

Inserting and Removing the PDA From the PlayStation

If a PocketStation (PDA) application is written so that it can operate when the PDA is inserted in the PlayStation (PS), the functions described below must be implemented so that the PDA can communicate (be recognized) as a Memory Card. This is unnecessary, however, if the application is written so that it exits and returns to the clock screen or another screen while the PDA is plugged into the PS.

Communication with the PS is performed by the PDA kernel. When the PDA is inserted or removed, the PDA application must explicitly switch communication with the PS on and off. The following is a description of kernel operations that take place when the PDA is inserted or removed from the PS as well as the operations that the PDA application should perform.

- Kernel operations performed when the PDA is inserted or removed from the PS

When the PDA is inserted or removed from the PS, a BATIRQ is generated. The kernel performs the following operations during BATIRQ interrupt handling. BATIRQ must be enabled beforehand.

- Insertion
 - Invert LCD. Turn off DAC, speaker, IR, LED.
 - Set bit8 of PDA status to 1.
- Removal
 - Normal LCD display.
 - Set bit8 of PDA status to 1.

- Kernel operations during communication with the PS

A SPIFIQ is generated during communication with the PS. Communication is performed by the kernel in the SPIFIQ interrupt handling routine. Since communication takes up significant processor resources, PDA application speed can be affected as a result. Also, if TC2FIQ interrupt handling is resource intensive, normal communication with the PS may not be possible. Therefore, it is safer to disable TC2FIQ when the PDA is inserted into the PS.

- System calls

In order to enable communication with the PS, a system call must be used and operation must switch to kernel mode. The following describes the operations performed by the system calls used for communication with the PS.

[SWI 5]

Checks the PDA status and updates PDA status contents if the insertion/removal flag is set. Then performs the same operations as [SWI 17 Argument 1]. Does nothing if no insertion/removal has occurred.

[SWI 6]

Returns address held in PDA status. If there has been an insertion/removal, bit8 of the PDA status is set to 1. If communication with the PS is enabled, bit9 is set to 1.

[SWI 17 Argument 1]

If the PDA is inserted in the PS, turns on communication with the PS and sets the clock frequency to 4 MHz.

If the PDA is not inserted in the PS, turns off communication with the PS.

[SWI 17 Argument 0]

Turns off communication with the PS.

- Operations that should be performed by the PDA application.

The following are examples of operations that should be performed by the PDA application.

1. Monitor the PDA status in the main loop to determine if insertion/removal has taken place.
2. If insertion/removal has taken place, determine if it was insertion or removal.
3. If insertion, turn on communication with the PS. (Clock frequency set to 4 MHz).
Disable TC2FIQ.
4. If removal, restore clock.
If necessary, turn devices such as DAC back on.

- Other issues

If the PDA application is started in response to a PS request from libmcx, the "connected to PS" status cannot be determined from the PDA status at startup. This must be determined by checking the register directly.

(This is not required for applications that do not need to be started up from the PS.)

Also, upon exit the application must always turn off communication with the PS through [SWI 17 Argument 0].

Sample program

```

AREA      MAIN_BSS, DATA, READWRITE      ; RAM AREA
status_addr      %      4      ; PDA status address

CODE32
AREA      MAIN, CODE, READONLY
main
    ; startup operations
    BL      connect_check      ; check for connection with PS
    SWI     6      ; get PDA status address beforehand
    LDR     r5, =status_addr    ;
    STR     r4, [r5]
    :

main_loop
    :
    ; main routine operations
    LDR     r5, =status_addr
    LDR     r4, [r5]      ; r4 status address
    LDR     r5, [r4]      ; r5 status contents
    TST     r5, #0x100
    BLNE   switch_mode      ; change mode if there was insertion/removal
    :
    :

switch_mode
    STMDB   sp!, {r4-r5, lr}
    SWI     5      ; If insertion, enable PS communication (4 MHz)
    ; If removal, disable PS communication

    LDR     r5, =status_addr    ; recheck status
    LDR     r4, [r5]      ; r4 status address
    LDR     r5, [r4]      ; r5 status contents
    TST     r5, #0x200

```

```
; PS communication not enabled, i.e. "removed"
MOVEQ r4, #5
SWIEQ 4 ; restore clock to 1MHz
; PS communication enabled, i.e. "inserted"
MOVNE r4, #0x0a000000 ; INTStatus
MOVNE r5, #0x2000
STRNE r5, [r4, #0x0c] ; disable TC2FIQ
LDMIA sp!, {r4-r5, pc}
```

connect_check

```
STMDB sp!, {r4-r5, lr}
MOV r5, #0x0d800000 ; PIOControl
LDR r4, [r5, #0x0c] ; check PIOReadInputData
TST r4, #0x10
; when connected to PS
MOVNE r5, #0x0d000000 ; LCDControl
LDRNE r4, [r5]
ORRNE r4, r4, #0x80
STRNE r4, [r5] ; invert LCD
MOVNE r4, #1
SWINE 17 ; enable communication with PS (4MHz)
LDMIA sp!, {r4-r5, pc}
```