the directories in the basic menu by clicking BL1, TSR IMAGE, BOOT, OS.

Default directory C:\WINCE600\OSDesigns\MVV210\RelDir\MVV210_ARMV4I_Release

BL1	→ BL1_UART.nb0
TSR IMAGE	→ EBOOT.nb0
BOOT	→ bootimage.nb0 (steplodr + eboot)
OS	→ NK.bin



After each image loads, check the option AUTO Reset and then click DETECT.



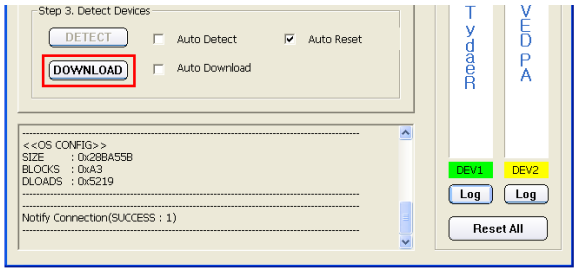
As demonstrated in the picture above, connect the USB cable and RS232 to the PC and power ON. If the DOWNLOAD button has been activated, click it.

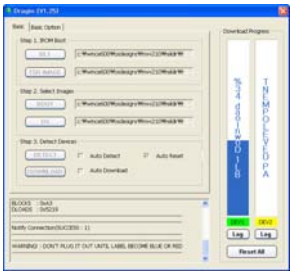
In the beginning, “BL1_UART.nb0” will be transferred through UART. The process will initialize DDR2 and EBOOT.nb0 will be downloaded to the Eboot program counter “0x47700000” through the USB. If the message “Hardware Found” appears for the USB, install the USB driver, which can be found in the \TOOL\WinXP USB Driver in the CD.

(Caution: This USB driver supports only 32Bit computers)

Once the USB installation ends, it automatically resets the power. Click the DOWNLOAD button and if you want to link your download with the power reset, check “Auto Download.”

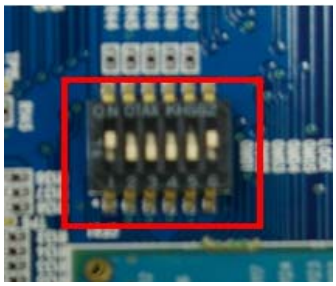
If USB driver is already installed in the PC, the DOWNLOAD button will start programming without the power reset.



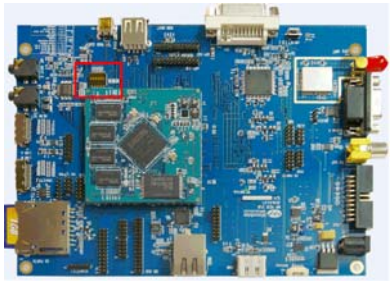


<Image Transfer>

The images are automatically transferred and Windows CE is automatically booted. After the programming has finished, set the DIP switch for the boot mode as shown below.



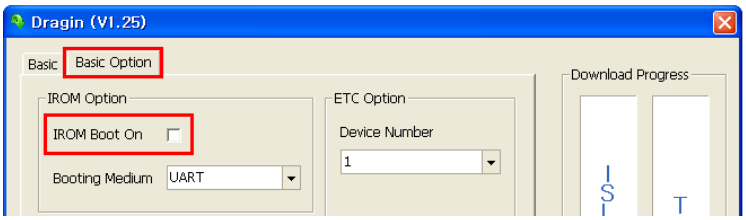
Automatic Boot Mode(NAND)



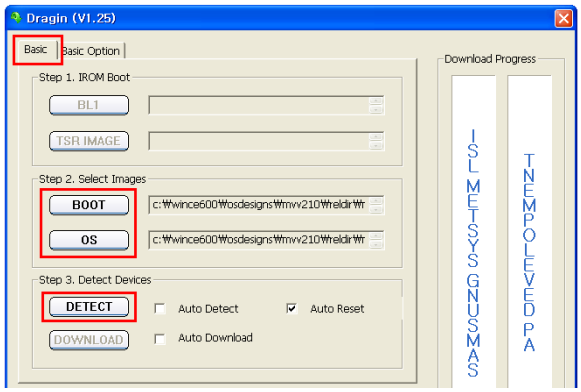
Boot switch location

Finally, once Bootloader has been uploaded, the followings changes must be made.

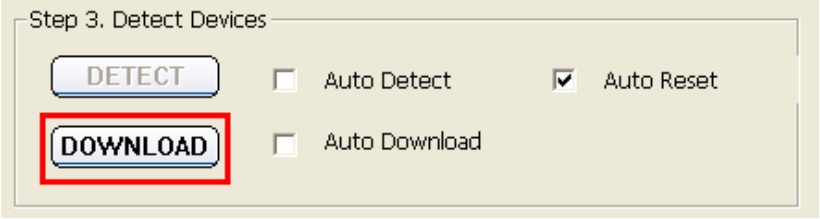
In the Basic Option menu, uncheck IROM Boot On.



In the Basic Menu, load the image only for BOOT and OS and then click DETECT.



If the power is connected, the DOWNLOAD window will pop up. Click to program automatically.



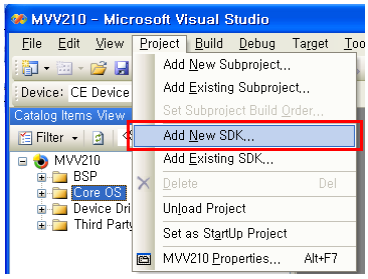
5. Application Program

5.1 Building SDK

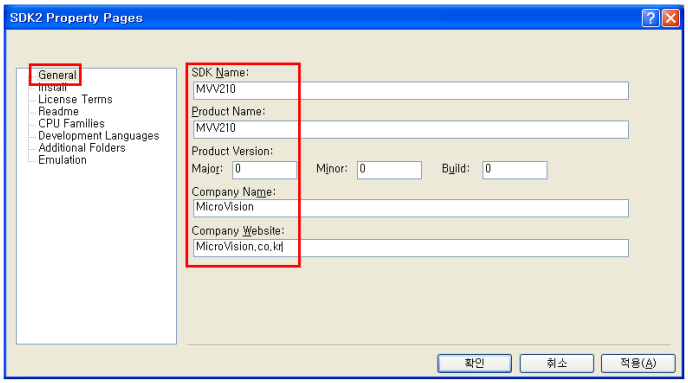
The Software Development Kit (SDK) is a package for building the application program. Including the board library and functions, SDK must be built from the compilation (Sysgen) BSP.

The following steps outline how to generate it (Left to Right, Top to Bottom)

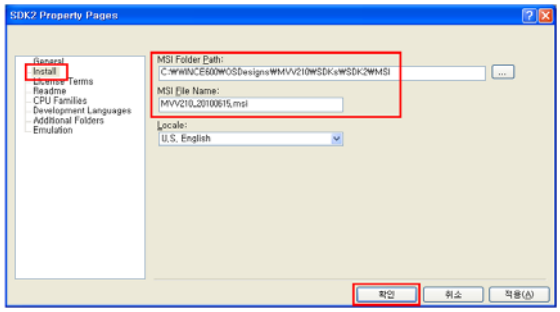
Project → "Add New SDK"



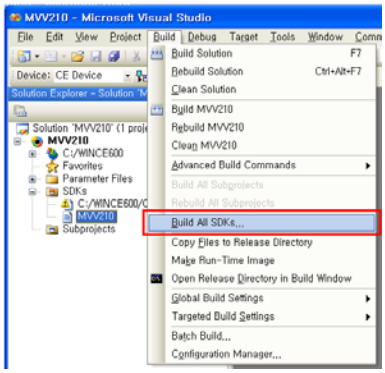
Name the project and click "OK"



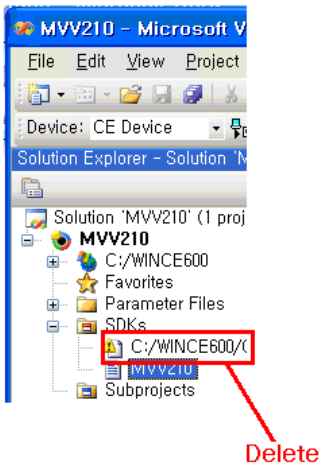
Accordingly fill in the file name and directory for storage and click "OK"

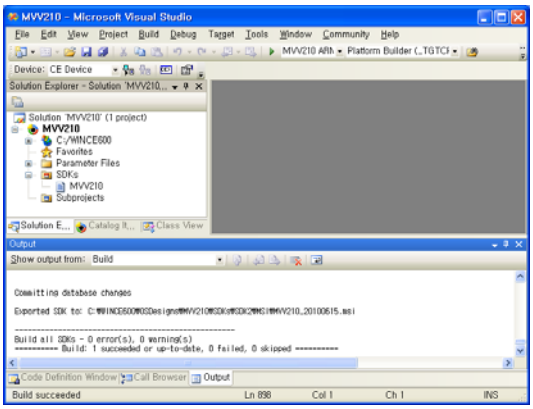


Build → "Build All SDKs"



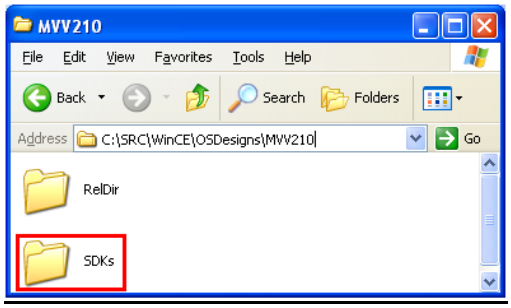
In case the build is not made, remove the SDK path.





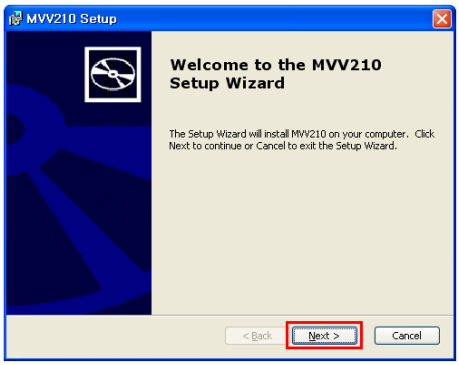
Finished state

After the build is over, default the SDK file location as shown below.

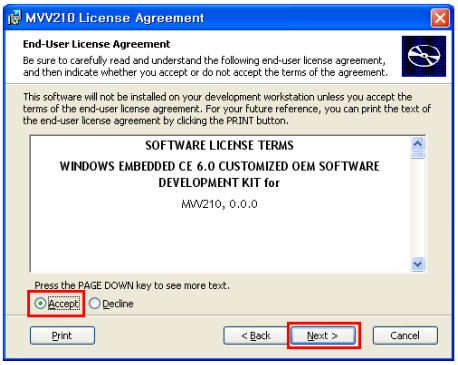


Execute MVV210_20100615.msi.mis in C:\WINCE600\OSDesigns\MVV210\SDKs\SDK2\MSI

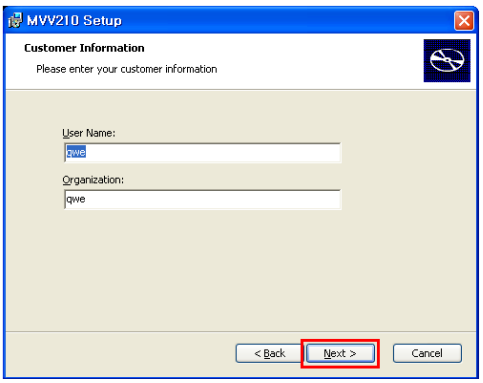
"NEXT"



"Accept" -> "NEXT"

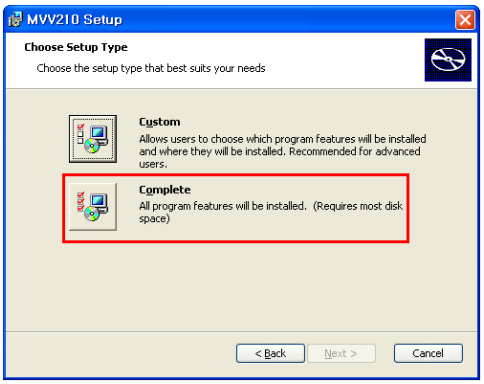


"NEXT"

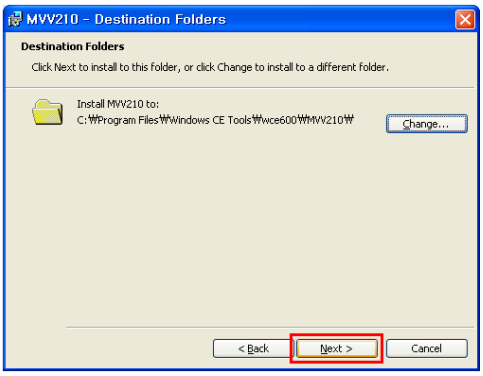


"Complete"

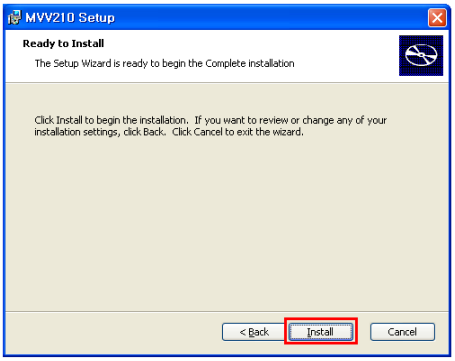
(For Microsoft Visual Studio 2008 users, click "Custom" and install by removing Document.



