



# PSP Inside v0.9q

(The multipurpos PSP information Toolbox)

27th November 2005

this is the result of myself pasting together various freely available documents aswell as adding some of my own findings. have fun... additions and corrections welcome :)

**THIS IS WORK IN PROGRESS! INFORMATION CONTAINED IN THIS DOCUMENT MAY BE MISSING, INCOMPLETE OR EVEN PLAIN WRONG! NO F\*\*\*\*\*' WARRANTY IMPLIED! IF THE USE OF THE INFORMATION CONTAINED HERE RESULTS IN ULTRA REALISTIC SMOKE EFFECTS, BRAIN DAMAGE OR LOSS OF PHYSICAL AND/OR MENTAL HEALTH PLEASE DON'T COME BACK AND SAY YOU HAVEN'T BEEN WARNED! YOU SHOULDN'T BE USING THIS IN THE FIRST PLACE!**

groepaz/hitmen (groepaz@gmx.net)

Hitmen-Console <http://www.hitmen-console.org>



## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Usermode Notice . . . . .	4
1.2	Rant . . . . .	4
1.3	Conventions . . . . .	4
<b>2</b>	<b>Prerequisites</b>	<b>6</b>
2.1	Installation . . . . .	6
2.2	Files . . . . .	6
<b>3</b>	<b>Startup</b>	<b>7</b>
3.1	autoload . . . . .	7
<b>4</b>	<b>Tabs Overview</b>	<b>8</b>
4.1	System Menu . . . . .	8
4.2	Memory Menu . . . . .	11
4.3	Disassembler Menu . . . . .	13
4.4	Register Menu . . . . .	15
4.5	Syscalls Menu . . . . .	16
4.6	IRQs Menu . . . . .	17
4.7	Videoram tab . . . . .	18
4.8	Patch Menu . . . . .	19
4.9	Console Tab . . . . .	20
4.10	Profiler Tab . . . . .	21
4.11	Config Tab . . . . .	22
4.12	About Tab . . . . .	23

<b>5</b>	<b>Exception Handler</b>	<b>24</b>
<b>6</b>	<b>Serial Console</b>	<b>25</b>
6.1	Parameters . . . . .	25
<b>7</b>	<b>Patch Engine</b>	<b>26</b>
7.1	Global Patches . . . . .	26
7.1.1	absolute memory writes . . . . .	26
7.2	Module Patches . . . . .	26
7.2.1	relative memory writes . . . . .	26
7.2.2	search and replace . . . . .	26
7.2.3	load additional modules . . . . .	27
7.2.4	run module . . . . .	27
7.2.5	suspend mode . . . . .	27
7.3	Example Patches . . . . .	27
<b>8</b>	<b>Usage Examples</b>	<b>28</b>
8.1	automatically suspend a thread . . . . .	28
8.2	debug Game . . . . .	28
<b>9</b>	<b>References</b>	<b>29</b>
9.1	Sources . . . . .	29
<b>10</b>	<b>Credits</b>	<b>30</b>
10.1	Contributions . . . . .	30

## 1 Introduction

Bored of playing another round on one of the countless emulators around? Thinking about what the hell is going on inside your small piece of electronic artwork? Look no further, now look inside! Here is a small plot what this app can do for you!

- ▷ Memory view of entire memory (Hex B/W/L, ASCII, Hex/ASCII)
- ▷ Disassembly of entire memory
- ▷ View coprocessor registers, even within disassembly
- ▷ Load/Start/Kill/Dump/... Threads & Modules
- ▷ Suspend/Resume threads to have full control over your PSP
- ▷ put PSPInside to background while running other apps (note button)

... and countless other Stuff to explore on yourself :)

All possible functions are listed on the bottom lines or within menus popping up if necessary.

### 1.1 Usermode Notice

The current built for 2.0 Firmware runs in Usermode so certain Functionality is not available. Please keep that in mind when using PSPInside on 2.0.

