Twórcy

O autorze

Danny Staple jest konstruktorem robotów i programistą. Od 2000 roku jest zawodowym inżynierem oprogramowania, zawodowo korzysta z Pythona oraz regularnie uczestniczy w projektach open source.

W domu Danny buduje roboty od 2004 roku i ma szafę pełną projektów, w tym roboty na kółkach, z kamerami, gąsienicami czołgowymi, nogami i ramionami wykonanymi z plastiku, tektury, metalu, tkanin, pudełek na drugie śniadanie czy zmodyfikowanych zabawek.

Danny jest autorem programu *Learn Robotics Programming*, opublikowanego w 2021 r. przez Packt Publishing oraz autorem artykułów dla magazynu dla *The MagPi*. Prowadzi kanał robotyki **Orionrobots** na YouTube'ie, a także zabiera swoje roboty na takie wydarzenia, jak Pi Wars czy Arduino Day. Danny jest także mentorem w CoderDojo KU, gdzie pokazuje dzieciom, jak programować w Pythonie, a także prowadzi kluby robotyki Lego.

Chciałbym podziękować społecznościom Pi Wars i Adafruit za odpowiedzi na moje trudne pytania, Mike'owi Moncrieffe'owi za sprawdzenie dla mnie diagramów, a mojemu zespołowi recenzentów za świetne informacje zwrotne uwzględnione w tej książce.

O recenzencie

Leo White jest zawodowym inżynierem oprogramowania i absolwentem University of Kent. Interesuje się elektroniką, drukiem 3D i robotyką. Najpierw zaczął programować na Commodore 64, a później napisał kilka aplikacji dla Acorn Archimedes. W ramach swojej codziennej pracy obecnie programuje dekodery. Wykorzystując jako podstawę Raspberry Pi, zmechanizował zabawki dla dzieci i sterował ramionami robotów, blogując w międzyczasie o swoich doświadczeniach i procesach. Prowadził prezentacje na Raspberry Jams i zgłaszał różne roboty do konkursu Pi Wars.

Spis treści

Kup książkę

3
Projektowanie podwozia robota w programie FreeCAD 47

4
Budowanie robota na bazie Pico..... 85

5
Sterowanie silnikami za pomocą Raspberry Pi Pico..... 109

Kup książkę

Część 2: Łączenie Raspberry Pi Pico z prostymi czujnikami i wyjściami

6
Pomiar ruchu za pomocą enkoderów na Raspberry Pi Pico 131

7
Planowanie i zakup większej liczby komponentów 159

Kup książkę

8
Mierzenie odległości w celu wykrywania obiektów za pomocą Pico 187

9
Zdalne sterowanie robotem Raspberry Pi Pico z użyciem Bluetooth LE 207

Część 3: Dodawanie kolejnych zachowań robotycznych do Raspberry Pi Pico

10
Użycie algorytmu PID do podążania wzdłuż ścian 231

Kup książkę

Kup książkę

13
Określanie położenia za pomocą metody Monte Carlo 301

14
Kontynuujemy podróż – kolejny robot..... 345

Kup książkę

Wprowadzenie

Robotyka to rozwijająca się dziedzina z zastosowaniami w każdej dziedzinie życia. Może się wydawać, że robotyka i związane z nią technologie są domeną dobrze wyposażonych laboratoriów uniwersytetów i firm high-tech. Jednak wielu aspektów robotyki – budowy i programowania – można się nauczyć i ćwiczyć je we własnym domu.

Główne obszary należące do robotyki to:

- Konstrukcje – projektowanie i budowanie mechanicznych platform
- Elektronika – czujniki, silniki i układy sterujące
- Oprogramowanie – kod dla bibliotek, interakcji czujników i zachowań

Ta książka ma na celu omówienie po trochu każdego obszaru, zwracając uwagę na podstawę w postaci projektu CAD, tworzenie fragmentów i składanie komponentów sprzętowych. Wprowadza nieco podstaw elektroniki cyfrowej, takich jak połączenia i magistrale danych. Jej celem jest zagłębienie się w czujniki i kod potrzebny do tworzenia interesujących zachowań z ich wykorzystaniem.

Istnieją książki o robotyce, które oferują teoretyczne wprowadzenie do robotyki, jednak celem tej pozycji książkowej jest zabranie Czytelnika w podróż pełną praktyki, zabawy i eksperymentów. Książka zawiera objaśnienia krok po kroku i ilustracje ułatwiające zrozumienie tematyki.