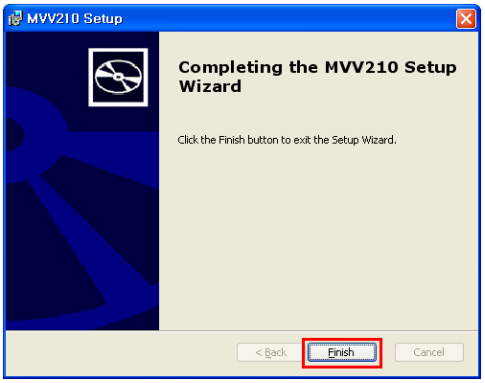
"NEXT"



"Install"

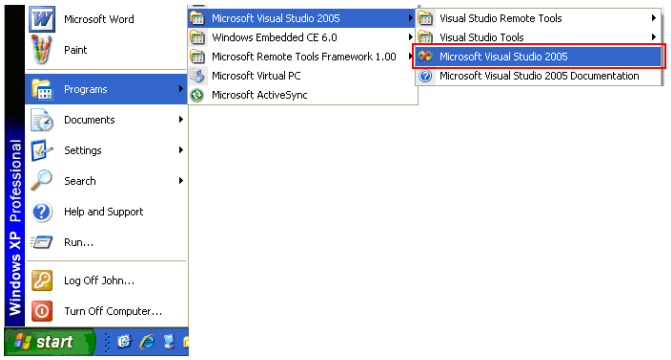


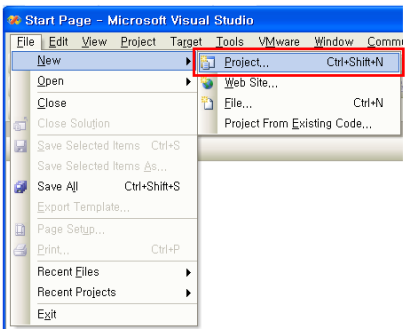
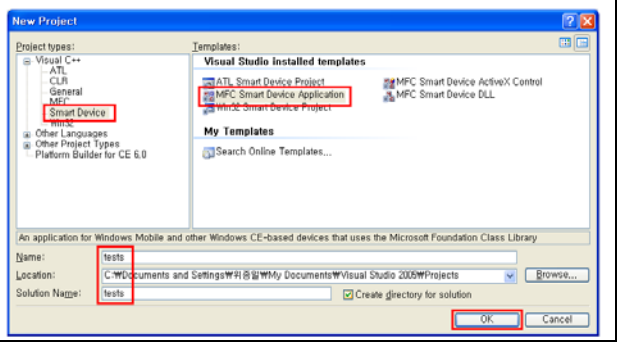
"Finish"



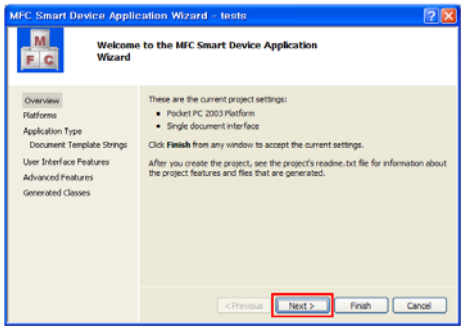
5.2 Building New Project for Application

Execute Visual Studio 2005

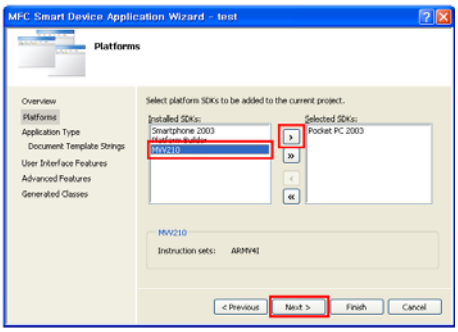


NEW -> Project	“Smart Device” -> “MFC Smart Device Application” -> Type in the project name and click “OK”
	

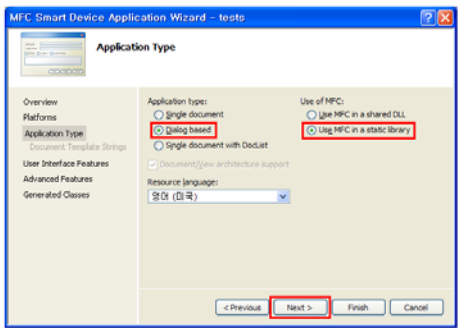
“NEXT”



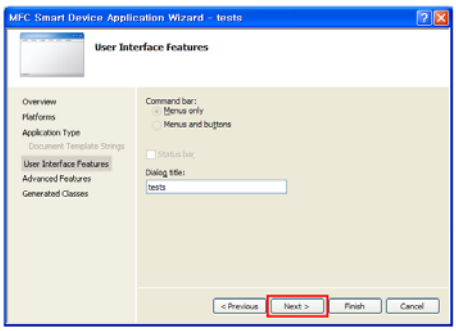
Select “MVV210” and click “>” and then “NEXT”



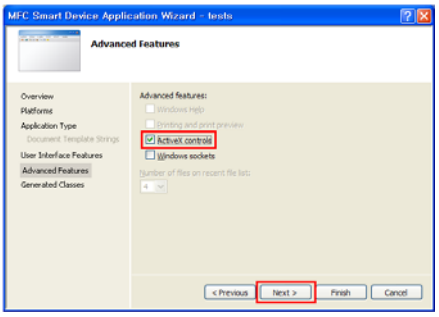
"Dialog based," "Use MFC in a static library" and click "NEXT"



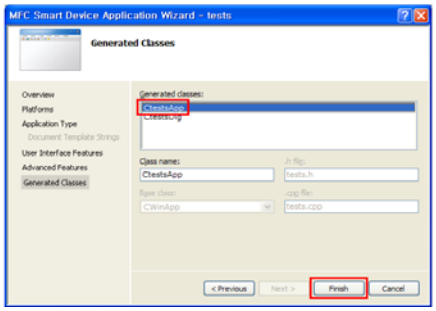
"NEXT"

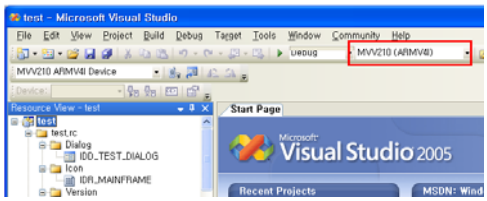


"ActiveX controls" and click "NEXT"



"CtestsApp" and click "Finish"



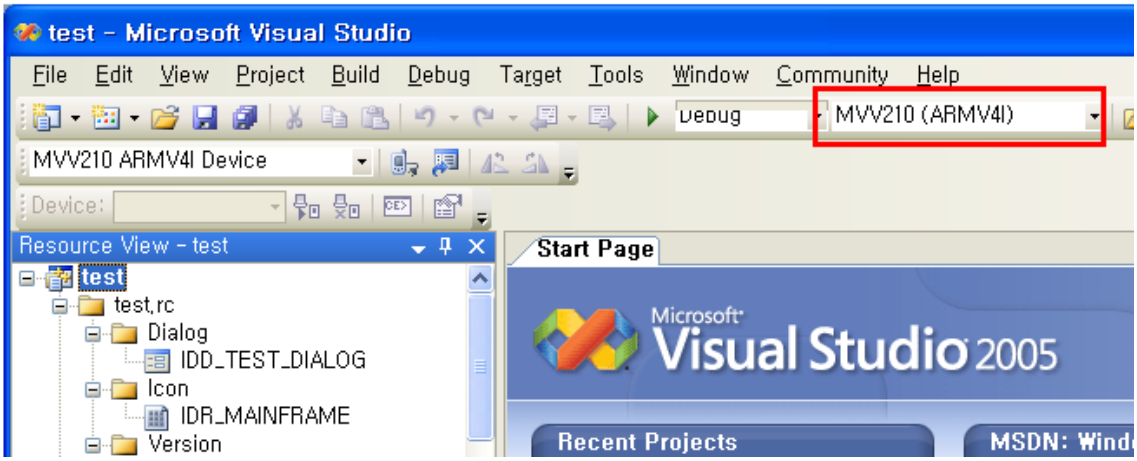


Finished SDK

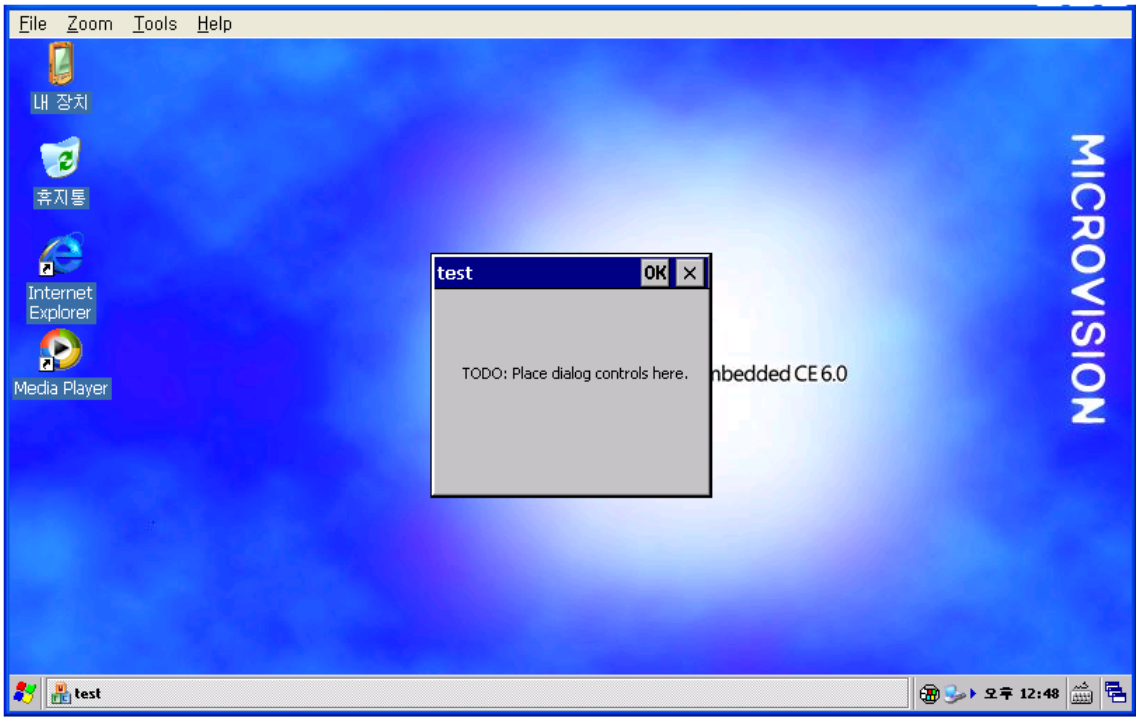
5.3 Building Application

After booting with Windows CE, use a USB cable to connect the board to the PC in order to enable ActiveSyne. If the installation window pops up, install the driver in the USB at the directory \TOOL\WinXP located within the CD.

