

# **1. Overview**

## **1.1 Scope**

This document describes how to use. NET Micro Framework (NETMF) with SH7264 M3A-HS64G50 board. This document mainly describes how to setup, build and run SH7264\_M3A\_HS64 solution available in the porting kit. This document also describes how to create, build, deploy and run NETMF based applications on SH7264 M3A-HS64G50 board using Microsoft visual studio.

## 1.2 Required Target System

- M3A-HS64G50

- 

- M3A-HS64G01

M3A-HS64G02 is also fine but the GPIO port name for Virtual Key input assignment is different.

- LCD panel which part number is TX09D55VM1CDA

If you have another LCD panel, you might need to modify the display driver.

Please go through the hardware user manual of the M3A-HS64G50 and M3A-HS64G01 before power ON and working with the board.
---

### **1.3 Supported Driver**

- Display (LCD)
- USB Host only for File System
- Key input
- Serial
- Timer & Power
- Nor Flash
- Serial Flash

In addition, the SD card driver for File System is already supported and we may be having other drivers. If needed, please contact us.

## **1.4 Supported Projects**

Following projects are supported.

- NativeSample
- TinyCLR

Following projects are not supported.

- Portbooter
- Tinybooter

However, these projects can be build properly so if needed you can customize them.

## **1.5 Requirements**

- Host system (Windows XP in this documentation)
- Target System (M3A-HS64G50 and M3A-HS64G01 Board)
- Microsoft Visual Studio 2010
- E10A-USB Emulator for SH2A and installation setup.
- C/C++ Compiler Package for SuperH RISC engine family
- .NET Micro framework porting kit (MicroFrameworkPK.msi)
- .NET Micro framework development kit (MicroFrameworkSDK.MSI)
- .NETMF Solution “SH7264\_M3A\_HS64” (included in the porting kit)
- NULL modem cable (Serial cable)
- Tera Term Pro or another serial program.

## **2. Setup**

### **2.1 Preparation**

Before setting up Porting Kit, below software have to be installed.

- Visual Studio 2010
- Renesas C/C++ Compiler Package for SuperH RISC engine family V.9.02 Release 00 or later

For these instructions, we will assume it is installed in

`c:\Program Files\Renesas\Hew\Tools\Renesas\Sh\x_y_z*`

## **2.2      Install .NET Micro Framework Porting Kit**

Install porting Kit by executing MicroFrameworkPK.msi.

## 2.3 Set Path

- 1) Open a command-line prompt
- 2) Change the current directory to  
C:\MicroFrameworkPK\_v4\_1
- 3) Enter below,  
setenv\_shc "c:\Program Files\Renesas\Hew\Tools\Renesas\Sh\x\_y\_z\*"

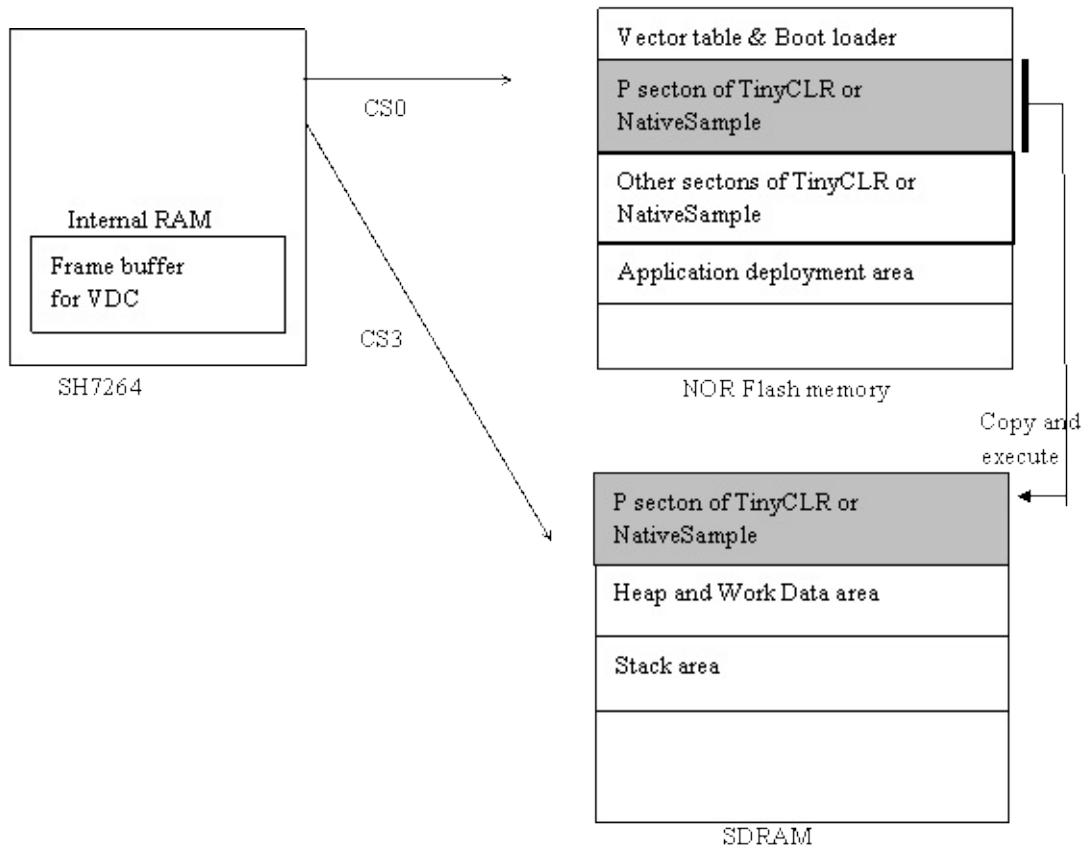
\*x\_y\_z is the version of the compiler being used for example 9\_2\_0.



### **3. Memory Map**

The memory map of SH7264 M3A-HS64G50 .NETMF Porting kit is shown below. Two types of memory configurations are supported.

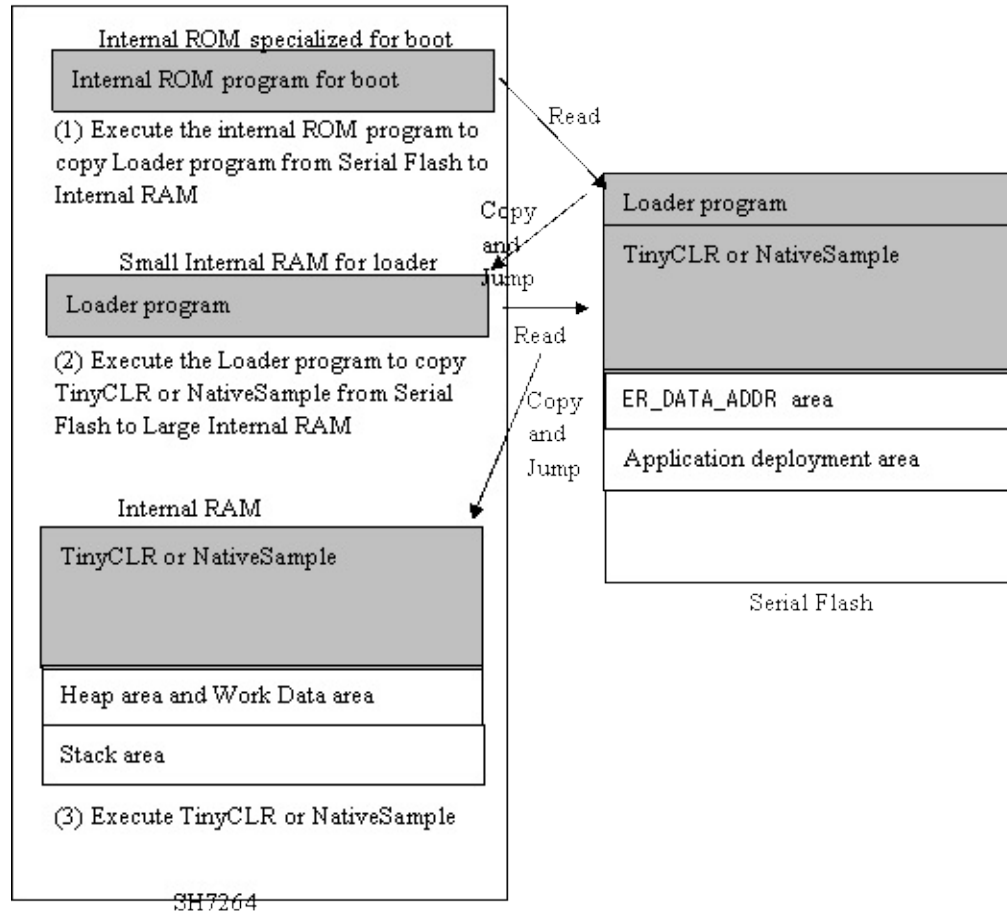
### 3.1 For full functions with external NOR Flash memory and SDRAM



**Figure 3.1 The memory map of SH7264 M3A-HS64G50 .NETMF Porting Kit - 1**

In this case, the memory size for TinyCLR can be large so full features can be included in TinyCLR. In addition, the big heap area can be allocated so it can keep large frame buffer for LCD display.

## 3.2 For limited functions with Low Cost Serial Flash memory and Internal RAM



**Figure 3.2 The memory map of SH7264 M3A-HS64G50 .NETMF Porting Kit - 2**

In this case, the TinyCLR and Work Data area are located in Internal RAM area so the external SDRAM is not required. However, Internal RAM is smaller than external SDRAM so features in TinyCLR are limited.

Notification: In the initial setting, Heap area is located in SDRAM.

## **4. The case of using full functions with external NOR Flash Memory and SDRAM**

In this section, we shall describe the way to build, download and execute SH7264 M3A-HS64G50 solution available in the porting kit with NOR Flash Memory and SDRAM.

## 4.1 How to Build

1) Using Command-line prompt, change the directory to  
“Solutions\SH7264\_M3A\_HS64” C:\MicroFrameworkPK\_v4\_1>cd  
solutions\SH7264\_M3A\_HS64

2) Run

**Msbuild dotnetmf.proj /t:build /p:flavor=debug;SERIALFLASH=false**

Flavor: <debug|release|rtm> SERIALFLASH : <ture|false>

If you want to debug your program using E10A-USB Emulator, please specify “debug” for “flavor” option.

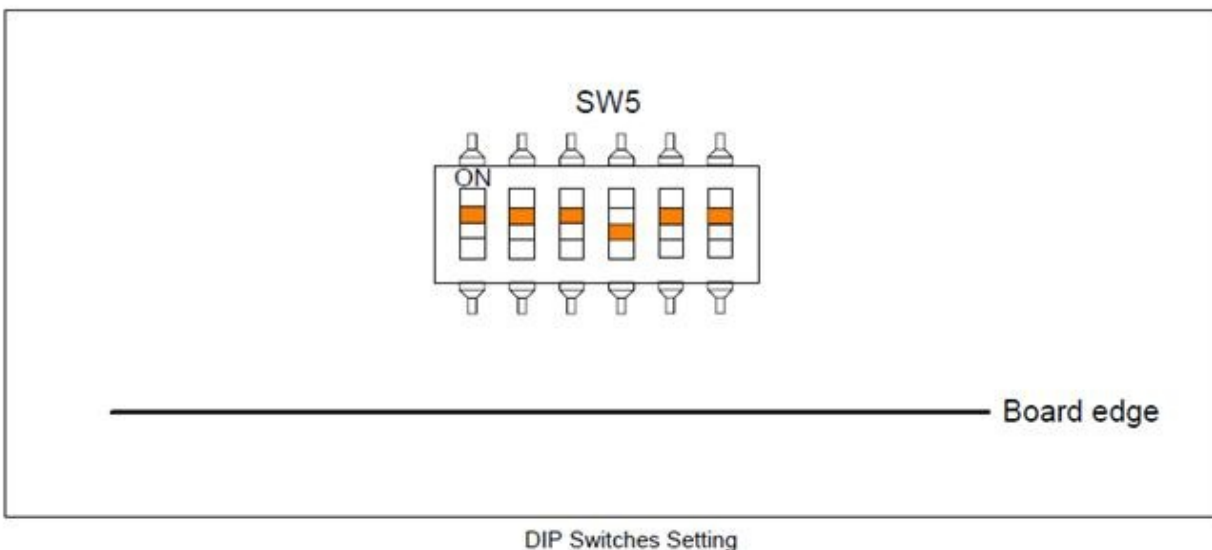
In this case, specify SERIALFLASH to “false”.

## 4.2 Board switch settings

There are two DIP switches located on the CPU board. The function of each switch and its connection are shown below:

### 1) SW5

Set SW5 of the M3A-HS64G50 board as below to startup from NOR Flash.



Mode Setting DIP Switches Setting (SW5)			
No.	Function	Default	Description
SW5-1	Clock operation mode	ON	Clock operation mode 2
SW5-2	Boot mode	ON	Boots from the memory which is connected to the CS0 space (boot mode 0)
SW5-3	Boot mode	ON	
SW5-4	Connection	OFF	Connected to the SDRAM
SW5-5	Connection	ON	Connected to the RSPI
SW5-6	Connection	ON	Connected to the expansion connector

**Figure 4.1 SW5 settings**

### 2) SW6

Regarding to the SW6, set all switches should be OFF.





## 4.3 Download using E10A-USB

### 1) Setup the E10A-USB Emulator

Install the E10A-USB Emulator software into your PC.

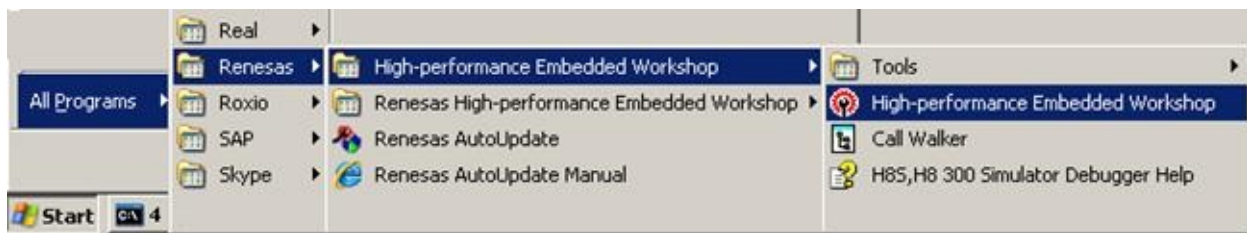
During install, you should select the device group for E10A-USB then specify “Super H RISC engine family SH-2A device group”

### 2) How to start downloading using E10A-USB

Extract the HEW workspace for SH7264 M3A-HS64G50 Porting Kit from file SH7264M3AHS64.zip at any place in your PC.

In the following instructions, we will assume it is installed in “c:\workspace”

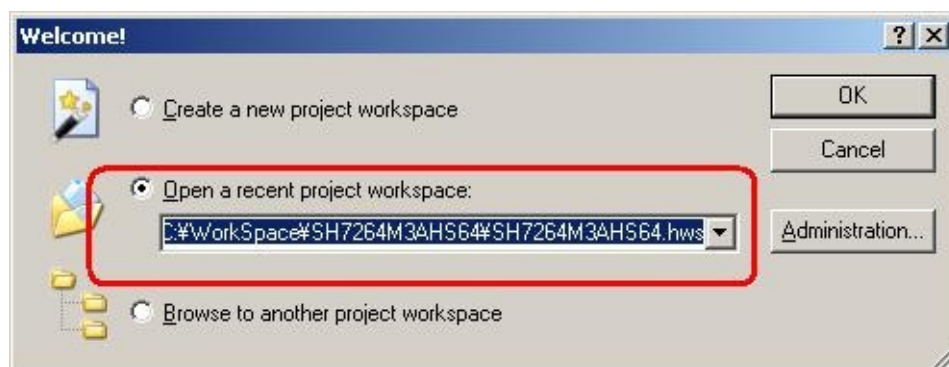
Start the Hew with choosing below menu



**Figure 4.1 Hew menu**

Then Hew will be show up and you can see below dialog box.

Specify the workspace as below, and press OK button.



**Figure 4.2 Welcome dialog box**



In below dialog box, set options like below.



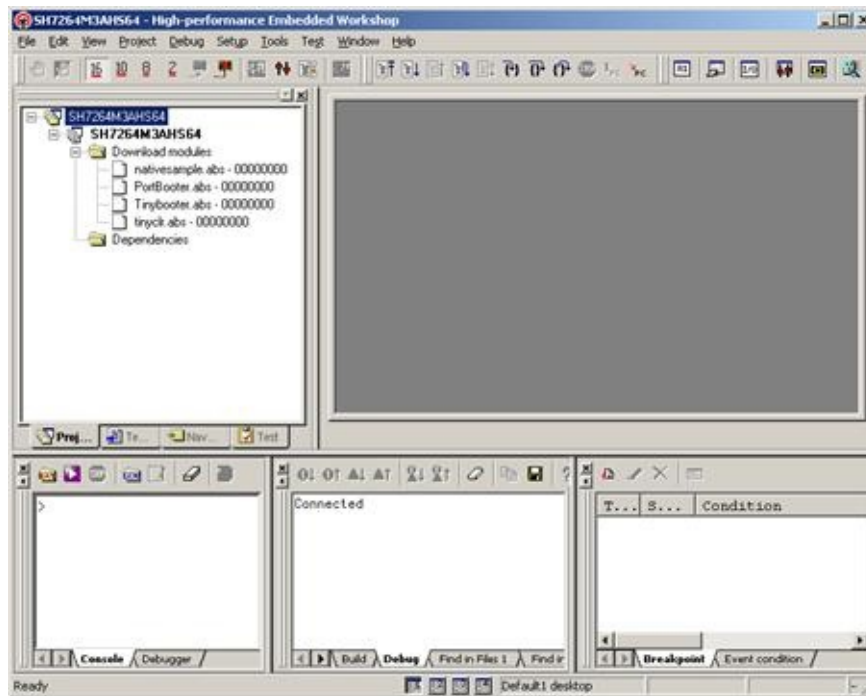
**Figure 4.3 Select Emulator mode dialog box**

When below message shows up, reset the target board by pressing the reset button on the board and then press OK button.