Budowanie własnych robotów w domu to świetny sposób na zdobycie umiejętności technologicznych. To doświadczenie technologii, które zastępuje nieprzeniknioną magię doświadczeniem w świecie rzeczywistym i pewnością siebie, aby można było budować więcej – każdy, kto zdobędzie nieco praktyki, też może zostać czarodziejem robotyki.

Dla kogo jest ta książka

Książka jest przeznaczona dla tych, którzy potrzebują praktycznego, krok po kroku wprowadzenia do projektowania, budowania i programowania robotów przy użyciu popularnego języka programowania Python. Jest ona również dla tych, którzy chcieliby uzyskać wprowadzenie do 3D w CAD-zie, sprzętu i czujników używanych w robotyce oraz zachowań robotów, które wykorzystują czujniki i sprzęt.

Ta książka będzie wartościowa dla twórców, uczniów i programistów, którzy chcą budować roboty w swoich domach lub warsztatach. Nie wymaga specjalistycznego warsztatu, a wszelkie potrzebne umiejętności i narzędzia zostaną wyjaśnione w odpowiednich jej częściach.

Kup książkę

Ci, którzy wcześniej mieli do czynienia z kodem, również uznają tę książkę za przydatną. Nie trzeba mieć żadnego doświadczenia z elektroniką czy tworzeniem różnych rzeczy. Spodziewaj się za to, że zdobędziesz pierwsze doświadczenia, ćwicząc techniki opisane w tej książce.

Oczekujemy, że będziesz posiadał chęci pogłębienia wiedzy i odrobinę odwagi w przeprowadzaniu eksperymentów z robotyką. Niezbędne jest praktyczne zastosowanie przedstawionych przykładów. Maksymalne wykorzystanie tej książki oznacza gotowość do stworzenia prawdziwego robota i przetestowania go.

O czym jest ta książka

Rozdział 1, Planowanie robota z użyciem Raspberry Pi Pico, przedstawia Raspberry Pi Pico w odniesieniu do innych głównych kontrolerów stosowanych w robotyce. Pokazuje zalety środowiska programistycznego CircuitPython i prowadzi przez proces tworzenia ogólnego planu budowy robota opartego na Pico. Rozdział ten zawiera listę zakupów sprzętu dla robota z pierwszej połowy książki, omawiając części i kompromisy przy ich wyborze.

Rozdział 2, Przygotowywanie Raspberry Pi Pico, przeprowadzi przez proces instalacji CircuitPython na Pico, a następnie przedstawia pierwsze kroki w pisaniu kodu w tym narzędziu. Omówi również lutowanie złączy szpilkowych (pinów) w Raspberry Pi Pico, dzięki czemu będzie można łączyć się z częściami robota.

Rozdział 3, Projektowanie podwozia robota w programie FreeCAD, przedstawia program FreeCAD przy okazji przekształcania ogólnego planu w projekt 3D w CAD-zie. Prezentuje, jak wykonać rysunki z projektu budowy części robota.

Rozdział 4, Budowanie robota na bazie Pico, pokazuje, w jaki sposób można wykorzystać rysunki CAD wraz z ręcznymi narzędziami do stworzenia części robota poprzez cięcie i wiercenie w kawałkach plastiku. Rozdział poprowadzi również poprzez montaż części, a następnie przez poprowadzenie przewodów i podłączenie elektroniki. W tym rozdziale robot zostanie po raz pierwszy podłączony do zasilania!

Rozdział 5, Sterowanie silnikami za pomocą Raspberry Pi Pico, wprowadza do sterowania silnikami za pomocą CircuitPython i Raspberry Pi Pico, pokazując, w jaki sposób można używać silników do wykonywania ruchów w linii prostej i skrętów oraz jak można kontrolować prędkość. W rozdziale tym pokazano, jak to połączyć w zaprogramowane sekwencje ruchu.

Rozdział 6, Pomiar ruchu za pomocą enkoderów na Raspberry Pi Pico, wprowadza pierwszy w książce robotyczny czujnik z enkoderami kół, pokazując, jak w kodzie wykryć ruch koła. W tym rozdziale omówiono urządzenie peryferyjne Raspberry Pi Pico PIO jako potężny sposób zarządzania tymi czujnikami.

Rozdział 7, Planowanie i zakup większej liczby komponentów, przygotowuje do następnej części książki, przedstawiając czujniki odległości, Bluetooth LE i **inercyjną jednostkę**

Kup książkę

pomiarową (ang. *inertial measurement unit*, **IMU**). Zawiera również porady dotyczące wyboru urządzeń i sposobu ich mocowania. Rozdział zawiera listę zakupów potrzebnych do dalszej części książki. Odświeżymy projekt części we FreeCAD, aby móc wykonać mocowania czujników, a następnie użyjemy narzędzi do ich wycięcia.