As shown in the picture below, select SDK (MVV210(ARMV4I)) and press “Ctrl + F5” to send to the board via ActiveSync as soon as the build finishes.

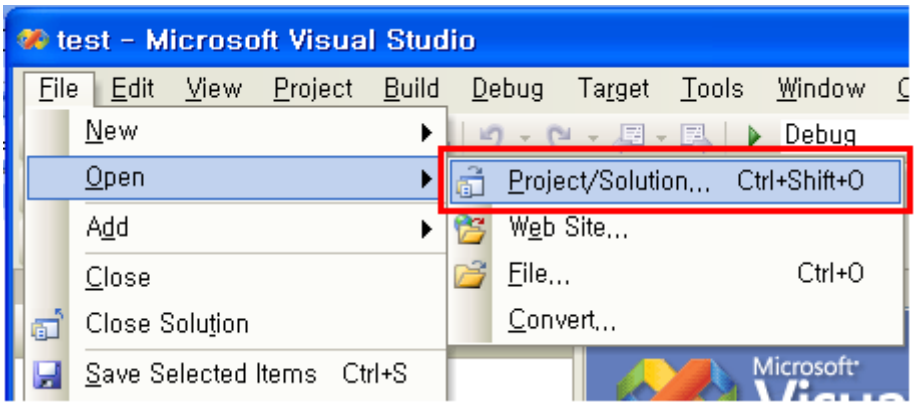


Windows CE image on board through the USB ActiveSync



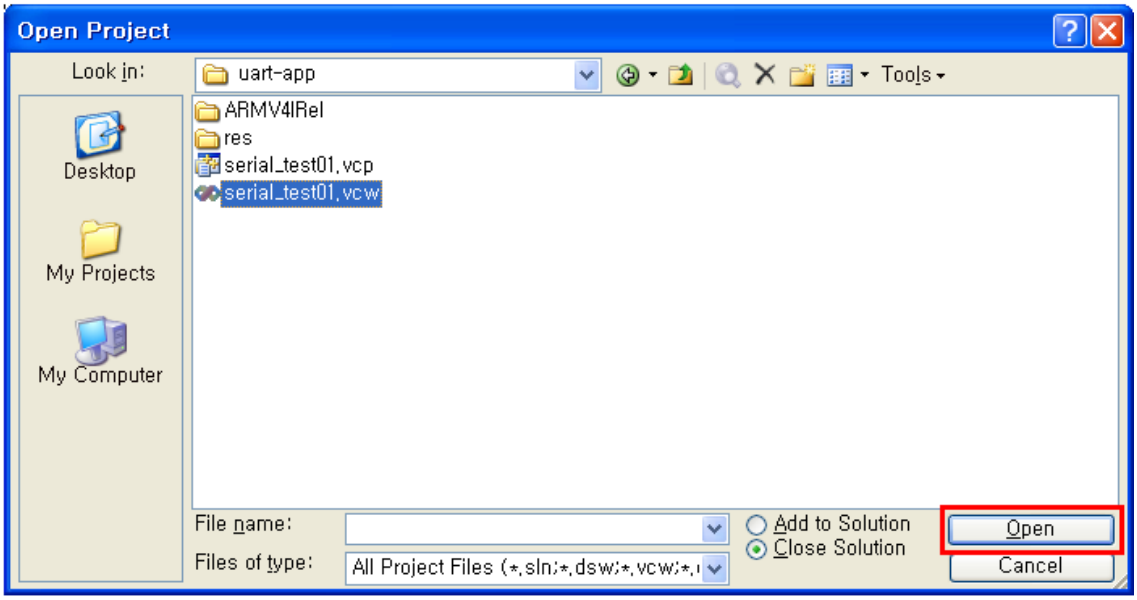
The screen above shows the window following execution, after the built image (execution file) has been sent to the DDR2 section on the board. Application developers normally send the build image to the board and then test it for any malfunctions.

If there is another source file, click “File -> Open -> Project/Solution” as displayed below and open the necessary file accordingly from the copied directory.

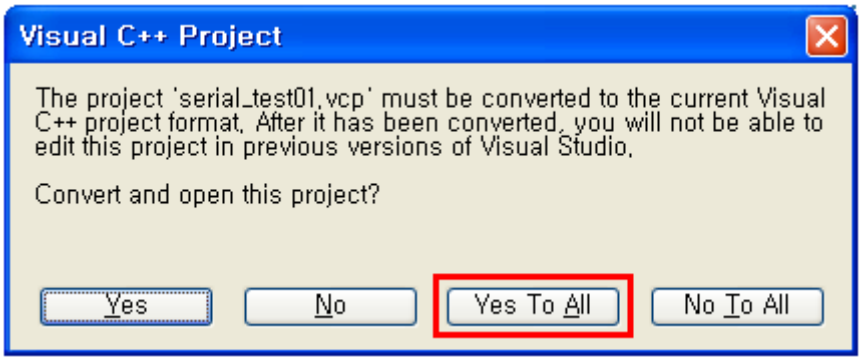


This document will use “uart-app” in the C:\Documents and Settings\20110101\My Documents\Visual Studio 2005\Projects. The directory name should not be in any language other than English.

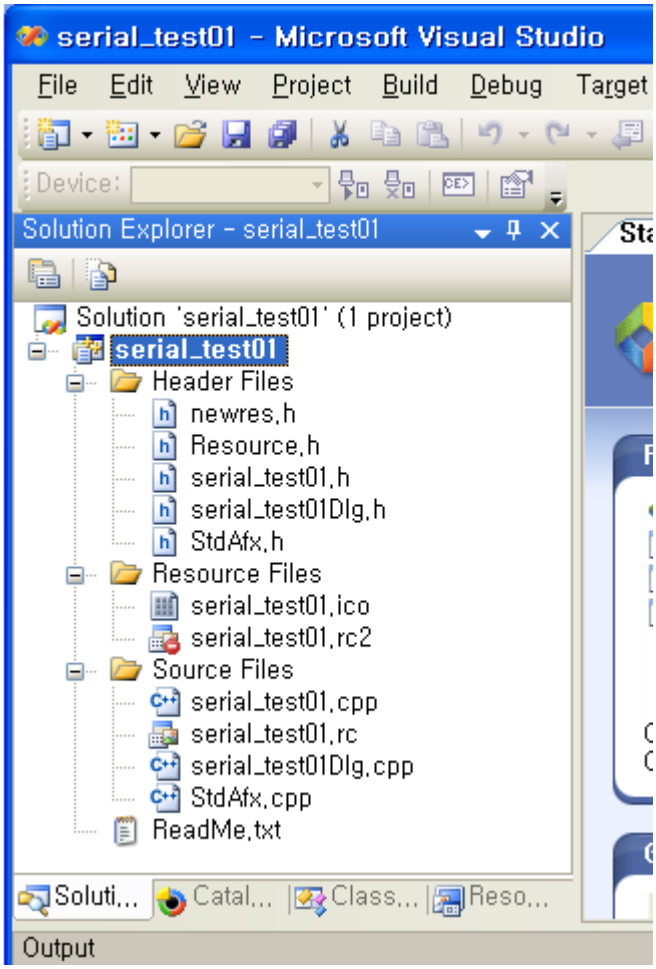
Select the source files and select “OPEN”.



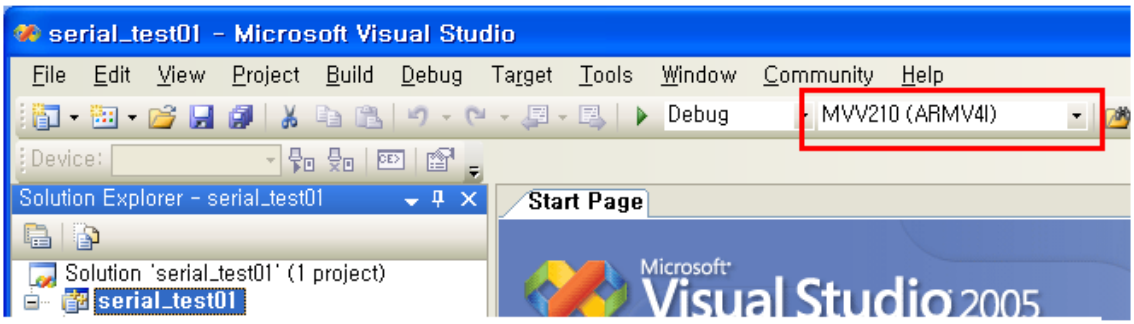
If this project is opened with another tool program, it will ask for a conversion. In that case, click “Yes To All”.



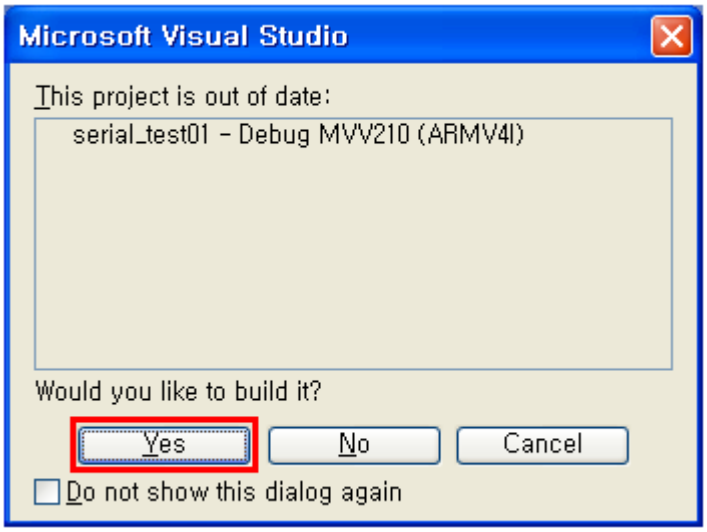
After the conversion has finished



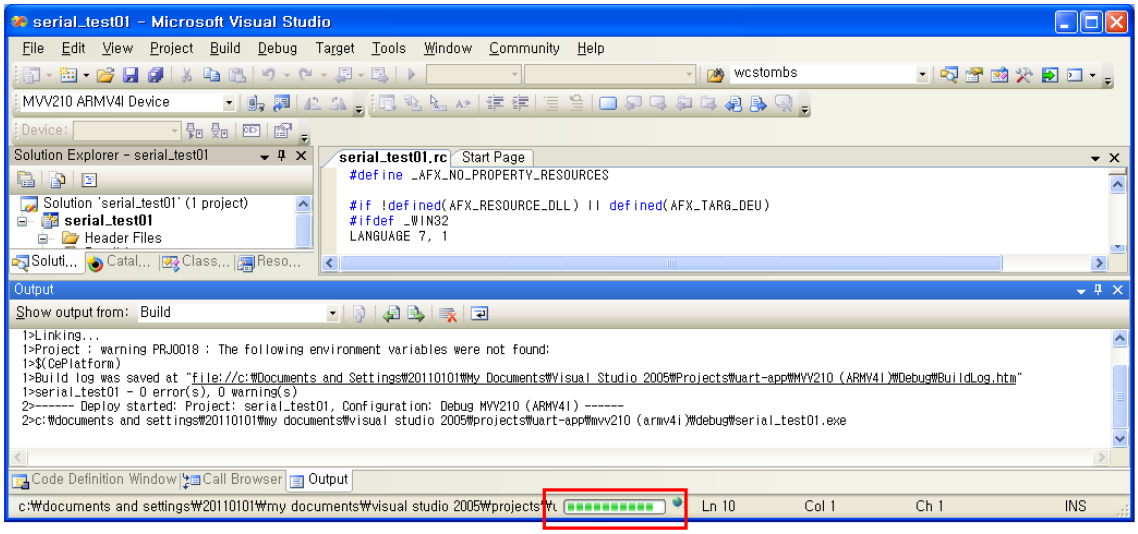
Select MVV210 SDK and press “Ctrl + F5” to build the project.



