Hardware Hacking. Edycja polska

Autorzy: Joe Grand, Ryan Russell
Tłumaczenie: Radosław Meryk
ISBN: 83-7361-549-0
Tytuł oryginału: Hardware Hacking
Format: B5, stron: 424

Hacking sprzętowy rozwija się od dawna. Za jego prekursorów uznaje się założycieli firmy Hewlett-Packard, którzy rozpoczęli budowanie swojego imperium od prób wykorzystania znanych urządzeń do celów, do których nie były przewidywane. Dziś hakerem sprzętowym można nazwać każdego, kto samodzielnie próbuje zmodyfikować swój komputer, palmtopa lub inne urządzenie tak, aby wycisnąć z niego maksimum możliwości. Każdy hobbysta-elektronik może zostać hakerem sprzętowym, wiedząc, co wykorzystać i co zmodyfikować.

Książka „Hardware Hacking. Edycja polska" to podręcznik dla wszystkich, którzy chcą spróbować sił przy samodzielnym modyfikowaniu swojego sprzętu. Autorzy — osoby na co dzień zajmujące się projektowaniem urządzeń elektronicznych i oprogramowania — dzielą się swoją wiedzą dotyczącą hackingu sprzętowego. Zarówno początkujący hobbysta, jak i zaawansowany elektronik znajdą tu coś dla siebie.

- Kompletowanie niezbędnych narzędzi
- Tworzenie macierzy RAID z dyskiem FireWire
- Budowa zestawu kina domowego wykorzystującego komputer
- Modyfikowanie konsol Atari 2600 i 7200
- Przeróbki komputerów ośmiobitowych
- Hacking konsoli Playstation
- Modyfikowanie urządzeń dostępowych w sieciach bezprzewodowych
- Przeróbki telefonów komórkowych i palmtopów
- Przegląd najważniejszych informacji o systemach operacyjnych i programowaniu w języku C

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Rozdział 7.

Hacking konsoli Playstation 2

W tym rozdziale:

- Otwieranie konsoli Playstation 2
- Instalacja portu szeregowego
- Ładowanie kodu z karty pamięci
- Inne projekty: wykorzystanie dowolnych twardych dysków

Wprowadzenie

Z 60 milionami egzemplarzy sprzedanymi na całym świecie — według stanu na sierpień 2003 roku — konsola PlayStation 2 firmy Sony posiada największą liczbę użytkowników spośród wszystkich współczesnych konsol gier. Dość nieoczekiwanie, w porównaniu z konsolami Xbox firmy Microsoft oraz Dreamcast firmy Sega, dla konsoli PS2 istnieje stosunkowo mało modyfikacji sprzętowych i hobbystycznych projektów oprogramowania. Istnieje aktywna społeczność programistów tworzących oprogramowanie dla konsoli PS2, ale jej liczebności nie da się porównać z analogicznymi grupami dla konsol Xbox lub Dreamcast. Poza producentami zmodyfikowanych układów, znanych jako modchipy, przeróbkami konsol PS2 zajmuje się bardzo niewielu hakerów sprzętowych.

Jednym z powodów, dla których nie istnieje zbyt wiele modyfikacji konsoli PS/2 jest fakt, że duża grupa utalentowanych hakerów zajmujących się wsteczną inżynierią sprzętu konsoli PS2 to producenci modchipów. Ludzie ci chronią informacje, traktując je jak tajemnice handlowe i rzadko ujawniają szerszemu gronu odbiorców (zazwyczaj dostarczają tylko tyle informacji, ile potrzeba użytkownikowi do zainstalowania modchipa). Dodatkowo, chociaż w konsoli PS2 zastosowano kilka standardowych interfejsów, takich jak USB czy IEEE 1394, jej architektura wewnętrzna całkowicie różni się od standardowych — na przykład od architektury PC, na której opiera się konsola Xbox. Zlokalizowanie szyn danych oraz zidentyfikowanie sygnałów wyjściowych generowanych przez niestandardowe układy płyty konsoli PS2 wymaga znacznie więcej wysiłku.

Komercyjny hacking sprzętowy: modchipy

W przypadku wszystkich konsol handel pirackim oprogramowaniem jest doskonałym interesem. Chociaż producenci konsol stosują zabezpieczenia, które utrudniają przeciętnemu graczowi kopiowanie gier, hakerzy sprzętowi stosują wyspecjalizowane techniki mające na celu pokonanie tych mechanizmów. Techniki te obejmują wyszukiwanie systemowych szyn danych za pomocą analizatorów logicznych i zrzucanie obrazów systemu BIOS w celu opracowania obejść programowych. Modchipy, których jedynym niegdyś celem było umożliwienie wykorzystania nielegalnych kopii gier, stały się skomplikowanymi urządzeniami o znacznie szerszych możliwościach.

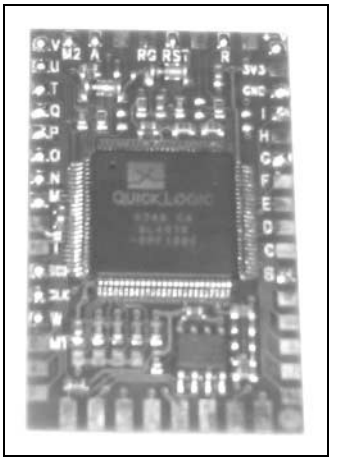
Często wykorzystuje się w nich specjalizowane eksploity i obejścia, mające na celu pokonanie określanego zabezpieczenia. Modchipy to niewielkie obwody drukowane, podłączane do różnych elementów na płycie głównej konsoli. Zazwyczaj podstawą tych układów są mikroukłady PIC lub PLD (*Programmable Logic Device* — programowalne urządzenia logiczne). Istnieją także nowoczesne modchipy, w których wykorzystuje się układy FPGA i Flash ROM, co umożliwia ich aktualizację, wprowadzanie poprawek i nowych funkcji. Zazwyczaj modchip wysyła sygnał do systemu zabezpieczeń konsoli i w ten sposób „oszukuje" ten mechanizm, który przyjmuje, że użytkownik użył legalnej płyty z grą. Za pomocą modchipów można również pokonać inne systemy zabezpieczeń, np. kody BIOS-u lub domyślny tryb wideo (PAL lub NTSC) karty graficznej konsoli.

W konsoli PS2 modchipy są stosowane głównie w celu pokonania sprzętowych mechanizmów uwierzytelniania dla dysków. Wykorzystuje się je również do aktualizowania BIOS-u, ściągania blokady Macrovision, uniemożliwiającej nagrywanie zawartości płyt DVD na magnetowid, pokonywania ograniczeń regionalnych dla płyt DVD oraz w celu modyfikacji systemu video wykorzystywanego w grach (np. aby umożliwić grę w gry przeznaczone dla systemu PAL w konsoli z systemem NTSC). Pierwsze modchipy były prostymi urządzeniami, zdolnymi do pokonywania mechanizmów uwierzytelniania dysków oraz wykonywania tzw. podmiany dysków (ang. *swap trick*). Sztuczka ta polega na zastosowaniu specjalnych dysków ładujących (tworzonych przez hakerów) z kodem zatrzymującym napęd DVD. Po zatrzymaniu napędu, użytkownik wymusza jego otwarcie, stosując specjalną modyfikację obudowy (tzw. *fliptop*) lub czasami za pomocą plastikowego noża lub płaskiego przedmiotu przypominającego kartę kredytową. Następnie umieszcza w napędzie skopiowaną grę PS2 i zamyka go. W ten sposób, dzięki uniemożliwieniu wykrycia wymuszonego otwarcia i zamknięcia napędu, następuje pokonanie systemu uwierzytelniania.

Największą wadą fizycznej podmiany dysków jest szybsze zużywanie się napędów DVD, spowodowane mechanicznym otwieraniem i zamykaniem napędu. W nowoczesnych modchipach zamiast destrukcyjnej podmiany zastosowano sprzętowe uwierzytelnianie dysków gier. W niektórych wykorzystuje się układy FPGA oraz pamięci Flash, pozwalające użytkownikom na wprowadzanie aktualizacji i poprawek błędów poprzez umieszczenie dysku w napędzie. Częścią tych układów jest specjalny mechanizm, wykonujący program zapisany na karcie pamięci konsoli PS2 w momencie włączenia

zasilania. Zazwyczaj program ten jest innym programem ładującym, tym razem z graficznym interfejsem użytkownika, umożliwiającym ładowanie programów i narzędzi z płyty CD na karcie pamięci lub na dysku twardym. Na rysunku 7.1 zaprezentowano modchip LisaZero (dostępny wyłącznie dla systemu PAL), w którym wykorzystano układ PLD QuickLogic ze zmodyfikowanym kodem ładującym.

Rysunek 7.1. *Modchip LisaZero*



W niektórych podrozdziałach niniejszego rozdziału zamieszczono przykłady kodu źródłowego. W większości przykładów wykorzystano bibliotekę open source — PS2Lib — dostępną pod adresem *http://ps2dev.sourceforge.net/ps2lib.html*.

Oto kilka uwag na temat konwencji zastosowanych w kodzie:

- W pliku nagłówkowym *tamtypes.h* zdefiniowano podstawowe typy wykorzystywane w bibliotece PS2Lib. Litera `u`, po której występuje numer, oznacza typ bez znaku o określonej liczbie bitów; litera `s` z numerem określa natomiast typ ze znakiem. Na przykład, `s8` oznacza ośmiobitową liczbę całkowitą ze znakiem, natomiast `u32` — trzydziestodwubitową liczbę całkowitą bez znaku.
- W pliku *tamtypes.h* zdefiniowano makra umożliwiające wygodny dostęp do rejestrów sprzętowych. Przypominają one makra `inb()` oraz `outb()`, spotykane w programowaniu niskopoziomowym komputera PC. Makro `_lw()` jest synonimem instrukcji MIPS `lw`, zwracającej trzydziestodwubitową liczbę odczytaną spod określonego adresu (adres jest określony jako 32-bitowa liczba całkowita bez znaku). Analogicznie, makro `_sw()` zapisuje trzydziestodwubitową liczbę pod wskazanym adresem. Każde z tych makr reprezentuje odpowiednią instrukcję MIPS. Tak więc istnieją makra `_lb()`, `_sb()`, `_lh()`/`_sh()` oraz `_ld()`/`_sd()`, służące do odczytywania wartości liczb — odpowiednio — ośmio-, szesnasto- i sześćdziesięcioczterobitowych.

Inżynierowie i hakerzy zajmujący się produkcją modchipów poświęcają wiele czasu i pieniędzy na prace nad wsteczną inżynierią sprzętu i testowaniem. Uzyskane przez nich informacje prawie nigdy nie są ujawniane szerszemu gronu odbiorców. Głównym powodem jest obawa przed konkurencyjnymi producentami modchipów. Czasami tworzą

oni swoje rozwiązanie na podstawie wstecznej inżynierii produktu konkurencji lub stosują tę samą technikę. Chociaż niektóre informacje odkrywane w wyniku analizy konsoli są przydatne tylko dla użytkowników chcących pokonać zabezpieczenia, większość przyda się każdemu, kto chce tworzyć własny sprzęt lub oprogramowanie dla konsoli PS2. Można tu wspomnieć choćby o układzie styków i sygnałów procesorów, szyn, portów rozszerzeń i o systemie BIOS.

