

**Szalony  
Geniusz**

— DONALD MORRIS

**RASPBERRY PI  
NIESAMOWITE  
PROJEKTY**

Miniaturowy komputer  
i jego wielka moc!

Helion



Tytuł oryginału: Raspberry Pi Projects for the Evil Genius

Tłumaczenie: Andrzej Watrak

ISBN: 978-83-246-9174-6

Original edition copyright © 2014 by McGraw-Hill Education  
All rights reserved

Polish edition copyright © 2014 by HELION SA  
All rights reserved

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie bierze jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Wydawnictwo HELION nie ponosi również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Projekt okładki: Studio Gravite/Olsztyn  
Obarek, Pokoński, Pazdrijowski, Zaprucki

Wydawnictwo HELION  
ul. Kościuszki 1c, 44-100 GLIWICE  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:  
<ftp://ftp.helion.pl/przyklady/rpszge.zip>

Drogi Czytelniku!  
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres  
<http://helion.pl/user/opinie/rpszge>  
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

O autorze .....	9
Podziękowania .....	9
Przedmowa .....	11
<b>1. Wprowadzenie do Raspberry Pi .....</b>	<b>13</b>
Sprzęt .....	14
Oprogramowanie .....	25
Podsumowanie .....	33
<b>2. Migacz LED .....</b>	<b>35</b>
Płytki prototypowa .....	35
Lutowanie .....	35
Dostęp do pinów GPIO .....	36
Polecenie apt-get .....	38
Projekt migacza LED .....	39
Podsumowanie .....	42
<b>3. Odtwarzacz MP3 .....</b>	<b>43</b>
Złącze prototypowe .....	43
Przenośny głośnik stereofoniczny .....	44
Formaty plików dźwiękowych .....	46
Wejścia przycisków sprzętowych .....	49
Myślenie w kategoriach RasPi .....	51
Wymagania projektu odtwarzacza MP3 .....	51
Podsumowanie .....	54
<b>4. Sterownik aparatu fotograficznego .....</b>	<b>55</b>
Wymagania projektu .....	56
Działanie migawki aparatu .....	56
Elektroniczna lampa błyskowa .....	58
Interfejsy optoizolacyjne .....	60
Moduły dźwiękowy i laserowy .....	61
Połączenia interfejsu .....	63
Funkcja fotografii poklatkowej .....	68
Podsumowanie .....	69
<b>5. System GPS .....</b>	<b>71</b>
Wprowadzenie .....	71
Podstawy działania systemu GPS .....	72
Odbiornik Ultimate GPS .....	75
Protokół NMEA .....	81
Aplikacje gpsd .....	83
Podsumowanie .....	85
<b>6. Sejsmograf .....</b>	<b>87</b>
Wprowadzenie .....	87
Sejsmologia i trzęsienia ziemi .....	87
Konwersja analogowo-cyfrowa .....	91
Interfejs SPI .....	92
Podłączenie i testowanie przetwornika MCP3008 z płytką RasPi .....	93

Analiza danych sejsmicznych .....	96
Gotowy system .....	99
Podsumowanie .....	99
<b>7. Automatyka domowa .....</b>	<b>101</b>
Wprowadzenie .....	101
Sieć mieszana Z-Wave .....	101
Interfejs pomiędzy RasPi a siecią Z-Wave .....	108
Logowanie SSH .....	109
Oprogramowanie Open Z-Wave .....	109
Podsumowanie .....	115
<b>8. Domowy system bezpieczeństwa .....</b>	<b>117</b>
Wprowadzenie .....	117
Wybór kamery .....	117
Montaż czujnika laserowego .....	118
Pakiet oprogramowania Motion .....	120
Korzystanie z kamery .....	123
Program obsługujący czujnik laserowy .....	125
Dodatkowe czujniki .....	126
Podsumowanie .....	127
<b>9. Czytnik NFC/RFID .....</b>	<b>129</b>
Wprowadzenie .....	129
Jak działa czytnik NFC? .....	129
Instalacja pakietu libnfc .....	131
Instalacja sprzętu .....	134
Wstępny test .....	136
Główny program .....	136
Przyszła rozbudowa .....	137
Podsumowanie .....	138
<b>10. Stacja meteorologiczna .....</b>	<b>139</b>
Wprowadzenie .....	139
Budowa stacji meteorologicznej .....	139
Protokół 1-Wire .....	142
Moduł analizujący 1-Wire .....	146
Podłączenie komponentów stacji meteorologicznej .....	146
Oprogramowanie owfs .....	147
Odczyt danych meteorologicznych .....	147
Analiza pakietów .....	150
Przyszła rozbudowa .....	151
Podsumowanie .....	151
<b>11. Sterownik logiczny .....</b>	<b>153</b>
Wprowadzenie .....	153
Zewnętrzna płytk sterująca 1-Wire .....	153
Instalacja i konfiguracja oprogramowania 1-Wire File System (owfs) .....	155
Linux, urządzenia i aplikacja FUSE .....	157
Testy płytki 8 Channel I/O .....	158
Program testowy Python .....	160
Monitorowanie ruchu za pomocą analizatora .....	161
Zdalne sterowanie za pomocą telefonu Android .....	162
Test serwera WWW w przeglądarce .....	165
Podsumowanie .....	167

<b>12.</b>	Pojazd samobieżny, część I .....	169
	Wprowadzenie .....	169
	Opis projektu .....	169
	Podwozie i silniki napędowe .....	170
	Serwomechanizmy .....	175
<b>13.</b>	Pojazd samobieżny, część II .....	179
	Wprowadzenie .....	179
	Schemat blokowy pojazdu .....	179
	Oprogramowanie I <sup>2</sup> C .....	181
	Oprogramowanie Bluetooth .....	181
	Program sterujący pojazdem .....	185
	Sterowanie pojazdem .....	187
	Przyszłe rozszerzenia .....	187
	Podsumowanie .....	188
<b>14.</b>	Wykrywacz radonu .....	189
	Wprowadzenie .....	189
	Radioaktywność i radon .....	189
	Licznik Geigera-Mullera K2645 .....	192
	Wstępna konfiguracja testowa .....	194
	Konstrukcja przenośnego wykrywacza radonu .....	197
	Obsługa przenośnego licznika Geigera .....	198
	Modyfikacje i rozbudowa .....	199
	Podsumowanie .....	199
<b>15.</b>	Serwer sygnału czasu .....	201
	Wprowadzenie .....	201
	Zegar czasu rzeczywistego (RTC) .....	201
	Oprogramowanie RTC .....	203
	Wprowadzenie do protokołu NTP .....	207
	Uruchomienie serwera NTP na płytce RasPi .....	208
	Podsumowanie .....	210
	Skorowidz .....	212



# Odtwarzacz MP3

W tym projekcie pokażę Ci, jak zbudować bardzo prosty, ale w pełni funkcjonalny odtwarzacz MP3. Opiszę również, jak wykorzystać piny GPIO jako wejścia, które pozwolą zastosować w odtwarzaczu przyciski do wybierania utworów do odtwarzania.

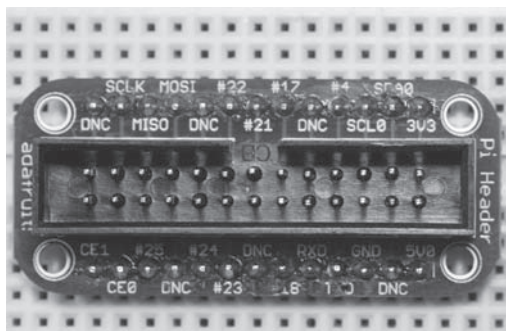
Poznasz również pakiet oprogramowania ALSA (*Advanced Linux Sound Architecture*, zaawansowana architektura dźwiękowa systemu Linux) służący do generowania dźwięku w systemie Linux, oferujący kilka podstawowych narzędzi ułatwiających zbudowanie tego projektu.

## Złącze prototypowe

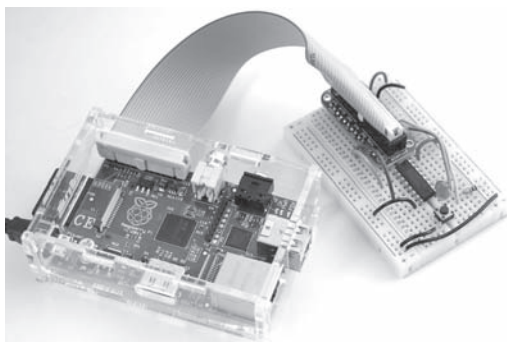
Najpierw skupmy się na komponencie, którego użyjemy do podłączenia przycisków do pinów GPIO płytki RasPi. Ten pomocniczy element różni się od płytki Pi Plate przedstawionej w poprzednim rozdziale. Nosi on nazwę Pi Cobbler i aby wyraźnie było widać jego piny, jest pokazany na rysunku 3.1 bez taśmy przyłączeniowej.