### 1.2 Rant

If you don't know what programming a machine down to the metal is all about, go away! no really, this document is not for you! if you are seeking for advice on using existing solutions, such as SDKs or libraries, you will find little to none information that is of any use for you and you might only become frustrated by figuring out how little you know. If you however aren't afraid of numbers and want to dare jumping into the snake-pit of semi-accurate information based on guesswork done by a bunch of freaks - feel invited. this was made to give you what you need in the most compressed and visually pleasing form possible. *Stuff that matters.*

### 1.3 Conventions

- ▷ we count bits starting from 0, the most significant bit of a byte is bit 7. when visualising a byte the most significant bit comes first (left), and the least significant bit comes last (right).
- ▷ when dealing with 16- or 32 byte values all figures are in big endian byte order. this means that the most significant byte comes first (left), and the least significant byte comes last (right).
- ▷ if known (from patents or other freely available sources) we use the same terminology as Sony does, in particular we try to use the same names and abbreviations for hardware registers, signals and the like as a weak attempt of providing consistency with other existing documentation.
- ▷ absolute memory addresses are shown as if the PSP had been initialized by the original BIOS and address translation had not been changed. For this matter we don't use physical addresses to avoid confusion for the majority of our readers.

- ▷ code snippets are in either real or pseudo C language. any logical or arithmetic expressions outside code snippets are loosely similar to C notation according to the following table:

Description	Symbol
logical or bitwise AND	&
logical or bitwise OR	
logical or bitwise exclusive OR	^
logical or bitwise NOT (inverse)	!
equality or assignment	=
addition	+
subtraction	-
multiplication	*
division	/

please notice that -outside code- we do not make a difference between logical and bitwise operations. if in doubt the operation is bitwise, it should however be clearly visible from the context.

## 2 Prerequisites

### 2.1 Installation

1. Mount your PSP with an USB cable.
2. In the PSP Operating-System (VSH) go to Settings -> USB Connection
3. Your PC's operation system should recognise your PSP as a removable media, and assign a drive letter to your PSP, in this example I:
4. Copy the directories `PSPINSIDE` and `PSPINSIDE%` (with all files and subdirectories) to your PSP `PSP\GAME\` directory.

Example:

`I:\PSP\GAME\PSPINSIDE`

`I:\PSP\GAME\PSPINSIDE%`

note: on Firmware v1.0 there is only one directory, same for 2.0

5. Now you can run PSPInside by going to Game -> Memory Stick -> PSPInside

Please note: The paths are fixed, so you should not change them, or the application will not run.

### 2.2 Files

#### **PSPINSIDE\CFG\SYSCALLS.TXT**

used to map systemcalls to their literal names, each line of the file contains the syscall number, its sha1 hash and the literal symbol name all separated by commas. for example:

`0x2000,0xca04a2b9,sceKernelRegisterSubIntrHandler`

#### **PSPINSIDE\CFG\PATCHES.TXT**

used to configure the patch engine

## 3 Startup

### 3.1 autoload

Files placed in the `PSPINSIDE\CFG\ALOAD` Directory are automatically loaded into the memory at startup of PSPInside if their filename consists of the loading address.

examples:

<code>88200000.bin</code> <- this file gets loaded to memory address 0x88200000 at PSPInside startup <code>88451234.bin</code> <- this file gets loaded to memory address 0x88451234 at PSPInside startup <code>-88188430.bin</code> <- this file is NOT loaded at startup (by placing the '-' in front)
--

## 4 Tabs Overview

### 4.1 System Menu

When you start PSPInside, you will be presented by a screen looking like this:

```

-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+----->
System   |Memory   |Disasm   |Register |Syscalls |IRQs
-----+-----+-----+-----+-----+-----+----->
Modulelist:                               Segments:
8800b6a8: sceSystemMemoryManager           88000000 - 8800f340
88012db8: sceLoaderCoreTool                8800f340 - 8800f86c
8801c700: sceExceptionManager
8801ed00: sceInterruptManager
8802c558: sceSysclib
8803b4d4: sceThreadManager
88045a00: sceDMAManager
8804a6f4: sceSystimer
8804d3a8: sceIOFileManager
Module 1 of 63 :
Module ID.....: 0022ef33
Name.....: sceSystemMemoryManager
Attributes.....: 00001007
Module entrypoint.: 8800b6a8
# of segments.....: 2
text seg. address.: 88000000
text seg. size....: 0000f338
data seg. size....: 0000052c
bss seg. size.....: 000005a8
gp.....: 88017860

Analog U/D : Thread -/+ | Square : Module functions
Select : Scrollmode    | Cross : Disasm(Addr.)
Triangle : Load functions | Circle : Change list
Start : USB on/off                                           \0ead0e00

```

You can select various screen menu's using the Trigger Left/Right buttons. Your options at this screen is:

- ▷ SELECT: Slow scrolling on/off.
- ▷ START: Enable/Disable USB mode, enabling this will allow you to mount/access the memory card from your PC, while working in PSPInside. Active USB mode will be indicated by a 'U' in the bottom right corner of the screen.
- ▷ TRIANGLE: This will bring up the file browser, allowing you to browse and load/execute files from Memory Stick, UMD and Flash.



```

+-----+
| /      |
+-----+
|ms0:/    |
|flash0:/ |
|flash1:/ |
|disc0:/  |
|         |
|         |
|         |
|         |
+-----+

```

- ▷ TRIANGLE : abort file browser
- ▷ CROSS: load selected file

```

+-----+
| Enter load Method |
+-----+
| 0xxxxxxx = LoadModule(<file>) |
| 1xxxxxxx = Assign disc0: to ms0: |
| 2xxxxxxx = LoadExec(<file>) |
| 4xxxxxxx = StartModule() |
| 8xxxxxxx = binload @ <addr> |
| x1xxxxxx = (whole directory) |
| x2xxxxxx = copy to ms0:/ |
| x4xxxxxx = PSPInside to back |
+-----+

```

```

+-----+-----+
|          | [0x00000000          ] |
|          | ----- |
| 01234 | Move analog Stick |
| f  5 | Cross      : Forward |
| e + 6 | Circle    : Back |
| d  7 | Square     : Return |
| cba98 | Select    : Return! |
|          | Triangle : Cancel |
+-----+-----+

```

- TRIANGLE: Cancel

- SQUARE: enter address with range checking
  - SELECT: enter address without range checking
- ▷ SQUARE: Will bring up the Module/Thread sub-menu, where you will be able to Start, Stop, Dump, Unload Modules and Threads.

```

+-----+
|Cross   : Start Module      |
+-----+
|Circle  : Stop Module       |
+-----+
|Triangle : Stop/Unload Module |
+-----+
|Start    : Dump Module      |
+-----+
|Select   : Dump all modules  |
+-----+
|Dig.Up   : Start module + sleep |
+-----+
|Dig.Down : Dump module (disasm) |
+-----+
|Square   : Return           |
+-----+

```

```

+-----+
|Cross    : Resume Thread     |
+-----+
|Circle   : Suspend Thread    |
+-----+
|Triangle : Kill Thread       |
+-----+
|Start     : Start Thread      |
+-----+
|Dig.Up    : Start Thread + sleep |
+-----+
|Square    : Return           |
+-----+

```

- ▷ CROSS: This will let you disassemble (realtime) the Module or Thread that you selected.
- ▷ CIRCLE: Switch between Modules and Threads.

## 4.2 Memory Menu

Pressing Trigger-R will advance us to the next menu, here we can browse the memory of the entire PSP.

```

-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+----->
System      |Memory      |Disasm      |Register    |Syscalls    |IRQs
-----+-----+-----+-----+-----+-----+-----
      [nX] Kernel@RAM(4MB):00000000
88000000: c0 ff bd 27 20 00 b4 af   ...' ...
88000008: 21 a0 80 00 34 00 bf af   !...4...
88000010: 2c 00 b7 af 21 b8 e0 00   ,...!...
88000018: 28 00 b6 af 1c 00 b3 af   (.....
88000020: 18 00 b2 af 21 90 c0 00   ....!...
88000028: 14 00 b1 af 21 88 a0 00   ....!...
88000030: 30 00 be af 24 00 b5 af   0...$....
88000038: 10 00 b0 af 3c 32 00 0e   ....<2..
88000040: 04 00 a0 af 21 20 80 02   ....! ..
88000048: ef 03 00 0e 21 b0 40 00   ....!.@.
88000050: 61 00 40 10 21 98 40 00   a.@.!.@.
88000058: 18 00 64 33 01 88 15 3c   ..d3...<
88000060: 2b 18 04 00 7c fc a4 8e   +...|...
88000068: 23 10 03 00 ff 00 46 30   #.....F0
88000070: 21 28 e0 02 23 1a 00 0e   !(..#...
88000078: 21 38 a0 03 52 00 40 14   !8..R.@.
88000080: 21 80 40 00 00 00 a5 8f   !.@.....
88000088: 01 88 1e 3c 2b 80 05 00   ...<+...
88000090: 42 00 00 12 d8 d9 c6 27   B.....'
88000098: 7c fc ac 8e 07 00 2d 26   |.....-&
880000a0: c2 58 0d 00 15 00 8a 91   .X.....
880000a8: c0 88 0b 00 01 00 46 32   .....F2
880000b0: 80 48 0a 00 21 80 a9 00   .H...!...
880000b8: 02 00 c0 14 21 18 20 02   ....!. .
Analog U/D : Addr. -/+
Triangle : Enter address      | Circle : Section change
Cross   : Disasm(Addr.)      | Square : Change viewmode
Select  : Scrollmode         | Start  : Dump section \0ead0e00

