

WinCE 实验教程

WinCE 实验教程之一嵌入式系统

嵌入式这个词现在变得越来越流行,然而关于它的很多概念可能还有一大部分人不是很明确,在本文中,我们不追求具体的概念是否准确,而是要对其有一个准确的理解。

嵌入式系统是不同于常见计算机系统的一种计算机系统,它不以独立设备的物理形态出现,即它没有一个统一的外观,它的部件根据主体设备以及应用的需要嵌入在设备的内部,发挥着运算、处理、存储以及控制作用。从体系结构上看,嵌入式系统主要由嵌入式处理器、支撑硬件和嵌入式软件组成。其中嵌入式处理器通常是单片机或微控制器;支撑硬件主要包括存储介质、通信部件和显示部件等;嵌入式软件则包括支撑硬件的驱动程序、操作系统、支撑软件以及应用中间件等。

可见,嵌入式系统是一个很大的概念,一旦嵌入式处理器和支撑硬件选定了,那么工作最多的就集中在嵌入式软件当中了。而嵌入式软件中的嵌入式操作系统部分和应用软件部分就成了重中之重。它们与通常说的操作系统与应用软件的概念是相似的,但也有区别。

嵌入式操作系统是与应用环境密切相关的,从应用范围角度来看,大致可以分为通用型的嵌入式操作系统如 Windows CE、VxWorks、嵌入式 Linux 等和专用型的嵌入式操作系统如 Palm OS、Symbian 等。从实时性的角度看,大致可以分为实时嵌入式操作系统和一般嵌入式操作系统。从原理上说,嵌入式操作系统仍旧是一种操作系统,因此它同样具有操作系统在进程管理、存储管理、设备管理、处理器管理和输入输出管理几方面的基本功能,但是由于硬件平台和应用环境与一般操作系统的不同,那么它也有自身的特点,最大的特点就是可定制性,也就是它能够提供可配置或可剪裁的内核功能和其他功能,可以根据应用的需要有选择的提供或不提供某些功能以减少系统开销。

前已述及,微软公司的 Windows CE 操作系统就是一种嵌入式操作系统,它 1996 年开始发布 Windows CE 1.0 版本,2004 年 7 月发布了 Windows CE .NET 5.0 版本,目前用得最多的是 Windows CE .NET 4.2 版本,其发展速度也是很快的,功能上自不必描述,它的主要应用领域有 PDA 市场、Pcket PC、Smartphone、工业控制、医疗等。

现代的嵌入式操作系统同嵌入式操作系统的定制或配置工具紧密联系,构成了嵌入式操作系统的集成开发环境。就 WinCE 来讲,你无法买到 WinCE 这个操作系统,你买到的是 Platform Builder for CE.NET 4.2 的集成开发环境,我们也简称为 PB,利用它你可以剪裁和定制出一个符合你自己需要的 WinCE.NET 4.2 的操作系统,因此,我们说的操作系统实际上完全是由自己定制出来的,这就是嵌入式操作系统最大的特点。

对于嵌入式的应用软件,通常就是指运行在嵌入式操作系统之上的软件了,这种软件由于不再针对常规的操作系统进行开发,因此很多如 VB、VC++ 等开发工具就不方便使用了,那么就有专门的 SDK 或集成开发环境来提供这种开发需要。在 WinCE 操作系统上的应用软件开发,微软就提供了 Embedded Visual Basic(简称 EVB)、Embedded Visual C++(简称 EVC)、Visual Studio.NET 等工具,它们是专门针对 CE 操作系统的开发工具,EVB 只支持到 CE 的 3.0 版本,目前用得最多的还是 EVC,把你的 CE 操作系统中的 SDK(软件开发包)导出然后安装在 EVC 下,就可以变成专门针对你这种设备或系统的开发工具了。而 VS.NET 中的 VB.NET 和 C# 也提供了对以 CE 为操作系统的智能设备开发的支持,而且也很方便,但必须要求这些设备中提供了对微软的 .NET Compact Framework 的支持才行,如果使用的话就要看具体情况了。

以上说了这么多,还是举个例子吧。比如说我要做一台医疗仪器,那么我就要选择好嵌入式的硬件环境,然后定制出符合我需要的 CE 操作系统,利用这个系统导出 SDK,然后利用 EVC 结合这个 SDK 来开发我的信号采集、处理和病情分析的应用程序,最后就形成了一台合适的利用嵌入式技术开发出的仪器了。

