POLITECHNIKA ŚLĄSKA

WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

KIERUNEK INFORMATYKA

Projekt inżynierski

Adaptacja systemu operacyjnego Rockbox na odtwarzacz Sandisk Sansa Connect

Autor: Tomasz Moń

Kierujący pracą: dr inż. Wojciech Mielczarek

Gliwice, styczeń 2012

Spis treści

Wstęp

Przenośny odtwarzacz muzyki Sandisk Sansa Connect został wprowadzony na rynek w 2007 roku. Zaletą urządzenia było wyposażenie go w port WiFi i odpowiednie oprogramowanie umożliwiające odtwarzanie muzyki z serwisu Yahoo! Music, polecanie utworów znajomym poprzez komunikator Yahoo! Messenger oraz przeglądanie zdjęć z serwisu społecznościowego Flickr.

Rok po premierze urządzenia, usługi firmy Yahoo! przestały być dostępne dla użytkowników odtwarzacza, co spowodowało całkowite usunięcie obsługi WiFi przez producenta (aktualizacja firmware z 2010 roku).

Cel pracy

Celem projektu jest opracowanie metody uruchomienia oprogramowania nieautoryzowanego przez producenta, wymagającej możliwie jak najmniejszej ingerencji sprzętowej, opartej na wykorzystaniu pól testowych pozostawionych na obwodzie drukowanym urządzenia.

Zakres pracy

Do realizacji projektu konieczne jest przeprowadzenie dokładnej analizy sposobu działania odtwarzacza. W ramach analizy sprzętowej można wyróżnić następujące elementy:

- identyfikacja wykorzystanych układów scalonych,
- określenie funkcji pełnionych przez poszczególne układy,
- ustalenie sposobu komunikacji poszczególnych składowych systemu,
- identyfikacja punktów testowych.

Projekt zakłada uzyskanie działającej wersji systemu operacyjnego Rockbox na odtwarzaczu Sandisk Sansa Connect, umożliwiającej odtwarzanie plików muzycznych. Ze względu na brak obsługi stosu TCP/IP w systemie Rockbox, niniejszy projekt nie obejmuje implementacji obsługi sieci bezprzewodowej WiFi.

Struktura dokumentu

Praca składa się z 5 rozdziałów oraz 3 dodatków.

Rozdział 1 zawiera podstawowe informacje z zakresu inżynierii wstecznej. W rozdziale 2 poruszono problem uruchamiania w systemach wbudowanych (ang. embedded) oprogramowania nieautoryzowanego przez producenta. Rozdział 3 jest opisem analizowanego urządzenia oraz modyfikacji sprzętowej umożliwiającej uruchomienie nieautoryzowanego oprogramowania wewnętrznego (ang. firmware). W rozdziale 4 przedstawiono tok postępowania prowadzącego do uruchomienia systemu Rockbox na odtwarzaczu Sandisk Sansa Connect. Rozdział 5 stanowi podsumowanie niniejszej pracy. W dodatku A opisano próby ominięcia sprawdzania autentyczności oprogramowania. W dodatku B znajduje się plik konfiguracyjny pakietu OpenOCD umożliwiający wykorzystanie modułu JTAG w odtwarzaczu Sandisk Sansa Connect. Dodatek C przedstawia nazwy i rozmieszczenie przycisków odtwarzacza.

1 Inżynieria wsteczna

Zadaniem inżynierii wstecznej ("odwrotnej") jest ustalenie w jaki sposób działa określony produkt: oprogramowanie lub urządzenie. W przypadku systemów komputerowych zastosowanie techniki odwracania pozwala na odzyskanie utraconej dokumentacji, bądź też opracowanie alternatywnych rozwiązań. Praktyka ta jest często stosowana przez twórców wolnego i otwartego oprogramowania celem zapewnienia współpracy z rozwiązaniami, do których nie ma publicznie dostępnej specyfikacji technicznej.

Spośród dostępnych metod inżynierii wstecznej, w przypadku systemów komputerowych można wyróżnić: deasemblacje (ang. disassembly), śledzenie wywołań funkcji oraz podgląd danych przesyłanych na magistrali (ang. sniffing).

1.1 Deasemblacja

Mając kod wykonywalny programu komputerowego, można dokonać próby odtworzenia kodu źródłowego. W systemie operacyjnym Linux standardowo wykorzystuje się format plików wykonywalnych ELF (ang. Executable and Linkable Format). Korzystając z odpowiednich narzędzi (np. objdump wchodzącego w skład GNU Binutils[1] lub IDA Pro[2], będącego standardem na rynku) można uzyskać kod programu w języku symbolicznym procesora, na którym działa dana aplikacja.

Standardowy format plików wykonywalnych ułatwia zadanie, ponieważ od razu można zidentyfikować poszczególne funkcje programu. Niestety, zazwyczaj nie są zachowane nazwy funkcji, ani też nazwy przekazywanych do nich parametrów. Rozpowszechniane pliki wykonywalne na ogół nie zawierają wymienionych informacji, ponieważ nie jest to niezbędne do działania programu, a ponadto mogłoby ułatwić analizę programu. Zaletą usunięcia nazw funkcji i parametrów jest natomiast mniejszy rozmiar pliku wykonywalnego. W przypadku bibliotek współdzielonych (w zależności od systemu operacyjnego mogą to być pliki .so bądź też .dll), często funkcje są udostępniane poprzez nazwę, co automatycznie powoduje, że jest ona znana dla osoby przeprowadzającej proces inżynierii wstecznej.

Metoda ta pozwala na zorientowanie się w jaki sposób działa program, jakkolwiek jest to zadanie żmudne, wymagające sporej wiedzy od osoby przeprowadzającej analizę (często analizowany kod był pisany w języku wysokiego poziomu, np. C, a kompilatory produkują kod, który jest przeznaczony dla docelowego procesora, a nie do analizy przez człowieka). Z powodu dużego stopnia złożoności, tylko nieliczne części systemu są poddawane szczegółowej analizie.

1.2 Śledzenie wywołań funkcji

Powszechnie wykorzystywane systemy operacyjne dostarczają różne biblioteki programowe, w tym bibliotekę standardową języka C. W przypadku inżynierii wstecznej zdarzają się sytuacje, w których kluczowe są parametry przekazywane do odpowiednich funkcji. Przykładem zastosowania śledzenia wywołań funkcji może być chęć opracowania sterownika dla jakiegoś urządzenia (np. procesor dźwięku, który może być sterowany za pomocą programu komputerowego).

Na podstawie wstępnej deasemblacji można ustalić, jakie zewnętrzne funkcje są wywoływane przez dany program oraz jakiego typu parametry są do nich przekazywane (w przypadku wywołań funkcji systemowych, bądź też ogólnodostępnych bibliotek, ustalenie parametrów nie wymaga dużego nakładu pracy).

W uniksowych systamch wykorzystujących format plików wykonywalnych ELF, korzystając ze zmiennej środowiskowej `LD_PRELOAD` można "podmienić" kod dowolnej funkcji wywoływanej przez program. Metoda ta jest wykorzystywana przez niektóre narzędzia służące do debugowania alokacji pamięci (np. malloctools[8]) poprzez podmenienie wywołań funkcji malloc(), calloc(), realloc() oraz free() na wersje umożliwiające tworzenie raportów dotyczących dokonanych alokacji.

1.3 Podgląd danych na magistrali

W systemach komputerowych występuje wiele interfejsów. Spośród interfejsów przeznaczonych dla urządzeń peryferyjnych do najpopularniejszych zaliczyć można USB, FireWire oraz RS-232. W systemach wbudowanych konieczna jest komunikacja pomiędzy różnymi układami scalonymi. Standardowe interfejsy wykorzystywane w systemach klasy embedded najczęściej oparte są na I^2C, SPI, UART. Podglądu danych na magistrali dokonuje się za pomocą odpowiednich urządzeń, takich jak np. USB Explorer 200[3], bądź też Bus Pirate[4], które przechwytują dane przesyłane pomiędzy elementami systemu. Na podstawie tych danych próbuje się następnie określić format komunikatu.

2 Problem uruchamiania nieautoryzowanego oprogramowania w systemach wbudowanych

Producenci sprzętu elektronicznego, poza nielicznymi wyjątkami, nie pozwalają użytkownikowi na ingerencję w strukturę oprogramowania wewnętrznego. Praktyka ta może być podyktowana ochroną własności intelektualnej osób trzecich wykorzystanej w systemie (przykładem może być implementacja cyfrowego zarządzania prawami DRM), nakładająca na producenta ograniczenia dotyczące dystrybucji wersji źródłowej oprogramowania.

W celu uruchomienia nieautoryzowanego oprogramowania w systemie wbudowanym należy ustalić, w jaki sposób do urządzenia została wgrana początkowa wersja oprogramowania, jak również sprawdzić, czy urządzenie posiada opcję aktualizacji oprogramowania. Można tu spotkać rozwiązania wykorzystujące programowanie wewnątrz systemu (ang. in-system programming, na przykład z wykorzystaniem interfejsu JTAG), a także przeprogramowywanie pamięci typu flash w trakcie procesu produkcyjnego urządzenia.

2.1 Metody aktualizacji oprogramowania

Urządzenia elektroniczne, takie jak przenośnie odtwarzacze muzyki, telefony komórkowe czy też konsole do gier, często posiadają możliwość aktualizacji oprogramowania wewnętrznego. Takie rozwiązanie pozwala na skrócenie czasu jaki upływa pomiędzy rozpoczęciem pracy nad produktem a jego wprowadzeniem na rynek (ang. time to market), ponieważ początkowa wersja firmware, obsługująca jedynie podstawowe funkcje, może później zostać rozbudowana poprzez aktualizacje (np. dodanie obsługi kart SDHC itp.).

