Name:

SID: _____

CSCE 351: Operating System Kernels

Lab 1 – Introduction to Windows CE .NET Platform Builder

Basic Setup:

• Windows 2000/XP workstation with Windows CE .Net 4.2 installed.

Objectives:

The objectives of this lab are as follows:

- Familiarize students with the development process in platform builder.
- Expose students to the basic debugging process in platform builder.
- Illustrate how to download a design to a device that can be either an actual system (e.g. E-box) or the provided emulator.
- Exploring some of the remote monitoring and debugging capability in platform builder 4.2.

Estimated Lab Time: 60 minutes

Introduction

Microsoft Platform Builder is an integrated development environment (IDE) for building customized embedded platforms based on the Microsoft Windows CE .NET operating system (OS). Platform Builder comes with all the development tools necessary for you to design, create, build, test, and debug a Windows CE–based platform. The IDE provides a single integrated workspace in which you can work on both platforms and projects.

As a student in CSE, you are able to download a copy of Windows CE .Net 4.2 with academic license. The file is available from <u>http://msdn08.e-academy.com/unl_cseng</u>. In order to download, you need to send an email to <u>manager@cse.unl.edu</u> to set up an account. Please specify that you need to download Windows CE .Net 4.2 and you are a student in CSCE 351. You should try to install Windows CE .Net 4.2 on your personal machine as well.

In this lab, we will learn how to create a platform, add/delete component to the platform, compile and debug the platform, download image to the device (the built-in emulator).

1. Platform Creation with the New Platform Wizard

- Step 1. Open platform builder
- Step 2. Once launched, you need to create a new platform workspace. To do so, you select:
 - File | New Platform and click <u>Next</u>

New Platform Wizard - Step 1		×
	Welcome to the New Platform Wizard	
	This wizard guides you through the process of creating a new Windows CE .NET-based platform. A platform is a specific implementation of the Windows CE operating system (OS) on a target device. A platform consists of an OS image and a board support package (BSP), which includes an OEM adaptation layer (OAL) and device drivers. This wizard helps you: - Select a BSP - Select a platform configuration - Select a configuration variant - Select additional features based on your configuration choices To continue, click Next.	
Q	< <u>B</u> ack <u>N</u> ext > Einish Cancel	

• Check EMULATOR: X86 and click Next. **Notice:** Windows CE .NET provides an Emulator, which is a tool that mimics the behavior of hardware that supports a Microsoft Windows CE–based platform. With the Emulator, you can design and build a Windows CE–based platform and test it using software that mimics hardware rather than testing the platform on a real system.

iew Platform Wizard - Step 2	X
Board Support Packages (BSPs) A BSP Catalog feature (.cec) file contains the or that are added to your platform when your platf	default set of device drivers 🛛 👌
Available BSPs:	Select one or more BSPs for your new platform.
NATIONAL GEODE: X86	A BSP for the Emulator platform. The platform uses the OS based on the x86 architecture.
	Note: Only BSPs supported by installed CPUs are displayed in the list.
() < Back	Next > Finish Cancel

Question: the list represents various embedded processors that the tool can support. As part of the post lab, please provide a complete specification for CEPC: X86 that include the most current processor clock rate and a sample device that uses that particular processor.

• Select Internet Appliance from a list of Available configurations, input platform name as MyLab1, and click Next.

Question: Notice, there are several types of devices that are supported by Windows CE .NET. As part of your post lab, answer the following questions: i) What is a set top box? ii) List one manufacturer of set-top-box that uses CE .NET as the operating system.

• Available configurations	Select the platform configuration that most
Digital Media Receiver Enterprise Terminal Enterprise Web Pad Gateway Industrial Controller Internet Appliance IP Phone Mobile Handheld Mobile Phone Set-Top Box	closely matches your device. <u>P</u> latform name:
Windows Thin Client	MyLab1
	Location:

 Make sure that Internet Brower is checked in the list available features and other features are not checked, and then click <u>Next</u>.

New Platform Wizard - Step 4 Application & Media Select the application and media options you w platform.	x] vant to include in your
Available features:	A feature that enables local playback support for Windows Media Audio and MP3 files in a small footprint. Includes DirectShow, Windows Media Technologies, and codec. Windows Media Player is not included.
	Estimated size of these features: 2619 KB
2 < Back	Next > Einish Cancel

click <u>N</u>ext

New Platform Wizard - Step 5			×
Networking & Communications Select the networking and c your platform.	ommunication options yo	u want to include in	1
<u>A</u> vailable features: [⊕-⊠ Local Area Network (LAN)	Acc	ommunications network	that connects
Personal Area Network (P Personal Area Network (P Personal Area Network (P Personal Area Network (P Personal Area Network (P	AN) geo on	graphically separated ar	reas.
		6	
	Esti	mated size of these feat	ures: 2764 KB
Q	< <u>B</u> ack <u>N</u> ext	> <u>F</u> inish	Cancel

• click <u>F</u>inish

New Platform Wizard - Step 6	
Completing the New Platform Wizard	
You have successfully completed the New Platfo Wizard.	9rm
You have created and begun customization of a basic platform configuration.	
Build options:	
 Build the debug version of your platform after this wizard closes. 	
 Build the release version of your platform after this wizar closes. 	<u>d</u>
 Modify the build options for your platform after this wizan closes. 	<u>d.</u>
	De
To close this wizard, click Finisn	
③ < <u>B</u> ack	Next 2 Einish Cancel



2. Customize and Build the Platform

In this section, we will perform simple modification to the kernel. This step will take about 30 minutes.

