



```
IMPORT RANDOM
```

Raspberry Pi

Przewodnik dla programistów Pythona

```
WORDS = ['CHICKEN', 'DOG', 'CAT', 'MOUSE', 'F
```

```
YES_REMAINING = 14
```

```
GUESS_ALLOWED_LETTERS = ""
```

```
DEF PLAY():
```

Zaprogramuj i rozwijaj własne projekty!

```
WORD = PICK_A_WORD()
```

```
WHILE TRUE:
```

```
GUESS = GET_GUESS(WORD)
```

```
IF F GUESS(GUESS, WORD):
```



```
WORD)
```

```
BREAK
```

```
DEF PICK_A_WORD():
```

```
WORD_POSITION = RANDOM.RANDINT(0, LE
```

```
RETURN WORDS[WORD_POSI
```

```
DEF GET_GUESS(WORD):
```

Simon Monk



Tytuł oryginału: Programming the Raspberry Pi: Getting Started with Python

Tłumaczenie: Jacek Janusz

ISBN: 978-83-246-8709-1

Original edition copyright © 2013 by The McGraw-Hill Companies.

All rights reserved.

Polish edition copyright © 2014 by HELION SA.

All rights reserved.

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.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: 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/rasppy.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/rasppy>

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
Wstęp	11
Rozdział 1. Wprowadzenie	13
Co to jest Raspberry Pi?	13
Co można zrobić przy użyciu Raspberry Pi?	14
Zapoznanie się z platformą Raspberry Pi	15
Uruchamianie platformy Raspberry Pi	16
<i>Kupowanie niezbędnych elementów</i>	16
<i>Łączenie wszystkiego w całość</i>	22
Uruchamianie systemu	22
Podsumowanie	24
Rozdział 2. Pierwsze kroki	25
Linux	25
Pulpit	25
Internet	27
Wiersz poleceń	27
<i>Poruszanie się po systemie za pomocą terminala</i>	29
<i>Polecenie sudo</i>	30
Aplikacje	30
Zasoby internetowe	32
Podsumowanie	32
Rozdział 3. Podstawy języka Python	33
IDLE	33
<i>Wersje języka Python</i>	33
<i>Powłoka języka Python</i>	34
<i>Edytor</i>	34
Liczby	36
Zmienne	37
Pętle for	38
Symulacja rzutów kostką do gry	39

Instrukcja if	41
<i>Porównywanie</i>	42
<i>Bycie logicznym</i>	42
<i>Instrukcja else</i>	43
Instrukcja while	44
Podsumowanie	45
Rozdział 4. Łańcuchy, listy i słowniki	47
Teoria łańcuchów	47
Listy	49
Funkcje	51
Wisielec	52
Słowniki	58
Krotki	59
<i>Wielokrotne przypisywanie</i>	60
<i>Wiele wartości powrotnych</i>	60
Wyjątki	61
Podsumowanie funkcji	61
<i>Liczby</i>	62
<i>Łańcuchy</i>	62
<i>Listy</i>	64
<i>Słowniki</i>	64
<i>Konwersje typów</i>	65
Podsumowanie	66
Rozdział 5. Moduły, klasy i metody	67
Moduły	67
<i>Użycie modułów</i>	67
<i>Przydatne biblioteki języka Python</i>	68
<i>Instalowanie nowych modułów</i>	69
Programowanie zorientowane obiektowo	70
Definiowanie klas	71
Dziedziczenie	72
Podsumowanie	74
Rozdział 6. Pliki i internet	75
Pliki	75
<i>Odczytywanie plików</i>	75
<i>Odczytywanie dużych plików</i>	77
<i>Zapisywanie plików</i>	78
<i>System plików</i>	78
Serializacja	79
Internet	80
Podsumowanie	82
Rozdział 7. Graficzne interfejsy użytkownika	83
Tkinter	83
Witaj, programisto	83
Konwerter temperatury	84

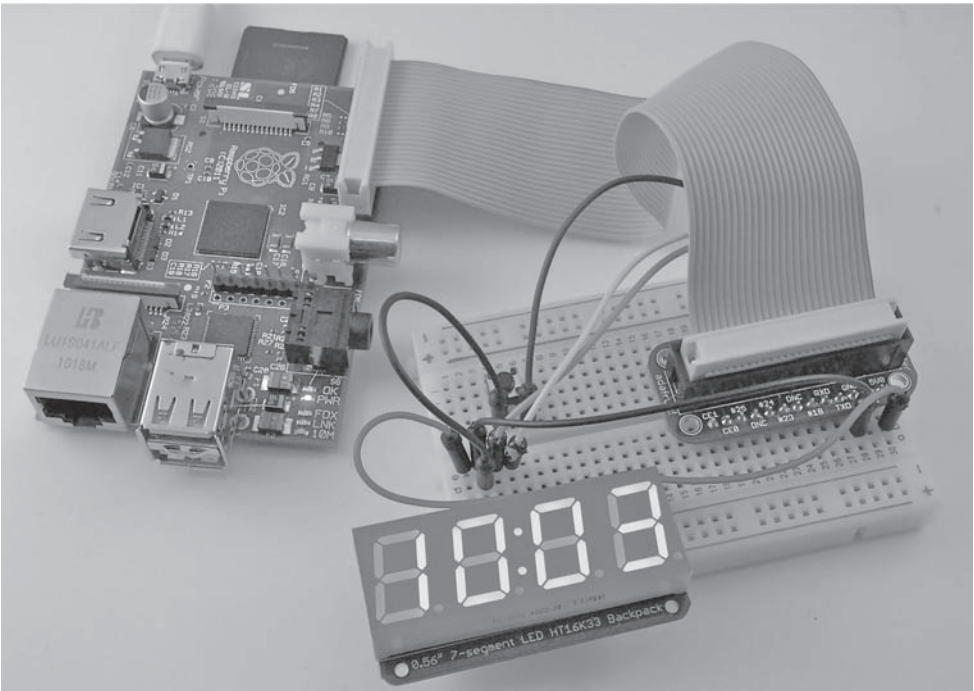
Inne widżety GUI	88
<i>Pole wyboru</i>	88
<i>Lista</i>	88
<i>Spinbox</i>	89
<i>Układy elementów</i>	89
<i>Pasek przesuwania</i>	92
Okna dialogowe	93
<i>Okno wyboru koloru</i>	94
<i>Okno wyboru pliku</i>	95
Menu	95
Obszar rysowania	96
Podsumowanie	97
Rozdział 8. Programowanie gier	99
Co to jest pygame?	99
Witaj, pygame	100
Malinowa gra	101
<i>Sterowanie myszą</i>	101
<i>Jedna malina</i>	103
<i>Wykrywanie złapania i punktacja</i>	104
<i>Odmierzanie czasu</i>	105
<i>Mnóstwo malin</i>	106
Podsumowanie	108
Rozdział 9. Współpraca ze sprzętem	109
Piny złącza GPIO	109
Bezpośrednie podłączenie do pinów GPIO	110
Karty rozszerzeń	111
<i>Pi Face</i>	111
<i>Slice of PI/O</i>	112
<i>RaspiRobotBoard</i>	113
<i>Gertboard</i>	114
Karty prototypowania	114
<i>Pi Cobbler</i>	115
<i>Pi Plate</i>	115
<i>Humble Pi</i>	116
Arduino i Raspberry Pi	117
<i>Współpraca Arduino i Raspberry Pi</i>	118
Podsumowanie	120
Rozdział 10. Projekt z wykorzystaniem karty prototypowania (zegar)	121
Elementy, których potrzebujesz	122
Montaż sprzętu	122
Oprogramowanie	124
Etap drugi	126
Podsumowanie	128