Uruchomienie nieautoryzowanego oprogramowania na urządzeniu wbudowanym często polega na przygotowaniu pliku, który zostanie potraktowany jako aktualizacja. W zależności od wykorzystywanego formatu to zadanie może mieć różny stopień trudności: w przypadku prostego sprawdzania poprawności pliku aktualizacji (takiego jak suma kontrolna) całość sprowadza się jedynie do ustalenia formatu pliku, jednakże w przypadku, gdy producent zastosował procedury kryptograficzne, w procesie aktualizacji konieczne jest opracowanie metody ominięcia sprawdzania podpisu cyfrowego, bądź też znalezienie błędów w implementacji.

Należy mieć na uwadze, że wgranie błędnego firmware może sprawić, że dane urządzenie elektroniczne będzie bezużyteczne (ang. brick). Część produktów posiadających funkcję aktualizacji oprogramowania wewnętrznego jest wyposażona w tryb odzyskiwania (ang. recovery mode) po nieudanej wymianie oprogramowania. Istnieją jednak

przypadki, gdy taka funkcjonalność nie jest dostępna (głównie w przypadku telefonów komórkowych lub konsol do gier) i w sytuacji nieudanej aktualizacji konieczna jest wizyta w serwisie celem dokonania naprawy.

2.1.1 Pamięć zewnętrzna

W przypadku urządzeń, które po podłączeniu do komputera działają jak pamięć masowa (np. odtwarzacze muzyki w trybie UMS (ang. USB mass storage)), część producentów korzysta z metody aktualizacji firmware polegającej na wgraniu pliku z aktualizacją do odpowiedniego katalogu. Przy każdym uruchomieniu oprogramowania wewnętrznego sprawdzana jest dostępność pliku aktualizacji, a jeżeli się taki znajdzie, to wyświetlany jest odpowiedni komunikat dla użytkownika (na przykład informujący o konieczności podłączenia zewnętrznego zasilacza do urządzenia celem rozpoczęcia procesu aktualizacji). Taki sposób aktualizacji jest wykorzystywany w odtwarzaczu iAudio X5[7] firmy Cowon.

2.1.2 USB DFU

Urządzenie wyposażone w interfejs USB może zostać zaktualizowane z wykorzystaniem mechanizmu DFU[5] (ang. Device Firmware Upgrade) o ile oczywiście producent zaimplementował taką funkcjonalność. Ponieważ w trakcie aktualizacji oprogramowania wewnętrznego urządzenie nie może być wykorzystywane w normalny sposób, wymagana jest możliwość przełączenia urządzenia w tryb DFU. Aby przełączyć tryb konieczna jest interwencja bądź to operatora (przykładowo uruchomienie urządzenia z przytrzymaną odpowiednią kombinacją przycisków), bądź też systemu operacyjnego wykorzystanego w urządzeniu.

Należy zwrócić uwagę, że klasa DFU jedynie definiuje sposób w jaki obraz aktualizacji jest przesyłany do urządzenia i nie nakłada żadnych ograniczeń na zawartość przekazywanego pliku, poza wymaganiem przyrostka zawierającego m.in. sumę CRC całego pliku oraz identyfikatory VendorID oraz ProductID urządzenia. Zawartość pliku jest w całości zależna od producenta, co wydaje się sensowne, ponieważ różne urządzenia działają często w całkowicie odmienny sposób.

2.1.3 Inne metody aktualizacji oprogramowania

Oprócz wyżej wymienionych sposobów aktualizacji oprogramowania istnieje także szereg innych rozwiązań. W przypadku ruterów popularną metodą aktualizacji jest wgranie pliku poprzez interfejs webowy. W przypadku konsol do gier aktualizacje oprogramowania wewnętrznego mogą być dostarczone na nośnikach z grami (w ten sposób

można dokonać aktualizacji firmware na konsoli Nintendo Wii), bądź też mogą zostać pobrane bezpośrednio przez konsolę poprzez sieć internet (wymaga skonfigurowania odpowiedniego połączenia, często bezprzewodowego - również obsługiwane przez Nintendo Wii).

2.2 Interfejs JTAG (IEEE 1149.1)

Standard IEEE 1149.1 definiuje logikę, która musi zostać załączona w układzie scalonym w celu udostępnienia następujących funkcjonalności[6]:

- testowanie połączeń pomiędzy układami scalonymi po osadzeniu ich na obwodzie drukowanym,
- testowanie danego układu scalonego,
- obserwowanie/modyfikacja stanu układu podczas normalnego działania.

Interfejs JTAG może zostać wykorzystany do zaprogramowania pamięci nieulotnych znajdujących się w systemie cyfrowym, na przykład na drodze skanowania granicznego (ang. boundary scan), które pozwala na dowolne sterowanie liniami wejścia/wyjścia układu scalonego. W tym przypadku konieczne jest posiadanie pliku BSDL (ang. boundary scan description language), zawierającego opis poszczególnych wyprowadzeń oraz rejestrów wykorzystywanych podczas skanowania granicznego. Jeżeli poprzez interfejs JTAG możliwy jest dostęp do pamięci operacyjnej (umożliwiają to na przykład układy zawierające rdzeń ARM), to można zaprogramować pamięć nieulotną poprzez uruchomienie w procesorze kodu, który ją zaprogramuje (pod warunkiem, że dany układ jest w stanie pobrać instrukcje z pamięci RAM).

Pozostawienie wyprowadzeń interfejsu JTAG na obwodzie drukowanym urządzenia może być wykorzystane w celu wprowadzenia nieautoryzowanych zmian do oprogramowania wewnętrznego. Poszczególne linie interfejsu zazwyczaj nie są oznaczone w produkcie konsumenckim. W celu sprawdzenia, czy dany punkt testowy odpowiada linii interfejsu JTAG można prześledzić połączenia w obwodzie drukowanym. Najprostszym sposobem dokonania tego jest wylutowanie układu scalonego wyposażonego w logikę testową (JTAG), a następnie sprawdzenie połączeń za pomocą testera ciągłości poszczególnych wyprowadzeń układu (korzystając z dokumentacji technicznej można ustalić funkcję danego wyprowadzenia).

3 Analiza odtwarzacza Sandisk Sansa Connect

Celem zaadaptowania systemu operacyjnego na określone urządzenie, konieczne jest posiadanie podstawowej wiedzy, jak urządzenie jest zbudowane. Ponieważ w przypadku odtwarzacza Sandisk Sansa Connect informacje te nie są publicznie dostępne, posłużono się metodami inżynierii wstecznej. W pierwszej kolejności zostały zidentyfikowane wykorzystane układy scalone. Następnym krokiem była analiza procesu odzyskiwania oprogramowania (w ten sposób można dokonać aktualizacji firmware).

3.1 Budowa odtwarzacza

Po otwarciu obudowy odtwarzacza oraz zdjęciu metalowej ramki pełniącej funkcję ekranu przystąpiono do identyfikacji układów scalonych na podstawie ich oznaczeń. W celu określenia przeznaczenia poszczególnych komponentów korzystano ze stron internetowych producentów układów. Określenie funkcji pełnionych w systemie przez poszczególne układy scalone nastąpiło na podstawie analizy oprogramowania wewnętrznego (patrz: 3.2).

Zidentyfikowane zostały następujące układy scalone:

- Texas Instruments TMS320DM320
  Mikrokontroler zawierający procesor ARM926EJS (zwany dalej procesorem głównym) oraz procesor sygnałowy TMS320vc5409. Producent nie udostępnia dokumentacji technicznej dla tego układu.

- Texas Instruments TPS65021
  Układ zarządzania zasilaniem, przeznaczony dla urządzeń wymagających kilku linii zasilania oraz czerpiących energię z jednego ogniwa litowo-jonowego, bądź też litowo-polimerowego. Komponent ten zawiera trzy przetwornice napięciowe typu step-down, przystosowane do zasilania procesora, urządzeń peryferyjnych, układów wejścia/wyjścia oraz pamięci operacyjnej. Komunikacja z głównym procesorem odbywa się poprzez magistralę I^2C.

- Texas Instruments TNETV105PAP
  Moduł odpowiadający za obsługę interfejsu USB 2.0. Producent nie udostępnia dokumentacji technicznej dla tego układu. Komunikacja z głównym procesorem odbywa się za pomocą magistrali VLYNQ.

- Texas Instruments TLV320AIC3106
  Kodek audio, zawierający przetworniki: cyfrowo-analogowy oraz analogowo-cyfrowy. Obsługuje kilka różnych metod przesyłu danych audio. Komunikacja z głównym

procesorem odbywa się za pomocą magistrali I^2C. Dane audio są przekazywane poprzez interfejs szeregowy procesora sygnałowego.

- Texas Instruments TPS61042
  Sterownik diod świecących. Oznaczenie znajdujące się na układzie to BHS.

- Atmel Mega165PV
  Ośmiobitowy mikrokontroler z rodziny AVR, komunikujący się z procesorem głównym za pomocą magistrali SPI, odpowiedzialny za obsługę przycisków, monitorowanie poziomu naładowania baterii, reset kodeka audio oraz za załączenie/wyłączenie wyświetlacza. Ponadto, układ ten może pełnić funkcję strażnika systemu (ang. watchdog).