Ukrywanie takich informacji utrudnia pracę hakerom sprzętowym i hobbystycznym społecznościom programistów. Jak przekonamy się w dalszej części niniejszego rozdziału, posiadanie niektórych ukrytych informacji o sprzęcie umożliwia lepszą kontrolę nad systemem i uzyskanie dostępu do zaawansowanych metod debuggowania. Moją główną motywacją podczas tworzenia eksploita *Independence* (niezależność) — patrz podrozdział „Ładowanie kodu z karty pamięci" — było umożliwienie pisania oprogramowania na konsolę PS2 bez konieczności fizycznego jej modyfikowania.

Otwieranie konsoli PS2

W tym podrozdziale podano wskazówki umożliwiające zidentyfikowanie wersji płyty głównej konsoli PS2 oraz opisano czynności umożliwiające rozmontowanie konsoli.

Wersje płyty głównej

W czasie prowadzenia prac nad niniejszą książką istniało 11 głównych numerów wersji konsoli PS2 oraz ponad dziesięć wersji BIOS-u. Wersje płyty głównej zwykle są oznaczane literą V i numerem, na przykład V7. Numer wersji płyty zwykle określa się jako numer wersji PS2. Numery rozpoczynają się od V0 — pierwszej wersji konsoli PS2 wyprodukowanej w Japonii.

Firma Sony aktualizuje wersje konsoli PS2 z kilku powodów:

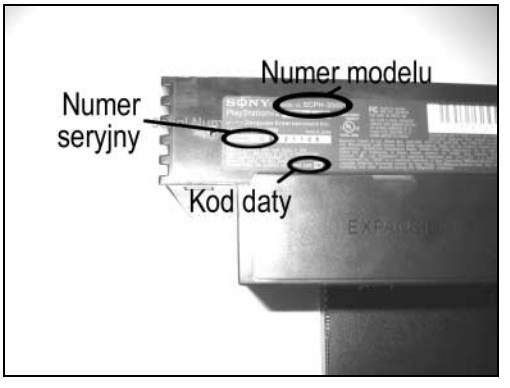
- aby poprawić błędy w sprzęcie i oprogramowaniu;
- aby umieścić oddzielne urządzenia w oddzielnych układach — po to, by zmniejszyć koszty produkcji;
- aby zastosować nowe mechanizmy zabezpieczeń.

Problem z głównymi wersjami płyty głównej polega na tym, że wraz ze zmianą wersji zmienia się fizyczne rozmieszczenie elementów na płycie. Oznacza to, że instrukcje wyszukania określonego komponentu lub punktu pomiarowego dla płyty V1 są inne niż dla V7. Układ płyt niektórych wersji (np. V5 i V6) jest zbliżony, a zatem, w takim przypadku instrukcje będą takie same. Modyfikacje opisane w niniejszym rozdziale zostały wykonane dla konsoli PS2 w wersji V4. Posiadacze innych wersji konsoli będą musieli dostosować instrukcje do swoich płyt głównych. Tam, gdzie to możliwe, będę się starał wskazać różnice pomiędzy płytami poszczególnych wersji.

Identyfikacja płyty głównej

Aby znaleźć numer wersji konsoli PS2, wykonaj następujące czynności:

- obróć konsolę PS2 i policz kwadratowe osłony śrub na dolnej krawędzi obudowy oraz w pobliżu gniazda rozszerzeń (lub PCMCIA);
- spójrz na naklejkę w tylnej części obudowy konsoli PS2. Najpierw przeczytaj tekst znajdujący się za napisem *Model Number* (numer modelu). Następnie zanotuj dwie pierwsze cyfry numeru seryjnego oraz dwucyfrowy kod daty (rysunek 7.2).



Rysunek 7.2. *Numer modelu, numer seryjny i kod daty*

- Poszukaj fizycznych cech odróżniających określony typ konsoli PS2. Na przykład, w gniazdo PCMCIA są wyposażone tylko konsole V0 (w wersji japońskiej); pozostałe posiadają gniazdo rozszerzeń umożliwiające podłączenie wewnętrznego dysku twardego. W konsolach w wersji V9 i wyższych nie ma portu IEEE 1394 (obok portów USB), a w niektórych, pomiędzy przyciskami Reset i Eject zamontowano odbiornik podczerwieni.
- Konsole o numerze modelu SCPH-10000 lub SCPH-15000 oraz takie, które mają gniazdo PCMCIA zamiast gniazda rozszerzeń, to urządzenia w wersji V0.
- Konsole z dziesięcioma śrubami w dolnej części obudowy to urządzenia V1, V2 lub V3. Dokładny numer wersji można stwierdzić posługując się informacjami w tabeli 7.1.
- Konsole z ośmioma śrubami w dolnej części to urządzenia w wersji V4 lub nowsze. Konsole V4 to modele o numerach od SCPH-30000 do SCPH-30006 oraz od SCPH-35001 do SCPH-35006. Jeżeli na wewnętrznej stronie pokrywy gniazda rozszerzeń zamontowany jest metalowy ekran, najprawdopodobniej mamy do czynienia z konsolą właśnie w tej wersji.
- Konsole PS2 w wersji V5 lub V6 (wersje te nie różnią się znacząco pod względem wewnętrznej budowy) to modele o numerach od SCPH-30000R do SCPH-30006 R oraz od SCPH-30000 do SCPH-30004. W celu odróżnienia konsoli w wersji V5 lub V6 od konsoli V4 należy zdjąć pokrywę gniazda rozszerzeń i odszukać niewielką śrubę w górnej części gniazda, w pobliżu jego lewej strony. Jeśli znajdziemy taką śrubę, oznacza to, że mamy do czynienia z wersją V5 lub V6. Dodatkowo wnętrze pokrywy gniazda rozszerzeń dla konsoli V5 lub V6 nie jest pokryte metalowym ekranem.

Tabela 7.1 Identyfikacja *konsol PS2 w wersjach V1, V2 i V3*

Wersja	Numer seryjny (pierwsze dwa znaki)	Kod daty
V1	U1	0D
V2	U0	0D
V3	U1	1A
V3	U2	0D

- Konsole o numerach modeli od SCPH-39000 do SCPH-39004 lub model SCPH-37000 to urządzenia w wersji V7.
- Jeżeli konsola PS2 pochodzi z Japonii, a jej numer to SCPH-39000 lub SCPH-39006, prawdopodobnie jest to wersja V8. Płyty główne konsol w wersjach V7 i V8 nie różnią się znacząco pomiędzy sobą.
- Jeżeli numer modelu mieści się w zakresie od SCPH-50000 do SCPH-50004, a kod daty jest różny od 3D, mamy do czynienia z konsolą w wersji V9. Jeżeli kod daty to 3D, nasza konsola to V10. W konsolach V9 i V10 nie ma portu IEEE 1394, a w niektórych pomiędzy przyciskami Reset i Eject zamontowany jest port podczerwieni.

Otwieranie konsoli

Pierwszą trudnością jest dostanie się do płyty głównej. Otwarcie konsoli PS2 dla kogoś, kto nigdy przedtem tego nie robił, może być trudne. Instrukcje podane poniżej dotyczą konsoli V4, a zatem być może trzeba je nieco zmodyfikować dla konsol wyprodukowanych po ukazaniu się tej wersji. Niektóre z opisanych tu czynności zupełnie nie pasują do modeli w wersji V3 i wcześniejszych.

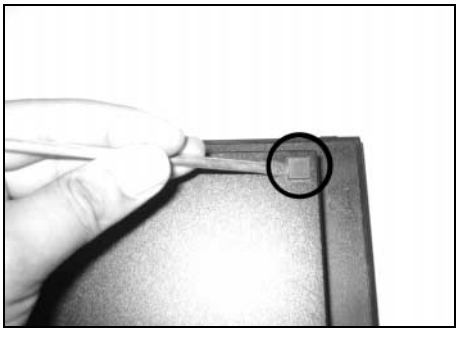
Ryzyko uszkodzenia sprzętu

Na płycie głównej konsoli PS2 i na podłączonych do niej układach znajduje się wiele elementów wrażliwych na ładunki elektrostatyczne. Dotknięcie elementów znajdujących się wewnątrz obudowy bez uprzedniego uziemienia może spowodować ich uszkodzenie. Najłatwiejszą metodą własnego uziemienia jest zakupienie antyelektrostatycznej opaski na nadgarstek i podłączenie jej do masy. Innym sposobem jest dotknięcie metalowego przedmiotu (np. kaloryfera) bezpośrednio przed dotknięciem płyty głównej

1. Obróć konsolę PS2 tak, aby jej dolna obudowa była skierowana w górę. Powinieneś ujrzeć osiem lub dziesięć kwadratowych wgłębień. Są to osłony śrub mocujących obudowę. Osłony te należy usunąć, podważając paznokciem lub płaskim wkrętakiem (rysunek 7.3).

2. Za pomocą wkrętaka odkręć śruby mocujące obudowę. Niektóre śruby mogą odkręcać się ciężko ze względu na klej użyty do montażu. Należy obrócić je energicznie do chwili usłyszenia trzasku. Teraz śruby powinny już odkręcać się bez oporu.

Rysunek 7.3. *Usuwanie osłon śrub*

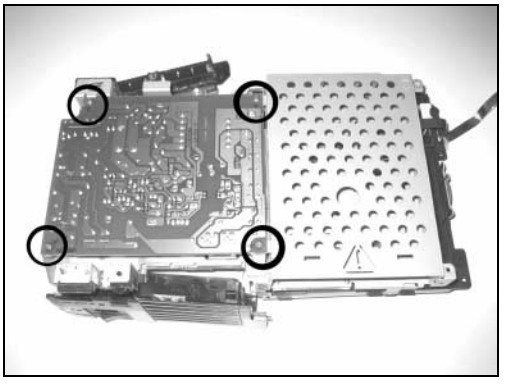


3. Jeżeli na obudowie przyklejona jest plomba gwarancyjna (zwykle obok złącza A/V z tyłu obudowy), zdejmij ją.
4. Ponownie obróć konsolę PS2 i ustaw ją w taki sposób, aby gniazdo rozszerzeń znalazło się po lewej stronie, a złącze A/V po prawej. Powoli podnieś górną część obudowy. W celu oddzielenia od układu joypada oraz napędu DVD, trzeba przesunąć ją nieco do przodu. Nie powinniśmy podnosić obudowy zbyt szybko, ponieważ jest ona w dalszym ciągu połączona z panelem przycisków Reset i Eject (rysunek 7.4).

