

Re: Setting up registers on PXA270?

Source:

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2008-04/msg00789.html>

- *From:* "Paul G. Tobey [eMVP]" <p space tobey no spam AT no instrument no spam DOT com>
 - *Date:* Tue, 29 Apr 2008 08:48:40 -0700
-

Then that's a characteristics of your Platform/BSP. That's where the code to configure the memory region where the DSP will be mapped should appear. I'd probably do it in the bootloader myself, but that's not mandatory. You could do it in or from a call made in OEMInit() in your platform code.

If you are **also** using GPIO to interface to this hardware, the initialization of the direction and initial state of those pins should, to my way of thinking, also be set up in the same place. Since all of this stuff is completely specific to your hardware design, you'd never use the driver anywhere else, but I prefer to have those things set in a central location where you can document the connection between the schematic for the hardware and what you're setting up the processor to do.

Paul T.

"Steve Conner" <connermcsteve@xxxxxxxxxxxxxxxx> wrote in message <news:a06a7353-5003-44ca-b6dd-a22f5aa49e4a@xx>

Hi FoolBlah, thanks for the reply...

I'm mapping the DSP's host port into memory using the PXA270's VLIO feature, roughly following the guidelines in the Intel app note for connecting an IDE hard drive.

To make this work, I need to change the direction, level and function of a GPIO pin (enabling the relevant chip select line) and change one of the memory controller registers to turn the VLIO on. I've tried this through RedBoot and seen the relevant things toggling on an oscilloscope.

My only worry is that the BSP already includes a driver for the GPIO pins that might get in my way. If it loads before my driver, does that mean it could stop me accessing the registers? And if it loads after mine, it could clobber the changes that I've made.

steve

Re: Setting up registers on PXA270?

On Apr 29, 12:04 pm, FoolBlah <foolb...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote:

Generally, a good practice is to setup the peripheral controller, GPIOs, etc. that you will be using in a specific driver at driver init time. This cannot always happen, but it makes debugging easier (for you and someone else that might look at the code someday) if you can architect it in this fashion.

What you are thinking is correct and yes you can use MmMapIoSpace() from the DDK to map the memory space. There is a possibility that what you plan on doing may clobber some other user of these peripherals. Do a search in the BSP for the various peripherals that you plan on using to be sure when you configure them you do not break something else in the process.

What controller/peripherals do you plan on utilizing?