- Atmel AT88SC
  Pamięć typu EEPROM zabezpieczona kryptograficznie. W pamięci przechowywany jest identyfikator urządzenia oraz adres MAC modułu WiFi. Komunikacja pamięci z procesorem głównym następuje poprzez magistralę I^2C.

- Sandisk SDINB1-4096
  Pamięć typu NAND flash o pojemności 4 GiB, zawierająca wbudowany kontroler, który umożliwia komunikację z pamięcią za pomocą interfejsu SD. Układ przechowuje pliki muzyczne oraz część systemu operacyjnego odpowiedzialną za interfejs użytkownika.

- Samsung K4M51323LC-DN75
  Pamięć SDRAM, zawierająca 4 banki pamięci w organizacji 4 MiB x 32 bit.

- Samsung K8D3216UBC
  Pamięć typu NOR flash o pojemności 4 MiB. W pamięci tej zapisane są: program rozruchowy (ang. bootloader), parametry programu rozruchowego, jądro systemu operacyjnego (Linux) oraz podstawowe aplikacje systemowe (initrd).

- Murata WIFI
  Moduł odpowiedzialny za komunikację bezprzewodową w standardzie 802.11g, wyposażony w chipset MARVELL W8686B12. Komunikacja modułu z procesorem głównym odbywa się za pomocą interfejsu SPI.

- ST M41T62
  Zegar czasu rzeczywistego z funkcją alarmu. Oprogramowanie wewnętrzne dostarczane przez producenta nie komunikuje się bezpośrednio z tym układem.

3.2 Funkcje i działanie oprogramowania firmowego

Oryginalne oprogramowanie pozwala na odtwarzanie plików muzycznych w formatach: MP3, AAC, WMA, bezpieczne WMA (ang. secure WMA) oraz plików wideo w formacie MPEG-4. Ponadto możliwe jest przeglądanie zdjęć w formatach PNG oraz JPEG. Po wprowadzeniu urządzenia na rynek, urządzenie posiadało obsługę serwisu Yahoo! Music, komunikatora internetowego Yahoo! Messenger oraz serwisu społecznościowego Flickr[9]. W połowie 2008 roku usługi Yahoo! Music oraz Yahoo! Messenger przestały być dostępne[10]. Wraz z aktualizacją oprogramowania wewnętrznego do wersji 1.2 obsługa WiFi została całkowicie usunięta (tzw. wersja No-WiFi)[11]. Ponadto, starsze wersje oprogramowania, po połączeniu z punktem dostępowym sieci WiFi sprawdzają dostępność połączenia z internetem poprzez wysłanie zapytania do serwera sandisk.ping.zing.net, który aktualnie jest wyłączony (co skutkuje poinformowaniem użytkownika o problemie w uzyskaniu połączenia z siecią).

Po połączeniu z internetem poprzez sieć WiFi, urządzenie ma uruchomioną usługę na porcie 8088. Przeznaczenie, jak również sposób działania usługi jest nieznany. Po wysłaniu pseudolosowych danych (wygenerowanych z /dev/random) na dany port możliwe jest wywołanie błędu naruszenia ochrony pamięci, co skutkuje wypisaniem stosownego komunikatu na wyświetlaczu urządzenia.

W urządzeniu znajdują się dwa procesory: główny, oparty na rdzeniu ARM oraz pomocniczny, oparty na rdzeniu AVR. Procesor pomocniczny, w tym projekcie inżynierskim został potraktowany jako "czarna skrzynka" - nie przeprowadzono deasemblacji kodu programu działającego w układzie, ani też nie przeprowadzono dokładnej analizy połączeń z innymi układami. Procesor główny ma możliwość zaprogramowania układu AVR, jednakże uznano, że chcąc umożliwić wybór systemu operacyjnego przy starcie (ang. dual-boot), oprogramowanie tego układu należy pozostawić niezmienione.

Oryginalne oprogramowanie dostarczane przez producenta sprzętu jest oparte na jądrze Linux. Firma Sandisk udostępniła zmodyfikowany kod źródłowy wykorzystanej wersji jądra systemu. Niestety, część funkcjonalności, taka jak obsługa procesora sygnałowego, obsługa dźwięku czy też obsługa sieci bezprzewodowej znajduje się we własnościowych modułach, do których nie jest dostępny kod źródłowy.

3.2.1 Tryb odzyskiwania

Urządzenie wyposażono w tryb odzyskiwania, który może posłużyć do aktualizacji oprogramowania wewnętrznego (początkowo była także możliwość aktualizacji poprzez WiFi). Celem skorzystania z tego trybu należy zainstalować oprogramowanie Sansa Connect Recovery Tool oraz wprowadzić urządzenie w tryb odzyskiwania poprzez naci-

Tabela 1: Struktura pliku .srr

Rozmiar	**Nazwa**	**Opis**
4 bajty	Identyfikator	`0xAA`, `0xBB`, `0xFF`, `0xEE`.
4 bajty	Adres ładowania	Adres (zapisany w formacie Little Endian[1]), pod który mają być załadowane dane.
4 bajty	Punkt startowy	Adres pierwszego rozkazu (Little Endian) do wykonania w przypadku pliku wykonywalnego. Jeżeli plik nie jest wykonywalny, w tym polu zapisana jest wartość `0xFFFFFFFF`.
4 bajty	Rozmiar pliku	Rozmiar pola danych i sygnatury (Little Endian).
Dowolny	Właściwe dane	Dane, które będą ładowane w odpowiednie miejsce pamięci operacyjnej.
2048 bajtów	Sygnatura	Podpis cyfrowy (nagłówka oraz danych). Niewykorzystane bajty wyzerowane.

śnięcie odpowiedniej kombinacji przycisków: NEXT, VOLUME UP oraz POWER, gdy urządzenie jest wyłączone. Proces ten jest podzielony na dwa główne etapy: załadowanie jądra systemu i podstawowego systemu plików oraz wgranie aplikacji użytkowych.

Pierwszy etap wykorzystuje aplikację `zsi_fw.exe` wchodzącą w skład pakietu Recovery Tool. Do urządzenia wysyłane są pliki w formacie .srr (patrz tabela 1) zawierające jądro systemu (vmlinux.srr) oraz podstawowy system plików (initrd.srr). W niektórych wersjach oprogramowania pliki z oprogramowaniem są zaszyfrowane, jednakże do urządzenia przesyłane są odszyfrowane ich wersje.

Pliki przesłane do urządzenia są zapisywane do odpowiednich sektorów pamięci NOR flash. Przy każdym starcie systemu następuje sprawdzenie poprawności podpisu tych modułów, w przypadku wykrycia nieprawidłowości urządzenie kasuje parametry programu rozruchowego oraz automatycznie przechodzi w tryb odzyskiwania. Część aplikacji `zsi_fw.exe` odpowiedzialna za komunikację z urządzeniem została dokładnie przebadana i zaimplementowana w programie `zsitool` napisanym przez autora niniejszego projektu inżynierskiego.

Podczas trwania drugiej fazy procesu odzyskiwania wykorzystywany jest program `zaprecover.exe`. W tym etapie następuje instalacja aplikacji użytkownika (zap.tar.gz). Poza samym oprogramowaniem, do urządzenia przesyłany jest także podpis cyfrowy dla tego pliku (zap.sig).

[1]Najpierw najmłodszy bajt

Aktualizacje firmware są podpisane kluczem kryptograficznym RSA o rozmiarze 2048 bitów, stąd też nie ma prostej metody na wgranie nieautoryzowanego oprogramowania. Z tego powodu konieczne jest opracowanie metody ominięcia procesu weryfikacji źródła firmware.

3.3 Zdalny dostęp

Korzystając z interfejsu JTAG możliwe jest uruchomienie zmodyfikowanej wersji oprogramowania wewnętrznego. Po wprowadzeniu urządzenia w tryb odzyskiwania można za pomocą interfejsu JTAG zmodyfikować kod programu rozruchowego działającego w urządzeniu tak, aby podpis cyfrowy był uznawany zawsze za poprawny, nawet wtedy, gdy nie występuje. Modyfikacje te dotyczą kodu znajdującego się w pamięci RAM. Bootloader po zainicjalizowaniu modułu zarządzania pamięcią (MMU), koprocesora, sygnałów zegarowych, linii wejścia/wyjścia, pamięci RAM oraz portu szeregowego kopiuje się do pamięci operacyjnej pod adres 0x1300180. Zmiany są przedstawione w postaci komend programu OpenOCD (Listing 1 i Listing 2).

Listing 1: Zmiany dla Bootloadera w wersji 24655

```
#Bootloader 24655
#zamien BL 0x13074A0 na MOV R0, #0
mww 0x1301904 0xe3a00000 # (stara wartosc: 0xeb0016e5)
#zamien LDR R12, [SP,#0x568+var55C] na MOV R12, #2
mww 0x1301914 0xe3a0c002 # (stara wartosc: 0xe59dc00c)
#zamien BL 0x1307634 na MOV R0, #0
mww 0x1301ab0 0xe3a00000 # (stara wartosc: 0xeb0016df)
```

Listing 2: Zmiany dla Bootloadera w wersji 49797

```
#Bootloader 49797
#zamien BL 0x1307604 na MOV R0, #0
mww 0x130190c 0xe3a00000 # (stara wartosc: 0xeb00173c)
#zamien LDR R12, [SP,#0x568+var55C] na MOV R12, #2
mww 0x130191c 0xe3a0c002 # (stara wartosc: 0xe59dc00c)
#zamien BL 0x1307798 na MOV R0, #0
mww 0x1301ab8 0xe3a00000 # (stara wartosc: 0xeb001736)
```

W ramach projektu wykonano modyfikacje dla dwóch wersji programu rozruchowego (nie jest znane, czy w urządzeniach dostępnych na rynku obecne są inne wersje bootloadera). Zmiany te zostały opracowane przy wykorzystaniu inżynierii wstecznej i powodują ominięcie wywołania funkcji odpowiedzialnych za:

- sprawdzenie czy podpis cyfrowy jest umieszczony w pliku .srr,

- odszyfrowanie podpisu cyfrowego,
- porównanie obliczonej sumy SHA-1 z wartością zawartą wewnątrz podpisu cyfrowego.

