1. **Overview**

1.1 Scope

This document describes how to use. NET Micro Framework (NETMF) with SH7619 EVB board. This document mainly describes how to setup, build and run SH7619_EVB solution available in the porting kit. This document also describes how to create, build, deploy and run NETMF based applications on SH7619 EVB board using Microsoft visual studio.

1.2 Required Target System

- SH7619 EVB

Please go through the document of the SH7619 EVB before power ON and working with the board.

1.3 Supported Driver

- Serial
- On chip Ethernet
- Timer & Power
- Nor Flash

Following drivers are just sample. In the SH7619 EVB there is no LCD panel and no Key Input.

If you do some wire arrangement, following sample drivers might help you to create driver.

- Display (LCD : 132 x 176 TFT LCD Display (HD66773R) with 262,144 colors)
- Key input using GPIO.

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 (SH7619 EVB)
- Microsoft Visual Studio 2010
- E10A-USB Emulator for SH2 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 "SH7619_EVB" (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

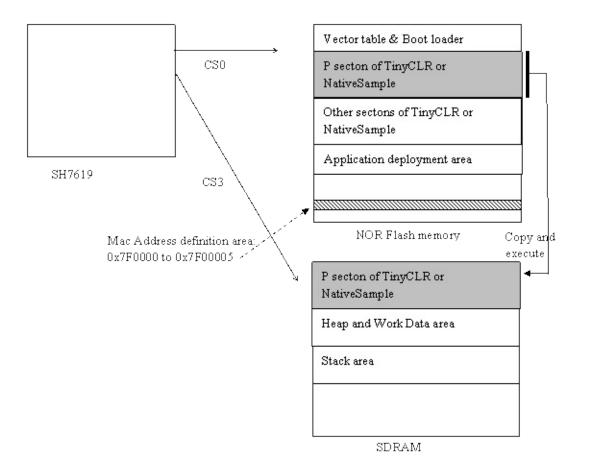
- 1) Open a command-line prompt
- 2) Change the current directory to C:\MicroFrameworkPK_v4_1
- 3) Enter below,

 $setenv_shc "c:\Program Files\Renesas\Hew\Tools\Renesas\Sh\x_y_z*"$

*x_y_z is the version of the compiler being used for example 9_2_0.

3. Memory Map

The memory map of SH7619 EVB .NETMF Porting kit is shown below.





4. How to build and execute

In this section, we shall describe the way to build, download and execute SH7619 EVB 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\SH7619_EVB"

C:\MicroFrameworkPK_v4_1>cd solutions\SH7619_EVB

2) Run

Msbuild dotnetmf.proj /t:build /p:flavor=debug;EnableTcpIp=true;TCP_IP_STACK=LWIP

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

TCP_IP_STACK : LWIP If you want to use RTIP, don't need to add

TCP_IP_STACK.

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

4.2 **Board switch settings**

Set SW1 of the SH7619 EVB board as below to startup from NOR Flash.

SW	Setting	Function
1	ON (Low)	MD0
2	OFF (High)	MD1
3	ON (Low)	MD2
4	ON (Low)	MD3
5	ON (Low)	MD5

The meaning of above setting is below,

- Clock mode

Clock Mode is MODE2.

Each frequency is below,

- Crystal 15.36MHz
- CPU 15.36MHz

FRQCR register is set by software as 0x1103 so CPU frequency will be 122.88MHz.

- BUS 61.44MHz
- Peripheral 30.72MHz
- CS0 Memory bus width 16 Bit
- Endian Big endian

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-2 device group"

2) How to start downloading using E10A-USB

Extract the HEW workspace for SH7619 EVB Porting Kit from file SH7619board.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

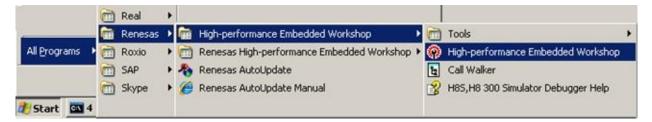


Figure 4.1 Hew menu

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

Specify the workspace as below, and press OK button.

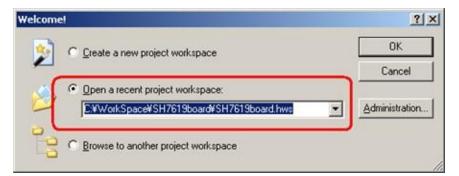


Figure 4.2 Welcome dialog box

In below dialog box, choose "SH7619".

CPU Select	×
CPU Select	
SH7619	•
SH7618	
SH7619	
SH7606 SH7618A	
SH-2_custom_MCU	

Figure 4.3 CPU Select dialog box

When below message shows up, power ON the target board and then press OK button.



Figure 4.4 heu7618 dialog box

Then Hew/E10A-USB will finish connection with SH7619.