Rozdział 8, Mierzenie odległości w celu wykrywania obiektów za pomocą Pico, przeprowadzi przez dodawanie i podłączanie dwóch czujników odległości do robota. Rozdział zawiera informacje na temat komunikacji I2C, a następnie pokazuje, jak zaprogramować robota do komunikacji z czujnikami. W kolejnym kroku zbudujemy kod dla robota, aby autonomicznie omijał ściany.

Rozdział 9, Zdalne sterowanie robotem Raspberry Pi Pico z użyciem Bluetooth LE, zawiera porównanie możliwości połączeń bezprzewodowych i pokazuje, dlaczego Bluetooth LE był odpowiednim wyborem projektowym. Podłączymy moduł Bluetooth LE do robota, a następnie do przesyłania danych z czujnika przez to połączenie i wyświetlania danych wyjściowych na smartfonie rozszerzymy dotychczasowy kod. Zobaczymy również, jak sterować robotem z poziomu smartfona.

Rozdział 10, Użycie algorytmu PID do podążania wzdłuż ścian, zawiera wprowadzenie do algorytmu PID, podstawowego w robotyce bloku składowego sterowania zachowaniem na podstawie czujników/wyjść. Zbudujemy demonstrację podążania wzdłuż ścian z użyciem czujnika odległości, a następnie pokażemy, jak dostroić PID z użyciem wykresów na smartfonie za pośrednictwem Bluetooth LE.

Rozdział 11, Sterowanie ruchem z użyciem enkoderów przez Raspberry Pi Pico, ponownie omawia enkodery, pokazując, jak przekonwertować ich dane wyjściowe na jednostki zrozumiałe dla człowieka. Dowiemy się, jak połączyć te czujniki z algorytmem PID, aby móc sterować prędkością silnika i jechać po linii prostej. Następnie zaprogramujemy robota tak, żeby przejechał określoną odległość po linii prostej z określoną prędkością.

Rozdział 12, Wykrywanie orientacji za pomocą IMU i Raspberry Pi Pico, wprowadza IMU, czujnik, który pozwala określić orientację robota. Rozdział ten zawiera przewodnik dotyczący podłączania czujnika i jego kalibracji. Użyjemy IMU z algorytmem PID do uzyskania zachowania, w którym robot będzie zawsze skierowany na północ. Na koniec rozdziału pokażemy, jak zaprogramować robota, aby wykonał określony skręt z użyciem IMU.

Rozdział 13, Określanie położenia za pomocą metody Monte Carlo, pokaże, jak zaprogramować robota w celu określenia jego prawdopodobnej lokalizacji na arenie. W tym rozdziale do zbudowania areny z płyty piankowej użyjemy rzutów i zamodelujemy ją w kodzie. Pokażemy, jak zwizualizować tę przestrzeń w komputerze za pomocą Bluetooth LE z Matplotlib. Następnie dowiemy się o ruchach robotem na podstawie danych wejściowych z czujników. W rozdziale pokazano, w jaki sposób wiele zachowań robota może ze sobą współpracować w pewnym kontekście. Przedstawimy wprowadzenie do używania w ruchach robota algorytmów prawdopodobieństwa, tworzenia prognoz i ich udoskonalania.

Kup książkę

Rozdział 14, Kontynuujemy podróż – kolejny robot, zawiera podsumowanie tematów poznanych w książce, wraz z informacjami o tym, jak zagłębić się w każdy z nich. Ten rozdział zawiera pomysły i obszary badawcze, w których można rozszerzyć poszczególne aspekty robota, a następnie dalsze sugestie dotyczące budowania bardziej ambitnych robotów i rozwijania swoich umiejętności. Rozdział poleca również społeczności związane z robotyką, w których warto uczestniczyć.

Aby jak najlepiej wykorzystać tę książkę

Będzie potrzebna znajomość kilku podstaw Pythona, takich jak zmienne, pętle, wyrażenia warunkowe i funkcje. Dla przedstawionych w tej książce aspektów związanych z budowaniem robotów zalecane jest dobrze oświetlone i wentylowane miejsce na biurku. Pomocny będzie dostęp do narzędzi ręcznych, aczkolwiek w dalszej części zamieścimy informację, jakie narzędzia należy zakupić. Przykładowy kod robota został przetestowany w CircuitPython 7.2.0 na Raspberry Pi Pico, natomiast powinien działać również z nowszymi wersjami. Przykłady kodu komputerowego zostały przetestowane w Pythonie 3.9.