Ominięcia wywołania wymienionych funkcji dokonano przez ustawienie wartości odpowiedniego rejestru oznaczającej poprawne wywołanie.

Po wprowadzeniu zmian przedstawionych na Listingach 1 i 2 z wykorzystaniem oprogramowania OpenOCD, do urządzenia można wprowadzić zmodyfikowaną wersję oprogramowania. Niestety, zmiany te są ulotne, po ponownym uruchomieniu ponownie sprawdzana jest autentyczność oprogramowania. Rozwiązanie tego problemu wymaga zmiany kodu programu rozruchowego opisanego w 3.4.

Plik initrd.srr zawiera skompresowany system plików CRAMFS. Po usunięciu nagłówka (pierwsze 16 bajtów) oraz sygnatury (ostatnie 2048 bajtów) możliwe jest wyodrębnienie plików tam zawartych za pomocą narzędzia cramfsck (wchodzącego w skład pakietu cramfsprogs). W celu rozpakowania plików do katalogu initrd można posłużyć się następującymi poleceniami (Listing 3):

Listing 3: Wyodrębnienie zawartości pliku initrd.srr do katalogu initrd

```
#Usuniecie naglowka oraz sygnatury
dd if=initrd.srr of=initrd.cramfs bs=1 skip=16 \
  count=$((`wc -c initrd.srr | cut -f1 -d' '`-2048-16))

#Wypakowanie plikow
cramfsck -x initrd initrd.cramfs
```

Po wprowadzeniu zmian (np. uruchomienie usługi telnet poprzez odkomentowanie odpowiedniej linii w pliku initrd/etc/inetd.conf) należy utworzyć nowy plik .srr. Można tego dokonać za pomocą następujących poleceń Listing 4):

Listing 4: Utworzenie pliku initrd-modified.srr na podstawie katalogu initrd

```
#Utworzenie nowego obrazu systmeu plikow
mkcramfs initrd initrd-modified.cramfs

#Pierwsze 12 bajtow naglowka
printf "\xAA\xBB\xFF\xEE\x20\x00\x40\x04\xFF\xFF\xFF\xFF" \
  > initrd-modified.srr

#Ostatnie 4 bajty naglowka - rozmiar danych i sygnatury (LE)
echo `wc -c initrd-modified.cramfs` | awk '{x=$1+2048; \
  printf "%c%c%c%c", x%256, (x/256)%256, (x/(256*256))%256, \
  (x/(256*256*256))%256}' >> initrd-modified.srr
```

```
#Dodanie wlasciwych danych
cat initrd-modified.cramfs >> initrd-modified.srr

#Dodanie pustej sygnatury
dd if=/dev/zero of=sygnatura bs=2048 count=1
cat sygnatura >> initrd-modified.srr
```

3.4 Uruchomienie własnej wersji oprogramowania

Z powodu wysokiej złożoności ataku siłowego, polegającego na wygenerowaniu fałszywego podpisu, który zostanie uznany za poprawny, rozważono opracowanie metody pozwalającej na całkowite ominięcie weryfikacji autentyczności oprogramowania. Podjęto liczne próby ominięcia weryfikacji autentyczności oprogramowania (patrz dodatek A), część z nich zakończyła się porażką, aczkolwiek udało się opracować metodę polegającą na zmianie programu rozruchowego. Z powodu zastosowania w odtwarzaczu Sandisk Sansa Connect sprzętowego zabezpieczenia przed zapisem sektorów pamięci flash przechowujących program rozruchowy, konieczna stała się ingerencja sprzętowa.

Po wylutowaniu wszystkich układów scalonych z obwodu drukowanego przystąpiono do ustalenia funkcji poszczególnych pól testowych. W rezultacie znalezione zostały wyprowadzenia interfejsu JTAG oraz portu szeregowego. Ponadto, odszukano ścieżkę łączącą wyprowadzenie $\overline{WP}/ACC$ pamięci NOR flash z zasilaniem. Po podaniu napięcia z przedziału od 8.5V do 12.5V na tą linię, urządzenie przechodzi w tryb przyśpieszonego programowania, w którym z wszystkich sektorów tymczasowo zdejmowana jest blokada przed zapisem.

3.4.1 Przygotowania do zapisu programu rozruchowego

Po uzyskaniu zdalnego dostępu, tak jak to opisano w punkcie 3.3, zgrano na włożoną do urządzenia kartę microSD kod programu rozruchowego za pomocą polecenia `cat /dev/mtd0 > /mnt/mmc/bootloader.bin`. Następnie, korzystając z edytora heksadecymalnego wxHexEditor wprowadzono odpowiednie zmiany wewnątrz pliku `bootloader.bin`, opierając się, w zależności od wersji programu rozruchowego, na listingu 1 albo 2. W trakcie wykonywania tego kodu procesor ARM działa w trybie Little Endian. W tabelach 2 oraz 3 znajduje się podsumowanie dokonanych zmian w zależności od wersji bootloadera. Po wprowadzeniu zmian, dane zapisano pod nazwą `bootloader-patched.bin`.

W celu umożliwienia zapisania zmodyfikowanej wersji programu rozruchowego w pamięci flash, konieczne było zdjęcie blokady przed zapisem i zaprogramowanie układu (posłużono się metodą akcelerowanego programowania). Najpierw za pomocą cieńkiego

Tabela 2: Zmiany pliku bootloader.bin dla wersji 24655

Offset	**Oryginalne dane**	**Zmienione dane**	**Oryginalna instrukcja**	**Zmieniona instrukcja**
0x1784	E5 16 00 EB	00 00 A0 E3	BL 0x13074A0	MOV R0, #0
0x1794	0C C0 9D E5	02 C0 A0 E3	LDR R12, [SP,#0x568+var55C]	MOV R12, #2
0x1930	DF 16 00 EB	00 00 A0 E3	BL 0x1307634	MOV R0, #0

Tabela 3: Zmiany pliku bootloader.bin dla wersji 49797

Offset	**Oryginalne dane**	**Zmienione dane**	**Oryginalna instrukcja**	**Zmieniona instrukcja**
0x178C	3C 17 00 EB	00 00 A0 E3	BL 0x1307604	MOV R0, #0
0x179C	0C C0 9D E5	02 C0 A0 E3	LDR R12, [SP,#0x568+var55C]	MOV R12, #2
0x1938	36 17 00 EB	00 00 A0 E3	BL 0x1307798	MOV R0, #0

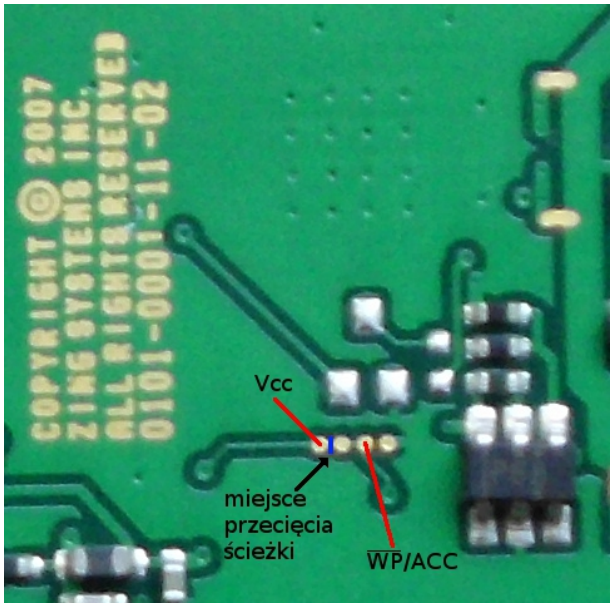
a zarazem ostrego narzędzia przerwano ścieżkę łączącą $\overline{WP}/ACC$ z V_{CC}. Na rysunku 1 miejsce przecięcia oznaczono niebieską linią. Kontakty $\overline{WP}/ACC$ oraz GND pamięci flash wyprowadzono poza obudowę układu za pomocą przewodów (rysunek 2).

