

**Jak zaprogramować robota : zastosowanie Raspberry Pi i Pythona w tworzeniu autonomicznych robotów / Danny Staple. – Gliwice , © 2022**

Spis treści

<b>O autorze</b>	<b>13</b>
<b>O korektorach</b>	<b>14</b>
<b>Wstęp</b>	<b>15</b>
<b>Część I. Podstawy robotyki</b>	<b>21</b>
<b>Rozdział 1. Wprowadzenie do robotyki</b>	<b>23</b>
Co oznacza słowo „robot“?	24
Przykłady zaawansowanych i imponujących robotów	25
Łaziki marsjańskie	27
Roboty w domu	27
Pralka	29
Inne roboty w domu	30
Roboty w przemyśle	31
Robotyczne ramiona	31
Roboty magazynowe	32
Roboty edukacyjne, hobbystyczne i biorące udział w zawodach	33
Podsumowanie	36
Ćwiczenia	37
Lektura uzupełniająca	37
<b>Rozdział 2. Odkrywanie elementów robota — kod i elektronika</b>	<b>38</b>
Wymagania techniczne	38
Z czego zbudowany jest robot?	39
Rodzaje części robota	42
Rodzaje silników	42
Inne elementy wykonawcze	44
Wskaźniki stanu — wyświetlacze, światła i dźwięki	45
Rodzaje czujników	45
Kontrolery oraz wejścia i wyjścia	47
Piny wejścia/wyjścia	48
Kontrolery	49
Wybór Raspberry Pi	51
Projekt z uwzględnieniem części i struktury kodu	53
Projekt fizycznej budowy robota	55
Podsumowanie	57
Ćwiczenia	58
Lektura uzupełniająca	58

<b>Rozdział 3. Odkrywanie Raspberry Pi</b>	<b>59</b>
Wymagania techniczne	59
Odkrywanie możliwości Raspberry Pi	60
Prędkość i moc	60
Złącza i praca w sieci	60
Wybór Raspberry Pi 3A+	61
Wybór złączy	61
Nakładki Hat do Raspberry Pi	63
Czym jest Raspberry Pi OS?	64
Przygotowanie karty SD za pomocą Raspberry Pi Imager	65
Podsumowanie	67
Ćwiczenia	67
Lektura uzupełniająca	68
<b>Rozdział 4. Przygotowanie Raspberry Pi pod robota</b>	<b>69</b>
Wymagania techniczne	69
Czym jest system „bez głowy” i dlaczego jest praktycznym rozwiązaniem dla robota?	70
Konfiguracja Wi-Fi na Raspberry Pi i włączenie dostępu do SSH	72
Znalezienie swojego Raspberry Pi w sieci	73
Instalacja programu Bonjour w systemie Windows	74
Test programu Bonjour	74
Wykrywanie i rozwiązywanie problemów	75
Łączenie się z Raspberry Pi za pomocą PuTTY lub SSH	76
Konfiguracja Raspberry Pi OS	77
Zmiana nazwy Raspberry Pi	78
Zabezpieczenie Raspberry Pi (choć w małym stopniu)	78
Ponowne uruchomienie Raspberry Pi i połączenie się z nim	80
Aktualizacja oprogramowania Raspberry Pi	82
Wyłączanie Raspberry Pi	83
Podsumowanie	84
Ćwiczenia	84
Lektura uzupełniająca	84
<b>Rozdział 5. Tworzenie kopii zapasowej kodu za pomocą Gita i karty pamięci SD</b>	<b>85</b>
Wymagania techniczne	86
Jak kod może zostać uszkodzony lub utracony?	86
Utrata lub uszkodzenie danych na karcie SD	86
Zmiany w kodzie i ustawieniach	87
Strategia 1. Zapisywanie kodu na PC i przesyłanie go do Pi	87
Strategia 2. Użycie Gita do cofania się w czasie	90
Strategia 3. Tworzenie kopii zapasowych na karcie SD	92
Windows	93
Mac	95
Linux	98