Oprogramowanie/sprzęt omówiony w książce	Wymagania dotyczące systemu operacyjnego
Thonny > 3.3 lub Mu Editor > 1.1	macOS, Linux lub Windows
Python 3.7 lub nowszy	macOS, Linux lub Windows
Matplotlib 3.6.1 lub nowszy	macOS, Linux lub Windows
NumPy 1.23.4 lub nowszy	macOS, Linux lub Windows
Bleak (biblioteka BLE Pythona) 0.19.0 lub nowsza	macOS, Linux lub Windows
Wolny port USB	macOS, Linux lub Windows
Smartfon/tablet z Bluetooth LE (Bluetooth > 4.0)	iOS lub Android
Adafruit Bluefruit LE Connect > 3.3.2	iOS lub Android
Laptop z obsługą Bluetooth LE (lub klucz sprzętowy BLE)	macOS, Linux lub Windows
FreeCAD	macOS, Linux lub Windows
Raspberry Pi Pico	
CircuitPython > 7.2.0	Raspberry Pi Pico

Thonny jest dostarczany z wbudowaną instalacją Pythona 3.x. Do instalowania pakietów w Thonny Pythonie można użyć menu **Narzędzia | Otwórz powłokę systemową**.

W przypadku korzystania z cyfrowej wersji tej książki radzimy wpisywać kod samodzielnie lub skorzystać z kodu w repozytorium GitHuba dla tej książki (link znajduje się w następnym punkcie). Pomoże to uniknąć potencjalnych błędów związanych z kopiowaniem i wklejaniem kodu.

Kup książkę

Pomoc do tej książki można znaleźć w:

- Zgłaszaniu błędów w repozytorium książki w GitHubie pod adresem `https://github.com/PacktPublishing/Robotics-at-Home-with-Raspberry-Pi-Pico`
- Wysyłając pytanie przez Discord na `https://discord.gg/2VHYY3FkXV`

Pobieranie plików z przykładowym kodem

Pliki z przykładowym kodem dla tej książki można pobrać z GitHub pod adresem `https://github.com/PacktPublishing/Robotics-at-Home-with-Raspberry-Pi-Pico`. Jeśli pojawi się aktualizacja kodu, zostanie ona wprowadzona w repozytorium GitHuba.

Dysponujemy również innymi pakietami kodu z naszego bogatego katalogu książek i filmów. Są one dostępne pod `https://github.com/PacktPublishing/`. Sprawdź je!

Pobieranie kolorowych grafik

Udostępniamy również plik PDF zawierający kolorowe zrzuty ekranu i diagramy użyte w tej książce. Można go pobrać pod tym adresem: `https://packt.link/7x3ku`.

Przyjęte konwencje

W tej książce zastosowano kilka konwencji w odniesieniu do tekstu.

`Kod w tekście`: wskazuje wyrażenia z kodu w tekście, nazwy tabel bazy danych, nazwy folderów, nazwy plików, rozszerzenia plików, ścieżki, fikcyjne adresy URL, dane wprowadzone przez użytkownika i nazwy użytkowników na Twitterze. Oto przykład: „Aby uruchomić ten kod, należy wysłać bibliotekę `pio_encoders.py`, zaktualizowany plik `robot.py`, a następnie `Measure_fixed_time.py`”.

Blok kodu zostanie przedstawiany w następujący sposób:

```
import time
import board
import digitalio
led = digitalio.DigitalInOut(board.LED)
led.direction = digitalio.Direction.OUTPUT
while True:
    led.value = True
    time.sleep(0.5)
    led.value = False
    time.sleep(0.5)
```

Kup książkę

Gdy będziemy chcieli zwrócić uwagę na określoną część bloku kodu, odpowiednie linie lub elementy zostaną pogrubione:

```
>>> print("Hello, world!")
Hello, World!
>>>
```

Wszelkie podawanie danych wejściowych lub wyjścia w wierszu poleceń będą zapisywane w następujący sposób:

```
code.py output:
4443 4522
```

Pogrubienie: oznacza nowy termin, ważne pojęcie lub słowa widoczne na ekranie. Na przykład będą pogrubiane słowa w menu lub oknach dialogowych. Oto przykład: „Uruchom Mu Editor, a gdy uruchomi się, kliknij przycisk **Mode**. Wybierz **CircuitPython**."

> **Wskazówki lub ważne uwagi**
>
> Wyglądają tak.

Kontakt

Opinie naszych czytelników są zawsze mile widziane.