Rysunek 7.4. *Zdejmowanie górnej osłony obudowy*

5. Zdemontuj panel przycisków *Reset* (*Eject*) poprzez pociągnięcie go do chwili usłyszenia trzasku. Wyciągnij panel przycisków poprzez otwór w obudowie. Połóż go obok konsoli, pozostawiając połączenie za pomocą taśmy.
6. Odkręć dwa wkręty mocujące panel joypada.
7. Odkręć wkręt znajdujący się po prawej stronie modułu wentylatora. Nie odkręcaj wkrętu znajdującego się bliżej napędu DVD.
8. Ostrożnie podnieś moduł. Pod nim, pomiędzy złączem dysku optycznego a złączem A/V, znajduje się kolejna śruba — odkręć ją. Wentylator jest połączony z płytą główną, a zatem zbyt szybkie podniesienie modułu może spowodować uszkodzenie połączenia.
9. Przytrzymując moduł joypada i wentylatora odwróć konsolę PS2. Upewnij się, że panel przedni jest skierowany w twoim kierunku. Powinieneś teraz bez trudu podnieść dolną część obudowy PS2. Odłóż ją na bok.

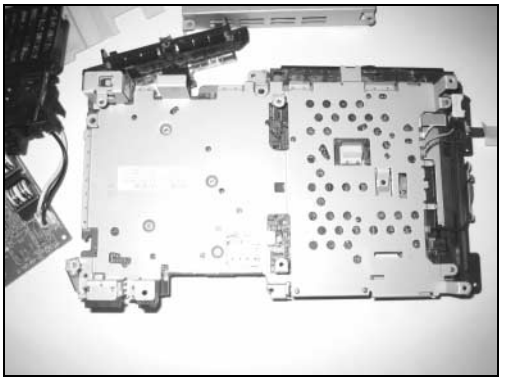
10. Konsolę po wykonaniu powyższych czynności pokazano na rysunku 7.5. Zielona płytka drukowana po lewej to zasilacz. Odkręć cztery wkręty, które go mocują.



Rysunek 7.5. *Dolna część konsoli PS2 z widocznym zasilaczem*

11. Zasilacz jest połączony z płytą główną konsoli PS2 za pomocą czterostykowego złącza. Ostrożnie podważ zasilacz, aby go rozłączyć. Na płycie znajdziesz niewielkie, dwuprzewodowe złącze, łączące wentylator z płytą główną. Pociągnij za to złącze, przytrzymując przewody jak najbliżej płyty głównej. Odłóż zasilacz i moduł wentylatora na bok.

12. Zdejmij plastikową płytkę z górnej części metalowej osłony. Zdemontuj również metalową obudowę wewnętrznego dysku twardego (zobacz rysunek 7.6).

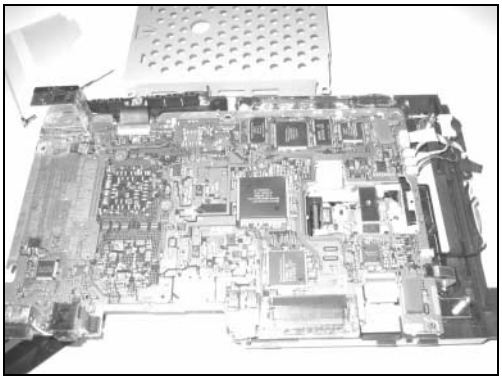


Rysunek 7.6. *Metalowa osłona u dołu*

13. Odkręć osiem małych śrubek. Cztery spośród nich znajdują się pod plastikową płytką. Dwie dodatkowe śrubki mocują złącze gniazda rozszerzeń, a ostatnie dwie znajdują się po prawej stronie metalowej osłony, nad napędem DVD. Odkręć małą czarną śrubkę znajdującą się pod napędem DVD.

14. Do napędu DVD przymocowana jest — za pomocą dwóch wypustek na dole i jednej z przodu — duża metalowa osłona. Podważ wypustki i ostrożnie zdejmij osłonę.

15. Płytę główną można podeprzeć po lewej stronie w pobliżu złącza A/V. Ja jako podpórki użyłem osłony gniazda rozszerzeń (zobacz rysunek 7.7).



Rysunek 7.7. *Dolna część płyty głównej konsoli PS2*

Konsola PS2 jest teraz całkowicie rozmontowana; można zatem przystąpić do wykonywania modyfikacji!

Instalacja portu szeregowego

W przypadku systemów wbudowanych port szeregowy często stanowi jedyny sposób komunikacji z systemem. Można go wykorzystać do ładowania programów, odbierania komunikatów o stanie urządzenia i debuggowania programów uruchomionych w systemie. Podobnie jak w przypadku większości układów SoC (*System on Chip* — system w jednym układzie), w procesorze Emotion Engine (EE) zamontowany jest szeregowy port wejścia-wyjścia, wykorzystywany wewnętrznie przez jądro systemu EE do wyprowadzania komunikatów diagnostycznych oraz informacji o statusie, a także do uruchamiania debuggera jądra. Więcej informacji na temat systemu EE konsoli PS2 można znaleźć w podrozdziale „Przegląd systemu PS2" blisko końca niniejszego rozdziału.

Port SIO można wykorzystać w tworzonym przez nas oprogramowaniu do wyprowadzania komunikatów debuggera lub w celu obsługi debuggera zdalnego, na przykład GDB (*http://sources.redhat.com/gdb*). Wykorzystując port SIO, można także uruchomić konsolę w systemie Linux dla PlayStation2. Główną korzyścią z zastosowania tego portu — w porównaniu ze standardowym kablem USB lub kartą sieciową PS2 — jest fakt, iż zapewnia on bezpośrednie połączenia z systemem EE, podczas gdy w innych wspomnianych metodach wykorzystywany jest procesor wejścia-wyjścia (IOP). W przypadku awarii procesora wejścia-wyjścia lub wystąpienia innych problemów, nie ma możliwości uzyskania danych z systemu EE. Dodatkowo, kabel SIO, który wykonamy, zapewnia dość szybką transmisję z prędkością do 115,2 kb/s.

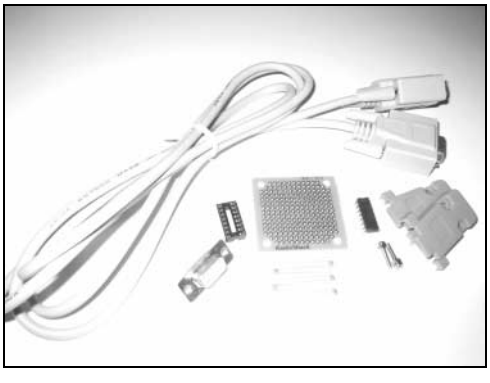
Do wykonania kabla potrzeba jedynie pięciu przewodów, które należy przylutować do płyty głównej konsoli PS2 oraz prostego układu interfejsu, wymagającego połączenia z 15 punktami.

Przygotowania

W tabeli 7.2 zestawiono elementy potrzebne do wykonania kabla SIO oraz płyty interfejsu, a na rysunku 7.8 zaprezentowano fotografię elementów. Układ MAX3323EEPE można zamówić na stronie WWW firmy Maxim (*www.maxim-ic.com*). Należy pamiętać, aby zamówić wersję układu w obudowie DIP. Pozostałe elementy można znaleźć w sklepach elektronicznych (także w sklepach internetowych zajmujących się sprzedażą części elektronicznych).

Tabela 7.2. *Lista elementów*

Liczba sztuk	Element	Uwagi
1	MAX3323EEPE	Maxim, *www.maxim-ic.com*
5	Kondensator monolityczny 0,1 μF	X
1	Żeńskie złącze DB9	x
1	Plastikowa obudowa złącza DB9	X
1	16-stykowa podstawka pod układ scalony	X
1	Płytka drukowana	X
5	Przewód 30AWG	o długości około 30 cm
1	Kabel szeregowy DB9	opcjonalnie
1	Pięciostykowe złącza męskie i żeńskie w obudowie plastikowej	opcjonalnie



Rysunek 7.8. *Materiały potrzebne do wykonania kabla SIO*

Kolory przewodów wybrane do zrealizowania połączenia zestawiono w tabeli 7.3.

Tabela 7.3. *Kolory przewodów w kablu szeregowym*

Kolor	Sygnał
Czerwony	+3.3 V (V_{CC})
Czarny	Masa (GND)
Biały	Napięcie zasilające procesor EE (V_{CORE})
Niebieski	EE_TXD oraz PC_TXD
Zielony	EE_RXD i PC_RXD

Wykonywanie projektu

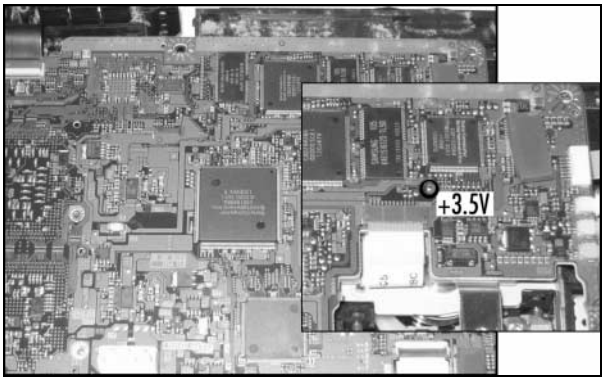
Ryzyko uszkodzenia sprzętu

Na płycie głównej konsoli PS2 znajduje się wiele montowanych powierzchniowo elementów wrażliwych na temperaturę. Nie należy używać lutownic o mocy przekraczającej 15 W. W przypadku lutownicy o regulowanej mocy przed lutowaniem upewnij się, że została nastawiona na wartość 15 W.

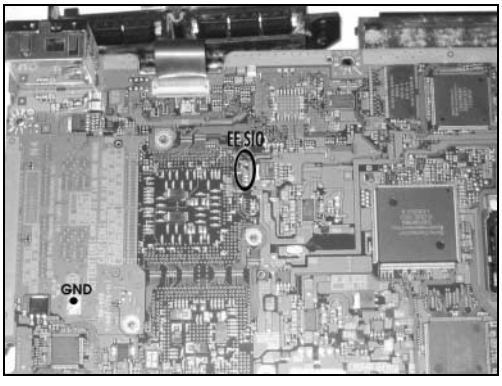
Płytę główną konsoli PS2 należy ustawić tak, aby złącze A/V znajdowało się bliżej nas, a złącze karty pamięci dalej. Sposób uzyskania dostępu do płyty głównej konsoli PS2 szczegółowo opisano w podrozdziale zatytułowanym „Otwieranie konsoli PS2". Płytę główną w wersji V4 w tym położeniu pokazano na rysunku 7.7. Rozpoczniemy od przymocowania przewodów łączących płytę PS2 z kablem szeregowym:

1. Odszukaj zacisk +3.3 V. Położenie tego zacisku na płycie V4 pokazano na rysunku 7.9. W większości witryn WWW poświęconych instalacji modchipów (np. *www.dms3.com*) można znaleźć ilustrację położenia zacisku +3.3 V dla innych wersji płyt głównych. Przylutuj jeden koniec czerwonego przewodu do wskazanego na ilustracji punktu.