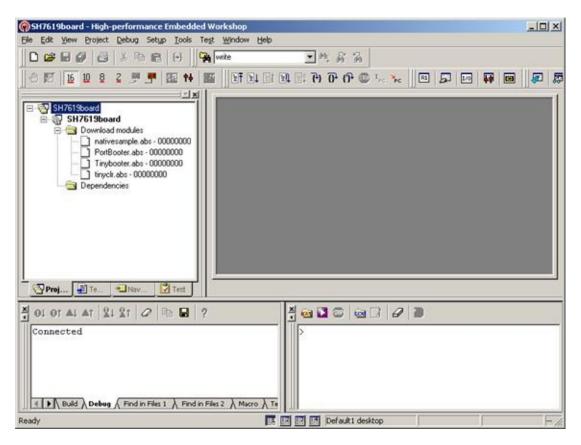


Figure 4.5 Hew window

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.

ebug Settings				?)>		
SessionSH7618_E10A-USB_SYSTEM_(_	Target Options					
	<u>I</u> arget					
-	SH7618 E10A-USB SYST	•				
	Default debug format:					
	Elf/Dwarf2	•				
	Download modules:	Download modules:				
	Filename	Offset Address	Format	Add		
	C:¥MicroFrameworkPK C:¥MicroFrameworkPK C:¥MicroFrameworkPK	00000000	Elf/Dwarf2 Elf/Dwarf2 Elf/Dwarf2	Modify		
	C:¥MicroFrameworkPK		Elf/Dwarf2	Bemove		
	1			<u>U</u> p		
				Dgwn		
			0	K Cancel		

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

Configuration		
General Loading flash memory		
Loading flash memory	C <u>D</u> isable C Enable	
Erasing flash memory	O Djsable O E <u>n</u> able	
<u>F</u> ile name	C:¥WorkSpace¥SH7619boa	Br
Bus width of flash memory	16-bit bus width	
Flash memory erasing time	D'3	minu
Entry point		
All erasing module address	H'AC001000	1
<u>₩</u> riting module address	H'AC001100	1
Access <u>s</u> ize	2	ľ.

Figure 4.7 Configuration dialog box

ii) 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 : SDRAM must be initialized.

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

-Specify the batch file

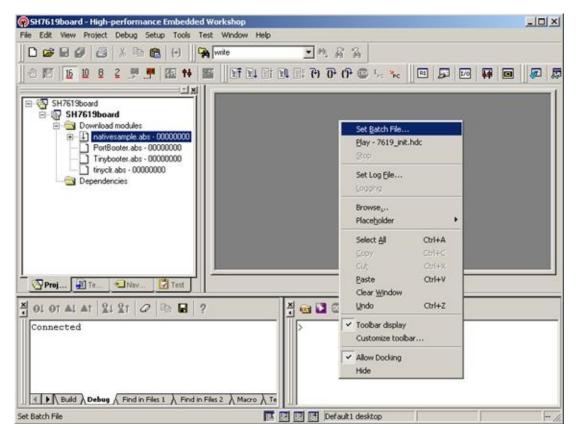
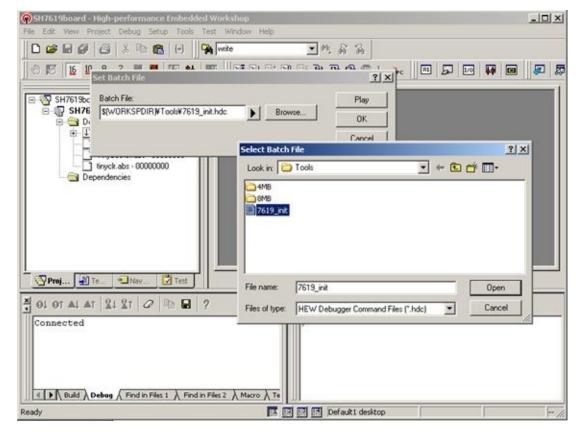


Figure 4.8 Set Batch File

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 below,

Figure 4.9 Set Batch File

-Run the batch file

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

iii) Download module to Flash

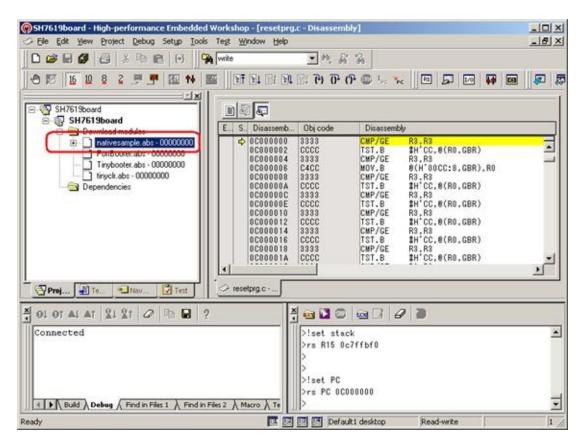


Figure 4.10 Hew window

In order to download, double click on the name of the download module which you want to download to flash. It takes more than 30 seconds.