3.4.2 Programowanie pamięci flash

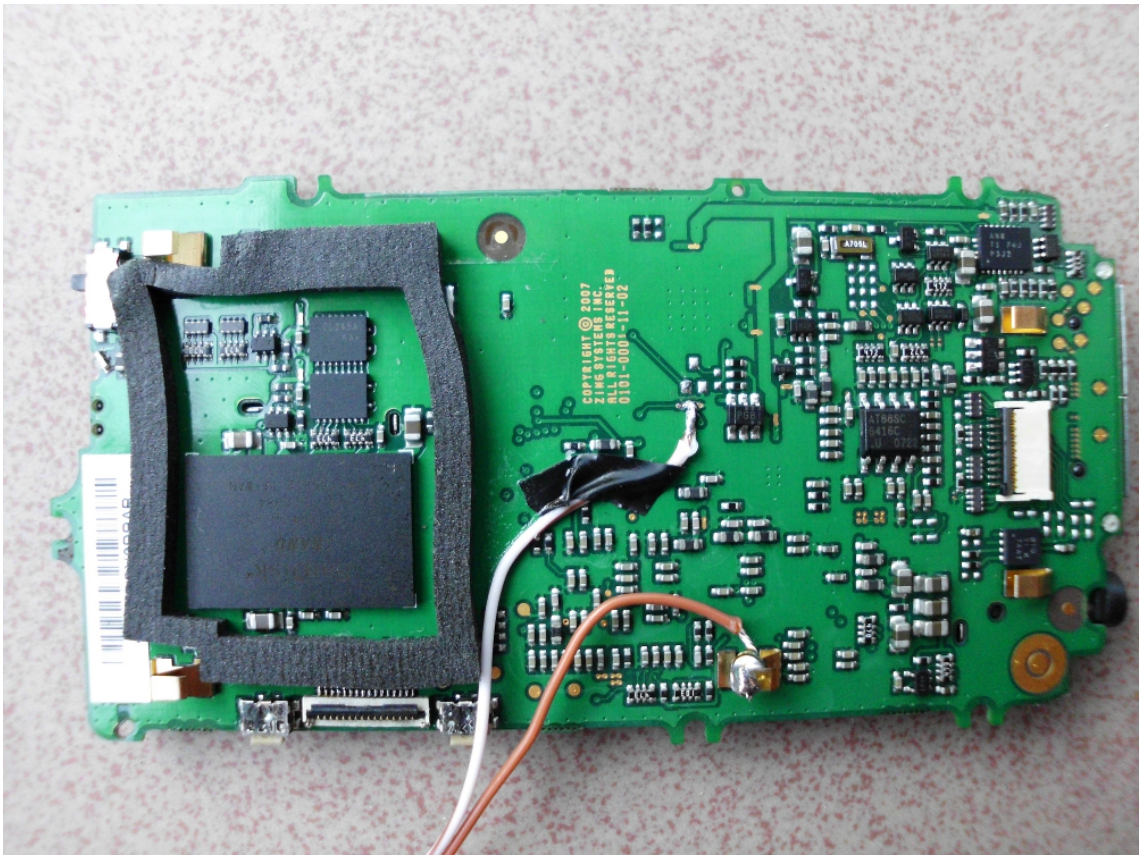
Do wyprowadzeń $\overline{WP}/ACC$ i GND pamięci flash (sposób udostępnienia tych wyprowadzeń został opisany w punkcie 3.4.1) podłączono zasilacz laboratoryjny HP E3631A (rysunek 4). Podczas startu systemu napięcie wyjściowe z zasilacza ustawiono na 0V. Po zalogowaniu się do urządzenia, zwiększono napięcie na wyjściu zasilacza do 9V (rysunek 3), aby wprowadzić pamięć NOR flash w tryb przyśpieszonego programowania.

"Wgrania" programu rozruchowego dokonano za pomocą polecenia `flash_eraseall /dev/mtd0;cat /mnt/mmc/bootloader-patched.bin > /dev/mtd0` Po zakończeniu procesu programowania pamięci, na zasilaczu laboratoryjnym ustawiono napięcie 0V, a następnie sprawdzono czy dane zostały poprawnie zapisane poprzez zgranie zawartości `/dev/mtd0` i porównanie sumy md5 otrzymanego pliku z sumą md5 pliku `bootloader-patched.bin`

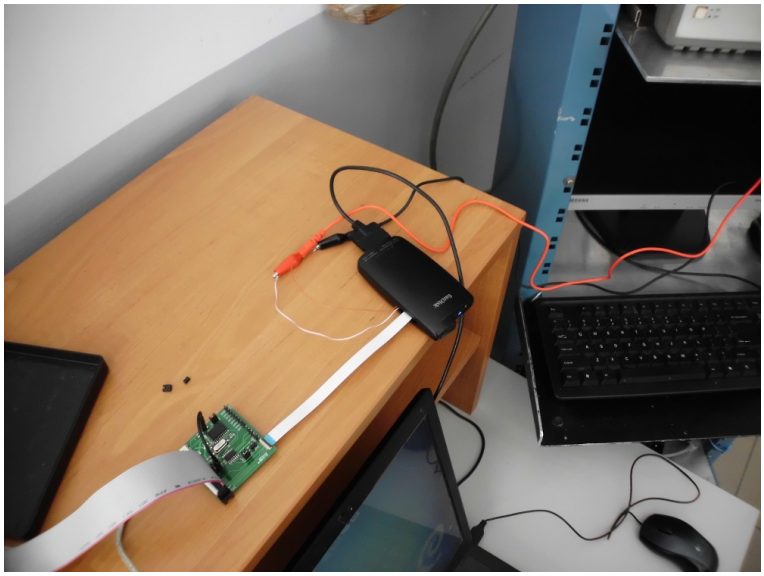
Po upewnieniu się, że wszystko działa poprawnie, wyłączono odtwarzacz oraz rozlutowano dołączone przewody. Połączenie $\overline{WP}/ACC$ z V_{CC} zostało odtworzone z wykorzystaniem małego kawałka drucika miedzianego.



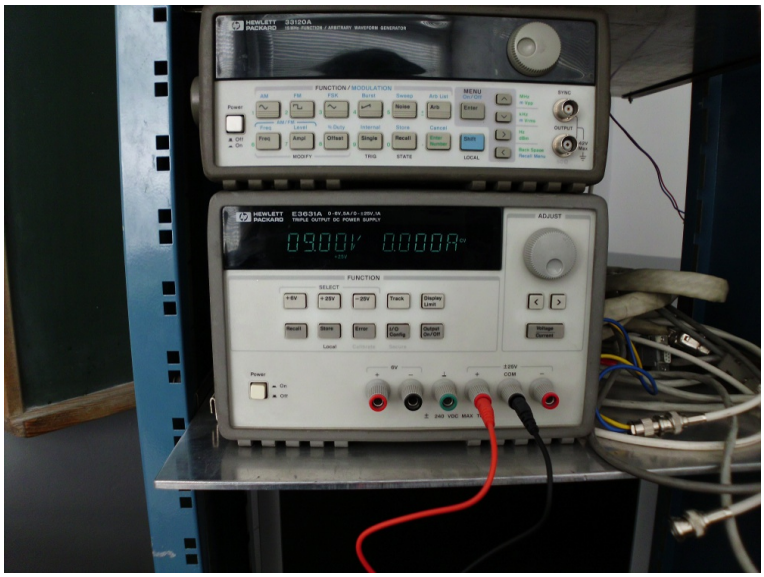
Rysunek 1: Połączenie $\overline{WP}/ACC$ z V_{CC}. Miejsce w którym przerwano ścieżkę oznaczono kolorem niebieskim.



Rysunek 2: Przewody służące do wyprowadzenia sygnałów $\overline{WP}/ACC$ oraz GND poza obudowę urządzenia. Taśmę samoprzylepną zastosowano celem zmniejszenia sił działających na miejsce lutowania.



Rysunek 3: Sygnały $\overline{WP}/ACC$ (czerwony) oraz GND (czarny) połączone z zasilaczem laboratoryjnym.



Rysunek 4: Napięcie $9V$ wykorzystane w procesie przeprogramowania programu rozruchowego.

4 Rockbox

Projekt Rockbox został zapoczątkowany w grudniu 2001 roku w intencji ulepszenia oprogramowania wewnętrznego odtwarzacza Archos Player. Grupa znajomych, zawodowo związanych z systemami wbudowanymi, rozpoczęła prace nad projektem alternatywnego oprogramowania, zastanawiając się jak skomplikowane może być to zadanie [13]. Po wykonaniu działającego zamiennika firmware, dodana została obsługa urządzenia Archos Recorder. Ponieważ projekt został wydany na otwartej licencji (GPLv2), zainteresowani z całego świata przesyłali swoje udoskonalenia oraz uwagi. W 2003 roku Rockbox obsługiwał również Archos FM Recorder. Obsługa urządzenia iRiver H100 została dodana w 2004 roku. Odtwarzacz ten różnił się od ówcześnie wspieranych przez system Rockbox sposobem realizacji dekodowania plików muzycznych, które w całości wykonywano w procesorze (ang. software codec), bez użycia specjalizowanych układów dekodujących. W 2005 roku pojawiła się obsługa odtwarzacza posiadającego kolorowy wyświetlacz (iRiver H320). W roku 2006 dodano obsługę odtwarzaczy iPod (modele: Nano, 5G, Mini 2G, 3G), kolejnych modeli iRiver (iFP, H10), iAudio X5, oraz Gigabeat F/X. Od tego czasu, każdego roku zwiększała się liczba wspieranych urządzeń.

Obecnie Rockbox działa natywnie na wielu urządzeniach (ponad 70 różnych modeli, działających w oparciu o architektury SH1, m68k, ARM oraz MIPS). W roku 2010 rozpoczęte zostały prace mające na celu uruchomienie projektu Rockbox jako aplikacji (ang. Rockbox as an Application) działającej pod kontrolą innych systemów operacyjnych (Android, iOS, MeeGo). Należy zwrócić uwagę, że Rockbox nie jest dystrybucją systemu Linux[12], lecz kompletnym, otwartym, systemem operacyjnym.

4.1 Zalety systemu Rockbox

System Rockbox przewyższa możliwościami oprogramowanie dostarczane przez producentów urządzeń.