**Figure 4.4 Heush2acustom1 dialog box**

Then Hew/E10A-USB will finish connection with SH7264.

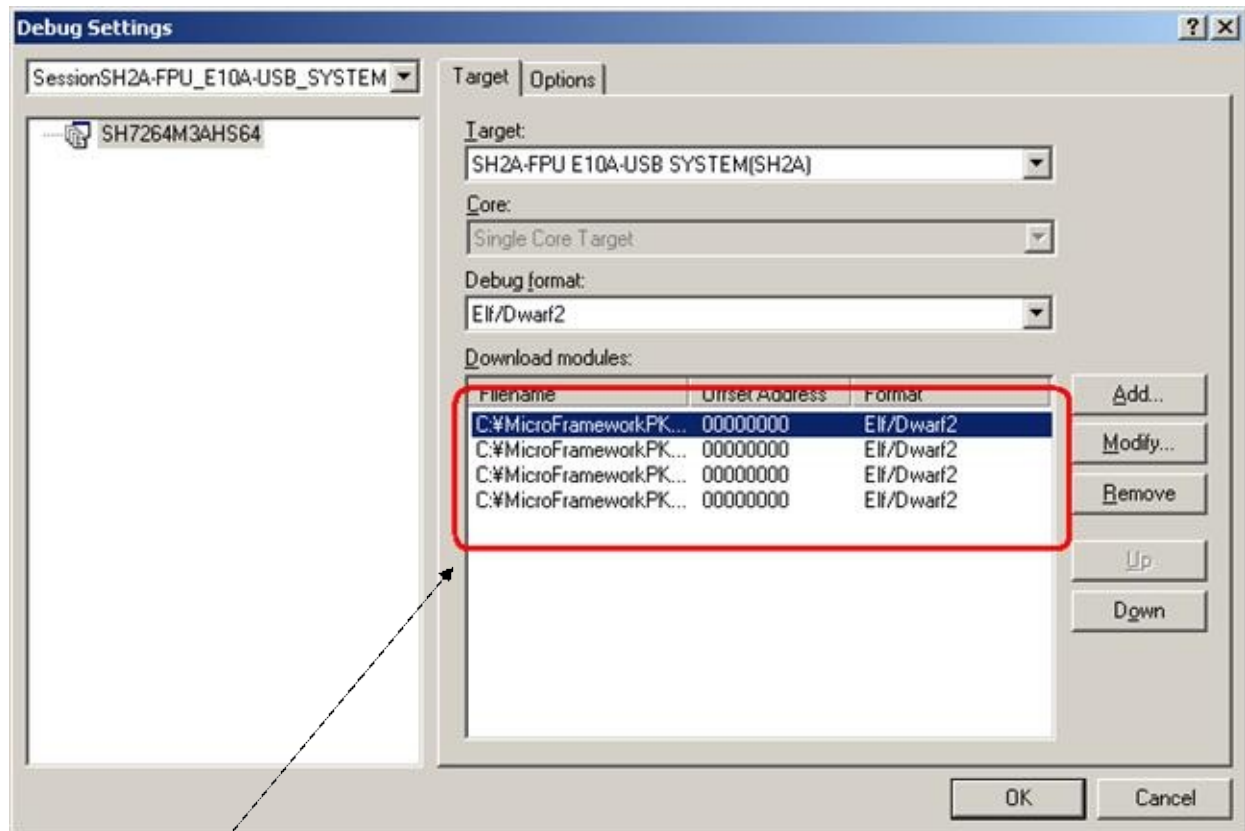


**Figure 4.5 Hew window**

3) How to download the program to the flash memory

i) Prepare the download module.

Select the [Debug] -> [Debug Settings...] from the menu bar of Hew then below dialog box will show up.



**Figure 4.6 Debug Settings dialog box**

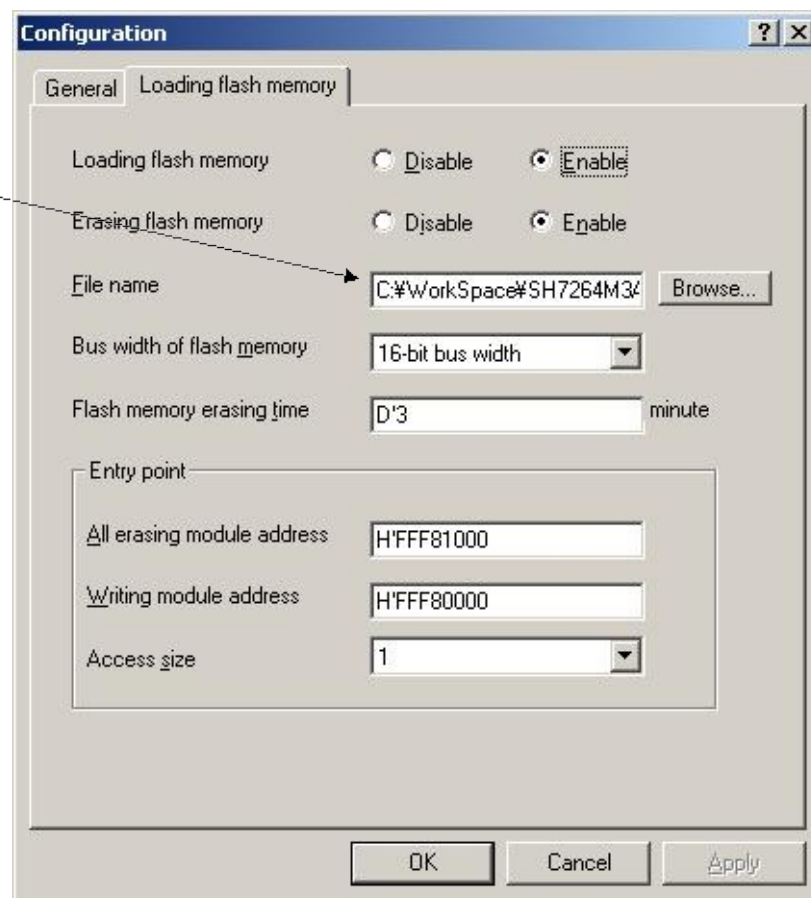
Here is the definition of download modules. Please change the Path setting for each download module by clicking “Modify” button.

ii) Prepare for downloading to Flash Memory.

Select the [Setup] -> [Emulator] -> [System...] from the menu bar of Hew then “Configuration” dialog box will show up.

Select “Loading flash memory” tab. Then, specify as same as below figure.

For “File name”, please specify <Workspace folder>\SH7264M3AHS64\fmttool\_hs64\fmttool\_hs64.mot.



**Figure 4.7 Configuration dialog box**

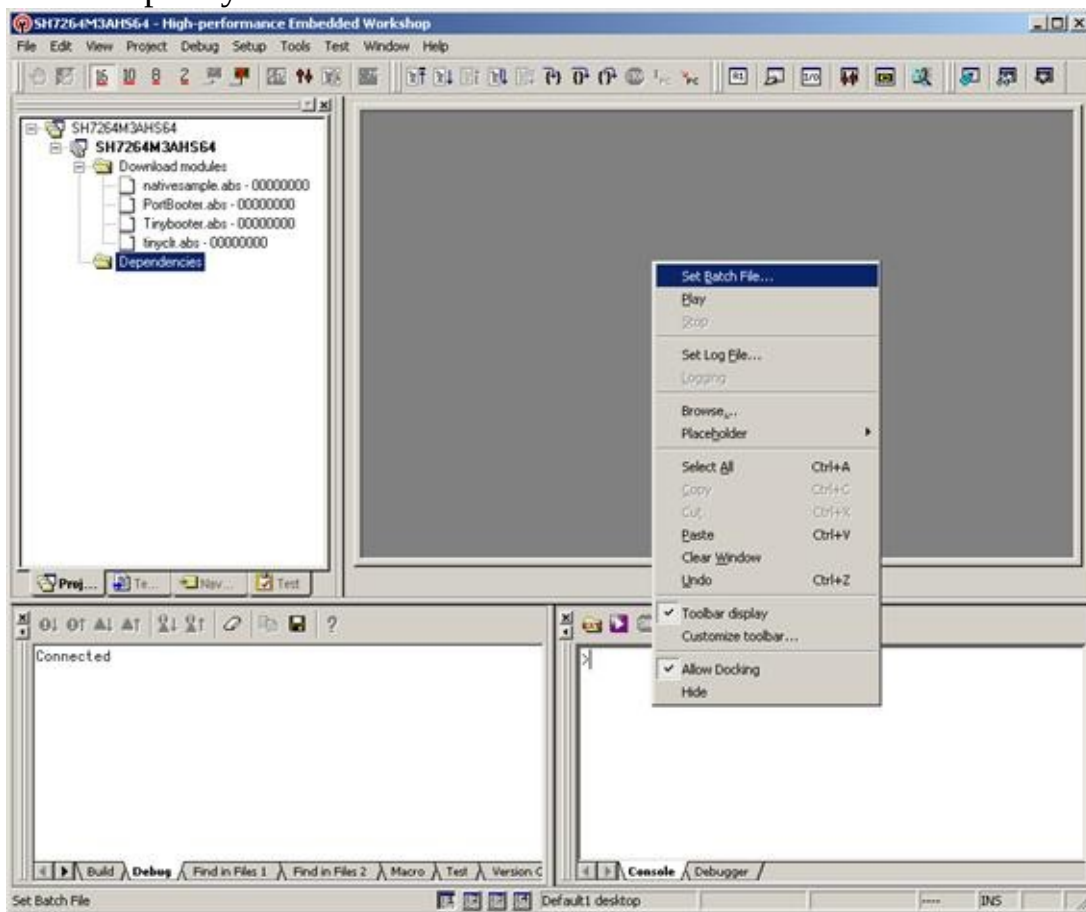
### iii) CPU and memory initialization

In order to download data to flash properly, CPU and memory should be initialized as following.

- CPU : Cache must be disabled.
- Memory : Port controller must be initialized.

In order to initialize them, run the batch file as below,

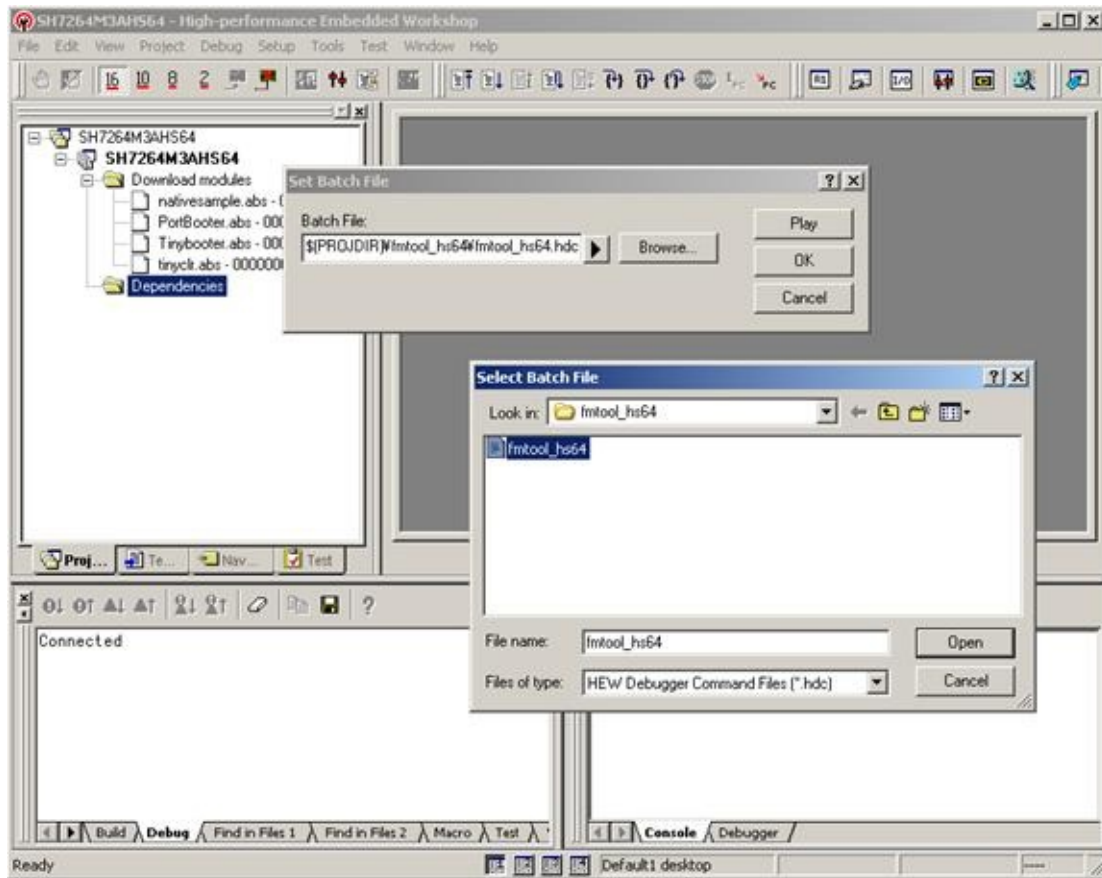
- Specify the batch file



**Figure 4.8 Popup menu on Command Line window**

On the command line window, do the right clicking so that the popup menu shows up.

Select “Set Batch file...” and specify the batch file name like the figure in next page,



**Figure 4.9 Select Batch File**

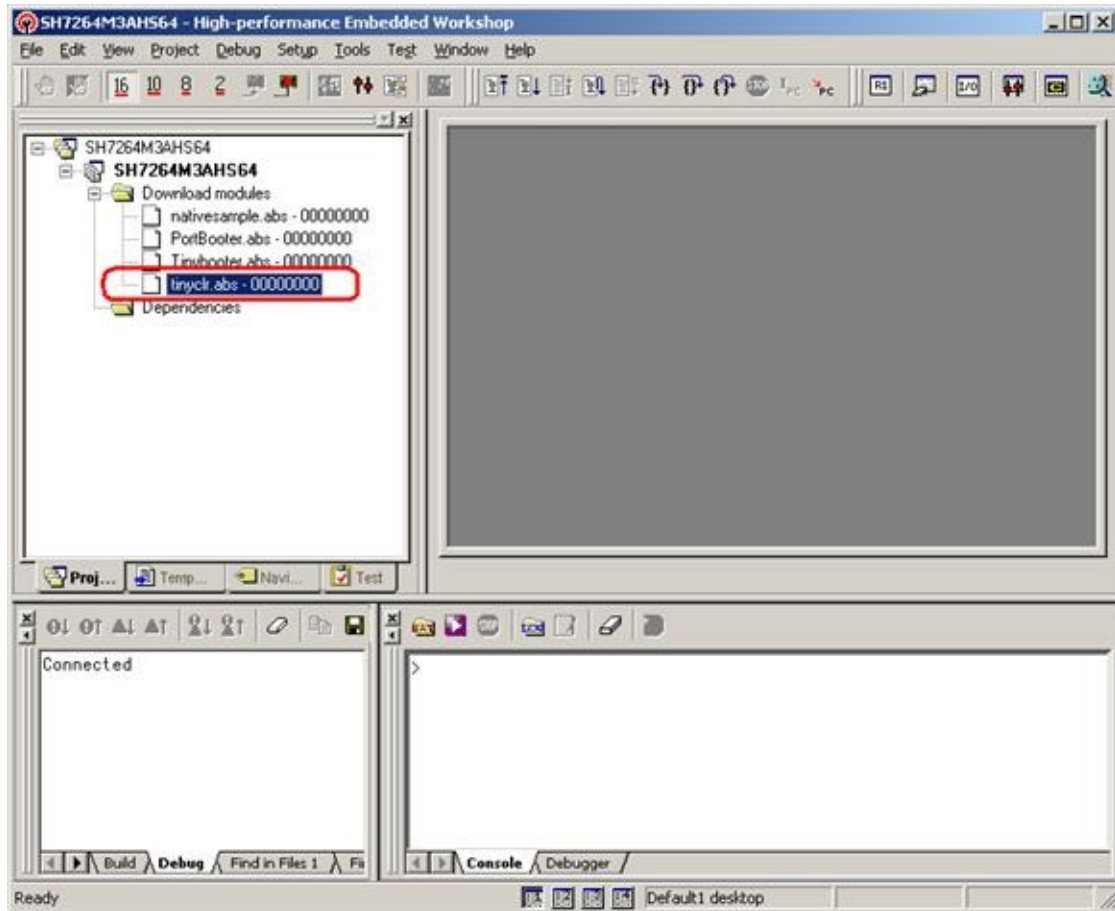
- Run the batch file

After specify the batch file, please press “Play” button so that batch file starts to run.



