The problem of Reading unique Device ID of MMC/SD card on PXA270

Source:
http://www.tech−archive.net/Archive/WindowsCE/microsoft.public.windowsce.embedded/2007−01/msg00236.html

- *From*: "alexu" <xalexu@xxxxxxxxx>
- *Date*: Wed, 24 Jan 2007 12:58:53 +0800

Hi!

the os's version is WindowsCE5.0.
I use these two functions to map the physical address to the virtual
address in
order to access physical registers in User Mode:

BOOL VirtualCopy(LPVOID lpvDest, LPVOID lpvSrc, DWORD cbSize, DWORD
fdwProtect);
LPVOID VirtualAlloc(LPVOID lpAddress, DWORD dwSize, DWORD
flAllocationType, DWORD flProtect);

Then i got the registers' physical address about controling MMC\SD card
successfully;After that, i send CMD2(after this ,
SD/MMC card will response with a 128 BITS serial number)via MMC_CMD
register.then i was waiting for the END_CMD_RES
interupt ,but the this interupt's bit was always not set,so i can't get the
serial number of SD/MMC;
i think that the reason maybe that i didn't pay attention to something
about accessing physical address in USER
MODE in WindowsCE; who can help me! Thanks!

these my codes:

```
int PXA270MMCSerialNumberRead( DWORD * buffer )
{
volatile PxaMMCReg * pMMCReg;

pMMCReg = ( volatile PxaMMCReg * )MmMapIoSpace(
(void*)MMC_PHY_BASE, sizeof(PxaMMCReg), FALSE );//this run successfully

if ( pMMCReg == NULL )
{
return FALSE;
}

//send STOP−TRANSMISSION to stop current data transfermation
```

```
unsigned int cmdatstop = pMMCReg−>rMMC_CMDAT;
cmdatstop |= CMDAT_RESP_R1;
cmdatstop |= (1<<10);

pMMCReg−>rMMC_CMDAT = cmdatstop;//there is a bit in MMC_CMDAT[10]
seems like
//that control STOPTRANS
pMMCReg−>rMMC_CMD = 0x4c;//cmd12
pMMCReg−>rMMC_CMDAT = cmdatstop;


unsigned int cmdat = pMMCReg−>rMMC_CMDAT;

//set the way of responsing
cmdat |= CMDAT_RESP_R2;

//we set command register
pMMCReg−>rMMC_CMD = 0x42;//cmd2

//pMMCReg−>rMMC_ARGH = 0;
//pMMCReg−>rMMC_ARGL = 0;
pMMCReg−>rMMC_CMDAT = cmdat;

//enable the END_CMD_RES interupts
//pMMCReg−>rMMC_I_MASK &= ~END_CMD_RES;
unsigned int imask = pMMCReg−>rMMC_I_MASK;
imask &= ~END_CMD_RES;
imask &= (0<<13)|(0xfff);
pMMCReg−>rMMC_I_MASK = imask;

while(!( pMMCReg−>rMMC_I_REG & ( 1<<2 ) ) )//weitting for MMC_RES
interupts occured, but this always didn't occured

//read the response from MMC disk
int i;
u32 resp[4];
u32 v = pMMCReg−>rMMC_RES & 0xffff;

for (i = 0; i < 4; i++)
{
u32 w1 = pMMCReg−>rMMC_RES & 0xffff;
u32 w2 = pMMCReg−>rMMC_RES & 0xffff;
resp[i] = v << 24 | w1 << 8 | w2 >> 8;
v = w2;
}

pMMCReg−>rMMC_I_MASK |= END_CMD_RES;

memcpy( buffer, resp, sizeof(u32)*4 );

VirtualFree((void *)pMMCReg, 0, MEM_RELEASE);
```

```
pMMCReg = NULL;

return TRUE;

}
```

.