Rysunek 7.9. *Położenie zacisku +3.3 V (V_{CC})*



2. Po lewej stronie płyty głównej widać litery *GH-*, po których następuje trzycyfrowa liczba. W pobliżu tego obszaru znajduje się prostokątne pole, które wykorzystamy jako masę. Położenie tego pola pokazano na rysunku 7.14. Na prostokącie mogą być widoczne ślady korozji. Przylutuj do wskazanego punktu jeden koniec czarnego przewodu.
3. Obszar z prawej strony tekstu to punkt znajdujący się bezpośrednio poniżej układu EE (zamontowanego po drugiej stronie). Obszar ten jest otoczony kilkoma małymi kondensatorami, a w środku znajduje się kilka dużych kondensatorów. W prawym górnym rogu układu EE można zauważyć cztery ułożone pionowo srebrne pola lutownicze. Pomiędzy górnym polem a pozostałymi trzema jest odstęp; jest to port SIO układu EE (rysunek 7.10).



Rysunek 7.10. *Położenie masy (GND) i portu SIO układu EE*

4. Najtrudniejsze będzie wyprowadzenie pierwszego punktu z portu SIO układu EE: będzie to styk EE_RXD. Górne kwadratowe pole lutownicze portu SIO to styk EE_TXD. Bezpośrednio nad stykiem EE_TXD znajduje się przelotka kontaktowa (ang. *via*) styku EE_RXD. Na rysunku 7.11 pokazano zbliżenie przelotki EE_RXD, pola EE_TXD oraz kondensatora V_{CORE}. Należy zachować ostrożność, aby nie podłączyć się do niewłaściwej przelotki kontaktowej w obszarze połączonym z niewielkim rezystorem. Przelotka styku EE_RXD nie jest połączona z żadnym elementem po tej stronie płyty. Ostrożnie przylutuj jeden koniec zielonego przewodu do omawianej tu przelotki.

Ryzyko uszkodzenia sprzętu

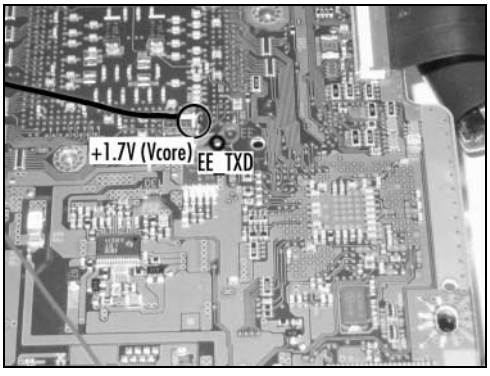
Przelotka kontaktowa (ang. *via*) to niewielki, okrągły otwór w płycie głównej. Zazwyczaj służy do łączenia jednej warstwy lub strony obwodu drukowanego z drugą stroną. W wykonywanej modyfikacji jeden z punktów (EE_RXD) jest podłączony do niewielkiej przelotki kontaktowej. Należy zwrócić uwagę na to, że przelotki kontaktowe różnią się od pól lutowniczych (ang. *pad*), czyli niewielkich kwadratowych lub okrągłych punktów pokrytych cyną.

Ponieważ przelotki łączą różne warstwy płyty głównej, ich uszkodzenie może spowodować uszkodzenie wielu warstw, a tym samym zniszczenie całej płyty głównej. Może się tak zdarzyć np. w przypadku zbytniego nagrzania przelotki. Aby uniknąć uszkodzenia płyty podczas łączenia styku EE_RXD, należy wykonać następujące czynności:

1. Za pomocą wkrętaka jubilerskiego delikatnie zdrap cynę osłaniającą przelotkę.
2. Nałóż pastę lutowniczą na odsłoniętą przelotkę.
3. Nałóż niewielką ilość cyny na lutownicę i szybko dotknij nią do przelotki tak, aby cyna spłynęła z lutownicy do przelotki.
4. Pokryj cyną przewód, który będzie użyty do wyprowadzenia sygnału EE_RXD.
5. Nałóż niewielką ilość żywicy na pokryty cyną przewód.
6. Przykładając pokryty cyną przewód do górnej części przelotki, dotknij lutownicą do kropli cyny na przelotce tak, aby połączyła się z pokrytym cyną przewodem. Podczas wykonywania tej czynności należy jak najkrócej podgrzewać przelotkę — jednak wystarczająco długo, aby zapewnić dobre połączenie lutowane pomiędzy przewodem a przelotką.

5. Zlutuj jeden koniec niebieskiego przewodu z kwadratowym polem lutowniczym styku EE_TXD, także oznaczonym na rysunku 7.11.

Rysunek 7.11. *Napięcie V_{CORE} oraz sygnały EE_TXD i EE_RXD*

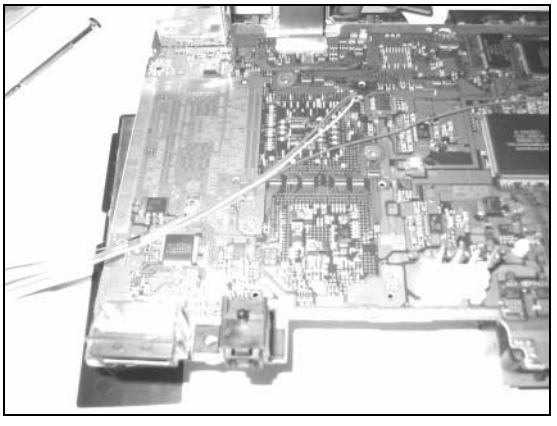


6. Ostatni punkt, jaki należy wyprowadzić z płyty, to napięcie +1,7 V, znane także jako napięcie V_{CORE}. Można je wyprowadzić z jednej z nóżek czarnego lub beżowego kondensatora znajdującego się pod układem EE. Zazwyczaj można go znaleźć obok drugiego kwadratowego pola lutowniczego, jak pokazano na rysunku 7.11. Przylutuj jeden koniec białego przewodu do nóżki kondensatora w miejscu oznaczonym małym beżowym punktem na płycie głównej.

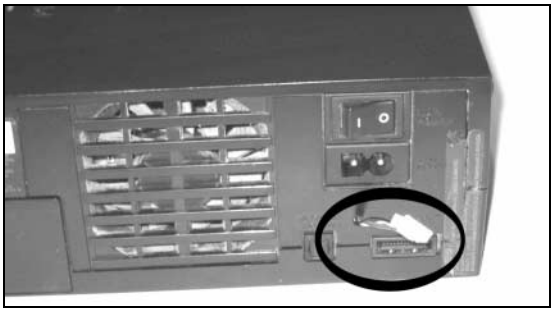
7. Po podłączeniu wszystkich pięciu przewodów (rysunek 7.12) przytwierdź je do płyty głównej za pomocą taśmy maskującej lub kleju epoksydowego, aby zabezpieczyć je przed zerwaniem. W przypadku płyty głównej w wersji V4 lub wyższej, po lewej stronie znajdziesz wycięcie, przez które przechodzą kable włącznika zasilania. Przeprowadź przewody przez to wycięcie tak, aby wychodziły z lewej strony płyty PS2 (rysunek 7.13).

Rysunek 7.12. *Przewody przylutowane do portu SIO*

8. Można teraz przystąpić do ponownego zmontowania konsoli PS2. Po zdjęciu metalowej osłony dolnej części płyty głównej można zauważyć niewielki otwór montażowy w obudowie, znajdujący się pomiędzy złączem A/V a złączem optycznym. Otwór ten jest wykorzystywany w celu przymocowania konsoli PS2 do wyświetlacza. Jeżeli chcesz zamontować kartę interfejsu na zewnątrz konsoli PS2, możesz przeprowadzić przewody przez ten otwór pod modułem wentylatora i włącznika zasilania. Sposób wyprowadzenia przewodów pokazano na rysunku 7.14. Aby zakończyć składanie konsoli, wykonaj — w odwrotnej kolejności — czynności opisane w podrozdziale „Otwieranie konsoli PS2”.



Rysunek 7.13. *Płyta główna przygotowana do zmontowania*

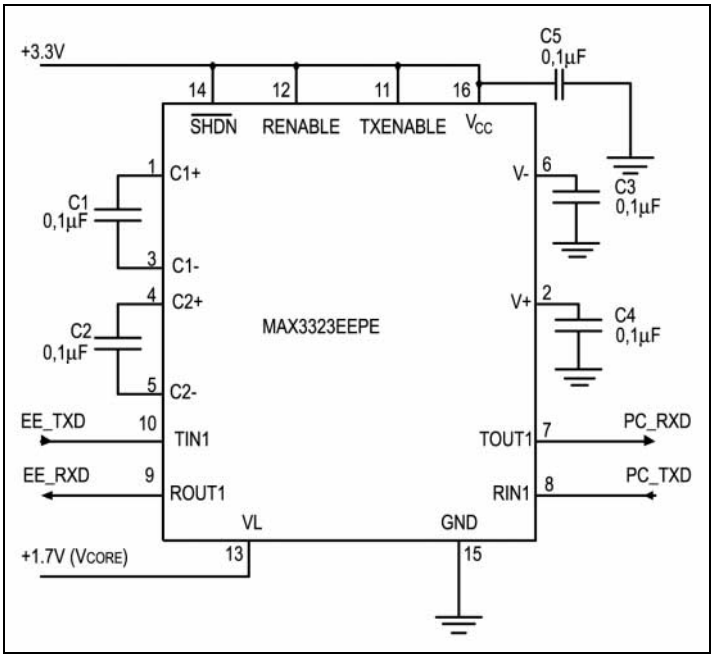


Rysunek 7.14. *Gotowe pięciostykowe złącze portu SIO*

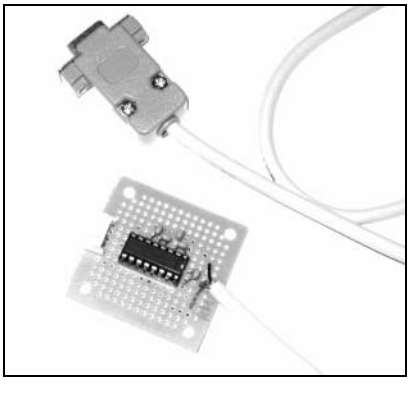
Jeżeli zdecydowałeś się na użycie pięciostykowego złącza w plastikowej obudowie, w tym momencie powinieneś przylutować przewody do jego żeńskiej części. W takim przypadku, po przymocowaniu przewodów do gniazda należy połączyć je za pomocą taśmy izolacyjnej albo koszulki termokurczliwej. Pięciostykowe gniazdo podłączone do przewodów pokazano na rysunku 7.14.

Po wyprowadzeniu przewodów z wnętrza konsoli PS2 nadszedł czas, aby wykonać zewnętrzną kartę interfejsu. Schemat karty pokazano na rysunku 7.15. Na złączu DB9 szeregowego kabla komputera PC sygnał PC_RXD będzie połączony ze stykiem 2, PC_TXD ze stykiem 3, a GND ze stykiem 5.

