

RE: IPSM w/ P30 Flash Results in Registry Corruption

Source:

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2006-05/msg00259.html>

- *From:* authorwjf <authorwjf@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 6 May 2006 21:33:03 -0700
-

Thank you for your response. I will look into CE 5 and see what the differences are.

"anonymous" wrote:

Until CE 5.0, we've always had the problem with IPSM drivers on K3, J3, and P30 parts not getting completely flushed to the flash chip. We have a soft reboot function which closes the file system gracefully which will flush to the flash chips, but otherwise, until CE 5.0, there isn't anyway to get the OS to flush to the FSD.

In CE 5.0, there was an IOCTL added which forces the file system to flush all data to the FSD. This solves this problem. You just have to add that IOCTL call after you do a regflush.

"authorwjf" wrote:

We recently upgraded our product to use P30 flash chips. Previously, our flash memory had no boot blocks. Specifically, our configuration consists of two banks of flash memory, each consisting of two 128MBit chips (for a total of 4 chips). The result is we have one application bank and one boot bank of flash. On the boot bank we keep our bootloader config block, along with the OS. The upper bank is used to store the windows registry and a user file volume. I think this is a pretty standard configuration for a WINCE / XSCALE platform.

Since changing to the P30 chips, I have been unsuccessful in getting the registry to work. When the registry gets flushed, it becomes corrupted. Our flash chips have a bottom boot block and I thought all I had to do was add an offset of 128K * 2 or 0x40000 to the physical and virtual starting addresses in PLAT_API.C. Apparently this is not enough. Below is my PLAT_API.C.

Not

part of this module are the defines for physical and virtual flash. These are 0x04000000 and 0x0x9D000000 respectively.

RE: IPSM w/ P30 Flash Results in Registry Corruption

```
#define P30_OFFSET 0x40000

#define P30_LENGTH 0x1FC0000

// note: see eboot/main.c & update EraseRegFromFlash parameters if these
are
changed:

UINT8 PLAT_DefineArrayParameters(PSM_OEMInfoPtr oem_info_ptr)

{

static const PSM_OEMInfo oem_info = {

/* PhysicalReadStartAddress */
(VOID_PTR)(APP_FLASH_BASE_PHYSICAL +
P30_OFFSET),

/* PhysicalWriteStartAddress */
(VOID_PTR)(APP_FLASH_BASE_PHYSICAL+
P30_OFFSET),

/* PhysicalBurstReadStartAddress */
(VOID_PTR)(APP_FLASH_BASE_PHYSICAL+
P30_OFFSET),

/* VirtualReadStartAddress */
(VOID_PTR)(APP_FLASH_BASE_C_VIRTUAL+
P30_OFFSET), /* cache address */

/* VirtualWriteStartAddress */
(VOID_PTR)(APP_FLASH_BASE_U_VIRTUAL+
P30_OFFSET), /* flash address */

/* VirtualBurstReadStartAddress */
(VOID_PTR)(APP_FLASH_BASE_C_VIRTUAL+
P30_OFFSET), /* cache address */

/* MaximumArrayLength */ P30_LENGTH,

/* MaximumRegistrySize */ 0x00080000,

/* ManagedAreaStart */ 0, /* boot block on P30 flash accounted for already
*/

/* ManagedAreaLength */ P30_LENGTH,

/* ReadBusWidth */ 32,

/* WriteBusWidth */ 32,
```

RE: IPSM w/ P30 Flash Results in Registry Corruption

```
/* BurstReadBusWidth */ 0
```

```
};
```

The PSM drive seems to format and the user volume is there and works. However if I do a regflush, on the next reboot the registry is corrupted. Does anyone know if this could be a problem with using (v)3.5 of PSM or (v)4.1 of WINCE? With so few parameters at my disposal it doesn't seem there is a limited number of places I could be at fault.

Thanks in advance to anyone who takes the time to comment :)