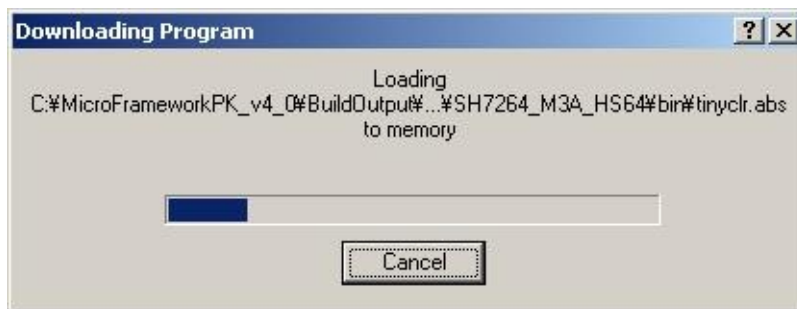
iv) Download module to Flash

In order to download, double click on the name of the download module which you want to download to flash.



**Figure 4.10 Hew window**

Below message should display during download.



**Figure 4.11 Downloading Program dialog box**

In the case of downloading NativeSample, please select nativesample.abs instead of tinyclr.abs.

## 5. The case of using limited functions with Serial Flash memory and Internal RAM

In this section, we shall describe the way to build, download and execute SH7264 M3A-HS64G50 solution available in the porting kit with Serial Flash Memory.

Since the Serial Flash Memory doesn't have XiP(execute-in-place) capability, CPU need to copy program data from Serial Flash Memory to Internal RAM and execute from RAM.

To copy data from the Serial Flash memory to the Internal RAM automatically, the loader program is required. The loader program is named "sh7264\_sflash\_loader\_prog.abs" in the E10A-USB workspace.

For detail for how to use the Serial Flash memory, refer the application note about "Boot from the Serial Flash Memory". This application note and sample source code can be downloaded from Renesas Web Site.

Please note, we modified the source code of sh7264\_sflash\_loader\_prog.abs little bit to suitable for .NET Micro Framework. The modified source code is in the sh7264\_sflash\_downloader\_for\_NETMF.zip.

## 5.1 How to Build

1) Using Command-line prompt, change the directory to  
“Solutions\SH7264\_M3A\_HS64”

```
C:\MicroFrameworkPK_v4_1>cd solutions\SH7264_M3A_HS64
```

2) Run

```
Msbuild dotnetmf.proj /t:build /p:flavor=debug;SERIALFLASH=true
```

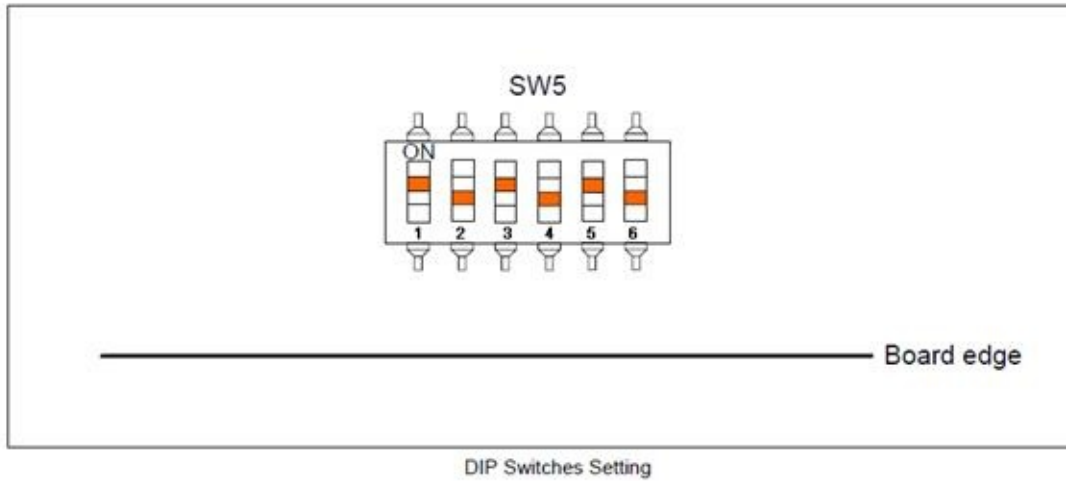
Flavor: <debug|release|rtm>      SERIALFLASH : <ture|false>

If you want to debug your program using E10A-USB Emulator, please specify “debug” for “flavor” option.

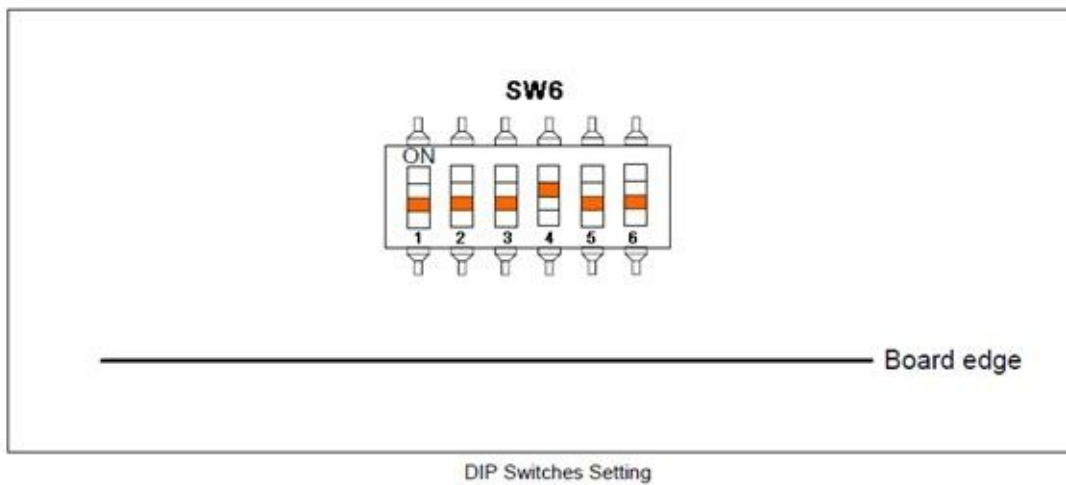
In this case, specify SERIALFLASH to “true”.

## 5.2 Board switch settings

There are two DIP switches located on the CPU board. The function of each switch and its connection are shown below:



**Figure 5.1 SW5 settings**



**Figure 5.2 SW6 settings**

## 5.3 Download using E10A-USB

### 1) Setup the E10A-USB Emulator

Install the E10A-USB Emulator software into your PC.

During install, you should select the device group for E10A-USB then specify “Super H RISC engine family SH-2A device group”

### 2) How to start downloading using E10A-USB

Extract the HEW workspace for SH7264 M3A-HS64G50 Porting Kit from file SH7264M3AHS64.zip at any place in your PC.

In the following instructions, we will assume it is installed in “c:\workspace”

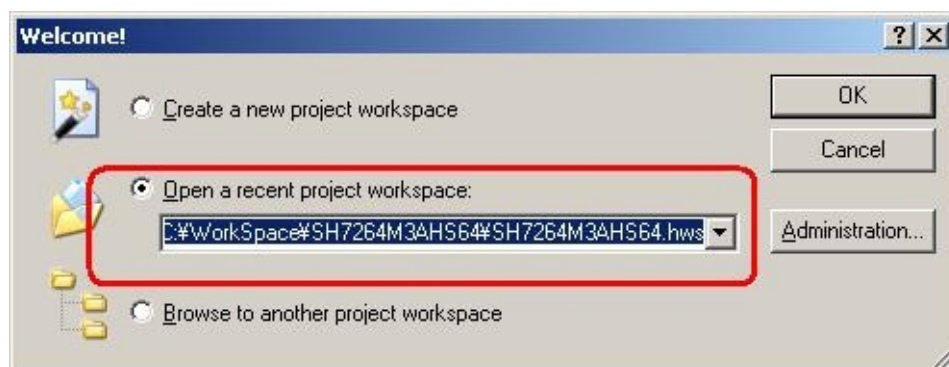
Start the Hew with choosing below menu



**Figure 5.1 Hew menu**

Then Hew will be show up and you can see below dialog box.

Specify the workspace as below, and press OK button.



**Figure 5.2 Welcome dialog box**



In below dialog box, set options like below.



**Figure 5.3 Select Emulator mode dialog box**

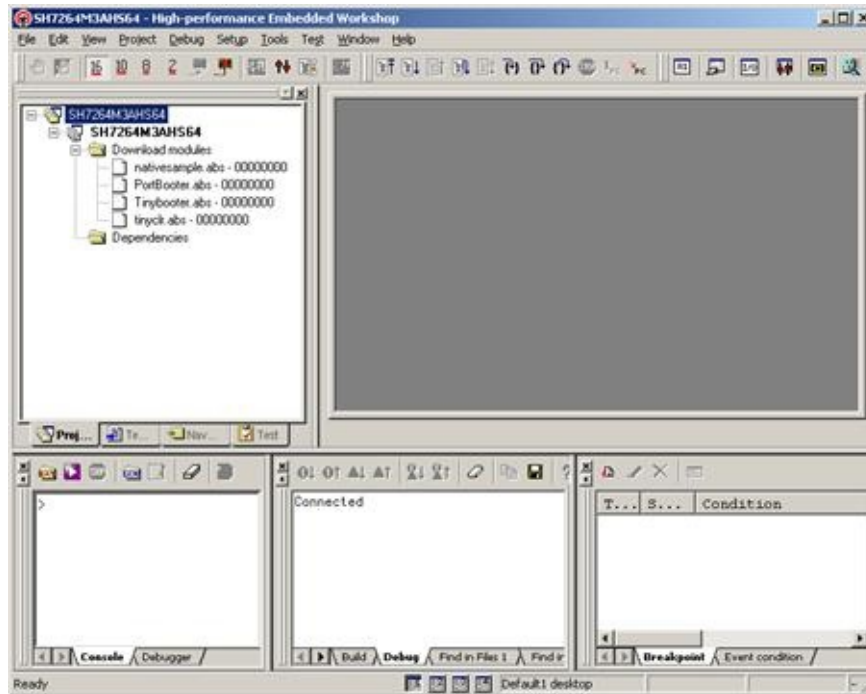
When below message shows up, reset the target board by pressing the reset button on the board and then press OK button.



**Figure 5.4 Heush2acustom1 dialog box**

Then Hew/E10A-USB will finish connection with SH7264.



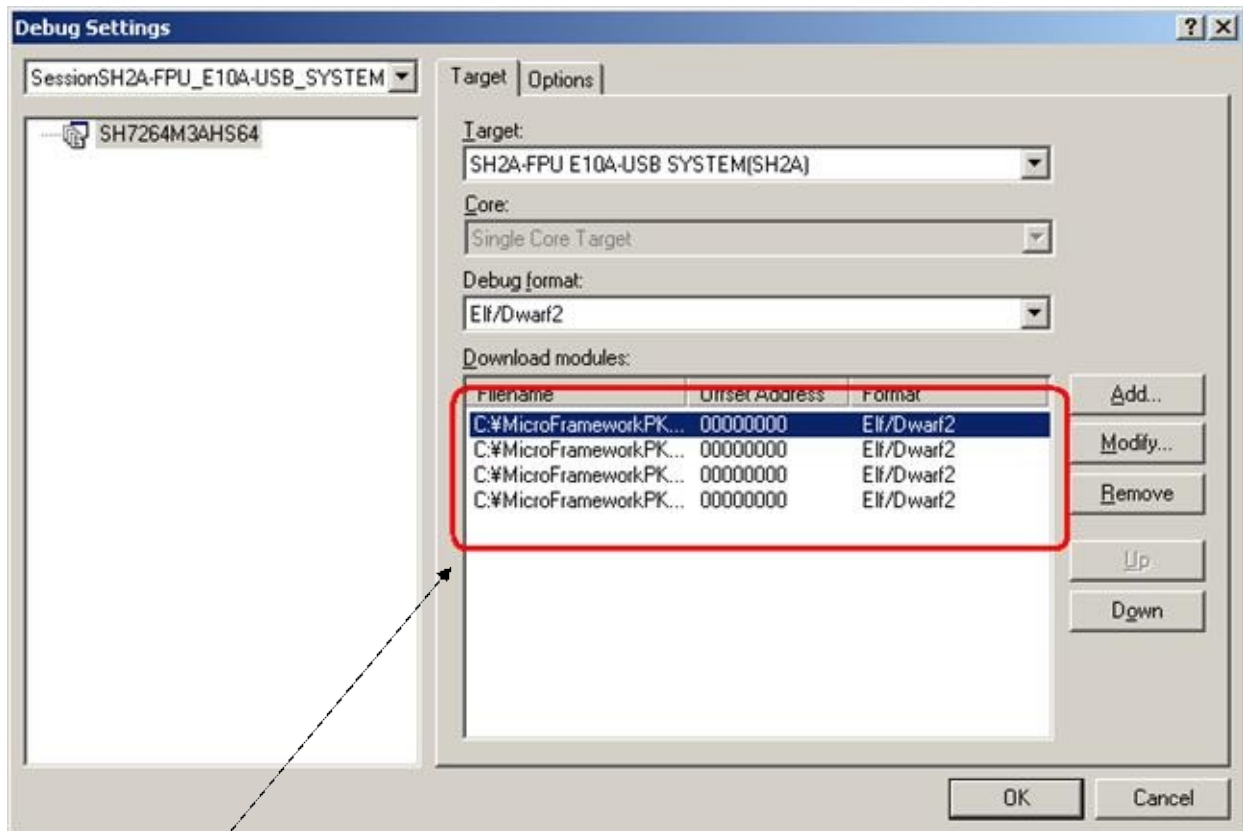


**Figure 5.5 Hew window**

3) How to download the program to the flash memory

i) Prepare the download module.

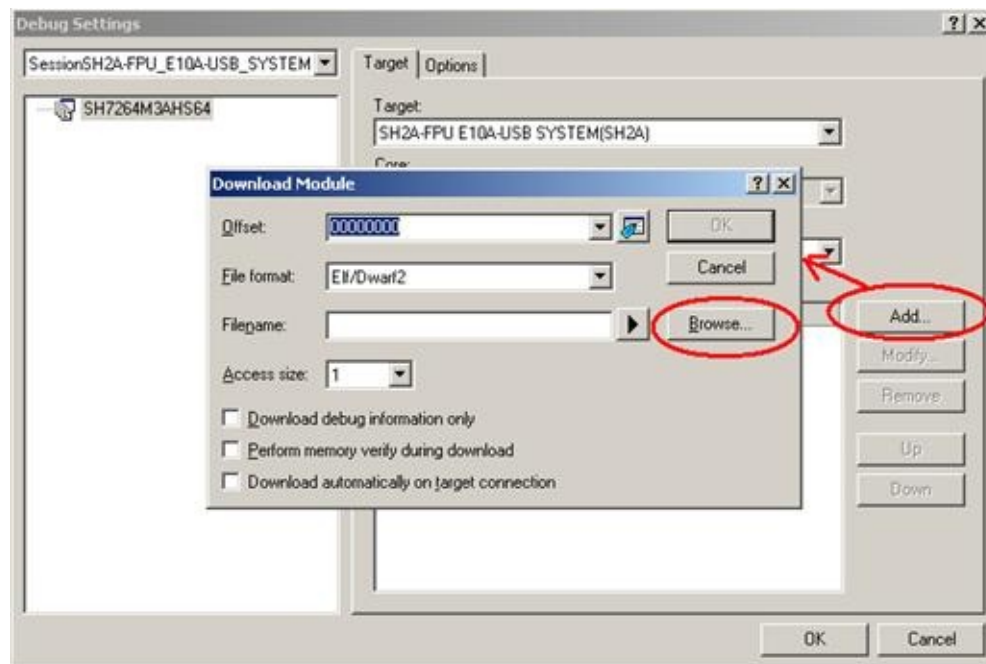
Select the [Debug] -> [Debug Settings...] from the menu bar of Hew then below dialog box will show up.



**Figure 5.6 Debug Settings dialog box**

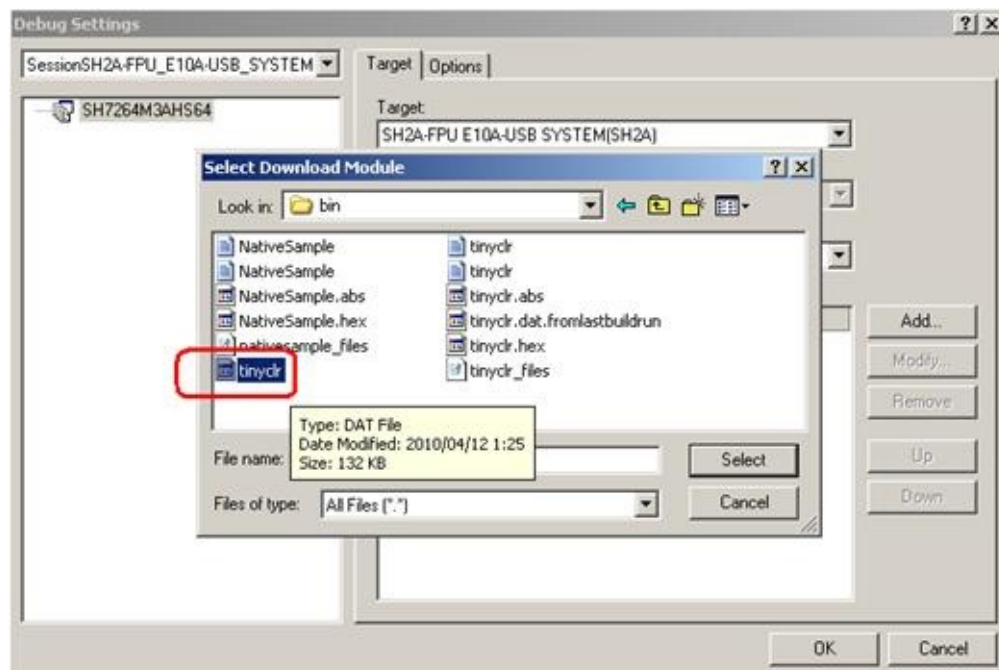
Here is the definition of download modules. Please change the Path setting for each download module by clicking “Modify” button.