W rzeczywistości jest to zwykły przedłużacz (albo akcesorium montażowe) do złącza GPIO, umożliwiający podłączenie wszystkich pinów do płytki prototypowej. Zestaw jest dostępny u wielu dostawców, między innymi Adafruit pod adresem [www.adafruit.com/products/914](http://www.adafruit.com/products/914). Na rysunku 3.2 jest pokazany złożony przedłużacz Pi Cobbler podłączony do płytki prototypowej i RasPi.

Użycie przedłużacza Pi Cobbler i płytki prototypowej umożliwia szybkie tworzenie nowych i łatwą modyfikację istniejących projektów. Ułatwi Ci podłączenie do płytki prototypowej wielu różnych wstępnie przygotowanych przewodów. Możesz również wykonać własny przewód ze znormalizowanego przewodu miedzianego lub kupić gotowy zestaw od swojego ulubionego dostawcy elementów elektronicznych. Rysunek 3.3 przedstawia wiązkę przewodów, którą kupiłem na potrzeby wykonania projektów opisanych w tej książce. Są to w rzeczywistości elastyczne oplecione przewody zakończone sztywną zaizolowaną końcówką. Przewody są warte swojej ceny i dobrze jest kupić taki zestaw.



**RYСУNEK 3.1.** Łącznik prototypowy Pi Cobbler



**RYСУNEK 3.2.** Przedłużacz Pi Cobbler umieszczony na płytce prototypowej, podłączony do płytki Raspberry Pi

## Przenośny głośnik stereofoniczny

Musisz w jakiś sposób odsłuchiwać utwory MP3. Ja używam bardzo małego i niedrogiego zestawu głośników stereofonicznych Veho 360° ([www.veho-uk.com](http://www.veho-uk.com)), pokazanego na rysunku 3.4.

Ten zestaw jest wyposażony we wtyk 3,5 mm, który możesz podłączyć do wyjścia audio RasPi. Zestaw posiada również akumulator, który można ładować poprzez podłączenie urządzenia do standardowego gniazda USB. Nie polecam jednak używać do tego celu gniazda USB RasPi ze względu na ograniczenia prądowe, o których pisałem w rozdziale 1. Regulator głośności w głośniku umożliwia ustawienie tylko dwóch stopni siły dźwięku. Ogólnie rzecz biorąc, ten zasilany z akumulatora głośnik dość dobrze odtwarza dźwięk, ale nie zamieniłbym na niego swojego domowego systemu dźwiękowego.

## STEROWNIK ALSA

ALSA jest podstawowym sterownikiem dźwiękowym wykorzystywanym przez większość dystrybucji systemu Linux dla RasPi. Jest to bardzo stabilne oprogramowanie, szeroko stosowane w różnych dystrybucjach systemu Linux. Jest całkowicie otwarte i doskonale udokumentowane na stronie [www.alsa-project.org](http://www.alsa-project.org). Tabela 3.1 zawiera listę ośmiu pakietów z krótkimi opisami, składających się na sterownik ALSA.

**TABELA 3.1.** Pakiety sterownika ALSA

Nazwa pakietu	Opis
<i>alsa-driver</i>	Niskopoziomowy sterownik łączący sprzęt z jądrem systemu Linux
<i>alsa-lib</i>	Interfejs API w języku C do sterownika
<i>alsa-utils</i>	Zestaw bardzo przydatnych narzędzi
<i>alsa-tools</i>	Zaawansowany zestaw narzędzi wykorzystywany z niektórymi kartami dźwiękowymi (ale nie w RasPi)
<i>alsa-firmware</i>	Oprogramowanie wbudowane w niektóre karty dźwiękowe (ale nie w RasPi)
<i>alsa-plugins</i>	Pakiet wykorzystywany z bibliotekami audio i sterownikami dźwiękowymi
<i>alsa-oss</i>	Starsza wersja pakietu o nazwie Open Sound System, dostarczana w celu zapewnienia kompatybilności
<i>pyalsa</i>	Moduł ALSA dla języka Python, wykorzystywany w systemie do edycji filmów wideo

Mimo że sterowniki ALSA powinny być już zainstalowane w Twojej dystrybucji systemu Linux, musisz wprowadzić poniższe polecenia w oknie terminala, aby sprawdzić, czy sterowniki rzeczywiście są dostępne:

```
sudo modprobe snd-bcm2835
```



**RYСУNEK 3.3.** Wysokiej jakości przewody do płytki prototypowej



**RYСУNEK 3.4.** Głośnik Veho 360°



Aplikacja *modprobe* ładuje pakiet LKM (*Loadable Kernel Modules*, ładowalne moduły jądra), który w tym przypadku jest sterownikiem dźwiękowym *snd-bcm2835* zgodnym z pakietem ALSA. Sprawdź również, czy w pliku o nazwie *modules* w katalogu */etc* jest zapisany sterownik *snd-bcm2835*. W tym pliku są zapisane wszystkie moduły LKM, które mają być załadowane podczas startu systemu.

Musisz również zainstalować pakiet *alsa-utils* zawierający kilka ważnych aplikacji potrzebnych do uruchomienia naszego projektu. Instalacja polega na użyciu polecenia *apt-get* opisanego w rozdziale 1. W oknie terminala wpisz następujące polecenie:

```
sudo apt-get install alsa-utils
```

Aby wygenerować dźwięk na wyjściu audio, trzeba wykonać jeszcze jeden krok. Wpisz następujące polecenie:

```
sudo amixer cset numid=3 1
```

Aplikacja *amixer* jest częścią pakietu *alsa-utils*, która umożliwia wybranie potrzebnego wyjścia audio. Po pierwszym uruchomieniu RasPi domyślnie jest ustawiane wyjście HDMI. Przekierowanie wyjścia audio następuje po użyciu powyższego polecenia. Liczba na końcu wiersza oznacza:

- 0 — automatyczny wybór,
- 1 — wyjście analogowe,
- 2 — wyjście HDMI.

## TESTOWANIE WYJŚCIA ANALOGOWEGO

Czas sprawdzić wyjście audio płytki RasPi. Umieść wtyk głośnika w złączu 3,5 mm i wpisz w oknie terminala następujące polecenie:

```
sudo speaker-test
```

W głośnikach powinieneś usłyszeć głośny szum. Jest to tzw. **różowy szum** generowany przez aplikację *speaker-test* będącą częścią pakietu ALSA. (Dokładniejszy opis różowego szumu znajdziesz w poniższym podrozdziale — „Charakterystyka częstotliwościowa sygnału audio”). Jeżeli nie słyszysz szumu, przejrzyj wymienione wyżej polecenia i upewnij się, że zostały wprowadzone w podany sposób. Oprócz generowania szumu aplikacja *speaker-test* oferuje dużo innych opcji. Więcej informacji znajdziesz na stronie <http://manpages.ubuntu.com/manpages/natty/man1/speaker-test.1.html>.

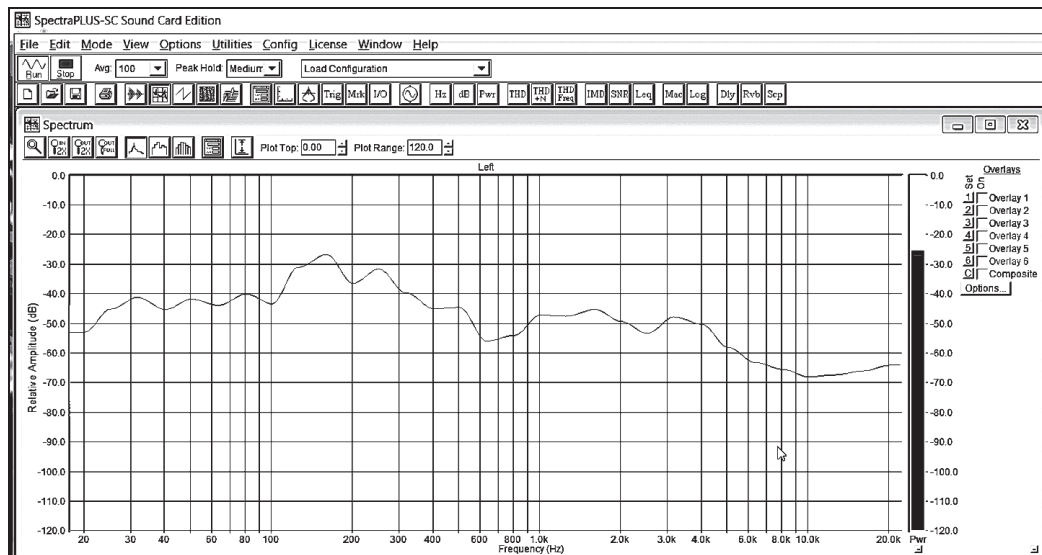
## CHARAKTERYSTYKA CZĘSTOTLIWOŚCIOWA SYGNAŁU AUDIO

Różowy szum generowany na wyjściu daje ciekawą możliwość pomiaru charakterystyki częstotliwościowej analogowego sygnału audio płytki RasPi. Dla mnie jako inżyniera akustyka z wieloletnim doświadczeniem jest to naturalna okazja do oceny wydajności akustycznej określonego podsystemu. Ale najpierw poznajmy nieco teorii różowego szumu i dowiedzmy się, dlaczego jest on tak przydatny przy określaniu charakterystyki częstotliwościowej systemu dźwiękowego. Szum w swojej istocie jest mieszaniną dowolnych dźwięków, których energia jest równomiernie rozłożona w całym paśmie częstotliwości. Tego typu szum jest zwany **szumem białym**, oznaczającym równomierny rozkład energii.

