Re: How do I port my CE 5.0 PCI bus writing code to 6.0?

Source:
http://www.tech−archive.net/Archive/WindowsCE/microsoft.public.windowsce.embedded/2007−03/msg00244.html

- *From*: "Dean Ramsier" <ramsiernospam@xxxxxxxxxx>
- *Date*: Wed, 21 Mar 2007 16:42:58 −0400

> In general CE 5.0 apps will run on CE 6.0.

"Well behaved" applications. Meaning those that don't do things normally associated with drivers, and don't use undocumented APIs

−−
Dean Ramsier − eMVP
BSQUARE Corporation

"Bruce Eitman [eMVP]" <beitman.nospam@xxxxxxxxxxxxxxxxxxxxxx> wrote in message news:eRGrn%23%23aHHA.4520@xxxxxxxxxxxxxxxxxxxxxx

> In CE 6.0 the rules have changed. You cannot access hardware from your application anymore. You will need a kernel mode driver, this is a new concept for CE so you will need to refer to the documentation.
>
> So yes, you need to write a driver or implement something in the kernel itself.
>
> In general CE 5.0 apps will run on CE 6.0.
>
> −−
> 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
>
> "Don" <Don@xxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
> news:7A34B4E0−F837−4650−9F44−39B9DDB72C30@xxxxxxxxxxxxxxx

Hi Blake,

Thanks for the reply. I am not sure what is meant by "move them to kernel mode"?
I am not even sure how to compile a C program in Visual studio 2005 since this is not one of the options in when I try to Open a new project.
Should I
still be using Platform Builder 5.0 just in some other mode?
−−
Don

"Blake" wrote:

> Hi Don,
>
> You cannot access all memory address in WinCE6.0. It has been divided
> into
> Kernel mode and User mode.
> You may not access hardwares directly via an application of User mode in
> WinCE6.0. Try to move them to Kernel mode or access through a driver.
>
> −−
> I hope it is useful ^_^
>
> Best Regards,
>
> Blake Chang
>
> "Don" <Don@xxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
> news:2A2B9EA1−DA38−41AF−B3FC−CCBB3B32B3CD@xxxxxxxxxxxxxxx
>
> > Hi
> >
> > Is the following code portable to the
> > windows CE 6.0?:
> >
> > // PCIenumerate.cpp : Defines the entry
> > point for the application.
> > //
> >
> > #include "stdafx.h"

```
//
// Copyright (c) Microsoft Corporation. All
rights reserved.
//
//
// Use of this source code is subject to the
terms of the Microsoft
end−user
// license agreement (EULA) under which
you licensed this SOFTWARE
PRODUCT.
// If you did not accept the terms of the
EULA, you are not authorized
to
use
// this source code. For a copy of the EULA,
please see the
LICENSE.RTF on
your
// install media.
//
#include <windows.h>

#ifndef RUN_ON_PC
#include "types.h"
#include "ceddk.h"
#include <fcntl.h>
#include <stdarg.h>
#include "PCIDriver.h"

PCI_COMMON_CONFIG AlteraPciConfig;


VOID
DumpPciConfig(int Bus, int Device, int
Function, PPCI_COMMON_CONFIG
pPciConfig);



int WINAPI
WinMain(HINSTANCE hInstance,
HINSTANCE hPrevInst, LPWSTR
lpCmdLine,
int
nCmShow)
{
PCI_SLOT_NUMBER slotNumber;
PCI_COMMON_CONFIG pciConfig;
int bus, device, function;
```

```
int length;

for (bus = 0; bus < PCI_MAX_BUS; bus++)
{
for (device = 0; device <
PCI_MAX_DEVICES; device++)
{
slotNumber.u.bits.DeviceNumber = device;

for (function = 0; function <
PCI_MAX_FUNCTION; function++)
{
slotNumber.u.bits.FunctionNumber =
function;

length = HalGetBusData(
PCIConfiguration, bus,
slotNumber.u.AsULONG,
&pciConfig, sizeof(pciConfig) −
sizeof(pciConfig.DeviceSpecific));

if (length == 0 || pciConfig.VendorID ==
0xFFFF)
{
break;
}

DumpPciConfig(bus, device, function,
&pciConfig);

if (function == 0 &&
!(pciConfig.HeaderType & 0x80))
{
break;
}
}

if (length == 0)
{
break;
}
}

if (length == 0 && device == 0)
{
break;
}
}
return 0;
}
```

```
PTCHAR
DeviceType(UCHAR ucBaseClass, UCHAR
ucSubClass, UCHAR ucProgIf)
{
static struct
{
ULONG ulClass;
PTCHAR pszDescription;
} ClassNames[] =
{
{ 0x000000, TEXT("Pre 2.0 Non VGA
device") },
{ 0x000101, TEXT("Pre 2.0 VGA device")
},
{ 0x010000, TEXT("SCSI Controller") },
{ 0x010100, TEXT("IDE Controller") },
{ 0x010180, TEXT("Busmaster IDE
Controller") },
{ 0x010200, TEXT("Floppy Controller") },
{ 0x010300, TEXT("IPI Controller") },
{ 0x010400, TEXT("RAID Controller") },
{ 0x018000, TEXT("Other Mass Storage
Device") },
{ 0x020000, TEXT("Ethernet Network
Controller") },
{ 0x020100, TEXT("Token Ring Network
Controller") },
{ 0x020200, TEXT("FDDI Network
Controller") },
{ 0x020300, TEXT("ATM Network
Controller") },
{ 0x028000, TEXT("Other Network
Controller") },
{ 0x030000, TEXT("VGA Compatible
Display Adapter") },
{ 0x030001, TEXT("8514 Compatible
Display Adapter") },
{ 0x030100, TEXT("XGA Compatible
Display Adapter") },
{ 0x038000, TEXT("Other Display
Adapter") },
{ 0x040000, TEXT("Video Multimedia
Device") },
{ 0x040100, TEXT("Audio Multimedia
Device") },
{ 0x048000, TEXT("Other Multimedia
Device") },
{ 0x050000, TEXT("RAM Memory
Controller") },
{ 0x050100, TEXT("Flash Memory
Controller") },
```

```
{ 0x058000, TEXT("Other Memory
Controller") },
{ 0x060000, TEXT("Host / PCI Bridge") },
{ 0x060100, TEXT("PCI / ISA Bridge") },
{ 0x060200, TEXT("PCI / EISA Bridge") },
{ 0x060300, TEXT("PCI / MCA Bridge") },
{ 0x060400, TEXT("PCI / PCI Bridge") },
{ 0x060500, TEXT("PCI / PCMCIA
Bridge") },
{ 0x060600, TEXT("NuBus Bridge") },
{ 0x060700, TEXT("CardBus Bridge") },
{ 0x068000, TEXT("Other Bridge") },
{ 0x070000, TEXT("8250 Compatible
Communications Device") },
{ 0x070001, TEXT("16450 Compatible
Communications Device") },
{ 0x070002, TEXT("16550 Compatible
Communications Device") },
{ 0x070100, TEXT("Parallel Port") },
{ 0x070101, TEXT("Bidirectional Parallel
Port") },
{ 0x070102, TEXT("ECP 1.X Parallel Port")
},
{ 0x078000, TEXT("Other Communications
Device") },
{ 0x0C0000, TEXT("Firewire (IEEE 1394)
Bus Controller") },
{ 0x0C0100, TEXT("ACCESS.bus
Controller") },
{ 0x0C0200, TEXT("SSA Controller") },
{ 0x0C0300, TEXT("UHCI USB Bus
Controller") },
{ 0x0C0301, TEXT("OHCI USB Bus
Controller") },
};
#define NUMBER_CLASS_NAMES
(sizeof(ClassNames) /
sizeof(ClassNames[0]))

ULONG ulClassKey;
int i;

ulClassKey = (ucBaseClass << 16) |
(ucSubClass << 8) | ucProgIf;

for (i = 0; i < NUMBER_CLASS_NAMES;
i++)
{
if (ClassNames[i].ulClass == ulClassKey)
{
return ClassNames[i].pszDescription;
```

```
}
}

return TEXT("Unknown Device Type");
}

VOID
DumpPciConfig(int Bus, int Device, int
Function, PPCI_COMMON_CONFIG
pPciConfig)
{
switch (pPciConfig−>HeaderType & 0x7F)
{
case 0:
if(pPciConfig−>VendorID == 0x1895)
{
AlteraPciConfig = *pPciConfig;
StartHere();
}
break;

}
}



VOID HwOutpl(UCHAR Port,ULONG
Value)
{

PULONG PortAddress;
PortAddress =
(PULONG)((AlteraPciConfig.u.type0.BaseAddresses[1]
&
~0x3) +
Port);
WRITE_PORT_ULONG(PortAddress,Value);
}



ULONG HwInpl(UCHAR Port)
{
ULONG Result;
// ULONG Read;
Result =
READ_PORT_ULONG((PULONG)((AlteraPciConfig.u.type0.BaseAddresses[1]
& ~0x3)+ Port ));
return Result;
}
```

```
VOID
DebugPrintf(PTCHAR pszFormat, ...)
{
static TCHAR szString[256];
static UCHAR szAnsiString[256];
static int hConsole;
va_list args;
char mbstr[256];

va_start(args, pszFormat);

wvsprintf(szString, pszFormat, args);

wcstombs( mbstr, szString, 255);
printf("%s\n",mbstr);

OutputDebugString(szString);

va_end(args);
}

#endif
```

I am trying to Port the above code to Windows CE 6.0. I used to compile the above code in Platform Builder 5.0 as a C program. I don't notice a way to write a C program in Visual Studio 2005 with the Windows CE 6.0 add on. Is there a way? When I was using Platform Builder I had decided that I did not need a driver in the form of a DLL and just wrote directly to the PCI Bus with the above code in a console app using Platform Builder 5.0. The exe built in Platform Builder 5.0 does not run under Windows CE 6.0. Is this what I should expect, exes built for 5.0 will not run on 6.0? Do I really need to write a proper driver DLL to talk to

> the
> PCI
> bus?
> −−
> Don

.