

PSX-Register

hacked by bITmASTER
eMail: bITmASTER@bigfoot.com

last update: 09.07.99

VAG-Format

VAG-Sample-Group

Eine Sample-Group besteht aus 16 Bytes.

```
typedef struct {  
    unsigned char pack_info;  
    unsigned char flags;  
    unsigned char packed[14];  
}
```

pack_info:
höherwertiges Nibble: Predictor-Nummer
niederwertiges Nibble: Shiftfactor

flags:
0x07: Sample-Ende
0x02: Samples gehören zu Repeat-Teil
0x06: Repeat-Start

0x04: Repeat-Point
0x03: Sample-Ende, beginne neu bei Repeat-Start

Register Size:
short: 2 Bytes
int: 4 Bytes

SPU-Register

| | | | |
|------------|-----|----|---|
| 0x1f8010c0 | int | rw | SPU-DMA Memory-Address |
| 0x1f8010c4 | int | rw | b15...b0 b31...b16: DMA-Block-Size (n * 64 Bytes) |
| 0x1f8010c8 | int | rw | b24 = 1: Start DMA when transfer complete b24 switch to 0 |

| | | | |
|------------|-----|-----|---------------------------------|
| 0x1f8010f4 | int | rw | DMA Interrupt Control |
| | | | |
| | | b15 | DMA channel ?? interrupt enable |
| | | | |

| | | | |
|------------|-------|----|--|
| 0x1f801c00 | short | rw | Volume left Channel 0 b0...b13: Volume (0...0x3fff) b14: Sign |
| 0x1f801c02 | short | rw | Volume right Channel 0 b0...b13: Volume (0...0x3fff) b14: Sign |
| 0x1f801c04 | short | rw | Frequency |
| 0x1f801c06 | short | rw | Startaddress Sound = SPU-Memory / 8 |
| 0x1f801c08 | short | rw | b7-b14 Attack-Rate (0x7f = nie) b15: 1 Exponential |
| 0x1f801c0a | short | rw | ADSR2 |
| 0x1f801c0c | short | rw | ADSR-Volume (0x0000 - 0x7fff) |
| 0x1f801c0e | short | r | current repeat address |

... weitere 23 mal für die Channels 1...23

s: Anzahl der Samples für eine Periode
f: gewünschte Wiedergabefrequenz
t: Teilerfaktor

$$t = \frac{s \cdot f}{44100} \cdot 4096 \quad f = \frac{44100 \cdot t}{4096 \cdot s}$$

| | | | |
|------------|-------|----|--|
| 0x1f801d80 | short | rw | MasterVolume left (0x0000-0x3fff) |
| 0x1f801d82 | short | rw | MasterVolume right (0x0000-0x3fff) |
| 0x1f801d84 | short | rw | ReverbDepth left (0x0000-0x7fff) , b15=1: inverted |
| 0x1f801d86 | short | rw | ReverbDepth right (0x0000-0x7fff), b15=1: inverted |
| 0x1f801d88 | short | w | start sound play b0 = 1 start channel 0 b1 = 1 start channel 1 : : : : : b15 = 1 start channel 15 |
| 0x1f801d8a | short | w | start sound play b0 = 1 start channel 16 b1 = 1 start channel 17 : : : : : b7 = 1 start channel 23 |
| 0x1f801d8c | short | w | stop sound play b0 = 1 stop channel 0 b1 = 1 stop channel 1 : : : : : b15 = 1 stop channel 15 |
| 0x1f801d8e | short | w | stop sound play b0 = 1 stop channel 16 |