W analizie częstotliwościowej systemów akustycznych zazwyczaj wykorzystuje się zestaw filtrów z zakresu jednej trzeciej oktawy, z pewnym przybliżeniem symulujących charakterystykę częstotliwościową ludzkiego ucha. Oznacza to, że filtry są węższe dla niższych częstotliwości i zwiększają się wraz z jej wzrostem. Podanie białego szumu na taki zestaw filtrów daje w rezultacie wznoszącą lub dodatnią krzywą widma akustycznego. Biały filtr jest wstępnie przefiltrowany (lub wstępnie zniekształcony, jeśli wolisz), a szum wyjściowy przetworzony przez filtry z jednej trzeciej oktawy daje płaską charakterystykę. Każde odchylenie od płaskiej charakterystyki będzie zatem

spowodowane właściwościami testowanego systemu, a nie podanym sygnałem szumu. Taki wstępnie przetworzony szum jest nazywany **szumem różowym**.

Rysunek 3.5 przedstawia widok programu do analizy akustycznej analogowego sygnału audio będącego różowym szumem generowanym przez aplikację *speaker-test*.



**RYСУNEK 3.5.** Analiza akustyczna płytki Raspberry Pi w zakresie jednej trzeciej oktawy

Jak widać, charakterystyka sygnału jest nieregularną krzywą, której daleko do płaskiej linii. Dobra wiadomość jest taka, że odkształcenia tej krzywej nie są na tyle duże, aby sygnał audio był zniekształcony. Entuzjaści systemów audio z pewnością użyją wielopasmowego korektora, który skompensuje odchylenia tej charakterystyki.

## Formaty plików dźwiękowych

Krótkie omówienie różnych formatów plików audio pozwoli Ci poznać ich zalety i wady ważne podczas tworzenia aplikacji dźwiękowych. Istnieją trzy ogólne kategorie formatów plików dźwiękowych:

1. bez kompresji — format bez kompresji, zajmuje najwięcej miejsca;
2. z kompresją bezstratną — pominięte są okresy ciszy, zapisane są tylko dane dźwiękowe;
3. z kompresją stratną — skompresowane są wszystkie dane, dźwięk jest nieznacznie zniekształcony.

Tabela 3.2 przedstawia kilka najbardziej znanych formatów z każdej kategorii.

## PORÓWNANIE FORMATÓW WAV I MP3

Omówię dokładniej powyższe formaty, ponieważ w tym projekcie płytka RasPi wykorzystuje format MP3. Umieściłem jednak również prostą demonstrację odtwarzania przez RasPi pliku w formacie WAV (*Waveform Audio File Format*, format zapisu fali dźwiękowej). Format WAV zapewnia najwierniejsze oddanie oryginalnego dźwięku, na jakie pozwala technologia cyfrowa. Dźwięk jest próbkowany z dużą częstotliwością i wykorzystuje duże wartości liczbowe do precyzyjnego zapisu amplitudy próbki. Wysoka częstotliwość próbkowania i duże liczby w rekordach

**TABELA 3.2.** Formaty plików dźwiękowych

Nazwa	Typ kompresji	Stopień kompresji	Uwagi
WAV	Bez kompresji	1:1	Zwany również PCM
AIFF	Bez kompresji	1:1	
AU	Bez kompresji	1:1	
FLAC	Kompresja bezstratna	1:1	
WMA bezstratny	Kompresja bezstratna	2:1	Windows Media Audio
M4A	Kompresja bezstratna	2:1	Bezstratny format Apple
MP3	Kompresja stratna	od 10:1 do 20:1	
Vorbis	Kompresja stratna	od 10:1 do 20:1	
ACC	Kompresja stratna	od 10:1 do 20:1	
WMA stratny	Kompresja stratna	od 10:1 do 20:1	

narzucają wysokie wymagania dotyczące miejsca na dysku. Na przykład 3-minutowy utwór symfoniczny może zajmować 30 MB. Dla porównania plik w formacie MP3 z tą samą treścią zajmuje około 2 MB, kompresja jest więc równa 15:1. Oczywiście odbywa się to kosztem obniżonej wierności odtwarzanego dźwięku. Dla większości ludzi format MP3 jest do przyjęcia, szczególnie gdy dźwięk jest odtwarzany przez takie urządzenia jak smartfony lub tablety. Ponadto odtwarzanie dźwięku w RasPi nie jest tak wierne, jak omówione wyżej.

## ODTWARZANIE PLIKU WAV

Odtworzenie pliku WAV będzie całkiem proste, jeżeli użyjesz innej aplikacji zawartej w pakiecie *alsa-utils*. Aplikacja nazywa się *aplay* i przy uruchamianiu jej musisz jedynie podać nazwę pliku WAV, który ma być odtworzony. Na stronie internetowej tej książki pod adresem [www.mhprofessional.com/raspi](http://www.mhprofessional.com/raspi) umieściłem dwa publicznie dostępne pliki w formatach WAV i MP3 o nazwach *class.wav* i *class.mp3*. Pobierz je i zapisz w katalogu *pi*. W następnej części rozdziału użyjesz pliku *class.mp3*.

A teraz otwórz okno terminala i wpisz następujące polecenie:

```
sudo aplay class.wav
```


Sprawdź, czy głośnik jest podłączony do gniazda audio wybranego za pomocą polecenia *amixer* opisanego wcześniej w tym rozdziale. Powinieneś usłyszeć przepiękny dźwięk fugi Bacha. Utwór trwa około 2,5 min. Odtwarzanie możesz w każdej chwili zatrzymać, naciskając klawisz `^c`.

Wersja tego samego pliku muzycznego w formacie MP3 ma wielkość około 1,2 MB, natomiast w formacie WAV — ponad 25 MB. W następnej części rozdziału, w której opisałem oprogramowanie odtwarzacza MP3, będziesz miał możliwość odtworzenia tej wersji. Będzie to doskonała okazja do porównania jakości dźwięku zapisanego w nieskompresowanym formacie WAV i w stratnym formacie MP3.

## ODTWARZANIE PLIKU MP3

Istnieje wiele aplikacji do odtwarzania plików MP3 w systemie Linux. Ja wybrałem jedną z nich, o nazwie *mpg123*, ponieważ jest bardzo stabilna i łatwo ją obsługiwać za pomocą wiersza poleceń, co jest funkcją potrzebną w odtwarzaczu RasPi. Do pobrania tej aplikacji użyj polecenia *apt-get* w następujący sposób:

```
sudo apt-get install mpg123
```

**UWAGA:**  W sieci jest dostępna inna wersja aplikacji *mpg123* o nazwie *mpg321*. Na pewno działa ona dobrze, ja jednak wolę pozostać przy oryginale.

Teraz wpisz poniższe polecenie, aby odtworzyć pobrany plik MP3:

```
sudo mpg123 class.mp3
```

Plik MP3 będzie odtwarzany przez dokładnie taki sam czas, jak w przypadku wersji WAV. Podobnie jak w aplikacji *aplay*, odtwarzanie można zatrzymać, naciskając klawisz `^c`.

Ja nie zauważyłem żadnych istotnych różnic pomiędzy oboma formatami, ale być może Ty je zauważysz. Ciekawe może być zapytanie znajomych albo rodziny, czy usłyszą różnicę pomiędzy obydwooma formatami plików dźwiękowych.

