

Cross-platform Transplant of Embedded Smart Devices

Jing Zhang

Department of Computer Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China
Email: zj1977@mail.gdufs.edu.cn

XinGuang Li

Department of Computer Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China
Email: lxx@mail.gdufs.edu.cn

Abstract—In the field of embedded software development, the total involvement of application software transplanted in a variety of different platforms occupies more parts than expected in a whole project. In the face of smart devices based on different operating systems and hardware platforms, it's of great significance to enhance the repeatability of application source code and the Cross-platform portability. Procedures for intelligent devices were designed according to the features of Windows Embedded CE6.0 as well as the characteristics of Visual Studio.net 2005, that the build environment. FmodMp3 player program was designed with managed language and transplanted to different intelligent devices, then the goal of cross-platform transplantation, that "code once written then ran in different platform" achieved. Paper also gave some advice about how to improve decompile capacity of managed programs.

Index Terms—Windows Embedded CE 6.0; Intelligent devices; VS2005; Cross-platform transplantation; Fmodmp3 player

I. INTRODUCTION

Various kinds of intelligent device are of great deference in user interface and performance, so, application development should be adapted to different device limit and demand, which will increase the operating costs. If the application can be crossing-platform transplanted in intelligent device based different OS and hardware platform, universal and portability of the application will be enhanced and development will be more rapidly and efficiently. Windows Embedded CE 6.0, promoted by Microsoft recently, supports windows .NET Compact Framework 2.0 to manage and develop the application, and supply Win32, MFC, ATL, WTL and STL to user. Cross-platform transplantation is not difficult any more thanks to the large and complete development environment supporting.

II. DEVELOPMENT ENVIRONMENT THE INTELLIGENT DEVICE DESIGN RELIES ON

The development of Windows Embedded CE 6.0 includes OS and application development. While, the OS development comprises the OS customization, drive

development and other bottom design demanded. The application development mainly refers to procedure development close to practical application.

Development Environment includes hardware and software platform. The hardware platform needs target platform except common computer, the target should be special board or target machine, or constructed by PC. Software platform includes OS, development and application software. In development of Windows Embedded CE 6.0 assembled all the software to Visual Studio.NET 2005.

A. Target Platform Introduction

The target device mainly refers to customized hardware environment Windows Embedded CE 6.0 needed. There are three platforms are mentioned:

- 1) *simulator*, which supplied an embedded analog environment, in which, the OS and applications developed could be ran and debugged in first step, then ,more time and cost could be saved.
- 2) *EElroidpxa270 development board*, the target platform was based on Intel PXA270 processor with Intel XScale architecture, supplied by Emdoor ltd, Shenzhen, China.
- 3) *Samsung I908E mobile, an intelligent mobile based on Windows Mobile 6.1.*

Application crossing-platform portability could be enhanced by means of comparing the matters when application arranged to different target device.

B. Build Software Development Environment

When you install Windows Embedded CE 6.0 and Visual Studio.NET 2005 and other related development software, the following installation sequence needs to be met:

- Install Visual Studio .NET 2005.
- Install Visual Studio.NET 2005 Service Pack Package 1.
- Install Platform Builder for CE 6.0, when asked to select support for CPU, for the existing target device, only select "ARMV4I" architecture CPU.
- Install Windows Embedded CE 6.0 Platform Builder Service Pack1.msi.
- Install synchronized software ActiveSync 4.5.

- Uninstall .NET Compact Framework 2.0 components which Visual Studio.NET 2005 has been installed, re-install .NET Compact Framework 2.0 SP2 (To improve .NET Compact Framework 2.0 compatibility).

- Install Windows Mobile 6 Professional SDK (Mainly provides to Windows Mobile 6 for Pocket PC and Windows Mobile 6 for Pocket PC Phone the Edition development support).

- Install Windows Mobile 6 Professional Images (CHS).msi.(Chinese version of Windows Mobile 6 emulator).

- Copy BSP package to the Platform Builder for CE 6.0 installation folder in the platform folder, the bsp package supports development board EeloidPxa270 customized Windows Embedded CE 6.0 operating system.

It needs to be emphasized that the installation of Windows Embedded CE 6.0 development related software, if the development time to take advantage of .NET Compact Framework, using the SP2 version of the best,.NET Framework update to the SP1 version, so that application testing can be reduced when problems.