| | | | |
|------------|-------|----|---|
| | | | b1 = 1 stop channel 17 : : : : : b7 = 1 stop channel 23 |
| 0x1f801d90 | short | rw | set channel in FM-Modus die Modulation kommt von Channel n-1 b0 = 1 channel 0 b1 = 1 channel 1 : : : : : b15 = 1 channel 15 |
| 0x1f801d92 | short | rw | set channel in FM-Modus die Modulation kommt von Channel n-1 b0 = 1 channel 0 b1 = 1 channel 1 : : : : : b15 = 1 channel 15 |
| 0x1f801d94 | short | rw | use channel as noise generator b0 = 1 start channel 0 b1 = 1 start channel 1 : : : : : b15 = 1 start channel 15 |
| 0x1f801d96 | short | rw | use channel as noise generator b0 = 1 start channel 16 b1 = 1 start channel 17 : : : : : b7 = 1 start channel 23 |
| 0x1f801d98 | short | rw | set channel to reverb modus (after sample end, the reverb modus automatic switch off !!) b0 = 1 start channel 0 b1 = 1 start channel 1 : : : : : b15 = 1 start channel 15 |
| 0x1f801d9a | short | rw | set channel to reverb modus b0 = 1 start channel 16 b1 = 1 start channel 17 : : : : : b7 = 1 start channel 23 |
| 0x1f801d9c | short | r | channel active b0 = 0 channel 0 active b1 = 0 channel 1 active : : : : : b15 = 0 channel 15 active |
| 0x1f801d9e | short | r | channel active b0 = 0 channel 16 active b1 = 0 channel 17 active : : : : : b7 = 0 channel 23 active |
| 0x1f801da0 | | | not used |
| 0x1f801da2 | | | start address Reverb Buffer |
| 0x1f801da4 | short | rw | function unknow |
| 0x1f801da6 | short | rw | SPU-Memory-Address |
| 0x1f801da8 | short | rw | Data to SPU |
| 0x1f801daa | short | rw | Spu-Control1 (0xc000) b14=0 DA-Converter mute b13...b8: Noise-Frequency b7 0 Reverb off 1 Reverb on b6: SpuIQR b5 b4 0 0 --- 0 1 write SPU non-DMA 1 0 write SPU with DMA 1 1 read SPU with DMA b3:EXT Reverb b2:CD Reverb b1:enable extern digital audio input b0:enable CD audio |
| 0x1f801dac | short | rw | Spu-Control2 (4) |

| | | | |
|------------|-------|----|---|
| 0x1f801dae | short | r | Spu-Status b7: 1, SPU is ready for DMA-Transfer b10: wrdy b11: current decoded page |
| 0x1f801db0 | short | rw | CD-Audio-Volume left |
| 0x1f801db2 | short | rw | CD-Audio-Volume right |
| 0x1f801db4 | short | rw | extern Volume left |
| 0x1f801db6 | short | rw | extern Volume right |
| 0x1f801db8 | short | r | ??? |
| 0x1f801dba | short | r | ??? |
| 0x1f801dbc | | | |
| 0x1f801dbe | | | |
| 0x1f801dc0 | | | SPU-Effects |
| 0x1f801dc2 | | | |
| 0x1f801dc4 | | | Filter für verzögertes Signal Tiefpaß 0x0000: niedrigste Grenzfrequenz 0x7fff: höchste Grenzfrequenz, keine Filterwirkung |
| 0x1f801dc6 | | | Amplitude verzögertes Signal 0...32767 -32768...0 (invertiert) |
| 0x1f801dc8 | | | |
| 0x1f801dca | | | |
| 0x1f801dcc | | | |
| 0x1f801dce | | | |
| 0x1f801dd0 | | | |
| 0x1f801dd2 | | | |
| 0x1f801dd4 | | | |
| 0x1f801dd6 | | | |
| 0x1f801dd8 | | | Delay, in 8er Steps (???) |
| 0x1f801dda | | | |
| 0x1f801ddc | | | |
| 0x1f801dde | | | |
| 0x1f801de0 | | | |
| 0x1f801de2 | | | |
| 0x1f801de4 | | | |
| 0x1f801de6 | | | |
| 0x1f801de8 | | | |
| 0x1f801dea | | | |
| 0x1f801dec | | | |
| 0x1f801dee | | | |
| 0x1f801df0 | | | |
| 0x1f801df2 | | | |
| 0x1f801df4 | | | |
| 0x1f801df6 | | | |
| 0x1f801df8 | | | offset left ??? |
| 0x1f801dfa | | | |
| 0x1f801dfc | short | | signed short volume left |
| 0x1f801dfe | short | | signed short volume right |

Timer

Timer-Frequenz: 33.872 MHz ??
33.868 MHz = 44100 * 256 * 3

Timer0

| | | | |
|------------|-------|----|-----------------------------------|
| 0x1f801100 | short | r | Up-Counter, aktueller Zählerstand |
| 0x1f801104 | short | rw | Timer-Mode-Register |
| 0x1f801108 | short | rw | Zähler Endstand |

Timer-Mode-Register

| | 0 | 1 |
|----|-----------------------------|--------------------------------|
| b0 | Timer running | Timer Stop |
| b1 | | |
| b2 | | |
| b3 | Timer zählt von 0 bis 65535 | Timer zählt von 0 bis Endstand |
| b4 | | Interrupt enable |
| b5 | | |
| b6 | | Interrupt enable |
| b7 | | |
| b8 | Vorteiler: ? | Vorteiler: ? |