如果你已经对嵌入式系统或 CE 产生了兴趣,那就请继续关注后续文章吧。

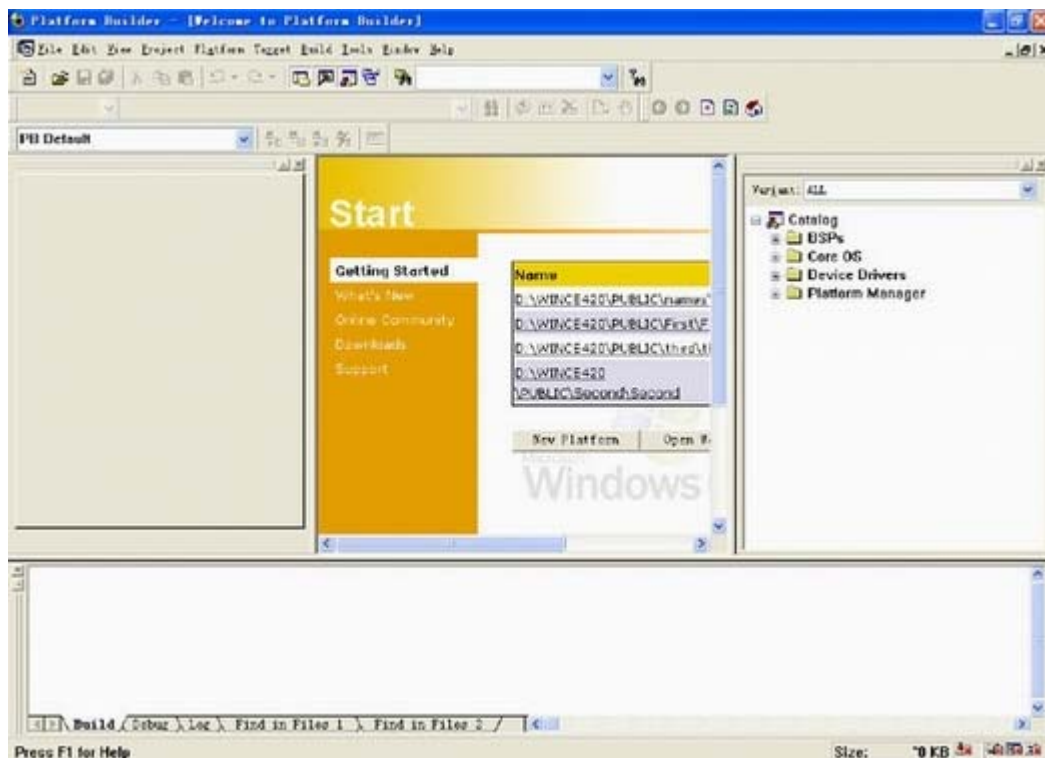
WinCE 实验教程之二集成开发环境

为了定制 CE 操作系统，微软公司为我们提供了 Platform Builder（简称 PB，但与 Power Builder 的数据库开发工具截然不同）的集成开发环境，下面我们就来认识一下这个开发环境并利用它生成一个 CE 操作系统。

先来说说 PB 的购买。PB 是商业软件，用户需要向微软在中国的各代理机构联系购买，价格大约在 10000~15000 之间，可以向各代理机构查询。如果出于非商业目的可以到微软公司的网站下载其评估版，这个版本好像只支持 CEPC 的体系结构，无法用在商业开发当中。本教程所用的 PB 是从微软购买的正版软件，版本号是 4.2，所以可能会和评估版有些差异，因本人没有使用过评估版，因此具体细节就不得而知了。

PB 的安装很简单，和通常 Windows 下的软件安装方法是一样的，只要插入安装光盘（现在购买到的 PB 都是一张 DVD 盘，以前的是 7 张 CD 盘）就会自动安装，或者手动执行光盘中的 setup.exe 安装程序即可。在安装过程中需要强调的有两点，一是要选择我们实际要用到的体系结构，CE 支持 x86、ARM、RISC 等很多种体系结构的嵌入，你要根据你的需要来选择，不要多选，很占用硬盘空间，也不要少选或错选导致以后用到的时候要重新安装，在这里我们用的是 x86 结构和其模拟器；二是尽量把操作系统文件夹即 WINCE420 文件夹放在非引导分区，即如果你的硬盘的引导区是 C 盘，那么 WINCE420 最好放在其他空间较大的盘中，防止将来引导分区损坏重装系统时把你作的 CE 系统也给格式化掉，因为 WINCE420 文件夹下的东西就是 CE 系统的组件和你自己做的 CE 系统。

PB 安装好后就可以运行它了，我们先来看一下它的窗口组成。



怎么样，是不是和 VC++ 的样子差不多？没错，左侧是 **Workspace** 窗口，它将包含你自己的 CE 系统，中间是启动窗口，没什么具体用处，右侧是 **Catalog** 窗口，是 PB 提供的可供你选择使用的 CE 内核组件包，你只要把其中你需要的组件选到左侧的 **Workspace** 窗口然后编译，那么就可以得到你自己的 CE 系统了，就是这么简单直观。底部是 **Output** 窗口，是编译、调试、日志、查找的输出窗口，和 VC++ 的一样。

接下来我们就先在模拟器上生成我们自己的 CE 系统。

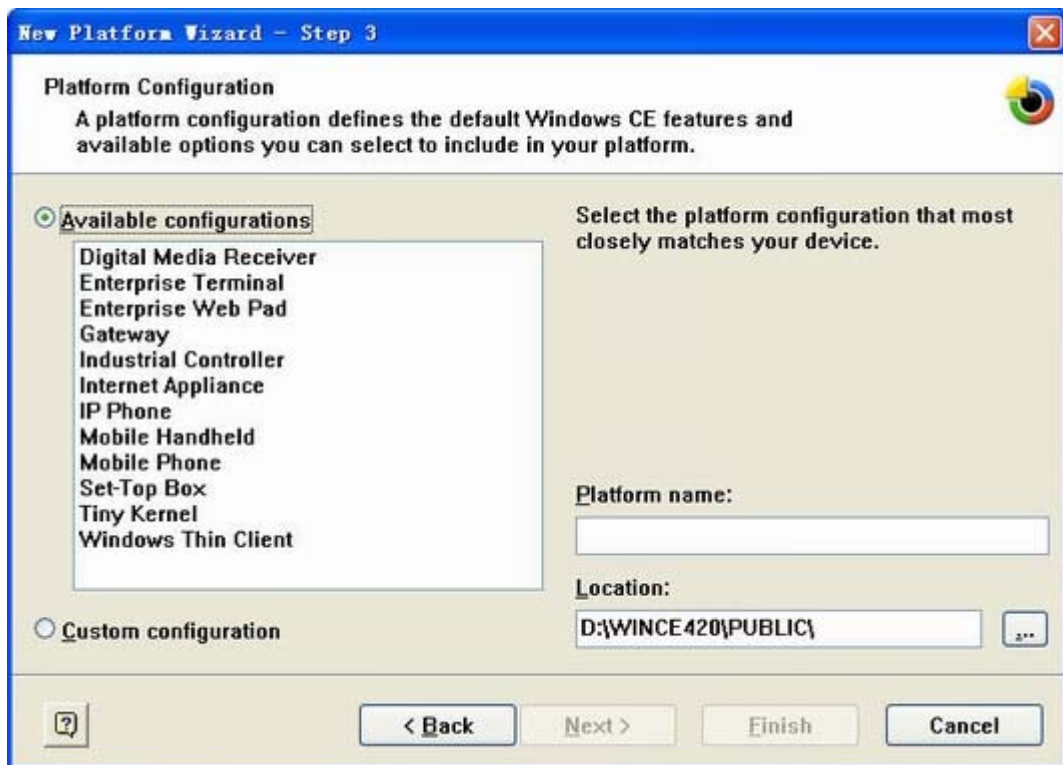
我们已经看到了在屏幕中间有个“**New Platform**”按钮，也可以在“**File**”菜单下发现这一命令，我们就从单击这个按钮开始。