Rozdział 11. RaspiRobot	129
Elementy, których potrzebujesz	130
Etap pierwszy: podstawowa wersja pojazdu	130
<i>Montaż sprzętu</i>	<i>130</i>
<i>Oprogramowanie</i>	<i>135</i>
Etap drugi: dodanie czujnika odległości i wyświetlacza	136
<i>Krok nr 1 — podłączenie szeregowego adaptera czujnika odległości</i>	<i>136</i>
<i>Krok nr 2 — podłączenie wyświetlacza</i>	<i>136</i>
<i>Krok nr 3 — aktualizacja oprogramowania</i>	<i>138</i>
<i>Krok nr 4 — uruchomienie</i>	<i>138</i>
<i>Zmodyfikowane oprogramowanie</i>	<i>139</i>
Podsumowanie	140
Rozdział 12. Co dalej?	141
Zasoby związane z systemem Linux	141
Zasoby związane z językiem Python	141
Zasoby związane z platformą Raspberry Pi	142
Inne języki programowania	143
<i>Język Scratch</i>	<i>143</i>
<i>Język C</i>	<i>143</i>
Aplikacje i projekty	144
<i>Centrum multimedialne (Raspbmc)</i>	<i>144</i>
<i>Automatyka domowa</i>	<i>145</i>
Podsumowanie	146
Skorowidz	147

Rozdział 10.

Projekt z wykorzystaniem karty prototypowania (zegar)

W tym rozdziale zajmiemy się czymś, co moglibyśmy nazwać nadmiernie skomplikowanym projektem cyfrowego zegara LED. Użyjemy platformy Raspberry Pi, karty Cobbler firmy Adafruit, uniwersalnej płytki prototypowej oraz czterocyfrowego wyświetlacza LED (patrz rysunek 10.1).



Rysunek 10.1. Zegar LED zrealizowany przy użyciu platformy Raspberry Pi

Na pierwszym etapie projektowania urządzenie umożliwi jedynie pokazywanie godzin. Drugi etap pozwoli jednak na ulepszenie projektu poprzez dodanie przycisku, którego naciśnięcie będzie powodować przełączanie trybu wyświetlania pomiędzy godzinami (i minutami), sekundami oraz datą.

Elementy, których potrzebujesz

Aby ukończyć projekt, będziesz potrzebować elementów przedstawionych w tabeli 10.1. Podano w niej nazwy producentów sprzętu oraz przykładowe sklepy internetowe, w których możesz kupić oryginalne części lub ich zamienniki.

Tabela 10.1. Lista elementów dla projektu zegara LED

Nazwa elementu	Producenci sprzętu i przykładowe sklepy internetowe	Orientacyjna cena
Raspberry Pi	Farnell, RS Components http://htpcentre.pl/sklep-2/raspberry-pi/ http://www.conrad.pl	160 zł
Pi Cobbler	Adafruit (nr produktu: 914) http://pl.farnell.com/	25 zł
Wyświetlacz 4-cyfrowy LED z interfejsem I2C	Adafruit (nr produktu: 880) http://nettigo.pl/products/317	70 zł
Uniwersalna płytki prototypowa	Adafruit (nr produktu: 64), SparkFun (SKU PRT-00112), Maplin (AG09K) http://botland.com.pl	15 zł
Przewody montażowe	Adafruit (nr produktu: 758), SparkFun (SKU PRT-08431), Maplin (FS66W) http://nettigo.pl/products/category/11 http://sklep.avt.pl	15 zł
Przełącznik naciskany, montowany na płycie drukowanej*	Adafruit (nr produktu: 367), SparkFun (SKU COM-00097), Maplin (KR92A) http://www.conrad.pl/ http://sklepelektroniczny.com	8 zł

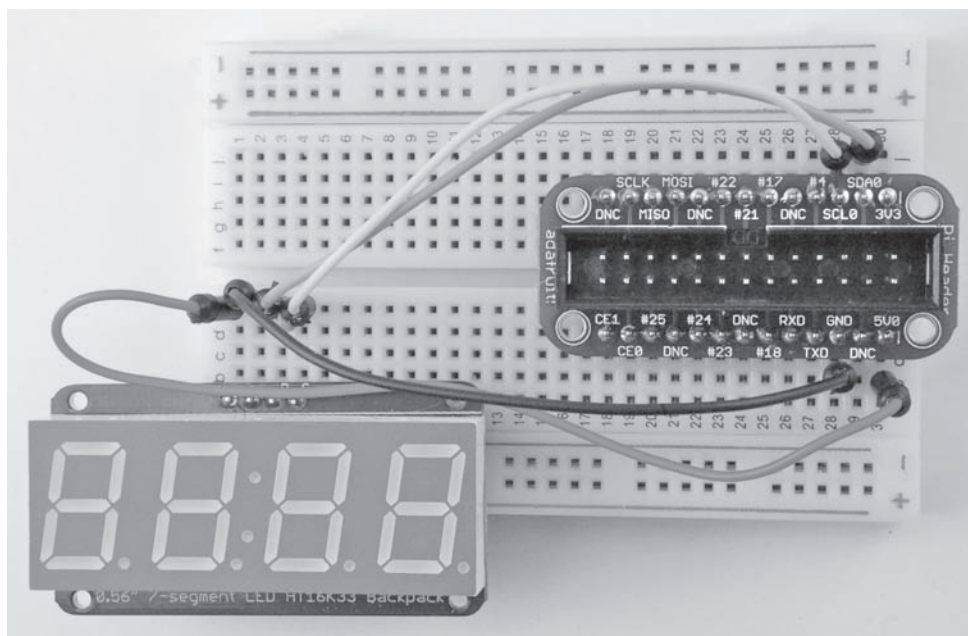
* Opcjonalne. Wymagane jedynie w etapie drugim.

Montaż sprzętu

Karta Pi Cobbler oraz moduł wyświetlacza z firmy Adafruit są udostępniane w postaci zestawów, które muszą zostać wcześniej odpowiednio zmontowane przy wykorzystaniu lutownicy, by mogły być używane. Operacje lutowania są proste do wykonania, a na stronie firmy Adafruit są dostępne dokładne instrukcje, prezentujące szczegółowo wszystkie wymagane działania.

Wyświetlacz zawiera tylko cztery piny (VCC, GND, SDA i SCL), którymi można go podłączyć do płytki uniwersalnej. Umieść go w ten sposób, by pin VCC był na płytce w rzędzie nr 1.

Karta Cobbler posiada 26 pinów, lecz w naszym projekcie użyjemy tylko kilku z nich. Powinna zostać ona umieszczona po przeciwnej stronie płytki uniwersalnej, a co najmniej tak daleko od wyświetlacza, by żaden z jej pinów nie znajdował się w pobliżu niego. Gniazdo karty Cobbler ma wycięcie po jednej stronie, aby zapewnić, że kabel płaski zostanie do niego włożony we właściwy sposób. Powinno być ono skierowane w stronę górnej krawędzi płytki uniwersalnej, jak przedstawiono na rysunku 10.2.



Rysunek 10.2. Umieszczenie elementów na płytce uniwersalnej

Poniżej otworów w płytce uniwersalnej znajdują się łączniki pozwalające na połączenie ze sobą pięciu otworów w określonym rzędzie. Ponieważ płytka została położona na boku, na rysunku 10.2 rzędy są widoczne jako pionowe.