Do kluczowych cech systemu [14] zaliczyć można między innymi[2]:

- Obsługę ponad 20 formatów audio.
  Wsród odtwarzanych typów plików znajdują się zarówno formaty stratne (m.in. MP3, Ogg/vorbis, MPC, AC3, WMA, AAC), bezstratne (m.in. WAV, FLAC, Wavpack, Shorten, TTA), jak również inne (m.in. SID, MIDI, MOD, SPC).
- Odtwarzanie bez przerw pomiędzy utworami (ang. gapless playback).
- 5-pasmowy equalizer (korektor graficzny).

[2]Lista nie dotyczy odtwarzaczy firmy Archos, które posiadają sprzętowe dekodowanie (ang. hardware codec) plików muzycznych.

- Regulację głośności cechującą się wysoką rozdzielczością.
  Oryginalne oprogramowanie w niektórych odtwarzaczach udostępnia małą liczbę poziomów głosności (np. w oryginalnym firmware Sandisk e200 dostępne są dwie opcje głośności - normalna oraz wysoka). System Rockbox pozwala na wielostopniową regulację głośności (w zależności od modelu od 64 do 100 różnych poziomów głosności, wyrażanych w decybelach).

- Możliwość zmiany wyglądu interfejsu użytkownika (tzw. obsługa motywów).

- Płynne przejścia pomiędzy utworami (ang. crossfade).
  Rockbox posiada zaawansowany crossfader. Możliwe jest także całkowite wyłączenie tego efektu.

- Normalizację głośności (ang. ReplayGain[15]).

- Obsługę serwisu społecznościowego Last.fm.
  W przypadku załączenia tej opcji, po podłączeniu odtwarzacza do komputera można za pomocą odpowiedniego oprogramowania (np. QTScrobbler[16]) uaktualinić listę odtworzonych utworów w serwisie Last.fm.

- Obsługę wtyczek (gry, aplikacje, dema).
  W systemie Rockbox dostępne są różne gry (m.in. Doom), aplikacje (m.in. metronom, edytor tekstu, kalendarz) oraz dema (m.in. spektogram odtwarzanego utworu).

4.2 Adaptacja systemu Rockbox na odtwarzacz

Celem uruchomienia systemu operacyjnego Rockbox na określonym urządzeniu, konieczne jest przygotowanie programu rozruchowego, posiadającego podstawową obsługę interfejsu użytkownika (przyciski, wyświetlacz) oraz pamięci masowej (z której ładowany jest właściwy system). Pożądaną cechą jest pozostawienie możliwości załadowania oryginalnego oprogramowania po naciśnięciu określonego przycisku w początkowej fazie uruchamiania systemu (w przypadku odtwarzacza Sansa Connect przyjęty został przycisk PREVIOUS).

Odtwarzacz Sansa Connect jest oparty na mikrokontrolerze TMS320DM320 firmy Texas Instruments. Przed rozpoczęciem pracy nad niniejszym projektem dyplomowym, Rockbox miał obsługę dwóch urządzeń opartych na wymienionym układzie: Olympus m:robe 500 oraz Creative Zen Vision M. Z tego powodu, część funkcjonalności (inicjalizacja procesora, jednostki MMU, procesora sygnałowego) było już zaimplementowanych. Creative Zen Vision M wykorzystuje wbudowany moduł I^2C, którego linie są

multipleksowane z interfejsem SPI1. Komunikacja z procesorem pomocnicznym odbywa się znacznie częściej aniżeli z układami przyłączonymi do magistrali I^2C (TPS65021 oraz TLV320AIC3106). Prawdopodobnie z tego powodu magistrala I^2C jest realizowana programowo na liniach GIO35 (SDA) oraz GIO36 (SCL). Zarówno Creative Zen Vision M jak i Olympus m:robe 500 wykorzystują interfejs SPI0. Podstawowa różnica pomiędzy SPI0 a SPI1, polega na tym, że ten pierwszy obsługuje tryb DMA. Sandisk Sansa Connect wykorzystuje SPI0 do komunikacji z modułem WiFi, a SPI1 do wymiany informacji z procesorem pomocniczym. Jest to rozwiązanie sensowne, ponieważ poprzez sieć (gdy się z niej korzysta) przesyłane jest stosunkowo dużo danych (np. strumieniowanie audio).

4.2.1 Procesor pomocniczy

Pomocniczy procesor Atmel Mega165PV jest odpowiedzialny za obsługę przycisków, zarządzanie zasilaniem, reset kodeka audio, jak również reset oraz całkowite wyłączenie systemu. Komunikacja z tym układem odbywa się za pośrednictwem interfejsu SPI1. Wektor stanu systemu zawiera informacje dotyczące wciśniętych przycisków oraz poziomu naładowania baterii, każda jego zmiana powoduje zgłoszenie przerwania do procesora poprzez ustawienie stanu niskiego na linii GIO0 głównego mikrokontrolera.

Monitorowanie stanu systemu

Procedura przyjęcia przerwania (na zboczu opadającym sygnału GIO0) uaktywnia wątek odpowiedzialny za monitorowanie stanu systemu. Wątek ten dokonuje odczytu oraz interpretacji wektora stanu z procesora pomocniczego. Początkowo, całość operacji odbywała się w procedurze przyjęcia przerwania, jednakże powodowało to problemy związane z utratą wrażenia pracy ciągłej innych części systemu (m.in. obsługa wyświetlacza). W systemach wbudowanych procedura przyjęcia przerwania powinna być możliwie jak najkrótsza, oddelegowanie części zadań do przestrzeni użytkownika (tutaj: do odpowiedniego wątku) jest często stosowaną praktyką celem zmniejszenia czasu reakcji urządzenia.

Inne

Układ Mega165PV nadzoruje proces ładowania baterii po podłączeniu zewnętrznego źródła zasilania oraz odpowiada za wykonanie resetu kodeka audio. Ponadto, ten komponent zarządza stanem wyświetlacza. Odpowiednie komendy służą do załączenia lub wyłączenia modułu LCD, jak również wprowadzenia wyświetlacza w stan uśpie-

nia. Dokładny sposób połączenia z wyświetlaczem nie został ustalony z powodu braku dokumentacji technicznej do zastosowanego ekranu LCD.

4.2.2 Wyświetlacz LCD

Wyświetlacz LCD połączony jest z modułem VENC (ang. Video Encoder), który wykorzystuje komponent OSD (ang. On Screen Display). Dane przesyłane są w formacie RGB666 (po 6 bitów na każdą ze składowych RGB). W początkowej wersji OSD korzystało bezpośrednio z bufora ramki obsługiwanego przez komponent odpowiedzialny za interfejs użytkownika (ang. GUI) systemu Rockbox. Rozwiązanie to powodowało wrażenie migotania w przypadku wykonywania dużej ilości operacji graficznych. Problem został wyeliminowany poprzez zastosowanie buforowania. Moduł GUI po zakończeniu wykonywania operacji graficznych na buforze ramki informuje sterownik o zmianach. Sterownik dokonuje aktualizacji bufora przechowującego aktualnie wyświetlany obraz.

Jasność wyświetlacza sterowana jest za pomocą sygnału PWM1 (linia GIO34). W przypadku wyłączenia wyświetlacza linia GIO34 przechodzi w stan niski (następuje wyłączenie funkcji PWM1) oraz zatrzymywany jest zegar modułu VENC (częstotliwość zegara modułu OSD jest skonfigurowana jako połowa częstotliwości zegarowej VENC).

4.2.3 Pamięć masowa

Sandisk Sansa Connect wykorzystuje interfejs MMC/SD do obsługi pamięci masowej (zarówno pamięci wewnętrznej, jak i zewnętrznej karty microSD). Zarówno Olympus m:robe 500, jak i Creative Zen Vision M nie korzystają z tego interfejsu, ponieważ tam funkcję pamięci masowej pełni dysk twardy.

Wybór urządzenia (iNAND lub zewnętrzna karta microSD) następuje poprzez linie sterujące buforami trójstanowymi. Układ iNAND wybierany jest stanem niskim na linii GIO5. Zewnętrzną kartę pamięci wybiera się za pomocą stanu niskiego sygnału GIO6. Linie GIO37 oraz GIO38 służą do załączenia zasilania, odpowiednio zewnętrznej karty pamięci oraz modułu iNAND. Zaimplementowany sterownik obsługuje transfer w trybie DMA.

4.2.4 Dźwięk

W urządzeniu zastosowano kodek audio TLV320AIC3106 firmy Texas Instruments. Konfiguracja parametrów (m.in. głośność, źródło oraz częstotliwość danych audio) odbywa się za pomocą magistrali I^2C. Referencyjny sygnał zegarowy służący do zapewnienia odpowiedniego tempa odtwarzania jest dostarczany z linii GIO16. Procesor należy

skonfigurować tak, aby na tej linii był wyprowadzony sygnał o częstotliwości 16 razy mniejszej od częstotliwości sygnału zegarowego PLLIN (funkcja CLKOUT0). Ten sam sygnał służy do taktowania interfejsu szeregowego procesora sygnałowego (chodzi tu o taktowanie modułu, nie transmisji).

Dane audio są przesyłane do kodeka audio za pośrednictwem interfejsu szeregowego procesora sygnałowego. Kodek audio jest odpowiedzialny za generację sygnału zegarowego wykorzystywanego w transmisji. Oryginalne oprogramowanie wykorzystuje protokół I^2S, jednakże w systemie Rockbox zastosowano format "right justified data". Czynnikiem motywującym do tej decyzji była możliwość uruchomienia na procesorze sygnałowym kodu wykorzystywanego w urządzeniu Olympus m:robe 500 (właśnie w formacie "right justified data").