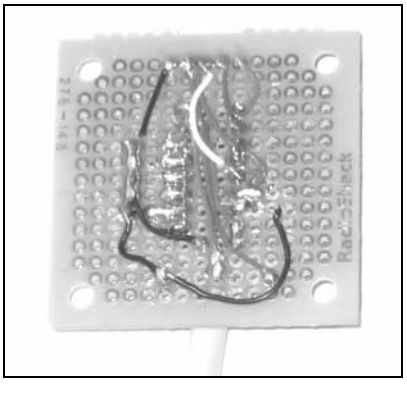
Ponieważ do połączenia karty interfejsu z konsolą użyłem pięciostykowego złącza, na jednym końcu karty zamontowałem wtyk tego złącza, natomiast na drugim przewody do utworzenia szeregowego kabla komputera PC. Do wykonania kabla użyłem starego kabla szeregowego DB9, od którego odciąłem obie końcówki. Następnie odizolowałem końce przewodów, aby przylutować je z jednej strony do karty interfejsu, a z drugiej do żeńskiego złącza DB9. Widok gotowego interfejsu z góry i z dołu pokazano na rysunkach 7.16 i 7.17.



Rysunek 7.15. *Schemat interfejsu SIO*



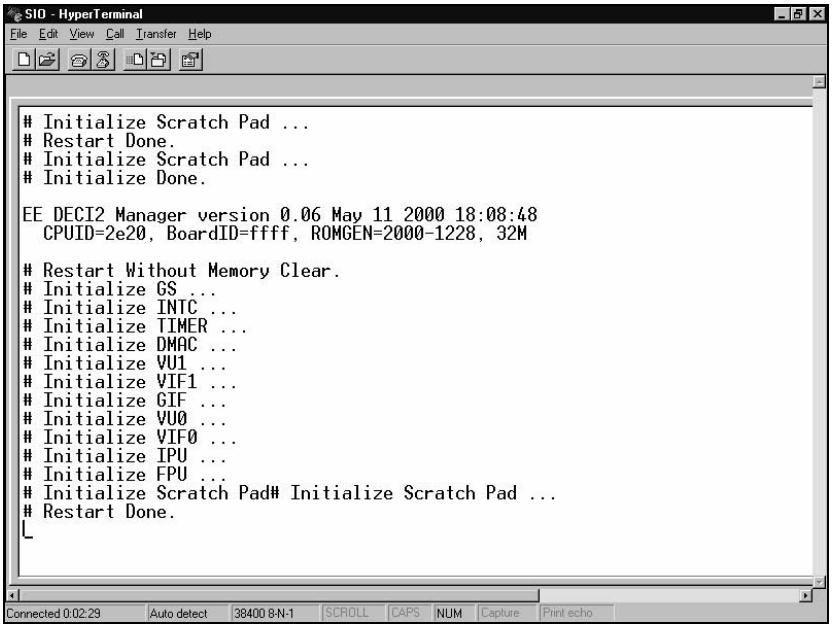
Rysunek 7.16. *Widok karty interfejsu SIO z góry*



Rysunek 7.17. *Widok karty interfejsu z dołu*

Testowanie

Aby przetestować utworzoną kartę interfejsu, podłącz ją z jednej strony do komputera PC, a z drugiej do konsoli PS2. Za pomocą programu terminalowego (np. HyperTerminal, dostarczanego wraz z systemem Windows) ustaw port szeregowy na 38400 bodów, 8 bitów danych, brak parzystości, 1 bit stopu oraz wyłącz sprzętowe sterowanie przepływem. Włącz zasilanie konsoli PS2. Jeśli kabel działa poprawnie, na ekranie wyświetlą się komunikaty pokazane na rysunku 7.18.



Rysunek 7.18. *Komunikaty układu EE*

Jak to działa?

Podstawą działania tej modyfikacji jest wykorzystanie nieudokumentowanej funkcji konsoli PS2 — portu szeregowego układu EE. Szczegółowe informacje dotyczące portu SIO można znaleźć w podrozdziale „Przegląd systemu PS2”. W standardowym porcie szeregowym RS232 w komputerze PC zazwyczaj wykorzystuje się napięcie +/–12 V do zdefiniowania logicznego zera oraz logicznej jedynki. Układ EE jest zasilany napięciem +1,7 V, a zatem, gdybyśmy bezpośrednio połączyli złącza portu SIO do portu szeregowego komputera PC, bez konwersji napięcia, sygnały przesyłane z komputera PC do układu EE mogłyby uszkodzić urządzenie. Układ MAX3323 to konwerter poziomów, przekształcający napięcia wyjściowe generowane przez układ EE do poziomu wymaganego przez komputer PC i *vice versa*. Do zasilania układu MAX3323 potrzebne jest napięcie +3,3 V, pobrane z konsoli PS2.

Ładowanie kodu z karty pamięci

15 sierpnia 2003 r opublikowałem eksploit *Independence* (Niezależność) dla konsoli PS2, który umożliwia ładowanie dowolnego oprogramowania zapisanego na standardowej karcie pamięci za pomocą mechanizmu uruchamiania gier na konsolę PS1. Można do tego wykorzystać dowolną grę konsoli PS1. Jedynym warunkiem jest zapisanie na karcie pamięci z eksploitem niepowtarzalnego identyfikatora tytułu gry. W przypadku włączenia konsoli PS2 z grą PS1 umieszczoną w napędzie DVD, eksploit przejmuje sterowanie w momencie, kiedy na ekranie pojawi się ekran powitalny z napisem Sony Computer Entertainment. Eksploit można także uruchomić poprzez umieszczenie w napędzie DVD płyty z grą PS1 i jej ręczne uruchomienie za pomocą przeglądarki PS2.

Witryna www poświęcona eksploitowi Independence

Oficjalna witryna WWW eksploita Independence to *www.0xd6.org/ps2-independence.html*. Można tu znaleźć najnowszą jego wersję, a także informacje o sposobach zapisywania plików na karcie pamięci oraz konfigurowania eksploita w systemie Windows. Od czasu jego publikacji w internecie pojawiło się wiele przewodników i instrukcji krok po kroku przedstawiających uruchamianie oprogramowania zapisanego na kartach pamięci. Niektóre z tych programów to *Naplink USB boot loader* (*http://naplink.napalm-x.com*), *Pukklink* oraz *ps2link* (*www.ps2dev.org*, sekcja *Loaders*), a także *MediaPlayer*, napisany przez grupę

PS2Reality (*www.ps2reality.net*). Obecnie opracowano wiele innych programów. Jak się okazało, wykorzystanie eksploita do załadowania programu ładującego jest najłatwiejszym sposobem uruchamiania programów użytkownika na konsoli. Chociaż termin „eksploit" brzmi nieco pompatycznie, faktem jest, że program ten „otworzył drzwi" dla społeczności hobbystycznych programistów konsoli PS2.

Przygotowania

Najtrudniejsze w wykonaniu tego projektu jest zapisanie plików na karcie pamięci. W tym celu można wykorzystać obraz dysku o nazwie *Exploit Installer*, napisany przez Nicholasa Van Veena. Aby to zrobić, można posłużyć się techniką „podmiany dysków" lub wykorzystać modchip opisany we wcześniejszej części niniejszego rozdziału. Przewodnik dotyczący instalacji można znaleźć pod adresem *http://ps2.consolevision.com/ps2homebrew3.shtml*.

Wybór odpowiedniej karty pamięci

Eksploit *Independence* działa z kartami pamięci do konsoli Sony PS2 lub kartami innych firm obsługującymi układ MagicGate; nie działa natomiast z kartami pamięci konsoli PS1 oraz z tymi, które nie obsługują wspomnianego układu.

W przypadku braku innego sposobu załadowania programu do konsoli PS2, można zakupić interfejs karty pamięci podłączany do komputera PC za pomocą kabla USB. Taki interfejs można zakupić w internecie, pod adresem *www.lik-sang.com*. Można też posłużyć się wyszukiwarką, wpisując w niej „Playstation 2" oraz „akcesoria".

Wykonywanie projektu: przygotowanie pliku TITLE.DB

Aby umożliwić uruchomienie programu przez eksploit, należy wprowadzić identyfikator tytułu gry PS1 w pliku *TITLE.DB*, zapisanym na karcie pamięci. Aby ułatwić wykonanie tego zadania, opracowałem narzędzie uruchamiane z wiersza polecenia; nazwałem je *titleman*. Wykonywalną wersję tego narzędzia, przeznaczoną dla systemu Windows, można pobrać w sekcji *Tutorials* macierzystej strony *Independence*. Można też pobrać źródła tego narzędzia i samodzielnie je skompilować.

Nie znam żadnego narzędzia z graficznym interfejsem użytkownika, za pomocą którego można by przetwarzać plik *TITLE.DB*. Aby posługiwać się nim, trzeba zatem umieć uruchamiać narzędzia wiersza polecenia. Można na przykład wykorzystać opcję *Uruchom*, dostępną w menu *Start* systemu Windows. Po wybraniu tego polecenia wyświetli się okno dialogowe, w którym można wpisać polecenie do wykonania i wcisnąć *Enter*, albo wpisać polecenie `cmd`, które spowoduje wyświetlenie okna konsoli. W przypadku systemów uniksowych informacji o sposobie uruchamiania narzędzi wiersza polecenia należy poszukać w dokumentacji systemu.

Opcje programu *titleman* wyszczególniono w tabeli 7.4. Najpierw należy skorzystać z opcji `-c`, aby utworzyć plik *TITLE.DB* na dysku i zapisać kod eksploita oraz kilka standardowych identyfikatorów tytułów.

Tabela 7.4. *Opcje programu titleman*

Opcja	Opis
`-c`	Utworzenie pliku *TITLE.DB* i zainicjowanie eksploita
`-a`	Dodanie co najmniej jednego identyfikatora tytułów do pliku *TITLE.DB*
`-d`	Usunięcie co najmniej jednego identyfikatora tytułów z pliku *TITLE.DB*
`-l`	Wyświetlenie wszystkich identyfikatorów tytułów zapisanych w pliku *TITLE.DB*
`-o`	Określenie alternatywnego pliku wynikowego
`-v`	Wyświetlanie opisowych komunikatów statusowych

W treści narzędzia *titleman* można znaleźć wykonywalny kod na konsolę PS2, uruchamiający plik *BOOT.ELF* z karty pamięci. Plik *BOOT.ELF* jest zapisany w folderze konfiguracyjnym konsoli, który można przeglądać za pomocą przeglądarki konsoli PS2. Nazwa tego folderu to:

- *BADATA-SYSTEM* dla konsol PS2 zakupionych w Ameryce Północnej;
- *BEDATA-SYSTEM* dla konsol PS2 zakupionych w Europie;
- *BIDATA-SYSTEM* dla konsol PS2 zakupionych w Japonii i innych krajach azjatyckich.

Podczas przygotowania zrzutu gry (zobacz podrozdział „Zapisywanie pliku TITLE.DB na karcie pamięci") należy pamiętać, aby użyć nazwy odpowiadającej właściwemu regionowi geograficznemu.

