How do I read data from or write data to EEPROM of XPAC

Applies to:

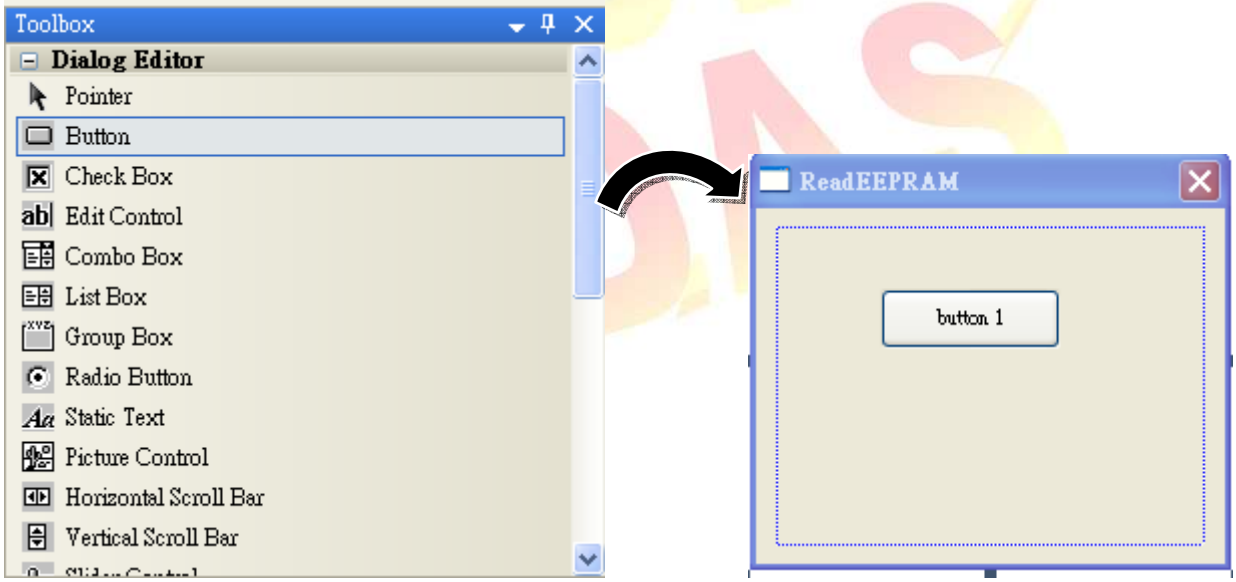
Platform	*OS version*	*XPAC utility version*
XPAC series	*All versions (WinCE6)*	*All versions*

The XPAC SDKs provides a complete solution to integrate with XPAC and compatible with Visual C#, Visual Basic .net and C++.

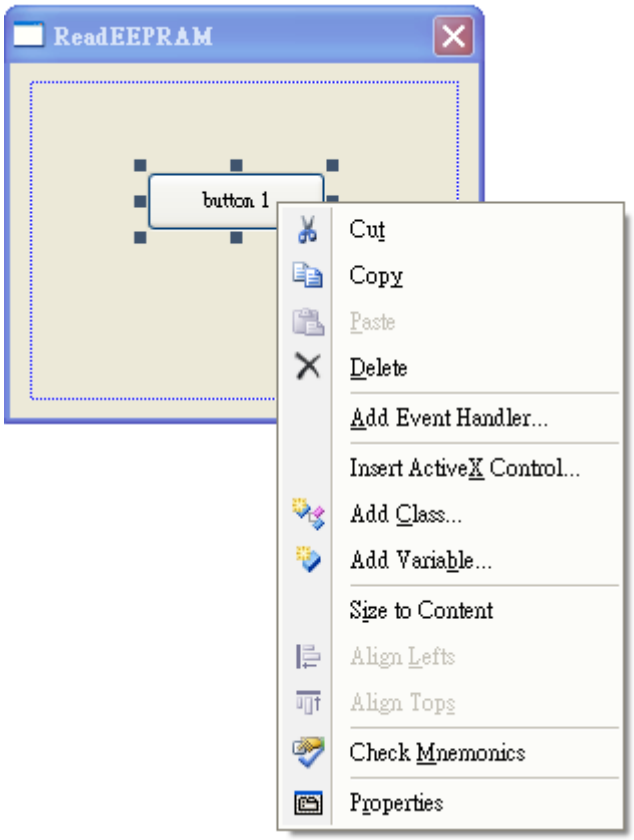
The example using MFC、C# and VB.Net demonstrates how to read and write data in EEPROM, please perform the following steps to build a program.