4.3 Ostateczny rezultat pracy

Rezultatem projektu inżynierskiego jest opracowanie metody uruchamiania oprogramowania wewnętrznego nieautoryzowanego przez producenta na odtwarzaczu Sandisk Sansa Connect. Uzyskano działającą wersję systemu operacyjnego Rockbox na wymienionym wyżej urządzeniu. Pod kontrolą systemu Rockbox możliwe jest odtwarzanie plików muzycznych (obsługiwana jest większa liczba formatów niż w przypadku oryginalnego oprogramowania), działają gry, aplikacje oraz dema dystrybuowane wraz z systemem. System Rockbox na urządzeniu uruchamia się znacznie szybciej aniżeli oryginalne oprogramowanie dostarczane przez producenta. Po zainstalowaniu programu rozruchowego systemu Rockbox możliwe jest uruchomienie oryginalnego oprogramowania poprzez przytrzymanie przycisku PREVIOUS po załączeniu odtwarzacza.

5 Podsumowanie

Praca została wykonana w pełnym zakresie i zgodnie z założonym celem. Opracowana została metoda uruchomienia alternatywnego oprogramowania wewnętrznego, nieautoryzowanego przez producenta sprzętu. Modyfikacje opracowane przez autora niniejszego projektu inżynierskiego zostały zamieszone w systemie śledzenia zgłoszeń projektu Rockbox dnia 2 listopada 2011 roku (identyfikator zgłoszenia FS#12363). Programiści systemu Rockbox, po przejrzeniu proponowanych zmian zgłosili swoje uwagi, które zostały uwzględnione przez autora projektu. Dnia 16 listopada 2011 roku, autor niniejszej pracy dyplomowej został oficjalnym programistą projektu Rockbox i uzyskał prawo zatwierdzania zmian w głównym repozytorium projektu. Od tego dnia, w gałęzi rozwojowej systemu znalazła się obsługa odtwarzacza Sandisk Sansa Connect (rewizja SVN nr 31000).

Dnia 13 listopada 2011 roku odtwarzacz Sandisk Sansa Connect działając pod kontrolą systemu operacyjnego Rockbox po raz pierwszy odtworzył utwór muzyczny. Był to Exlibris - W Objęciach Kruka. Zgodnie z tradycją [17] projektu zostało to ogłoszone na grupie dyskusyjnej.

Projekt ten można rozwinąć. Przykładowa lista usprawnień jakie można wykonać:

- Obsługa zegara czasu rzeczywistego.
  Wewnątrz urządzenia znajduje się zegar czasu rzeczywistego ST M41T62. Oryginalne oprogramowanie nie posiada funkcji umożliwiających użytkownikowi ustawienie aktualnej daty. Komponent ten nie jest przyłączony do magistrali I^2C obecnej na liniach GIO35 (SDA) oraz GIO36 (SCL).

- Dodanie obsługi interfejsu USB.
  W urządzeniu interfejs USB obsługiwany jest przez układ TNETV105PAP. Niedostępna jest dokumentacja techniczna tego komponentu. Na szczęście kod źródłowy jądra systemu Linux udostępniony przez firmę Sandisk zawiera odpowiedni sterownik.

- Dodanie obsługi modułu WiFi.
  System Rockbox nie zawiera stosu TCP/IP, stąd też celem uzyskania działającego połączenia sieciowego niezbędna jest implementacja całego podsystemu sieciowego. Dodatkowym problemem jest brak dokumentacji technicznej do zastosowanego modułu WiFi. Oryginalne oprogramowanie obsługuje sieć bezprzewodową z wykorzystaniem własnościowego (ang. proprietary) modułu jądra systemu, stąd też nie jest dostępny odpowiedni kod źródłowy.

Dodatek A Próby ominięcia weryfikacji autentyczności oprogramowania

Niniejszy dodatek zawiera opis kilku pomysłów na uzyskanie możliwości uruchomienia oprogramowania niepodpisanego cyfrowo przez producenta oraz informację o rezultatach ich zastosowania.

Wymiana pamięci iNAND na slot na kartę microSD

Komunikacja z pamięcią flash Sandisk iNAND odbywa się w oparciu o magistrale SD (ang. Secure Digital) lub SPI (ang. Serial Peripheral Interface), stąd też możliwe jest wlutowanie w miejsce tego układu slotu na karty microSD.

Uzasadnienie

Wymiana pamięci iNAND na slot na kartę microSD umożliwia uzyskanie zdalnego dostępu, poprzez telnet, do systemu uruchomionego na odtwarzaczu. Zdalny dostęp można uzyskać przez zmienę plików znajdujących się na karcie.

Rezultat

Wylutowano układ iNAND i przylutowano w odpowiednie miejsce slot na karty microSD. Po zakończeniu procesu odzyskiwania, na karcie microSD znalazł się system plików ext3 zawierający część aplikacji wchodzących w skład oprogramowania wewnętrznego odtwarzacza. Edycja odpowiedniego pliku (assets/System/runzap) umożliwiła wystartowanie usługi telnet.

Następnie zalogowano się do działającego systemu i przekonano się, że nie można skasować programu rozruchowego (komenda `flash_eraseall /dev/mtd0` zwróciła błąd wejścia/wyjścia). Po wgraniu zmodyfikowanego obrazu podstawowego systemu plików (do pliku urządzenia `/dev/mtd3`), okazało się, że ponowne uruchomienie odtwarzacza powoduje jego przejście do trybu odzyskiwania, co sugeruje, że przy każdym starcie następuje sprawdzenie poprawności podpisu cyfrowego. Naruszenie ochrony pamięci po wysłaniu dużej ilości losowych danych na port 8088 odtwarzacza powoduje naruszenie ochrony pamięci w funkcji konwertującej łańcuch znaków na małe litery. Nie udało się ustalić, czy ten błąd można wykorzystać do uruchomienia zdalnego kodu.

Doprowadziło to do wniosku, że sama zmiana zawartości pamięci iNAND jest niewystarczająca do uruchomienia nieautoryzowanego oprogramowania wewnętrznego (możliwe jest natomiast uruchomienie nieautoryzowanych aplikacji działających pod kontrolą systemu Linux stanowiącego oprogramowanie dostarczone przez producenta), przy-

stąpiono do szczegółowej analizy wykorzystanych układów scalonych oraz identyfikacji punktów testowych znajdujących się na powierzchni obwodu drukowanego odtwarzacza.

Identyfikacja punktów testowych

Identyfikacja punktów testowych została przeprowadzona za pomocą testera ciągłości po uprzednim wylutowaniu wszystkich układów scalonych.

Uzasadnienie

W przypadku gdy w urządzeniu (na obwodzie drukowanym) wyprowadzony jest interfejs JTAG możliwe jest przeprogramowanie układu lub jego uruchamianie (ang. debugging).

Rezultat

Wśród 16 ułożonych obok siebie padów, znalezione zostały wyprowadzenia interfejsu JTAG oraz portu szeregowego (UART). Po dokładnym zmierzeniu rozstawu i przeszukaniu dostępnych złącz, ustalone zostało, że pasującym elementem jest Hirose FH19SC-16S-0.5SH(05).

Połączenie poprzez JTAG

Zaprojektowany został pomocniczy obwód drukowany pełniący funkcje konwertera UART⇔USB oraz przejściówki na standardowe wyprowadzenia interfejsu JTAG. Do odtwarzacza zostało wlutowane odpowiednie złącze.

Uzasadnienie

Interfejs JTAG umożliwia programowanie układu jak również prace krokową.

Rezultat

Działanie w trybie krokowym umożliwiło lepsze zrozumienie kodu programu rozruchowego. Opracowane zostały modyfikacje poszczególnych instrukcji powodujące ominięcie sprawdzania poprawności podpisu cyfrowego. Pamięć NOR flash jest zabezpieczona przed zapisem, stąd też nie udało się zapisać w niej zmodyfikowanej wersji bootloadera.

Zmiana adresu startowego

Próbą obejścia problemu zabezpieczonej pamięci flash była zmiana adresu, spod którego procesor wykonuje instrukcje. Układ TMS320DM320 ma dwie linie informacyjne BTSEL0 oraz BTSEL1, które określają sekwencję startową po wystąpieniu sygnału RESET. Możliwe są następujące kombinacje stanu sygnałów BTSEL[1:0]:

- 00, 11 – Start z zewnętrznej pamięci NOR flash
- 01 – Tryb startowy zewnętrznego hosta (ang. Boot-up mode external host)
- 10 – Start z wewnętrznej pamięci ROM

Uzasadnienie

Zmiana kodu wykonywanego po starcie systemu umożliwi obejście sprawdzania poprawności podpisu cyfrowego.

Rezultat

Na obwodzie drukowanym znalezione zostały rezystory typu pull-down dla linii BTSEL0 oraz BTSEL1. Wewnętrzna pamięć ROM zawiera kod służący do odczytu programu rozruchowego użytkownika (ang. user boot-loader) z pamięci NAND flash. Niestety, kod zawarty w pamięci ROM nie obsługuje kart SD, jak również nie ma możliwości przeprogramowania tej pamięci. Rozważona została także możliwość wykorzystania trybu startowego zewnętrznego hosta. Znalezione zostały miejsca, gdzie "można się podlutować" celem sterowania liniami SCLK0 oraz SENZ, jednakże punkty te znajdują się w miejscu praktycznie niedostępnym bez poważnej ingerencji sprzętowej. Ponadto, z powodu konieczności fizycznego dodania dodatkowego układu, który by odpowiednio programował procesor po wystąpieniu sygnału RESET, pomysł ten został całkowicie odrzucony.