III. KEY TECHNOLOGY ANALYSIS OF THE INTELLIGENT DEVICE DEVELOPMENT

A The Characteristics of Intelligent Device Program

Intelligent device program, compared with the traditional PC program, has the characteristics shown in the following aspects.

1) API function and MFC

Subset is a very good concept to describe intelligent devices and conventional PC programming.

- The API function of intelligent devices and MFC controls are both corresponds to the traditional PC, Win32 API function and a subset of MFC controls. Partial feature of function has been streamlined. such as the API functions of Windows CE, does not support the security attributes, so a place with security attributes must be set to the NULL.

- Intelligent devices for the use of exception handling have been somewhat restricted.

- Some intelligent devices, API functions of Windows CE have a number of special extended attributes for the intelligent devices.

2)Transplantation of intelligent device program

API function in intelligent device is a subset of the Win32 API function ,MFC controls in intelligent device are also in a subset of the traditional MFC.Therefore, if you want the program under the traditional PC transplantation to intelligent devices, you need to consider the following issues:

- When the user wants to transplant the existing Win32 applications in Windows platform to intelligent devices, they often can not find a suitable API Function and its counterpart, so need to re-write code to achieve the corresponding function of the original function.

- When the user wants to write programs using MFC controls and transplant to intelligent devices, it needs to

check if the controls used in the intelligent devices is supported; if MFC in intelligent device that does not contain the appropriate controls, Designers want to approach implementation in the code to "bypass" the original control features or functions to achieve through their own preparation of the corresponding function.

- When the users transplant the original program on a PC device to the intelligent device, not only need to successfully run on the simulator, also need to program to the actual physical device, the only way it can guarantee good performance after transplantation to intelligent devices.

B. Characteristics of the Compile environment Visual Studio.net 2005

Visual Studio.NET 2005 in support of intelligent devices including Windows CE, Pocket PC, Smart Phone and Windows Mobile 6 and so on.

From a programming point of view, Windows CE programming interface is supported by a subset of Win32 API, which supports more than 600 kinds of the most common Win32API. It has specially designed for real-time embedded applications, and preemptive multitasking operating system kernel, the latest version is 6.0.Pocket PC is a handheld device that can store and retrieve e-mail, contacts and appointments, play multimedia files, play video games, and exchange text messages with MSN Messenger, browse Web content. Pocket PC Phone Edition, including Pocket PC Phone and Smart Phone. Pocket PC Phone which is based on the Pocket PC architecture, smart phone system that not only retains the full functionality of Pocket PC, you can also wirelessly access the Internet via GPRS, browsing news, e-mail. The target device mentioned in article is the Samsung i908 smart phone as a Pocket PC Phone. The Smart Phone with a physical keyboard, integrated PDA and mobile internet access. Windows Mobile 6 operating system is a custom product of Windows Embedded CE, which launched by Microsoft and used in the field of smart phones and PDA.

C. .NET Compact Framework

Windows Embedded CE 6.0 provides.NET Compact Framework support, this project uses .NET Compact Framework 2.0 version. Although in the occupancy of the system resources, .NET Compact Framework occupied more space compared with Windows API , and it's efficiency in the running is smaller than MFC, but given the reality of intelligent devices support the expansion of the basic large-capacity SD card or CF card, use the .NET Compact Framework will greatly facilitate the development of limits to staff. And using it brings platform-independent nature is also attractive, develop language-independent, intermediate code, distributed support, etc. are all the reasons for choice .NET Compact Framework.

.NET Compact Framework bring managed code and Web services to intelligent devices, which allows secure, downloadable applications, such as personal digital assistants (PDA), mobile phones and set-top boxes and other devices. In the .NET Compact Framework support,

can be easily and efficiently develop a suitable mobile device application, without the need to consider the specific hardware environment. .NET Compact Framework shields the details of the underlying hardware to developers, so that developers can concentrate on business logic solutions.

.NET Compact Framework is more suited to the field of mobile and embedded development, for Windows Embedded CE features, its library and other areas have done a lot of feature additions and removal. For example, remove the functionality for server, ASP.NET, remoting, reflection issue, C++ Development, J#, and JSL development, code access security, and XML architecture support. However, an increase of the following class name space to adapt to mobile and embedded applications:

- Mobile DirectX programming class.
 - Specifically for Windows CE window class.
 - supports the use of infrared transmission of a series of classes.
 - Used to support the SQL Server Mobile classes.
- Language provided that comply .NET Compact Framework 2.0, then it developed program can be run on Windows CE.