Using MFC to read and write data in EEPROM

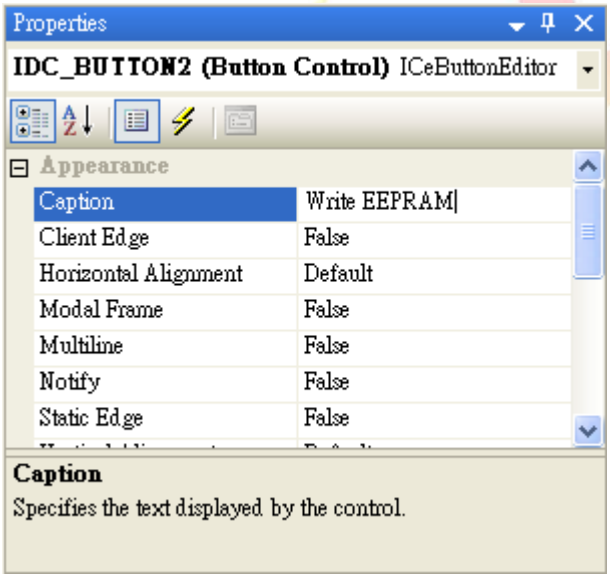
Step 1: From the Toolbox, drag a Button control onto the form



Step 2: Right-click the Button control, and then click Properties



Step 3: In the Properties window, type “Write EEPROM”, and press ENTER to set the Caption property.



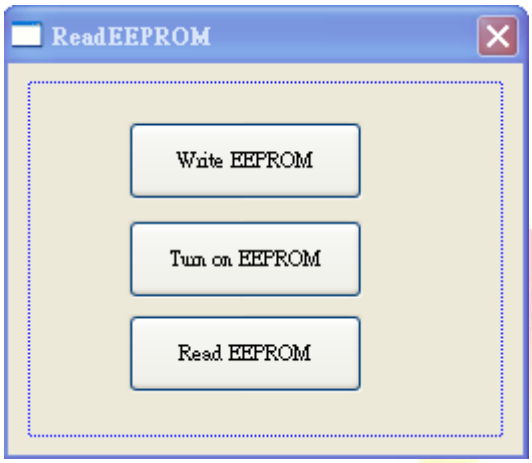
Step 4: Repeat the Step 1~3 to add a button named as “Read EEPROM”.

Step 4: Add “#include "XPacSDK_CE.h" in main file.

```
// ReadSRAMDlg.cpp : implementa
//

#include "stdafx.h"
#include "ReadSRAM.h"
#include "ReadSRAMDlg.h"
#include "XPacSDK_CE.h"
```

Step 5: Double-click the buttons on the form



Step 6: Inserting the following code.

Insert following code in the click event of “Turn on EEPROM” button.

```
void CReadEEPROMDlg::OnBnClickedButton3()
{
    // TODO: Add your control notification handler code here
    pac_EnableEEPROM(true);
}
```

Using “pac_EnableEEPROM” to turn on EEPROM, this function parameter is turn on/off EEPROM.

Syntax

```
void pac_EnableEEPROM(bool);
```

Insert following code in the click event of “Write EEPROM” button.

```
void CReadSRAMDlg::OnBnClickedButton2()
{
    // TODO: Add your control notification handler code here
    BYTE data='a';

    pac_WriteMemory(0,&data,1,1);

    printf("write data: %c\n",data);
}
```

Using “pac_WriteMemory” to write data to EEPROM, 1st parameter of this function is memory address, 2nd parameter is write data, 3rd parameter is data length and 4th parameter is memory type (0: SRAM, 1:EEPROM).