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

5. 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:

Port:	COM1 - OK
aud rate:	115200 💌
<u>D</u> ata:	8 bit 💌 Cant
P <u>a</u> rity:	none 💌
<u>S</u> top:	1 bit 💌 Hel
	none 🔻

Figure 5.1 Serial port setup

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

File Edit Setup C	ontrol	Window H	elp		
TypeRef	=	1144 bytes	(286 elements)	
FieldRef	=	128 bytes	(32 elements)	
MethodRef	-	1696 bytes	(424 elements)	
TypeDef	=	3808 bytes	(476 elements)	
FieldDef	=	1748 bytes	(870 elements)	
MethodDef	=	5056 bytes	(2524 elements)	
DebuggingInfo	-	2544 bytes			
Attributes	Ξ	48 byte	s (6 elements)	
TypeSpec	=	36 byte	s (9 elements)	
Resources Files	=	72 byte	s (3 elements)	
Resources	=	304 byte	s (38 elements)	
Resources Data	=	1184 byte	s		
Strings	=	22309 byte	s		
Signatures	=	10127 byte:	s		
ByteCode	2	55001 byte	s		
Ready. Cannot find any en	Inno	int l			
Done.	ici ypo	inc.			
/Sdbg∀1S7, ₩		- for detring			100

Figure 5.2 Serial terminal

If you can see above like above, 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.

6. How to set the MAC address

The initial value of MAC address is FF.FF.FF.FF.FF.FF.

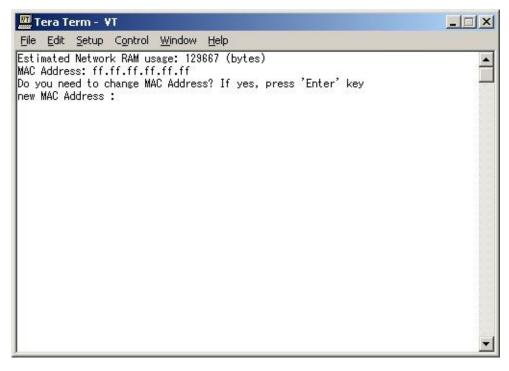
There are two ways for changing MAC address.

6.1 Using Serial Terminal

Using Serial Terminal, you can change the MAC address without modifying any source code.

- 1) Open the Serial terminal and connect the serial port of the SH7619 EVB with PC via Serial cross cable.
- 2) Press Enter key
- 3) Power on the SH7619 EVB with pressing Enter key.

Please don't release the Enter key until you can find following message in terminal window.



4) Enter new MAC address.

If MAC address is expected 1a.e.f.f8.6.f3, type below,

1a.0e.0f.f8.06.f3

6.2 Modifying source file

You can change the MAC address modifying source code.

1) Change the sequence for getting MAC address.

Open

```
C:\MicroFrameworkPK_v4_1\DeviceCode\Targets\Native\sh2\DeviceCode\sh76
So that you can find RTP_BOOL SH7619_EDMAC_open(PIFACE pi).
```

Please remove all of following codes.

```
int addr = MAC_address_area;
```

for(i=0; i<macLen; i++)</pre>

```
g_NetworkConfig.NetworkInterfaces[NETWORK_INTERFACE_INDEX_SH7619EMAC].macAddressBut
= *(volatile char *)(addr+i);
```

```
debug_printf( "MAC Address: %x.%x.%x.%x.%x.%x*f¥n",
```

```
(UINT8)g NetworkConfig.NetworkInterfaces[NETWORK INTERFACE INDEX SH7619EMAC].mai
  (UINT8)g_NetworkConfig.NetworkInterfaces[NETWORK_INTERFACE_INDEX_SH7619EMAC].ma
  (UINT8)g NetworkConfig.NetworkInterfaces[NETWORK INTERFACE INDEX SH7619EMAC].ma
  (UINT8)g_NetworkConfig.NetworkInterfaces[NETWORK_INTERFACE_INDEX_SH7619EMAC].mai
  (UINT8)g_NetworkConfig.NetworkInterfaces[NETWORK_INTERFACE_INDEX_SH7619EMAC].mail
  (UINT8)g NetworkConfig.NetworkInterfaces[NETWORK INTERFACE INDEX SH7619EMAC].mai
debug_printf( "Do you need to change MAC Address? If yes, press 'Enter' key¥r¥n" );
c[0] = 0x0;
for (i=0; i<0xff; i++){
 DebuggerPort_Read( HalSystemConfig.DebugTextPort, c, 1);
 if (c[0] == 0x0d)
  hal printf( "new MAC Address :" );
    while(c[0] == 0x0d) DebuggerPort_Read( HalSystemConfig.DebugTextPort, c, 1);
   for(i=0; i<17; ){
      Events = Events WaitForEvents( SYSTEM EVENT FLAG COM IN, 100 );
     if(Events & SYSTEM_EVENT_FLAG_COM_IN){
        Events_Clear( SYSTEM_EVENT_FLAG_COM_IN );
          DebuggerPort_Read( HalSystemConfig.DebugTextPort, c, 1);
          Buffer[i] = c[0];
          i++:
      }
    for (i=0; i<17; i++) hal printf( "%c",Buffer[i] );</pre>
   hal_printf( "¥r¥n");
   for(i=0; i<macLen; i++) {</pre>
```

```
Buff[i]=SH7619_EDMAC_AtoH(Buffer[i*3])*0x10;
Buff[i]+=SH7619_EDMAC_AtoH(Buffer[i*3+1]);
```

```
}
```

```
for(i=0; i<macLen; i++) {</pre>
```

```
g_NetworkConfig.NetworkInterfaces[NETWORK_INTERFACE_INDEX_SH7619EMAC].macAd }
```

debug_printf("Updating...¥r¥n");

g_AM29DL_16_BS_DeviceTable.InitializeDevice(pBLOCK_CONFIG); g_AM29DL_16_BS_DeviceTable.EraseBlock(pBLOCK_CONFIG,MAC_address_area); g_AM29DL_16_BS_DeviceTable.Write(pBLOCK_CONFIG,MAC_address_area,macLen,Buff,0x0); debug_printf("Done¥r¥n");

```
i=0x100;
}
```