D. Native Code and Managed Code to Work Together

In order to develop Windows Embedded CE 6.0 applications can now use Visual Basic, C#, and C/C++ 3 kinds of programming languages. Generally, code involves the core algorithm and core technology use C/C++ language and the program has been prepared as the core dynamic library (.DLL), yet VB or C# language is used in the external code such as interfaces. Second, Windows CE operating system itself is to use C/C++ language, and it gives users the C/C++ form of Win32 API, which developers use VB or C# to write Windows CE application. It is sometimes necessary to communicate with the operating system, but also inevitable to call the Win32 API, so, native code and managed code is very important to work together.

1) P/Invoke (Platform Invoke)

P/Invoke is to enable managed code to call unmanaged DLL functions implemented in the service, which allows managed code to directly call native code based on the Win32 API, there are also some third-party components / resources, some of which form a dynamic link library provided, or has been already a COM component. P/Invoke could locate and call the exported DLL functions, across the interaction margin of managed code and native code, and arrange their parameters.

To sum up, the occasion using P/Invoke include:

- .NET Compact Framework does not implement some function, it needs Windows CE API.
- Existing resources such as DLL or COM components, hoping to make full use of, reduce development costs and risks.
- In view of the implementation and decompile capacity of DLL may be higher than the .NET Compact Framework, using DLL to improve program performance and security. Of course, the implementation of the DLL is

whether higher than the managed code or not can not be generalized.

The Mp3 player program developed in this project called the local code Fmodce.dll.

2) *Decompile Technology Based on Fuzzy Processing*

The purpose of fuzzy processing is to hide the intention of program without changing its runtime behavior. It is not encryption, but for .NET managed code, the fuzzy processing is more easily achieved decompile. The trick of Fuzzy processing is to make Snoopers confused and at the same time, the program can still be submitted to the same product to the CLR. Fuzzy processing is applied to compile .NET assemblies, rather than the source code process. Fuzzy processing program will never read or change the source code. The project made use of fuzzy processing' identifier renaming and overload induction method to enhance the ability to decompile the program.

IV. CROSS-PLATFORM TRANSPLANTATION OF PLAYER

A. MP3 Player Design Process

This project used Fmod components and combined with P/Invoke platform invoke design program to achieve the basic functions of MP3 player. Concrete steps are:

- Make use of PB to custom a multimedia system (intelligent device interface based on Windows CE). Simulate multimedia system which has been customized on EEIoidpxa270 development board. And then export the SDK, and install the SDK and emulator to the PC. The native code and managed code in development Language Options supported by SDK are chosen. Simulator in the configuration of the property, select the screen size of the item, the maximum is 99, while the actual need is a 640 × 480, directly into the input box 640 and 480, finally appears as 40 and 80, but the PB has been the default is 640 × 480 screen size.

- Start Visual Studio 2005, Create a new project, select the "Visual Basic", and select "Windows CE 6.0" in the "Intelligent Device", then select "Device Application" in the template; After the set up, the importation of works will be entitled "FmodMP3", then enter the Visual Basic development environment.

- Copy the resources of picture and components named Fmod.dll which program requires to the project folder named FmodMP3. To re-enter just the development environment, and add items to existing projects, new images and components would be shown in the project directory if the above steps successfully implemented.

- Program interface designing. The available Controls in .NET Compact Framework is subset of controls based on the establishment of PC-desktop application, such as: buttons, and form the properties of items where there is no function to add a background image. The use of images to complete the beautification of the control interface and the associated click event is triggered image function realization operations (simulate button function). The project need to be added a Textbox control, 7 PictureBox controls, a Progress Bar control, Timer control

and an OpenFileDialog dialog box, then set their corresponding attributes.

- Interface layout has been completed as shown in Figure 1.

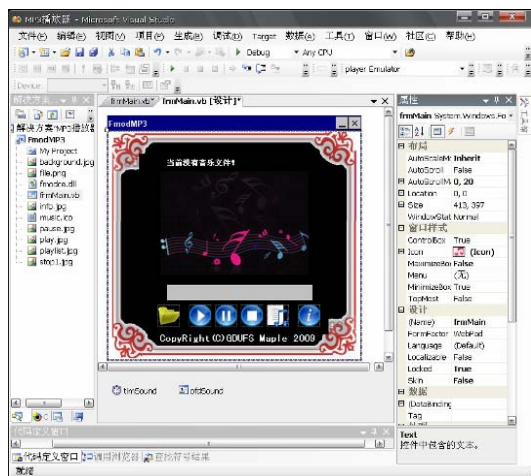


Figure 1. Player interface design

- After the completion of interface design, the other six icon works as button should be set respectively except the background image. Then write code to complete the process-related functions. Due to space constraints, the specific program code is not supplied in detail here.