Po utworzeniu pliku *TITLE.DB* dodajemy identyfikatory wszystkich gier konsoli PS1, za pomocą których chcemy uruchomić eksploit. Identyfikator tytułu zazwyczaj można znaleźć pod ikoną z oceną ESRB gry na zadrukowanej stronie płyty. Po odczytaniu identyfikatora z płyty, należy go przekształcić na postać zrozumiałą dla konsoli PS2. Na przykład, na płycie z grą *Street Fighter Alpha* znajduje się identyfikator *SLUS-00197*. Aby był on „zrozumiały" dla konsoli PS2, należy zamienić myślnik na znak podkreślenia oraz wprowadzić kropkę pomiędzy trzecią a czwartą cyfrą w pięciocyfrowej liczbie za myślnikiem. Tak więc identyfikator *SLUS-00197* należy przekształcić na *SLUS_001.97*.

Identyfikator tytułu można także odczytać z pliku *SYSTEM.CNF* po otwarciu płyty z grą w komputerze PC. Wartość w opcji `BOOT` za znakiem równości (`=`) to właśnie poszukiwany identyfikator. Ponieważ jest zapisany w pliku *SYSTEM.CNF*, posiada już właściwy format i nie trzeba go przekształcać.

Po uzyskaniu identyfikatorów tytułów dla wszystkich gier dla konsoli PS1, które chcemy uruchomić, należy wprowadzić je do pliku *TITLE.DB*. Można to zrobić pojedynczo lub użyć wsadowego trybu narzędzia *titleman*. Składnia polecenia służącego do wprowadzenia pojedynczego zapisu do pliku *TITLE.DB* jest następująca:

```
titleman -a title_id
```

Zatem w moim przypadku, powinienem wpisać `titleman -a SLUS_001.97`.

Znacznie wygodniejsze jest jednak użycie trybu wsadowego; umożliwia dodanie za pomocą jednego polecenia wielu identyfikatorów tytułów. Aby skorzystać z trybu wsadowego, należy umieścić wszystkie identyfikatory tytułów w oddzielnych wierszach pliku tekstowego i użyć opcji `-a` z nazwą pliku wsadowego poprzedzoną symbolem *@*. W pliku wsadowym można umieszczać komentarze; są to wiersze rozpoczynające się od znaku średnika. Przykład pliku wsadowego pokazano na listingu 7.1.

Listing 7.1. *Przykład pliku wsadowego programu titleman*

```
; Xenogears (dysk 1)
SLUS_006.64

; Xenogears (dysk 2)
SLUS_006.69

; Broken-Helix
SLUS_002.89

; Suikoden
SLUS_002.92

; Suikoden II
SLUS_009.58

; Sentient
SLUS_941.10

; Blood Omen: Legacy of Kain
SLUS 000.27
```

Polecenia wymagane do wprowadzenia pojedynczego tytułu do pliku *TITLE.DB* (*Street Fighter Alpha*) zaprezentowano na listingu 7.2.

Listing 7.2. *Polecenia do umieszczenia zapisu w pliku TITLE.DB*

```
$ titleman -c
$ titleman -a SLUS_001.97
```

W przypadku popełnienia błędu podczas dodawania identyfikatora lub w celu usunięcia wcześniej dodanego identyfikatora, należy skorzystać z opcji `-d`. Pojedyncze identyfikatory usuwa się z pliku *TITLE.DB* analogicznie do ich dodawania (różnica polega na użyciu opcji `-d` zamiast opcji `-a`). Można też skorzystać z trybu wsadowego.

Aby sprawdzić, jakie identyfikatory tytułów zostały dodane do pliku *TITLE.DB*, można wykorzystać opcję `-l`.

Wybór pliku BOOT.ELF

Wersja 0.1 eksploita ładuje wykonywalny plik *BOOT.ELF* z karty pamięci natychmiast po przejęciu sterowania przez eksploit. Plik ten jest zapisany w tym samym folderze co plik *TITLE.DB*. Dla użytkowników chcących tworzyć własne oprogramowanie dla konsoli PS2, najlepszym programem *BOOT.ELF* jest *ps2link* — program Open Source umożliwiający ładowanie programów przez kartę sieciową konsoli PS2. Najnowszą wersję programu *ps2link* można pobrać pod adresem *www.thethirdcreation.net/tools* lub *www.ps2dev.org* (w sekcji *Loaders*).

Użytkownicy zainteresowani odtwarzaniem strumieni wideo, plików MP3 oraz Ogg Vorbis przez sieć mogą zainstalować program *MediaPlayer* firmy PS2Reality. Podręczniki objaśniające sposób wykorzystania programu *MediaPlayer* z eksploitem są dostępne pod adresem *www.ps2reality.net* (uwaga: witryna jest hiszpańskojęzyczna!).

Wreszcie, jeśli ktoś chce uruchomić jedną ze starych gier na konsolę Sega Genesis, może wykorzystać emulator konsoli Sega Genesis *PGEN* autorstwa Nicholasa Van Veena. Emulator PGEN można pobrać z internetu; znajdziemy go m.in. pod adresem *http://pgen.gamebase.ca*.

Aby wykorzystać dowolny z poprzednio wymienionych programów z eksploitem, należy je odpowiednio skonfigurować — w taki sposób, aby załadowały się natychmiast po przejęciu sterowania przez eksploit. Niektóre programy są rozprowadzane wraz ze szczegółowymi instrukcjami w pliku *README*. Jeżeli nie ma takich instrukcji, należy zmienić nazwę głównego pliku *ELF* programu na *BOOT.ELF* (na przykład z *PGEN_11.ELF* na *BOOT.ELF*).

Zapisywanie pliku TITLE.DB na karcie pamięci

Po utworzeniu w pliku *TITLE.DB* listy wszystkich tytułów gier, za pomocą których chcemy uruchomić eksploit, należy zapisać ten plik na karcie pamięci. Uzyskamy to na kilka sposobów. Możemy na przykład zapisać plik:

- **za pomocą programu Exploit Installer Nicholasa Van Veena**. Skorzystanie z tej metody wymaga zainstalowania modchipa lub zastosowania techniki podmiany dysków. Chociaż technika podmiany dysków jest skuteczna, osobiście nie polecam jej stosowania, ponieważ grozi to zniszczeniem konsoli. Program *Exploit Installer* można pobrać pod adresem *www.ps2newz.net/forums/showthread.php?threadid=14803*.
- **za pomocą interfejsu kart pamięci podłączanego do komputera PC poprzez port USB**. Urządzenia te umożliwiają zapisanie programów z dysku komputera PC na karcie pamięci umieszczonej w interfejsie. Aby skorzystać z tej metody, należy utworzyć nowy zrzut gry i nadać mu nazwę *Your System Configuration*, a następnie zapisać folder zrzutu odpowiadający regionowi geograficznemu, w którym zakupiono konsolę, albo otworzyć istniejący plik i zmienić jego nazwę. Po utworzeniu zrzutu gry, należy skopiować do niego pliki *TITLE.DB* i *BOOT.ELF*. Należy pamiętać, że nazwy plików *TITLE.DB* oraz *BOOT.ELF* muszą składać się wyłącznie z wielkich liter; w innym przypadku eksploit nie będzie działać. Po utworzeniu zrzutu należy skopiować go na kartę pamięci, wykorzystując oprogramowanie dostarczane wraz z interfejsem.
- **za pomocą narzędzia nPort**. Napalm — grupa programistów, twórców programu *Naplink*, służącego do ładowania programów poprzez port USB, opracowała także narzędzie *nPort*, umożliwiające kopiowanie zrzutów gier pomiędzy komputerem PC a konsolą PS2 przez istniejące połączenie USB (nawiązane za pomocą programu *Naplink*) albo przez kartę sieciową (w programie *Pukklink* lub *ps2link*). Na oficjalnej stronie WWW *Independence* jest dostępne archiwum *.npo* (format obsługiwany przez *nPort*) zawierające wymagane pliki. Z witryny można również pobrać przewodnik opisujący sposób wykorzystania programu *nPort* do zapisywania eksploita. Program *nPort* jest dostępny pod adresem *http://wire.napalm-x.com*.

Niezależność!

Po zapisaniu eksploita na karcie pamięci włóż kartę do konsoli PS2 i umieść w napędzie DVD dowolną grę, której tytuł umieściłeś w pliku *TITLE.DB*. Po załadowaniu dysku najpierw wyświetli się biały ekran, a po krótkiej chwili ukaże się ekran początkowy programu *BOOT.ELF*. Jeżeli uruchomi się zwykły emulator konsoli PS1, sprawdź zawartość pliku *TITLE.DB* za pomocą opcji *-l* i upewnij się, czy nie popełniłeś pomyłki w pisowni identyfikatora. Sprawdź także, czy w napędzie znajduje się płyta z właściwą grą. Jeżeli podczas ładowania eksploita wyświetli się czerwony ekran, upewnij się, że umieściłeś plik *BOOT.ELF* w tym samym folderze, co plik *TITLE.DB*. Inne błędy, jakie mogą wystąpić podczas ładowania, są specyficzne dla aplikacji zapisanej w pliku *BOOT.ELF*.

Jak to działa?

Konsola PS2 umożliwia emulację konsoli PS1 za pomocą kombinacji emulacji sprzętowej i graficznego, programowego emulatora PS1 o nazwie *PS1DRV*. W przypadku próby załadowania gry na konsolę PS1 w konsoli PS2, przeglądarka systemowa najpierw uruchamia emulator *PS1DRV* z systemu BIOS. Emulator *PS1DRV* wykonuje kilka

czynności konfiguracyjnych, takich jak ustawienie szybkości dysku oraz wczytanie parametrów graficznych dla wybranej gry. Na koniec inicjuje emulator graficzny i przestawia procesor IOP w tryb konsoli PS1. Teraz sterowanie przejmuje procesor IOP, który ładuje grę na konsolę PS1 z dysku. Grafika jest emulowana za pomocą specjalnego kanału SIF DMA w procesorze EE pomiędzy procesorem IOP a emulatorem *PS1DRV*.

Po umieszczeniu dysku PS1 w konsoli PS2 przeglądarka systemowa odczytuje identyfikator tytułu z zapisanego na dysku pliku o nazwie *SYSTEM.CNF*. W pliku tym podane są także inne parametry ładowania, takie jak domyślny tryb wideo, dla którego napisano grę. Identyfikator tytułu gry jest przekazywany do emulatora *PS1DRV*, dzięki czemu można wybrać parametry graficzne dopasowane do danej gry. Jeżeli przeglądarka systemowa nie może znaleźć pliku *SYSTEM.CNF* na dysku PS1, do emulatora *PS1DRV* przekazywana jest wartość *???*.

Kiedy emulator *PS1DRV* poszukuje parametrów grafiki dla określonej gry, przeszukuje trzy lokalizacje: wbudowaną tabelę, plik *SYSTEM.CNF* zapisany na dysku oraz inny plik, zapisany na karcie pamięci, noszący nazwę *TITLE.DB*. Ten ostatni znajduje się w folderze systemowym, zarezerwowanym dla programów BIOS-u. W przypadku konsol PS2 zakupionych w Japonii lub Azji, folder ten ma nazwę *BIDATA-SYSTEM*; w przypadku konsol przeznaczonych dla krajów europejskich jest to folder *BEDATA-SYSTEM*, natomiast w Ameryce Północnej nazywa się on *BADATA-SYSTEM*.