}

2) Set MAC address

Open

 $\label{eq:linear} MicroFrameworkPK_v4_1\Solutions\SH7619_EVB\DeviceCode\Network\Ne So that you can find g_NetworkConfig.$

```
NETWORK CONFIG g NetworkConfig =
            { TRUE },
                               // interface count
            1.
            {
                         {
                                                                                                                                                                       //0.
                                       //SOCK_MAKE_IP_ADDR(192,168, 84, 84), /*ip address */
                                       //SOCK_MAKE_IP_ADDR(255,255,255, 0), /*subnet mask*/
                                       //SOCK_MAKE_IP_ADDR(192,168, 84,251), /*gateway */
                                       //SOCK MAKE IP ADDR(192,168, 84,200), /*dns1
                                                                                                                                                                                                                                                                                                                                            */
                                       //SOCK MAKE IP ADDR(192,168, 84,202), /*dns2
                                                                                                                                                                                                                                                                                                                                            */
                                                                                                                                                                   SOCK_NETWORKCONFIGURATION_FLAGS_DHCP,
                                     SOCK MAKE IP ADDR( 0, 0, 0, 0), /*ip address */
                                     SOCK_MAKE_IP_ADDR(255,255,255, 0), /*subnet mask*/
                                     SOCK_MAKE_IP_ADDR( 0, 0, 0, 0), /*gateway */
                                     SOCK MAKE IP ADDR( 0, 0, 0, 0), /*dns1
                                                                                                                                                                                                                                                                                                            */
                                     SOCK_MAKE_IP_ADDR( 0, 0, 0, 0), /*dns2
                                                                                                                                                                                                                                                                                                            */
                                     // SOCK NETWORKCONFIGURATION FLAGS DHCP |
SOCK NETWORKCONFIGURATION FLAGS DYNAMIC DNS |
SOCK NETWORKCONFIGURATION FLAGS TYPE set(SOCK NETWORKCONFIGURATION FL
                                    // SOCK MAKE IP ADDR LITTLEEND(192,168, 84, 9), /*ip address */
                                    // SOCK MAKE IP ADDR LITTLEEND(255,255,255, 0), /*subnet mask*/
                                     // SOCK MAKE IP ADDR LITTLEEND(192,168, 84, 251), /*gateway */
                                     // SOCK_MAKE_IP_ADDR_LITTLEEND(192,168, 84, 200), /*dns1
                                                                                                                                                                                                                                                                                                                                                                                                                            */
                                     // SOCK_MAKE_IP_ADDR_LITTLEEND(192,168, 84, 202), /*dns2
                                                                                                                                                                                                                                                                                                                                                                                                                             */
                                     SOCK NETWORKCONFIGURATION INTERFACETYPE ETHERNET,
                                                                                        /*mac address length*/
                                     6,
                                     {
                                                                                        /*mac address*/
                                                 0xFF, 
0xFF,
                                                 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 
0xFF,
                                                 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 
0xFF,
                                                 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 
0xFF,
```

} }, }, };

Put the MAC address into above red portion.

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.

	Edit View Debug Team Data Tools Test	100	low <u>H</u> elp <u>Project</u>	Ctrl+Shift+N	·
ĵ	Qpen	080	Web Site Team Project File	Shift+Alt+N Ctrl+N	Solution Explorer 7
	Save Selected Items CtrI+S Save Selected Items As. Save All CtrI+Shift+S		Project From Existing Code		
	Export Template. Source Control				
1	Page Setup. Print. Ctri+P				
	Recent Eiles Recent Projects and Solutions				
	E <u>x</u> it Alt+F4				
1	cror List	0		~ ₽)	<
	0 Errors (1) 0 Warnings (1) 0 Messages Description File #	5	Line 🔺 Co., 🔺	Project 🔺	
	🔧 Error List 🔲 Output 🖳 Find Results 1				🖏 Sol••• 📑 Te••• 💐 Cla

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.

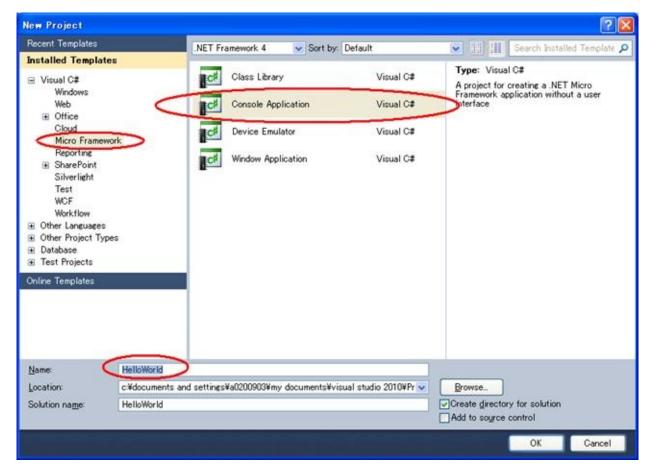


Figure 7.2 New Project

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....