B. Analysis of Fmodce.dll Correlation Function

FMOD is a very simple general-purpose audio engine. It runs very well in Windows, WinCE, Linux, PSP, Game Cube Xbox and other platforms. Before using fmod play music, we must first initialize the code is as follows:

```
FSOUND_Init(44100,32,0);
```

The first parameter is the output rate of music, its units are Hz, where is set to 44100. The second parameter is to set the maximum number of channels. The third parameter, you can specify a number of identities.

FMOD has two API, they are FSOUND and FMUSIC. All MUSIC, such as mod, s3m, xm, it, mid, rmi, sgt, fsb through FMUSIC API to play.

FSOUND API is for the compression format using. Like wav, mp3, ogg, raw and other file formats, you can convert each other with other software. Sound of music, like artillery shells fired short, then you can convert these voices "Sample" file. Samples will be decompressing into memory before playing, and can play many times. If you want to play the music as long as background music, the music can be converted to flow. Because the files are read from the disk and then converted into a flow, so it will use some CPU and memory. Also need to note that, at the same time FMUSIC can not be repeated play.

- Fmusic

Play with FMUSIC need a handle, the code is as follows:

```
handle = FMUSIC_LoadSong("YourFileName");
FMUSIC_PlaySong(handle);
FMUSIC_SetMasterVolume(handle,256)
FMUSIC_SetPaused(handle,true);
FMUSIC_SetPaused(handle,false);
```

```
FMUSIC_SetLooping(handle,true);
FMUSIC_StopSong(handle);
FMUSIC_FreeSong(handle);
• Fsound
handle=
SOUND_Sample_Load(0,"yourFileName",0,0,0);
FSOUND_PlaySound(0,handle)
FSOUND_SetVolume(handle,255);
FSOUND_SetPaused(handle,true);
FSOUND_SetPaused(handle,false);
FSOUND_StopSound (handle);
FSOUND_Sample_Free(handle);
• Streams
handle=FSOUND_Stream_Open("YourFileName",0, 0, 0);
FSOUND_Stream_Play (0,handle);
FSOUND_Stream_Stop(handle);
FSOUND_Stream_Close(handle);
The project is mainly used some of the functions of Fsound and Fstream .
```