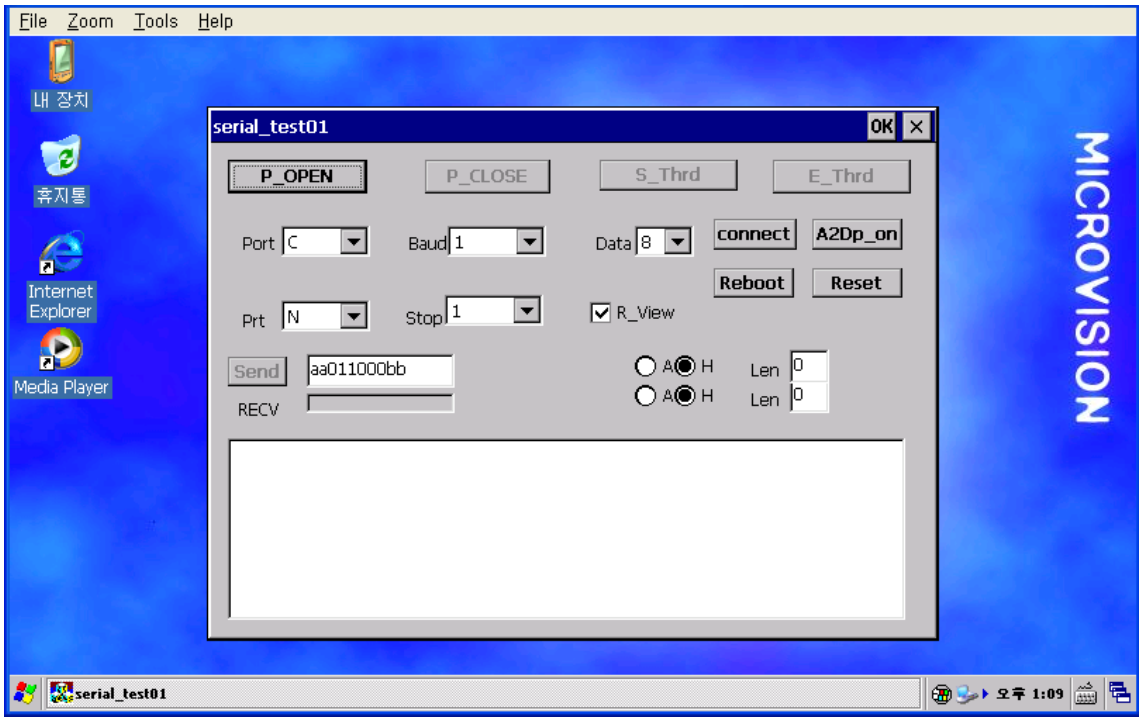
Click “Yes”



Transferring State

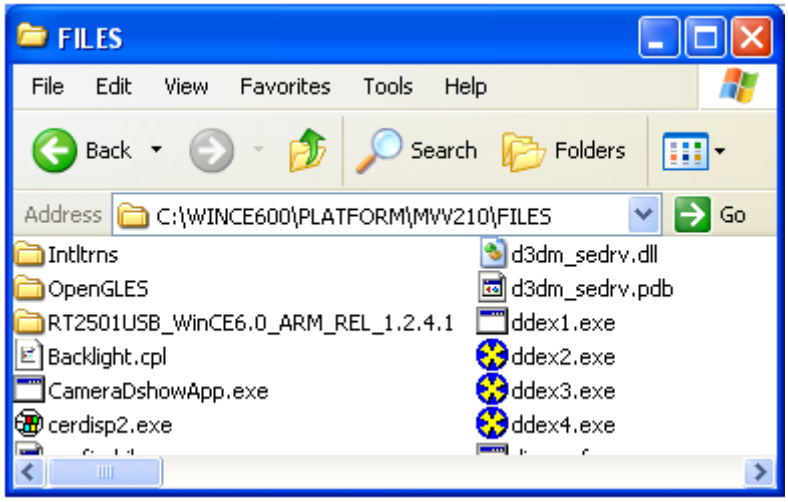


After execution

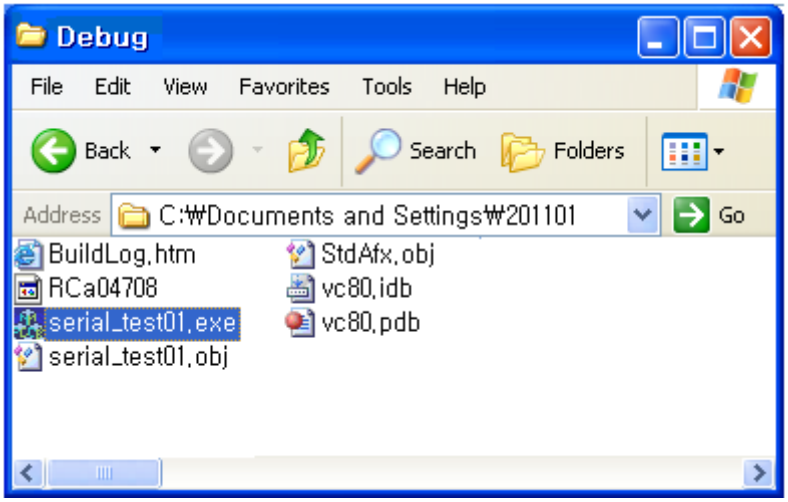


Note that if the execution program is running on DDR2 and not on NAND, after rebooting, the program will disappear. Therefore, in the following descriptions, we explain how to upload only the images that are successfully built and executed on the NAND flash memory of the board.

When you look inside the directory C:\WINCE600\PLATFORM\MVV210\FILES, many files will be present, as shown below.

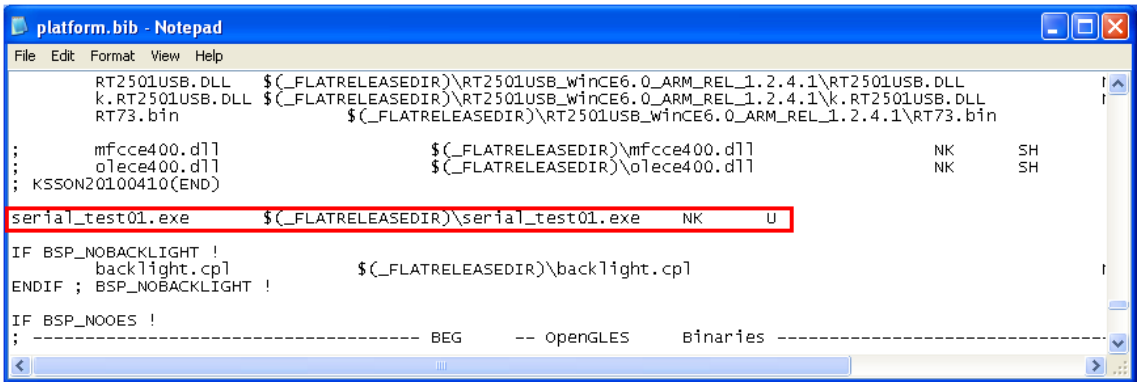


The application execution file “serial_test01.exe” can be found in the directory C:\Documents and Settings\20110101\My Documents\Visual Studio 2005\Projects\uart-app\MVV210 (ARMV4I)\Debug.

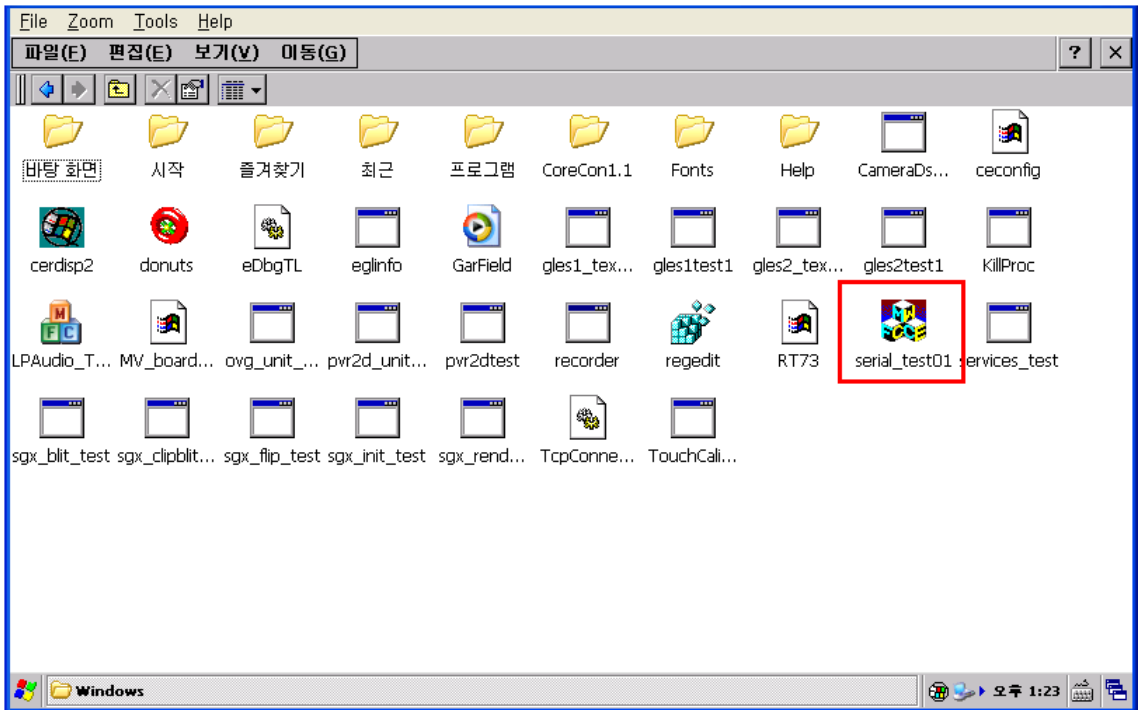


Copy this file “serial_test01.exe” to the directory C:\WINCE600\PLATFORM\MVV210\FILES and open the “platform.bib” at C:\WINCE600\PLATFORM\MVV210\FILES using file editor. Add the line serial_test01.exe $(_FLATRELEASEDIR)\serial_test01.exe NK U, as shown below, and then build again so that the file “serial_test01.exe” will be included in the NK.bin.

The location $(_FLATRELEASEDIR) indicates “Internal” -> “Windows” folder.

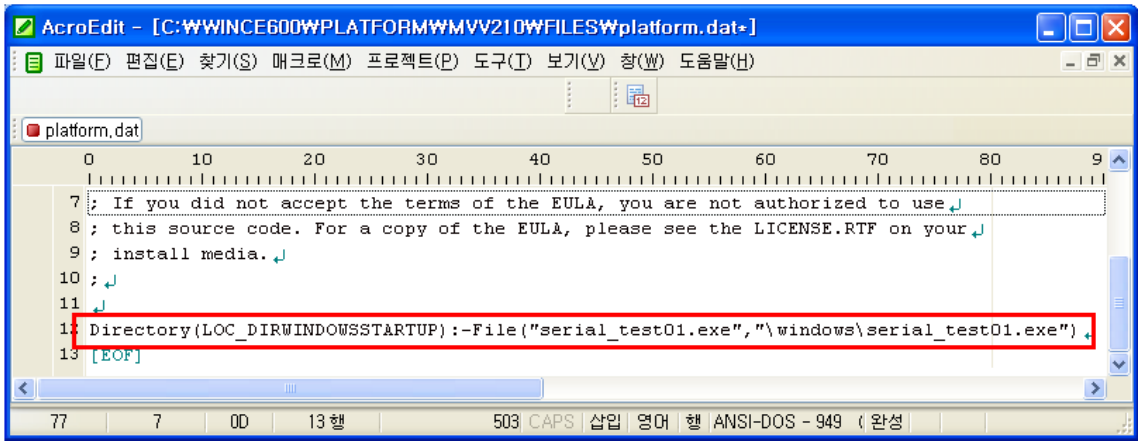


New NK.bin programmed onto the board



If you want automatic booting with Windows CE, open “platform.dat” in the directory C:\WINCE600\PLATFORM\MVV210\FILES and insert the script listed below.