In the case of downloading TinyCLR, additional one file needs to be downloaded.



**Figure 5.7 Download Modules dialog box**

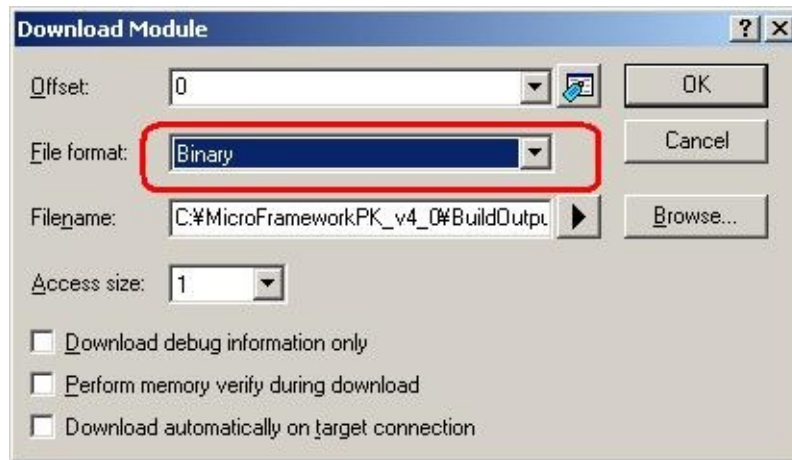
Press [Add...] button so the [Download Module] dialog box shows up. Then press [Browse...] button.



### **Figure 5.8 Select Download Module**

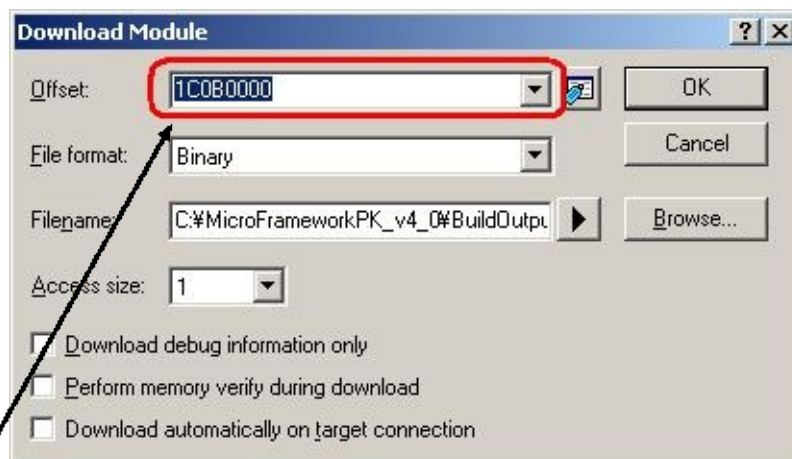
In this dialog box, select  
C:\MicroFrameworkPK\_v4\_1\BuildOutput\SH2A\SHC9.2\Be\FLASH\  
<option>\SH7264\_M3A\_HS64\bin\tinyclr.dat

In the next step, specify [File format:] to “Binary”.



**Figure 5.9 Download Module dialog box**

In addition, [Offset:] should be changed.



**Figure 5.10 Download Module dialog box**

The offset is decided from the definition in Scatterfile\_definition.proj file.

To know the offset, open

C:\MicroFrameworkPK\_v4\_1\Solutions\SH7264\_M3A\_HS64\TinyCLR\Scatterf

```

<!-- A simple scatterfile for SH -->
<Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003" >
<PropertyGroup Condition="''$(SERIALFLASH)'!='true' ">
  <ER_PROG_ADDR>20000</ER_PROG_ADDR>
  <ER_DATA_ADDR>20200000</ER_DATA_ADDR>
  <ER_CONFIG_ADDR>203F0000</ER_CONFIG_ADDR>
  <ER_INIT_LOAD>0100000</ER_INIT_LOAD>
  <ER_INIT_EXEC>0c000000</ER_INIT_EXEC>
  <ER_RW_ADDR>0C010000</ER_RW_ADDR>
  <ER_VTR_ADDR>00</ER_VTR_ADDR>
  <ER_PReset_ADDR>20001000</ER_PReset_ADDR>
  <STACK_ADDR>0c06fff8</STACK_ADDR>
  <HEAP_ADDR>0c200000</HEAP_ADDR>
  <HEAP_END>0c5fffff</HEAP_END>
  <PROG_RAM_ADDR>0c100000</PROG_RAM_ADDR>
  <PROG_RAM_END>0c10fff8</PROG_RAM_END>
  <ENTRY_ADDR></ENTRY_ADDR>
</PropertyGroup>

<PropertyGroup Condition="''$(SERIALFLASH)'=='true' ">
  <ER_PROG_ADDR>1c001100</ER_PROG_ADDR>
  <ER_DATA_ADDR>000b2000</ER_DATA_ADDR>
  <ER_CONFIG_ADDR>1c0ad000</ER_CONFIG_ADDR>
  <ER_INIT_LOAD>1c0ada00</ER_INIT_LOAD>
  <ER_INIT_EXEC></ER_INIT_EXEC>
  <ER_RW_ADDR>FFF80000</ER_RW_ADDR>
  <ER_VTR_ADDR>1C000000</ER_VTR_ADDR>
  <ER_PReset_ADDR>1C000E00</ER_PReset_ADDR>
  <STACK_ADDR>fff88800</STACK_ADDR>
  <HEAP_ADDR>1c0ae800</HEAP_ADDR>
  <HEAP_END>1c0dc3FC</HEAP_END>
  <ENTRY_ADDR></ENTRY_ADDR>
</PropertyGroup>

</Project>

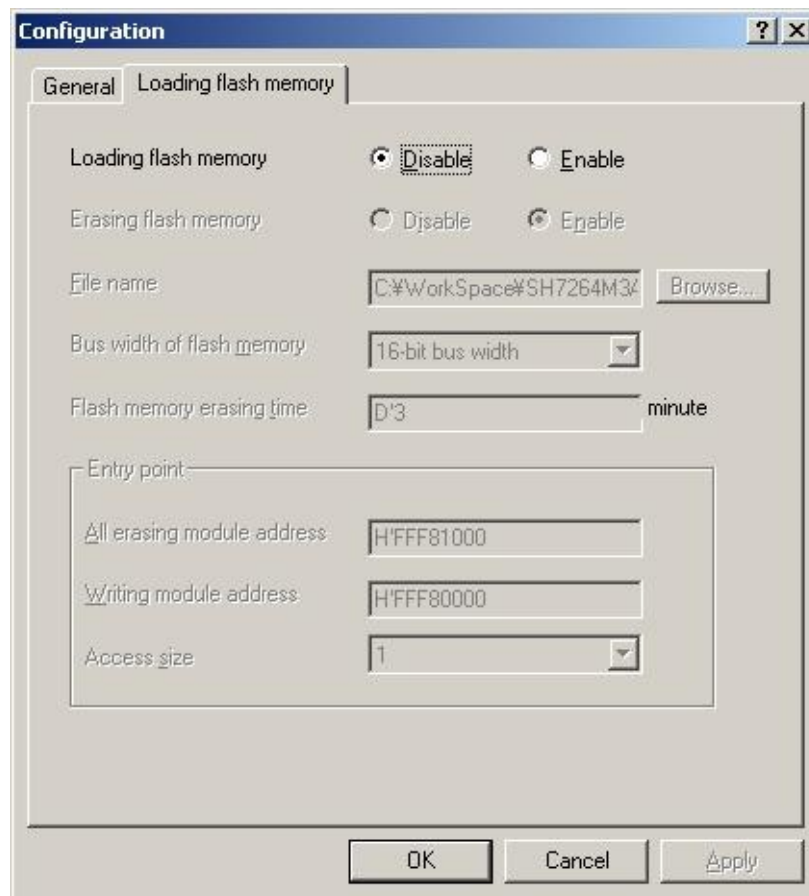
```

Offset = This value +  
0x1c000000 - 0x2000

ii) Prepare for downloading to Flash Memory.

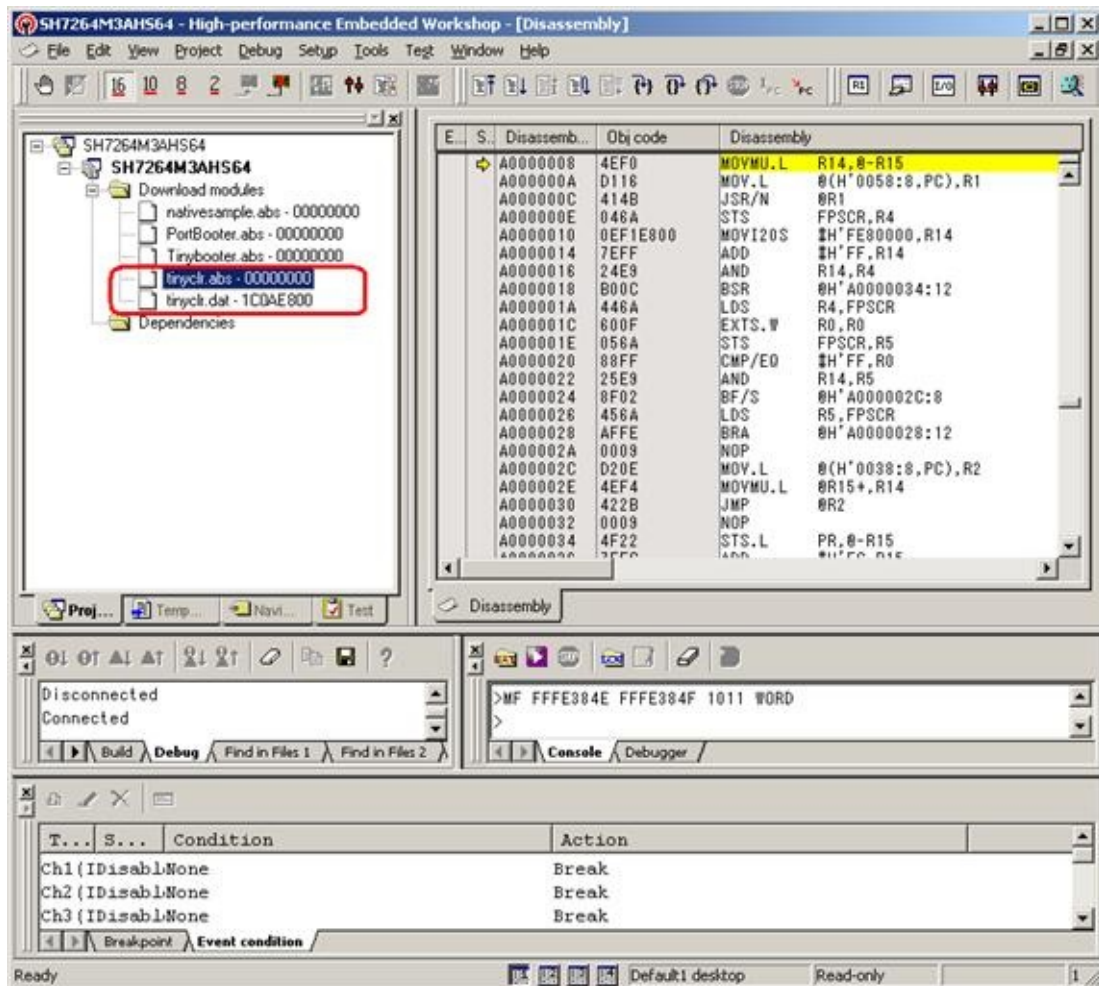
Select the [Setup] -> [Emulator] -> [System...] from the menu bar of Hew then “Configuration” dialog box will show up.

Select “Loading flash memory” tab. Then, specify as same as below figure.



**Figure 5.11 Configuration dialog box**

iii) Download module to RAM temporary



**Figure 5.12 Hew window**

In order to download, double click on the name of the download module which you want to download to RAM.

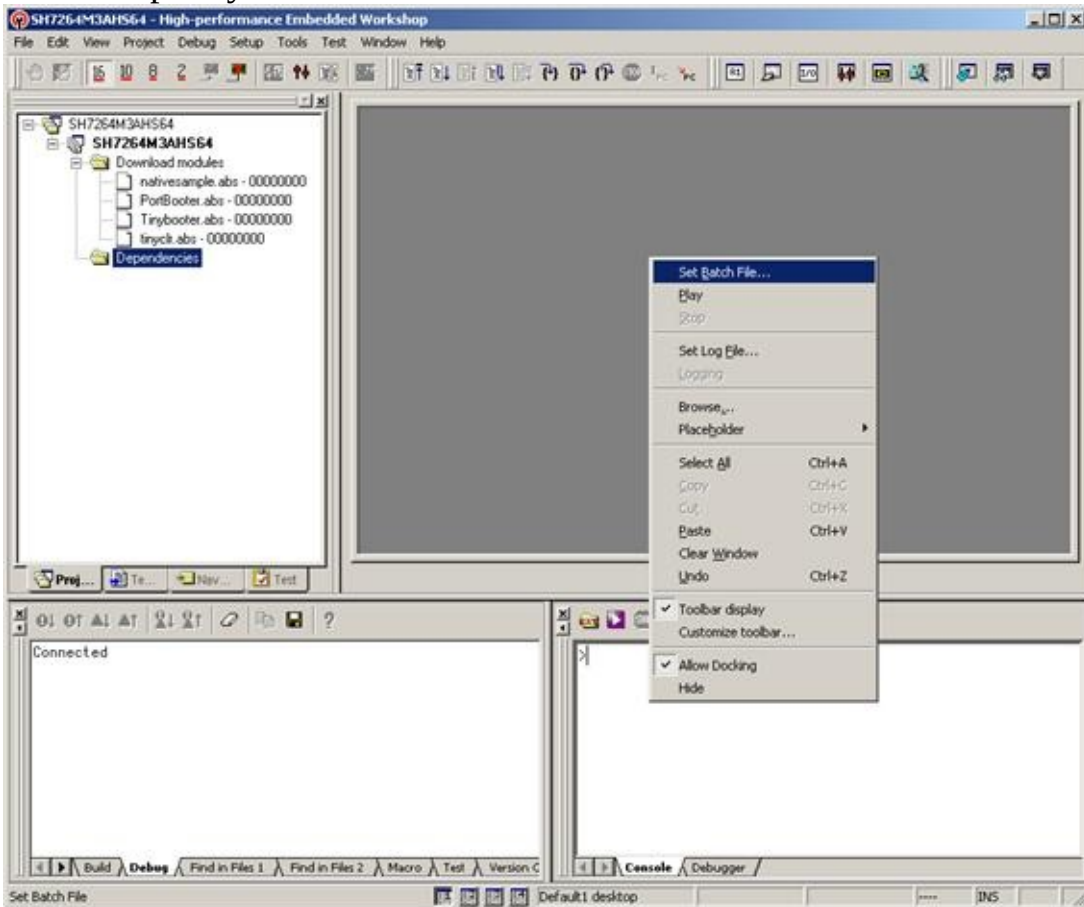
In the case of downloading TinyCLR, download both of TinyCLR.abs and tinyclr.dat. In the case of downloading NativeSample, download only nativesample.abs.



iv) Specify and Run the Batch file to write data to Serial Flash Memory.

In order to download data to Serial flash, the batch file should be executed.

- Specify the batch file



**Figure 5.13 Popup menu on Command Line window**

On the command line window, do the right clicking so that the popup menu shows up.