```

Your options at this screen is:

- ▷ **START:** Will let you dump the active section to memory stick in the directory PSPINSIDE\DUMPS
- ▷ **SELECT:** Select will allow you to control the browsing controls, slow/fast and if you want to be able to advance in chunks with analog left/right.
- ▷ **CROSS:** Will switch between memory/disassembly view of the current memory address.
- ▷ **CIRCLE:** Will switch between several preprogrammed sections, in the future we might put this into a configuration file.
- ▷ **SQUARE:** Will change the view modes: normal hex view, byte view, short word view, long word view, pure ASCII.

- ▷ TRIANGLE: Will bring up a virtual hex keyboard, allowing you to quickly go any address.

```
+-----+-----+
|          | [0x00000000          ] |
|          |-----+
| 01234 | Move analog Stick |
| f  5 | Cross   : Forward  |
| e + 6 | Circle  : Back    |
| d  7 | Square  : Return   |
| cba98 | Select  : Return!  |
|          | Triangle : Cancel |
+-----+-----+
```

- ▷ TRIANGLE: Cancel
- ▷ SQUARE: enter address with range checking
- ▷ SELECT: enter address without range checking

### 4.3 Disassembler Menu

Pressing Trigger-R will advance us to the next menu, this is the Disassembler menu, it supports custom Allegrex and VFPU opcodes.

```

-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+----->
System   |Memory   |Disasm   |Register |Syscalls |IRQs
-----+-----+-----+-----+-----+-----+-----
[nX] RAM uncached(24MB):00080bfc
08880bfc: f8 e5 04 25 addiu    $a0, $t0, 0xe5f8
08880c00: 21 28 00 00 addu     $a1, $zero, $zero
08880c04: a6 1b 22 0e jal      0x08886e98
08880c08: 21 30 00 00 addu     $a2, $zero, $zero
08880c0c: 89 08 03 3c lui      $v1, 0x0889
08880c10: 48 20 65 24 addiu    $a1, $v1, 0x2048
08880c14: 01 00 04 24 addiu    $a0, $zero, 0x0001
08880c18: 14 80 82 af sw       $v0, 0x8014($gp)
08880c1c: 4c 02 a0 af sw       $zero, 0x24c($sp)
08880c20: 54 02 a0 af sw       $zero, 0x254($sp)
08880c24: 58 02 a0 af sw       $zero, 0x258($sp)
08880c28: 60 02 a0 af sw       $zero, 0x260($sp)
08880c2c: 64 02 a0 af sw       $zero, 0x264($sp)
08880c30: 84 01 22 0e jal      [sceUmdActivate]
08880c34: 68 02 a0 af sw       $zero, 0x268($sp)
08880c38: 68 1d 22 0e jal      0x088875a0
08880c3c: 00 00 00 00 nop
08880c40: 8c 08 0a 3c lui      $t2, 0x088c
08880c44: 40 66 50 25 addiu    $s0, $t2, 0x6640
08880c48: 01 00 04 24 addiu    $a0, $zero, 0x0001
08880c4c: 21 28 00 02 addu     $a1, $s0, $zero
08880c50: e8 03 06 24 addiu    $a2, $zero, 0x03e8
08880c54: 06 01 22 0e jal      [sceKernelGetThreadmanIdLi]
08880c58: 21 38 00 00 addu     $a3, $zero, $zero
Analog U/D : Addr. -/+
Triangle : Enter address | Circle : Section change
Cross : Memory(Addr.)   | Square : More functions
Select : Scrollmode      | Start : Dump section \0ead0e00

```

Your options at this screen is:

- ▷ **START:** Will let you dump the disassembly of the active section to memory stick in the directory PSPINSIDE\DUMPS
- ▷ **SELECT:** Select will allow you to control the browsing controls, slow/fast and if you want to be able to advance in chunks with analog left/right.
- ▷ **CROSS:** Will switch between memory/disassembly view of the current memory address.
- ▷ **CIRCLE:** Will switch between several preprogrammed sections, in the future we might put this into a configuration file.
- ▷ **SQUARE:** Will bring up a sub-menu allowing you to Call or Patch the selected address/instruction.

```

+-----+
| Cross  | Call 0x88000000 |
+-----+
| Circle | NOP @ 0x88000000 |
+-----+
| Select | jr $ra @ 0x88000000 |
+-----+
| Triangle | <long> @ 0x88000000 |
+-----+
| Square | Return |
+-----+