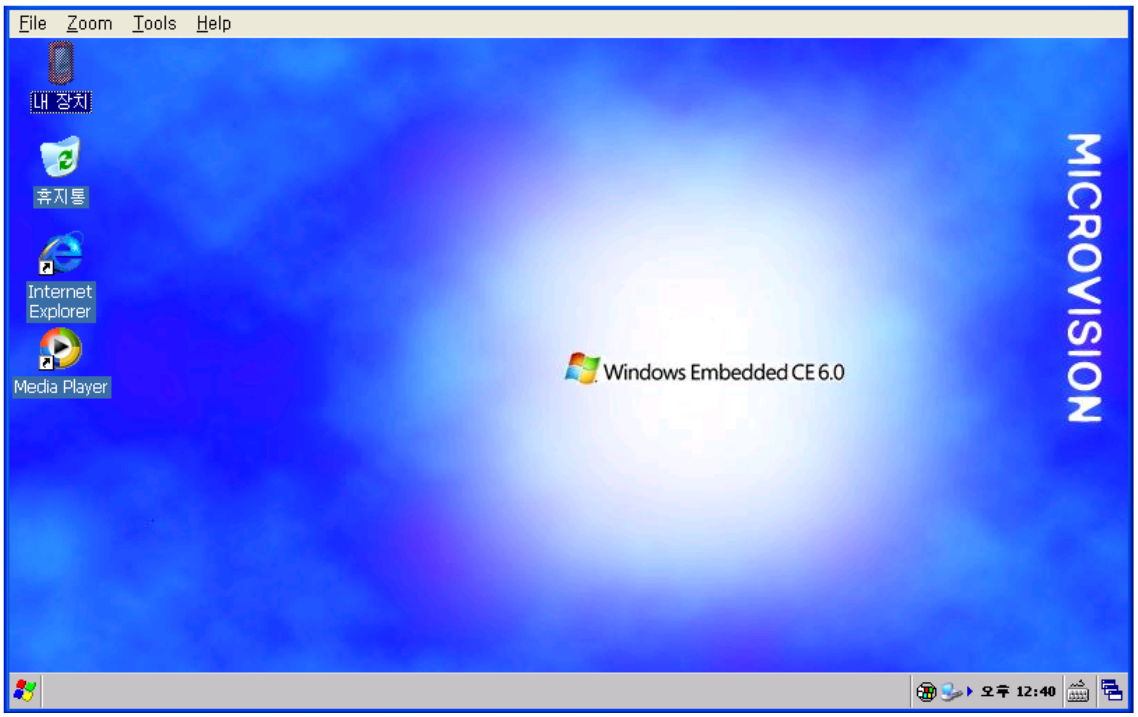
Directory(LOC_DIRWINDOWSSTARTUP):-
File("serial_test01.exe","\windows\serial_test01. exe")



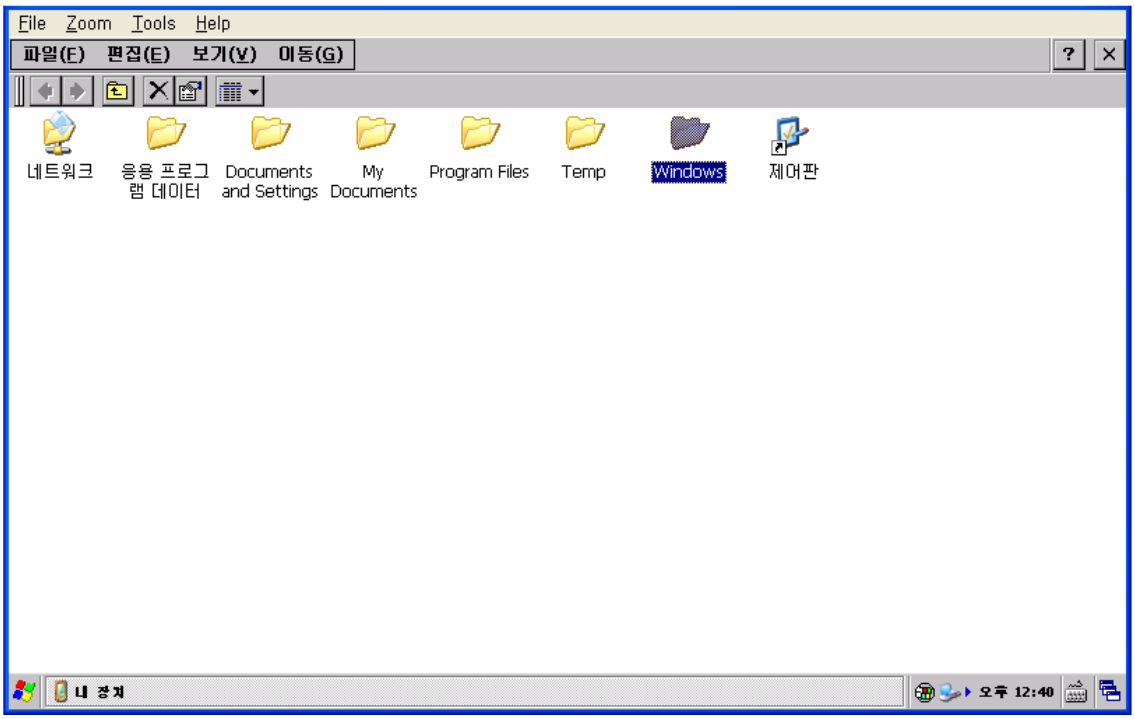
5.4 Operating Camera

Connect the camera module to the board.

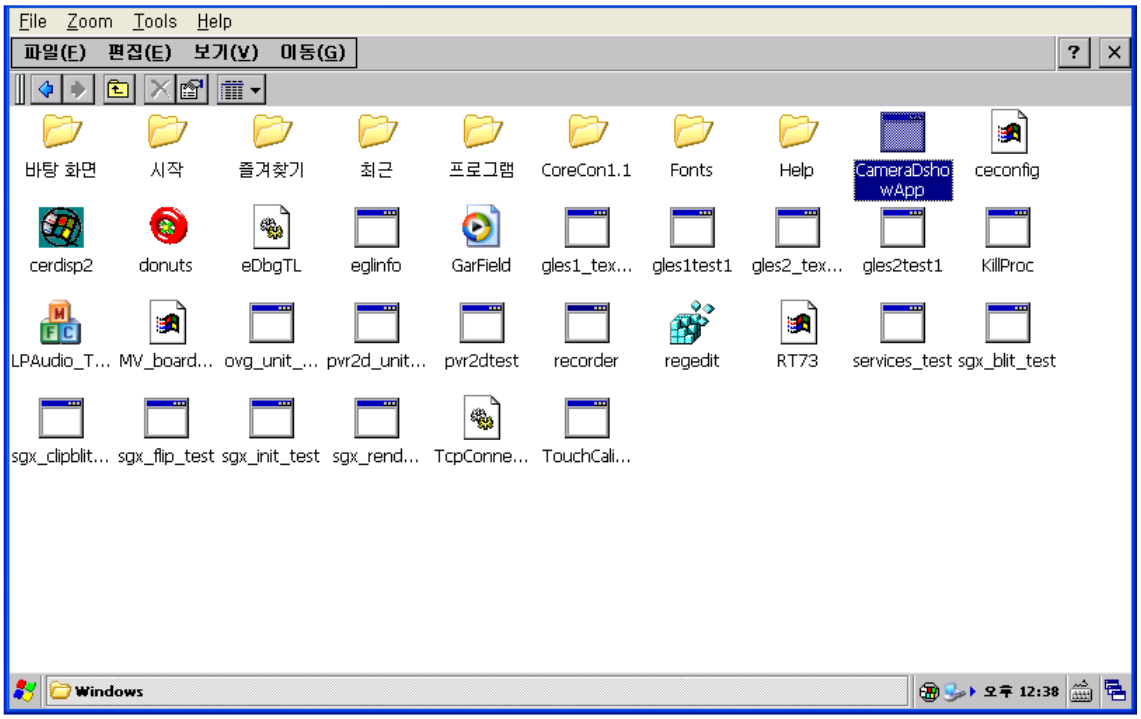
Once Windows CE has booted, click “My Device”.



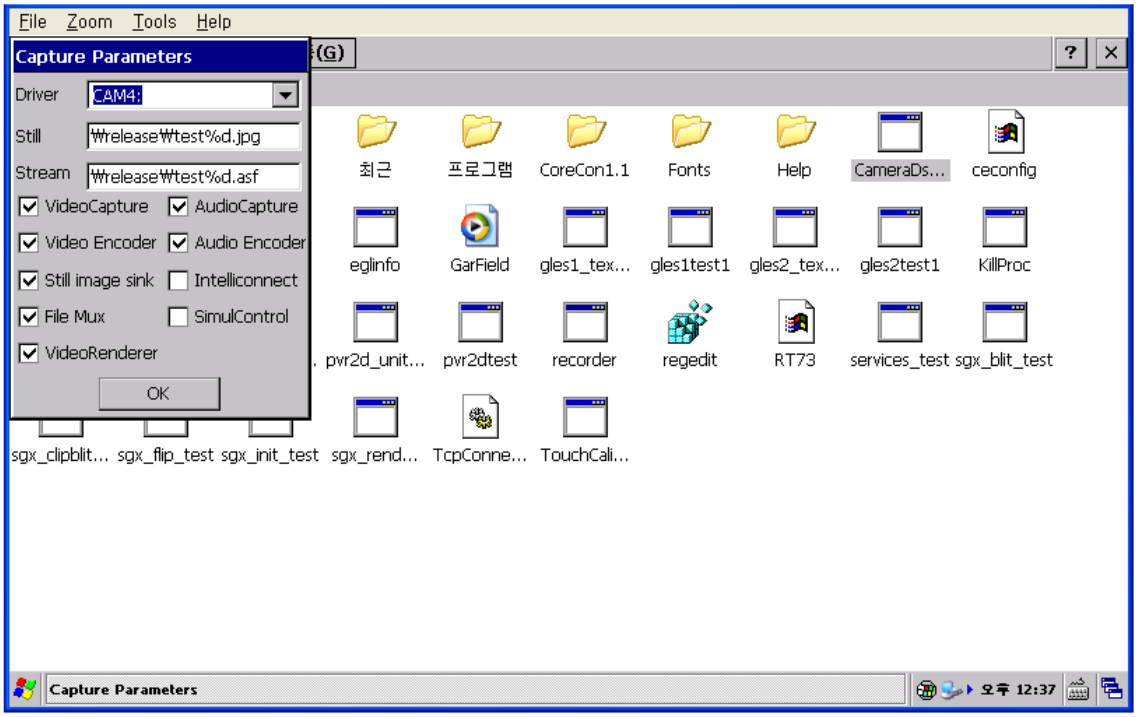
Click Windows Folder.