Na pomysł stworzenia eksploita *Independence* wpadłem analizując procedury przetwarzające identyfikator tytułu z pliku *TITLE.DB*. Główna procedura ładująca informacje z pliku *TITLE.DB*, którą nazwałem `load_mc_title.db()`, służy do ładowania pliku *TITLE.DB* z karty pamięci do pamięci RAM. Procedura ta wywołuje inna procedurę — *find_title_params()*, której zadaniem jest odszukanie identyfikatora tytułu w załadowanym pliku *TITLE.DB* i zwrócenie ciągu znaków z wartościami parametrów. Warto zauważyć, że sposób załadowania pliku *TITLE.DB* do pamięci RAM powoduje, że implementacja eksploita jest dziecinnie łatwa. Cała zawartość pliku *TITLE.DB* jest ładowana pod stały adres RAM 0x20800000. Oznacza to, że w pliku *TITLE.DB* możemy umieścić cały program, który przejmie sterowanie po zakończeniu działania eksploita i będziemy dokładnie znali miejsce załadowania tego programu!

Do procedury `find_title_params()` są przekazywane trzy parametry: adres określający, gdzie jest ładowany plik *TITLE.DB* (`title_db`), adres zmiennej łańcuchowej, pod który zostaną przekazane parametry (`params`) oraz poszukiwany identyfikator tytułu (`title_name`). Procedura wykonuje pętlę przeszukującą każdy wiersz — wiersze są zakończone znakiem wysuwu wiersza (`LF`), powrotu karetki (`CR`) lub kombinacją tych znaków — w poszukiwaniu identyfikatora tytułu. W przypadku znalezienia identyfikatora przeszukiwana jest dalsza część wiersza za znakiem równości, aż do znaku końca wiersza. Po osiągnięciu końca wiersza uzyskany łańcuch znaków jest kopiowany do zmiennej łańcuchowej `params`. Z tej operacji kopiowania skorzystamy w utworzonym eksploicie.

W idealnej sytuacji, rozmiar parametru powiązanego z identyfikatorem tytułu powinien wynosić około 25 bajtów, łącznie ze znakiem końca wiersza. W procedurze `load_mc_title_db()` do zapisania tej wartości przydzielany jest bufor o rozmiarze 256 bajtów. Bufor ten jest zapisany w pamięci RAM obok bardzo ważnego rejestru układu EE — adresu powrotu, czyli rejestru `$ra`. Kiedy w architekturze MIPS procedura wykonuje

inną procedurę, zapisuje rejestr $ra do pamięci RAM, ponieważ procesor automatycznie aktualizuje rejestr $ra w taki sposób, że wskazuje na adres ostatniej instrukcji procedury wywołującej. Po zakończeniu wykonywania procedury wywoływanej, rejestr $ra w dalszym ciągu wskazuje na tę instrukcję, a zatem przed wyjściem z procedury należy odtworzyć rejestr $ra z pamięci RAM. W procedurze load_mc_title_db() rejestr $ra zapisywany przed wywołaniem procedury find_title_params() znajduje się przed miejscem przydzielonym do zapisania wartości ciągu znaków params.

Kiedy procedura find_title_params() kopiuje łańcuch znaków do zmiennej params, wykorzystuje funkcję języka C strcpy(), która kopiuje ciąg dowolnej długości do innego ciągu. W funkcji strcpy() nie jest przeprowadzane sprawdzanie rozmiaru ciągu, a zatem funkcja kopiuje ciągi do momentu napotkania znaku NUL (znak o kodzie ASCII 0). Oznacza to, że gdybyśmy skonstruowali w pliku *TITLE.DB* ciąg znaków o rozmiarze przekraczającym 256 bajtów przydzielonych na zmienną params, moglibyśmy zastąpić zapisaną wartość rejestru $ra (ponieważ jest ona zapisana w pamięci RAM za wartością zmiennej params). Wartość, która zastąpi adres zapisany w rejestrze $ra, stanie się adresem następnej instrukcji wykonywanej po zakończeniu procedury load_mc_title_db(). Może to być dowolny adres w pamięci RAM poprawny dla układu EE.

Ten rodzaj eksploita, wykorzystujący przepełnienie bufora, jest powszechnie wykorzystywany w programach, w których nie wykonuje się sprawdzenia rozmiaru ciągów znaków lub innych wartości ładowanych z plików danych. Ma on jednak istotną wadę, polegającą na tym, że bardzo łatwo zapobiec możliwości jego użycia. Dotyczy to również eksploita *Independence* emulatora *PS1DRV*. Wykorzystując standardową funkcję C strncpy(), można określić maksymalną długość kopiowanego łańcucha znaków. Gdyby firma Sony użyła funkcji strncpy() z maksymalną długością 256 znaków, utworzenie eksploita nie byłoby możliwe.

A zatem, na jaki adres ustawimy rejestr $ra podczas konstruowania ciągu znaków w pliku *TITLE.DB*? Jak pamiętamy, procedura load_mc_title_db() ładuje całą zawartości pliku *TITLE.DB* do pamięci RAM pod stały adres 0x20800000. Rejestr $ra możemy ustawić na dowolny adres za adresem ładowania pliku *TITLE.DB*. W eksploicie *Independence* wykorzystałem stały adres 0x20810110, co pozwala na zapisanie w pliku *TITLE.DB* około 200 wpisów. Po zakończeniu działania procedury load_mc_title_db(), rejestr $ra wskazuje na ten adres, a mój kod przejmuje sterowanie nad konsolą PS2.

Inne projekty: niezależne dyski twarde

Wraz z wprowadzeniem na rynek pakietu *PlayStation 2 Linux Kit* (w maju 2002 r.) oraz oficjalną premierą karty sieciowej (w sierpniu 2002), wprowadzono obsługę dysku twardego dla konsoli PS2. Oprócz oficjalnego twardego dysku firmy Sony, dostarczanego wraz z pakietem *Linux Kit*, użytkownicy konsoli PS2 mogą wykorzystywać dowolne dostępne na rynku dyski twarde, które można podłączyć do złącza IDE karty sieciowej. Choć firma Sony zapowiedziała oficjalną premierę dysku twardego dla użytkowników (tych, którzy nie zakupili pakietu *Linux Kit*) na marzec 2004 roku, hobbyści już dużo wcześniej pisali oprogramowanie wykorzystujące dyski twarde.

W listopadzie 2003 r. Nicholas Van Veen oraz kilku innych programistów (w tym ja sam) opracowali bibliotekę *libHDD* — zbiór bibliotek i sterowników umożliwiających wykorzystywanie dysku twardego w programach użytkowników. W bibliotece tej (dostępnej pod adresem *http://ps2dev.org/kb.x?T=967*) zawarta jest także obsługa oficjalnego systemu plików firmy Sony dla programów wykorzystujących dysk twardy. Dzięki bibliotece *libHDD*, programy przeznaczone do uruchomienia z karty pamięci mogą wykorzystywać dane lub nawet inne programy zapisane na dysku twardym. Otwiera to drogę dla wielu projektów, od emulatorów obsługujących ładowanie zrzutów gier z dysku twardego (np. PGEN) do programów multimedialnych służących do odtwarzania plików audi o i wideo zapisanych na dysku twardym. Można przypuszczać, że w miarę pojawiania się coraz większej liczby programów obsługujących dysk twardy, konsola PS2 zyska większe uznanie jako platforma hackingu sprzętowego.

Przegląd systemu PS2

Konsola PS2 jest systemem o architekturze równoległej — wysoka wydajność uzyskiwana jest dzięki podziałowi zadań programów na wiele procesorów i koprocesorów. Logika typowej gry na konsolę PS2 wykonuje się na głównym procesorze, dane wejściowe wprowadzane przez użytkownika są obsługiwane na procesorze pomocniczym, a trójwymiarowe przekształcenia geometryczne na jednym z dwóch ultraszybkich koprocesorów. Dla porównania, w tradycyjnej architekturze komputera PC, do obsługi danych wejściowych użytkownika, logiki gry oraz przetwarzania grafiki wykorzystywany jest tylko jeden procesor. Oczywiście nowoczesne karty graficzne 3D są wyposażone w szybkie, programowalne procesory graficzne, które przejmują obliczenia związane z oświetleniem oraz przetwarzaniem wierzchołków, odciążając w ten sposób procesor główny.

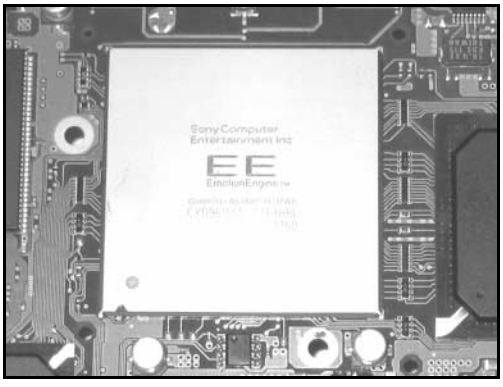
Układ Emotion Engine

Zasadniczym elementem konsoli PS2 jest 64-bitowy procesor MIPS o nazwie *Emotion Engine* — EE (rysunek 7.19). Procesor EE jest poprzednikiem linii procesorów SoC (*System on a Chip* — system w jednym układzie) TX79 firmy ArTile Micro. W procesorach SoC wszystkie urządzenia końcowe zarządzające systemem są zintegrowane w jednym układzie. Na przykład w popularnym układzie rodziny TX79 — TMPR7901 — zintegrowano między innymi główny procesor MIPS, kontroler pamięci SDRAM, kontroler szyny PCI oraz kontroler sieci Ethernet. Największą zaletą procesorów SoC są zmniejszone koszty produkcji oraz większa wydajność zintegrowanych peryferiów. W architekturze SoC zazwyczaj występuje co najmniej jedna ultraszybka szyna wewnętrzna, która komunikuje się z urządzeniami zewnętrznymi za pomocą wolniejszej, wspólnej szyny systemowej.

Na procesor EE składają się następujące udokumentowane urządzenia końcowe:

- procesor główny MIPS R5900 (CPU);
- dwa procesory przetwarzania wektorów (VU lub VPU);
- procesor zmiennoprzecinkowy (FPU);

Rysunek 7.19. *Procesor Emotion Engine*



- kontroler DMA (DMAC);
- kontroler przerwań (INTC);
- programowalne zegary;
- interfejs pomocniczego procesora (SIF);
- dwa interfejsy procesorów VU (VIF);
- interfejs układu syntezy grafiki (GIF);
- procesor przetwarzania obrazów (IPU).

Można także wymienić trzy urządzenia nieudokumentowane:

- kontroler RDRAM [(R)DRAMC];
- szeregowy port wejścia-wyjścia (SIO, UART);
- interfejs JTAG (IEEE 1149.1).