有人已经猜想会出现什么了，没错，就是一个微软惯用的向导。

第一步，是一屏介绍，告诉你该向导可以帮你作什么，直接点击“**Next**”；

第二步，是让你选择 **BSP**，所谓 **BSP** 即板支持包，是由主板厂家提供给你的 CE 组件，以 x86 为例，有三种选择，**CEPC** 是在 x86 架构上以 PC 机的硬件为基础的 **BSP**；**EMULATOR** 是在 x86 架构上的模拟器，即在 PC 机上用软件模拟一个硬件平台，如果我们手中没有硬件平台那么就可以利用这个模拟器在自己的 PC 机上模拟一个平台，当然功能是有限的；**NATIONAL GEODE** 是利用 **GEODE** 微处理器的 x86 结构，如果你使用的 CPU 是这种那么就应先则此项 **BSP**。此处我们就先选择模拟器吧。

第三步，配置你的平台，如下图所示：



可以在可用的配置中选择一项如数字多媒体设备、企业终端、移动手机等，也可以选

择自定义配置，同时选择你的系统的存储路径和平台名称，这里我们就先选一个 **Internet Appliance**，放在默认路径，平台名称就叫 **Hello** 吧。

第四步，配置你想包含在你系统中的应用程序和多媒体选项，例如是否包含对 **.NET Compact Frameworks** 的支持，是否包含 **IE** 浏览器，是否包含 **MP3** 播放或 **MPEG-4** 播放等等，此外取默认值。

第五步，像第四步一样配置网络通讯组件，取默认值。

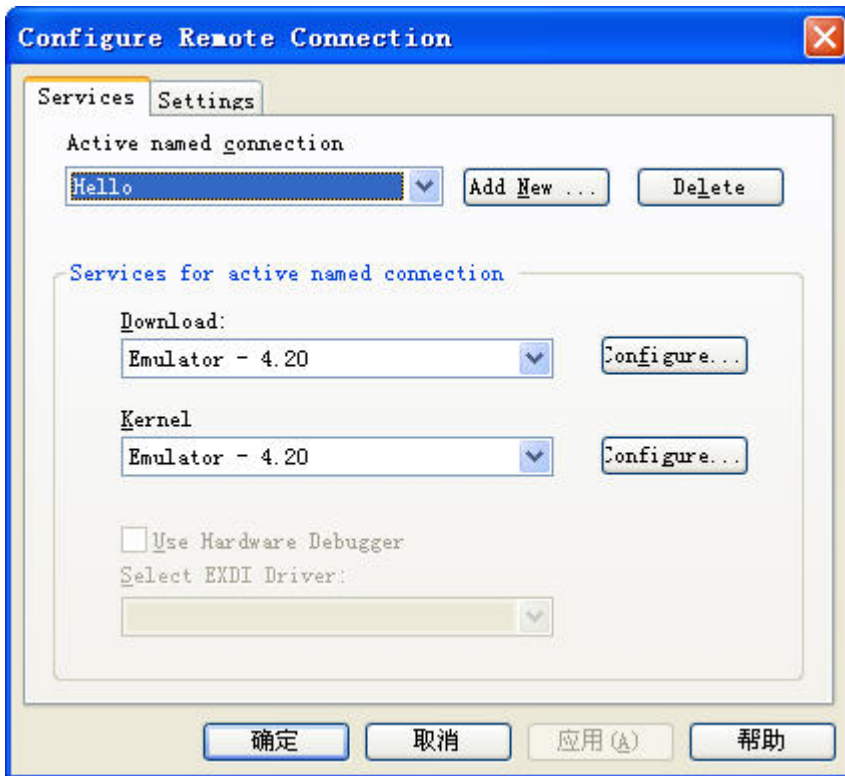
第六步，完成。

至此，我们自己的第一个 **CE** 操作系统就定制完成了，接下来要做的就是编译了，通过“**Build**”菜单下的“**Build Platform**”或“**Rebuild Platform**”命令来进行，注意在编译之前通过“**Build**”菜单下的“**Set Active Configuration**”来确定是编译 **Debug** 版还是 **Release** 版，编译的时间比较长，一般要 10 分钟以上，看你选择的组件的情况了，耐心等待一会吧，也可以先休息一会了。

好了，编译完成了，我们来看一下 **WINCE420\Public\Hello** 文件夹下的变化，在 **RelDir** 文件夹下分别有 **Debug** 和 **Release** 版本的文件夹，我们到 **Release** 文件夹下可以发现很多编译过程中和编译后的文件，找到 **NK.bin** 文件，它就是我们的 **CE** 系统打包后的文件了，也就是我们最后要的 **CE** 系统。

接下来我们来利用模拟器运行一下我们自己的系统吧。

我们首先执行“**Target**”菜单下的“**Configure Remote Connection**”命令，在弹出的对话框中配置如下：



然后运行“Target”菜单中的“Download/Initialize”命令把生成的操作系统下载到设备中去并初始化设备，在这里我们上图已经配置过了，就是下载到模拟器中去。在此期间弹出的对话框都可以选择“Yes”，最后我们就看到了如图所示的系统模拟器界面，这个就是我们自己定制的 Hello 的 CE 系统。



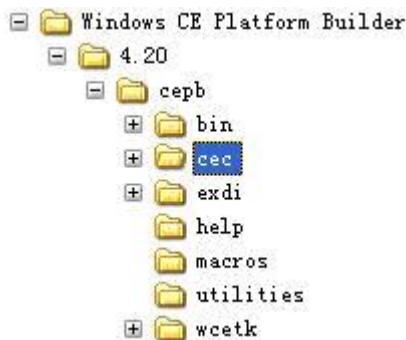
如果运行完成,那么除了关闭模拟器窗口以外,还要执行“Target”菜单下的“Disconnect”命令断开与设备的连接。

好了, PB 这个集成开发环境就介绍到这里,利用 PB 做自己的 CE 操作系统,你学会了吗?