```

- ▷ TRIANGLE: Will bring up a virtual hex keyboard, allowing you to quickly go any address.

```

+-----+-----+
|      | [0x088000cc ] |
+-----+-----+
| 01234 | Move analog Stick |
| f 5   | Cross : Forward |
| e + 6 | Circle : Back |
| d 7   | Square : Return |
| cba98 | Select : Return! |
|      | Triangle : Cancel |
+-----+-----+

```

- ▷ TRIANGLE: Cancel
- ▷ SQUARE: enter address with range checking
- ▷ SELECT: enter address without range checking

#### 4.4 Register Menu

```

-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+-----+----->
System   |Memory   |Disasm   |Register |Syscalls |IRQs
-----+-----+-----+-----+-----+-----+-----+----->
                                COP0 (mfc/mtc)
00: 00000000 00000000000000000000000000000000
01: 00000000 00000000000000000000000000000000
02: 00000000 00000000000000000000000000000000
03: 00000000 00000000000000000000000000000000
04: 00000000 00000000000000000000000000000000
05: 00000000 00000000000000000000000000000000
06: 00000000 00000000000000000000000000000000
07: 00000000 00000000000000000000000000000000
08: 00000000 00000000000000000000000000000000 BadVAddr
09: 00000000 00000000000000000000000000000000 Count
10: 00000000 00000000000000000000000000000000
11: 00000000 00000000000000000000000000000000 Compare
12: 00000000 00000000000000000000000000000000 Status
13: 00000000 00000000000000000000000000000000 Cause
14: 00000000 00000000000000000000000000000000 EPC
15: 00000000 00000000000000000000000000000000 PRId
16: 00000000 00000000000000000000000000000000 Config
17: 00000000 00000000000000000000000000000000
18: 00000000 00000000000000000000000000000000
19: 00000000 00000000000000000000000000000000
20: 00000000 00000000000000000000000000000000
21: 00000000 00000000000000000000000000000000 SC-code< <2
22: 00000000 00000000000000000000000000000000
23: 00000000 00000000000000000000000000000000
Analog U/D : Register -/+ | Select: Scrollmode
Cross : Disasm(Addr.)      | Circle: Change regbank

                                \0ead0e00

```

- ▷ DPad or Analog Up/Down to scroll
- ▷ CROSS to disassemble at Address
- ▷ CIRCLE to switch through Register Banks

## 4.5 Syscalls Menu

```

-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+----->
System   |Memory   |Disasm   |Register |Syscalls |IRQs
-----+-----+-----+-----+-----+-----+----->

0x2000 sceKernelRegisterSubIntrHandler
0x2001 sceKernelReleaseSubIntrHandler
0x2002 sceKernelEnableSubIntr
0x2003 sceKernelDisableSubIntr
0x2004 sceKernelSuspendSubIntr
0x2005 sceKernelResumeSubIntr
0x2006 sceKernelIsSubInterruptOccurred
0x2007 QueryIntrHandlerInfo
0x2008 sceKernelRegisterUserSpaceIntrStack
0x2009 _sceKernelReturnFromCallback
0x200a sceKernelRegisterThreadEventHandler
0x200b sceKernelReleaseThreadEventHandler
0x200c sceKernelReferThreadEventHandlerStatus
0x200d sceKernelCreateCallback
0x200e sceKernelDeleteCallback
0x200f sceKernelNotifyCallback
0x2010 sceKernelCancelCallback
0x2011 sceKernelGetCallbackCount
0x2012 sceKernelCheckCallback
0x2013 sceKernelReferCallbackStatus

Function: sceKernelRegisterSubIntrHandler
NID:      0xa04a2b90
Syscall:   0x2000
Address:   0x8801fce0

\0ead0e00