Rysunek 10.2 przedstawia płytkę uniwersalną z czterema pinami wyświetlacza po lewej stronie oraz kartą Cobbler po prawej. Jeśli chcesz skorzystać z instrukcji przedstawionych w tym rozdziale, musisz umieścić swoje moduły w taki sam sposób, jak zaprezentowano na tym rysunku.

UWAGA Przewody można będzie dużo łatwiej podłączyć do płytki uniwersalnej, jeśli kabel płaski nie zostanie dołączony do karty Cobbler.

Połączenia, które muszą zostać wykonane, są opisane w tabeli 10.2.

Schemat kolorów przedstawiony w powyższej tabeli jest jedynie sugestią. Powszechnie stosuje się jednak kolor czerwony dla oznaczenia przewodu dodatniego, a czarny lub niebieski dla połączeń o potencjale zerowym.

Tabela 10.2. Lista połączeń do wykonania

Sugerowany kolor przewodu	Z	Do
Czarny	Pin GND karty Cobbler	Pin GND wyświetlacza (drugi pin od lewej strony)
Czerwony	Pin 5V0 karty Cobbler	Pin VCC wyświetlacza (pierwszy pin od lewej strony)
Pomarańczowy	Pin SDA0 karty Cobbler	Pin SDA wyświetlacza (trzeci pin od lewej strony)
Żółty	Pin SCL0 karty Cobbler	Pin SCL wyświetlacza (pierwszy pin od prawej strony)

OSTRZEŻENIE W tym projekcie podłączamy moduł wyświetlacza 5 V do platformy Raspberry Pi, która zasadniczo wykorzystuje napięcie 3,3 V. Możemy to jednak wykonać w sposób bezpieczny, ponieważ użyty tutaj moduł wyświetlacza działa jedynie jako urządzenie podrzędne i dlatego nasłuchuje na portach SDA i SCL. Inne urządzenia I2C mogą być jednak nadrzędnymi i w tych przypadkach zasilanie ich napięciem 5 V może spowodować uszkodzenie systemu Raspberry Pi. Wynika stąd, że przed każdym podłączeniem urządzenia I2C do Raspberry Pi powinienes się upewnić, że dokładnie wiesz, co zamierzasz zrobić.

Teraz możemy podłączyć kartę Cobbler do platformy Raspberry Pi przy użyciu kabla płaskiego, dostarczonego w zestawie z tą kartą. Wcześniej jednak urządzenie Raspberry Pi powinno zostać wyłączone. Gniazdo na karcie Cobbler jest zabezpieczone przed niepoprawnym włożeniem kabla, lecz takiej ochrony nie posiada platforma Raspberry Pi. Upewnij się więc, że czerwony przewód na kablu zostanie umieszczony przy krawędzi płytki urządzenia Raspberry Pi, jak pokazano na rysunku 10.1.

Włącz urządzenie Raspberry Pi. Jeśli diody LED na jego płycie nie zaświecą się, wyłącz je od razu i sprawdź wszystkie połączenia.

Oprogramowanie

Wszystko zostało podłączone, a platforma Raspberry Pi uruchomiła się poprawnie. Jednakże na wyświetlaczu nic się nie pojawia, ponieważ nie napisaliśmy jeszcze żadnego oprogramowania, które by nim sterowało. Zamierzamy rozpocząć od prostego zegara, który będzie po prostu wyświetlał systemowy czas z Raspberry Pi. Platforma Raspberry Pi nie posiada zegara czasu rzeczywistego informującego o bieżącym czasie. Może jednak automatycznie zsynchronizować swój czas z sieciowym serwerem czasu, jeśli zostanie podłączona do internetu.

Raspberry Pi wyświetla czas w prawym dolnym narożniku ekranu. Bez dostępu do internetu możesz samodzielnie ustawić czas przy użyciu poniższego polecenia (użyj oczywiście bieżącej daty!):

```
sudo date -s "Aug 24 12:15"
```

Będziesz musiał jednak powtarzać tę czynność po każdym restarcie. Dużo lepiej jest więc podłączyć Raspberry Pi do internetu.

Jeśli zsynchronizujesz czas z siecią, może się zdarzyć, że minuty wyświetlą się prawidłowo, lecz godziny będą błędne. Oznacza to, iż Twój system Raspberry Pi „nie wie”, w jakiej strefie czasowej się znajduje. Problem może zostać usunięty poprzez użycie poniższego

polecenia, które otwiera okno pozwalające na wybranie kontynentu oraz miasta odpowiadających Twojej strefie czasowej:

```
sudo dpkg-reconfigure tzdata
```

W momencie pisania niniejszej książki dystrybucja Raspbian Wheezy wymagała wcześniejszego podania kilku instrukcji, aby umożliwić korzystanie z magistrali I2C przez program w języku Python, który zostanie przez nas napisany. Jest prawdopodobne, że kolejne wersje dystrybucji Raspbian (i również inne dystrybucje) mogą mieć już odpowiednio skonfigurowane porty, przez co wykonanie przedstawionych poleceń nie będzie już konieczne. Obecnie należy jednak wprowadzić następujące instrukcje:

```
sudo apt-get install python-smbus
sudo modprobe i2c-dev
sudo modprobe i2c-bcm2708
```

UWAGA *Może się okazać, że po każdym restarcie urządzenia Raspberry Pi będzie zawsze konieczne wykonanie dwóch ostatnich instrukcji z powyższej listy.*

Obecnie Raspberry Pi pokazuje już poprawny czas, a magistrala I2C jest dostępna. Możemy więc rozpocząć pisanie programu w języku Python, który będzie wysyłał informacje o czasie do wyświetlacza. Aby uprościć ten proces, stworzyłem w języku Python moduł biblioteczny, który jest przeznaczony do obsługi tego typu wyświetlaczy. Można go pobrać ze strony <http://i2c7segment.googlecode.com/files/i2c7segment-1.0.tar.gz>.

Podobnie jak miało to miejsce z innymi modułami, należy zapisać plik, rozpakować go przy użyciu polecenia `tar -xzf`, a następnie wykonać poniższe polecenie pozwalające na jego instalację dla języka Python 2:

```
sudo python setup.py install
```

Faktyczny program obsługi zegara znajduje się w pliku `10_01_clock.py` na stronie <http://helion.pl/ksiazki/rasppy.htm>. Poniżej przedstawiono jego listing:

```
import i2c7segment as display
import time

disp = display.Adafruit7Segment()

while True:
    h = time.localtime().tm_hour
    m = time.localtime().tm_min
    disp.print_int(h * 100 + m)
    disp.draw_colon(True)
    disp.write_display()
    time.sleep(0.5)
    disp.draw_colon(False)
    disp.write_display()
    time.sleep(0.5)
```

Program jest krótki i prosty. Pętla działa przez cały czas, odczytując bieżącą godzinę i minutę, a następnie prezentując je w odpowiednim miejscu wyświetlacza. Jest to realizowane poprzez mnożenie godzin przez 100, by wyświetlić je po lewej stronie, a następnie dodanie minut, które będą się pojawiać po prawej.

Biblioteka `i2c7segment` wykonuje większość działań za nas. Jest ona wykorzystywana w wierszach, które definiują, co powinno zostać wyświetlone (`print_int` oraz `draw_colon`). Funkcja `write_display`, pochodząca z tej biblioteki, aktualizuje czas przedstawiany na wyświetlaczu.