Aplikacja *mpg123* jest zwana odtwarzaczem konsolowym, ponieważ obsługuje się ją, używając klawiatury i obserwując informacje wyświetlane w oknie terminala. Właśnie taka funkcja jest potrzebna do utworzenia odtwarzacza MP3 sterowanego za pomocą RasPi. Trzeba jednak wiedzieć, jakie są dostępne polecenia sterujące i ich funkcje. Można to łatwo sprawdzić, naciskając klawisz `h` podczas działania aplikacji. Rysunek 3.6 przedstawia efekt naciśnięcia tego klawisza podczas odtwarzania pliku MP3.

```

pi@raspberrypi: ~
File Edit Tabs Help
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
  version 1.14.4; written and copyright by Michael Hipp and others
  free software (LGPL/GPL) without any warranty but with best wishes
Decoder: ARM

Terminal control enabled, press 'h' for listing of keys and functions.

Playing MPEG stream 1 of 5: class1.mp3 ...

MPEG 2.0, Layer: III, Freq: 22050, mode: Joint-Stereo, modext: 2, BPF : 209
Channels: 2, copyright: No, original: Yes, CRC: No, emphasis: 0.
Bitrate: 64 kbit/s Extension value: 0
Title:  onclassical_demo_demicheli_geminiani_pieces_allegro-in-f-major_small-ve
rsion
Artist:
Comment:
Album:
Year:
Genre: Unknown
Frame# 1040 [ 2969], Time: 00:27.16 [01:17.55], RVA:  off, Vol: 100(100)

.-= terminal control keys =-
[s] or [ ]   interrupt/restart playback (i.e. '(un)pause')
[f]         next track
[d]         previous track
[b]         back to beginning of track
[p]         loop around current position (don't combine with output buffer)
[.]         forward
[.]         rewind
[:]         fast forward
[.]         fast rewind
[>]        fine forward
[<]        fine rewind
[+]         volume up
[-]         volume down
[r]         RVA switch
[v]         verbose switch
[l]         list current playlist, indicating current track there
[t]         display tag info (again)
[m]         print MPEG header info (again)
[h]         this help
[q]         quit
[c] or [C]   pitch up (small step, big step)
[x] or [X]   pitch down (small step, big step)
[w]         reset pitch to zero

Frame# 4008 [ 1], Time: 01:44.69 [00:00.02], RVA:  off, Vol: 114(114)
[1:44] Decoding of class1.mp3 finished.

Terminal control enabled, press 'h' for listing of keys and functions.

Playing MPEG stream 2 of 5: class.mp3 ...
Title:  onclassical_demo_latry_bach_fugue_in_d_major_bhw-532_live_cut-version
Artist:
Comment:
Album:
Year:
Genre: Unknown
Frame# 5891 [ 1], Time: 02:33.88 [00:00.02], RVA:  off, Vol: 114(114)
[2:33] Decoding of class.mp3 finished.

```

**RYСУNEK 3.6.** Lista klawiszy sterujących aplikacji *mpg123* w oknie terminala

Tabela 3.3 zawiera prostą listę klawiszy sterujących wraz dodatkowymi opisami.

**TABELA 3.3.** Klawisze sterujące aplikacją mpg123

Klawisz sterujący	Działanie
<i>s</i>	Przełączenie pauzy (naciśnij raz, aby wstrzymać odtwarzanie, naciśnij ponownie, aby kontynuować)
<i>f</i>	Następny utwór
<i>d</i>	Poprzedni utwór
<i>b</i>	Powrót na początek utworu
<i>p</i>	Odtwarzanie w pętli wokół bieżącej pozycji
<i>.</i>	Przewijanie w przód
<i>,</i>	Przewijanie wstecz
<i>:</i>	Szybkie przewijanie w przód
<i>;</i>	Szybkie przewijanie wstecz
<i>+</i>	Zwiększenie głośności
<i>-</i>	Zmniejszenie głośności
<i>r</i>	Przełącznik względnego dopasowania głośności (ang. <i>Relative Volume Adjustment, RVA</i> ), wykorzystujący znacznik MPEG ID3V2 do regulacji głośności albumu
<i>v</i>	Przełączanie wyświetlania szczegółowych informacji
<i>l</i>	Wyświetlenie bieżącej listy odtwarzania
<i>t</i>	Wyświetlenie informacji o tagu pliku
<i>m</i>	Wyświetlenie informacji z nagłówka MPEG
<i>h</i>	Pomoc
<i>q</i>	Wyjście
<i>c</i>	Niewielkie podwyższenie tonu
<i>C</i>	Duże podwyższenie tonu
<i>x</i>	Niewielkie obniżenie tonu
<i>X</i>	Duże obniżenie tonu
<i>w</i>	Reset wysokości tonu

Tak się składa, że odtworzenie wszystkich plików MP3 w bieżącym katalogu jest bardzo proste po wprowadzeniu następującego polecenia:


```
sudo -vC *.mp3
```

Parametr *v* w powyższym poleceniu powoduje wyświetlenie szczegółowych informacji, a *C* przełącza aplikację w tryb pełnej kontroli, w którym działają wszystkie klawisze sterujące. Gwiazdka *\** w powyższym poleceniu jest symbolem wieloznacznym zlecającym aplikacji odtworzenie wszystkich plików MP3 znajdujących się w bieżącym katalogu. Podsumowując, jest to bardzo rozbudowany, a jednocześnie całkiem prosty odtwarzacz MP3, dobrze nadający się do zastosowania w naszym projekcie. Teraz potrzebujemy jedynie kilku przycisków do wybierania utworów i kodu Python do automatyzacji czynności.

## Wejścia przycisków sprzętowych

Zróbmy przerwę w omawianiu oprogramowania i sprawdźmy, jak podłączyć przyciski sprzętowe do płytki RasPi i jak urządzenie ma reagować na naciśnięcie przycisku przez użytkownika. Jak pamiętasz z rozdziału 2., każdy pin GPIO może być ustawiony w trybie wejścia lub wyjścia. W projekcie migacza LED ustawiliśmy pin nr 18 w trybie wyjścia. W tym rozdziale ustawimy pin

nr 23 jako wejście, a pin nr 18 pozostawimy w trybie wyjścia. Naszym celem jest zapalenie diody LED dołączonej do pinu nr 18 w chwili naciśnięcia przycisku dołączonego do pinu nr 23. Do sterowania pinem nr 23 użyjemy napięcia 3,3 V. Warto przypomnieć sobie ostrzeżenie z rozdziału 1.

**OSTRZEŻENIE:**  Wszystkie piny GPIO są przystosowane do napięcia 3,3 V, a nie 5 V. Na płycie RasPi nie ma zainstalowanych zabezpieczeń przeciwprzepięciowych. Jeżeli niechcący podasz napięcie 5 V na pin GPIO, uszkodzisz płytke.

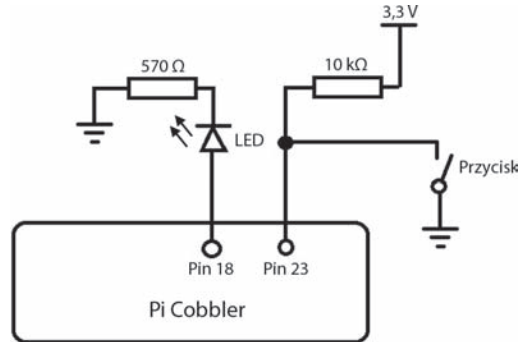
Na nakładce Pi Cobbler jest dostępne napięcie 3,3 V i nietrudno jest nieumyślnie podłączyć je zamiast napięcia 3,3 V. Dlatego bądź szczególnie ostrożny i sprawdź dwa razy połączenia, zanim podłączysz zasilanie.

Połącz diodę LED, przycisk i dodatkowe rezystory w testowy obwód zgodnie z rysunkiem 3.7. Wykorzystaj nielutowaną płytkę prototypową, ponieważ łatwo można na niej umieścić nakładkę Pi Cobbler i inne elementy.

Poniżej przedstawiony jest kod Python obsługujący ten mały obwód. Program jest dostępny pod nazwą *Test\_Switch.py* na stronie internetowej książki.

**UWAGA:** 

W operacji porównywania stosowane są dwa znaki równości, na przykład `GPIO.input(23) == False`. Jeżeli się pomyliś i wpiszesz tylko jeden znak, powstanie instrukcja przypisania, która może, ale nie musi, być potraktowana jako wartość logiczna. Wynikiem błędnej instrukcji `GPIO.input(23) = False` prawdopodobnie będzie `False`, ponieważ w ten sposób nie można programowo ustawić pinu w trybie wyjścia. W każdym razie program będzie działał, ale niezgodnie z oczekiwaniami. Jest to błąd logiczny, często popełniany przez początkujących programistów.