```

▷ DPad or Analog Up/Down to scroll

▷ CROSS to disassemble at Address



## 4.6 IRQs Menu

```

-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+----->
System   |Memory   |Disasm   |Register |Syscalls |IRQs
-----+-----+-----+-----+-----+-----+----->

00000000      0 (unknown)
00000000      0 (unknown)
00000000      0 (unknown)
00000000      0 (unknown)
8807b1ee      2770 GPIO
00000000      0 ATA_ATAPI
880f9c26      0 UmdMan
8813cf26      3558 MScm0
00000000      0 Wlan
00000000      0 (unknown)
880c2c76      0 Audio
00000000      0 (unknown)
8807d74e      49 I2C
00000000      0 (unknown)
880e7aea      0 SIRCS_IrDA
8804a4e2      0 Systimer0
8804a4e2      0 Systimer1
8804a4e2      0 Systimer2
8804a4e2      0 Systimer3
88037afa      280 Thread0

Interrupt '(unknown)'          Calls 0
Handler 0 Entry 00000000 Common 00000000 GP 00000000
Intrcode 0 SubCount 0 IntrLevel 0 Enabled 0
field_1c 00000000 totalclk_lo 00000000 totalclk_hi 00000000
Minclock_lo FFFFFFFF Minclock_hi FFFFFFFF
maxclock_lo 00000000, maxclock_hi 00000000

                                /0ead0e00

```

▷ DPad or Analog Up/Down to scroll

▷ CROSS to disassemble at Address

## 4.7 Videoram tab

```
-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+
<Videoram |Patch!   |Console  |Profiler |Config  |About
-----+-----+-----+-----+-----+-----+

04000000 00000000                                     \0ead0e00
```

▷ DPad or Analog to scroll around

## 4.8 Patch Menu

```
-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----  
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit  
-----+-----+-----+-----+-----+-----+  
<Videoram |Patch!   |Console  |Profiler |Config  |About  
-----+-----+-----+-----+-----+-----+
```

Not yet !

/0ead0e00

## 4.9 Console Tab

PSPInside will redirect any stdout and kernel debug messages to its internal console aswell as the serial output.

```
-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----  
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
```

```
-----+-----+-----+-----+-----+-----+  
<Videoram |Patch!   |Console |Profiler |Config  |About  
-----+-----+-----+-----+-----+-----+
```

```
stdio.c: stdoutReopen: id=0x00000001  
stdio.c: stdoutReopen: id=0x00000002  
PSPInside: enabled Kernel Messages  
PSPInside: enabled stdout
```

```
/0ead0e00
```

### 4.10 Profiler Tab

```
-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
```

```
-----+-----+-----+-----+-----+-----+
<Videoram |Patch! |Console |Profiler |Config |About
-----+-----+-----+-----+-----+-----+

```

```

enable          :          0
systemck        :          0 [cycles]
cpu ck          :          0 [cycles]
stall           :          0 [cycles]
+(internal)     :          0 [cycles]
+--(memory)     :          0 [cycles]
+----(COPz)     :          0 [cycles]
+----(VFPU)     :          0 [cycles]
sleep           :          0 [cycles]
bus access      :          0 [cycles]
uncached load   :          0 [times]
uncached store  :          0 [times]
cached load     :          0 [times]
cached store    :          0 [times]
I cache miss    :          0 [times]
D cache miss    :          0 [times]
D cache wb      :          0 [times]
COP0 inst       :          0 [inst]
FPU inst        :          0 [inst]
VFPU inst       :          0 [inst]
local bus       :          0 [cycles]

```

```
/0ead0e00
```

### 4.11 Config Tab

```
-----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >-----  
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit  
-----+-----+-----+-----+-----+-----+-----  
<Videoram |Patch!   |Console  |Profiler |Config   |About  
-----+-----+-----+-----+-----+-----+-----
```

Not yet !

/0ead0e00

## 4.12 About Tab

```

----< PSPInside V0.9q (c) 2005 by /-/itmen Productions >----
Trigger L/R : Tab -/+ | TrigL/R+Home : Quit
-----+-----+-----+-----+-----+-----+
<Videoram |Patch!   |Console  |Profiler |Config   |About
-----+-----+-----+-----+-----+-----+
Build on: Oct 27 2005 for Firmware v1.5 (detected v1.5)
working in directory: ms0:/PSP/GAME/KP/

skywalker@psp: cat AboutPSPInside.txt

>>> PSPInside - THE multipurpose PSP information toolbox

- Memory / disassembly viewer
- Module/thread list with alot of manipulation functions
- COP0 register viewer, important registers named
- Patchengine
- Fileselector for loading modules and binaries
- countless more ...

Load and start modules, disassemble them, dump the disasm
to your memstick, stop threads or just do whatever you like
or need with the software-parts running inside your PSP ...
All possible buttons are commented, some of them pop up
menues with even more commented functions ... its on you
to find every function it contains ... and there are alot :)

Many thanks to groepaz, jihad and xor37h of the Hitmen crew
for their great help on this project !

Special greetings to : maniac, eas, squid, hamstiglue, kk

- Visit us at www.hitmen-console.org -
                                     |0ead0e00