Select “Set Batch file...” and specify the batch file name like the figure in next page,

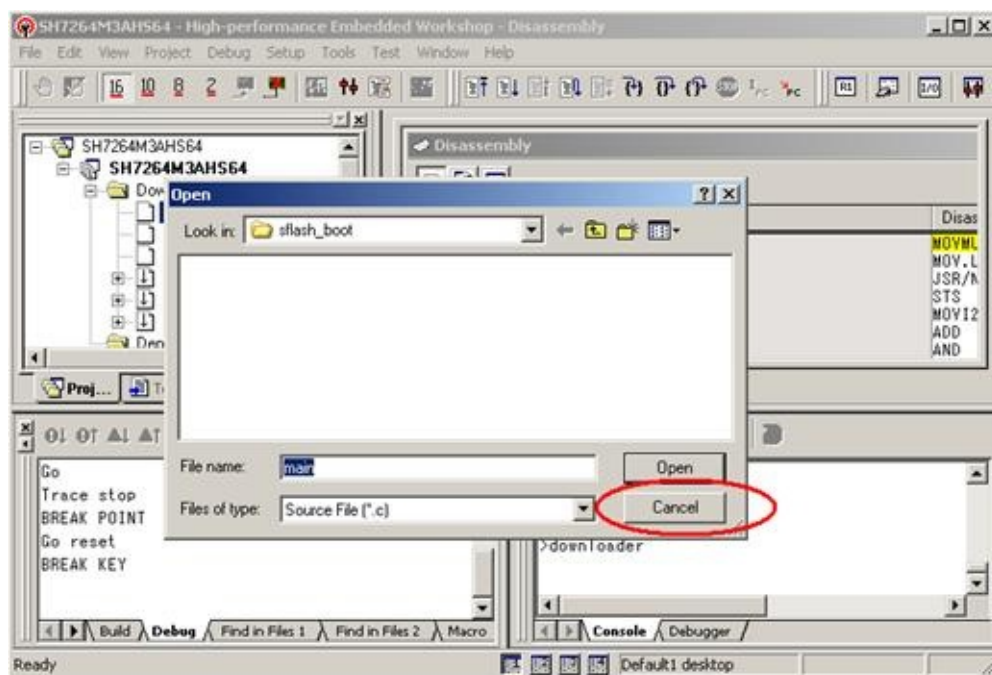


**Figure 5.14 Select Batch File**

Specify ...\\sfash\_boot\\downloader.hdc.

- Run the batch file

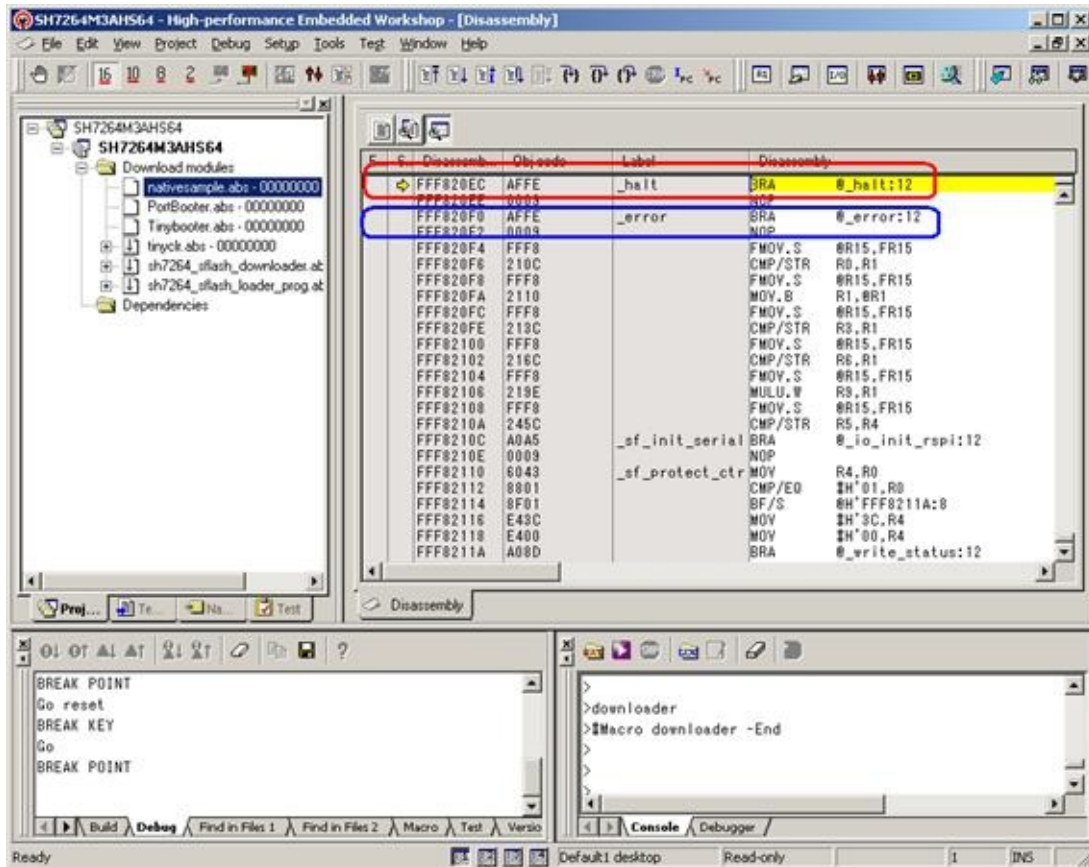
After specify the batch file, please press “Play” button so that batch file starts to run. During running the batch file, following dialog box might be opened several times.



**Figure 5.15 Open dialog box**

Press “Cancel” button.

After several seconds or minutes, the program will be stop.



**Figure 5.16 Hew window**

If the program stops by red portion, the download to Serial Flash is finished properly.

If the program stops by blue portion, the download to Serial Flash is not finished properly.

3) How to copy the program from Serial Flash to Internal RAM and run

In order to copy program and run, you just simply need to press the [Reset Go] button.

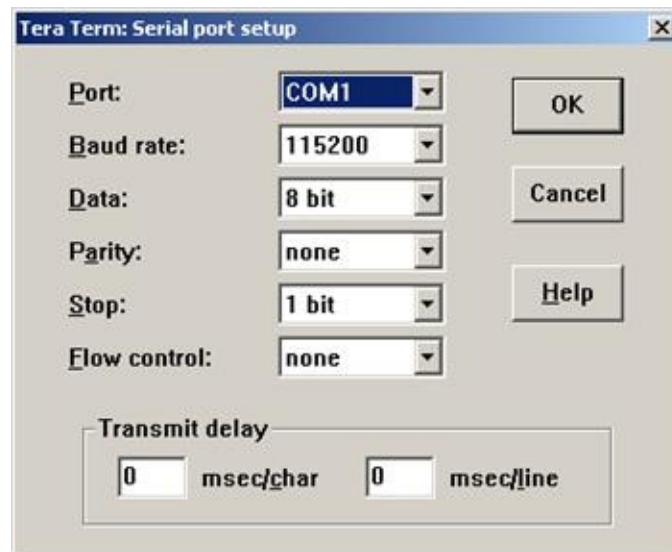


Figure 5.17 Reset Go button

## 6. Running TinyCLR

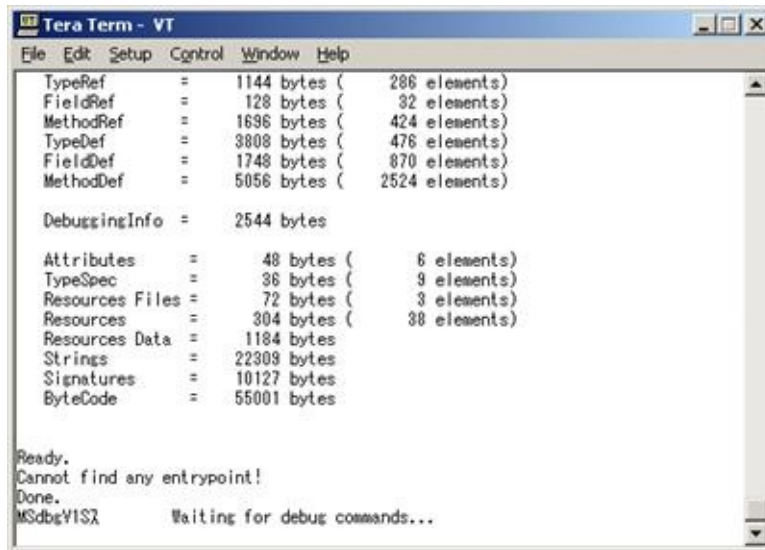
Hit “Stop” button in the toolbar or Select “Halt Program” option from Debug menu. Disconnect E10A-USB from HEW, power OFF the board and disconnect E10A connection from board.

To verify that TinyCLR is up and running connect the serial port of the target with PC and start the terminal program Tera Term Pro with following settings:



**Figure 6.1 Serial port setup**

Power ON the board and you should see following messages in the serial terminal:



**Figure 6.2 Serial terminal**

If you can see above messages, congratulations! Your TinyCLR is up and running. This example is the case of using TinyCLR.abs by debug build.

Please note that don't forget to close Tera Term Pro before proceeding forward.

## **7. How to deploy application**

.NET Micro Framework has the feature of developing Application using Visual Studio2010.

In this section, the way to deploy application is explained by using Sample application.

## **7.1      Install .NET Micro Framework SDK**

In order to use .NET Micro Framework with Visual Studio, you need to install .NET Micro Framework SDK (MicroFrameworkSDK.msi) into your PC.

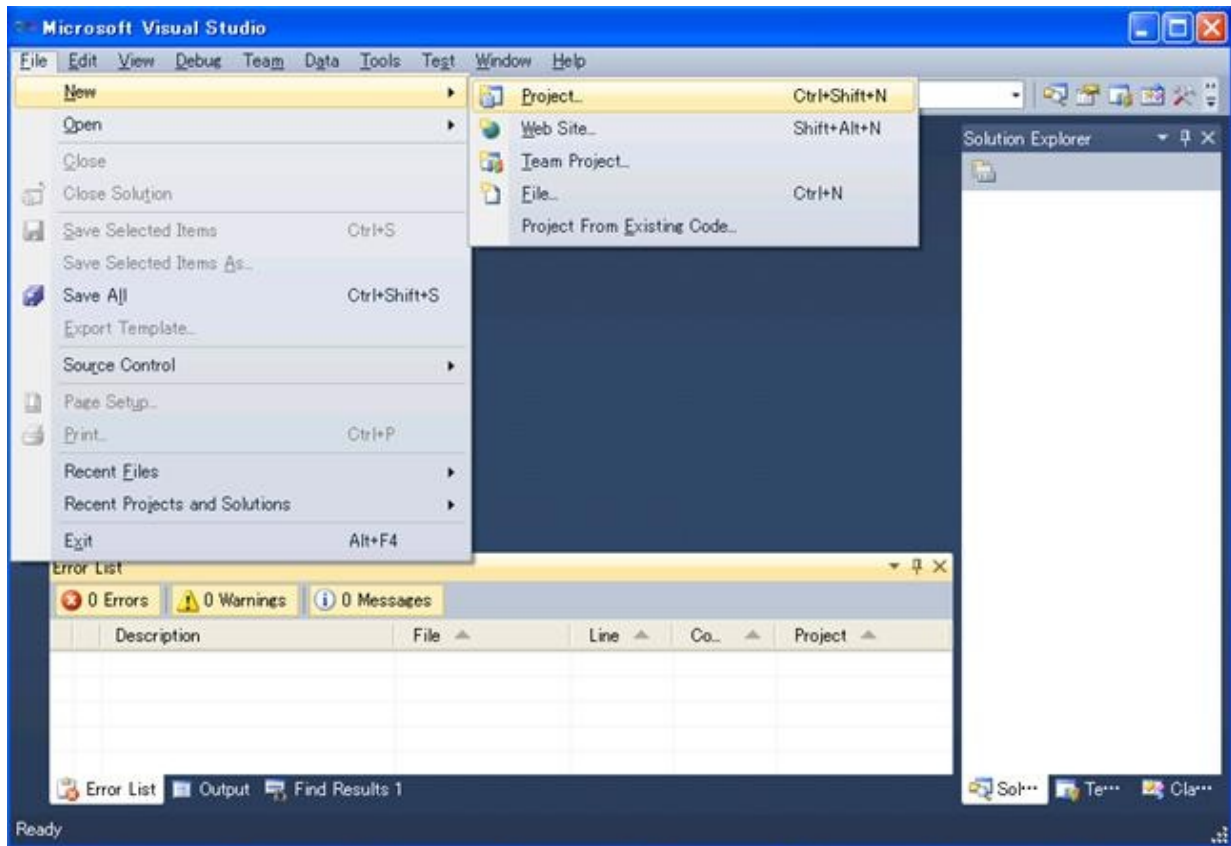


## 7.2 Prepare Sample application

At first, you need to create the application. Please open Visual Studio.

i) Create new project

Select File menu -> New -> Project.



**Figure 7.1 Microsoft Visual Studio**

Please note, these screens are captured from Visual Studio2010 professional. You will see difference screens and menus in the case of Visual Studio 2010 Express Edition. But, you will find similar features in the menus of Visual Studio 2010 Express Edition.

ii) Select the Project type, Template and project name

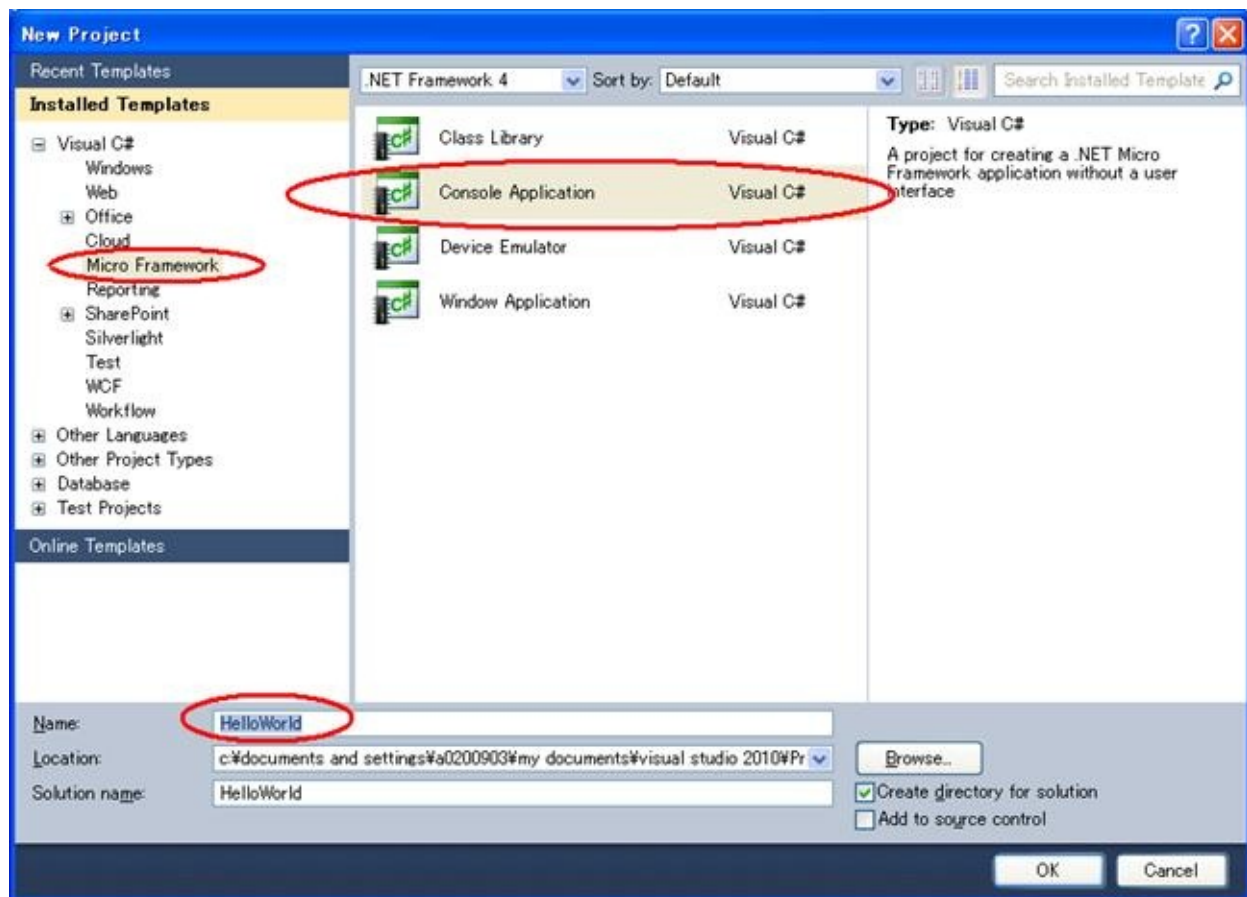
Select below,

Project Type: Micro Framework

Template: Console Application

Name: HelloWorld

And press OK button.