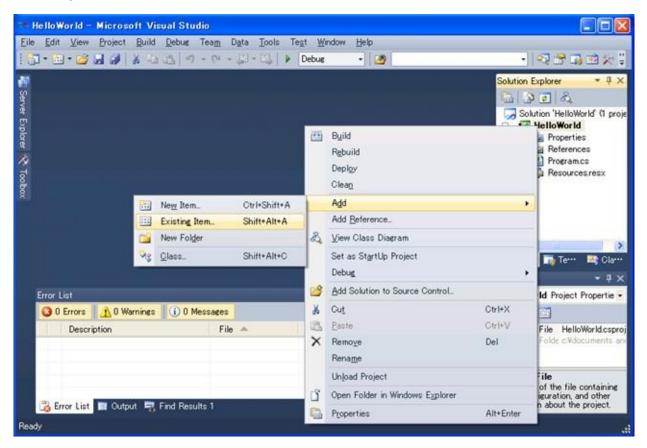


Figure 7.3 Microsoft Visual Studio

Select the below file,

 $C:\MicroFrameworkPK_v4_1\Product\Sample\HelloWorld\Main.cs$

And	press	Add	button.
-----	-------	-----	---------

Add Existing Iter	m - HelloWorld				×
Look jn:	HelloWorld		•	G 👂 📴 🖽 -	
Desktop Projects My Computer	HelloWorld.csp	roj			
	Object <u>n</u> ame:	Main.cs		•	<u>A</u> dd ▼
	Objects of type:	All Files (*.*)			Cancel

Figure 7.4 Add Existing Item dialog box

iv) Remove unnecessary file

Remove Program.cs file from the HelloWorld project.

👐 HelloWorld – Microsoft Visual Stud	io						
Eile Edit View Project Build Debug		Tegt <u>₩</u> indow	Help	81			
10-11-12 1 2 1 × 4 15 10 -	· G · Ð · Ū I	Debug	• 12	11			J B X I
🏝 Server Explorer 🛠 Toobox						Solution Explorer	lloWorld' (1 proje orld erties rences
× box				ũ	Open Open With		ces.resx
			-	2	View <u>C</u> ode <u>V</u> iew Class Diagram	F7	
					Exclude From Proje	ct	• 📑 Cla
Error List				×	Cu <u>t</u> Cop <u>y</u>	Ctrl+X Ctrl+C	• 4 ×
0 Errors 0 Warnings 0 Me	ssares			×	Delete	Del	at hopertie
Description	File 🔺	Line 🔺	Co	-	Rename		lloWorld.csproj
				1	Properties	Alt+Enter	documents and
🐻 Error List 🔳 Output 📑 Find Result	s 1					Project File The name of the build, configuratio information about	n, and other
Ready							đ

Figure 7.5 Microsoft Visual Studio

v) Build solution

= HelloWorld - Microso	oft V	isual Stud	lio										
<u>Eile Edit View Project</u>	Build	Debug	Tea <u>m</u> D) <u>a</u> ta <u>I</u> oo	ls Te <u>s</u> t	₩indow	Help				1010		
i 🔂 • 🖻 • 🗳 🗐 🥔 🔰	凿	Build Solut	tion		F6		- 6	2			- 4	2 🗗 🖬 🖻	ジョン
The Server		Bebuild So Deploy Sol Olean Solu	lution								and an	lorer [] [] කිද n 'HelloWorld'	• ₽ ×
📲 Server Explorer 🛠 Toobox	**	Build Hello Rebuild He Deploy Hel Clean Hello	HoWorld NoWorld		Shift+F6						8- 12 8- 12 9- 12	HoWorld Properties References Main.cs Program.cs	ti proje
obox		Batch Buil Configurati		er_							@- 5 5	Resources.re	sx
											< ≪⊒ Soh…	Te*** 🗵	Cla…
											Properties	2	• 7 ×
Error List			_							• 1 >		s File Propert	ies 🔹
3 0 Errors 🔒 0 Wa	mines	1 0 M	essages								21	1	
Description			File	*		Line 🔺	Co_	*	Project	*			ору
强 Error List 🔳 Outpu		Find Resul	lts 1									on relates to th ient processe	
Ready			inter										
1.000)													

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.

Application	Configuration: Active (Debug)	
Build	Platform: Active (Any CPU)	
Build Events		
)ebug	Deployment Transport:	^
Resources	Serial	
Reference Paths	Device:	_
NET Micro Framework		
	Generate native stubs for internal methods	

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.

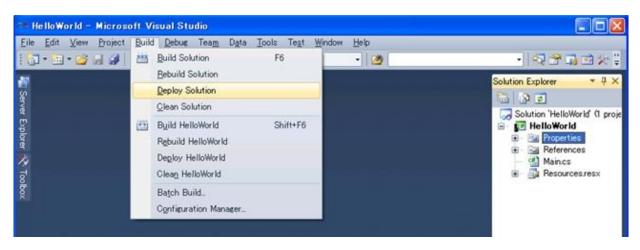


Figure 7.9 Microsoft Visual Studio

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.