WinCE 实验教程之三文件夹结构

这里所述的文件夹结构，不是 WinCE 系统的文件夹结构，而是集成开发环境 Platform Builder 的文件夹结构，该文件夹结构十分复杂，共有 3 万多个文件，2400 多个子文件夹，如果不清楚的话在以后的开发过程当中将会带来很大的麻烦，也时甚至会发生找不到文件的现象。对于这个文件夹结构，我们可以大体上分成两个部分，一个是 PB 的安装文件夹，一个是 CE 文件夹，下面我们就分别来了解一下。

先来看 PB 的安装文件夹。该文件夹一般装在系统盘的 Program Files\Windows CE Platform Builder\4.20 文件夹下，文件夹结构如下图所示：



在这个文件夹结构中，cec 文件夹是很重要的，它是包组件文件（.CEC 文件）的安放位置，在 PB 安装以后这里面包含了很多标准的操作系统组件、设备驱动程序组件、板支持包组件、平台管理组件等，如果用户想要扩展组件的话，只需要把相应的 CEC 文件安放在这个文件夹中即可，因此它是系统组件的配置文件所在的文件夹。

Utilities 文件夹中包含的是一个有用的工具，通过它可以生成系统的启动盘，从而可以引导我们自己定制的 CE 操作系统。这个在下一章中会有集中的应用。

Wcetk 文件夹中包含的是另一个有用的工具，通过它可以测试 CE 的性能，有由我没有用过这项，所以更多的功能我也不是很清楚。

接下来我们看一下 CE 的文件夹，其结构如下图所示：



其中 PLATFORM 文件夹下存放的是与具体平台相关的程序，当你修改某一平台的内核时就要到具体的平台所在的文件夹下去修改，比如 EMULATOR 平台即模拟器的 KERNEL 部分，那就要到 EMULATOR 的文件夹下改其 KERNEL 子文件夹下的源程序。

SDK 文件夹包含了 PB 在编译时用到的如 LINK.exe 等程序，如果我们需要手工编译些什么东西那么可以到这个文件夹下来找相应的工具程序。

PUBLIC 文件夹下是各平台要用到的公共的源程序，也是子文件夹最多的一个文件夹，它的结构如下：



其中大部分都是系统组件的源程序比如 IE、SHELL，如果我们想要修改某个组件的行为就可以到相应的文件夹下去找。

用得最多的是其中的 COMMON 文件夹，在该文件夹下的 SDK 文件夹下的 SAMPLES 子文件夹中有一些示范样例程序，比如大键盘的输入法的源程序等，我们可以更改这些源程序。在该文件夹下的 OAK 文件夹中的 CSP 文件夹为 CPU 支持组件，里面的各子文件夹都是针对特定的 CPU 的内容，比如针对 ARM、I486、SA11X1 等，如果我们需要处理和特定 CPU 相关的部分就可以到此文件夹下来操作。在该文件夹下的 DRIVERS 文件夹为微软做好的各种典型设备的驱动程序的源程序，比如 1394 的驱动、网卡的驱动、串口的驱动等，如果我们想要修改驱动或重新驱动，都可以以这个文件夹下的源程序做参考。

说了这么多文件夹结构，那么如此复杂的文件夹结构在编译 CE 操作系统的时候是如何组织起来的呢，编译器是如何找到所需要的文件的，又是由谁来告诉编译器如何编译的呢？这些问题的答案就是 DIRS 文件和 SOURCES 文件，顾名思义，一个是负责连接各级文件夹的，一个是负责编译选项的。

DIRS 文件是位于需要编译的文件夹中的一个特殊的文本文件，它指出了要编译的源程序所在的文件夹名。

SOURCES 文件是位于源程序文件夹中的一个特殊的文本文件，它包含了一些宏定义，编译程序就是利用这些宏定义作为选项来决定如何编译和链接这些源程序的。

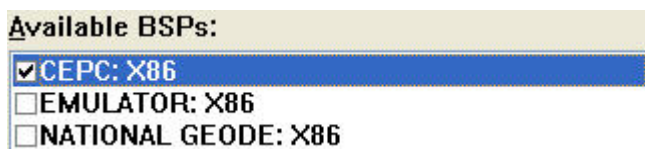
我们可以自己用记事本工具来打开这样的文件看看，如果想了解具体每个宏的含义请参阅 PB 的帮助文档。

好了，这章就到这里了，有了这些基础，更多精彩内容马上就会开始了。

WinCE 实验教程之四 CE 的引导

好了，以上我们已经生成了自己的系统，也明了了 PB 的文件夹结构，这一次我们一起来看看如何把我们自己的系统在目标设备上运行起来，此处我们的目标设备是 X86 结构的，因此我们就以这种设备为例，当然如果你没有自己的硬件平台，也可以用另一台 PC 机来代替。

按以前学过的内容，我们还是用 Hello 这个系统，但是要重新做一个了，因为上一次我们做的那个是在模拟器上用的，这次我们要做一个用在设备上的，我们可以在向导中选择 CEPC 来做，如图：



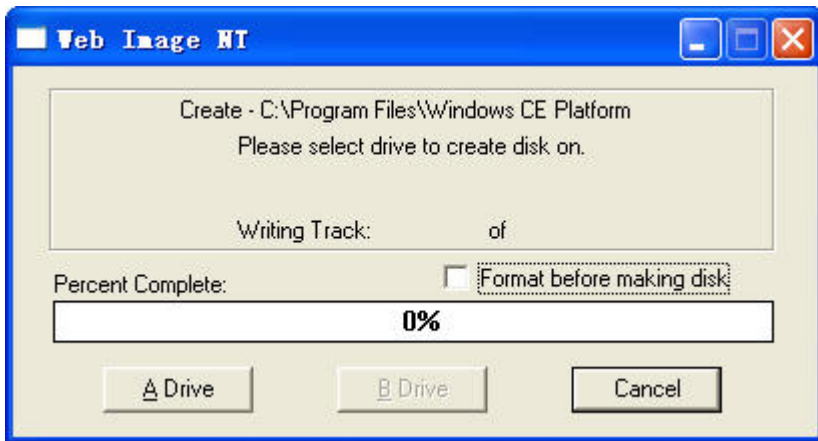
当然我们也可以同时选中 CEPC 和 EMULATOR，这样就既可以在 CEPC 设备上使用，也可以在模拟器上使用了，不过不推荐这样用。