Podsumowanie	99
Ćwiczenia	99
Lektura uzupełniająca	100
<b>Część II. Budowanie autonomicznego robota – podłączanie czujników i silników do Raspberry Pi</b>	<b>101</b>
<b>Rozdział 6. Podstawy budowania robota – koła, zasilanie i połączenia</b>	<b>103</b>
Wymagania techniczne	104
Wybór podwozia robota	104
Rozmiar	104
Liczba kół	105
Koła i silniki	106
Prostota	108
Cena	108
Wnioski	108
Wybór sterownika silników	109
Stopień integracji	110
Wykorzystanie pinów	110
Rozmiar	111
Lutowanie	112
Zasilanie	112
Złącza	112
Wnioski	113
Zasilanie robota	114
Testowe dopasowanie elementów robota	116
Składanie podstawy robota	119
Montaż płytek enkodera	122
Montaż wsporników	122
Montaż kółka samonastawnego	125
Zakładanie kół	126
Przygotowanie przewodów	126
Montaż Raspberry Pi	127
Dodanie baterii	128
Gotowa podstawa robota	130
Podłączanie silników do Raspberry Pi	131
Podłączanie sterownika do silników i baterii	132
Niezależne zasilanie	135
Podsumowanie	137
Ćwiczenia	137
Lektura uzupełniająca	138
<b>Rozdział 7. Jazda do przodu i skręcanie –wprawianie silników w ruch za pomocą Pythona</b>	<b>139</b>
Wymagania techniczne	140
Testowy kod dla silników	140

Przygotowanie bibliotek	140
Test — wyszukanie nakładki sterownika silników	141
Test — pokazanie, że silniki się kręcą	143
Rozwiązywanie problemów	143
Omówienie kodu	144
Sterowanie robotem	146
Rodzaje sterowania	146
Kierowanie budowanym przez nas robotem	149
Obiekt Robot — kod do eksperymentów związanych z komunikacją z robotem	150
Dlaczego warto utworzyć ten obiekt?	151
Z czego się składa obiekt Robot?	152
Skrypt pokonania z góry określonej ścieżki	156
Podsumowanie	159
Ćwiczenia	159
Lektura uzupełniająca	159

## **Rozdział 8. Programowanie czujników odległości za pomocą Pythona** **160**

Wymagania techniczne	160
Wybór między czujnikami optycznymi a ultradźwiękowymi	161
Optyczne czujniki odległości	161
Ultradźwiękowe czujniki odległości	163
Stany logiczne i przesuwanie poziomów napięcia	163
Dlaczego dwa czujniki?	166
Podłączanie czujnika ultradźwiękowego i odczytywanie z niego danych	167
Montaż czujników	167
Dodawanie przełącznika zasilania	169
Podłączanie czujników odległości	171
Instalacja bibliotek Pythona do komunikacji z czujnikiem	173
Odczytywanie odległości z czujnika ultradźwiękowego	174
Wykrywanie i rozwiązywanie problemów	176
Unikanie ścian — skrypt omijania przeszkód	177
Dodawanie czujników do klasy Robot	177
Zachowania polegające na omijaniu przeszkód	178
Podsumowanie	184
Ćwiczenia	184
Lektura uzupełniająca	185

## **Rozdział 9. Programowanie pasków LED RGB za pomocą Pythona** **186**

Wymagania techniczne	187
Czym jest pasek LED RGB?	187
Porównanie technologii stosowanych w paskach świetlnych	187
Wartości RGB	189
Podłączanie pasków LED RGB do Raspberry Pi	190
Podłączanie paska LED	191