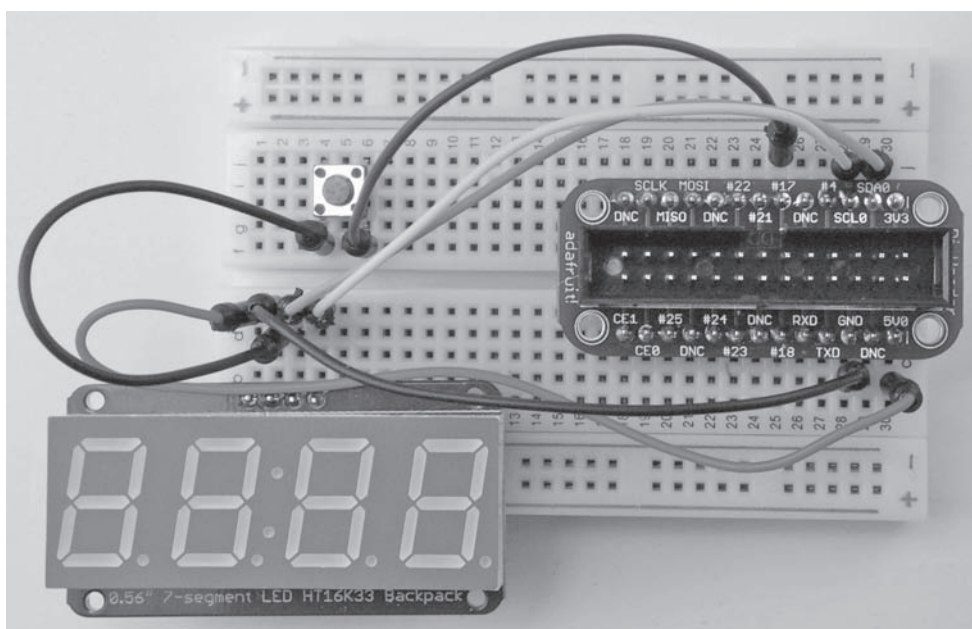
Migotanie przecinka zostało zrealizowane poprzez wyświetlanie go i wygaszanie co pół sekundy. Dostęp do portu I2C jest możliwy jedynie dla administratorów, dlatego poniższe polecenie musisz wykonać z wysokimi uprawnieniami:

```
sudo python 10_01_clock.py
```

Jeśli wszystko zostało wykonane prawidłowo, wyświetlacz powinien pokazać bieżący czas.

Etap drugi

Gdy podstawowa funkcjonalność została już zaimplementowana, spróbujmy teraz ulepszyć sprzęt i oprogramowanie w ten sposób, by kolejne naciśnięcie przycisku powodowało cykliczną zmianę trybu prezentowania informacji, przełączając między wyświetlaniem godzin i minut, sekund oraz daty. Na rysunku 10.3 przedstawiono płytke uniwersalną z umieszczonym na niej przyciskiem, a także dwoma nowymi połączeniami. Zauważ, że obecnie modyfikujemy układ elektroniczny etapu pierwszego poprzez dodanie do niego przycisku; nic innego nie zostaje zmienione.



Rysunek 10.3. Dodawanie przycisku do układu

UWAGA Zamknij system operacyjny oraz wyłącz zasilanie platformy Raspberry Pi, zanim zaczniesz modyfikować układ elektroniczny na płytce uniwersalnej.

Przycisk ma cztery końcówki i musi zostać umieszczony we właściwym położeniu. Jeśli tak się nie stanie, przełącznik przez cały czas będzie rozpoznawany jako włączony. Kończówki powinny wystawać z boków przełącznika w kierunku dolnej i górnej krawędzi płytki (rysunek 10.3). Nie martw się jednak, jeśli go niewłaściwie umieścisz — nic się nie uszkodzi, jedynie wyświetlacz będzie w sposób ciągły zmieniał tryb wyświetlania bez naciskania przycisku.

Aby podłączyć przycisk do układu, potrzebne są dwa dodatkowe przewody. Jeden prowadzi z przełącznika (patrz rysunek 10.3) do pinu GND wyświetlacza. Drugi jest podłączony do pinu o numerze 17 w karcie Cobbler. W rezultacie przy każdym naciśnięciu przycisku port GPIO o numerze 17 w Raspberry Pi będzie zwierany do masy.

Odpowiedni program znajduje się w pliku *10_02_fancy_clock.py*. Poniżej prezentujemy jego kod źródłowy:

```
import i2c7segment as display
import time
import RPi.GPIO as io

switch_pin = 17
io.setmode(io.BCM)
io.setup(switch_pin, io.IN, pull_up_down=io.PUD_UP)
disp = display.Adafruit7Segment()

time_mode, seconds_mode, date_mode = range(3)
disp_mode = time_mode

def display_time():
    h = time.localtime().tm_hour
    m = time.localtime().tm_min
    disp.print_int(h * 100 + m)
    disp.draw_colon(True)
    disp.write_display()
    time.sleep(0.5)
    disp.draw_colon(False)
    disp.write_display()
    time.sleep(0.5)

def display_date():
    d = time.localtime().tm_mday
    m = time.localtime().tm_mon
    disp.print_int(d * 100 + m)
    disp.draw_colon(True)
    disp.write_display()
    time.sleep(0.5)

def display_seconds():
    s = time.localtime().tm_sec
    disp.print_str('----')
    disp.print_int(s)
    disp.draw_colon(True)
    disp.write_display()
    time.sleep(0.5)

while True:
    key_pressed = not io.input(switch_pin)
    if key_pressed:
        disp_mode = disp_mode + 1
        if disp_mode > date_mode:
            disp_mode = time_mode
        if disp_mode == time_mode:
```

```

display_time()
elif disp_mode == seconds_mode:
    display_seconds()
elif disp_mode == date_mode:
    display_date()

```

Po pierwsze: należy zauważyć, że ze względu na to, iż wymagany jest dostęp do pinu nr 17 GPIO w celu sprawdzenia, czy przycisk został wciśnięty, konieczne staje się użycie biblioteki `RPi.GPIO`. Korzystaliśmy już z niej w rozdziale 5. — w przykładzie instalowania modułu. Jeśli więc nie masz jeszcze zainstalowanej biblioteki `RPi.GPIO`, wróć do rozdziału 5. i wykonaj teraz tę operację.

Przy użyciu poniższej instrukcji pin przycisku definiujemy jako port wejściowy:

```
io.setup(switch_pin, io.IN, pull_up_down=io.PUD_UP)
```

To polecenie również aktywuje wewnętrzny rezystor pull-up, który zapewnia, że port wejściowy będzie się zawsze znajdował pod napięciem 3,3 V (w stanie wysokim), z wyjątkiem sytuacji, w której naciśnięto przycisk, a przez to zwarto wejście do masy.

Większość działań, które realizowano w pętli, została umieszczona w funkcji zwanej `display_time`. Dodano również dwie nowe funkcje: `display_seconds` i `display_date`. Nie wymagają one dodatkowych komentarzy.

Aby wiedzieć, jaki tryb jest aktywny, przy użyciu poniższych wierszy kodu dodaliśmy kilka nowych zmiennych:

```

time_mode, seconds_mode, date_mode = range(3)
disp_mode = time_mode

```

W pierwszym wierszu przypisujemy każdej z trzech zmiennych inną liczbę. W drugim inicjalizujemy zmienną `disp_mode` wartością `time_mode`, której później użyjemy w głównej pętli.