```

## 5 Exception Handler

```
Exception - Bus Error (data)
EPC       - 889049B0
Cause     - 0000001C
Status    - 20008603
BadVAddr  - 00000000
zr:00000000 at:00000000 v0:00000000 v1:00000000
a0:00000000 a1:00000000 a2:00000000 a3:00000000
t0:00000000 t1:00000000 t2:00000000 t3:00000000
t4:00000000 t5:00000000 t6:00000000 t7:00000000
s0:00000000 s1:00000000 s2:00000000 s3:00000000
s4:00000000 s5:00000000 s6:00000000 s7:00000000
t8:00000000 t9:00000000 k0:00000000 k1:00000000
gp:00000000 sp:00000000 fp:00000000 ra:00000000
```

```
Press X to reload
      O to restart
      ^ to return to VSH
```

- ▷ CROSS: reload and restart using LoadExec
- ▷ CIRCLE: restart using direct jump to main()
- ▷ TRIANGLE: quit PSPInside and go back to VSH



## 6 Serial Console

If you connect a suitable serial cable to the remote/headphone port you will be able to see kernel debug messages and stdout messages on a connected terminal.

### 6.1 Parameters

- ▷ 115200 Baud
- ▷ 8 Databits
- ▷ No Parity
- ▷ 1 Stopbit
- ▷ **no** Hardware Handshake (**no** RTS/CTS, **no** DTR)
- ▷ **no** Software Handshake (**no** XON/XOFF)

## 7 Patch Engine

patches are contained in the file `PATCHES.TXT`. the syntax of this file is defined like this:

- ▷ each line that starts with a `:` is an instruction
- ▷ each line starting with a `#` is a comment

### 7.1 Global Patches

global patches are executed immediatly when PSPInside is started, regardless of where they appear in the Patch File

#### 7.1.1 absolute memory writes

Syntax	Symbolic Meaning	Function
<code>:poba xxxxxxxx yy</code>	<code>poke.b \$xxxxxxx, \$yy</code>	Poke once absolute 8 bit
<code>:powa xxxxxxxx yyyy</code>	<code>poke.w \$xxxxxxx, \$yyyy</code>	Poke once absolute 16 bit
<code>:pola xxxxxxxx yyyyyyyy</code>	<code>poke.l \$xxxxxxx, \$yyyyyyy</code>	Poke once absolute 32 bit
<code>:paba xxxxxxxx yy</code>	<code>poke.b \$xxxxxxx, \$yy</code>	Poke always (every frame) absolute 8 bit
<code>:pawa xxxxxxxx yyyy</code>	<code>poke.w \$xxxxxxx, \$yyyy</code>	Poke always (every frame) absolute 16 bit
<code>:pala xxxxxxxx yyyyyyyy</code>	<code>poke.l \$xxxxxxx, \$yyyyyyy</code>	Poke always (every frame) absolute 32 bit

### 7.2 Module Patches

modules patches are applied after a specific module is loaded. each module patch block starts with a `mod` command

Syntax	Function
<code>:modX &lt;Module-Name&gt;</code>	Start Patch Block for Section X of specified Module
<code>:mode &lt;Module-Name&gt;</code>	Start Patch Block for the Section that contains the Entrypoint of specified Module

#### 7.2.1 relative memory writes

Syntax	Symbolic Meaning	Function
<code>:pobr xxxxxxxx yy</code>	<code>poke.b MODBASE+\$xxxxxxx, \$yy</code>	Poke once relative 8 bit
<code>:powr xxxxxxxx yyyy</code>	<code>poke.w MODBASE+\$xxxxxxx, \$yyyy</code>	Poke once relative 16 bit
<code>:polr xxxxxxxx yyyyyyyy</code>	<code>poke.l MODBASE+\$xxxxxxx, \$yyyyyyy</code>	Poke once relative 32 bit
<code>:pabr xxxxxxxx yy</code>	<code>poke.b MODBASE+\$xxxxxxx, \$yy</code>	Poke always (every frame) relative 8 bit
<code>:pawr xxxxxxxx yyyy</code>	<code>poke.w MODBASE+\$xxxxxxx, \$yyyy</code>	Poke always (every frame) relative 16 bit
<code>:palr xxxxxxxx yyyyyyyy</code>	<code>poke.l MODBASE+\$xxxxxxx, \$yyyyyyy</code>	Poke always (every frame) relative 32 bit

#### 7.2.2 search and replace

Syntax	Function
<code>:sobr yy zz</code>	Search yy and replace with zz over module segment , once (8bit)
<code>:sowr yyyy zzzz</code>	Search yy and replace with zz over module segment , once (16bit)
<code>:solr yyyyyyyy zzzzzzzz</code>	Search yy and replace with zz over module segment , once (32bit)
<code>:sabr yy zz</code>	Search yy and replace with zz over module segment , always (8bit)
<code>:sawr yyyy zzzz</code>	Search yy and replace with zz over module segment , always (16bit)
<code>:salr yyyyyyyy zzzzzzzz</code>	Search yy and replace with zz over module segment , always (32bit)
<code>:d2ms</code>	disc0: to msfat: replacer over whole stated module segment.