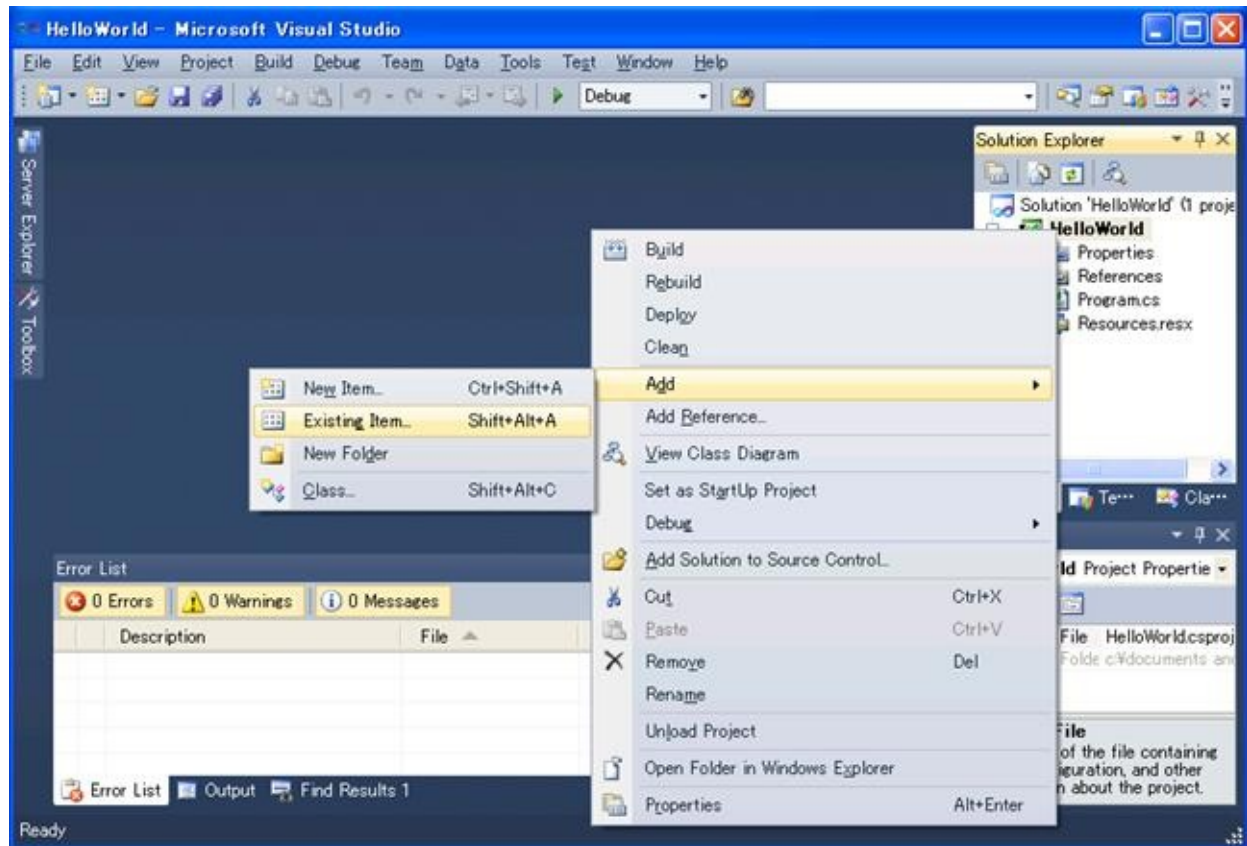


**Figure 7.2 New Project**

Note depending upon your application please select the appropriate options. If you are creating a GUI based application it is advised to select “Window Application” option.

iii) Copy the source file of Sample program into this project

Right click on “HelloWorld” in the Solution Explorer and select Add -> Existing Item....

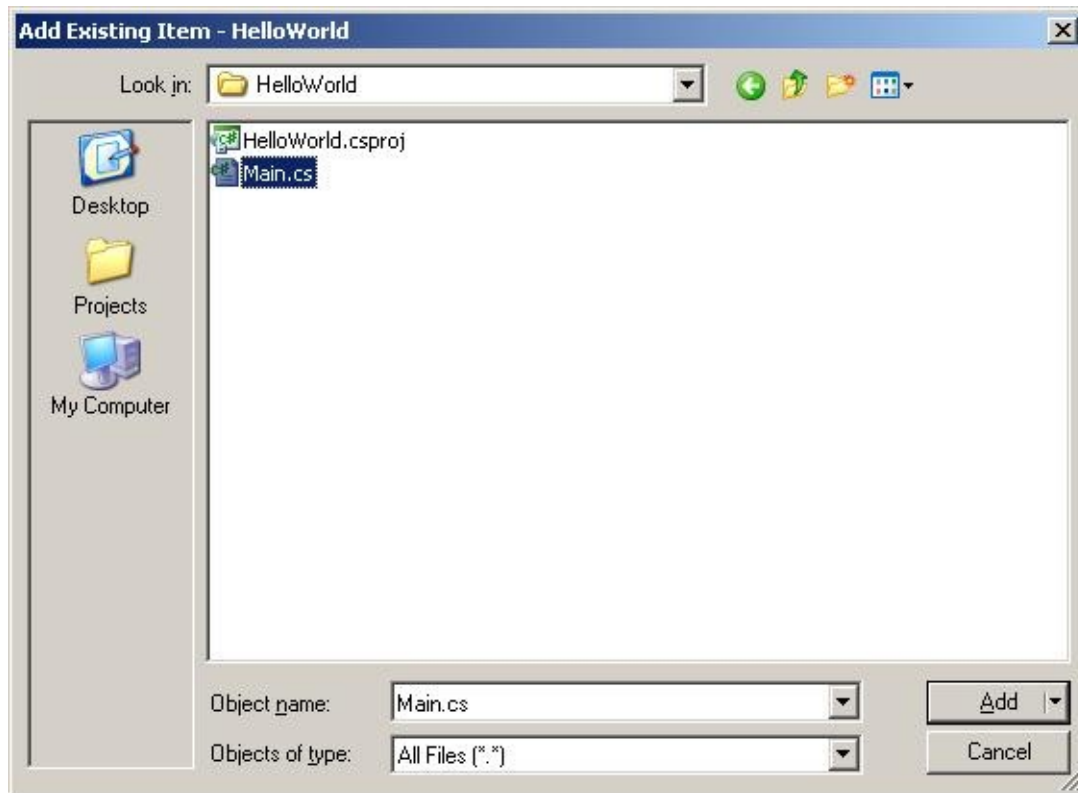


**Figure 7.3 Microsoft Visual Studio**

Select the below file,

C:\MicroFrameworkPK\_v4\_1\Product\Sample\HelloWorld\Main.cs

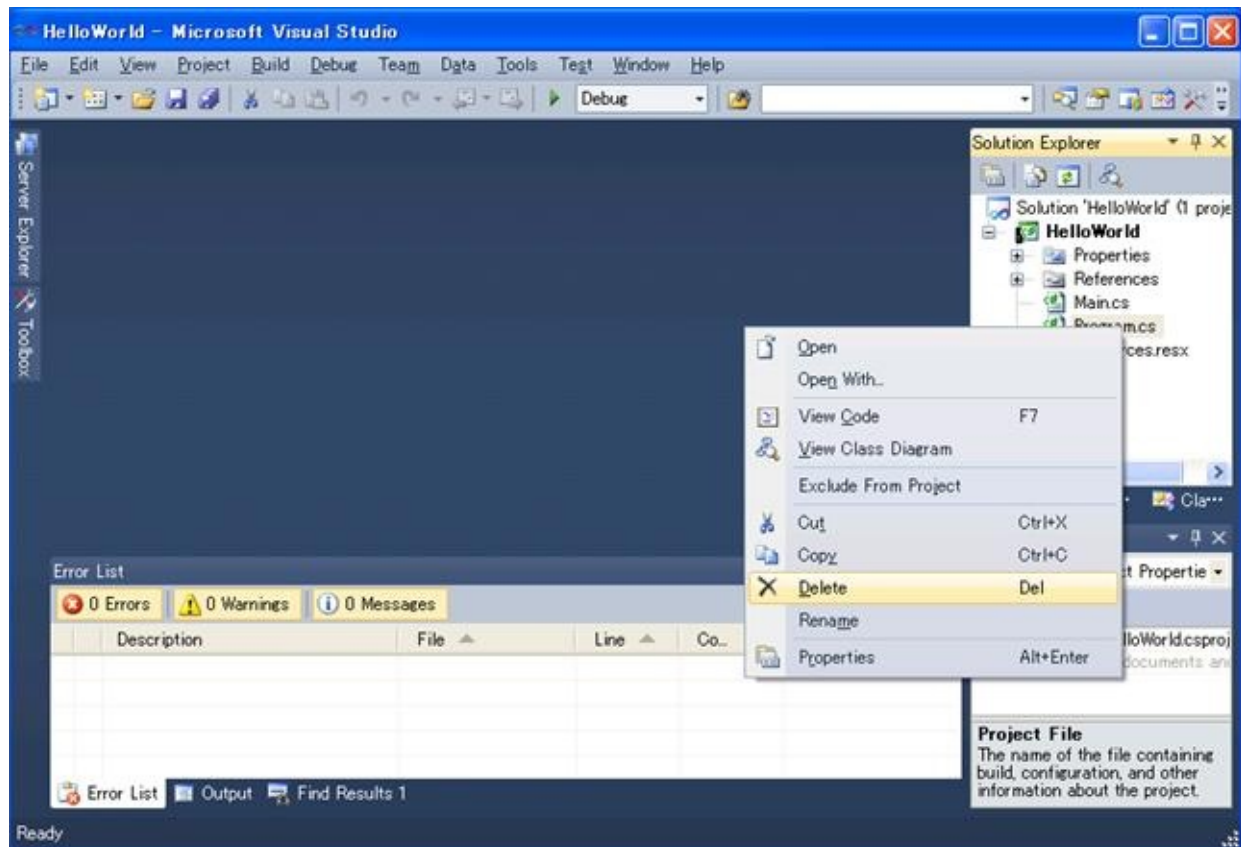
And press Add button.



**Figure 7.4 Add Existing Item dialog box**

iv) Remove unnecessary file

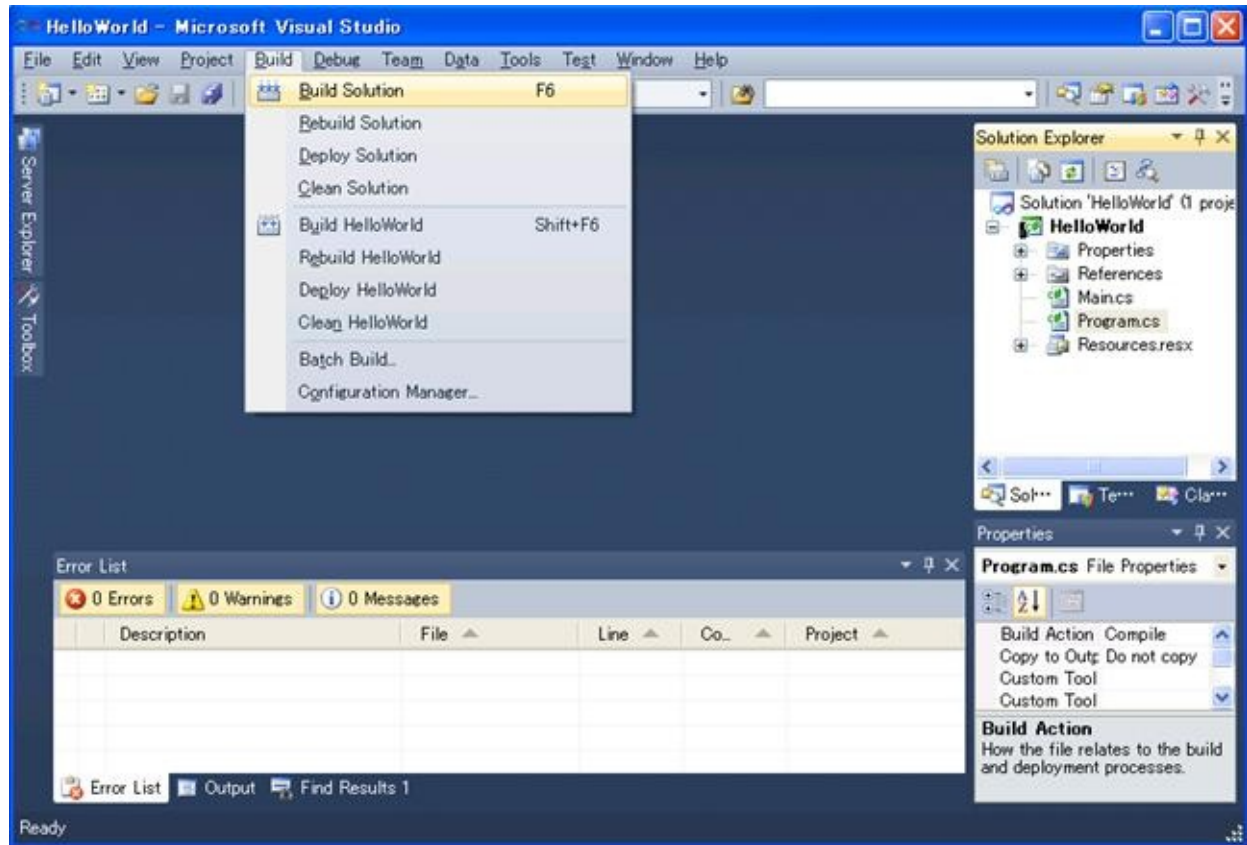
Remove Program.cs file from the HelloWorld project.



**Figure 7.5 Microsoft Visual Studio**

v) Build solution

Build Solution by selecting Build Solution menu,



**Figure 7.6 Microsoft Visual Studio**

Then build shall be completed successfully.

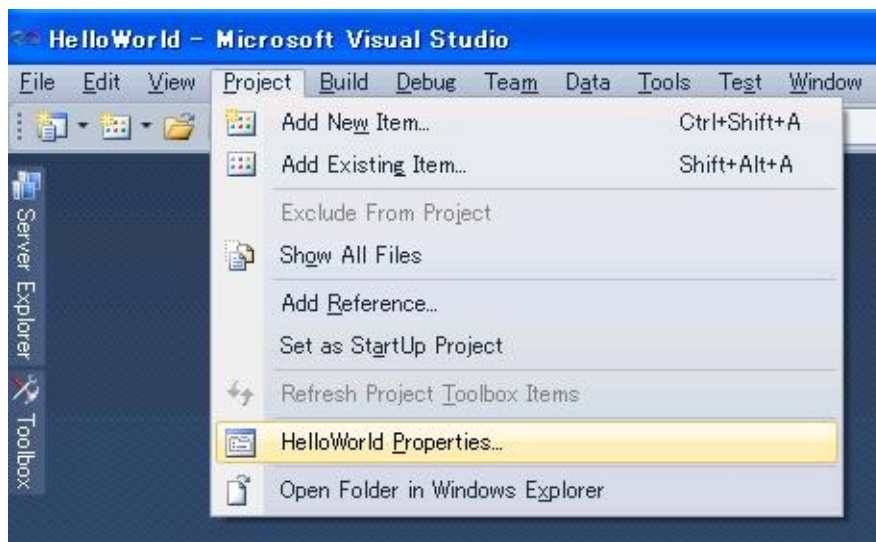
## 7.3 Deployment and run the application

In order to communicate with visual studio and your board, you need to download TinyCLR into the flash memory on the board.

It is assumed that you have build and downloaded TinyCLR into the flash memory on the board. If not, please build your Porting kit and downloading tinyclr.abs into the flash memory referring to section 4.

i) Change properties for current project.

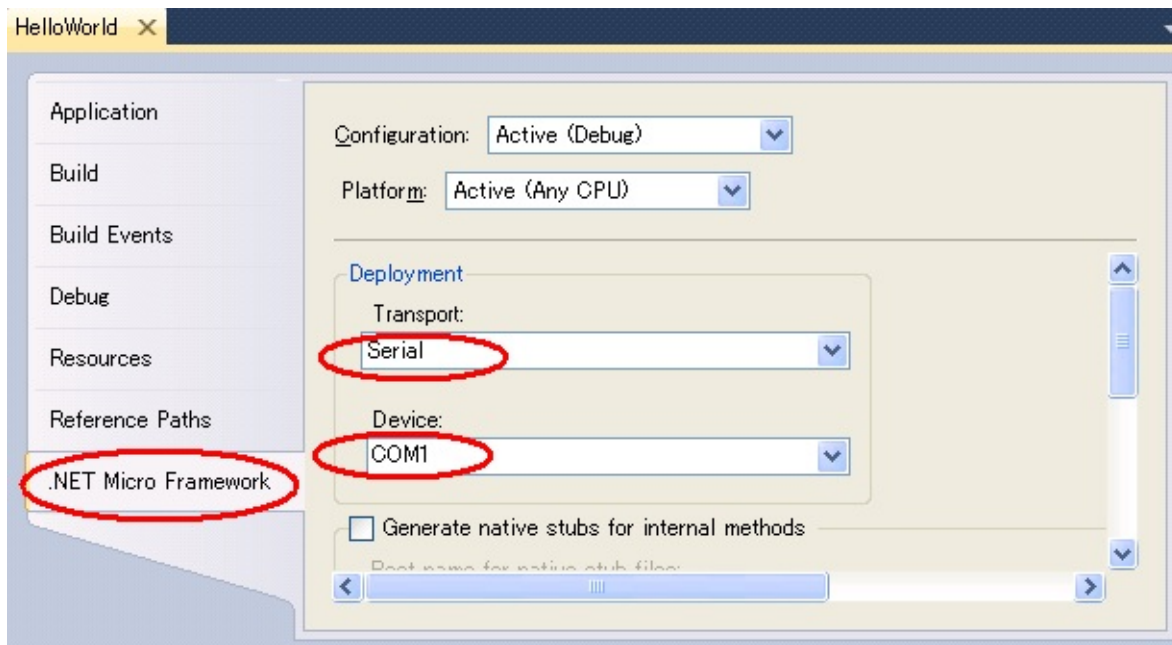
Select Project menu -> HelloWorld Properties... so that you can modify properties.



**Figure 7.7 Microsoft Visual Studio**

Select “.NET Micro Framework” sheet and change deployment option to serial as shown below, save and close this window.





**Figure 7.8 Properties...**

ii) Re-Build

In order to affect these changing, Re-build is required.