Główna pętla została zmodyfikowana w taki sposób, by sprawdzać, czy przycisk został wciśnięty. Jeśli jest on wciśnięty, wówczas następuje zwiększenie zmiennej `disp_mode` o 1 i zmiana trybu wyświetlania. Gdy osiągnie ona wartość maksymalną, jest ponownie inicjalizowana zmienną `time_mode`.

Występujące później bloki instrukcji `if` pozwalają na wybór i wywołanie odpowiedniej funkcji wyświetlającej, zależnej od bieżącego trybu.

Podsumowanie

Sprzęt używany w projektach może zostać dość łatwo zaadaptowany do innych zastosowań. Po odpowiedniej modyfikacji programu mógłbyś na przykład prezentować na wyświetlaczu również inne informacje. Oto kilka pomysłów:

- Bieżąca przepustowość połączenia z internetem (prędkość danych).
- Liczba nieprzeczytanych e-maili w skrzynce pocztowej.
- Liczba dni pozostałych do końca roku.
- Liczba osób odwiedzających stronę internetową.

W następnym rozdziale zajmiemy się kolejnym projektem sprzętowym. Tym razem będzie to poruszający się robot, którego „mózgiem” stanie się platforma Raspberry Pi.

Skorowidz

`__init__`, 71, 73, 85

A

a, 78

- `.append(x)`, 64
- `.count(x)`, 64
- `.index(x)`, 64
- `.insert(i, x)`, 64
- `.pop()`, 64
- `.remove(x)`, 64
- `.reverse()`, 64
- `.sort()`, 64

AbiWord, 31

`abs(x)`, 62

adapter Wi-Fi, 21

administrator, 30

and, 43

aplikacje, 30

- AbiWord, 31
- biblioteka pygame, 100
- do konwersji temperatury, 84
- Gnumeric, 31
- IDLE, 34
- LXTerminal, 28
- tworzenie w języku Python, 33
- zaktualizowanie bazy pakietów, 31
- Zlewozmywak kuchenny, 88

`append`, 78

`apt-get`, 30

Arch Linux ARM, 19

Arduino, 117

- oprogramowanie, 118

- współpraca z Raspberry Pi, 118

atrybuty

- `command`, 93

- `fill`, 93

- `sticky`, 91

automatyka domowa, 145

B

biblioteka

- `i2c7segment`, 126

- języka Python, 68

- Pi Face, 112

- pygame, 99

- PySerial, 133

- RaspiRobotBoard, 133

- RPi.GPIO, 74, 133, 128

bieżący katalog, 26

- polecenie zmiany, 29

`bin(x)`, 62

Biuro, 31

`blit`, 101

`boot_behaviour`, 23

`break`, 44

BROWSE, 89

C

C, 143

Canvas, 96

`cd`, 29

centrum przetwarzania danych, 16

`change directory`, 29

`clock`, 105

`clock.tick`, 105

`columnconfigure`, 91, 92

command, 93
 complex(r, i), 62
 convert, 72, 87, 101
 converters, 87
 copytree, 79
 curselection, 89
 czytanie z pliku, 75

D

d.clear(), 65
 date -s, 124
 datetime, 68
 def, 52
 definiowanie klas, 71
 del(), 64,65
 description, 72, 73
 dodanie nowego programu, 30
 dodatkowe pakiety, 31
 Don't Repeat Yourself, 44
 DoubleVar, 86
 dpkg-reconfigure tzdata, 125
 DRY, 44
 dystrybucje systemu Raspberry Pi Linux, 19
 dziedziczenie, 72

E

edytor IDLE, 35
 tworzenie pliku tekstowego, 76
 edytor nano, 35
 ekran konfiguracyjny, 23
 else, 43
 escape character, 62
 exit, 77, 102
 expand_rootfs, 23
 EXTENDED, 89

F

f.close(), 76
 False, 42
 fill, 93
 find, 55
 float(x), 65
 for, 38, 39
 Fundacja Raspberry Pi, 142
 strona domowa, 32

funkcje, 51
 abs(x), 62
 bin(x), 62
 check_for_catch, 104, 105
 complex(r, i), 62
 copytree, 79
 d.clear(), 65
 del(a[i:j]), 64
 del(d[key]), 65
 display, 104, 105
 display_date, 128
 display_seconds, 128
 display_time, 128
 exit, 102
 f.close(), 76
 find, 55
 float(x), 65
 get(key, default), 65
 get_guess, 54
 get_range, 139
 get_range_cm, 139
 hex(x), 62
 input, 55
 insert, 50
 int(x), 65
 is_caught, 106
 key in d, 65
 len, 49
 len(d), 65
 len(numbers), 50
 list(x), 65
 math.acos, 62
 math.asin, 62
 math.atan, 62
 math.cos, 62
 math.factorial(n), 62
 math.log(x), 62
 math.pow(x, y), 62
 math.sin, 62
 math.sqrt(x), 62
 math.tan, 62
 oct(x), 62
 pack, 85
 play, 53
 print_word_with_blanks, 54
 randint, 67
 raw_input, 55
 read, 78

- readline, 78
- rmtree, 79
- round(x, n), 62
- showinfo, 93
- sort, 50
- str, 54
- update, 106
- update_display, 139
- update_raspberry, 103, 104
- update_spoon, 103
- wm_title, 85
- write_display, 126
- funkcje przetwarzające
 - liczby, 62
 - listy, 64
 - łańcuchy, 63
 - słowniki, 65

G

- gcc, 144
- General Purpose Input/Output, 15, 109
- generowanie liczby losowej, 39
- Gertboard, 114, 115
- get, 89
- get(key, default), 65
- get_range, 139
- get_range_cm, 139
- glob, 79
- glob.glob, 79
- globbing, 79
- gniazdo micro-USB, 16
- gniazdo RJ-45, 15
- Gnumeric, 31
- GPIO, 15
- graficzny interfejs użytkownika, 83
 - lista, 88
 - menu, 95
 - obszar rysowania, 96
 - okna dialogowe, 93
 - pasek przesuwania, 92
 - pole wyboru, 88
 - siatka, 85
 - spinbox, 89
 - Tk, 83
 - Tkinter, 83
 - układy elementów, 89
 - widżety, 88
 - obszaru rysowania, 96
- grid, 85, 87

- grupy funkcji, 67
- GUI, 83
 - widżety, 88

H

- HDMI, 13
- hermetyzacja, 72
- hex(x), 62
- High-Definition Multimedia Interface, 13
- HTML, 80
- Humble Pi, 116
- Hypertext Markup Language, 80
- Hypertext Transfer Protocol, 80

I

- I2C, 110
- IDLE, 33
 - edytor, 34
 - menu Run, 36, 40
 - New Window, 34
 - powłoka języka Python, 34
 - tworzenie nowego pliku, 34
 - wczytywanie plików, 36
 - wersje języka Python, 33
 - zapisywanie programów, 34, 36
- IDLE 3, 33, 99
- if, 41
- ignore, 103
- import, 67
- importowanie biblioteki, 39
- init, 101
- initialize, 101
- input, 55
- insert, 50
- instalator modułu, 70
- install, 70, 119
- instalowanie nowych modułów, 69
- instancja klasy, 70
- instrukcja, 51
 - break, 44
 - clock.tick, 105
 - elif, 43
 - else, 43
 - except, 77
 - if, 41
 - logika, 42
 - porównywanie, 42
 - while, 44

int(x), 65
 interfejs
 Canvas, 96
 DVI, 13
 GPIO, 15
 HDMI, 15
 szeregowy, 110
 internet, 27, 80
 IntVar, 88
 is_caught, 106
 iteracje, 38

