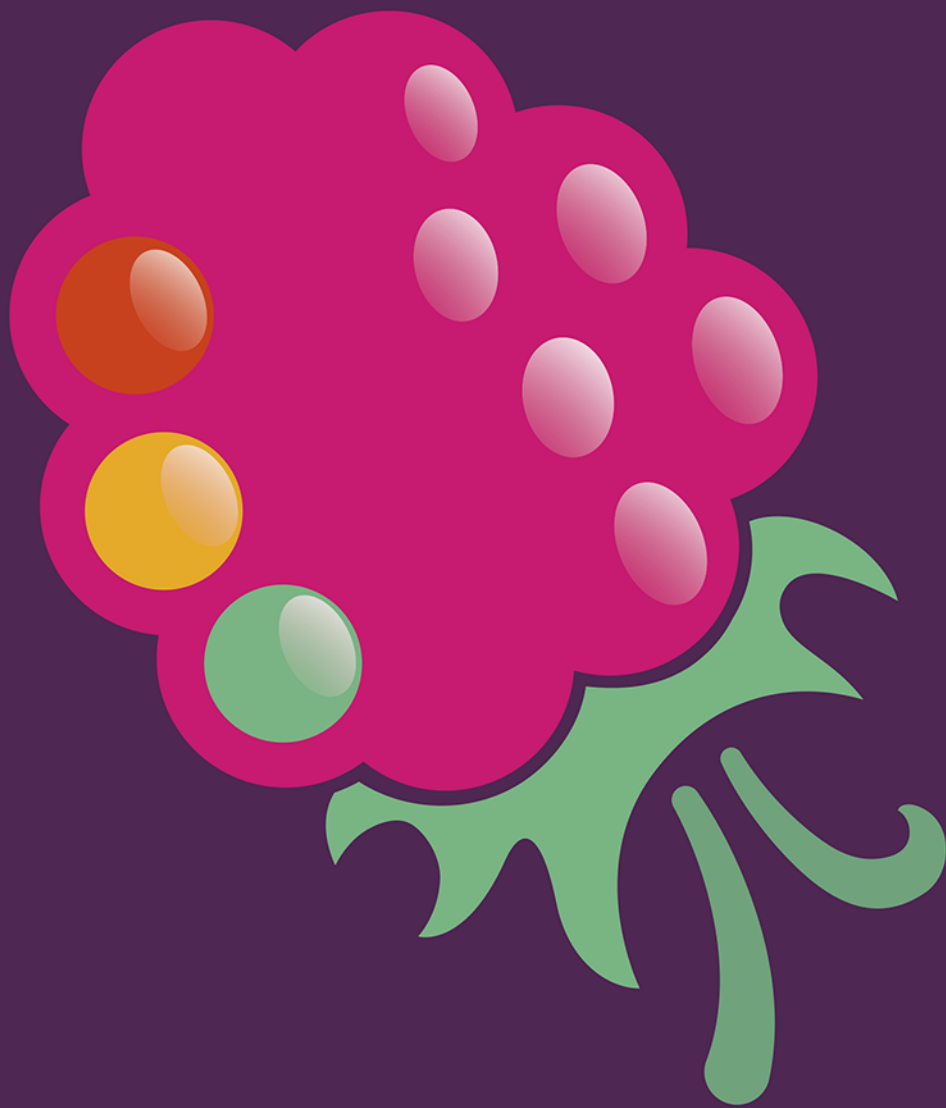


Witold Wrotek



Raspberry Pi

od podstaw

Helion 

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

Redaktor prowadzący: Małgorzata Kulik

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

Grafika na okładce została wykorzystana za zgodą AdobeStock.com.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
WWW: helion.pl (księgarnia internetowa, katalog książek)

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

ISBN: 978-83-289-2311-9

Copyright © Helion S.A. 2025

Printed in Poland.

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

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

SPIST TREŚCI

1	PODSTAWY RASPBERRY PI 5	7
	Opis modelu Raspberry Pi 5	7
	Specyfikacja techniczna	9
	Jak przygotować Raspberry Pi 5 do pracy?	12
	Instalacja systemu operacyjnego (Raspberry Pi OS)	14
2	PIERWSZE KROKI Z RASPBERRY PI	34
	Jak się zalogować do skonfigurowanego systemu Raspberry Pi 5?	34
	Jak podłączać elementy i układy elektroniczne do GPIO?	36
	1. Konstrukcja wewnętrzna płytki stykowej	37
	2. Jak podłączać elementy do płytki stykowej?	38
	3. Przykładowy obwód na płycie stykowej (LED i rezystor)	40
	Funkcje pinów GPIO w Raspberry Pi 5	43
	Identyfikacja pinów	49
	Fizyczny układ pinów	49
	Numeracja GPIO (BCM — Broadcom SOC Channel)	50
	Wymagania, które powinny spełniać sygnały podawane i odbierane z GPIO	51
	1. Poziomy napięcie GPIO w Raspberry Pi	52
	2. Akceptowane natężenia prądu na pinach GPIO	52
	3. Ochrona przed zbyt wysokim napięciem	52
	4. Sterowanie prądami wyjściowymi GPIO	53
	5. Bezpieczeństwo i ochrona układów	53
	6. Sygnały komunikacyjne	57
	7. Typowe elementy podłączane do GPIO	57
	Podsumowanie	57
3	PROGRAMOWANIE RASPBERRY PI	58
	Dlaczego Python?	58
	Instalacja i uruchomienie środowiska programistycznego	59
	Pierwszy skrypt w Pythonie	61
	Uruchomienie skryptu	62