Step 1. Switch from Release Build to Debug Build

1 🖬 X 🖷 🖬	£ * £ * 🖪 🎮 📰 😚 🦮 🔄	
EMULATOR: X	86 Win32 (WCE emulator) Debug	
EMULATOR: X	86 Win32 (WCE emulator) Release	
EMULATOR: X	86 Win32 (WCE emulator) Debug	
AB1 features		

Step 2. Select Platform | Setting | Build Options and *Enable Kernel Debugger*



Step 3. Select Build | Build Platform

This step takes a bout 15-20 minutes to finish. While you are waiting for the build process to complete, please read Appendix A, *History of Windows CE*.

The build system builds a platform in four sequentially executed phases: the Sysgen phase, the Feature Build phase, the Release Copy phase, and the Make Image phase. Please go to this link for more information about build phases:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcepb40/html/ wcepb Building an OS Image.asp

If the build succeeds, the build information should contain 0 errors.

3. Download OS image to the emulator

Step 1. In platform builder, Select Target | Configure Remote Connection Step 2. Select *Emulator – 4.20* for both *Download* and *Kernel Transport*

Configure Remote Connection	×
Services Settings	
Active named connection	
MyLab1 Add New Delete	
Services for active named connection	
Download:	
Emulator - 4.20 Configure	
Kernel Transport:	
Emulator - 4.20 Configure	
Use Hardware Debugger Select EXDI Driver:	
OK Cancel Apply Help	

Step 3. Increase the memory size to 64MB

	the second s		
Screen Size (WxHxD):	800 🛨 600 🛨 16	z	
C Skin File:		1944	
System			
Memory (MB):	64 Î	<u>.</u>	
Host Key:	Right Alt	•	
In Background:	Betain foreground priority		
Communication			
Communication	NAT (Outgoing only)	•	
Communication Ethernet Serial Port <u>1</u> :	NAT (Outgoing only)	•	
Communication Ethernet Serial Port <u>1</u> :	NAT (Outgoing only) None	× •	
Communication Ethernet Serial Port <u>1</u> : Serial Port <u>2</u> :	NAT (Outgoing only) None None	•	
Communication Ethernet Serial Port <u>1</u> : Serial Port <u>2</u> :	NAT (Outgoing only) None None		

Step 4. Make sure that the NAT is selected for *Download* configure in order to connect to internet

Step 5. Click the setting Tab, and make sure Download Image, Initialize (Jump to Image), and Connect to Device are all checked, and then click OK.

VICES Deaning		
Download / Initialize De	vice	
Download Image	an canada an	
Initialize (Jump to Im	iage)	
I Connect to Device		
Reset		
Download always		🔽 Eorce clean
C Download if image	changed	
C Jump to image only		
Target Output Message	18	
✓ Timestamps	Process IDs	✓ Thread IDs
Example Output Text:	987654 PID:ABCD1234	TID:FFFF8765
Send to Output Wind	low F	Annend now line
		Whelle new me
Send to File:		

Step 6. Select Target | Download/Initialize

Step 7. After download is finished, the following window will show up in the emulator display (look in your task bar to find the emulator display program).



4. Platform Debugging

We can debug emulator-based platform. After the download is finished, the platform builder enters the debug mode. We can watch the debug message, the process and thread states. We can also set break points at source code and watch the variable values.



- To debug, you can Select what you want to see in View | Debug Windows
- To set break point,
 - Select Debug | Break
 - $\circ~$ Open the source code from File | Open
 - $\circ~$ Set the break point in the source code
 - $\circ \ \ Select \ Debug \ | \ Go$

We give an example in the following:

- Step 1. Download the image into the Emulator
- Step 2. Select View | Debug Windows | Processes
- Step 3. Select View | Debug Windows | Variables
- Step 4. Open an application in the emulator window, e.g., internet explorer
- Step 5. Select Debug | Break
- Step 6. In the platform builder, click file | open to open the file %WINCEROOT\PRIVATE\WINCEOS\COREOS\NK\KERNEL\schedule.c
- Step 7. Set up a break point in function SC_NKTerminateThread

```
11-
//-----
void
SC_NKTerminateThread(
   DWORD dwExitCode
   )
                                                             Ι
{
   int loop;
LPCRIT pCrit;
   PCALLSTACK pcstk, pcnextstk;
   LPCLEANEVENT 1pce;
   PTHREAD pdbg;
HANDLE hClose, hWake;
   LPCRITICAL_SECTION 1pcs;
   LPTHREADTIME 1ptt;
   LPPROXY pprox, nextprox;
   RTs RT:
   FILETIME ftExit;
   DEBUGMSG(ZONE_ENTRY,(L"SC_NKTerminateThread entry: %8.81x\r\n",dwExitCode));
   SETCURKEY(0xffffffff);
   if (GET_DYING(pCurThread))
       dwExitCode = GetUserInfo(hCurThread);
   SET_DEAD(pCurThread);
```

- Step 8. Select Debug | Continue.
- Step 9. Close an application in the Emulator window, (e.g. Internet Explorer).
- Step 10. Now the kernel stops at the breakpoint.
- Step 11. In the Variable window, click the cell under Name and input RunList.
- Step 12. *RunList* contains a list of threads, the current running thread is called *pth*. You can open it to see detail information.

Name	Value	
🛱 RunList	()	1
🖽 pRunnable	0x83e20800	
👎 pth	0x83d4e800	che ne ne ne ne ne
wInfo	0x000c	
– bSuspendCnt	0×00 ''	
- bWaitState	0x02 ' "	
🕀 pProxList	0×0000000	
🗄 pNextInProc	0x83db7890	
🕀 pProc	0x81fa2148	
🕀 pOwnerProc	0x81fa2148	
🛛 🕂 aky	0xffffffff	
🕀 pcstkTop	0×00000000	
- dwOrigBase	0×10560000	1

Step 13. Feel free to debug other functions and open more watch windows.