J

jedynka logiczna, 109, 112
 języki
 C, 143
 programowania, 33, 143
 Python, 33
 Scratch, 143

K

karta SD, 16, 18
 przygotowanie, 19
 karty prototypowania, 114
 Humble Pi, 116
 Pi Cobbler, 115
 Pi Plate, 115
 projekt zegara LED, 121
 karty rozszerzeń, 111
 Gertboard, 114
 Pi Face, 111
 RaspiRobotBoard, 113
 Slice of PI/O, 112
 karty z systemem Arduino, 117
 katalog
 config, 134
 domowy, 26
 pulpitu, 26
 roboczy, 29
 zakładanie dla programów języka Python, 35
 key in d, 65
 klasa, 70
 App, 85
 definiowanie, 71
 dziedziczenie, 72
 Entry, 88
 hermetyzacja, 72
 Label, 85, 88

nadrzędna, 73
 Raspberry, 106
 zamiana na moduł, 74
 klawiatura, 18
 klucz, 59
 komentarz, 40
 kompilator C, 144
 koncentrator USB, 21
 kontrolka
 listy, 88
 spinbox, 89
 konwencje nazewnicze, 37
 konwersje typów, 65
 Konwerter temperatury, 84
 kopiowanie plików, 79
 krotki, 59
 wiele wartości powrotnych, 60
 wielokrotne przypisywanie, 60

L

len, 49
 len(d), 65
 len(numbers), 50
 liczby, 36
 funkcje, 62
 liczby całkowite, 37
 liczby zmiennoprzecinkowe, 37
 o podwójnej precyzji, 86
 Linux, 25
 język C, 143
 zasoby, 141
 list, 29
 list(x), 65
 listingi kodów, 12
 listy, 49, 88
 funkcje, 64
 łańcuchów, 53
 łączenie, 50
 przypisanie wartości elementowi, 49
 sortowanie, 50
 tworzenie, 49
 usuwanie elementu, 50
 wstawianie elementu w określonej pozycji, 50
 wyodrębnienie elementów, 49
 złożone, 50
 ls, 29
 LXDE, 14
 LXTerminal, 28, 69

L

- łańcuchy, 47
 - funkcje, 62
 - łączenie, 49
 - stałe łańcuchowe, 62
- w programowaniu zorientowanym
 - obiekto, 70
 - wycinanie fragmentów, 48
 - wyświetlanie pierwszego znaku, 48

M

- magazyn sieciowy, 142
- Magician Chassis, 130, 131
- magistrala szeregową, 110
- Malinowa gra, 101
- math, 68
- math.acos, 62
- math.asin, 62
- math.atan, 62
- math.cos, 62
- math.factorial(n), 62
- math.log(x), 62
- math.pow(x, y), 62
- math.sin, 62
- math.sqrt(x), 62
- math.tan, 62
- mechanizm wyjątków, 61
- menedżer plików, 26, 35
 - python_games, 100
- menu, 95
- menu aplikacji, 28
- menu startowe
 - Biuro, 31
 - Internet, 27
 - menedżer plików, 35
 - Programowanie, 33
- message, 105
- metoda, 70, 71
 - przesyłanie, 73, 74
- metody
 - __init__, 71, 73, 85
 - columnconfigure, 92
 - convert, 72, 87
 - convert(), 101
 - curselection, 89
 - description, 72, 73
 - get, 89
 - grid, 85, 87

- init, 101
- pack, 84
- rowconfigure, 92
- set, 93
- showerror, 93
- showwarning, 93
- metody reprezentowania danych, 47
- Midori, 27
- MISO, 110
- MIT, 143
- modalne okna dialogowe, 93
- model A, 15
- model B, 15
- moduł, 67
 - funkcje, 68
 - instalowanie, 69
 - obsługi wyświetlacza, 125
 - użycie, 67
 - wyświetlacza, 124
 - zamiana klas na moduł, 74
- moduły
 - converters, 87
 - datetime, 68
 - math, 68
 - pickle, 69
 - pygame, 69, 101
 - pygame.locals, 102
 - random, 67
 - string, 68
 - tkinter, 69, 83
 - urllib.request, 69
- monitor, 18
- more INSTALL.txt, 70
- MOSI, 110
- MULTIPLE, 89
- mysz, 18

N

- nazewnictwo
 - funkcji, 52
 - zmiennych, 37
- not, 43

O

- obiekt
 - Frame, 85
 - Menu, 95
 - text, 105
 - Tk, 84

obszar powiadamiania, 104
 obszar rysowania, 96
 obudowa, 19
 Occidentalis, 19
 oct(x), 62
 odczytywanie plików, 75, 77
 odmierzenie czasu, 105
 offvalue, 88
 okno dialogowe, 93

- okno wyboru koloru, 94
- okno wyboru pliku, 95
- z alertem, 93

 onvalue, 88
 operator

- and, 43
- not, 43
- or, 43
- porównania, 42
- przypisania, 47

 oprogramowanie XBMC, 145
 or, 43

P

pack, 84, 85, 93
 pakiet

- glob, 79
- shutil, 78
- tkinter.colorchooser, 94
- tkinter.filedialog, 94, 95
- tkinter.messagebox, 93

 pakiet binarny, 69
 parametr, 39

- self, 71

 pasek przesuwania, 92
 pętla for, 38

- symulacja rzutów kostką do gry, 39

 pętla while, 103
 Pi Cobbler, 115, 116, 121, 122
 Pi Face, 111

- diody LED, 112
- przełączniki, 112

 Pi Plate, 115, 116
 pickle, 69
 pickling, 79
 pin, 109, 117

- bezpośrednie podłączenie, 110
- MISO, 110
- MOSI, 110

PWM, 110
 SCL, 110
 SCLK, 110
 SDA, 110
 platforma Raspberry Pi, 11, 13, 15

- aplikacje i projekty, 144
- Arduino, 118
- automatyka domowa, 145
- centrum multimedialne, 144
- czas systemowy, 124
- elementy zestawu, 17
- gniazdo micro-USB, 16
- karta SD, 16
- magazyn sieciowy, 142
- moduł wyświetlacza 5 V, 124
- podłączanie kabli, 124
- podłączenie do sieci bezprzewodowej, 21
- Raspbmc, 144
- sposoby użycia, 141
- uruchamianie, 16
- Wi-Fi, 15
- współpraca z Arduino, 119
- współpraca ze sprzętem, 109
- wykorzystanie, 14
- zasoby, 142
- zasoby internetowe, 32

 plik

- 04_01_list_and_for, 51
- 04_02_polite_function, 52
- 04_03_hello_n, 52
- 04_09_hangman_full_solution.py, 58
- 04_09_stats, 60
- 05_01_converter, 71
- 05_02_converter_offset_bad, 72
- 05_03_converters_final.py, 74
- 06_01_hangman_file.py, 76
- 06_02_hangman_file_try.py, 77
- 06_03_file_readline, 77
- 06_04_helion_scraping, 81
- 07_01_hello.py, 83
- 07_02_temp_framework.py, 84
- 07_03_temp_ui.py, 86
- 07_04_temp_final.py, 87
- 07_05_kitchen_sink.py, 88
- 07_06_resizing.py, 90
- 07_07_scrolling.py, 93
- 07_08_gen_dialogs.py, 93
- 07_09_color_chooser.py, 94
- 07_10_menus.py, 95