Timer1

| | | | |
|------------|-------|----|-----------------------------------|
| 0x1f801110 | short | r | Up-Counter, aktueller Zählerstand |
| 0x1f801114 | short | rw | Timer-Mode-Register |
| 0x1f801118 | short | rw | Zähler Endstand |

Timer-Mode-Register

| | 0 | 1 |
|----|-----------------------------|--------------------------------|
| b0 | Timer running | Timer Stop |
| b1 | | |
| b2 | | |
| b3 | Timer zählt von 0 bis 65535 | Timer zählt von 0 bis Endstand |
| b4 | | Interrupt enable |
| b5 | | |
| b6 | | Interrupt enable |
| b7 | | |
| b8 | Vorteiler: ? | Timer wird mit HBL getaktet |

Timer2

| | | | |
|------------|-------|----|-----------------------------------|
| 0x1f801120 | short | r | Up-Counter, aktueller Zählerstand |
| 0x1f801124 | short | rw | Timer-Mode-Register |
| 0x1f801128 | short | rw | Zähler Endstand |

Timer-Mode-Register

| | 0 | 1 |
|----|-----------------------------|--------------------------------|
| b0 | Timer running | Timer Stop |
| b1 | | |
| b2 | | |
| b3 | Timer zählt von 0 bis 65535 | Timer zählt von 0 bis Endstand |
| b4 | | Interrupt enable |
| b5 | | |
| b6 | | Interrupt enable |
| b7 | | |
| b8 | Vorteiler: ? | Vorteiler: ? |

Interrupt-Register

| | | | |
|------------|-----|----|--------------------------------|
| 0x1f801070 | int | rw | Interrupt Acknowledge Register |
| 0x1f801074 | int | rw | Interrupt Mask Register |

| | l=on | HW-Desc |
|-----|-------------------------------|------------|
| b0 | Interrupt für VBL erlauben | 0xf0000001 |
| b1 | GPU | 0xf0000002 |
| b2 | CD-ROM | 0xf0000003 |
| b3 | DMA-Controller | 0xf0000004 |
| b4 | Interrupt für Timer0 erlauben | 0xf0000005 |
| b5 | Interrupt für Timer1 erlauben | 0xf0000006 |
| b6 | Interrupt für Timer2 erlauben | 0xf0000007 |
| b7 | Controller | 0xf0000008 |
| b8 | SIO | 0xf0000009 |
| b9 | SPU | 0xf000000a |
| b10 | PIO | 0xf000000b |

For acknowledge an interrupt you must write ~(int-bit) to the interrupt acknowledge register.
i.e. VBL

```
li    t0,0x1f800000
li    t1,~1          # ~1 = 0xfffffffffe
sw    t1,0x1070(t0)
```

SIO

The SIO has a 8 Byte FIFO. The SIO can read (write not tested) 1, 2, 4 or 8 bytes at once. For use the multiple byte feature you must enable the RX interrupt. This interrupt occurs after the sio has get 1, 2, 4 or 8 bytes. The interrupt service routine must get the amount of bytes in following manner:

| Buffersize | |
|------------|--|
| 1 | li t0,0x1f800000 lb t1,0x1050(t0) |
| 2 | li t0,0x1f800000 lhu t1,0x1050(t0) |
| 4 | li t0,0x1f800000 lw t1,0x1050(t0) |
| 8 | li t0,0x1f800000 lw t1,0x1050(t0) # first 4 bytes lw t1,0x1050(t0) # secound 4 bytes |

This multiple byte feature avoids some interrupt load !