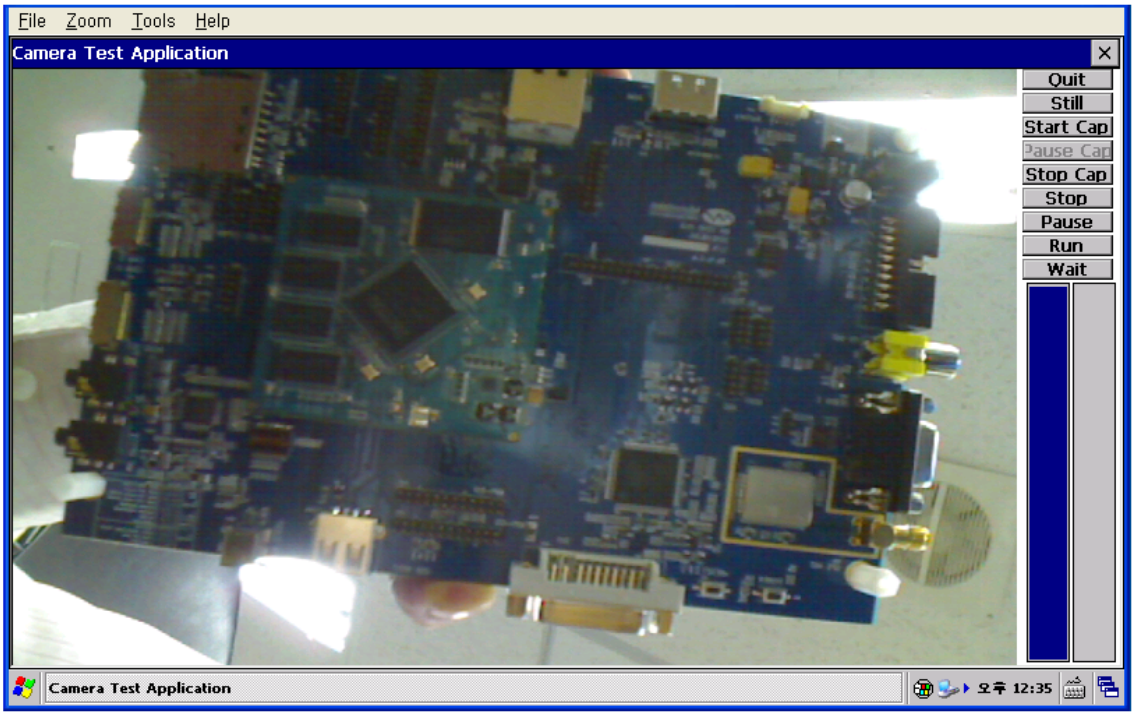
Run CameraDshowApp.



Click “OK”



Camera View



6. Bi-directional Communication using Camera

This program allows the user to preview the image by connecting the camera to each board and outputting the image from the other side via Ethernet or WiFi. Codec is used for H.264 and the image can be saved in H.264 AVI format.

6.1 File Descriptions

Binary File

(1) EncMux.dll

:DirectShow Filter that gets Vidio/Audio source on the DirectShow Capture port and performs H.264 video encoding, G711 Audio Encoding

(2) MfcLib.lib

:MFC H/W Encoding/Decoding Library such as H.264/MPEG4/XVID

(3) DispLib.lib

:Display Library that displays on the LCD via the video buffer

(4) G711Lib.lib

: G.711 uLaw 8Khz Encoding/Decoding Library

(5) WavOutLib.lib

: Audio output library to output audio PCM data to the speaker

(6) NetTest.exe

: Application program controlling video communications such as transmitter/receiver, network module, or capture/renderer filter

File description for each Module

EncMux.dll		
EncMux\	EncMux.h	CLSID or I/F for EncMux filter header
	EncMux.dll	DirectShow filter

MfcLib.lib		
MfcLib\	Inc\MfcLib.h	Declare MFC Encoding/Decoding class and structure
	Lib\MfcLibD.lib	MfcLib Library for debugging
	Lib\MfcLib.lib	MfcLib Library for release

DispLib.lib		
DispLib\	Inc\DispLib.h	Display Class Declaration
	Lib\DispLibD.lib	DispLib Library for debugging
	Lib\DispLib.lib	DispLib Library for release
	DispLib\DispLib.cpp	Display Class Source Code
	SVEDriverAPI.h	Device IO Map and Type declare header

G711Lib.lib		
G711Lib\	Inc\G711Lib.h	G711 SW Encoding/Decoding API header
	Lib\G711LibD.lib	G711 Codec Library for debugging
	Lib\G711Lib.lib	G711 Codec Library for release

WavOutLib.lib		
WavOutLib\	Inc\WavOut.h	Audio Output Class Declaration
	Lib\WavOutLibD.lib	WavOutLib Library for debugging
	Lib\WavOutLib.lib	WavOutLib Library for release
	WavOut.cpp	Audio Output Class Source Code

NetTest.exe		
NetTest\	CPropertyBag.cpp/h	COM Property Qwerty Class Source
	DShowCam.cpp/h	DirectShow Capture/Preview Source
	NetTest.cpp/h	Video Chat Application Source Code
	NetTestDlg.cpp/h	Video Chat Window Source Code
	Utils.cpp/h	Utility Source Code

Major Function Description

EncMux Filter

EncMux.h
STDMETHODIMP set_HWND(OAHWND hParWnd); ▶ Function indicating the handle of parent window with filter. Parent handle definition before hParWnd : Parent Window's handle
STDMETHODIMP set_EncodeControl(UINT uEncCtrl); ▶ Function Enable/Disable the Encoding - uEncCtrl : When sending network data uEncCtrl=1 Enable, uEncCtrl=0 Disable

MfcLib Library

MfcLib.h
eMFCLIB_RET_CODE Open(eMFCLIB_CODEC eCodec); ▶ MFC H/W Encoder or Decoder Device Assigned Function. - MFC_ENC MFC_UNKNOWN_TYPE = 0x0, - MFC_COM_CTRL = 0x1, - MFC_MPEG4_ENC = 0x100, - MFC_H263_ENC, - MFC_H264_ENC, - MFC_MPEG4_DEC = 0x200, - MFC_H264_DEC, - MFC_H263_DEC, - MFC_MPEG2_DEC, - MFC_DIVX_DEC, - MFC_XVID_DEC, - MFC_VC1_DEC
eMFCLIB_RET_CODE InitEnc(void* pEncParam); ▶ Function initializing the encoder parameter. - MFC_ENC_H264_PARAM : H.264 Encoding Parameter Structure - MFC_ENC_H263MPEG4_PARAM: MPEG4/H263 Encoding Parameter Structure
eMFCLIB_RET_CODE InitDec(LPBYTE pRawStrm, INT iStrmLen, INT iNumExtFrmBuf=5, INT iFrmDelay=0); ▶ Function for Decoder Initialization - pRawStrm : Stream Buffer Pointer - iStrmLen : Stream Size in the Stream Buffer - iNumExtFrmBuf: Extra Frame Buffer Size -

iFrmDelay: Frame Delay (0~16)
eMFCLIB_RET_CODE Decode(LPBYTE* ppY, LPBYTE* ppCbCr, LPBYTE pRawStrm, INT iStrmLen, INT iLast, UINT uOutType); ▶ Output Function of Decoded Image(Y,Cb,Cr) from Video Stream. - ppY, ppCbCr: Y,Cb,Cr Decoding Image Buffer Pointer - pRawStrm: Input Video Stream Buffer Pointer - iStrmLen: Input Video Stream Buffer Size - iLast: Last Stream Flag (Last:1 or 0) - uOutType: Output Decoding Image Format
eMFCLIB_RET_CODE Decode(LPBYTE* ppY, LPBYTE* ppCbCr, UINT uOutType); ▶ Output Function of Decoded Image(Y,Cb,Cr) in the Frame Buffer - ppY, ppCbCr: Y,Cb,Cr Decoding Image Buffer Pointer - uOutType: Output Decoding Image Format
eMFCLIB_RET_CODE Encode(LPBYTE pYuvBuf, LPBYTE pRawStrm, UINT* puStrmLen, UINT uInType); ▶ Output Function of Encoding Stream from Y,Cb,Cr Image - pYuvBuf: Y,Cb,Cr Input Image Buffer Pointer - pRawStrm: Encoded Stream Buffer Pointer - puStrmLen: Encoded Stream Size - uInType: Input Image Format
eMFCLIB_RET_CODE Close(); ▶ Function Clearing MFC Instance Assigned

DispLib Library

DispLib.cpp/h
HRESULT Open(); ▶ Function Allocating Display Device
HRESULT Draw(LPBYTE pY, LPBYTE pCb, LPBYTE pCr, int iX, int iY, int iW, int iH, int iSrcType); ▶ Output Function displaying Y,Cb,Cr Image Buffer - pY,pCb,pCr: Y,Cb,Cr Input Image Buffer Pointer - iX, iY: Screen Axis for Upper Left Image - iW, iH: Size(Horizontal, Vertical) of the Output Image - iSrcType: Input Image Buffer Format FRMSRC_RGB16 = 0, FRMSRC_RGB565 = 0, FRMSRC_RGB24, FRMSRC_YUV420, FRMSRC_YUV422_YCBYCR,

FRMSRC_YUV422_CBYCRY, FRMSRC_YUV422_YCRYCB, FRMSRC_YUV422_CRYCBY, FRMSRC_NV12
HRESULT Close(); ▶ Function of Clearing the Display Device.

G711Lib Library

DShowCam.cpp
BOOL G711Encode(LPBYTE pStrmBuf, UINT* puStrmSz, LPBYTE pPcmBuf, UINT uPcmSz); ▶ Encoding Function of G711 uLaw Stream from PCM Audio Data - pStrmBuf: Encoded G.711 uLaw Stream Buffer Pointer - pPcmBuf: PCM Audio Buffer Pointer for Encoding
BOOL G711Decode(LPBYTE pPcmBuf, UINT* pPcmSz, LPBYTE pStrmBuf, UINT uStrmSz); ▶ Decoding Function of PCM Audio Data from G.711 uLaw Stream - pPcmBuf: Decoded PCM Audio Buffer Pointer - pStrmBuf: G.711 uLaw Stream Buffer Pointer for Decoding - uStrmSz: Stream Buffer Size for Decoding

WavOutLib Library

WavOutLib.cpp/h
HRESULT Open(HWND hWnd, UINT uSamplesPerSec, UINT uBitPerSample, UINT uChannels); ▶ Function Initializing Audio Output Device - hWnd: Handle of Parent Window for Message - uSamplesPerSec: Audio PCM Samples per Second - uBitPerSample: Bits of Audio PCM Samples - uChannels: Number of Audio Channel
HRESULT PutData(WPARAM wParam, LPARAM lParam, LPBYTE pBuf, UINT uBufSz); ▶ Output Function for Audio from PCM Data - wParam: WPARAM Message Parameter - lParam: WAVEHDR Structure Pointer - pBuf: Input Audio PCM Buffer Pointer -

uBufSz: Input Audio PCM Buffer Size
HRESULT Stop(); ▶ Function Stopping the Audio Output
HRESULT Close(); ▶ Function Closing Audio Output Device

NetTest Application

NetTestDlg.cpp/h
void OnBnClickedOpenDev(); ▶ Function initializing the modules for video communication - MFC Decoder Initialization - Display Devie Initialization – Camera and MIC Device Initialization - Network UDP Tx/Rx Module Initialization, Video Port(200), Audio Port(300) – Start Camera Preview
void OnBnClickedStart()/OnBnClickedStop(); ▶ Function Transmitting and Stopping the Network Stream by Encoding the Input Stream - ECTRL_RUN/ECTRL_STOP Control Encoding of the EncMux.dll
void OnBnClickedCloseDev(); ▶ Function Clearing the Module Device and Library, which is Initialized
void OnBnClickedExit(); ▶ Function for Finishing Program
LRESULT OnSendVideoStream(WPARAM wParam, LPARAM lParam); ▶ Function transmitting Encoded Video Stream via Network – Encoded Stream at EncMux will be transmitted to the Application by using the message WM_SEND_VIDEO_STREAM - wParam: Encoded Stream Buffer Pointer - lParam: Encoded Stream Buffer Size
LRESULT OnSendAudioStream(WPARAM wParam, LPARAM lParam); ▶ Function Transmitting the Encoded Audio Stream to the Network – Encoded Stream at EncMux will be transmitted to the Application by using the message of WM_SEND_AUDIO _STREAM - wParam: Encoded Stream Buffer Pointer - lParam: Encoded Stream Buffer Size
LRESULT OnRcvVideoStream(WPARAM wParam, LPARAM lParam); ▶ Output Function displaying video stream by network after decoding – Decoding

the received Video Stream by MFC and display it by using DisLib - wParam: Received Video Stream Buffer Pointer - lParam: Received Video Stream Buffer Size
LRESULT OnRcvAudioStream(WPARAM wParam, LPARAM lParam); ▶ Output Function displaying the audio stream received by network after decoding - After decoding the received audio stream by using G711 library, Output to Speaker by WavOutLib - wParam: Received Audio Stream Buffer Pointer - lParam: Received Audio Stream Buffer Size
LRESULT OnWavOutOpen(WPARAM wParam, LPARAM lParam); ▶ Function calling back when Audio Device is initialized
LRESULT OnWavOutData(WPARAM wParam, LPARAM lParam); ▶ Function setting the output buffer for Audio PCM data. – Call WavOutLib by network and input the decoded PCM data
LRESULT OnWavOutClose(WPARAM wParam, LPARAM lParam); ▶ Function calling back when Audio Device is cleared
HRESULT CNet::CreateSocket(UINT uPort); ▶ Function generating the Network Socket - uPort: Network Port Number
HRESULT CNet::ConnectTo(PTCHAR szIPAddr); ▶ Function to connect the network terminal – IP Address connected
HRESULT CNet::SendData(LPBYTE pDat, UINT uSz); ▶ Function Receiving Data through Network - pDat: Transmit Data Buffer Pointer - uSz: Transmit Data Buffer Size
void CNet::OnReceive(int nErrorCode) ▶ Function Receiving Data through Network
void CNet::DestroySocket(); ▶ 네트워크 소켓을 해제하는 함수