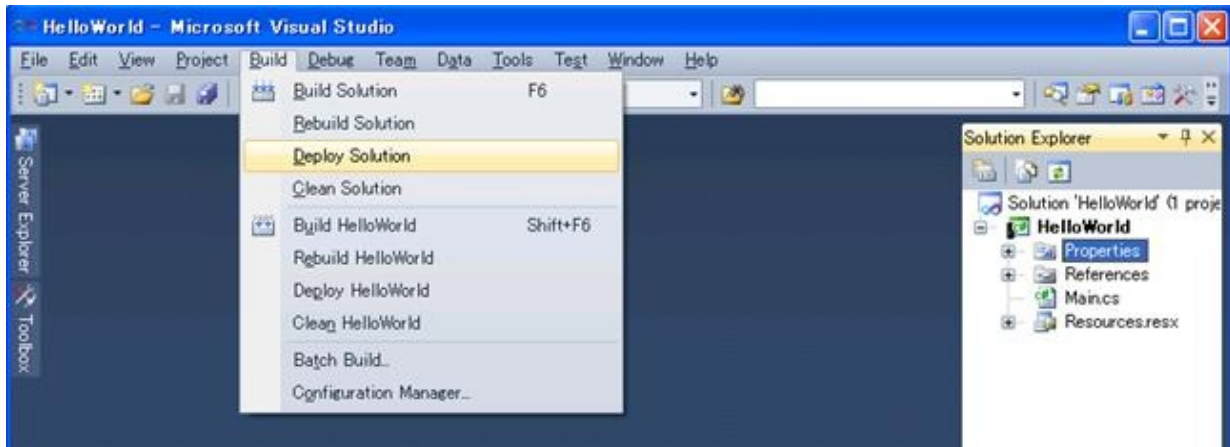
iii) Turn on the board

First, connected the serial terminal of board with that of PC using serial cable and then turn on the board.

iv) Deploying application

Select Build menu -> Deploy Solution.





**Figure 7.9 Microsoft Visual Studio**

Below message is shown up in the output window at the start of deployment process.

“Incrementally deploying assemblies to device”

When deployment gets finished without problem, you can see below message.

“Assemblies successfully deployed to device.”

If you have any trouble please make sure if you use correct flash memory configuration file.

v) Run the application

Select Debug menu -> Start Debugging so that you can see the application running and use break point via Visual Studio.

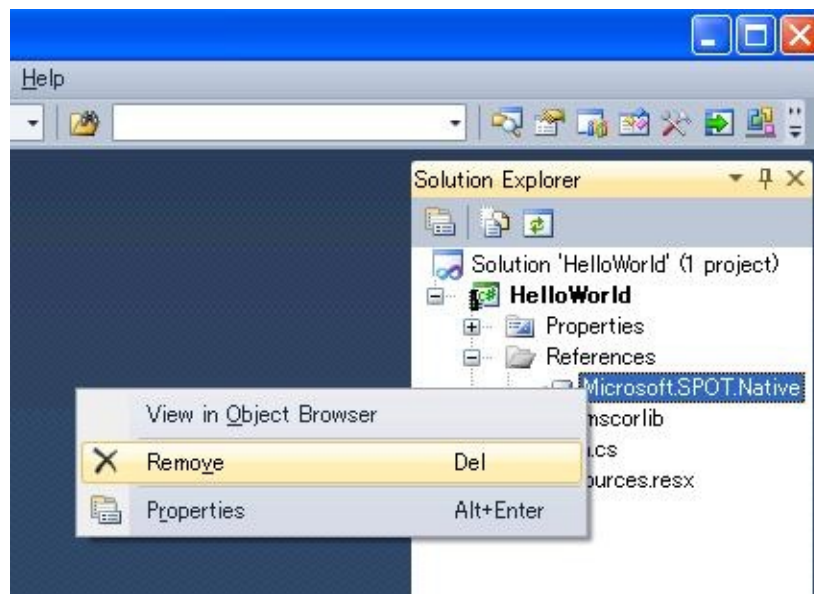
If the deployment fails, it might be caused by the unexpected data in the Deployment area in Flash memory. One of the way to avoid this is to erase all data in the deployment area using MFDeploy.exe. How to use MFDeploy.exe is described in Section 9.3.

## 7.4 Big-endian support

Normally, in the initial setting, the reference dlls are supporting Little-Endian. So it is recommended to change the reference dlls to for Big-Endian.

### i) Remove the Reference for Little-Endian

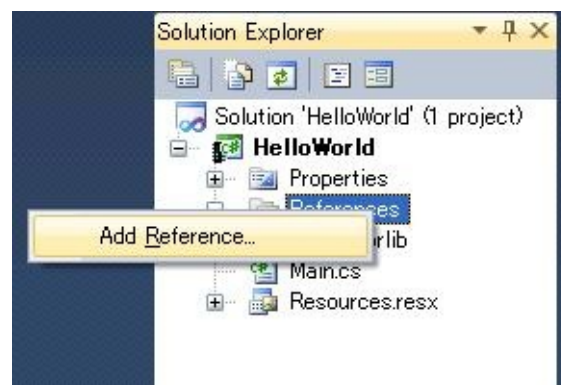
Right Click the Reference which you want to remove as shown below (Microsoft.SPOT.Native) and select “Remove” option from the popup menu.



**Figure 7.10 Solution Explorer**

### ii) Add the Reference for Big-Endian

Right Click on the “References” as shown below and select “Add reference...” option from the popup menu.



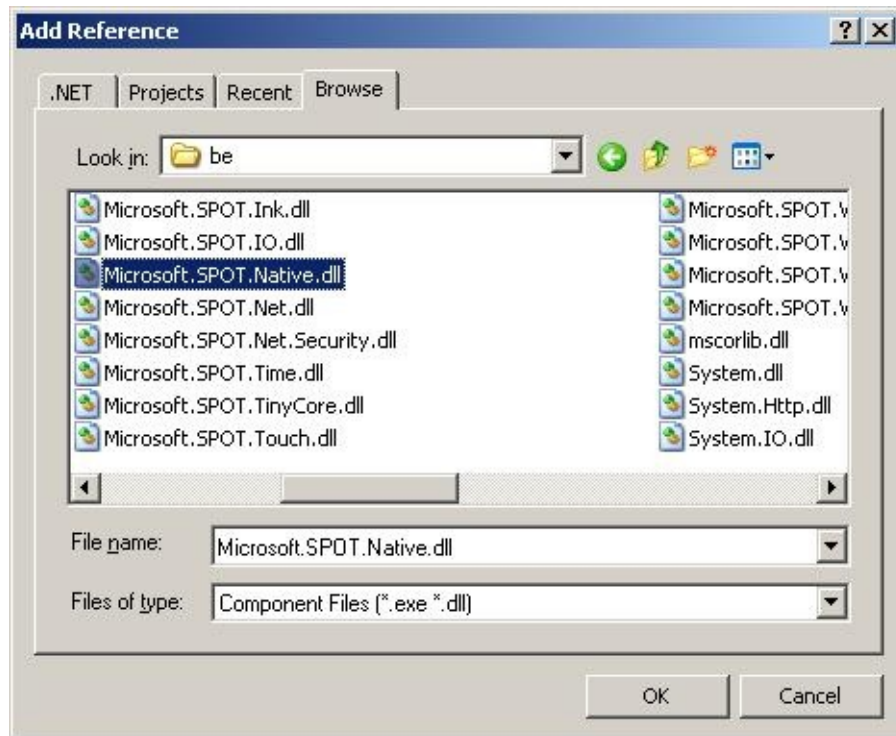
**Figure 7.11 Solution Explorer**



Then, “Add reference” window shall show up.

Select “Browse” tab and change “Look in” path to below mentioned path:

C:\Program Files\Microsoft .NET Micro Framework\v4.1\Assemblies\be

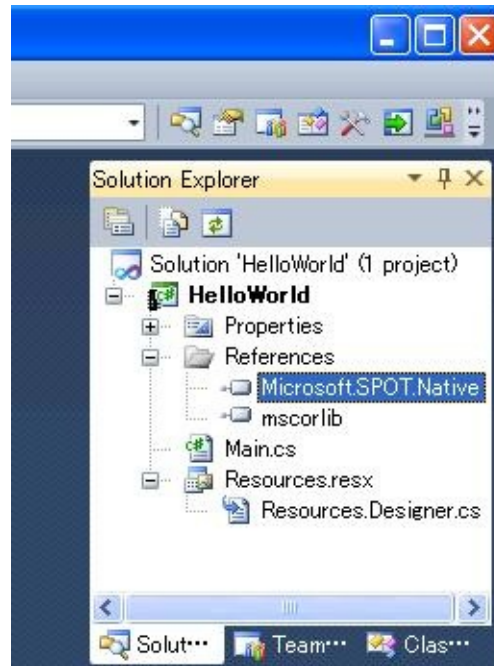


**Figure 7.12 Add Reference**

Then, select the dll which you want to add. (In this case, select Microsoft.SPOT.Native.dll.)

iii) Make sure that selected reference dll is correct.

Double Click the Reference which you want to see information.



**Figure 7.13 Solution Explorer**

An object browser tab / window shall be displayed. Verify that you see the correct path information for dll (Microsoft.SPOT.Native) as shown below.

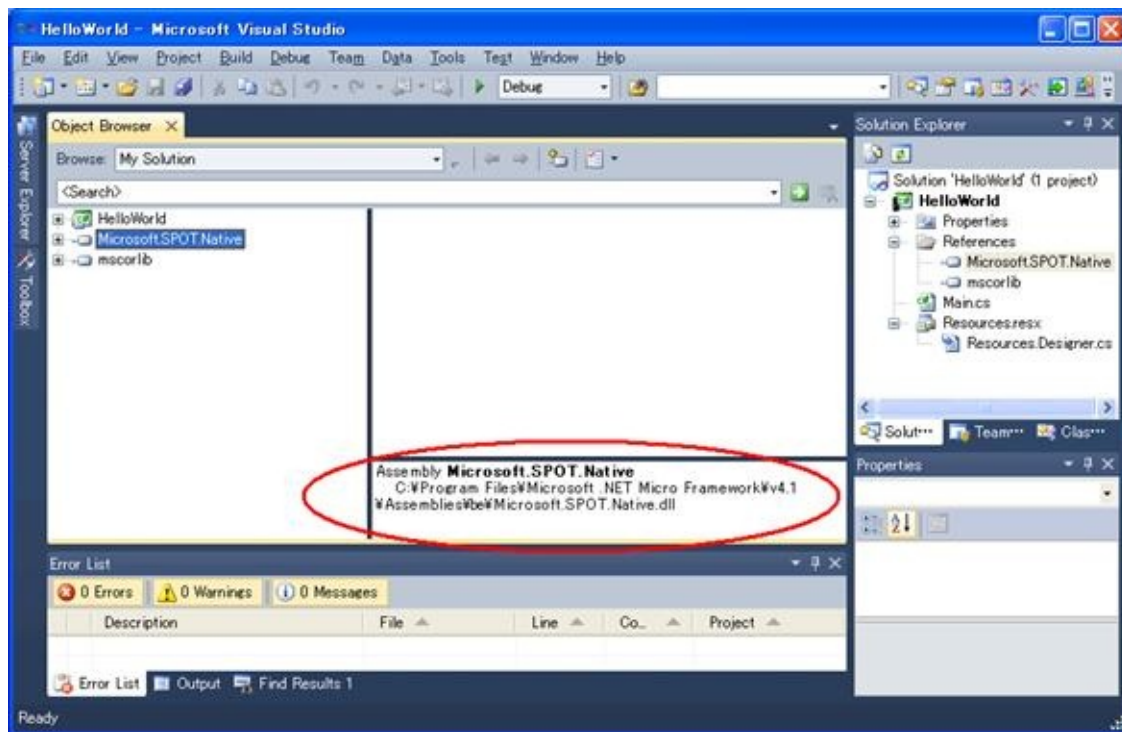


Figure 7.14 Object Browser

## 8. How to include application in TinyCLR

The application can be included into `tinyclr.abs`. The simplest thing to do is just to add the PE files to your `TinyCLR.proj` file.

### 1) Build generic application in the Porting Kit

Change current folder to “C:\MicroFrameworkPK\_v4\_1’ and execute below command,

```
MSBUILD.EXE build.dirproj
```

### 2) Build an application without Visual Studio 2010.

Build your application by following command,

```
MSBUILD.EXE build.dirproj
```

For example, if you want to build HelloWorld Sample application, use following commands,

Change current folder to “C:\MicroFrameworkPK\_v4\_1\Product\Sample’ and execute below command,

```
MSBUILD.EXE build.dirproj
```

### 3) How to include application into `tinyclr.abs`

Change below red lines for your application

Please add the following red lines after the property section of the `TinyCLR.proj` file. You will have to make sure the `.pe` files have been built.

...

```
<Import Condition="" Project="$(SPOCLIENT)\Framework\Features\Diagnostics.featureproj" />
<Import Condition="" Project="$(SPOCLIENT)\Framework\Features\Core.featureproj" />
<Import Condition="" Project="$(SPOCLIENT)\Framework\Features\Serialization.featureproj" />

<ItemGroup>
  <MMP_DAT_CreateDatabase
Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\mscorlib.pe"/>
  <MMP_DAT_CreateDatabase
Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\Microsoft.SPOT.Native.pe"/>
  <MMP_DAT_CreateDatabase
Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\Microsoft.SPOT.Net.pe"/>
  <MMP_DAT_CreateDatabase
```

```
Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\Microsoft.SPOT.HelloWorld.pe"/>
  <MMP_DAT_CreateDatabase Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\System.pe"/>
  <MMP_DAT_CreateDatabase
Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\Microsoft.SPOT.Graphics.pe"/>
  <MMP_DAT_CreateDatabase
Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\Microsoft.SPOT.TinyCore.pe"/>
  <MMP_DAT_CreateDatabase
Include="$(BUILD_TREE_CLIENT)\pe\$(ENDIANNESS)\Microsoft.SPOT.Hardware.pe"/>
</ItemGroup>
<Import Project="$(SPOCLIENT)\tools\targets\Microsoft.SPOT.System.Interop.Settings" />
```



## **9. MFDeploy Tool**

In Porting Kit, there is a useful Tool, MFDeploy.exe. Using this tool, you can make sure if TinyCLR works fine, see the Flash memory mapping information, and erase the data in Deployment area.

## 9.1 How to build and run MFDeploy.exe

Following is the way to build and run MFDeploy.exe.

1) Make sure the Timer driver, Power driver and the Serial driver work fine

Before using MFDeploy.exe, please make sure the Timer driver, Power driver and the Serial driver works fine using NativeSample.

2) Build MFDeploy.exe

You can build MFDeploy.exe with the command “MSBUILD.EXE build.dirproj” under the C:\MicroFrameworkPK\_v4\_1 folder.

Using this command, you can build not only MFDeploy.exe but also all the managed tests in the PK as well.

MFDeploy is located in

\BuildOutput\public\Debug\Server\dl\MFDeploy.exe.

3) Run the MFDeploy.exe

i) Connect COM Port to the board

Before run the MFDeploy.exe, please make sure the COM port in your PC is not used by another application like Tera Term.

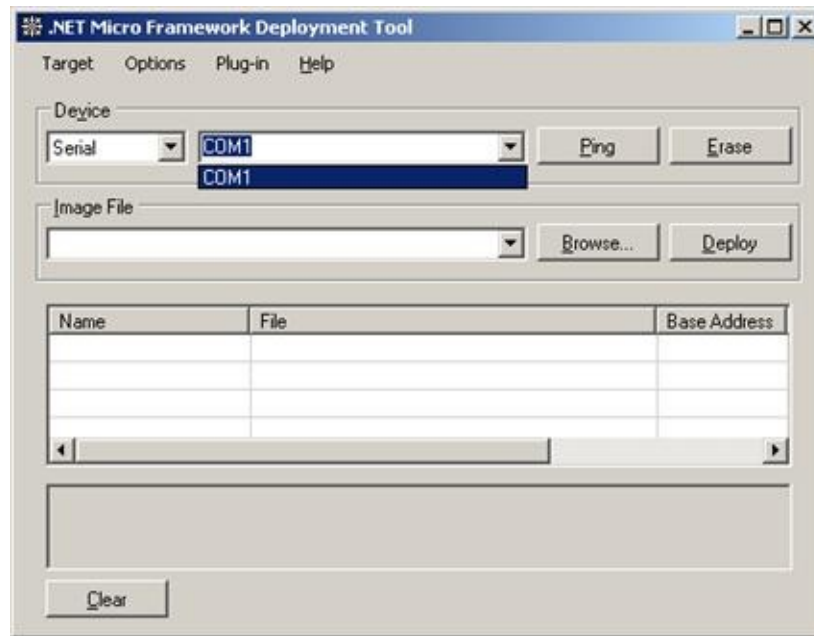
Then connect the COM port in your PC to the Serial connector on the board by Serial Cross Cable.

ii) Run the MFDeploy.exe

Run the MFDeploy.exe by double-clicking it.

iii) Select the COM Port

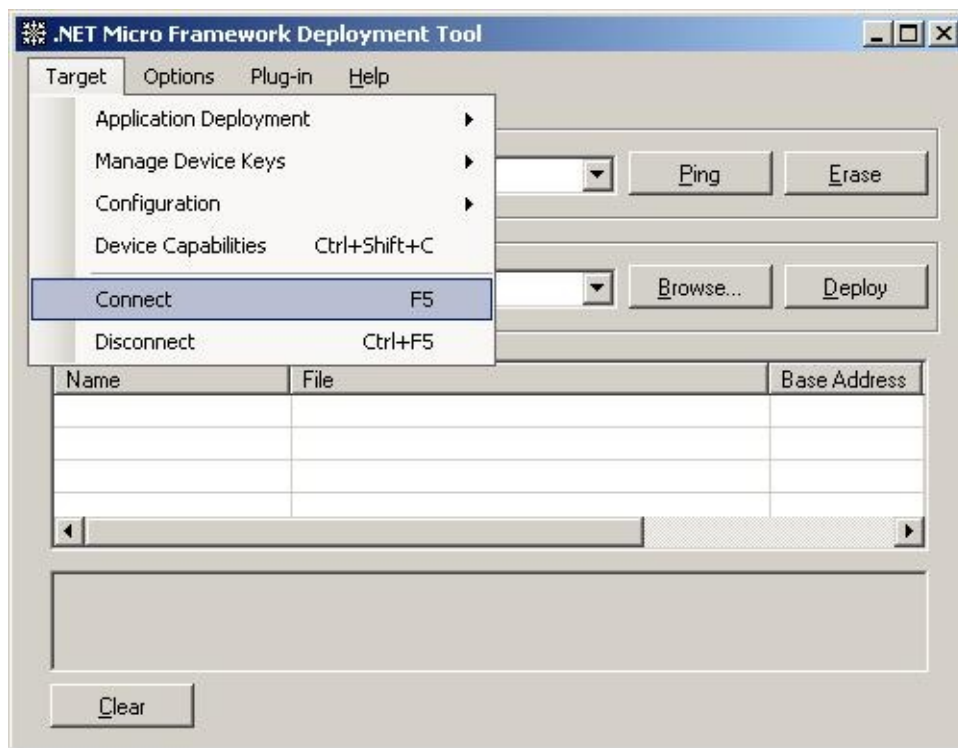
Select the COM Port which you want to use for MFDeploy as shown below (List Box).