**RYSunEK 3.7.** Testowy obwód z przyciskiem dołączonym do wejścia

#### Test\_Switch.py

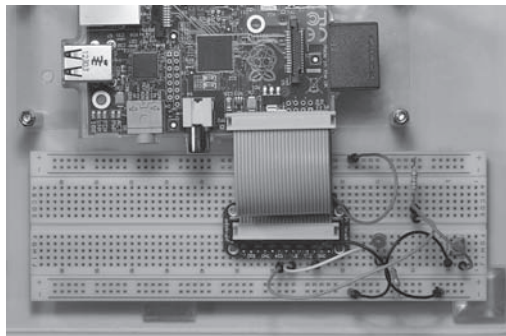
```
# Import biblioteki GPIO umożliwiającej dostęp do pinów.
import RPi.GPIO as GPIO
# Zastosowanie numerów pinów według BCM.
GPIO.setmode(GPIO.BCM)
# Ustawienie pinu 18 jako wyjścia.
GPIO.setup(18, GPIO.OUT)
# Ustawienie pinu 23 jako wejścia.
GPIO.setup(23, GPIO.IN)
# Początek nieskończonej pętli.
while True:
    # Sprawdzenie, czy został naciśnięty przycisk.
    if GPIO.input(23) == False:
        # Ustawienie stanu wysokiego na pinie.
        GPIO.output(18, GPIO.HIGH)
    # Przycisk nienaciśnięty, ustawienie stanu niskiego na pinie.
    else:
        GPIO.output(18, GPIO.LOW)
```

W powyższym programie użyłem instrukcji warunkowej `if/else` do sprawdzania, czy pin nr 23 ma stan niski, czyli `False`. Jeżeli tak, wówczas na pinie nr 18 ustawiany jest stan wysoki (`GPIO.HIGH`).



W przeciwnym wypadku instrukcja warunkowa przejdzie bezpośrednio do części `else`, gdzie ustawiany jest stan niski (`GPIO.LOW`) pinu nr 18.

Uruchom program i sprawdź, czy po każdorazowym naciśnięciu przycisku zapali się dioda LED i będzie świecić przez cały czas, gdy przycisk będzie naciśnięty. Rysunek 3.8 przedstawia gotowy obwód zbudowany na płytce prototypowej z płytką RasPi dołączoną do nakładki Pi Cobbler za pomocą płaskiej taśmy przyłączeniowej.



**RYСУNEK 3.8.** Gotowy obwód testowy z przyciskiem

## Myślenie w kategoriach RasPi

Zawsze podczas tworzenia nowego programu musisz pamiętać o prędkości działania płytki RasPi. Pętla `while` w powyższym programie jest wykonywana ponad milion razy na sekundę, co oznacza, że wszystkie polecenia wewnątrz pętli są nieustannie powtarzane. Z tego powodu użyłem instrukcji warunkowej `if/else` do ciągłego ustawiania na pinie wartości `HIGH` przez cały czas, gdy naciśnięty jest przycisk. Gdybym pominął część `else`, dioda LED zapaliłaby się na stałe po pierwszym naciśnięciu przycisku. Spróbuj usunąć część `else`, aby potwierdzić takie zachowanie układu. Ale dla dioda świeci się cały czas?

Odpowiedź jest prosta: bez części `else` nie będzie polecenia wyłączającego diodę. Na pinie zostanie zapamiętany taki stan, jaki został ustawiony za pomocą ostatnio użytego polecenia. To właśnie miałem na myśli, pisząc o myśleniu w kategoriach RasPi. Przekonałem się, że początkujący (a czasem nie całkiem początkujący) programiści mikroprocesorów mogą być zdezorientowani aspektem programowania w czasie rzeczywistym. Sterowanie fizycznym sprzętem może być wyzwaniem. Abyś uniknął niespodzianek, postaram się opisać w projektach zawartych w tej książce te **ambitne** obszary programowania.

## Wymagania projektu odtwarzacza MP3

Pierwszym krokiem podczas tworzenia każdego projektu jest określenie wymagań, które muszą być spełnione, aby zadowolić użytkownika. W tym przypadku wymagania są raczej proste. Oto one:

- użycie aplikacji *mpg123* jako odtwarzacza plików MP3,
- odtwarzanie plików (utworów) MP3 umieszczonych w określonym katalogu,
- użycie monitora komputera do wyświetlania danych pliku (utworu),
- użycie gniazda audio 3,5 mm jako wyjścia sygnału,
- użycie klawiatury do wprowadzania poleceń sterujących aplikacją *mpg123* podczas odtwarzania utworu,
- użycie jednego z trzech przycisków do odtwarzania wybranego utworu.

Zbudowanie tego projektu jest prostym procesem, ponieważ większość elementów i oprogramowanie zostały już omówione.

## NAPISANIE PROGRAMU

Kluczową kwestią jest uruchomienie w programie Python aplikacji *mpg123* ze wszystkimi potrzebnymi argumentami. Na szczęście język Python oferuje funkcję o nazwie `system()` umożliwiającą realizację tego zadania w całkiem prosty sposób. Funkcja ta jest częścią pakietu *os*. Przedstawiony

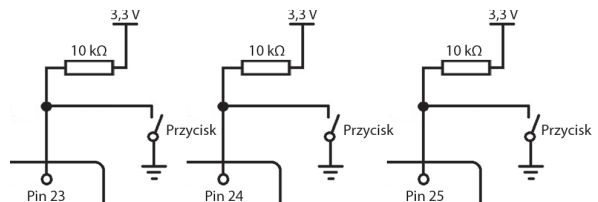
niziej program, który napisałem w języku Python, nosi nazwę *MP3\_Player.py* i można go pobrać ze strony internetowej książki.

### MP3\_Player.py

```
# Import pakietu os zawierającego funkcję system().
import os
# Import biblioteki GPIO umożliwiającej dostęp do pinów.
import Rpi.GPIO as GPIO
# Zastosowanie numerów pinów według BCM.
GPIO.setmode(GPIO.BCM)
# Ustawienie pinu 23 jako wejścia.
GPIO.setup(23, GPIO.IN)
# Ustawienie pinu 24 jako wejścia.
GPIO.setup(24, GPIO.IN)
# Ustawienie pinu 25 jako wejścia.
GPIO.setup(25, GPIO.IN)
# Sprawdzenie, że wybrane jest wyjście analogowe.
os.system('amixer cset numid=3 1')
while True:
    # Sprawdzenie, czy został naciśnięty przycisk dołączony do pinu nr 23.
    if(GPIO.input(23) == False):
        # Uruchomienie aplikacji mpg123 odtwarzającej plik class.mp3 w trybie pełnej
        # kontroli z klawiatury i z wyświetlaniem szczegółowych informacji na monitorze.
        os.system('mpg123 -vC class.mp3')
    # Sprawdzenie, czy został naciśnięty przycisk dołączony do pinu nr 24.
    if(GPIO.input(24) == False):
        # Uruchomienie aplikacji mpg123 odtwarzającej plik class1.mp3 w trybie pełnej
        # kontroli z klawiatury i z wyświetlaniem szczegółowych informacji na monitorze.
        os.system('mpg123 -vC class1.mp3')
    # Sprawdzenie, czy został naciśnięty przycisk dołączony do pinu nr 25.
    if(GPIO.input(25) == False):
        # Uruchomienie aplikacji mpg123 odtwarzającej plik Maid.mp3 w trybie pełnej
        # kontroli z klawiatury i z wyświetlaniem szczegółowych informacji na monitorze.
        os.system('mpg123 -vC Maid.mp3')
```

## SCHEMAT ODTWARZACZA MP3

Na rysunku 3.9 pokazany jest sposób podłączenia wszystkich pinów i przycisków. Możesz po prostu podłączyć dwa dodatkowe przyciski i rezystory do istniejącego obwodu zbudowanego według rysunku 3.7. Jeżeli chcesz, możesz również pozostawić podłączoną diodę LED, jednak w tym obwodzie nie jest ona uwzględniona.



RYСУNEK 3.9. Schemat odtwarzacza MP3

Rysunek 3.10 przedstawia gotowy obwód zbudowany na płytce prototypowej z płytką RasPi dołączoną do nakładki Pi Cobbler za pomocą płaskiej taśmy przyłączeniowej i głośnikiem Veho 360° dołączonym do wyjścia analogowego. Trzy przyciski służące do wybierania trzech utworów są umieszczone po prawej stronie płytki prototypowej.



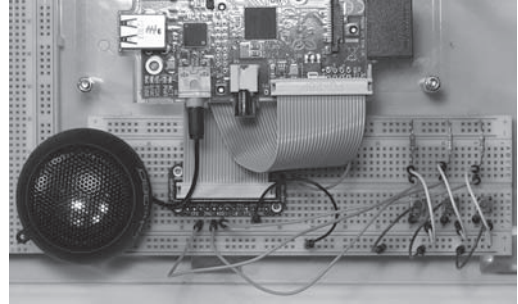
## TESTOWANIE PROJEKTU

Umieść program *MP3\_Player.py* w katalogu *pi*, a następnie otwórz terminal i wpisz polecenie:

```
sudo python MP3_Player.py
```

Odkryłem, że próba uruchomienia tego programu w powłoce Python IDLE może skutkować jego niestabilnym działaniem, ponieważ aplikacja *mpg123* wymaga bezpośredniego dostępu do klawiatury, który powłoka może blokować. Po uruchomieniu programu dostępne są wszystkie polecenia klawiatury wymienione w tabeli 3.3. Poeksperymentuj z różnymi poleceniami, aby przekonać się do wielu funkcji oferowanych przez aplikację *mpg123*.