C. The key Method Calling VB

In order to call the Visual Basic program in the Windows CE API and fmodce.dll function, the steps and code is as follows:

```
• Import Namespace,
Imports System.Runtime.InteropServices
• Statement Windows CE API
<DllImport("coredll.dll", SetLastError:=True)> _
Public Shared Function LocalAlloc(ByVal uFlags
As UInt32, ByVal uBytes As UInt32) As IntPtr
End Function
<DllImport("coredll.dll", SetLastError:=True)> _
Public Shared Function LocalFree(ByVal hMem As
IntPtr) As IntPtr
End Function
<DllImport("coredll.dll", SetLastError:=True)> _
Public Shared Function LocalReAlloc(ByVal hMem
As IntPtr, ByVal uBytes As UInt32, ByVal fuFlags As
UInt32) As IntPtr
End Function
Public Const LMEM_FIXED As Integer = 0
Public Const LMEM_MOVEABLE As Integer = 2
Public Const LMEM_ZEROINIT As Integer =
&H40
Private Function fmod_getStream_New(ByVal
filename As String) As IntPtr
Dim filenames As Byte() =
Encoding.Default.GetBytes(filename & vbNullChar)
Dim p As IntPtr=
LocalAlloc(Convert.ToUInt32(LMEM_FIXED Or
LMEM_ZEROINIT),
Convert.ToUInt32(filenames.Length))
If Not p.Equals(IntPtr.Zero) Then
Marshal.Copy(filenames, 0, p, filenames.Length)
End If
fmod_getStream_New = p
End Function
```

```

Friend WithEvents Panel1 As
System.Windows.Forms.Panel
#End Region
•Statement Fmodce.dll function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Init", SetLastError:=True,
CharSet:=CharSet.Unicode,
CallingConvention:=CallingConvention.Winapi)> _
Public Shared Function fmod_Init(ByVal mixrate As
Integer, ByVal maxsoftwarechannels As Integer, ByVal
flags As Integer) As Boolean
End Function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Stream_GetLength",
SetLastError:=True, CharSet:=CharSet.Unicode, CallingC
onvention:=CallingConvention.Winapi)> _
Public Shared Function fmod_GetLength(ByVal fstream
As IntPtr) As Integer
End Function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Stream_GetPosition",
SetLastError:=True, CharSet:=CharSet.Unicode,
CallingConvention:=CallingConvention.Winapi)> _
Public Shared Function fmod_GetPosition(ByVal fstream
As IntPtr) As UInt32
End Function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Stream_Open",
SetLastError:=True, CharSet:=CharSet.Unicode,
CallingConvention:=CallingConvention.Winapi)> _
Public Shared Function fmod_Open(ByVal data As IntPtr,
ByVal mode As Integer, ByVal offset As Integer, ByVal
length As Integer) As IntPtr
End Function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Stream_Play",
SetLastError:=True, CharSet:=CharSet.Unicode,
CallingConvention:=CallingConvention.Winapi)> _
Public Shared Function fmod_Play(ByVal channel As
Integer, ByVal fstream As IntPtr) As Integer
End Function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Stream_SetPosition",
SetLastError:=True, CharSet:=CharSet.Unicode,
CallingConvention:=CallingConvention.Winapi)> _
Public Shared Function fmod_SetPosition(ByVal
fstream As IntPtr, ByVal position As UInt32) As Boolean
End Function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Stream_Stop",
SetLastError:=True, CharSet:=CharSet.Unicode,
CallingConvention:=CallingConvention.Winapi)> _
Public Shared Function fmod_Stop(ByVal fstream
As IntPtr) As Boolean
End Function
<DllImport("fmodce.dll",
EntryPoint:="FSOUND_Close", SetLastError:=True,
CharSet:=CharSet.Unicode,
CallingConvention:=CallingConvention.Winapi)> _

```

```

Public Shared Sub fmod_Close()
End Sub
Private Function fmod_getStream(ByVal filename
As String) As IntPtr
Dim filenames As Byte() =
Encoding.Default.GetBytes(filename & vbNullChar)
Dim hfile As GCHandle =
GCHandle.Alloc(filenames, GCHandleType.Pinned)
If Environment.Version.Major = 1 Then
fmod_getStream = New
IntPtr(hfile.AddrOfPinnedObject().ToInt32 + 4)
Else
fmod_getStream = hfile.AddrOfPinnedObject()
End If
If hfile.IsAllocated Then
hfile.Free()
End If
End Function
#End Region

```

Do a good job after the above two steps can use Fmodce.dll function, combined with the control layout to achieve FmodMp3 player functions.

D. Change the target device platform

The player program is based on the ARM architecture, the program need no change when it is migrated to the real target device EEloidpxa27x embedded development board, the program can be downloaded to EEloidpxa270 development board if player device is selected. The interface when player in the development board run is shown in Figure 2. Next, transplant the project's program to Samsung smartphone I908E, before this, the program running environment should be simulated by Microsoft's Windows Mobile 6 emulator because of the reason that



Figure 2. the Interface layout in Eloidpxa27x

Samsung's smartphone I908E is using a Windows Mobile 6.1 operating system. The following changes to the procedures applicable to smartphone available on the simulator program.

The most important first step is that, the project Properties has item "Change the target platform" only in the program written by managed Language (precisely, the intelligent devices platform with the same .NET Compact Framework version).As shown in Figure 3.



Figure 1. Change the target platform.

Interface layout of the entire program in Visual Studio 2005 development environment is shown in Figure 4. Adjust the interface to adapt to the simulator interface, the running screen of modified program in the simulator to Windows Mobile 6 Professional is shown as Figure 5.