**Figure 9.1 .NET Micro Framework Deployment Tool**

### iii) Connect COM Port

Connect MFdeoply.exe to COM Port by selecting Target menu -> Connect option as shown below.



**Figure 9.2 .NET Micro Framework Deployment Tool**

## 9.2 How to make sure if TinyCLR works fine

If you can not deploy the application properly, “MFDeloy.exe” is a good tool to make sure if the TinyCLR works fine.

Turn on the board so that you can see like below log output.

Connecting to COM1...Connected

TinyCLR (Build 4.0.2037.0)

Starting...

Created EE.

Started Hardware.

No debugger!

Create TS.

Loading start at 95000, end b5f88

Attaching file.

Assembly: mscorlib (4.0.2037.0) (3572 RAM - 29944 ROM - 17631 METADATA)

AssemblyRef	=	0 bytes (	0 elements)
TypeRef	=	0 bytes (	0 elements)
FieldRef	=	0 bytes (	0 elements)
MethodRef	=	0 bytes (	0 elements)
TypeDef	=	1032 bytes (	129 elements)
FieldDef	=	232 bytes (	115 elements)
MethodDef	=	1448 bytes (	724 elements)

Attributes	=	0 bytes (	0 elements)
TypeSpec	=	16 bytes (	4 elements)
Resources	=	232 bytes (	29 elements)
Resources Files	=	16 bytes (	2 elements)
Resources Data	=	437 bytes	
Strings	=	967 bytes	
Signatures	=	2015 bytes	
ByteCode	=	10500 bytes	

Attaching file.

Assembly: Microsoft.SPOT.Native (4.0.2037.0) (1064 RAM - 5752 ROM - 4159 METADATA)

AssemblyRef	=	4 bytes (	1 elements)
TypeRef	=	80 bytes (	20 elements)
FieldRef	=	0 bytes (	0 elements)
MethodRef	=	60 bytes (	15 elements)
TypeDef	=	328 bytes (	41 elements)
FieldDef	=	132 bytes (	65 elements)
MethodDef	=	216 bytes (	108 elements)

Attributes	=	48 bytes (	6 elements)
TypeSpec	=	0 bytes (	0 elements)

Resources = 72 bytes ( 9 elements)  
Resources Files = 8 bytes ( 1 elements)  
Resources Data = 747 bytes  
Strings = 207 bytes  
Signatures = 587 bytes  
ByteCode = 413 bytes

Attaching file.

Assembly: Microsoft.SPOT.Hardware (4.0.2037.0) (1752 RAM - 11404 ROM - 7365 METADATA)

AssemblyRef = 8 bytes ( 2 elements)  
TypeRef = 124 bytes ( 31 elements)  
FieldRef = 24 bytes ( 6 elements)  
MethodRef = 120 bytes ( 30 elements)  
TypeDef = 496 bytes ( 62 elements)  
FieldDef = 176 bytes ( 88 elements)  
MethodDef = 444 bytes ( 222 elements)

Attributes = 0 bytes ( 0 elements)  
TypeSpec = 0 bytes ( 0 elements)  
Resources = 0 bytes ( 0 elements)  
Resources Files = 0 bytes ( 0 elements)  
Resources Data = 0 bytes  
Strings = 1329 bytes  
Signatures = 1061 bytes  
ByteCode = 2579 bytes

Attaching file.

Assembly: Microsoft.SPOT.Hardware.SerialPort (4.0.2037.0) (508 RAM - 3440 ROM - 1527 METADATA)

AssemblyRef = 8 bytes ( 2 elements)  
TypeRef = 96 bytes ( 24 elements)  
FieldRef = 0 bytes ( 0 elements)  
MethodRef = 80 bytes ( 20 elements)  
TypeDef = 16 bytes ( 2 elements)  
FieldDef = 32 bytes ( 16 elements)  
MethodDef = 92 bytes ( 46 elements)

Attributes = 0 bytes ( 0 elements)  
TypeSpec = 0 bytes ( 0 elements)  
Resources = 0 bytes ( 0 elements)  
Resources Files = 0 bytes ( 0 elements)  
Resources Data = 0 bytes  
Strings = 667 bytes  
Signatures = 239 bytes  
ByteCode = 1118 bytes

Attaching file.

Assembly: Microsoft.SPOT.IO (4.0.2037.0) (716 RAM - 4432 ROM - 2459 METADATA)

AssemblyRef = 12 bytes ( 3 elements)  
TypeRef = 72 bytes ( 18 elements)

FieldRef = 0 bytes ( 0 elements)  
MethodRef = 96 bytes ( 24 elements)  
TypeDef = 120 bytes ( 15 elements)  
FieldDef = 68 bytes ( 34 elements)  
MethodDef = 140 bytes ( 70 elements)

Attributes = 0 bytes ( 0 elements)  
TypeSpec = 0 bytes ( 0 elements)  
Resources = 0 bytes ( 0 elements)  
Resources Files = 0 bytes ( 0 elements)  
Resources Data = 0 bytes  
Strings = 646 bytes  
Signatures = 335 bytes  
ByteCode = 1199 bytes

Attaching file.

Assembly: System.IO (4.0.2037.0) (1548 RAM - 13264 ROM - 5862 METADATA)

AssemblyRef = 8 bytes ( 2 elements)  
TypeRef = 168 bytes ( 42 elements)  
FieldRef = 36 bytes ( 9 elements)  
MethodRef = 392 bytes ( 98 elements)  
TypeDef = 144 bytes ( 18 elements)  
FieldDef = 76 bytes ( 37 elements)  
MethodDef = 392 bytes ( 195 elements)

Attributes = 0 bytes ( 0 elements)  
TypeSpec = 8 bytes ( 2 elements)  
Resources = 0 bytes ( 0 elements)  
Resources Files = 0 bytes ( 0 elements)  
Resources Data = 0 bytes  
Strings = 356 bytes  
Signatures = 790 bytes  
ByteCode = 6919 bytes

Attaching file.

Assembly: Microsoft.SPOT.Graphics (4.0.2037.0) (388 RAM - 2268 ROM - 1357 METADATA)

AssemblyRef = 8 bytes ( 2 elements)  
TypeRef = 24 bytes ( 6 elements)  
FieldRef = 0 bytes ( 0 elements)  
MethodRef = 20 bytes ( 5 elements)  
TypeDef = 40 bytes ( 5 elements)  
FieldDef = 16 bytes ( 8 elements)  
MethodDef = 96 bytes ( 48 elements)

Attributes = 0 bytes ( 0 elements)  
TypeSpec = 0 bytes ( 0 elements)  
Resources = 0 bytes ( 0 elements)  
Resources Files = 0 bytes ( 0 elements)  
Resources Data = 0 bytes  
Strings = 537 bytes  
Signatures = 293 bytes  
ByteCode = 242 bytes

Attaching file.

Assembly: Microsoft.SPOT.TinyCore (4.0.2037.0) (5080 RAM - 61564 ROM - 23446 METADATA)

AssemblyRef = 16 bytes ( 4 elements)  
TypeRef = 224 bytes ( 56 elements)  
FieldRef = 52 bytes ( 13 elements)  
MethodRef = 456 bytes ( 114 elements)  
TypeDef = 1104 bytes ( 138 elements)  
FieldDef = 728 bytes ( 363 elements)  
MethodDef = 1576 bytes ( 787 elements)

Attributes = 0 bytes ( 0 elements)  
TypeSpec = 4 bytes ( 1 elements)  
Resources = 0 bytes ( 0 elements)  
Resources Files = 0 bytes ( 0 elements)  
Resources Data = 0 bytes  
Strings = 12916 bytes  
Signatures = 3122 bytes  
ByteCode = 25075 bytes

Attaching file.

Assembly: Microsoft.SPOT.Time (4.0.2037.0) (508 RAM - 2976 ROM - 1552 METADATA)

AssemblyRef = 12 bytes ( 3 elements)  
TypeRef = 60 bytes ( 15 elements)  
FieldRef = 0 bytes ( 0 elements)  
MethodRef = 36 bytes ( 9 elements)  
TypeDef = 96 bytes ( 12 elements)  
FieldDef = 40 bytes ( 20 elements)  
MethodDef = 84 bytes ( 41 elements)

Attributes = 0 bytes ( 0 elements)  
TypeSpec = 0 bytes ( 0 elements)  
Resources = 0 bytes ( 0 elements)  
Resources Files = 0 bytes ( 0 elements)  
Resources Data = 0 bytes  
Strings = 895 bytes  
Signatures = 220 bytes  
ByteCode = 403 bytes

Loading Deployment Assemblies.

Resolving.

Total: (12884 RAM - 135044 ROM - 65358 METADATA)

AssemblyRef = 76 bytes ( 19 elements)  
TypeRef = 848 bytes ( 212 elements)  
FieldRef = 112 bytes ( 28 elements)  
MethodRef = 1260 bytes ( 315 elements)  
TypeDef = 3376 bytes ( 422 elements)  
FieldDef = 1500 bytes ( 746 elements)  
MethodDef = 4488 bytes ( 2241 elements)



DebuggingInfo = 2252 bytes

Attributes = 48 bytes ( 6 elements)  
TypeSpec = 28 bytes ( 7 elements)  
Resources Files = 72 bytes ( 3 elements)  
Resources = 304 bytes ( 38 elements)  
Resources Data = 1184 bytes  
Strings = 18520 bytes  
Signatures = 8662 bytes  
ByteCode = 48448 bytes

GC: 1msec 15276 bytes used, 4178820 bytes available

Type 0F (STRING ): 24 bytes  
Type 15 (FREEBLOCK ): 4178820 bytes  
Type 17 (ASSEMBLY ): 15180 bytes  
Type 34 (APPDOMAIN\_HEAD ): 72 bytes

Total: (12884 RAM - 135044 ROM - 65358 METADATA)

AssemblyRef = 76 bytes ( 19 elements)  
TypeRef = 848 bytes ( 212 elements)  
FieldRef = 112 bytes ( 28 elements)  
MethodRef = 1260 bytes ( 315 elements)  
TypeDef = 3376 bytes ( 422 elements)  
FieldDef = 1500 bytes ( 746 elements)  
MethodDef = 4488 bytes ( 2241 elements)

DebuggingInfo = 2252 bytes

Attributes = 48 bytes ( 6 elements)  
TypeSpec = 28 bytes ( 7 elements)  
Resources Files = 72 bytes ( 3 elements)  
Resources = 304 bytes ( 38 elements)  
Resources Data = 1184 bytes  
Strings = 18520 bytes  
Signatures = 8662 bytes  
ByteCode = 48448 bytes

Ready.

Cannot find any entrypoint!

Done.

Waiting for debug commands...

After the above messages, you should try and connect from MFdeploy by clicking “Ping” button from MFdeploy. If you see that the TinyCLR responds, then it means that the TinyCLR is up and running.

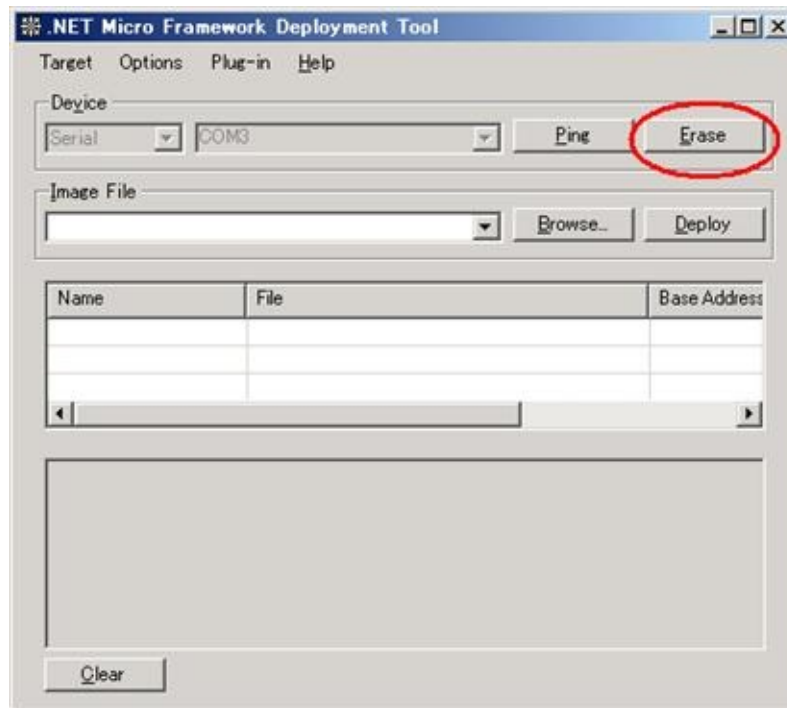
Pinging... TinyCLR



### 9.3 Erase data in the deployment area

Following is the way to erase data in the deployment area.

- 1) Connect the Target Platform referring section 9.1.
- 2) Press Erase button so that erasing starts.



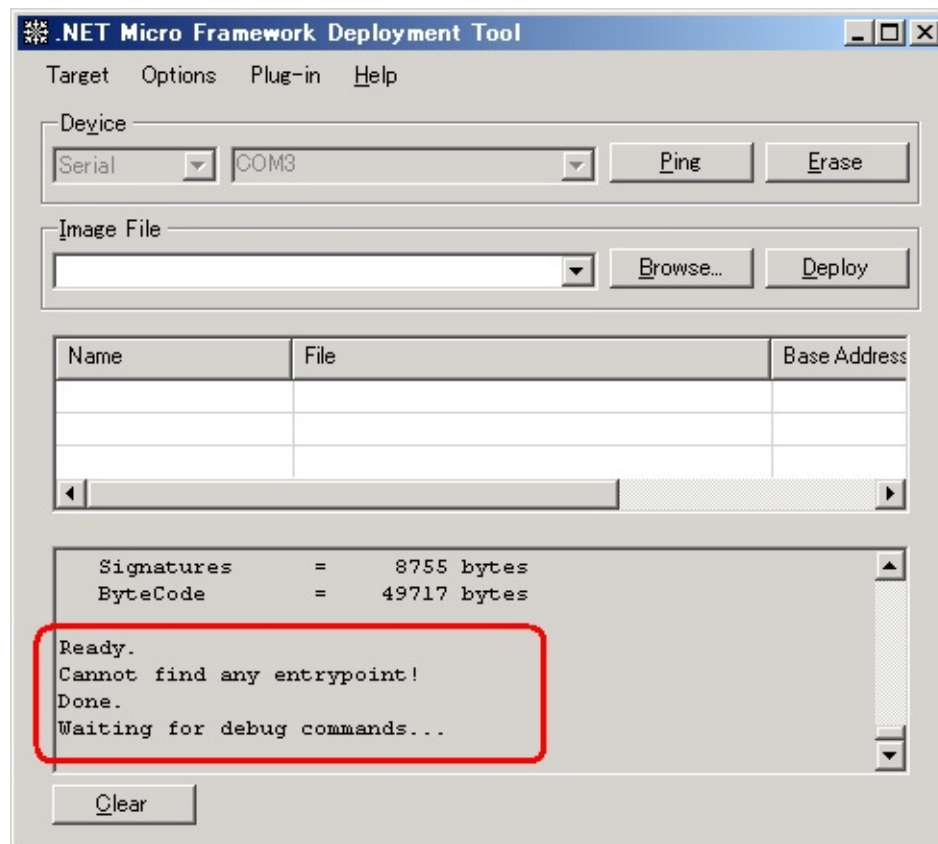
**Figure 9.3 .NET Micro Framework Deployment Tool**

- 3) If below error shows up, don't mind.



**Figure 9.4 Error message**

4) Following message shows up when the data in deployment area has erased.



**Figure 9.5 .NET Micro Framework Deployment Tool**