Ogólna opinia: Jeśli masz pytania dotyczące dowolnego aspektu tej książki, wyślij do nas wiadomość e-mail na adres `customercare@packtpub.com` i w temacie wiadomości podaj tytuł książki.

Errata: Chociaż dołożyliśmy wszelkich starań, aby zapewnić prawidłowość naszych treści, błędy się zdarzają. Jeśli znalazłeś błąd w tej książce, bylibyśmy wdzięczni, gdybyś nam to zgłosił. Odwiedź stronę `www.packtpub.com/support/errata` i wypełnij formularz.

Piractwo: Jeśli w Internecie natkniesz się na jakiekolwiek nielegalne kopie naszych utworów w jakiejkolwiek formie, będziemy wdzięczni za podanie adresu lokalizacji lub nazwy strony internetowej. Prosimy o kontakt pod adresem `copyright@packt.com` z linkiem do materiału.

Jeśli chcesz zostać autorem: Jeśli jest temat, w którym jesteś ekspertem i jesteś zainteresowany napisaniem lub współtworzeniem książki, odwiedź stronę `authors.packtpub.com`.

Kup książkę

Część 1: Podstawy – Wprowadzenie do robotyki z Raspberry Pi Pico

W tej części wykonamy pierwsze kroki w poznawaniu Raspberry Pi Pico, następnie zaplanujemy i zbudujemy na jego bazie robota oraz napiszemy dla niego pierwszy kod, który wprawi go w ruch.

Ta część zawiera następujące rozdziały:

- Rozdział 1, Planowanie robota z Raspberry Pi Pico
- Rozdział 2, Przygotowanie Raspberry Pi Pico
- Rozdział 3, Projektowanie podwozia robota w programie FreeCAD
- Rozdział 4, Budowanie robota na bazie Pico
- Rozdział 5, Sterowanie silnikami za pomocą Raspberry Pi Pico

Kup książkę

Kup książkę

1

Planowanie robota z użyciem Raspberry Pi Pico

Planując, stwarzamy największą szansę na powodzenie misji. Chcemy budować roboty w osiągalny sposób. Zacznijmy od planu! Wykorzystamy ten plan do zbadania, dlaczego **Raspberry Pi Pico** świetnie się do tego nadaje i sporządzimy listę zakupów!

W tym rozdziale poznamy możliwości Raspberry Pi Pico. Odkryjemy **CircuitPython** i dowiemy się, dlaczego jest to świetny język dla Raspberry Pi Pico. Dodatkowo, zaplanujemy projekt robota i poznamy kompromisy w dokonywaniu wyborów dotyczących robota już na wczesnym etapie projektu. Sprawdzimy, czy wszystko w naszym robocie pasuje do siebie, opracowując potrzebne części i narzędzia, a także przedstawiając sugestie, jak je zdobyć.

Na końcu rozdziału będziemy mieli zarówno plan, jak i części, dzięki którym będziemy gotowi do zbudowania robota. Ponadto, będziemy znali wstępny proces przydatny w przypadku tworzenia innych robotów i przygotowania się do odniesienia sukcesu z nimi.

W tym rozdziale omówimy następujące główne tematy:

- Co to jest Raspberry Pi Pico i dlaczego nadaje się do robotyki?
- Co to jest CircuitPython?
- Planowanie robota z użyciem Raspberry Pi Pico
- Test dopasowania robota z Raspberry Pi Pico
- Zalecana lista zakupów dla podstaw robota

Kup książkę

Wymagania techniczne

W miarę postępów w tym rozdziale będziemy zajmować się niezbędnym sprzętem oraz listą zakupów. Dlatego w tym podrozdziale skupimy się tylko na tym, czego będziemy potrzebować fizycznie i na komputerze, aby móc rozpocząć.

Będziemy potrzebować:

- Trochę cienkiego kartonu
- Linijkę, ołówek i nożyczki
- Dobrą przeglądarkę internetową z dostępem do Internetu

Co to jest Raspberry Pi Pico i dlaczego nadaje się do robotyki?