### 7.2.3 load additional modules

Syntax	Function
:lxmd <module-path>	Loads and executes all Modules in a Directory once the stated 'mod0' module gets loaded
:lxmf <module-name>	Loads and executes a Module File once the stated 'mod0' module gets loaded

### 7.2.4 run module

Syntax	Function
:run0	Start stated module, keep PSPInside in front
:run1	Start stated module, put PSPInside to back (get to front with "note" button)

### 7.2.5 suspend mode

Syntax	Function
:sus0	suspend Thread after loading
:sus1	mark Thread for note-button suspending

## 7.3 Example Patches

```

mod0 Bomb
run0
lxmd fatms:/MYMODULES/
lxmf fatms:/PRX/SPECIAL.PRX

```

mod0 Bomb - this is the module which gives the offset for the "relative" pokes, when Module "Bomb" is now loaded, the offset for the following relative pokes will the start-address of the first segment of the module (mod1 - for the start-address of 2.segment, mode - for the entryptpoint as offset)

run0 - Run that module with PSPInside in front (run1 - Run that module with PSPInside to back)

lxmd fatms:/MYMODULES/ - if "bomb" gets loaded, the modules in this directory get loaded

lxmf fatms:/PRX/SPECIAL.PRX - if "bomb" gets loaded, this module gets loaded

## 8 Usage Examples

### 8.1 automatically suspend a thread

This shows how to automatically suspend a specific thread when PSPInside is switched to foreground using the note button

1. in `PATCHES.TXT` create a patch like this

```
:thd0 user_main  
:sus1
```

### 8.2 debug Game

This shows how to run PSPInside in the Background of Wipeout

1. in `PATCHES.TXT` create a patch for Wipeouts main thread

```
:mod0 WO_Game  
:run1
```

this will put PSPInside into background when the game is run

2. load PSPInside
3. press TRIANGLE
4. browse to `disc0:/PSP_GAME/SYSDIR/EBOOT.BIN` and press CROSS
5. press SQUARE (00000000)

now the game will start and you can switch between PSPInside/Wipeout with note button once the language selection shows up.

## 9 References

- ▷ U.S. Pat. 6,817,021 (Disk device and guide member)
- ▷ U.S. Pat. 6,345,747 (Strap Assembly)
- ▷ U.S. Pat. Application 20040266529 (Methods and systems for remote execution of game content and presentation on a wireless portable device) - PS3 to PSP connection
- ▷ Debug Information in 'Puzzle Bobble' (Error Codes, Kernel API Names etc...)
- ▷ WM8750 Datasheet

### 9.1 Sources

- ▷ <http://www.uspto.gov>
- ▷ <http://www.mips.com>
- ▷ <http://www.sdmi.org>
- ▷ <http://www.sony.com>
- ▷ <http://www.sony.net>
- ▷ <http://www.lik-sang.com/psp.html>
- ▷ <http://www.chipworks.com>
- ▷ <http://www.extremetech.com>
- ▷ <http://www.rsasecurity.com>
- ▷ <http://pinouts.ru>
- ▷ <http://www.edcheung.com/automa/sircs.htm>
- ▷ <http://www.hifi-remote.com/sony/>
- ▷ <http://www.ecma-international.org>

## 10 Credits

Skywalker	Main Program, Patch Engine
Xor37h	Disassembler
Jihad	Testing, Hardware Hacking
Groepaz	Docs, additional Code

### 10.1 Contributions

Tyranid	fixes for v1.0 Firmware reported additional allegrex opcodes foundation of SIO and IRQ related lib functions
shazz	hint on fixing loading of plaintext prxes, provided such prx for testing
sherpya	python script to make disassembly output nicer
Fanjita Ovem	testing on Firmware 2.0 2.0 Firmware Syscall research
Vampire	2.0 Firmware Syscall research
Wil	testing on Firmware 2.0