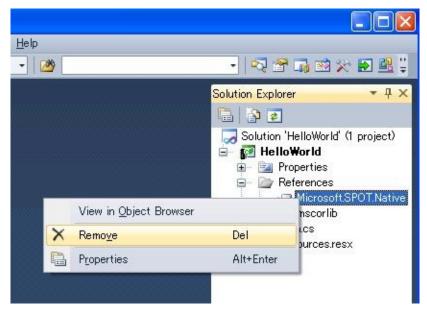


Figure 7.10 Solution Explorer

ii) Add the Reference for Big-Endian

Right Click on the "References" as shown below and select "Add reference..." option form the popup menu.

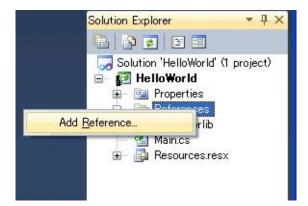


Figure 7.11 Solution Explorer

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 F	Framework\v4.1\Assemblies\be
---	------------------------------

Reference		
	s Recent Browse	
Look jn: 🗲	-	- 3 🕫 🗁 🖽 -
-	SPOT.Ink.dll	Microsoft.SPOT.
	SPOT.IO.dll	Microsoft.SPOT.
	SPOT.Native.dll	Microsoft.SPOT.
Microsoft.	SPOT.Net.dll	Microsoft.SPOT.
👏 Microsoft.	SPOT.Net.Security.dll	🔊 mscorlib.dll
🔊 Microsoft.	SPOT.Time.dll	🔊 System.dll
🖄 Microsoft.	SPOT.TinyCore.dll	🔊 System. Http. dll
🔊 Microsoft.	SPOT.Touch.dll	👏 System.IO.dll
•		•
File <u>n</u> ame:	Microsoft.SPOT.Native.dll	•
Files of <u>type</u> :	Component Files (*.exe *.dll)	Ţ
		OK Cancel

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.

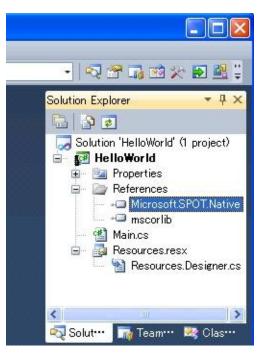


Figure 7.13 Solution Explorer

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

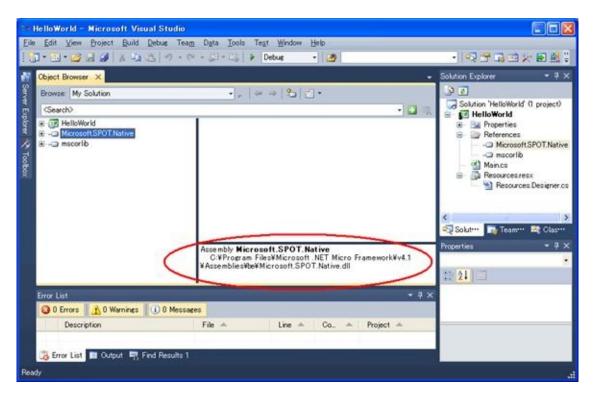


Figure 7.14 Object Browser

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

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.

... lines appropriate for your application <Import Condition="" Project="\$(SPOCLIENT)\Framework\Features\Diagnostics.featureproj" /> <Import Condition="" Project="\$(SPOCLIENT)\Framework\Features\Core.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.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 MFDeply.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 MDFelpoy.exe but also all the managed tests in the PK as well.

MFdeploy is located in

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

3) Run the MFDeply.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 MFDeply.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).

	COM1 COM1	Eing	<u>E</u> rase
		Browse	<u>D</u> eploy
Name	File		Base Address
•			,



iii) Connect COM Port

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

arget Options Plug-in <u>H</u> e Application Deployment Manage Device Keys Configuration			<u>P</u> ing	<u>E</u> rase
Connect	Shift+C F5 Itrl+F5	J	<u>B</u> rowse	<u>D</u> eploy
Name File				Base Address
•			1	D

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)
                                  0 elements)
 MethodRef
               =
                      0 bytes (
 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)
                =
                                  29 elements)
 Resources
               =
                    232 bytes (
                                    2 elements)
 Resources Files =
                       16 bytes (
 Resources Data =
                      437 bytes
  Strings
                  967 bytes
             =
                   2015 bytes
 Signatures
               =
  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)
                     60 bytes (
                                  15 elements)
 MethodRef
                =
 TypeDef
                   328 bytes (
                                 41 elements)
              =
 FieldDef
                   132 bytes (
                                 65 elements)
              =
 MethodDef
                =
                     216 bytes (
                                  108 elements)
 Attributes
                    48 bytes (
                                 6 elements)
              =
                     0 bytes (
 TypeSpec
                                  0 elements)
                =
```