Rysunek 3.11 zawiera widok ekranu pokazujący działający odtwarzacz MP3. W ostatnim wierszu widoczna jest informacja z tagu nagłówka formatu MPEG (*Moving Picture Experts Group*, grupa ekspertów ds. ruchomych obrazów).



RYСУNEK 3.10. Gotowy odtwarzacz MP3

```
pi@raspberrypi ~ $ sudo python MP3_Player.py
numid=3,iface=MIXER,name='PCM Playback Route'
; type=INTEGER,access=rw-----,values=1,min=0,max=2,step=0
; values=1
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
version 1.14.4; written and copyright by Michael Hipp and others
free software (LGPL/GPL) without any warranty but with best wishes
Decoder: ARM

Terminal control enabled, press 'h' for listing of keys and functions.

Playing MPEG stream 1 of 1: class.mp3 ...

MPEG 2.0, Layer: III, Freq: 22050, mode: Joint-Stereo, modext: 2, BPF : 209
Channels: 2, copyright: No, original: Yes, CRC: No, emphasis: 0.
Bitrate: 64 kbit/s Extension value: 0
Title:  onclassical_demo_latry_bach_fugue_in_d_major_bwv-532_live_cut-version
Artist:
Comment:                                     Album:
Year:                                         Genre:  Unknown
Frame#  680 [ 5212], Time: 00:17.76 [02:16.15], RVA:  off, Vol: 100(100)█
```

RYСУNEK 3.11.  
Widok odtwarzanego  
pliku MP3

## ROZBUDOWA PROJEKTU

Zaprezentowałem Ci podstawowy odtwarzacz MP3, który może być rozbudowany na wiele sposobów w celu spełnienia przyszłych wymagań. Wyświetlacz może być zmieniony ze standardowego monitora na kilkuwierszowy **wyświetlacz ciekłokrystaliczny (LCD)**. Na razie jednak należy skupić się na prostszym zagadnieniu użycia RasPi do sterowania urządzeniem.

Innym aspektem, nad którym możesz się zastanowić, jest użycie osobnego przycisku pełniącego określoną funkcję, na przykład przejście do następnego utworu. Aby to teraz zrobić, wystarczy naciśnięcie klawisz *f*. Implementacja przycisku z osobną funkcją pociąga za sobą wiele pracy programistycznej, ponieważ aplikacja *mpg123* oczekuje naciśnięcia klawisza sterującego, czyli wejścia logicznego zwanego *stdin* lub standardowego urządzenia wejściowego, w tym przypadku klawiatury. Zmiana lub równoległe użycie urządzenia wejściowego nie jest trywialnym zadaniem, dlatego uznałem, że nie należy opisywać go na początku książki.

Dodawanie utworów MP3 do odtwarzania jest raczej proste, wystarczy tylko, że skopiujesz je za pomocą portu RasPi USB do domyślnego katalogu. Program musi być zmieniony tak, aby

uwzględniał nowe nazwy plików. Można jednak tak go zmienić, aby wyświetlał wszystkie utwory MP3 zapisane w katalogu, dzięki czemu będziesz mógł wybrać jeden z nich do odtworzenia, co stanowi istotę listy odtwarzania.

Oczywiście, do zasilenia RasPi możesz użyć baterii, dzięki czemu urządzenie będzie całkowicie przenośne, o ile użyjesz wyświetlacza LCD do interakcji z użytkownikiem.

## Podsumowanie

Rozpocząłem rozdział od omówienia nakładki prototypowej Pi Cobbler, będącej nieocenionym narzędziem umożliwiającym eksperymentowanie z większością pinów GPIO dostępnych na płytce RasPi.

W opisie oprogramowania Linux ALSA pokazałem, jak jest obsługiwany dźwięk w systemie operacyjnym RasPi. Instalacja ALSA jest konieczna, abyś mógł korzystać z aplikacji *mpg123* stanowiącej podstawę projektu odtwarzacza MP3. Wprowadziłem Cię również w podstawy formatów plików dźwiękowych, w tym MP3, wykorzystanego do zakodowania muzyki odtwarzanej w tym projekcie. Zrobiłem też krótką dygresję na temat charakterystyki częstotliwościowej układu dźwiękowego RasPi wyznaczonej z wykorzystaniem różowego szumu.

Następnie pokazałem, jak podłączyć przyciski do sterowania płytką RasPi, wykorzystując piny GPIO jako wejścia. Przyciski zostały użyte jako część projektu odtwarzacza MP3.

Zamieściłem wyczerpujący opis aplikacji *mpg123* wraz ze szczegółową listą wszystkich dostępnych klawiszy sterujących tym wszechstronnym narzędziem.

Rozdział zakończyłem krótkim opisem, jak rozbudować projekt MP3, aby bardziej przypominał nowoczesny przenośny odtwarzacz.

# Skorowidz

## A

adapter Bluetooth, 181  
adres  
  adaptera Bluetooth, 183  
  IP, 20  
  IP płytki, 147  
akumulator  
  litowo-jonowy, 172  
  litowo-polimerowy, 180  
analiza  
  akustyczna, 46  
  częstotliwościowa, 45  
  danych sejsmicznych, 96  
  komunikatu GPS, 82  
  pakietów, 150  
analyzer  
  1-Wire, 146  
  pakietów, 161, 162  
API, Application Program Interface, 15  
aplikacje gpsd, 83 *Patrz także*  
  program, oprogramowanie  
automatyka domowa, 102

## B

biblioteka GPIO, 37  
Bluetooth, 181  
budowa  
  czujnika sejsmicznego, 88  
  koła, 171  
  stacji meteorologicznej, 139

## C

certyfikat  
  FCC, 211  
  próbki uranowej, 195  
ciśnienie atmosferyczne, 149  
CSI-2, 20  
częstotliwość  
  Nyquista, 91  
  próbekowania, 91  
czujnik  
  ciśnienia, 142  
  dźwięku, 126

laserowy, 118, 125  
ruchu, 115, 127  
światła, 62  
  U1, 141  
  wilgotności, 142  
czytnik NFC, 129

## D

DAC, Digital-to-Analog Converter, 91  
dawka promieniowania, 190  
demon ntpd, 211  
detektor sejsmiczny, 88  
diagram  
  modułu Z-Wave ZW3102N, 106  
  przetwornika MCP3008, 91  
  sterownika aparatu fotograficznego, 57  
dioda LED, 39  
dokładność zegara, 202  
donationware, 162  
dostawca nieprodukcyjny, 14  
dostęp do  
  pinów GPIO, 36  
  płytki 8 Channel I/O, 166  
DRAM, Dynamic Random Access Memory, 15  
DSI, Display Serial Interface, 20  
duplex, 92  
dylatacja czasu, 74  
działanie  
  czytnika NFC, 129  
  migawki aparatu, 56  
  sieci Z-Wave, 103  
  systemu GPS, 72, 74  
dzielnik, 201

## E

edytor nano, 40, 122  
ekran konfiguracyjny, 27  
elektroniczna lampa błyskowa, 58  
elementy interfejsu licznika GM, 193  
emulacja RS-232, 77

## F

fala sejsmiczna, 88  
formaty plików dźwiękowych, 47  
fotografia poklatkowa, 55, 68  
fototranzystor, 62  
funkcje pakietu Motion, 120  
FUSE, 158

## G

głośnik stereofoniczny, 44  
GPS, Global Positioning System, 71  
GPU, Graphic Processing Unit, 14  
graficzny interfejs użytkownika, GUI, 79  
GUI, Graphical User Interface, 79

## H

half-duplex, 143  
hasło, 27  
hierarchia serwerów NTP, 207

## I

I2C, Inter-Integrated Circuit, 24  
identyfikatory RFID, 129  
IDLE, 31  
instalacja  
  libnfc, 131  
  owfs, 155  
interfejs  
  API, 15  
  CSI-2, 20  
  I2C, 24  
  Pi Cobbler  
    lista pinów, 64  
    podłączenie modułu dźwiękowego, 62  
    połączenia, 63  
  SPI, 24, 91, 92  
  SSI, 24  
  UART, 25

interfejsy optoizolacyjne, 60  
 IP, Internet Protocol, 20  
 izolator optotriakowy, 60