- 07_11_canvas.py, 96
- 08_01_hello_pygame.py, 100
- 08_02_rasp_game_mouse, 102
- 08_03_rasp_game_one.py, 103
- 08_04_rasp_game_scoring.py, 104
- 08_05_rasp_game_refactored.py, 106
- 08_06_rasp_game_final.py, 107
- 10_01_clock.py, 125
- 10_02_fancy_clock.py, 127
- 11_01_rover_basic.py, 133
- 11_02_rover_plus.py, 138
- 3_1_dice.py, 40
- 3_2_double_dice, 41
- 3_3_double_dice_solution.py, 43
- 3_4_double_dice_while, 44
- 3_5_double_dice_while_break, 44
- 4_4_hangman_words, 53
- 4_5_hangman_play.py, 54
- 4_6_hangman_get_guess.py, 55
- 4_7_hangman_print_word.py, 56
- 4_8_hangman_full.py, 57
- converters.py, 74
- hangman_words.txt, 76
- mylist.pickle, 79
- random.py, 67
- raspberry.jpg, 101
- raspirobot_basic.desktop, 134
- raspirobot_plus.desktop, 138
- pliki, 75
 - kopiowanie, 79
 - odczytywanie, 75, 77
 - przenoszenie, 79
 - serializacji, 80
 - system plików, 78
 - tryby otwarcia, 78
 - zapisywanie, 78
 - danych do pliku, 78
 - zawartości zmiennej, 79
- pobieranie danych ze stron internetowych, 81
- początkowa konfiguracja, 22
- podłączanie urządzeń, 109
- podwozie, 131
- pole wyboru, 88
- polecenia
 - apt-get, 30
 - blit, 101
 - break, 44
 - cd, 29
 - columnconfigure, 91
 - date -s, 124
 - dpkg-reconfigure tzdata, 125
 - exit, 77
 - for, 39
 - glob.glob, 79
 - import, 67
 - install, 70, 119
 - instalowanie biblioteki, 133
 - kontynuowania wiersza, 105
 - ls, 29
 - more INSTALL.txt, 70
 - pop, 50
 - print, 47
 - pwd, 29
 - random, 39
 - range, 39
 - read, 80
 - remove, 32
 - return, 52
 - rowconfigure, 91
 - shutil.copy, 79
 - shutil.move, 79
 - sudo, 30
 - try, 76
 - ustawianie czasu, 124
 - w edytorze IDLE, 35
 - w powłoce języka Python, 34
 - while, 44
 - znak dwukropka, 38
- pop, 50
- porównywanie, 42
- port USB
 - alternatywa, 109
 - wejściowy, 128
- powłoka języka Python, 34
 - liczby, 36
- print, 47
- print working directory, 29
- programowanie gier, 99
 - biblioteka pygame, 99
 - Malinowa gra, 101
 - obszar powiadamiania, 104
 - odmierzanie czasu, 105
 - punktacja, 104
 - sterowanie myszą, 101
 - wykrywanie złapania, 104
 - wyświetlanie komunikatu na ekranie, 105

programowanie zorientowane obiektowo, 50, 70
 instancja, 70
 klasa, 70
 metoda, 70

programy
 Konwerter temperatury, 84
 Malinowa gra, 101
 symulacja rzutów kostką do gry, 39
 Wisielec, 52
 Witaj, programisto, 83
 zegar, 121
 Zlewozmywak kuchenny, 88

programy użytkowe powłoki systemowej, 78

przechowywanie plików, 27

przeglądarka internetowa, 27

przenoszenie plików, 79

pulpit, 25

punktacja, 104

pwd, 29

PWM, 110

pygame, 69, 99
 inicjalizacja, 101
 oficjalna strona, 108
 RaspiRobot, 132
 wygląd aplikacji, 100
 wyświetlanie tekstu, 105
 zasoby, 142
 zegar, 105

pygame.locals, 102

PySerial, 119
 RaspiRobot, 133

Python, 33
 biblioteka pygame, 99
 biblioteki, 68
 działania arytmetyczne, 34
 edytor, 34
 funkcje, 51
 IDLE, 33
 instalowanie modułów, 69
 instrukcja else, 43
 instrukcja if, 41
 instrukcja while, 44
 internet, 80
 iteracje, 38
 komentarz, 40
 konwersje typów, 65
 krotki, 59
 liczby, 36
 liczby losowe, 39

listy, 49
 logika, 42
 łańcuchy, 47
 moduły, 67
 operator porównania, 42
 parametry, 39, 48
 pętle for, 38
 pliki, 75
 powłoka języka, 34
 serializacja, 79
 słowniki, 58
 wartości logiczne, 42
 wersje języka, 33
 wyjątki, 61
 zasoby, 141
 zmienne, 37
 znaki zachęty, 34

Python 2, 99
 Python 3, 99
 Python Games, 100

Q

QtonPi, 19
 QUIT, 102

R

r, 78
 r+, 78
 randint, 67
 random, 39, 67
 range, 39
 Raspberry Pi
 aplikacje, 30
 dodatkowe pakiety, 31
 dystrybucje systemu, 19
 internet, 27
 usuwanie pakietów, 32

Raspbian Wheezy, 11, 19, 25, 30
 dostępne pakiety, 31
 języki programowania, 143
 kompilator GNU C, 143
 korzystanie z magistrali I2C, 125
 wersje języka Python, 99

Raspbmc, 144

RaspiRobot, 129
 autostart, 134
 dodanie czujnika odległości, 136
 kod źródłowy pierwszego etapu projektu, 135

- lista elementów dla projektu, 131
 - łączenie czujnika odległości i adaptera, 137
 - montaż sprzętu, 130
 - instalacja oprogramowania w środowisku Raspberry Pi, 132
 - karta RaspiRobotBoard, 132
 - montaż podwozia, 131
 - podłączanie silniczków, 134
 - test działania, 135
 - zamiana pudełka na baterie, 132
 - odległość od przeszkody, 140
 - odłączanie zasilania, 137
 - okablowanie wyświetlacza, 137
 - oprogramowanie, 135
 - aktualizacja, 138, 139
 - podłączenie wyświetlacza, 136
 - podstawowa wersja pojazdu, 130
 - potrzebne elementy, 130
 - sprawdzanie naciśnięcia klawiszy, 136
 - szeregowy moduł czujnika odległości, 136
 - uruchomienie, 138
 - wygląd ekranu, 139
 - RaspiRobotBoard, 113, 140
 - montaż karty w RaspiRobot, 132
 - pliki źródłowe biblioteki, 133
 - zasilanie bateriami, 130
 - raw_input, 55
 - read, 78, 80
 - readline, 78
 - refaktoryzacja, 106
 - reguły nazewnictwa, 37
 - remove, 32
 - return, 52
 - rmtree, 79
 - round(x, n), 62
 - rowconfigure, 91, 92
 - RPi.GPIO, 69, 74, 128
 - RaspiRobot, 133
 - Run Module, 36
- ## S
- s
 - .capitalize(), 63
 - .center(width), 63
 - .endswith(str), 63
 - .find(str), 63
 - .format(args), 63
 - .isalnum(), 63
 - .isalpha(), 63
 - .isspace(), 63
 - .ljust(width), 63
 - .lower(), 63
 - .replace(old, new), 63
 - .split(), 63
 - .splitlines(), 63
 - .strip(), 63
 - .upper(), 63
 - SCL, 110
 - SCLK, 110
 - Scratch, 111, 143
 - edycja programu, 144
 - SDA, 110
 - selectmode, 89
 - self, 71
 - serializacja, 79
 - set, 93
 - showerror, 93
 - showinfo, 93
 - showwarning, 93
 - shutil, 78
 - .copy, 79
 - .move, 79
 - sieć Wi-Fi, 21
 - SINGLE, 89
 - Slice of PI/O, 112
 - cechy, 112
 - słowa kluczowe
 - def, 52
 - słowniki, 58
 - funkcje, 64
 - sort, 50
 - SPI, 110
 - spinbox, 89
 - stabilizator, 113
 - stałe łańcuchowe, 62
 - stałe numeryczne, 91
 - standardowa biblioteka, 68
 - sticky, 91
 - str, 54
 - string, 68
 - string index out of range, 48
 - StringVar, 86, 89
 - struktura interfejsu użytkownika, 85
 - stubs, 54
 - sudo, 30
 - super-user do, 30
 - system on a chip, 16