```
Resources
                     72 bytes (
                                  9 elements)
               =
 Resources Files =
                       8 bytes (
                                   1 elements)
                      747 bytes
 Resources Data =
                  207 bytes
 Strings
             =
 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)
                    120 bytes (
 MethodRef
                                   30 elements)
               =
              =
                   496 bytes (
                                 62 elements)
 TypeDef
 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)
                       0 bytes (
                                   0 elements)
 Resources Files =
 Resources Data =
                       0 bytes
 Strings
                 1329 bytes
             =
                   1061 bytes
 Signatures
               =
 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)
                     0 bytes (
                                 0 elements)
 Resources
               =
 Resources Files =
                       0 bytes (
                                   0 elements)
                       0 bytes
 Resources Data =
                  667 bytes
 Strings
             =
                    239 bytes
 Signatures
               =
 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)
              =
                     96 bytes (
 MethodRef
                                   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)
                     0 bytes (
 Resources
               =
                                  0 elements)
                       0 bytes (
                                    0 elements)
 Resources Files =
 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)
              =
                                 37 elements)
 FieldDef
              =
                    76 bytes (
 MethodDef
                =
                     392 bytes (
                                   195 elements)
 Attributes
              =
                     0 bytes (
                                 0 elements)
 TypeSpec
                      8 bytes (
                                  2 elements)
                =
 Resources
               =
                     0 bytes (
                                  0 elements)
                                    0 elements)
 Resources Files =
                       0 bytes (
                        0 bytes
 Resources Data =
 Strings
                  356 bytes
             =
                    790 bytes
 Signatures
               =
 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)
                      0 bytes (
 TypeSpec
                                  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)
                     4 bytes (
 TypeSpec
                                  1 elements)
               =
 Resources
               =
                     0 bytes (
                                 0 elements)
                       0 bytes (
                                   0 elements)
 Resources Files =
 Resources Data =
                       0 bytes
                 12916 bytes
 Strings
             =
 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
                                  41 elements)
                =
                     84 bytes (
                                 0 elements)
 Attributes
              =
                    0 bytes (
 TypeSpec
                     0 bytes (
                                  0 elements)
               =
 Resources
               =
                     0 bytes (
                                 0 elements)
 Resources Files =
                       0 bytes (
                                   0 elements)
 Resources Data =
                       0 bytes
 Strings
             =
                  895 bytes
                    220 bytes
 Signatures
               =
 ByteCode
               =
                    403 bytes
Loading Deployment Assemblies.
Resolving.
Total: (12884 RAM - 135044 ROM - 65358 METADATA)
  AssemblyRef
                      76 bytes (
                                   19 elements)
                =
 TypeRef
                   848 bytes (
                                212 elements)
              =
                   112 bytes (
 FieldRef
                                28 elements)
              =
 MethodRef
               =
                    1260 bytes (
                                  315 elements)
 TypeDef
                   3376 bytes (
                                 422 elements)
              =
 FieldDef
              =
                  1500 bytes (
                                 746 elements)
                    4488 bytes ( 2241 elements)
 MethodDef
                =
```

```
DebuggingInfo =
                     2252 bytes
 Attributes
                   48 bytes (
                                6 elements)
              =
 TypeSpec
               =
                    28 bytes (
                                 7 elements)
                      72 bytes (
 Resources Files =
                                   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)
              =
                  112 bytes (
 FieldRef
              =
                                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)
              =
                    28 bytes (
 TypeSpec
                                 7 elements)
               =
                      72 bytes (
                                   3 elements)
 Resources Files =
 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 Terget Plarform referring section 9.1.
- 2) Press Erase button so that erasing starts.

mage File		<u>B</u> rowse	Deploy
Name	File		Base Addres
d			<u>.</u>



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.

檾.NET Micro Framewo	rk Deployment Tool		_ 🗆 🗵
Target Options Plug ⊢Device	-in <u>H</u> elp		
Serial COM3	ſ	<u>P</u> ing	<u>E</u> rase
Image File			
]	Browse	Deploy
Name	File		Deve Address
iname	File		Base Address
•			F
Signatures ByteCode	= 8755 bytes = 49717 bytes		1
Ready. Cannot find any en Done. Waiting for debug			Ţ
<u>C</u> lear			

Figure 9.5 .NET Micro Framework Deployment Tool

10. Wiring arrangement

In this section, the wiring arrangement for hardware for existing LCD driver and Keypad driver is described.

1) Wiring arrangement for LDC panel

The 132 x 176 TFT LCD Display (HD66773R) is interfaced to area CS6B of the SH7619 Bus State Controller (BSC). 3.3V power is provided by the SH7619 board, while 5V power is provided by separate DC power supply connected to the LCD Expansion board. A 40-pin header connector is used for the LCD module interface; the pin assignment for the 40-pin header is shown below.