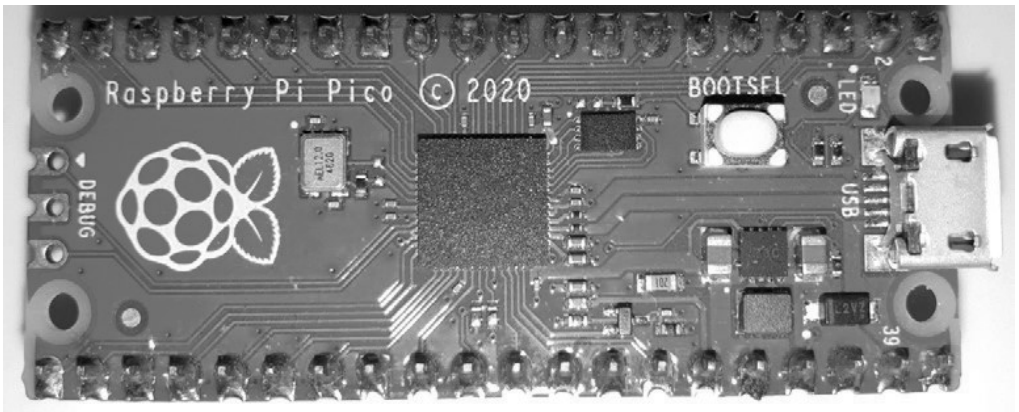
Sercem każdego robota jest **kontroler**. Zwykle jest to urządzenie komputerowe, które jest odpowiedzialne za wykonywanie kodu robota w celu realizacji jego zadań i zachowań. Wybór kontrolera jest kluczowym wyborem w projektowaniu robotów. Można albo wychodzić z założenia *Mam ten kontroler, co mogę z nim zrobić?* albo z założenia *Które kontrolery mają możliwości, których potrzebuję dla tego konkretnego robota?*

W tym punkcie przyjrzymy się bliżej temu, co Raspberry Pi Pico oferuje jako kontroler i kompromisom, jakie zawiera. Zbadamy, dlaczego jest on przydatny w robotyce i dlaczego może być częścią większego, bardziej interesującego systemu.

Dodatkowo, zagłębimy się w szczegóły jego interfejsów i tego, w jaki sposób będą one dla nas przydatne.

Mikrokontroler obsługujący język Python

Zacznijmy od przyjrzenia się Raspberry Pi Pico i odkrycia, co zawiera. Poniższe zdjęcie przedstawia Raspberry Pi Pico:



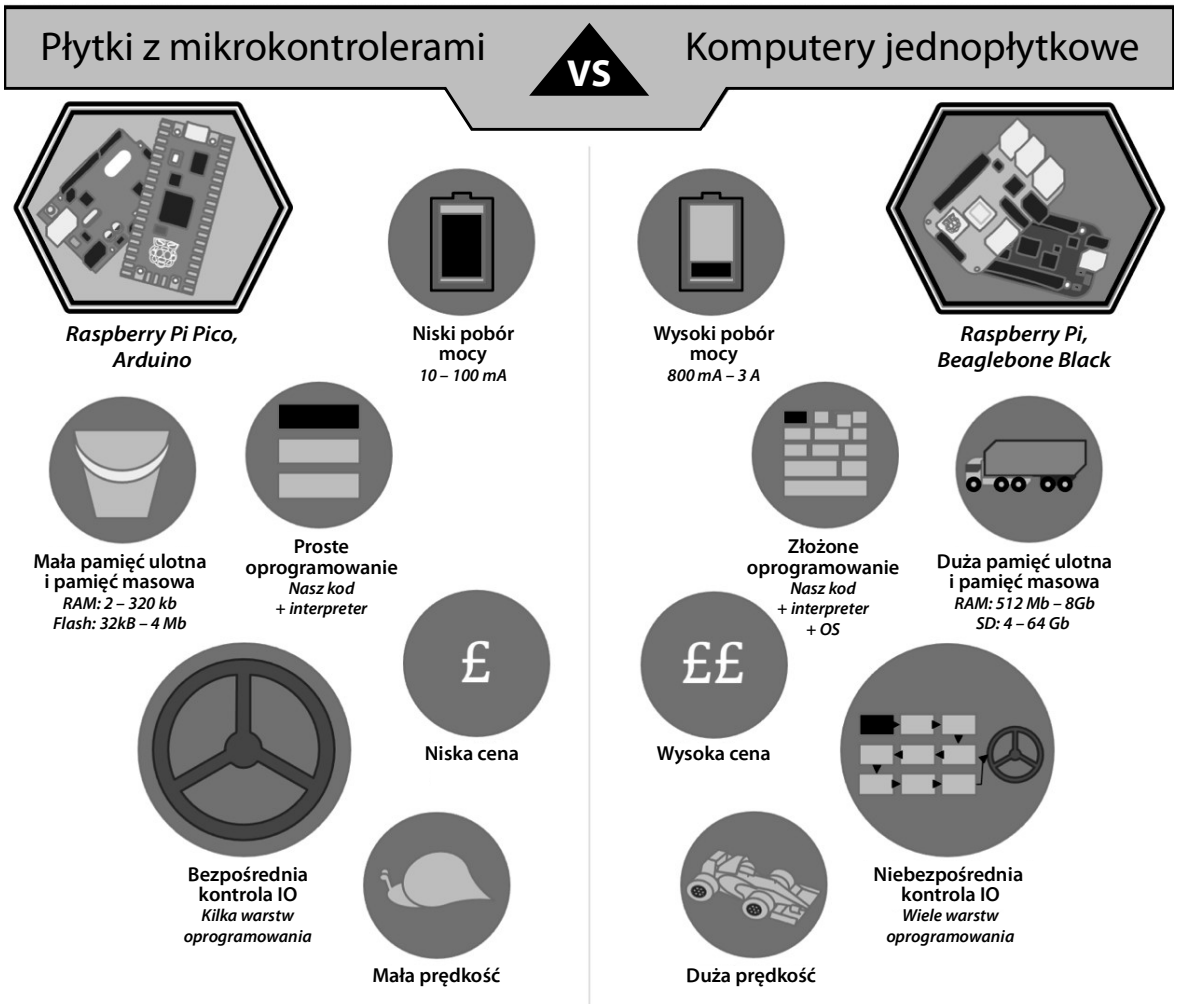
Rysunek 1.1 — Raspberry Pi Pico

Kup książkę

Raspberry Pi Pico, jak pokazano na rysunku 1.1, to mikrokontroler *RP2040* na płycie zaprojektowanej dla Raspberry Pi. Ten **mikrokontroler** jest małym modułem obliczeniowym zaprojektowanym tak, aby gładko współpracował ze sprzętem. Po prawej stronie posiada złącze USB służące do zasilania lub programowania za pomocą komputera. Dioda LED jest przydatna do debugowania. Ponadto, blisko krawędzi znajduje się wiele pinów **wejścia/wyjścia** (ang. input/output, IO), które służą do podłączania komponentów. To właśnie za ich pomocą we/wy dzieje się magia, jeśli chodzi o sterowanie robotami!

Kontrolery używają pinów IO do zapisu i odczytu z dołączonego sprzętu. W celu wymiany danych z innymi urządzeniami mogą grupować piny w magistrale (które omówimy bardziej szczegółowo później). Dodatkowo, na wyjściach mogą generować różne kształty fali w celu sterowania silnikami i diodami LED.

Brzmi to bardzo podobnie jak w przypadku innych modeli **Raspberry Pi**. Jest to jednak zupełnie inna klasa komputerów. **Raspberry Pi Pico** ma więcej wspólnego z płytką **Arduino**. Przyjrzyjmy się bliżej na poniższym diagramie, co oznacza ta różnica:



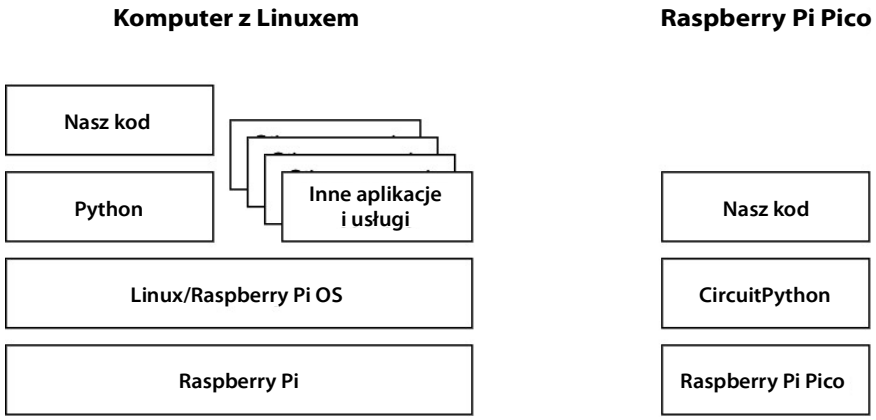
Rysunek 1.2 — Płytki z mikrokontrolerami vs. komputery jednopłytkowe

Kup książkę

Rysunek 1.2 pokazuje, że chociaż płytki z mikrokontrolerami, takie jak Raspberry Pi Pico i Arduino, mogą wyglądać podobnie do **komputerów jednopłytkowych** (ang. single-board computers, **SBC**), takich jak Raspberry Pi 4 lub BeagleBone, różnią się w kluczowych obszarach. Dla przykładu, różnią się pamięcią masową, szybkością procesora, kosztem, złożonością oprogramowania i tym, jak blisko sprzętu działa oprogramowanie.