DShowCam.cpp/h
BOOL CreateCam(HWND hWnd); ▶ Function of the Filter Graph for DirectShow.- hWnd: Handle of Generated Parent Window
void SetVideoWindowSize(LPRECT rt); ▶ Function of Screen Size and Location of Video Renderer - rt: Video Screen Location/Size
void DestroyCam(); ▶ Function Clearing the Filter Graph of the DirectShow

void Start()/Run()/Stop()/Pause(); ▶ Function Run, Stop or Pause of the Filter Graph

void EnAudio(BOOL enAudio);
▶ Function of Audio Graph when Filter Graph is generated (enAudio=1 Gererated)
void EnPreview(BOOL enPreview);
▶ Function of Video Preview Graph when Filter Graph is generated (enPreview=1 Gererated)
void EnVideo(BOOL enVideo);
▶ Function of Video Capture Graph when Filter Graph is generated (enVideo=1 Gererated)

void SetPrvMode(int iIdx); ▶ Function Setting the Video Preview Mode - iIdx: Video Preview Mode Index Void SetCapMode(int iIdx); ▶ Function setting the Video Capture Mode - iIdx: Video Capture Mode Index
int GetPrvMode(); ▶ Function Returning the current Preview Mode ▶ int GetCapMode(); Function Returning the current Video Capture Mode
void SetEncMuxEncCtrl(UINT uEncCtrl); ▶ Function Start/Stop the Encoding the A/V Capture Source by EncMux - uEncCtrl : Encoding Start(ECTRL_RUN=1), Encoding Stop(ECTRL_STOP=100)
HRESULT SetAudioSrcMode(IBaseFilter* pAudioCaptureFilter, int iIdx); ▶ Function setting Audio Capture Mode when the Filter Graph is generated - iIdx: Audio Capture Mode Index

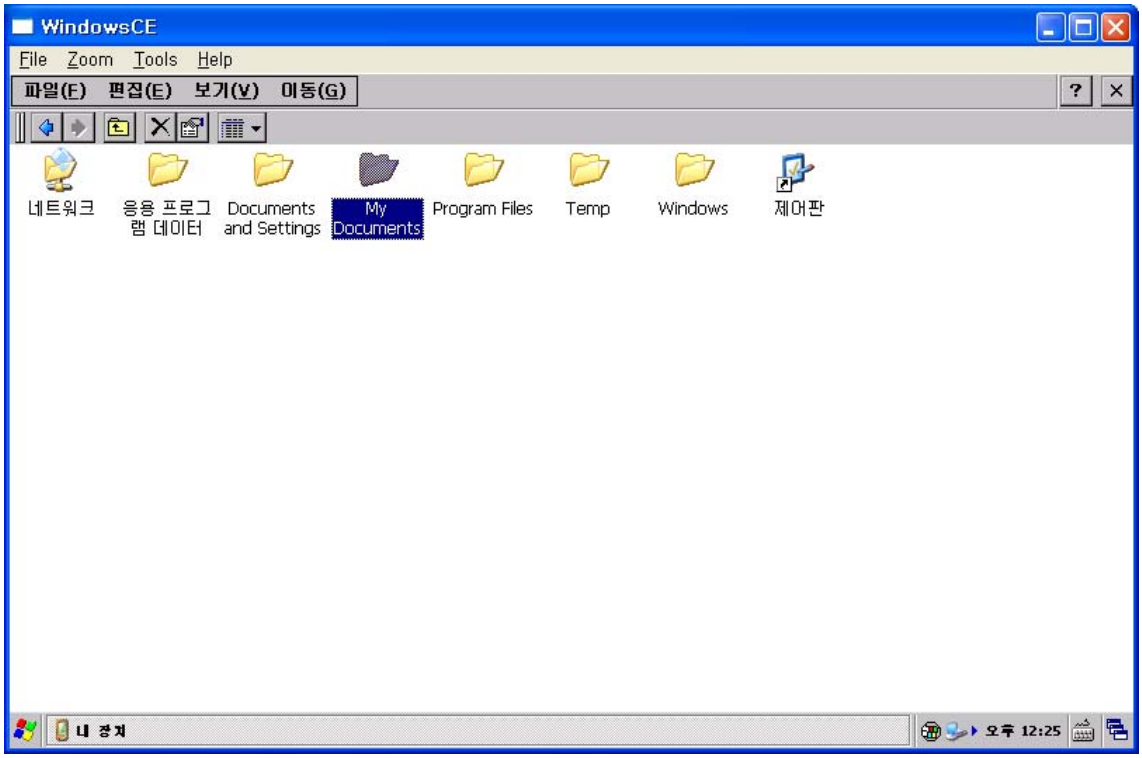
CPropertyBag.cpp/h
HRESULT Read(LPCOLESTR pszPropName, VARIANT *pVar, IErrorLog *pErrorLog); ▶ Function Reading Property Value of COM Object - pszPropName: Property Name Reading - pVar: Pointer of the Property Value Reading
HRESULT Write(LPCOLESTR pszPropName, VARIANT *pVar); ▶ Function Changing the Property Value of COM Object - pszPropName: Properly Name Changing - pVar: Property Value for a change

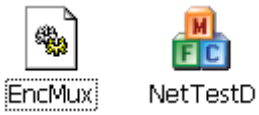
Utils.cpp/h
HRESULT GetPin(IBaseFilter *pFilter, PIN_DIRECTION PinDir, IPin **ppPin); ▶ Function Returning the Interface Pointer on Input or Output Pin of the Filter - pFilter : Filter Pointer Returning the Pin Interface - PinDir : Pin Direction (Input or Output) - ppPin : Returning Pin Interface Pointer
void RegSvr(WCHAR *wzDll); ▶ Function Registering EncMux/NetSrcFlt/G711uLawDec Filter to the Registry - wzDll : Path name of the DirectShow Filter to be registered

6.2 Camera Operation

Instructions for building, installing, and compiling SDK are described in the MV-V210_WindowsCE_Guide.pdf at the directory of \DOC\WinCE in the CD.
Please refer to pages 24-29.

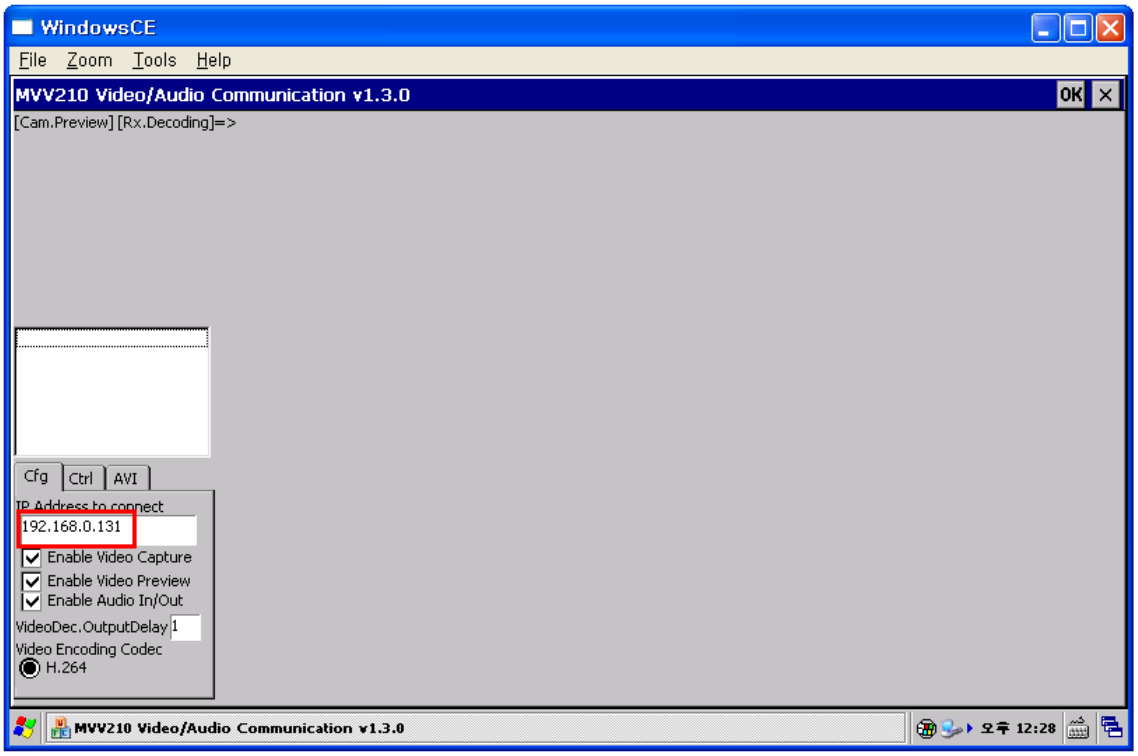
Once the compilation has ended, the following files will be generated. Copy EncMux.dll, NetTest.exe to the SD card or copy them to “My Documents” using Active Sync.



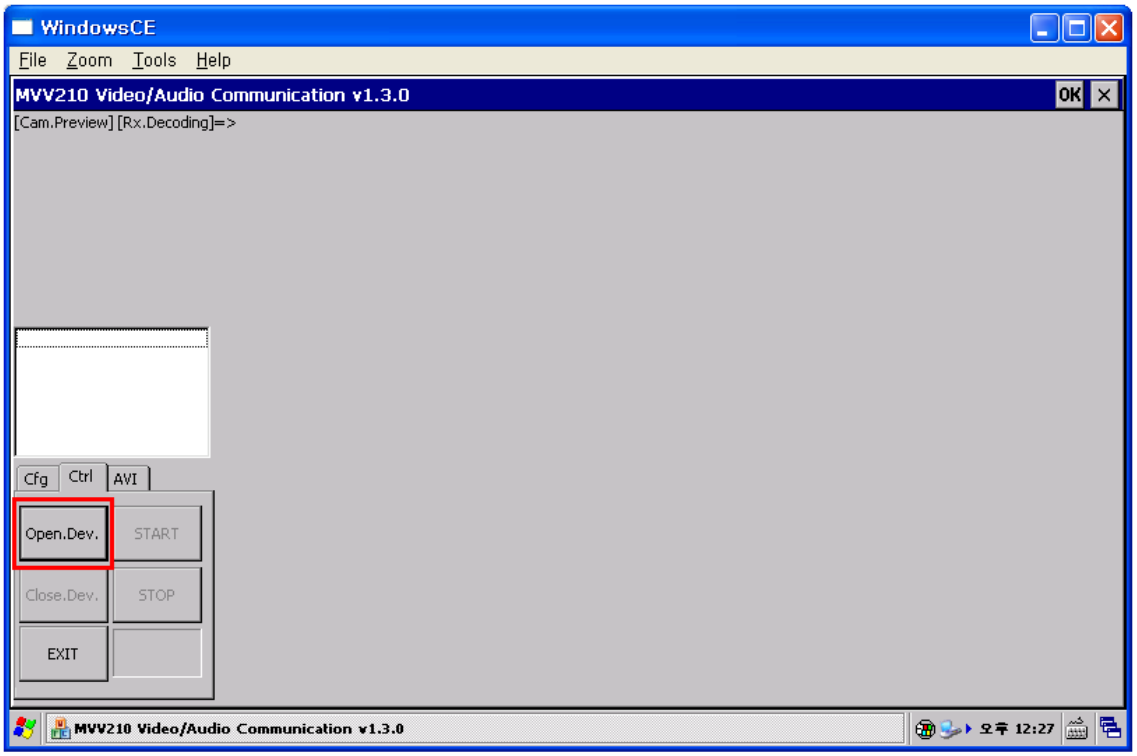


Connect the camera module to both boards and connect the boards with Ethernet or WiFi.