Pisanie kodu dla diod LED	191
Tworzenie interfejsu LED	192
Dodawanie klasy Leds do klasy Robot	193
Test jednej diody LED	195
Test wszystkich diod LED	196
Wyświetlanie tęczy za pomocą diod LED	197
Modele przestrzeni barw	197
Wyświetlanie tęczy na pasku LED	200
Wykorzystanie paska LED RGB do rozwiązywania problemów z unikaniem przeszkód	202
Dodawanie diod LED do zachowania unikania przeszkód	202
Dodawanie kolorów tęczy	204
Podsumowanie	205
Ćwiczenia	206
Lektura uzupełniająca	206
<b>Rozdział 10. Sterowanie serwomotorami za pomocą Pythona</b>	<b>207</b>
Wymagania techniczne	207
Czym są serwomotory?	208
Budowa serwomotoru	209
Wysyłanie pozycji do serwomotorów	209
Ustawianie serwomotoru za pomocą Raspberry Pi	211
Kod obracający serwomotorem	213
Wykrywanie i rozwiązywanie problemów	215
Sterowanie silnikami prądu stałego i serwomotorami	216
Kalibracja serwomotorów	217
Dodawanie mechanizmu uchylnno-obrotowego	218
Budowa mechanizmu uchylnno-obrotowego	218
Montaż mechanizmu uchylnno-obrotowego na robocie	223
Kod dla mechanizmu uchylnno-obrotowego	224
Obiekt serwomotoru	224
Dodawanie serwomotoru do klasy robota	226
Kręcenie mechanizmem uchylnno-obrotowym	227
Uruchamianie mechanizmu uchylnno-obrotowego	229
Wykrywanie i rozwiązywanie problemów	229
Budowanie sonaru	230
Montaż czujnika	231
Instalacja bibliotek	234
Kod zachowania	234
Podsumowanie	237
Ćwiczenia	237
Lektura uzupełniająca	238
<b>Rozdział 11. Programowanie enkoderów za pomocą Pythona</b>	<b>239</b>
Wymagania techniczne	240
Pomiar przejechanego dystansu za pomocą enkoderów	240
Zastosowanie enkoderów	240

Rodzaje enkoderów	241
Określanie położenia bezwzględnego i względnego	242
Określanie kierunku i prędkości	243
Enkodery w naszym robocie	244
Montaż enkoderów	245
Przygotowanie enkoderów	246
Podnoszenie Raspberry Pi	246
Przymocowanie enkoderów do podwozia	247
Podłączanie enkoderów do Raspberry Pi	248
Wykrywanie pokonanej odległości za pomocą Pythona	249
Zapisywanie informacji w logu	250
Proste zliczanie	250
Dodawanie enkoderów do obiektu robota	253
Przeliczanie tyknień na milimetry	255
Jazda po linii prostej	257
Korygowanie toru jazdy za pomocą regulatora PID	257
Obiekt regulatora PID w Pythonie	259
Kod jazdy po linii prostej	260
Wykrywanie i rozwiązywanie problemów dotyczących tego zachowania	263
Pokonanie zadanego dystansu	263
Refaktoryzacja zamiany jednostek w klasie EncoderCounter	263
Inicjalizacja stałych	264
Zachowanie polegające na pokonywaniu zadanej odległości	265
Skręcanie w określony sposób	267
Funkcja jazdy po łuku	271
Podsumowanie	272
Ćwiczenia	273
Lektura Uzupełniająca	273
<b>Rozdział 12. Programowanie IMU za pomocą Pythona</b>	<b>274</b>
Wymagania techniczne	275
Urządzenia nawigacji inercyjnej	275
Polecane modele IMU	276
Lutowanie — dodawanie złączy do IMU	277
Połączenia lutowane	277
Montaż IMU na robocie	279
Umieszczenie czujnika na robocie	279
Podłączanie IMU do Raspberry Pi	283
Pomiar temperatury	284
Instalacja oprogramowania	285
Wykrywanie i rozwiązywanie problemów	286
Odczytywanie pomiarów temperatury	286
Wykrywanie i rozwiązywanie problemów	290
Uproszczenie linii poleceń systemu VPython	291
Odczytywanie danych z żyroskopu za pomocą Pythona	291
Zasada działania żyroskopu	291
Dodawanie żyroskopu do interfejsu	294

Wykres danych z żyroskopu	295
Odczytywanie danych z akcelerometru za pomocą Pythona	297
Zasada działania akcelerometru	297
Dodawanie akcelerometru do interfejsu	298
Wyświetlanie danych z akcelerometru w postaci wektora	298
Praca z magnetometrem	300
Zasada działania magnetometru	300
Dodawanie magnetometru do interfejsu	302
Wyświetlanie danych z magnetometru	302
Podsumowanie	304
Ćwiczenia	304
Lektura uzupełniająca	304