Zdjęcie zabezpieczenia bloków w pamięci NOR flash

Z powodu wcześniejszych niepowodzeń, prace skupiły się na ustaleniu możliwości zdjęcia ochrony przed zapisem z bloków pamięci NOR flash zawierających kod programu uruchomieniowego.

Uzasadnienie

Zdjęcie blokady przed zapisem umożliwi zapisanie zmodyfikowanej wersji programu uruchomieniowego.

Rezultat

Zgodnie z dokumentacją techniczną wykorzystanej pamięci, założenie bądź też zdjęcie ochrony poszczególnego bloku wymaga podania wysokiego (z przedziału $8.5V \sim 12.5V$) napięcia na linie RESET (lub z wykorzystaniem alternatywnej metody – na linię A9). Odnaleziono układ TI TPS3106 dokonujący resetu pamięci NOR flash w przypadku skoku napięcia (ang. supervisor). Z racji tego, że układ ten jest w obudowie dość łatwej do rozlutowania, rozważana była możliwość wylutowania go i podania na linię RESET wysokiego napięcia. Niestety, okazało się, że sygnał ten jest również połączony z sygnałem RESET procesora głównego. Ponieważ procesor główny nie jest w stanie przyjąć na wejście tak dużego napięcia, metoda ta może zostać zastosowana jedynie w przypadku wylutowania układu z obwodu drukowanego i zdjęcia blokady w zewnętrznym programatorze. Z racji wysokiego kosztu takiej operacji (obudowa BGA, konieczność wylutowania metalowej ramki pełniącej funkcję ekranu) możliwość ta została odrzucona.

Szybkie programowanie układu (ang. Accelerated Program Operation)

Układ K8D3216UBC posiada wejście $\overline{WP}/ACC$, które w zależności od wartości napięcia do niego przyłożonego wpływa na zachowanie układu. Możliwe wartości napięcia na tym wejściu to:

- V_{IL} $(-0.5V \sim 0.8V)$ – w tym przypadku pierwsze dwa bloki są zawsze zabezpieczone przed zapisem,
- V_{IH} $(0.7V_{CC} \sim V_{CC}+0.3V)$ – w tym przypadku zabezpieczenie bloków jest zależne od tego, czy dokonano procedury zabezpieczania z wykorzystaniem wysokiego napięcia,
- V_{HH} $(8.5V \sim 12.5V)$ – wszystkie bloki pamięci zostają tymczasowo odbezpieczone, urządzenie przechodzi w tryb Unlock Bypass (wymagane tylko dwa cykle magistrali na zaprogramowanie; standardowa procedura programowania wymaga cztery cykle) oraz zmniejsza się czas programowania układu.

Należy zwrócić uwagę, że pin $\overline{WP}/ACC$ może być w stanie V_{HH} wyłącznie w trakcie przyśpieszonego programowania.

Uzasadnienie

Tryb ten powoduje tymczasowe zdjęcie zabezpieczenia przed zapisem, stąd też możliwa jest zmiana bootloadera z wykorzystaniem akcelerowanej metody programowania.

Rezultat

Na obwodzie drukowanym znaleziona została ścieżka łącząca sygnał $\overline{WP}/ACC$ z zasilaniem. Możliwe jest przerwanie tej ścieżki. Dzięki podaniu na linie $\overline{WP}/ACC$ napięcia V_{HH} udało się przeprogramować program rozruchowy tak, aby podpis cyfrowy oprogramowania zawsze był uznawany za poprawny.

Dodatek B Konfiguracja OpenOCD

Niniejszy dodatek zawiera plik konfiguracyjny pakietu OpenOCD umożliwiający wykorzystanie adaptera JTAGKey firmy Amontec do przeprowadzenia sesji uruchamiania oprogramowania na odtwarzaczu Sandisk Sansa Connect.

Listing 5: openocd.cfg

```
# zalaczenie pliku konfiguracyjnego wykorzystanego adaptera JTAG
source [find interface/jtagkey.cfg]

# konfiguracja resetu interfejsu JTAG poprzez linie TRST
reset_config trst_only
jtag_rclk 8

# Texas Instruments TMS320DM320
if { [info exists CHIPNAME] } {
    set  _CHIPNAME $CHIPNAME
} else {
    set  _CHIPNAME dm320
}

# Procesory z serii TMS320 zawieraja modul icepick
source [find target/icepick.cfg]

# ARM ETB11
if { [info exists ETB_TAPID ] } {
   set _ETB_TAPID $ETB_TAPID
} else {
   set _ETB_TAPID 0x00000000
}


#konfiguracja portu dostepu testowego modulu icepick
#parametr irlen okresla dlugosc rejestru rozkazow
jtag newtap $_CHIPNAME etb -irlen 4 -expected-id $_ETB_TAPID
jtag configure $_CHIPNAME.etb -event tap-enable \
    "icepick_c_tapenable $_CHIPNAME.jrc 1"

# ARM926ejs
if { [info exists CPU_TAPID ] } {
   set _CPU_TAPID $CPU_TAPID
} else {
   set _CPU_TAPID 0x0792602f
}
```

```
#konfiguracja portu dostepu testowego rdzenia ARM
jtag newtap $_CHIPNAME arm -irlen 4 -expected-id $_CPU_TAPID
jtag configure $_CHIPNAME.arm -event tap-enable \
    "icepick_c_tapenable $_CHIPNAME.jrc 0"

set _TARGETNAME $_CHIPNAME.arm

target create $_TARGETNAME arm926ejs -chain-position $_TARGETNAME

# ustalenie szybkosci interfejsu JTAG (sygnal TCK)
jtag_rclk 1500
$_TARGETNAME configure -event "reset-start" { jtag_rclk 1500 }

#opcjonalne, funkcje przyspieszajace dostep do pamieci dla ARM7/9
arm7_9 fast_memory_access enable
arm7_9 dcc_downloads enable

#konfiguracja modulu pamieci NOR Flash, poprzez interfejs CFI
#0x100000 - adres pod ktory zamapowana jest pamiec
#0x400000 - rozmiar pamieci (4 MB)
#2 2 - 16 bitow danych podpiete do 16 linii danych procesora
flash bank samsung cfi 0x100000 0x400000 2 2 dm320.arm
```

Dodatek C Przyciski odtwarzacza Sansa Connect

Niniejszy dodatek przedstawia zdjęcie odtwarzacza wraz z nazwami przycisków (rysunek 5) jakie zostały przyjęte w systemie Rockbox.

Rysunek 5: Zdjęcie odtwarzacza wraz z naniesionymi nazwami przycisków.

Literatura

[1] http://sourceware.org/binutils/ (link aktywny: 2011-12-06)

[2] http://www.hex-rays.com/products/ida/index.shtml (link aktywny: 2011-12-06)

[3] http://www.ellisys.com/products/usbex200/index.php (link aktywny: 2011-12-06)

[4] http://dangerousprototypes.com/bus-pirate-manual/ (link aktywny: 2011-12-06)

[5] Universal Serial Bus Device Class Specification for Device Firmware Upgrade, Version 1.1 Aug 5, 2004.

[6] 1149.1-2001 IEEE Standard Test Access Port and Boundary-Scan Architecture. http://ieeexplore.ieee.org/servlet/opac?punumber=7481 (link aktywny: 2011-11-15)

[7] http://www.rockbox.org/wiki/IaudioBoot (link aktywny: 2011-12-06)

[8] http://hg.atheme.org/malloctools (link aktywny: 2011-12-06)

[9] SanDisk Sansa® Connect™ MP3 Player User Manual

[10] http://www.maximumpc.com/article/demise_of_yahoo_music_to_kill_the_sansa_connect (link aktywny: 2011-12-08)

[11] http://kb.sandisk.com/app/answers/detail/a_id/863/~/sansa-connect-recovery-tool (link aktywny: 2011-12-08)

[12] http://www.linuxjournal.com/article/10835 (link aktywny: 2011-12-11)

[13] http://daniel.haxx.se/blog/2011/12/07/ten-years-of-rockbox/ (link aktywny: 2011-12-10)

[14] http://www.rockbox.org/wiki/WhyRockbox (link aktywny: 2011-12-10)

[15] http://wiki.hydrogenaudio.org/index.php?title=ReplayGain_1.0_specification (link aktywny: 2011-12-11)

[16] http://qtscrob.sourceforge.net (link aktywny: 2011-12-11)

[17] http://www.rockbox.org/wiki/GentlemenMails (link aktywny: 2011-12-11)

Zawartość płyty CD

Płyta CD załączona do niniejszej pracy zawiera następujące katalogi:

- `rockbox` - kod źródłowy systemu Rockbox (SVN r31537),
- `rockbox-bin` - wersja binarna systemu operacyjnego Rockbox (SVN r31537) przeznaczona na odtwarzacz Sandisk Sansa Connect,
- `rbdev-dl` - pliki wykorzystywane przez skrypt `rockboxdev.sh` (znajdujący się w katalogu `rockbox/tools`) celem uzyskania zestawu narzędzi umożliwiających skompilowanie systemu,
- `zsitool` - kod źródłowy programu umożliwiającego wgranie pliku w formacie .srr do urządzenia pracującego w trybie odzyskiwania,
- `tex` - wersja źródłowa niniejszego dokumentu,
- `pdf` - niniejszy dokument w formacie PDF.