CZĘŚĆ I. PROSTSZE PROJEKTY DLA POCZĄTKUJĄCYCH 63

4	PROJEKT 1. STEROWANIE DIODĄ LED	65
5	PROJEKT 2. MIGAJĄCA DIODA	73
6	PROJEKT 3. SYGNALIZATOR ŚWIETLNY	79
7	PROJEKT 4. ELEKTRONICZNY LICZNIK	84
8	PROJEKT 5. TERMOMETR I CZUJNIK WILGOTNOŚCI NA RASPBERRY PI 5	96
9	PROJEKT 6. REJESTRATOR TEMPERATURY I WILGOTNOŚCI	106
	Funkcja readFirstLine	108
	Główna część programu	108
	Odczyt temperatury	108
	Przechwytywanie przerwania programu	109
10	PROJEKT 7. PROSTY ALARM Z CZUJNIKIEM RUCHU	110
11	PROJEKT 8. STACJA POGODOWA	121

CZĘŚĆ II. DALSZE PROJEKTY DLA BARDZIEJ ZAAWANSOWANYCH 133

12	PROJEKT 9. WYŚWIETLACZE I INTERFEJSY	135
13	PROJEKT 10. ZEGAR CYFROWY	146
14	PROJEKT 11. WYŚWIETLANIE DANYCH Z CZUJNIKA	155
15	PROJEKT 12. MAGISTRALA 1-WIRE	164
16	PROJEKT 13. MONITORING I POWIADOMIENIA	178
	Sandbox WhatsApp w Twilio	184
	Częste błędy i sposoby ich rozwiązania	185
	FileNotFoundError: [Errno 2] No such file or directory	185
	Błąd 63007	188
17	PROJEKT 14. ZDALNE STEROWANIE DIODĄ LED PRZESIEĆ	192
18	PROJEKT 15. KOMUNIKACJA BEZPRZEWODOWA	200
	Jak sparować Raspberry Pi 5 z telefonem komórkowym?	202
19	PROJEKT 16. MEDIA CENTER NA RASPBERRY PI	214
	Połączenia sieciowe	219
	Konfiguracja zasobów sieciowych w Kodi	220
	Menedżer dodatków	222

Kroki podstawowe	222
Instalacja dodatków z oficjalnego repozytorium Kodi	222
Spotify	225
Serwisy VOD (Netflix, Amazon Prime Video, Max)	225
Dostosowywanie ustawień audio i wideo	226
Kroki do dostosowywania ustawień audio i wideo	226
Aplikacja na smartfona do sterowania Kodi	227
Jak korzystać z Kore (Android)?	228
Jak korzystać z Official Kodi Remote (iOS)?	228
20 PROJEKT 17. STEROWANIE SILNIKIEM KROKOWYM	229
21 PROJEKT 18. STEROWANIE PWM	243
A RASPBERRY PI 5 I RASPBERRY PI OS. ROZWIĄZYWANIE PROBLEMÓW X	251
B I2C I 1-WIRE. PORÓWNANIE	269
C BEZPIECZEŃSTWO I NIEZAWODNOŚĆ UKŁADÓW ELEKTRONICZNYCH OKIEM PRAKTYKA	273
Jaka jest różnica między przewodem uziemiającym a przewodem neutralnym i dlaczego uziemienie jest tak ważne?	273
Dlaczego układy elektroniczne mogą działać w różnych warunkach klimatycznych?	275
Jak pogodzić wodę i prąd?	277

6

PROJEKT 3. SYGNALIZATOR ŚWIETLNY

Dodając do układu diodę w kolorze zielonym i rozbudowując program, można zbudować symulator sygnalizatora świetlnego.

Zanim podłączymy diodę do Raspberry Pi 5, trzeba dobrać rezystor, aby nie zostały przekroczone:

- maksymalne dopuszczalne natężenie prądu do jednego pinu,
- sumaryczne maksymalne dopuszczalne natężenie prądu czerpanego ze wszystkich pinów.