Syntax

```
bool pac_WriteMemory(DWORD address, LPBYTE lpBuffer, DWORD dwLength, int mem_type);
```

Insert following code in the click event of “Read EEPROM” button.

```
void CReadSRAMDlg::OnBnClickedButton1()
{
    // TODO: Add your control notification handler code here
    BYTE data=0;

    pac_ReadMemory(0,&data,1,1);

    printf("read data: %c\n",data);
}
```

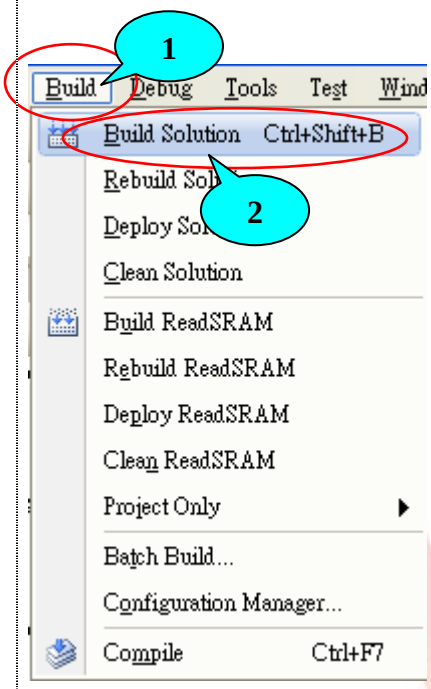
Using “pac_ReadMemory” to read data from EEPROM, 1st parameter of this function is memory address, 2nd parameter is read data buffer, 3rd parameter is data length and 4th parameter is memory type (0: SRAM, 1:EEPROM).

Syntax

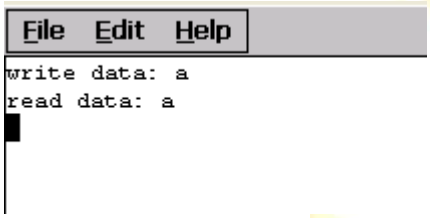
```
bool pac_ReadMemory(DWORD address, LPBYTE lpBuffer, DWORD dwLength, int mem_type);
```

Step 7: Build and execute

Click “Build”->”Build Solution” to build the project, and a execute file will be obtained in the project folder. Put this execute file in your XPAC and execute it.



This program results the following picture.



Tips & Warnings

Refer to the FAQ documents below to upload the execute file to XPAC.

- X5-02_How_to_debug_XPAC_programs_in_Visual_Studio_2005(2008)_online_through_the_TCPIP_english
- X5-27_How to write a MFC application with XPAC SDK in visual studio 2005
- X5-30_How to write a MFC application with XPAC SDK in visual studio 2008

Using C# to read and write data in EEPROM

Step 1: From the Toolbox, drag three buttons control onto the form. Three buttons text properties are “Read EEPROM”, “Write EEPROM” and “Turn on EEPROM”. (The steps are the same with the Step 1~3 of **Using MFC to read and write data in EEPROM**).

Step 2: Get the XPacNet.dll and copy it to the project folder. The XPacNet.dll can be obtained from any C# demo program that has been provided on the CD or by downloading the latest version from ICP DAS web site.

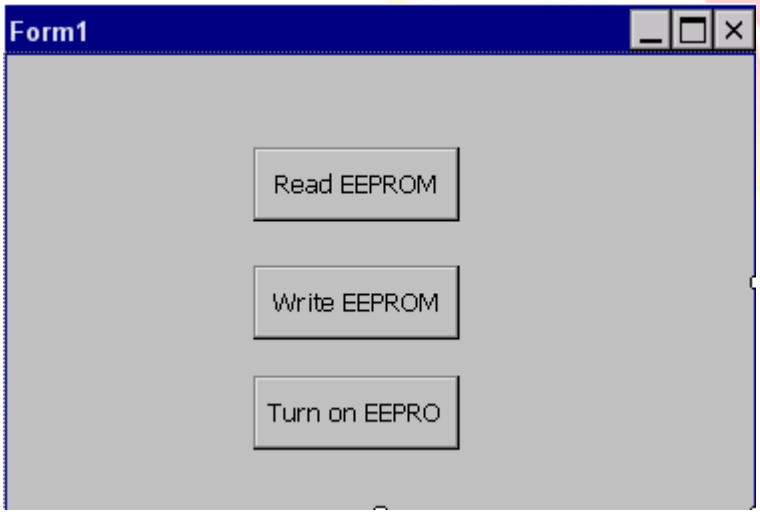
1. CD:\SDK\XPacNET
2. ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/xpacnet/
3. ftp://ftp.icpdas.com/pub/cd/XP-8000-Atom-CE6/SDK/XPacNet/

Tips & Warnings

Refer to the FAQ documents below to add XPacNet.dll to the project.