当编译完以后，我们到 public 文件夹找到这个项目的文件夹，在 RelDir 文件夹下有两个子文件夹，一个是 debug 版，一个是 Release 版，前者是允许调试的，包含一些调试信息，后者是发布用的。通常在调试阶段我们都使用 debug 版。我们在这个文件夹下找到 nk.bin 文件，以前说过，这个就是我们系统的最终文件了。但此时我们还没有办法让它运行，下面我们来看让它运行的方法。

CE 的启动程序也叫做 BootLoader 程序，它可以引导起 CE 操作系统，即将 nk.bin 文件装入内存并解包，然后运行操作系统的内核。以后我将会专就 BootLoader 程序来写的。这里我们由于刚接触 CE，所以先不用写专用的启动程序，我们就用 DOS 引导盘来引导系统然后加载 CE 就行了。

说到 DOS 引导盘，在这个 WINDOWS XP 都普及的年代看来是很难找了，还好 PB 为我们想到了这一点，它带了一个工具能够制作 DOS 引导盘，下面我们就先来做这件事情。

我们先到 PB 的程序文件夹下找到这个文件夹“Program Files\Windows CE Platform Builder\4.20\cepb\utilities”，在这里有一个 websetup.exe 的可执行文件，让我们先来运行它。它会提示你安装，单击“Install”按钮，瞬间就安装完成了。它安装完成以后我们就可以接下来运行那个奇怪的 cepcboot.144 文件了，运行后的界面如下：



找张软盘来吧，一个 DOS 引导盘马上就能做出来了。好像还是 DOS6.22 的。做完启动盘后，这张盘就可以引导 CE 系统了，我们来看一下这张盘中文件组成：



其

中 DOS 部分的文件在这里就不说了，我们来看 eboot.bin，这个文件是用来通过网络来加载 nk.bin 的文件，sboot.bin 文件是用来通过串口加载 nk.bin 的文件，如果你要通过网络或串口来加载 nk.bin 那么可以选择其中的一个。Loadcepc.exe 是一个可以在 DOS 下执行的加载 CEPC 架构 nk.bin 的程序，就是因为它才能通过网络或串口或根文件夹来启动 nk.bin。

对于 config.sys 文件，它里面为我们提供了一个启动菜单，我们可以选择使用什么样的屏幕分辨率以什么方式启动系统，而 autoexec.bat 则对应每个菜单项执行了不同配置的

loadcepc.exe 程序。具体的 loadcepc.exe 的用法可以参见 readme.txt 文件。

另外,如果你要想在目标机上直接启动 C: 盘根目录上的 nk.bin, 那么请更改 autoexec.bat 文件, 让它加载 nk.bin 而不要加载 eboot.bin。如果你只用到其中的一项, 那么也可以按你的要求把 config.sys 和 autoexec.bat 文件进行修改。

以上只是一种实现方式, 如果你手中已经有了启动盘, 也可以直接到 WINCE420\PUBLIC\COMMON\OAK\CSP\I486\DOS\BOOTDISK 下把其中的文件 COPY 到你的启动盘, 这与用上述方法得到的结果是一样的。

有了以上的准备, 我们就可以把 nk.bin 在目标设备上引导了, 如果你目标设备的网卡是如下类型的:

SMC9000

NE2000

DEC21140

RTL8139

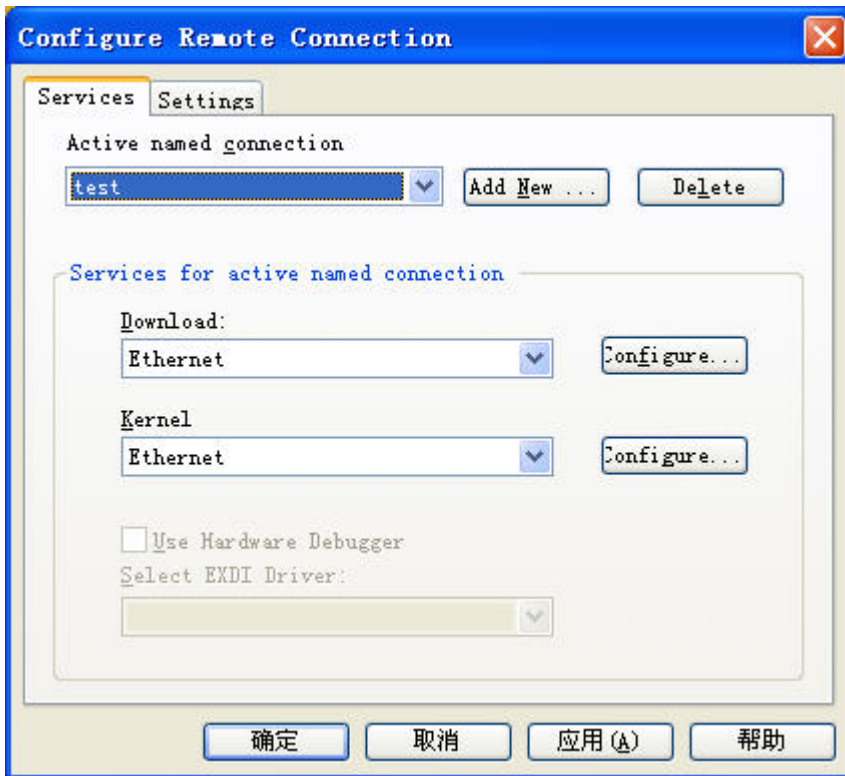
DP83815

3C90X

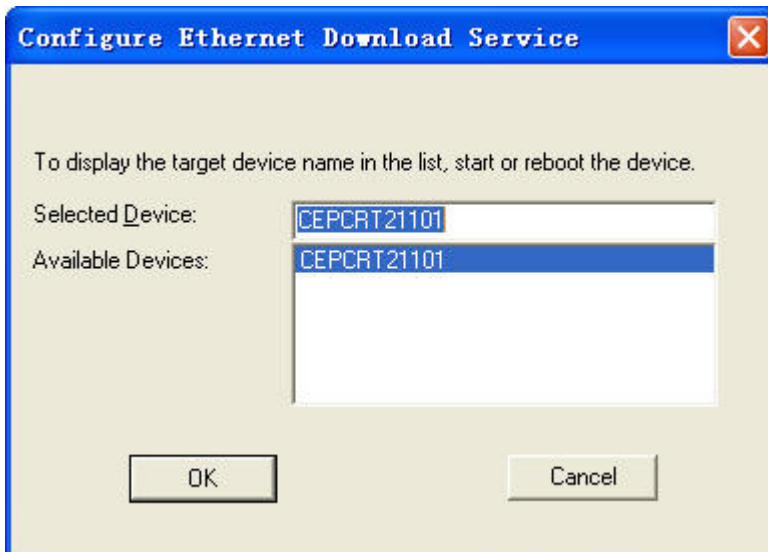
那么都是支持远程调试的, 可以直接在 PB 中 Download 到设备中去, 否则可以把 nk.bin 文件 COPY 到目标设备的 C: 盘根文件夹下。

我们下面具体看一下使用网线远程启动目标设备的过程。

首先将启动盘插入目标设备的软驱中, 打开目标设备的电源, 当目标设备启动后我们看到屏幕上提示 Jump XXXXXXXX, 此时该目标设备的网卡向网络广播 BootMe 的消息, 我们可以在开发机上使用 PB 中的"target"菜单下的"Configure Remote Connection", 在弹出的对话框中配置如下:



单击此对话框中的“Configure”按钮，在出现的对话框中会找到设备的名字：



一切 OK，执行“target”菜单下的“Download/Initialize”命令就可以把 nk.bin 下载到目标设备并启动了。

如果无法找到目标设备可试着将目标设备重新启动或更改引导盘中的 autoexec.bat 中的网络 IP 地址等信息以重试。

好了，快让你的系统在真实的设备上启动起来吧。

WinCE 实验教程之五高级调试

回顾一下我们以前学过的内容，我们学习了嵌入式系统的基本概念，学习了如何定制一个操作系统，还学习了如何引导一个 CE 操作系统。这次我们来学习 CE 系统的高级调试技术，主要是利用 PB 的集成开发环境来调试操作系统的内核和设备驱动程序。

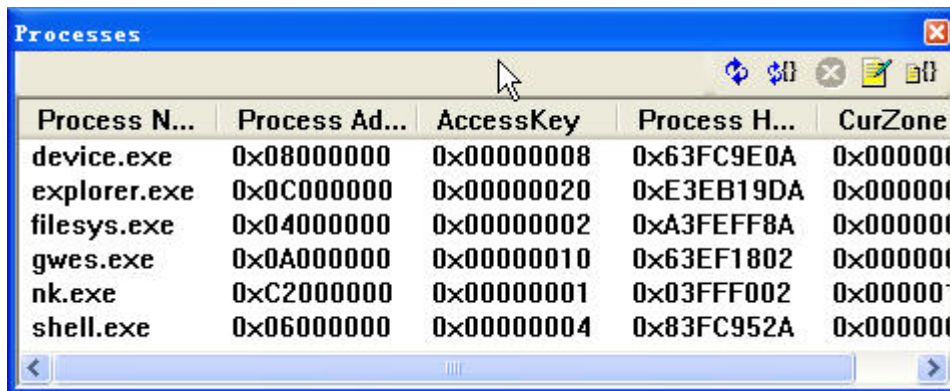
PB 集成开发环境强大的调试功能为我们提供了可以深入调试系统程序的途径，它的操作方式与常规的集成开发环境如 VC++ 等非常相似，但功能却比它们强得多，因为利用 PB 我们可以调试到一些系统内核部分的程序如内核组件、IE 组件，也可以调试各种设备驱动程序如串口驱动、网卡驱动等，因此，我们才能够更加方便的定制专用的系统。

下面我们先来打开上次我们用的 Hello 的那个系统，我们定制它时是在模拟器上来模拟的，为了讲解方便，也为了更多的人能够方便的学习，我们以后的内容尽量使用模拟器。但是我们必须清楚意识到，模拟器毕竟不是真实的目标机器，哪怕你用一台 PC 机来作为 CE 的目标机器都比使用模拟器好，因此有条件的学习者还是使用真实的目标设备为好。

我们使用 Debug 的配置来重新编译 Hello 平台，假设目前我们要调试串口设备的驱动程序，那么我们先要找到串口驱动程序的源程序，这个可以通过 Workspace 窗口来找到，在设备驱动程序下的串口设备中找到并双击 com16550.c 源程序，就可以在主窗口中打开这个源程序了，在程序中设置断点以后当程序执行到断点处后就会中断，然后由我们通过在 PB 中使用调试工具栏上的单步跟踪等调试方法来进行驱动程序的调试，在调试工具栏中还可以查看到内存、调用堆栈、反汇编、变量、寄存器等调试工具。由于此处虽然使用核心调试功能，但程序的调试与常规的程序调试方法是相同的，所以此处不做更多的举例。