Pin No.	Signal Name	I/O	Pin No.	Signal Name	I/O
1	GND	_	21	D3	I/O
2	/RESET	OUT	22	D2	I/O
3	3.3V supply	_	23	GND	_
4	NC	_	24	D1	I/O
5	NC	_	25	D0	I/O
6	D15	I/O	26	GND	-
7	D14	I/O	27	/LCD-RD	OUT
8	GND	—	28	/LCD-WR	OUT
9	D13	I/O	29	LCD-RS1	OUT
10	D12	I/O	30	LCD-RS2	OUT
11	D11	I/O	31	GND	_
12	D10	I/O	32	3.3V supply	_
13	GND	_	33	5V supply	_
14	D9	I/O	34	5V supply	_
15	D8	I/O	35	GND	_
16	D7	I/O	36	/LCD-CS1	OUT
17	D6	I/O	37	/LCD-CS2	OUT
18	GND	_	38	LCD-TMG1	IN
19	D5	I/O	39	LCD-TMG2	IN
20	D4	I/O	40	LCD-DIR	OUT

Figure 10.1 LCD header pin assignment

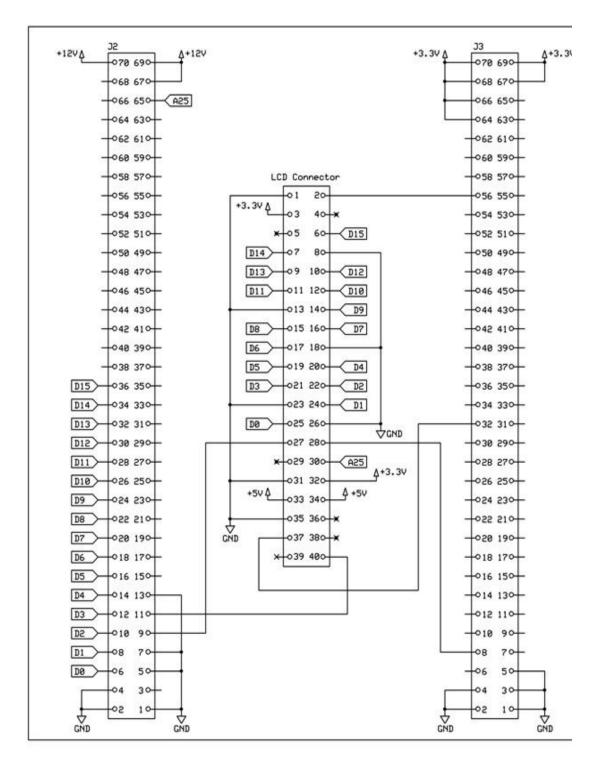


Figure 10.2 Schematic of LCD interface

2) Wiring arrangement for GPIO for Button Keypad

In this Porting Kit, it is assumed that a 25-button keypad matrix is connected to ten GPIOs of the SH7619. Five GPIOs (PC08, PC10, PC09, PC11, and PC13) should be configured for output. The other five GPIOs (PC00, PC01, PC02, PC03, and PC17) should be configured for input and they have pull-up resistors meaning the buttons are active low.

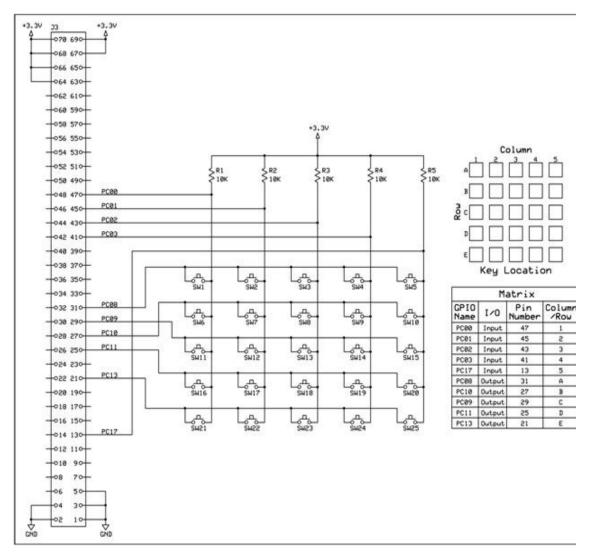


Figure 10.3 GPIO button configuration and perspective orientation

3) Wiring arrangement for secondary Serial Interface

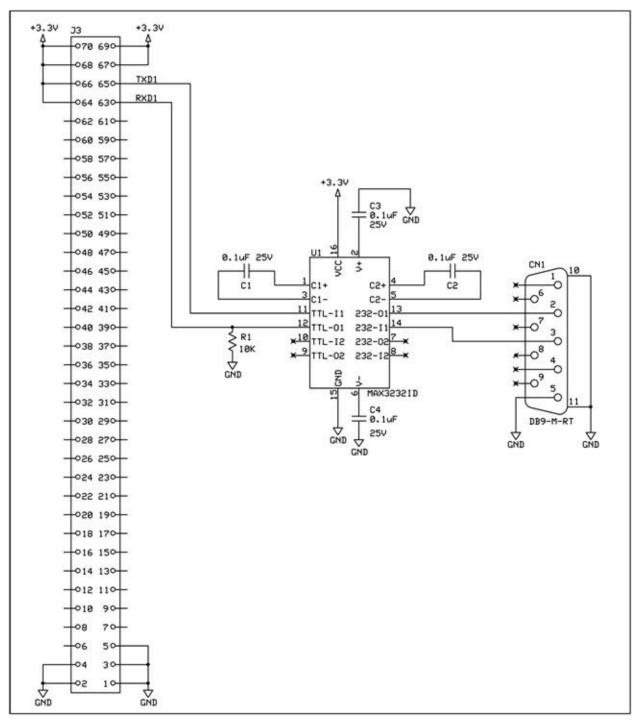


Figure 10.4 Schematic of Secondary Serial Interface