**J**

język Python, 30

**K**

kamera, 123  
 kamera internetowa, 117  
 karta pamięci SD, 16, 28  
 klawisze sterujące aplikacją  
 mpg123, 49  
 klucz Z-Stick, 108  
 kod  
 pakietu libnfc, 132  
 wysyłający powiadomienia,  
 124  
 kodek H264, 15  
 komponenty elektryczne  
 pojazdu, 179  
 komunikacja  
 I2C, 177  
 NFC, 129, 130  
 komunikat GPS, 82  
 koncentrator USB, 20  
 konfiguracja  
 owfs, 155  
 pakietu Motion, 120  
 płytki RasPi, 26  
 portu szeregowego, 77  
 sieci, 105  
 SIS, 105  
 zegara RTC, 205  
 kontroler  
 sieci Z-Wave, 105  
 Wiimote, 170, 183  
 konwerter  
 cyfrowo-analogowy, 91  
 HDMI, 18  
 USB/1-Wire, 140  
 krzywa kalibracyjna, 191

**L**

licznik Geigera-Mullera, 189,  
 192, 197, 199  
 linie  
 danych SPI, 92  
 sygnalizacyjne SPI, 24

Linux, 29  
 LKM, Loadable Kernel Module,  
 45, 204  
 logowanie SSH, 109  
 LSB, Least Significant Bit, 93  
 lutowanie, 35, 36

**Ł**

ładowalne jądro modułu, 204  
 łącznik prototypowy Pi Cobbler,  
 43

**M**

mechanizm wielodostępu, 103  
 miganie diody, 41  
 mikroprocesor BCM2835, 14, 22  
 moduł  
 analizujący 1-Wire, 146  
 dźwiękowy, 61  
 laserowy, 62  
 LKM, 206  
 zasilający, 140  
 ZW3102N, 107  
 moduły stacji meteorologicznej,  
 140  
 monitor  
 analogowy, 19  
 pakietów GPS, 85  
 monitorowanie ruchu, 161  
 montowanie  
 czujnika laserowego, 118  
 ładowarki, 173  
 monitora, 175  
 mrem, 195  
 MSB, Most Significant Bit, 93

**N**

najbardziej znaczący bit, 93  
 najmniej znaczący bit, 93  
 napięcie  
 sterujące lampą błyskową, 59  
 sygnału logicznego, 22  
 narzędzie  
 apt, 39  
 pakujące, 38  
 NFC, Near Field  
 Communication, 129  
 NMEA, 81  
 NTP, Network Time Protocol,  
 201

**O**

obraz systemu Wheezy Raspian,  
 109  
 obsługa licznika Geigera, 198  
 obwód  
 wysokiego napięcia, 61  
 wzmacniacza, 90  
 odbiornik Ultimate GPS, 75, 78  
 odczyt danych  
 meteorologicznych, 147  
 odtwarzanie pliku  
 mp3, 47  
 wav, 47  
 okno  
 aplikacji  
 cgps, 83  
 gpsmon, 85  
 putty, 109  
 Tera Term, 150  
 xgps, 84  
 serwera lightscontrol, 111  
 terminala Raspberry Pi, 110  
 z modułem LKM, 206  
 z obrazem wideo, 123  
 okres  
 drgań wahadła, 89  
 połowicznego rozpadu, 189  
 oprogramowanie, *Patrz także*  
 program  
 Bluetooth, 181  
 I2C, 181  
 Motion, 117, 120  
 NTP, 208  
 Open Z-Wave, 109  
 RTC, 203  
 optoizolator NTE3041, 60  
 oscylator, 201  
 oscylator zegara RTC, 202

**P**

pakiet  
 alsa-utils, 43–45  
 cwiid, 184  
 libnfc, 131, 133, 136, 138  
 LKM, 45  
 Motion, 120  
 python-smbus, 177  
 Z-Wave, 103  
 pamięć  
 DRAM, 15  
 flash, 15  
 SD, 15

- parametry
  - dokładnościowe, DOP, 85
  - serwomechanizmu, 176
  - urządzenia Z-Wave, 112
- piny
  - dotatkowe GPIO, 23
  - GPIO, 22
  - GPIO, 21, 22, 36
  - interfejsu, 64
  - złącza UART, 76
- plik
  - alarm.py, 125
  - CameraControl.py, 66
  - cmdline.txt, 131
  - extract\_data.py, 136
  - geiger.py, 195
  - inittab, 132
  - INSTALL.txt, 38
  - modules, 205
  - motion.conf, 121
  - MP3\_Player.py, 52
  - ntp.conf, 208, 209
  - ow\_eight.py, 161
  - owfs.conf, 147, 148
  - rc.local, 206
  - Robot\_Car.py, 186, 187
  - Seismic\_Monitor.py, 98
  - SoundModTest.py, 67
  - Test\_ADC.py, 93, 96
  - Test\_Switch.py, 50
  - TimeLapse.py, 68
  - uart\_posix.c, 133
  - Wiimote\_Test.py, 184, 185
- pliki
  - dźwiękowe, 46
  - pakietu libnfc, 133
- plytka
  - 1-Wire 8 Channel I/O, 153, 167
  - 8 Channel I/O, 153
  - Arduino, 25
  - ładowarki, 172
  - modułu zasilającego, 141
  - odbiornika Ultimate GPS, 75
  - Pi Plate, 35
  - PN532/NFC, 130
  - RasPi, 15, 16
  - sterownika
    - serwomechanizmów, 174, 177
    - zegara RTC, 203
  - pobór prądu, 23
- podłączenie
  - detektora sejsmicznego, 95
  - diody, 40
  - przewodu USB/TTL, 76
- podwozie, 170–172
- pojazd samobieżny, 169
- polecenia systemu Linux, 30
- polecenie
  - apt-get, 38
  - date, 207
  - import, 40
  - killall gpsd, 210
  - killOZW, 114
  - lsusb, 118, 156
  - print, 32
  - ROM, 150
- połączenia interfejsu, 63
- połączenie
  - SPI, 92
  - z czujnikiem laserowym, 119
  - z siecią, 108
  - ze sterownikiem PN532, 135
- pomiar
  - aktywności próbki, 196
  - aktywności tła, 196
  - ciśnienia, 142, 149
  - radioaktywności, 190
  - temperatury, 148, 149
  - wilgotności, 148, 149
- potencjometr, 90
- powiadomienie e-mail, 123
- powłoka IDLE, 33
- poziomy radioaktywności radonu, 190
- prawo Ohma, 39
- prędkość fali, 88
- procesor
  - Broadcom VideoCore IV, 15
  - graficzny, GPU, 14
- program, *Patrz także*
  - oprogramowanie
    - amixer, 45
    - cgps, 83
    - CuteCom, 81
    - DrGPIO, 164
    - FUSE, 157
    - gpsmon, 85
    - IDLE, 31
    - libnfc, 131, 132
    - migacza LED, 40, 41
    - modprobe, 45, 204
    - mpg123, 48
    - nfc-poll, 136
- ntpdate, 211
- obsługujący czujnik laserowy, 125
- odtworacza MP3, 51
- ow\_eight, 161
- owfs, 139, 147, 155
- putty, 109
- realizujący protokół szeregowy, 76
- sejsmografu, 99
- sterownika aparatu fotograficznego, 68
- sterujący pojazdem, 185
- systemu NFC, 136
- tar, 38
- Tera Term, 77, 80, 150
- terminalowy ZOC, 77
- Test\_ADC.py, 95
- Test\_File\_ADC.py, 96
- Test\_Switch.py, 50
- testowy Python, 160
- testujący moduł dźwiękowy, 66
- Wiimote\_Test.py, 184
- xgps, 84
- programy testujące, 68
- projekt
  - automatyka domowa, 101
  - czytnik NFC/RFID, 129
  - domowy
    - system bezpieczeństwa, 117
  - Migacz LED, 35
  - Odtwarzacz MP3, 43
  - pojazd samobieżny, 169, 179
  - sejsmograf, 87
  - serwer sygnału czasu, 201
  - stacja meteorologiczna, 139
  - sterownik aparatu fotograficznego, 55
  - sterownik logiczny, 153
  - system GPS, 71
  - wykrywacz radonu, 189
- promieniowanie jonizujące, 189
- protokół
  - 1-Wire, 142
  - DHCP, 20
  - I2C, 24, 181
  - I2C, 176
  - NMEA, 81
  - NTP, 207
  - SPI, 24
  - UART, 25, 78, 131

przedłużacz Pi Cobbler, 43  
 przepływ danych, 130, 143  
 przerwania, 23  
 przetaktowanie, overclocking, 15  
 przetwornik  
   analogowo-cyfrowy, 143  
   MCP3008, 95  
 pula serwerów, 208  
 pulpit  
   interfejsu graficznego, 28  
   systemu Occidentalis, 29  
 PWM, Pulse Width Modulation, 175