system plików, 78
 system Raspberry Pi
 karty prototypowania, 114
 karty rozszerzeń, 111
 podłączanie urządzeń, 109
 pulpit, 25
 RaspiRobot, 129
 sterowanie robotem, 113
 terminal, 29
 uruchamianie, 22
 wiersz poleceń, 27
 szeregowy moduł czujnika odległości, 136

Ś

środowisko graficzne, 14
 pulpit, 26

T

teoria łańcuchów, 47
 terminal, 29
 text.yview, 93
 textvariable, 86
 The MagPi, 142
 Tk, 83
 tkinter, 69, 83, 86
 .colorchooser, 94
 .filedialog, 94, 95
 .messagebox, 93
 podręcznik użytkownika modułu, 97
 total, 41
 True, 42
 try, 76
 tryb
 a, 78
 r, 78
 r+, 78
 w, 78
 typy
 DoubleVar, 86
 IntVar, 88
 StringVar, 86, 89

U

układy elementów, 89
 waga elementów siatki, 92
 update, 106

update_display, 139
 urllib.request, 69
 uruchamianie
 platformy, 16
 systemu, 22
 automatyczne uruchamianie środowiska
 graficznego, 23
 usuwanie pakietów, 32
 użycie modułów, 67

W

w, 78
 waga elementów siatki, 92
 wartości logiczne, 42
 web scraping, 81
 wewnętrzny rezystor pull-up, 128
 while, 44, 103
 widzety GUI, 88
 lista elementów, 88
 modyfikowalne, 90
 niezmienne, 90
 pasek przesuwania, 92
 pole wyboru, 88
 spinbox, 89
 układy elementów, 89
 widżet tekstowy, 92
 wiele wartości powrotnych, 60
 wielokrotne przypisywanie, 60
 wiersz poleceń, 27
 menu aplikacji, 28
 poruszanie się po systemie, 29
 Wi-Fi, 15
 write, 78
 write_display, 126
 współpraca ze sprzętem, 109
 wyjątki, 61
 wykrywanie złapania, 104
 wyrażenia regularne, 81
 wzmacniacz prądu w układzie Darlingтона, 111

X

XBMC, 144

Y

yscrollcommand, 93

Z

- załączki kodu, 54
- zapisywanie plików, 78
- zarządzanie złożonością, 51
- zasada DRY, 44
- zasilacz, 14, 17
- zasoby
 - internetowe, 32
 - związane z językiem Python, 141
 - związane z platformą Raspberry Pi, 142
 - związane z systemem Linux, 141
- zawartość katalogu, 79
- zdarzenia, 103
 - QUIT, 103, 136
- zegar LED, 121
 - dodawanie przycisku do układu, 126
 - lista elementów dla projektu, 122
 - lista połączeń do wykonania, 124
 - modyfikowanie układu elektronicznego, 126
 - montaż sprzętu, 122
 - oprogramowanie, 124
 - port wejściowy, 128
 - program obsługi zegara, 125
 - ulepszenie, 126
 - umieszczenie elementów na płycie
 - uniwersalnej, 123
- zero logiczne, 110
- zestaw Raspberry Pi, 17
 - karta SD, 18
 - klawiatura, 18
 - koncentrator USB, 21
 - łączenie elementów, 22
 - monitor, 18
 - mysz, 18
 - obudowa, 19
 - sieć Wi-Fi, 21
 - zasilacz, 17
- złącze GPIO, 109
 - bezpośrednie podłączenie do pinów, 110
 - piny, 109
- zmienna
 - bumpy-case, 38
 - camel-case, 38
 - ignore, 103
 - root, 84
 - total, 41
- zmienne, 37
 - globalne, 53
 - komunikat błędu, 38
 - konwencje nazewnictwa, 37
 - listy, 49
 - nazwy plików, 77
 - reguły nazewnictwa, 37
 - typu łańcuchowego, 55
 - zawierające łańcuch, 47
 - znak równości, 37
- znak ucieczki, 62
- znaki
 - #, 40
 - \$, 29
 - dwukropek, 38
 - gwiazdka, 68
 - końca, 78
 - kropka, 50
 - podwójny znak równości, 41, 42
 - pojedynczy znak równości, 41
 - pusty łańcuch, 78
 - równości, 47
 - szablonowe, 79

Ż

żądania HTTP, 80

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**

Odkryj możliwości małego Pi!

Zastanawiasz się, czy komputer wielkości lekko przerośniętej karty kredytowej podola typowym zadaniom? Oczywiście! Oprócz niewielkich rozmiarów ma on jeszcze jeden atut — kosztuje naprawdę niewiele. Raspberry Pi, bo o nim mowa, został zaprojektowany jako platforma do nauki programowania dla dzieci. Jego możliwości jednak rozrosły się błyskawicznie. W tej chwili jest wykorzystywany w wielu interesujących i nowatorskich projektach, sprawdza się również jako miniaturowy serwer WWW, sterownik lub platforma do odtwarzania filmów. Jeżeli masz pomysł, jak go wykorzystać — ta książka pomoże Ci osiągnąć cel.

To przewodnik, dzięki któremu nauczysz się tworzyć oprogramowanie dla Pi przy użyciu języka Python i edytora IDLE. W trakcie lektury dowiesz się, jak skonfigurować Pi, jakie akcesoria będą Ci potrzebne oraz jak to wszystko połączyć w całość. W kolejnych rozdziałach zaznajomisz się z podstawami obsługi systemu operacyjnego Linux oraz przekonasz się, że programowanie w języku Python wcale nie jest takie trudne. Ponadto zdobędziesz wiedzę na temat podłączania sprzętu do Pi za pomocą GPIO, a następnie zaczniesz budować robota — RaspiRobota. To prawdziwie fascynująca książka a zrealizowany w niej inspirujący projekt da Ci zajęcie na długie wieczory. Przekonaj się sam!

Dzięki tej książce:

- poznasz platformę Raspberry Pi
- opanujesz podstawy języka Python
- wykonasz prosty projekt oparty na Pi
- zbudujesz swojego robota — RaspiRobota

helion.pl
księgarnia
internetowa

Nr katalogowy: 17957



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



Zamówienia telefoniczne:
0 801 339900



0 601 339900



Helion

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
<http://helion.pl>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-246-8709-1



9 788324 687091

cena: 34,90 zł

Informatyka w najlepszym wydaniu