### **Część III. Słyszenie i widzenie – wyposażenie robota w inteligentne czujniki** **305**

<b>Rozdział 13. System wizyjny robota z wykorzystaniem bibliotek PiCamera i OpenCV</b>	<b>307</b>
Wymagania techniczne	308
Konfiguracja kamery dla Raspberry Pi	308
Montaż kamery na mechanizmie uchylno-obrotowym	309
Podłączanie kamery	312
Konfiguracja oprogramowania do rozpoznawania obrazów	314
Konfiguracja oprogramowania kamery Pi	315
Zdjęcie z Raspberry Pi	315
Instalacja OpenCV i bibliotek pomocniczych	316
Tworzenie aplikacji dla Raspberry Pi do przesyłania obrazu	317
Projektowanie serwera kamery OpenCV	317
Obiekt CameraStream	318
Tworzenie głównej aplikacji serwera do przesyłania obrazów	320
Budowa szablonu	322
Uruchamianie serwera	322
Wykrywanie i rozwiązywanie problemów	323
Wykonywanie zadań w tle w trakcie przesyłania obrazu	323
Podążanie za kolorowymi obiektami za pomocą Pythona	329
Zamiana obrazu na informacje	331
Rozbudowa regulatora PID	333
Dodawanie pozostałych elementów zachowania	334
Uruchamianie zachowania	339
Wykrywanie i usuwanie problemów	340
Śledzenie twarzy za pomocą Pythona	341
Szukanie obiektów na obrazie	341
Projektowanie naszego zachowania	345
Kod odpowiedzialny za śledzenie twarzy	345
Uruchamianie zachowania polegającego na śledzeniu twarzy	349
Wykrywanie i rozwiązywanie problemów	349
Podsumowanie	350

Ćwiczenia	350
Lektura uzupełniająca	351

## **Rozdział 14. Śledzenie linii z wykorzystaniem kamery i Pythona** **353**

Wymagania techniczne	354
Śledzenie linii — wprowadzenie	354
Czym jest śledzenie linii?	354
Zastosowanie w przemyśle	355
Rodzaje technik śledzenia linii	356
Tworzenie trasy testowej dla funkcji śledzenia linii	357
Przygotowanie niezbędnych materiałów	357
Wytaczanie linii	358
Proces śledzenia linii z wykorzystaniem komputerowego rozpoznawania obrazów	359
Algorytmy śledzenia linii za pomocą kamery	359
Proces rozpoznawania linii	360
Testowanie widzenia komputerowego za pomocą przykładowych obrazów	362
Dlaczego należy używać obrazów testowych?	362
Przygotowanie obrazów testowych	363
Kod Pythona znajdujący krawędzie linii	364
Określanie położenia linii na podstawie krawędzi	367
Obrazy testowe z niewyraźną linią	369
Śledzenie linii z wykorzystaniem algorytmu PID	371
Tworzenie schematu zachowania	372
Dodawanie czynnika czasu do regulatora PID	373
Tworzenie wstępnej wersji zachowania	373
Regulacja wartości PID	380
Wykrywanie i rozwiązywanie problemów	380
Ponowne odnajdowanie linii	380
Podsumowanie	381
Ćwiczenia	382
Lektura uzupełniająca	382

## **Rozdział 15. Komunikacja głosowa z robotem za pomocą programu Mycroft** **383**

Wymagania techniczne	384
Wprowadzenie do programu Mycroft — terminologia asystenta głosowego	384
Zamiana mowy na tekst	385
Słowa wybudzające	385
Wypowiedzi	385
Intencja	385
Dialogi	386
Słownictwo	386