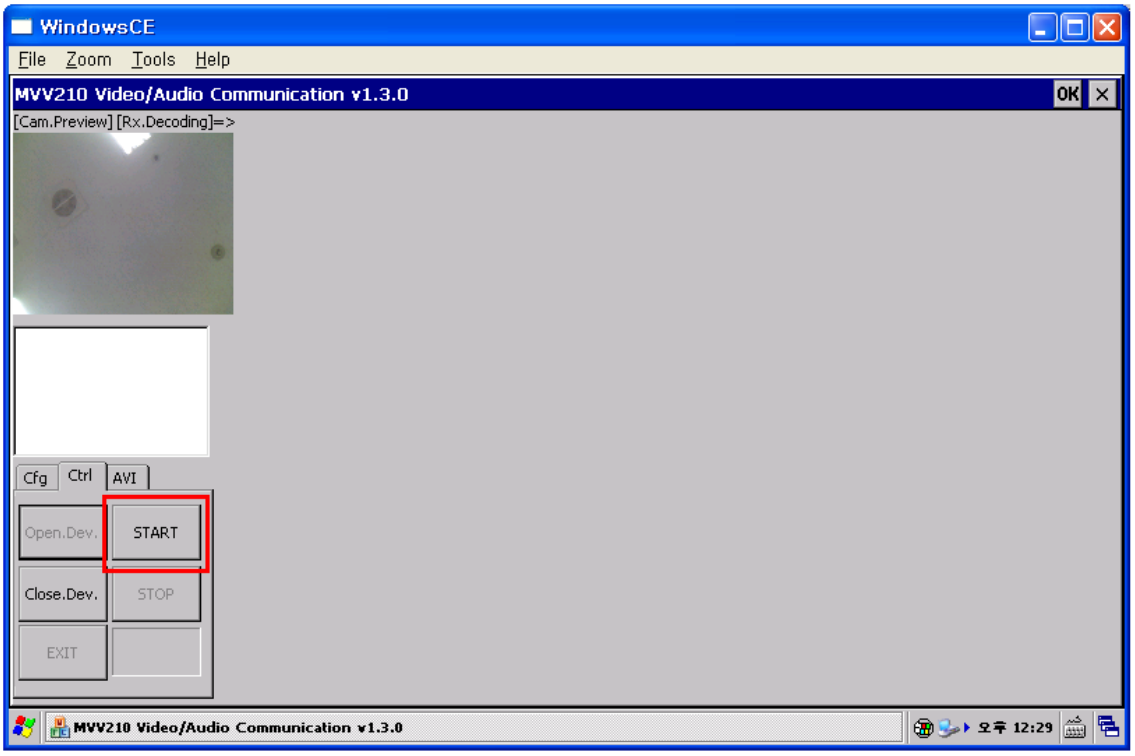
Execute the NetTest.exe file and type the IP address for the other side if the screen comes up.

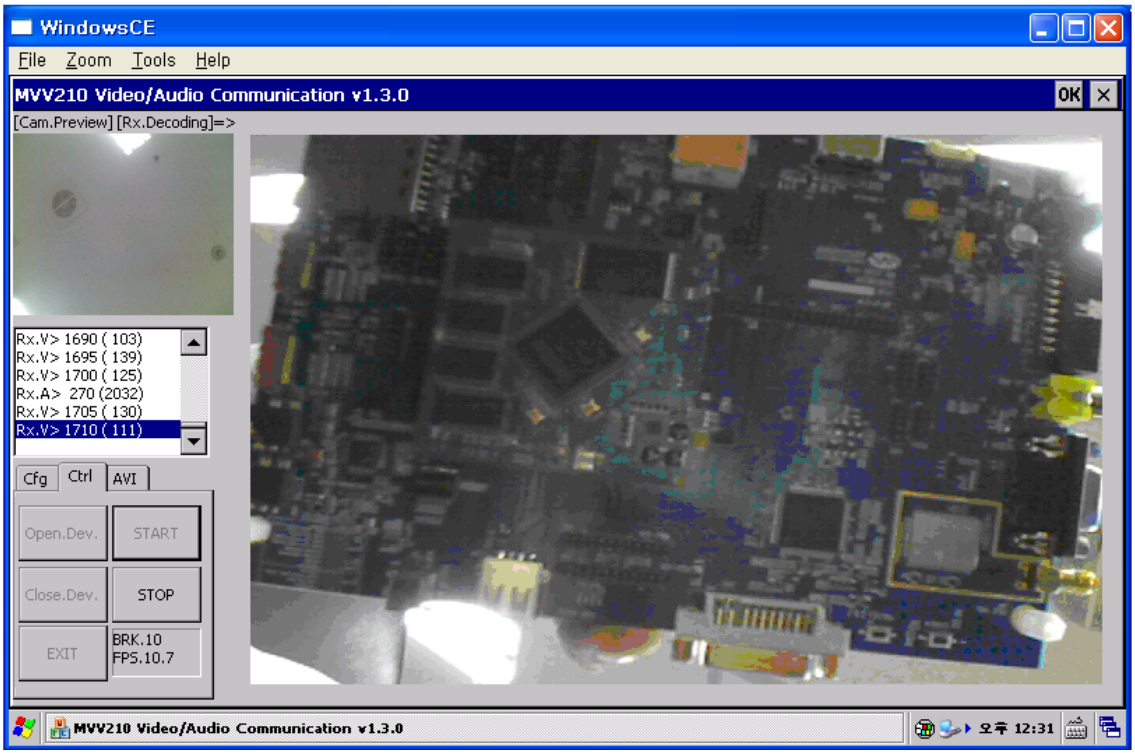


Click Open.Dev for both boards



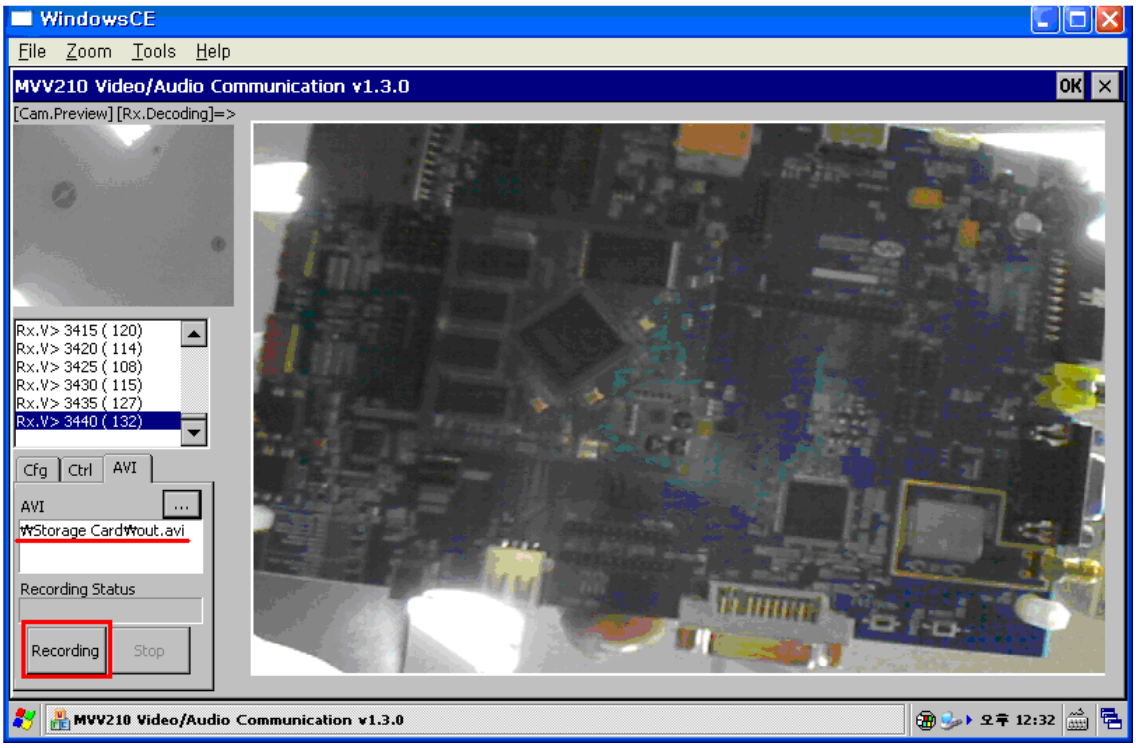
If the Preview Screen comes up, as shown below, click START for both boards.



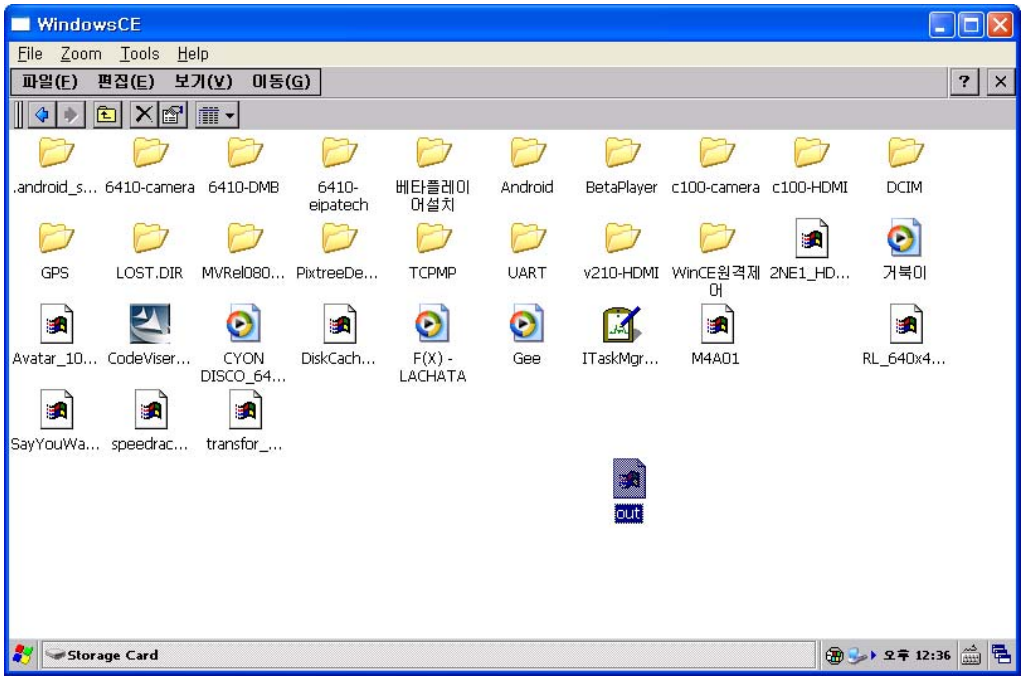


To save the movie clip, click Recording.

The saving pathway is \Storage Card\out.avi. (Saved in SD card with the name of out.avi.)



Stored



Save this file to the PC and open it with GOM player to see the movie. Note that the voice needs a codec (ffdshow_rev3529_20100811_clsid.exe). After installation, the option should be set to play the voice.