| | | | | | |
|------------|-------|----|--|---------------------------|--------------------|
| 0x1f801050 | int | rw | SIO-Data (8, 16 or 32 Bits) | | |
| 0x1f801054 | short | r | SIO Status | | |
| | | | b0 | ready for write byte ->TX | |
| | | | b1 | ready for read byte RX-> | |
| | | | b2 | ? | |
| | | | b3 | parity error | |
| | | | b4 | overrun error | |
| | | | b5 | frame error | |
| | | | b6 | 0 | |
| | | | b7 | DSR | |
| | | | b8 | CTS | |
| | | | b9 | interrupt | |
| 0x1f801058 | short | rw | SIO Mode | | |
| | | | b1 | b0 | Baudrate Prescaler |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 16 |
| | | | 1 | 1 | 64 |
| | | | b3 | b2 | char length |
| | | | 0 | 0 | 5 Bit |
| | | | 0 | 1 | 6 Bit |
| | | | 1 | 0 | 7 Bit |
| | | | 1 | 1 | 8 Bit |
| | | | b4 | parity | |
| | | | 0 | off | |
| | | | 1 | on | |
| | | | b5 | parity mode | |
| | | | 0 | even | |
| | | | 1 | odd | |
| | | | b7 | b6 | stop bit length |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1,5 |
| | | | 1 | 1 | 2 |
| 0x1f80105a | short | rw | SIO Control | | |
| | | | b0 | TX enable | |
| | | | b1 | DTR, OpenDrain-Output | |
| | | | b2 | RX enable | |
| | | | b3 | ? | |
| | | | b4 | Interrupt Flag | |
| | | | b5 | RTS, OpenDrain-Output | |
| | | | b6 | SIO Reset | |
| | | | b7 | ? | |
| | | | b9 | b8 | Buffersize |
| | | | 0 | 0 | 1 Byte |
| | | | 0 | 1 | 2 Bytes (short) |
| | | | 1 | 0 | 4 Bytes (int) |
| | | | 1 | 1 | 8 Bytes (two int) |
| | | | b10 | TX Interrupt enable | |
| | | | b11 | RX Interrupt enable | |
| | | | b12 | DSR Interrupt enable | |
| 0x1f80105e | short | rw | SIO Baudrate = 2,1168 Mhz / wanted baudrate | | |

DMA-Control

| | | | | |
|------------|-----|----|-----------------------|-------------------------------|
| 0x1f8010f0 | int | rw | DMA Enable | |
| | | | b3 | enable DMA 0 (MDECin) |
| | | | b7 | enable DMA 1 (MDECout) |
| | | | b11 | enable DMA 2 (GPU) |
| | | | b15 | enable DMA 3 (CD-ROM) |
| | | | b19 | enable DMA 4 (SPU) |
| | | | b23 | enable DMA 5 (PIO) |
| | | | b27 | enable DMA 6 (OTC) |
| 0x1f8010f4 | int | rw | DMA Interrupt Control | |
| | | | b0 | fast DMA 0 Transfer ??? |
| | | | b1 | fast DMA 1 Transfer ??? |
| | | | b2 | fast DMA 2 Transfer ??? |
| | | | b3 | fast DMA 3 Transfer ??? |
| | | | b4 | fast DMA 4 Transfer ??? |
| | | | b5 | fast DMA 5 Transfer ??? |
| | | | b6 | fast DMA 6 Transfer ??? |
| | | | b7 | |
| | | | b8 | DMA 0 ??? |
| | | | b9 | DMA 1 ??? |
| | | | b10 | DMA 2 ??? |
| | | | b11 | DMA 3 ??? |
| | | | b12 | DMA 4 ??? |
| | | | b13 | DMA 5 ??? |
| | | | b14 | DMA 6 ??? |
| | | | b15 | |
| | | | b16 | enable DMA 0 Interrupt |
| | | | b17 | enable DMA 1 Interrupt |
| | | | b18 | enable DMA 2 Interrupt |
| | | | b19 | enable DMA 3 Interrupt |
| | | | b20 | enable DMA 4 Interrupt |
| | | | b21 | enable DMA 5 Interrupt |
| | | | b22 | enable DMA 6 Interrupt |
| | | | b23 | should be 1 |
| | | | b24 | acknowledge interrupt DMA 0 |
| | | | b25 | acknowledge interrupt DMA 1 |
| | | | b26 | acknowledge interrupt DMA 2 |
| | | | b27 | acknowledge interrupt DMA 3 |
| | | | b28 | acknowledge interrupt DMA 4 |
| | | | b29 | acknowledge interrupt DMA 5 |
| | | | b30 | acknowledge interrupt DMA 6 |
| | | | b31 | should be 1 for acknowledging |

Before the DMA interrupt service routine ends, you should acknowledge the DMA interrupt.

```
li    t0,0x1f800000
lw     t1,0x10f4(t0)
li     t2,0xffffffff      # mask
and    t1,t1,t2
li     t2,0x81000000      # i.e. DMA 0
or     t1,t1,t2
sw     t2,0x10f4(t0)
```

Don't forget to acknowledge the interrupt self:

```
li     t1,~4
sw     t1,0x1070(t0)
```

Memory-Variables

| | |
|-------|----------------------------|
| 0x120 | Zeiger auf Event-Table |
| 0x124 | Größe Event-Table in Bytes |