我们来看一下对 CE 目标设备的进程、线程等的控制。

将编译好的平台下载到目标设备上，我们打开 Target 菜单，使用“CE Processed”命令打开一个 CE 的进程管理窗口如下图：

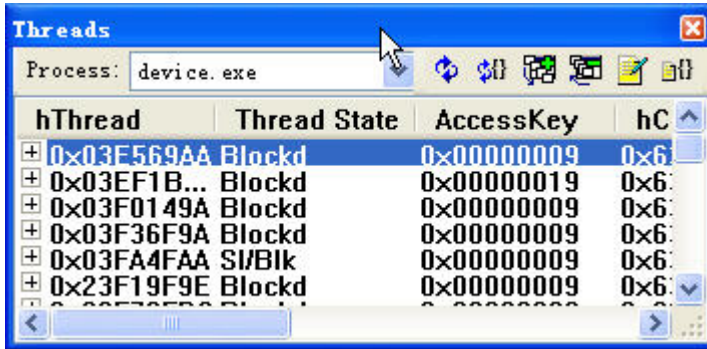


Process N...	Process Ad...	AccessKey	Process H...	CurZone
device.exe	0x08000000	0x00000008	0x63FC9E0A	0x00000001
explorer.exe	0x0C000000	0x00000020	0xE3EB19DA	0x00000001
filesys.exe	0x04000000	0x00000002	0xA3FEFF8A	0x00000001
gwes.exe	0x0A000000	0x00000010	0x63EF1802	0x00000001
nk.exe	0xC2000000	0x00000001	0x03FFF002	0x00000001
shell.exe	0x06000000	0x00000004	0x83FC952A	0x00000001