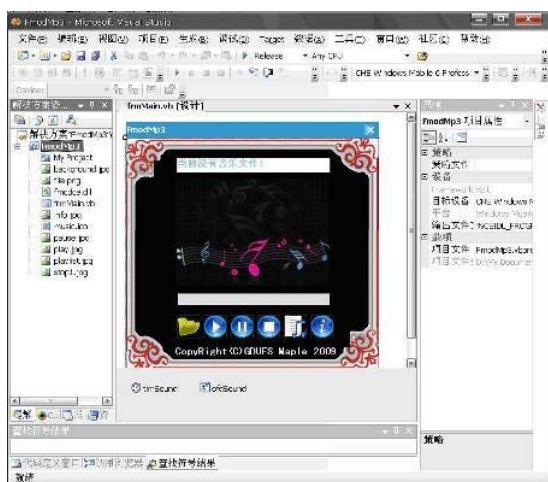


Figure 4. the Interface layout in VS2005

Transplant the modMP3 program on the emulator to Samsung smart phones I908E, a true smart phone's screen is not fully compatible with Microsoft simulator; download the program after slight modifications made. Running interface on the phone is shown as Figure 6.

E. The Reflected Decompile Ability of Core Algorithm

The last generated .exe or .DLL file written in managed language is a kind of intermediate language (MSIL) procedure, the source code could be restored using .NET Framework, MISL disassemble tool, or the Reflector decompile tools, such as the Figure 7 shows the screenshots that FmodMp3 program was decompiled by Reflector tool, it can be seen from the chart that fmodce.dll (C / C + + write) can not be decompiled, while the VB code in almost all the languages displayed . Through the core algorithm written uses C / C + + written code while interface with peripherals such as VB and

other managed languages to safeguard the intellectual property of software development and better realize the commercial value of the product.



Figure.5. Windows Mobile 6 simulator interface



Figure 6. FmodMp3 running interface on the Samsung smartphone



Figure 7. Decompile FmodMp3 with Reflector

V. CONCLUSION

Use .NET Compact framework 2.0 development process, to some extent to achieve the program cross-platform migration. Fmodmp3 player mentioned in the article was developed based ARM architecture, which can run on PC machine (simulator), embedded development board and intelligent mobile phones, to

achieve cross-platform migration. But some features can not be fully realized, this is caused by the reason that .NET framework is not supported in the .NET Compact framework exception mechanism and security mechanism. The design used only for the unique function of intelligent devices is also a relationship. But With the.NET Compact framework3.5, VS2008 and other technological advances, such as the program in the intelligent device to achieve true cross-platform will not be far away.

ACKNOWLEDGMENT

The authors wish to thank the Scientific and technological research project Foundation of Guangdong Province contract 2008B080701007 and Natural Science Foundation of Guangdong Province for contract 9151042001000017 under which the present work was possible.

REFERENCES

- [1] XUE DaLong, CHEN ShiDi, WANG Yun. Windows CE embedded systems development from basic to practice. Beijing: Electronic Industry Press, 2008
- [2] JIANG Bo.Windows Embedded CE6.0 Program designed to combat [M]. Beijing: Mechanical Industry Press, 2009
- [3] (U.S.A) Powers-L, Snell • M (wrote) ;LIU YanBo, XIAO Na, JIA Han (translated) Visual Studio2005 technology collection (M). Beijing: Posts & Telecom Press, 2008.1
- [4] (France) Smacchia. P. (wrote); SHI Fan (translated). C# and .NET2.0 Practice: Platform, language and framework [M]. Beijing: Posts & Telecom Press, 2008.1

- [5] CE6 OAL you have to know something[OL]. <http://blog.sina.com.cn/s/blog-537bca0100corr.html>.
- [6] Microsoft. Advanced P/Invoke on .NET Compact Framework [J/OL].MSDN Library.
- [7] Microsoft. NET framework decompiles technology [J/OL].MSDN Library.

Jing .Zhang was born in Dalian city,Liaoning province, November 24, 1977.Zhang graduated from Shenyang University of Technology in July of 2000 and received a bachelor of engineering degree. Then entered Guangdong University of Technology and Received a Master of Engineering degree in July of 2003.The major field of study is embedded system and Artificial Intelligence.

Her current job is TEACHING and works in .Guangdong University of Foreign Studies. Previous publications are: Circuit analysis and electronic technology(Beijing: Machinery Industry Press,2007),WINCE operating system transplantation based Intel PXA270(Beijing: Micro Computer Information,2008),Development and design of SQL Server Mobile database(Shanghai:Computer Application and Software,2008) .Now she is interested in embedded system and Artificial Intelligence.

Xinguang .Li was born in Nanxian city, Hunan province, July 6, 1965. Li graduated from South China University of Technology in July of 1998 and received a Dr. Degree. The major field of study is embedded system and Artificial Intelligence.

His current job is TEACHING and works in .Guangdong University of Foreign Studies. Now he is interested in embedded system and Artificial Intelligence.