- X5-28_How to write a C#.net application with XPAC SDK in visual studio 2005
- X5-31_How to write a C#.net application with XPAC SDK in visual studio 2008
-

Step 3: Double-click the buttons on the form



Step 4: Inserting the following code

Insert following code in the click event of “Turn on EEPROM” button.

```
private void button3_Click(object sender, EventArgs e)
{
    XPacNET.XPac.pac_EnableEEPROM(true);
}
```

Using “pac_EnableEEPROM” to turn on EEPROM, this function parameter is used to turn on/off EEPROM.

Syntax

```
void pac_EnableEEPROM(bool);
```

Insert following code in “Write EEPROM” button click event.

```
private void button2_Click(object sender, EventArgs e)
{
    byte[] data = new byte[20];
    data[0] = Convert.ToByte('a');
    XPacNET.XPac.pac_WriteMemory(0, data, 20, 0);

}
```

Using “pac_WriteMemory” to write data to EEPROM, the 1st parameter is the memory address, 2nd parameter is write data, 3rd parameter is data length and 4th parameter is memory type (0: SRAM, 1:EEPROM).

pac_WriteMemory” Syntax

```
bool pac_WriteMemory(uint index, byte[] Buffer, uint Length, int mem_type);
```

Insert following code in the click event of “Read EEPROM” button.

```
private void button1_Click(object sender, EventArgs e)
{
    byte[] data = new byte[20];
    XPacNET.XPac.pac_ReadMemory(0, data, 1, 0);
    string str = Encoding.ASCII.GetString(data, 0, 20);
    MessageBox.Show("Read memory: "+ str);
}
```

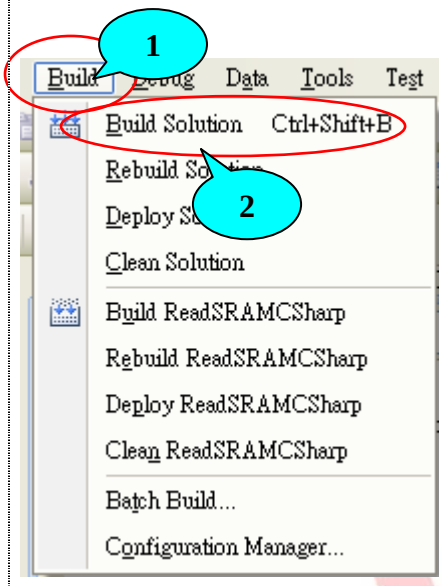
Using “pac_ReadMemory” to reading data from EEPROM, the 1st parameter is memory address, 2nd parameter is read data buffer, 3rd parameter is data length and 4th parameter is memory type (0: SRAM, 1:EEPROM).

pac_ReadMemory” Syntax

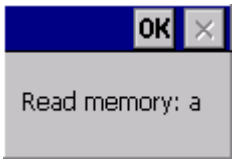
```
bool pac_ReadMemory(uint index, byte[] Buffer, uint Length, int mem_type);
```

Step 5: Build and execute

Click “Build”->”Build Solution” to build the project, and a execute file will be obtained in the project folder. Put this execute file in your XPAC and execute it.



This program results the following picture.



Tips & Warnings

Refer to the FAQ documents below to upload the execute file to XPAC.

- X5-02_How_to_debug_XPAC_programs_in_Visual_Studio_2005(2008)_online_through_the_TCPIP_english
- X5-28_How to write a C#.net application with XPAC SDK in visual studio 2005
- X5-31_How to write a C#.net application with XPAC SDK in visual studio 2008

Using VB.Net to read and write data in EEPROM

Step 1: From the Toolbox, drag three buttons control onto the form. Three buttons text properties are “Turn on EEPROM”、“Read EEPROM” and “Write EEPROM”. (The steps are the same with the Step 1~3 of **Using MFC to read and write data in EEPROM**).

Step 2: Get the XPacNet.dll and copy it to the project folder. The XPacNet.dll can be obtained from any VB.Net demo program that has been provided on the CD or by downloading the latest version from ICP DAS web site.