Umiejętności	386
Ograniczenia nasłuchiwania mowy przez robota	387
Dodawanie wejścia i wyjścia audio do Raspberry Pi	387
Montaż nakładki	388
Instalacja asystenta głosowego na Raspberry Pi	389
Instalacja oprogramowania nakładki ReSpeaker Pi	390
Komunikacja programu Mycroft z kartą dźwiękową	392
Pierwsze kroki w programie Mycroft	393
Wykrywanie i rozwiązywanie problemów	395
Programowanie API za pomocą modułu Flask	396
Zarys sterowania robotem za pomocą Mycroftu	396
Zdalne uruchamianie zachowania	397
Programowanie interfejsu sterującego	399
Wykrywanie i rozwiązywanie problemów	401
Programowanie asystenta głosowego w programie Mycroft	402
Tworzenie intencji	402
Wykrywanie i rozwiązywanie problemów	408
Dodawanie kolejnej intencji	408
Podsumowanie	410
Ćwiczenia	411
Lektura uzupełniająca	412
<b>Rozdział 16. Więcej o IMU</b>	<b>413</b>
Wymagania techniczne	413
Programowanie wirtualnego robota	414
Tworzenie modelu w VPythonie	414
Wykrywanie obrotu za pomocą żyroskopu	418
Kalibracja żyroskopu	419
Obracanie wirtualnym robotem za pomocą żyroskopu	421
Wykrywanie pochylenia i przechylenia za pomocą akcelerometru	424
Odczyt pochylenia i przechylenia z wektora akcelerometru	424
Wygładzanie odczytów akcelerometrów	427
Fuzja danych pomiarowych z akcelerometru i żyroskopu	428
Wykrywanie odchylenia za pomocą magnetometru	431
Kalibracja magnetometru	432
Odczytywanie przybliżonej wartości odchylenia robota z magnetometru	437
Zestawienie odczytów z czujników w celu ustalenia orientacji	439
Sterowanie robotem na podstawie danych z IMU	445
Podsumowanie	447
Ćwiczenia	447
Lektura uzupełniająca	447
<b>Rozdział 17. Sterowanie robotem za pomocą telefonu i Pythona</b>	<b>449</b>
Wymagania techniczne	450
Gdy nie działa sterowanie głosem — dlaczego musimy mieć możliwość sterowania	450
Menu — wybieranie zachowań dla robota	451

Zarządzanie trybami robota	452
Wykrywanie i rozwiązywanie problemów	453
Usługa sieciowa	454
Szablon	455
Uruchamianie aplikacji	457
Wykrywanie i rozwiązywanie problemów	459
Wybór kontrolera — jak będziemy sterować robotem i dlaczego	459
Projekt i ogólny zarys	460
Przygotowanie Raspberry Pi do zdalnego sterowania — przygotowanie podstawowego systemu sterowania	462
Rozbudowa podstawowej aplikacji do obsługi obrazów	464
Budowa systemu ręcznego sterowania	464
Szablon (strona internetowa)	467
Arkusz stylów	469
Programowanie suwaków	472
Uruchamianie ręcznego sterowania	475
Wykrywanie i rozwiązywanie problemów	476
Robot w pełni sterowany za pomocą telefonu	477
Tryby menu kompatybilne z zachowaniami opartymi na module Flask	477
Wgrywanie usług wideo	477
Nadanie menu stylu	480
Menu startowe dla Raspberry Pi	481
Dodawanie diody do serwera menu	482
Automatyczne uruchamianie robota za pomocą systemu	482
Podsumowanie	485
Ćwiczenia	485
Lektura uzupełniająca	486
<b>Część IV. Kontynuacja przygody z robotyką</b>	<b>489</b>
<b>Rozdział 18. Rozwijanie umiejętności z zakresu robotyki</b>	<b>491</b>
Spółeczności konstruktorów robotów w sieci — fora i media społecznościowe	492
Kanały w serwisie YouTube, które warto znać	493
Pytania natury technicznej — gdzie szukać pomocy?	494
Spotkania konstruktorów robotów — zawody, miejsca dla twórców, spotkania	494
Przestrzeń dla twórców	495
Targi twórców, Raspberry Jams i Doja	495
Zawody	496
Propozycje nowych umiejętności do zdobycia — druk 3D, lutowanie, PCB i CNC	497
Projektowanie	497
Umiejętności związane z formowaniem i budowaniem	498
Umiejętności związane z elektroniką	500
Wzbogacanie wiedzy o rozpoznawaniu obrazów	503
Książki	503

Kursy internetowe	503
Media społecznościowe	504
Wzbogacanie swojej wiedzy o uczenie maszynowe	504
Platforma programistyczna ROS	505
Podsumowanie	505
Lektura uzupełniająca	506
<b>Rozdział 19. Projekt kolejnego robota – podsumowanie</b>	<b>507</b>
Wymagania techniczne	508
Wizualizacja Twojego następnego robota	508
Tworzenie schematu blokowego	510
Wybór części	511
Planowanie kodu dla robota	512
Przedstawienie świata swojego projektu	515
Podsumowanie	516

oprac. BPK