在这个窗口中我们可以看到每一个 CE 进程的信息，同时在工具栏上还可以控制中止某

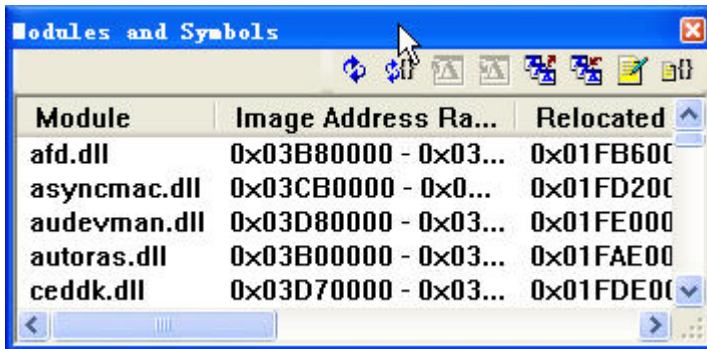
一个进程或打开某一个进程的线程。

同样的，我们使用“CE Threads”命令可以打开线程窗口如下图：

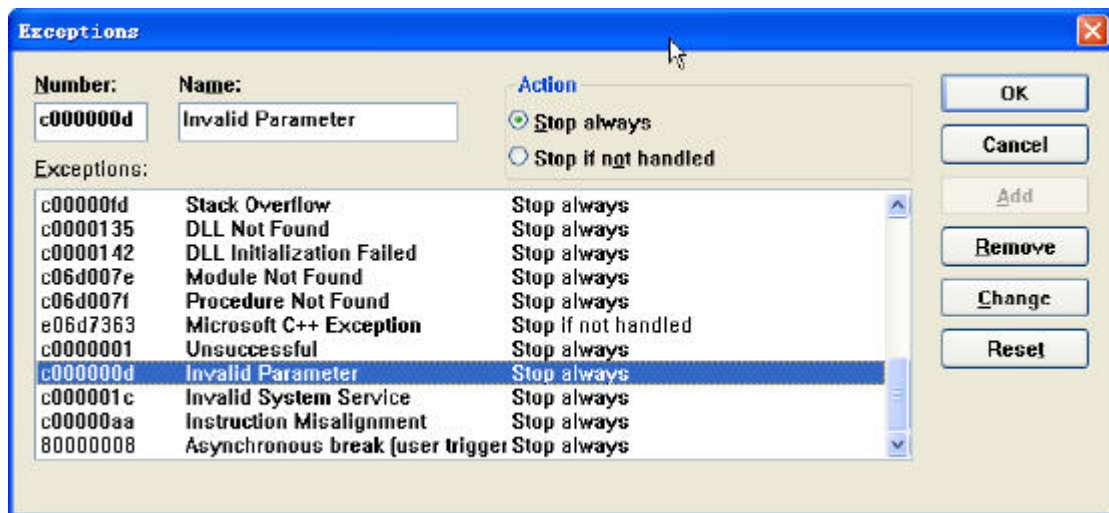


在这个窗口中我们可以通过工具栏上的进程列表框来选择查看某个进程的所有线程的详细信息，包括调用堆栈及各种时间信息等。

如果了解目标平台中的模块的地址信息和重定位信息等，我们就可以使用“CE Modules and Symbols”菜单命令打开如下的窗口：

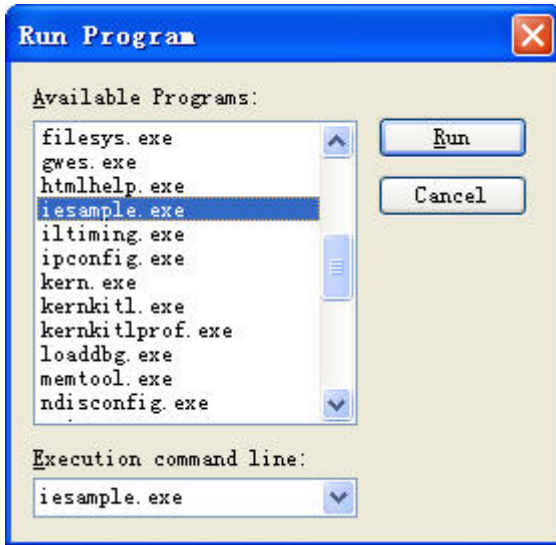


利用“CE Exceptions”菜单命令则可以改变 CE 对异常响应的行为，窗口如下图所示：



利用以上的四个功能我们可以对 CE 的进程、线程、模块和异常响应进行必要的管理，也是以后常用的命令。如果想进一步的控制 CE，我们还可以使用以下三个命令。

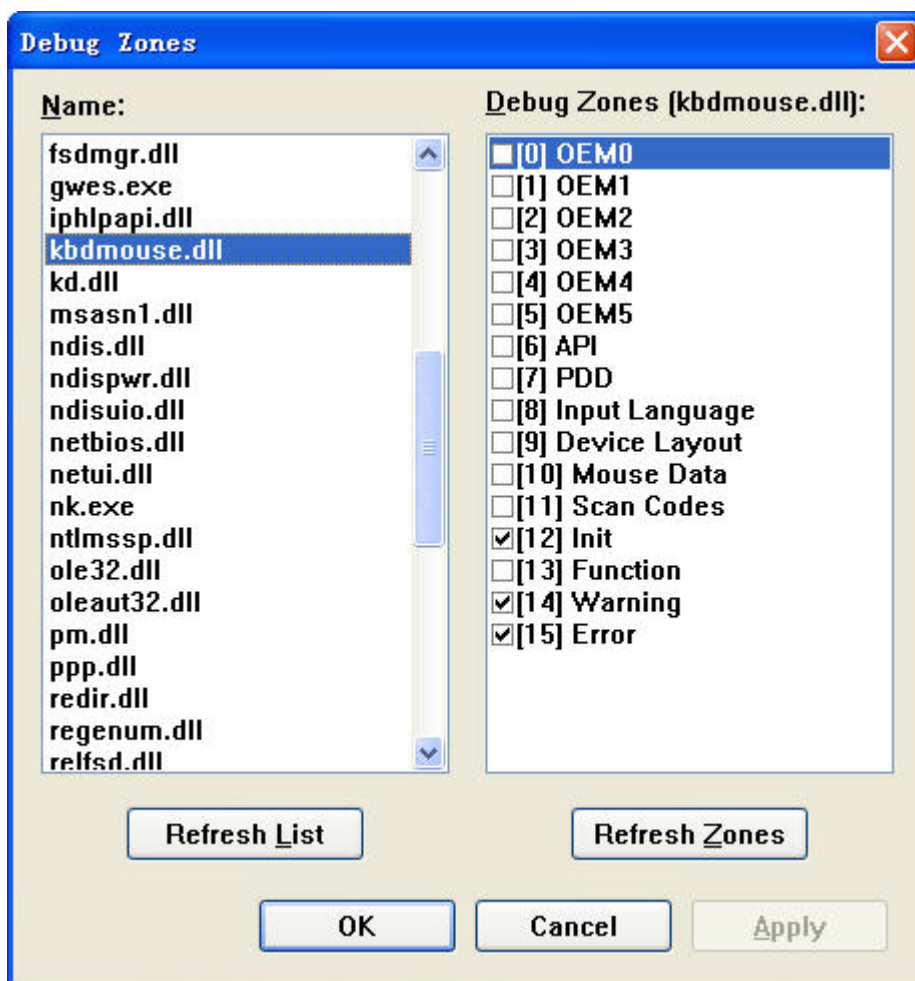
使用“Run Programs”命令可以在开发机上控制目标机中运行某个指定的程序，此命令运行后打开的窗口如下图：



如果我们在其中选择 iesample.exe 然后点击“Run”按钮，则会在目标机上发现 IE 浏览器已经运行起来了。如果开发的目标设备上无显示功能，则可以通过这个无窗运行其中的程序。

还有一个“CE Target Control”的命令，它可以打开一个设备控制窗口，在其中输入设备控制命令后也能控制目标设备动作，如果有兴趣使用这些设备控制命令的话可以查一下 PB 的帮助。

另一个高级调试功能是使用调试区，它是 CE 中很常用的调试功能，具体它的代码实现及要求我们将在设备驱动程序中进行讲解，这里我们主要看一下它的应用方式，举个具体的例子，如果我们要跟踪目标机上键盘的动作以查看键盘驱动程序的工作情况，那我们就要用到调试区了。执行“CE Debug Zone”命令打开调试区窗口如下图：



左侧列表是模块名称，右侧是某一模块对应的调试区域，如上，我们选中 kbdmouse.dll 模块，右侧就列出了该模块的 16 个调试区，默认已经勾选了三个，所以当这个 dll 被调入内存执行时就会在 PB 的 Debug 窗口中输出它的初始化信息、警告信息和错误信息，如果我们想在 Debug 窗口中得到某个按键的扫描码，那么只要勾选 Scan Codes 项就可以了，确定以后，我们到目标设备上按键盘，就可以在开发机 PB 的 Debug 窗口上看到如下内容：

```
1468296 PID:63ef1802 TID:e3ec1eda 0x83ebe9f8: KeybdPdd_GetEventEx2: scan code
0x0000001c, code in progress 0x00000000, previous 0x00000000
```

```
1468296 PID:63ef1802 TID:c3ec4062 0x83ef1948: KeybdEventThreadProc: keybd 0 scan
code 0x0000001c Down
```

```
1468296 PID:63fc9e0a TID:e3ed3e22 0x83eeee44: OpenSwMixer - no mixer found
```

```
1468296 PID:63fc9e0a TID:e3ed3e22 0x83eeee44: SPS - open failed : 0x1
```

```
1468300 PID:63fc9e0a TID:e3fc5a32 0x83eb1830: OpenSwMixer - no mixer found
```

```
1468300 PID:63fc9e0a TID:e3fc5a32 0x83eb1830: SPS - open failed : 0x1
```

```
1468425 PID:63ef1802 TID:e3ec1eda 0x83ebe9f8: KeybdPdd_GetEventEx2: scan code
```

0x000000f0, code in progress 0x00000000, previous 0x0000001c

1468430 PID:63ef1802 TID:e3ec1eda 0x83ebe9f8: KeybdPdd_GetEventEx2: scan code
0x0000001c, code in progress 0x00000000, previous 0x0000001c

1468430 PID:63ef1802 TID:c3ec4062 0x83ef1948: KeybdEventThreadProc: keybd 0 scan
code 0x0000001c Up

这其中就是调试区返回给 Debug 窗口的信息，其实质就是在做 kbdmouse.dll 这个驱动程序时实现了对应的调试区。

好了，常规调试也好，高级调试也罢，目的只有一个，就是要排除程序中的错误，得到必要的信息，快试试吧！