Dokumentacja to doskonałe źródło wiedzy

Szczegółowa analiza wszystkich zintegrowanych urządzeń końcowych układu EE zajęłaby kilka rozdziałów. Podręczniki dostarczone wraz z pakietem *PlayStation2 Linux Kit* (*www.playstation2-linux.com*) są nieocenionym źródłem informacji na temat działania procesora *Emotion Engine* oraz układu syntezy grafiki. Użytkownicy, którzy nie są właścicielami pakietu, znajdą podstawowe informacje na temat procesora Emotion Engine w podręcznikach dostarczanych z procesorem TX79 — potomkiem układu EE. Podręczniki te można pobrać ze strony pod adresem *www.semicon.toshiba.co.jp/eng/index.html*. Przewodniki programowania konsoli PS2 są natomiast dostępne pod adresami *www.ps2dev.org* oraz *www.oopo.net/consoledev*.

Szeregowy port wejścia-wyjścia

Port SIO jest realizacją szybkiego układu UART z ośmiobajtową kolejką FIFO dla transmisji oraz szesnastobajtową kolejką FIFO dla odbioru. Układ obsługuje także standardowe sygnały CTS i RTS, wykorzystywane do sprzętowego sterowania przepływem. Chociaż w konsolach PS2 dostępnych na rynku styki portu SIO nie są podłączone, BIOS układu EE oraz jądro systemu *runtime* wykorzystują port SIO do wyprowadzania komunikatów o stanie urządzenia podczas procesu ładowania kodu w konsoli PS2.

Port SIO nie jest udokumentowany w podręczniku użytkownika procesora EE. Z tego powodu kod inicjalizacji oraz komunikaty wyjściowe odczytałem z BIOS-u konsoli PS2. Mnóstwo informacji o przerwaniach portu SIO oraz rejestrach sprzętowych znalazłem w podręczniku *Core Architecture Manual* procesora Toshiba TX79. Okazuje się, że podręcznik ten jest niemal identyczny z podręcznikiem *Core User's Manual* procesora EE, poza tym, że z tego drugiego usunięto informacje dotyczące portu SIO.

Po przestudiowaniu BIOS-u, kodu obsługi portu SIO w jądrze oraz dostępnej dokumentacji i po napisaniu kilku testowych programów, mogłem sporządzić dość dokładną listę rejestrów i definicji. Większość rejestrów portu SIO w układzie EE jest identyczna z rejestrami tego portu w procesorze TX49, jednak nic nie wskazuje na to, że występuje w nim obsługa DMA. Można również znaleźć kilka rejestrów analogicznych do układu UART procesora TX7901. Mapę rejestrów portu SIO zaprezentowano w tabeli 7.5.

Tabela 7.5. *Mapa rejestrów portu SIO*

Adres	Nazwa	Opis
0x1000f100	SIO_LCR	Rejestr sterowania linią
0x1000f110	SIO_LSR	Rejestr stanu linii
0x1000f120	SIO_IER	Rejestr zezwolenia na przerwanie
0x1000f130	SIO_ISR	Rejestr stanu przerwania
0x1000f140	SIO_FCR	Rejestr sterowania kolejką FIFO
0x1000f150	SIO_BGR	Rejestr sterowania szybkością transmisji
0x1000f180	SIO_TXFIFO	Rejestr kolejki FIFO transmisji
0x1000f1c0	SIO_RXFIFO	Rejestr kolejki FIFO odbioru

Z podręcznika procesora TX79 dowiedziałem się, że jeśli port SIO ma potrzebę przerwania działania procesora CPU, generuje wyjątek `Debug` i ustawia 12 bit rejestru przyczyny `COP0`. Procesor CPU dekoduje wówczas rejestr stanu procesora, aby określić przyczynę przerwania. Jądro procesora EE wykorzystuje wyjątek SIO jako mechanizm uruchamiania wbudowanego debuggera w jądrze.

Aby zainicjować port SIO, należy najpierw zapisać do rejestru `SIO_LCR` wartość określającą liczbę bitów danych i bitów stopu oraz informację o włączeniu lub wyłączeniu kontroli parzystości. Można również określić źródło zegara, potrzebne do określenia szybkości transmisji. W następnym kroku obie kolejki FIFO są wyzerowane oraz — opcjonalnie — włączane są przerwania. Na koniec należy obliczyć dzielnik i wartość zegara potrzebne do określenia szybkości transmisji. Na listingu 7.3 zaprezentowano przykładowy kod inicjalizacji portu SIO z podaną szybkością transmisji i standardowymi parametrami 8N1 (8 bitów danych, bez kontroli parzystości, 1 bit stopu).

Listing 7.3. *Przykład kodu inicjalizacji portu SIO*

```
#define SIO_FCR_FRSTE 0x01      /* Zezwolenie na zerowanie kolejek FIFO. */
#define SIO_FCR_RFRST 0x02      /* Zerowanie kolejki FIFO RX. */
#define SIO_FCR_TFRST 0x04      /* Zerowanie kolejki FIFO TX. */

#define CPUCLK  294912000       /* Wartość wykorzystywana do określenia dzielnika
szybkości transmisji. */
```

```
void sio_init(u32 baudrate)
{
    u32 brd;         /* Dzielnik szybkości transmisji.  */
    u8 bclk = 0;     /* Zegar do obliczenia szybkości transmisji.  */

    /* 8 bitów danych, 1 bit stopu, bez kontroli parzystości, z wykorzystaniem
zegara CPU do obliczenia szybkości transmisji.  */
    _sw((1<<5), SIO_LCR);
    /* Wyłączenie wszystkich przerwań.  */
    _sw(0, SIO_IER);

    /* Wyzerowanie kolejek FIFO.  */
    _sw(SIO_FCR_FRSTE|SIO_FCR_RFRST|SIO_FCR_TFRST, SIO_FCR)
    /* Włączenie kolejek FIFO.  */
    _sw(0, SIO_FCR) ;

    brd = CPUCLK/(baudrate * 256);

    while ((brd >= 256) && (++bclk < 4))
            brd /= 4;

    _sw((bclk << 8) | brd, SIO_BGR);
}
```

Sposób wysyłania i odbierania znaków jest bardzo prosty: aby wysłać znak, należy zapisać wartość do rejestru `SIO_TXFIFO`; aby go odebrać, należy odczytać wartość z tego rejestru. Trzeba również sprawdzić rejestr `SIO_ISR`, aby upewnić się, czy w buforze `TX_FIFO` jest miejsce na kolejny znak lub czy w kolejce `RX_FIFO` jest chociaż jeden znak, który można odczytać. Kod służący do wysyłania i odbierania znaków przypominający standardowe funkcje ANSI C `putc()` oraz `getc()` pokazano na listingu 7.4.

Listing 7.4. *Przykład kodu wejścia-wyjścia portu SIO*

```
int sio_putc(int c)
{
    /* Blokowanie do momentu uzyskania gotowości do transmisji  */
    while ((_lw(SIO_ISR) & 0xf000) == 0x8000);

    _sb(c, SIO_TXFIFO);
    return c;
}

int sio_getc()
{
    /* Czy w buforze RX_FIFO jest jakiś znak? */
    if (_lw(SIO_ISR) & 0xf00)
            return _lb(SIO_RXFIFO);
            /* Zwrócenie znaku końca pliku (EOF).  */
            return -1;
}
```

Procesor wejścia-wyjścia

Procesor wejścia-wyjścia (*I/O processor* — IOP) zarządza pracą większości wbudowanych i zewnętrznych urządzeń końcowych, włącznie z kartami pamięci, jednostką przetwarzania dźwięku, kontrolerami oraz napędem DVD. Jest to struktura SoC ###from LSI Logic###, opracowana na podstawie oryginalnej konsoli PlayStation (PS1), spełniająca wszystkie główne funkcje konsoli PS1 i zrealizowana w jednym układzie scalonym. Rdzeń procesora IOP tworzy procesor MIPS R3000A, taktowany zegarem 36,864 MHz. Wewnętrzna szybkość procesora IOP stanowi w przybliżeniu 1/8 szybkości procesora EE taktowanego zegarem 294,9 MHz. Podczas emulacji sprzętu PS1 procesor IOP działa z oryginalną szybkością konsoli PS1 — tzn. 33 MHz. Układ IOP może bezpośrednio zaadresować do 2 MB pamięci RAM. Z urządzeniami wewnętrznymi i peryferiami zewnętrznymi komunikuje się za pomocą szyny SBUS.

Interfejs procesora pomocniczego

Procesor IOP czasami określa się jako procesor pomocniczy. W tym przypadku układ EE jest procesorem głównym. Interfejs procesora pomocniczego (*Sub-CPU Interface* — SIF) jest szybkim połączeniem pomiędzy procesorami IOP i EE, realizowanym poprzez kanał DMA. Dzięki interfejsowi SIF, każdy z procesorów może przesyłać dane bezpośrednio do pamięci RAM drugiego procesora. Najpopularniejszym zastosowaniem interfejsu SIF jest interfejs wywołania zdalnej procedury (*Remote Procedure Call* — RPC), umożliwiający procesorowi EE wywoływanie procedur w procesorze IOP. Procedury te wykorzystują niskopoziomowy sterownik sprzętu, odpowiadający mechanizmowi RPC, i za pomocą interfejsu SIF zwracają dane do układu EE. W ten sposób układ EE może odczytywać informacje z pliku znajdującego się na dysku DVD nie przerywając swojego działania. Kiedy procesor IOP zakończy obsługę żądania odczytu, generuje przerwanie do procesora EE. Dodatkowo, dzięki interfejsowi SIF, procesor IOP może zaplanować wysyłanie danych kontrolera do bufora w układzie EE co okres *Vblank* (*vertical blank* — dla urządzeń działających w systemie NTSC ten czas wynosi 1/60 sekundy, co w przybliżeniu jest równe czasowi wyświetlenia jednej ramki wideo). Układ EE może korzystać z tych danych bez konieczności wysyłania jawnego żądania co każdą ramkę.

Dodatkowe zasoby internetowe

- **Oficjalna strona programu ps2dev**: *www.ps2dev.org* — można w niej znaleźć przewodniki, przykłady kodu, programy ładujące i inne zasoby ułatwiające pisanie oprogramowania dla konsoli PS2.
- **The Third Creation**: *www.thethirdcreation.net* — strona miesięcznych pokazów demo. Można tu znaleźć doskonałe demonstracje. Wszystkie są napisane za pomocą oprogramowania open source. Gorąco polecam pobranie programów pisanych przez programistę o pseudonimie *adresd* — należą do najlepszych.

- **PlayStation 2 Linux Kit**: *www.playstation2-linux.com* — oficjalna strona zestawu PlayStation 2 Linux Kit firmy Sony — pakietu sprzętu i oprogramowania umożliwiającego korzystanie z systemu Linux na konsoli PS2.
- **Witryna WWW Dana Peori'ego**: *www.oopo.net/consoledev* — przykłady kodu i przewodniki, między innymi dotyczące grafiki oraz programowania interfejsu SIF.
- **Witryna WWW** Lukasza Brunna: *www.mouthshut.net* — strona macierzysta *libITO* — jednej z pierwszych bibliotek graficznych dostępnych dla programistów PS2.