Aby przez diodę LED w kolorze zielonym płynął prąd o takim samym natężeniu jak przez diodę LED w kolorze czerwonym (wówczas jasności świecenia obu diod będą zbliżone), musi na niej powstać spadek napięcia wynoszący 2,5 V (rysunek 4.2). Pozostałe napięcie, czyli 0,8 V, powinno odłożyć się na rezystorze R. Z charakterystyki (rysunek 4.1) wynika, że przy napięciu 2,5 V przez diodę LED w kolorze zielonym płynie prąd o natężeniu około 8 mA. Z prawa Ohma można wyliczyć wartość R:

$$R = \frac{U_r}{I}$$

Po podstawieniu wartości liczbowych jest to:

$$R = \frac{0,8 \text{ V}}{8 \text{ mA}} = 0,1 \text{ k}\Omega = 100 \text{ }\Omega$$

Najpopularniejsze rezystory należą do szeregu E20. Zatem należy użyć rezystora o wartości 100 Ω .

Obie diody będą w sumie pobierały z układu prąd o natężeniu 16 mA. Jest ono niższe od maksymalnej wartości.

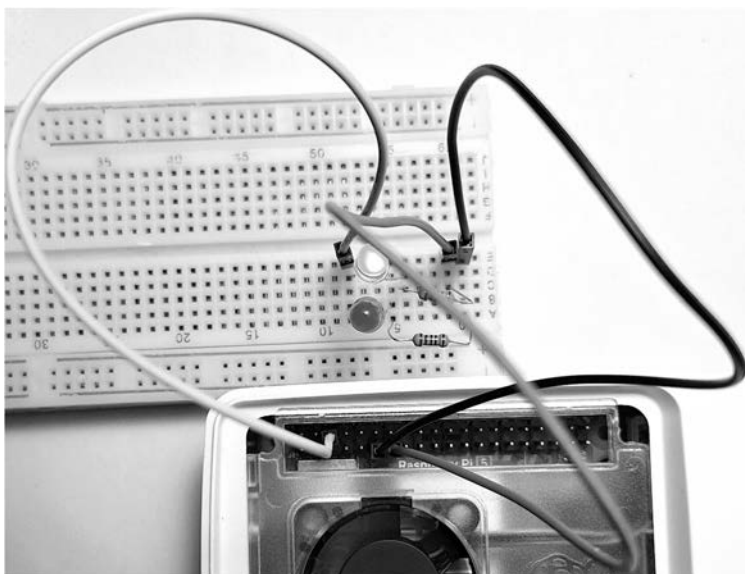
Do podłączenia diody LED wykorzystam pin, który na złączu GPIO ma numer 13, ale przypisana mu jest nazwa GPIO27.



W przykładzie podłączę zieloną diodę LED za pośrednictwem rezystora o takiej samej rezystancji, jakiej użyłem do podłączania czerwonej diody LED. Dlaczego? Ma on wyższą wartość niż wyliczona dla diody zielonej, a zatem nie spowoduje przeciążenia GPIO. Jeżeli dioda zielona będzie świeciła z podobną intensywnością jak dioda czerwona, ograniczę zużycie prądu, co ma znaczenie, jeżeli będę chciał korzystać z zasilania bateryjnego. Uzyskam unifikację elementów: zamiast dwóch nominalów rezystorów będę używał jednego.



Elektronik jak saper — myli się tylko raz. Potem musi zmontować zniszczony układ od początku. Połącz elementy (rysunek 6.1) i zanim włączysz zasilanie, sprawdź dwa, a najlepiej trzy razy, czy elementy są połączone poprawnie.



Rysunek 6.1. Do GPIO podłączone zostały dwie diody LED

W układzie połączyłem pin 6 (GND) z rzędem 8. płytki stykowej. Do rzędu 8. podłączone są katody obu diod LED.



Katoda diody oznaczona jest na schemacie kreską. Po podłączeniu do niej niższego napięcia niż do anody (oznaczonej strzałką) dioda spolaryzowana jest w kierunku przewodzenia.

Dioda LED w kolorze czerwonym ma anodę podłączoną do rzędu 5. płytki stykowej. Rząd ten zasilany jest przez rezystor o wartości 220 Ω , podłączony do rzędu 0. Do rzędu 0 podłączony jest również przewód biegnący do pinu 11 „malinki”.

Dioda LED w kolorze zielonym ma anodę podłączoną do rzędu 4. płytki stykowej. Rząd ten zasilany jest przez rezystor o wartości 220 Ω , podłączony do rzędu 1. Do rzędu 1. podłączony jest również przewód biegnący do pinu 13 Raspberry Pi 5.