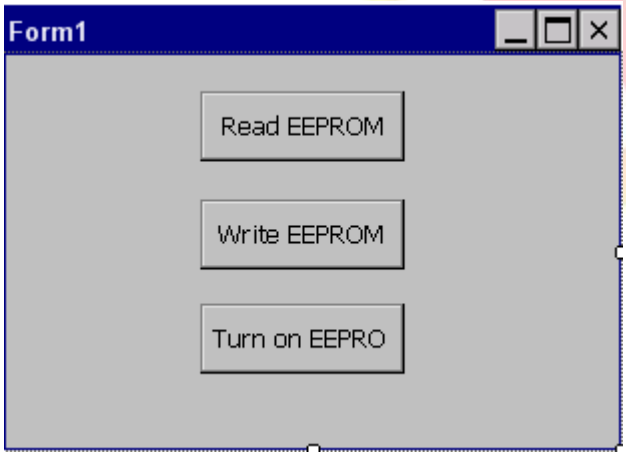
1. CD:\SDK\XPacNET
2. ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/xpacnet/
3. ftp://ftp.icpdas.com/pub/cd/XP-8000-Atom-CE6/SDK/XPacNet/

Tips & Warnings

Refer to the FAQ documents below to add XPacNet.dll to the project.

- X5-29_ How to write a VB.net application with XPAC SDK in visual studio 2005
- X5-32_How to write a VB.net application with XPAC SDK in visual studio 2008

Step 3: Double-click the buttons on the form



Step 4: Inserting the following code

Insert following code in the click event of “Turn on EEPROM” button.

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    XPacNET.XPac.pac_EnableEEPROM(True)
End Sub
```

Using “pac_EnableEEPROM” to turn on EEPROM, this function parameter is used to turn on/off EEPROM.

"pac_EnableEEPROM" Syntax

```
void pac_EnableEEPROM(bEnable AS Boolean);
```

Insert following code in the click event of "Write EEPROM" button.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim data(20) As Byte
    Dim encoding As New System.Text.UTF8Encoding()
    data = encoding.GetBytes("a")
    XPacNET.XPac.pac_WriteMemory(0, data, 20, 0)
End Sub
```

Using "pac_WriteMemory" to write data to EEPROM, the 1st parameter is memory address, 2nd parameter is write data, 3rd parameter is data length and 4th parameter is memory type (0: SRAM, 1:EEPRAM).

pac_WriteMemory" Syntax

```
bool pac_WriteMemory(index AS UInteger, Buffer() AS byte, Length AS UInteger,
mem_type AS Integer);
```

Insert following code in the click event of "Read EEPROM" button.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim data(20) As Byte
    Dim str As String
    Dim enc As New System.Text.UTF8Encoding()
    XPacNET.XPac.pac_ReadMemory(0, data, 20, 0)
    str = enc.GetString(data, 0, 20)
    MsgBox("read data:" + str)
End Sub
```

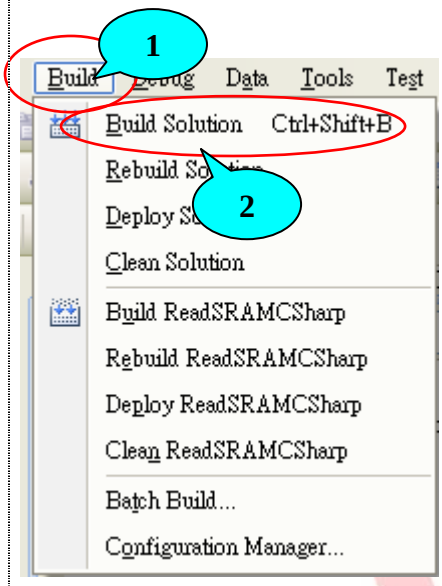
Using "pac_ReadMemory" to read data from EEPROM, the 1st parameter is memory address, 2nd parameter is read data buffer, 3rd parameter is data length and 4th parameter is memory type.

pac_ReadMemory" Syntax

```
bool pac_ReadMemory(uint index, byte[] Buffer, uint Length, int mem_type);
```

Step 5: Build and execute

Click “Build”->”Build Solution” to build the project, and a execute file will be obtained in the project folder. Put this execute file in your XPAC and execute it.



This program results the following picture.

Tips & Warnings

Refer to the FAQ documents below to upload the execute file to XPAC.

- X5-02_How_to_debug_XPAC_programs_in_Visual_Studio_2005(2008)_online_through_the_TCPIP_english
- X5-29_ How to write a VB.net application with XPAC SDK in visual studio 2005
- X5-32_How to write a VB.net application with XPAC SDK in visual studio 2008