## R

Radioaktywność, 189  
 radon, 189  
 Raspberry Pi, 13  
 REST, 166  
 rodzaje pamięci, 15  
 router Netgear, 123  
 rozdzielczość HD, 15  
 rozszerzenie pinów GPIO, 23

## S

schemat  
   blokowy  
     interfejsu licznika GM, 193  
     pojazdu, 179  
     sieci 1-Wire, 155  
     stacji meteorologicznej, 139  
     systemu NFC/RFID, 129  
     układu DS1307, 202  
     układu DS18B20, 145  
     licznika GM K2645, 192  
     odtwarzacza MP3, 52  
     regulatora napięcia, 179  
     wejść płytki 8 Channel I/O, 154  
 SD, Secure Digital, 15  
 sejsmograf, 95  
   bramowy, 89  
   z odwróconym wahadłem, 89  
 sejsmologia, 87  
 serwer  
   lightscontrol, 110  
   NTP, 201, 207, 208  
   owfs, 147, 150, 166  
 serwer  
   WebIOPi, 166  
   WWW, 163

serwomechanizm, 171, 175  
 sieć  
   mieszana Z-Wave, 101  
   Wi-Fi, 123  
   Z-Wave, 103, 104  
 silniki napędowe, 170  
 skala Richtera, 87  
 SoC, System on a Chip, 13  
 specyfikacja płytki 8 Channel I/O, 154  
 SPI, Serial Peripheral Interface, 24, 92  
 sprawdzanie daty, 206  
 SSH, Secure Shell, 109  
 SSI, Synchronous Serial Interface, 24  
 stacja  
   lutownicza, 36  
   meteorologiczna, 139, 147  
 standard  
   HDMI, 18  
   HDMI-CEC, 18  
   OpenGL ES 2.0, 15  
 sterowanie  
   migawką, 57  
   pojazdem, 187  
   smartfonem, 162, 164  
 sterownik  
   ALSA, 44  
   aparatu fotograficznego, 64  
   PL2303, 77  
   PN532, 135  
   serwomechanizmu, 175  
 strefa bezpieczeństwa, 118  
 strona  
   podstawowego serwera, 115  
   serwera lightscontrol, 112  
   serwera WebIOPi, 166  
   startowa serwera owfs, 148, 150  
   z czynnościami, 113  
   z grupami urządzeń, 113  
 strumień danych GPS, 80, 81  
 sufiks .py, 40  
 superużytkownik, 29  
 sygnał PWM, 175  
 system  
   GPS, 71, 74  
   operacyjny Android, 162  
   operacyjny Linux, 29  
 systemy automatyki domowej, 102  
 szeregowa transmisja danych, 142

szum  
   biały, 45  
   różowy, 46

## Ś

środowisko IDLE, 31

## T

tag TPV, 84  
 technologia Z-Wave, 103  
 temperatura, 147  
 terminal  
   LXDE, 32, 37  
   Raspberry Pi, 110  
 test  
 komunikacji I2C, 177  
 testowanie  
   licznika GM, 194  
   odbiornika GPS, 79  
   odtwarzacza MP3, 53  
   połączeń interfejsu, 65  
   płytki 8 Channel I/O, 158  
   przetwornika MCP3008, 93  
   serwera WWW, 164  
   wejść, 160, 165  
   wyjść, 159, 164  
 TFT, Thin-Film Transistor, 169  
 TPV, Target, Position, Velocity, 71  
 transmisja  
   danych odbiornika, 78  
   UART, 76  
 tranzystor MOSFET, 145  
 trioda, 60  
 trzęsienia ziemi, 87  
 typy komunikatów NMEA GPS, 82

## U

UART, 25, 76, 131  
 układ  
   DS1307 DIP, 201  
   DS18B20, 145  
   DS2438A, 144  
   Maxim DS18B20, 141, 144  
   Maxim DS2438A, 143  
   NTE3041, 60  
   SoC, 14  
   SOIC, 143  
   Z-WAVE, 105  
 uruchomienie serwera NTP, 208

urządzenia  
 Bluetooth, 184  
 NFC, 130  
 NFC/RFID, 129  
 sieciowe, 104

urządzenie  
 docelowe, 129  
 inicjujące, 129  
 master, 143  
 slave, 143  
 Z-Wave, 101, 112

ustawienia konfiguracyjne, 27

użytkownik  
 pi, 29  
 root, 29

**W**

wahadło, 89

warstwa łącza, 103

warstwy  
 modelu sieci ISO, 103  
 stratum, 207

wątek, 120

wejścia  
 przycisków sprzętowych, 49  
 swobodne, 165

wejście  
 niezasilane, 154  
 zasilane, 154

wilgotność, 148

wirtualny port szeregowy, 77

wtyk  
 RJ12, 140  
 RJ45, 140

wydajność urządzenia, 14

wykrywacz radonu, 197

wykrywani  
 kolizji, 103  
 modułów 1-Wire, 158  
 płytki zegara RTC, 204  
 ruchu, 121, 122  
 sterownika  
 serwomechanizmów, 178

wymagania systemu sterownika,  
 56

wystukiwanie bitów, 25

wysyłanie powiadomień, 124

wyświetlacz  
 ciekłokrystaliczny, 53  
 TFT, 169  
 danych GPS, 83

wzmacniacz, 90

wzmacniacz operacyjny, 90

**Z**

zapis danych do pliku, 96

zasilanie  
 monitora, 180  
 pasożytnicze, 145  
 szyny 1-WIRE, 145  
 wsteczne, 179

zatrzymanie  
 demona gpsd, 210  
 serwera, 114

zdalny dostęp, 166

zegar  
 czasu rzeczywistego, 201  
 RTC, 201

złącza  
 RASPI, 16  
 rozszerzeń, 21

złącze  
 1-Wire, 139  
 audio, 19  
 do sterowania migawką, 57  
 DVI, 18  
 DSI, 20  
 Ethernet, 19  
 GPIO, 20, 21  
 HDMI, 17  
 kompozytowe, 18  
 lampy błyskowej, 59  
 prototypowe, 43  
 typu gorąca stopka, 59  
 UART, 76  
 USB, 19  
 VGA, 18  
 wideo, 19  
 zasilania micro USB, 17  
 zwiększanie wydajności, 14





# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

**Szalony  
Geniusz**

# RASPBERRY PI

## NIESAMOWITE PROJEKTY

**Prawdziwy komputer wielkości karty kredytowej?** Tak, to możliwe! Raspberry Pi to mały komputer o ogromnym potencjale, w dodatku naprawdę tani. Mało kto wie, że został on zaprojektowany z myślą o nauce programowania dla dzieci. Obecnie Pi sprawdza się w roli serwera WWW, odtwarzacza filmów lub sterownika automatyki. Masz pomysł na własny projekt i szukasz interesujących przykładów? Trafisz na doskonałą książkę poświęconą tej niezwykłej platformie!

**W trakcie lektury poznasz budowę Pi**, przygotujesz go do pracy i pierwszego uruchomienia. To dopiero wstęp do niesamowitej przygody! Z kolejnych rozdziałów dowiesz się, jak zbudować migacz LED, odtwarzacz MP3, sterownik aparatu fotograficznego oraz system GPS. Ponadto przy niewielkim nakładzie pracy będziesz mógł opracować system automatyki domowej, stację meteorologiczną oraz system zabezpieczeń domu. Książka ta jest bogatym źródłem praktycznych projektów, które zapewnią Ci wiele godzin świetnej zabawy!

Sięgnij po tę książkę i:

- uruchom platformę Raspberry Pi i poznaj jej budowę
- zbuduj system automatyki domowej
- opracuj własną stację meteorologiczną
- przygotuj własny serwer czasu rzeczywistego (RTC)
- baw się świetnie przez długie godziny!

### Fascynujące projekty dla platformy Pi!

Patronat medialny:

[mikrokontrolery.blogspot.com](http://mikrokontrolery.blogspot.com)



**helion.pl**  
księgarnia  
internetowa

Nr katalogowy: 23576

Księgarnia internetowa:  
<http://helion.pl>

Zamówienia telefoniczne:  
**0 801 339900**  
**0 601 339900**



Sprawdź najnowsze promocje:  
• <http://helion.pl/promocje>  
Książki najchętniej czytane:  
• <http://helion.pl/bestsellery>  
Zamów informacje o nowościach:  
• <http://helion.pl/nowości>

Helion SA  
ul. Kościuszki 1c, 44-100 Gliwice  
tel.: 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYŚCI

ISBN 978-83-246-9174-6



Cena: 39,90 zł

Informatyka w najlepszym wydaniu

9 788324 691746