Na rysunku 6.2 pokazałem program, który steruje naprzemiennym zapalaniem diod LED.

```

1 import gpiozero
2 import time
3 delay = 2
4 GPIO_RED = 17
5 GPIO_GREEN = 27
6 chip = gpiozero.Chip('gpiochip4')
7 led_red = chip.get_line(GPIO_RED)
8 led_green = chip.get_line(GPIO_GREEN)
9 led_red.request(consumer="LED_RED", type=gpiozero.LINE_REQ_DIR_OUT)
10 led_green.request(consumer="LED_GREEN", type=gpiozero.LINE_REQ_DIR_OUT)
11 try:
12     while True:
13         led_red.set_value(1)
14         led_green.set_value(0)
15         time.sleep(delay)
16         led_red.set_value(0)
17         led_green.set_value(1)
18         time.sleep(delay)
19 finally:
20
21     led_red.release()
22     led_green.release()

```

Rysunek 6.2. Program naprzemiennie włączający diody LED podłączone do GPIO 17 i GPIO 27

Jak działa ten program?

Linie rozpoczynające się od słów `import` powodują wczytanie bibliotek: `gpio` (odpowiedzialnej za obsługę GPIO w Raspberry Pi 5) oraz `time` (służy ona do obsługi operacji związanych z czasem, takich jak zatrzymywanie działania programu na określony czas, pomiar czasu wykonania kodu, uzyskiwanie aktualnej daty i godziny).

Następnie zmiennej `delay` przypisywana jest wartość 2. Jest ona wykorzystywana do ustalania czasu cyklicznego włączenia (wyłączenia) diod LED.

W kolejnych dwóch liniach zmiennym `LED_RED` i `LED_GREEN` przypisywane są wartości 17 oraz 27. Są to numery GPIO „widziane” z wnętrza układu scalonego. Nie są one identyczne z kolejnymi numerami kołków w GPIO (tabela 2.1).

W liniach zaczynających się do słów `led_red.request` oba piny są ustawione jako wyjściowe, co umożliwia ich kontrolę.

Następnie rozpoczyna się naprzemienne wykonywanie nieskończonej pętli `while True`. Zawarte w niej polecenia:

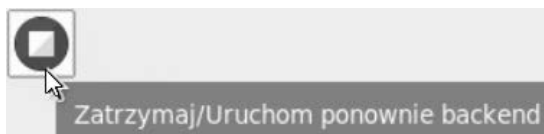
- zapalają diodę podłączoną do pinu 11 (`led_red.set_value(1)`) i gaszą diodę zasilaną z pinu 13 (`led_green.set_value(0)`).
- następnie odwracają stany — gaszą diodę zasilaną z pinu 11 (`led_red.set_value(0)`) i zapalają diodę podłączoną do pinu 13 (`led_green.set_value(1)`).



Pamiętaj, że w układzie zmienione zostaną piny fizyczne i należy także odpowiednio zmienić numery GPIO (`GPIO_RED` i `GPIO_GREEN`).



Program możesz zatrzymać kliknięciem przycisku *Zatrzymaj/Uruchom ponownie backend* (rysunek 6.3).



Rysunek 6.3. Przycisk do zatrzymywania działania programu bez usuwania go z pamięci „malinki”

W rozdziale 4. napisałem, że pierwszym celem przy nauce języka programowania jest napisanie programu wyświetlającego tekst „Hello, World”, a jego odpowiednikiem przy programowania komputera jednopłytkowego jest włączenie diody LED. Obecnie mamy już za sobą trzy projekty związane z włączaniem diody LED. W kolejnym rozdziale zajmiemy się odczytywaniem stanu przycisku oraz zliczaniem jego naciśnień.

PROGRAM PARTNERSKI

— GRUPY 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
Helion

Raspberry Pi od podstaw

Twój przewodnik po świecie
jednopłytkowych komputerów!



Czy wiesz, jak:

- dobrać odpowiednie obciążenie do Raspberry Pi, by działało stabilnie i wydajnie?
- rozpocząć programowanie i sterować czujnikami, silnikami i wyświetlaczami?
- stworzyć inteligentny alarm, stację pogodową, rejestrator temperatury i wilgotności?
- wyświetlać dane z czujników na ekranach i komunikować Raspberry Pi z innymi urządzeniami?
- sterować diodą LED przez internet lub użyć magistrali 1-Wire do zaawansowanej komunikacji?
- zmienić Raspberry Pi w centrum multimedialne lub platformę do monitoringu i powiadomień?

Ta książka krok po kroku wprowadzi Cię w podstawy Raspberry Pi, programowanie i praktyczne projekty — od prostych układów dla początkujących po bardziej zaawansowane systemy. Znajdziesz tu także porady dotyczące rozwiązywania problemów i poprawy niezawodności swoich projektów.

**Zacznij przygodę z Raspberry Pi już dziś
i twórz własne projekty elektroniczne!**

Helion



helion.pl



HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI

Sięgnij po więcej! ▶



ISBN 978-83-289-2311-9



Cena: 69,00 zł