

Re: Accessing peripherals from a managed application

Source:

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.app.development/2006-06/msg00068>.

- *From:* "Bruce Eitman \ (eMVP)" <beitman.nospam@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 16 Jun 2006 08:25:08 -0400
-

You don't, because you don't call those functions.

You call the Win32 APIs for serial ports. Look in help for the Serial Communications Functions. You could probably find some code at OpenNetCF.org like OpenNetCF.IO.Serial

Bruce Eitman (eMVP)
Senior Engineer
beitman AT applieddata DOT net

Applied Data Systems
www.applieddata.net
An ISO 9001:2000 Registered Company
Microsoft WEP Gold-level Member

"Mario" <alphantommy@xxxxxxxxxxxxx> wrote in message
news:1150418975.424374.64740@xx

Paul,

Thanks for your input. I used dumpbin utility to find out the headers of the dll files. I did not find anything in the coredll.dll that resembled functions for uart. However, I did find functions possible related to the serial port on com16550.dll (output at the end of this post) that was in my project release directory. Now, I can do a P/Invoke on this functions, but how do I find what parameters each one of these functions has? I tried the P/Invoke Wizard from Paul Yao's site, but it kept crashing my system.

Regards,

Mario

Re: Accessing peripherals from a managed application

Dump of file com16550.dll

File Type: DLL

Section contains the following exports for SERIAL.dll

00000000 characteristics
44903EE6 time date stamp Wed Jun 14 09:52:54 2006
0.00 version
1 ordinal base
12 number of functions
12 number of names

ordinal hint RVA name

1 0 00001A44 COM_Close = COM_Close
2 1 00001C0C COM_Deinit = COM_Deinit
3 2 000022E0 COM_IOCTLControl = COM_IOCTLControl
4 3 000030E8 COM_Init = COM_Init
5 4 00003380 COM_Open = COM_Open
6 5 000020F0 COM_PowerDown = COM_PowerDown
7 6 000020B0 COM_PowerUp = COM_PowerUp
8 7 0000198C COM_PreClose = COM_PreClose
9 8 00001BA4 COM_PreDeinit = COM_PreDeinit
10 9 00001D30 COM_Read = COM_Read
11 A 000020A8 COM_Seek = COM_Seek
12 B 00003618 COM_Write = COM_Write

Summary

1000 .data
1000 .pdata
1000 .reloc
7000 .text

Paul G. Tobey [eMVP] wrote:

How would a C/C++ program talk to the driver? You'll do exactly the same thing from managed code. Maybe you open the driver with CreateFile(), and read and write to it with ReadFile() and WriteFile(). If that's the case, you'd P/Invoke to those API calls from managed code, yes. You don't somehow import everything in coredll.dll. You declare the functions that you want to call, indicating that they are in coredll.dll (*if* they are in coredll.dll and not some other DLL), and then call them.

Re: Accessing peripherals from a managed application

Paul T.

"Mario" <alphantommy@xxxxxxxxxxxx> wrote in message
news:1150392387.307256.105420@xx

Bruce,

Thanks for the post. I do have the drivers for the hardware since they came with the devevelopment board. What I have not figure out is how to access to the driver from a managed application. From what I have gathered, this is the sequence for developing a managed application with VS2005 and PB (WinCE 5.0)

1. Create an OS image wtih PB.
2. Somehow figure out the headers contained in coredll.dll
3. use dllimport to import coredll.dll in the managed application in VS2005.
4. Call on the imported functions to access the hardware.

Am I far off? The first experiment that I want to do is just to send some data through the SPI or UART from a managed app. Again, thanks for any additional input.

Best Regards,

Mario

Bruce Eitman (eMVP) wrote:

Mario:

Since nobody else replied, I will. I am not a Managed App developer, but I can tell you how this is typically done.

Typically, the hardware is accessed via a driver written in C/C++, then a managed app accesses the driver to access

Re: Accessing peripherals from a managed application