Chociaż Raspberry Pi Pico doskonale nadaje się do kontrolowania sprzętu, takiego jak roboty, to nie nadaje się już tak dobrze do zadań wymagających dużej ilości pamięci lub procesora, takich jak sztuczna inteligencja czy rozpoznawanie obrazów. Istnieje rodzaj systemów robotów znany jako **koń i jeździec**, który łączy SBC (na przykład Raspberry Pi 4) odpowiedzialny za złożone przetwarzanie z mikrokontrolerem (na przykład Pico) odpowiedzialnym za sterowanie sprzętem.

Niska złożoność oznacza, że czas uruchamiania kodu na mikrokontrolerze jest bardzo krótki, co oznacza, że nasz kod nie musi współistnieć z innym oprogramowaniem w systemie operacyjnym. Przyjrzyjmy się poniższemu schematowi blokowemu:



Rysunek 1.3 – Uruchamianie kodu na Raspberry Pi w porównaniu z Pico

Powyższy diagram przedstawia architekturę oprogramowania w Raspberry Pi w porównaniu z Raspberry Pi Pico. Pokazuje, że komputer z systemem Linuks, taki jak Raspberry Pi, ma dodatkowe warstwy oprogramowania wraz z konkurującymi aplikacjami działającymi równolegle z naszym kodem.

Oprócz tego kontrolery mają **przerwania**. Za ich pomocą można powiadomić kod, że coś się zmieniło, na przykład stan pinów IO. Znajdziemy je także w innych modelach Raspberry Pi, ale ponownie są one kontrolowane przez ten natrętny system operacyjny. W Pico i innych mikrokontrolerach mamy większą kontrolę nad tym, co się dzieje. A gdy coś się zmienia na pinach IO, umożliwia stworzenie responsywnego kodu z przewidywalnym czasem reakcji.

Jak więc Raspberry Pi Pico wypada w porównaniu z Arduino Uno? Poniższa tabela przedstawia szczegóły z ich specyfikacji i kart katalogowych:

Kup książkę

Specyfikacja	Raspberry Pi Pico	Arduino Uno
Liczba cyfrowych pinów IO	26	14
Liczba analogowych pinów IO	4	6
Procesor	Dwurdzeniowy RP2040, taktowanie 133 MHz	Jednordzeniowy ATmega, taktowanie 16 MHz
Pamięć flash (przechowywanie kodu)	2 MB	32 kB (plus 1 kB EPROM)
RAM (pamięć do uruchamiania kodu)	264 kB	2 kB

Tabela 1.1 – Porównanie Pico z Arduino Uno

Powyższa tabela pokazuje, że Raspberry Pi Pico ma szybszy procesor wielordzeniowy, a także więcej pamięci masowej i cyfrowych pinów IO. Ponadto, Raspberry Pi Pico ma unikalny system **programowalnych IO** (ang. programmable IO, **PIO**), oferujący wyjątkową elastyczność w organizowaniu danych do i z tych pinów. Oficjalne płytki Pico są również tańsze niż oficjalne płytki Arduino.

Innym miejscem, w którym Raspberry Pi Pico wypada korzystniej w porównaniu z Arduino, jest możliwość użycia Pythona (CircuitPython lub MicroPython). Wiele mikrokontrolerów, takich jak Arduino, do programowania wymaga C/C++, co może być trudne dla początkujących. Python jest łatwiejszy do zrozumienia, pozwala na tworzenie złożonych i interesujących struktur danych, a także można uzyskać dostęp do wielu bibliotek kodu.

W skrócie, kluczowe cechy Raspberry Pi Pico są następujące:

- Mikrokontroler – charakteryzuje się niskim poborem mocy, a ponadto jest mały w porównaniu z SBC.
- Posiada kontrolowalne i elastyczne możliwości związane z IO.
- Niski koszt w porównaniu z wieloma płytkami z mikrokontrolerami i większością SBC.
- Możliwość programowania w języku Python.

Wiele właściwości, które są przypisane Raspberry Pi Pico, wynika z *RP2040* – głównego układu Pico, dostępnego również w formach innych niż w Raspberry Pi Pico.

Elastyczność IO jest najciekawszą cechą Raspberry Pi Pico, więc przyjrzyjmy się temu bliżej.

Interfejsy Raspberry Pi Pico dla czujników i urządzeń

Raspberry Pi Pico posiada wiele interfejsów do podłączania sprzętu, a także unikalny system PIO. W tym punkcie przyjrzymy się każdemu typowi interfejsu.

Cyfrowy pin IO to podstawowy system IO w Raspberry Pi Pico. Wyjście może być włączone lub wyłączone, co świetnie nadaje się do włączania i wyłączania diod LED, natomiast

Kup książkę

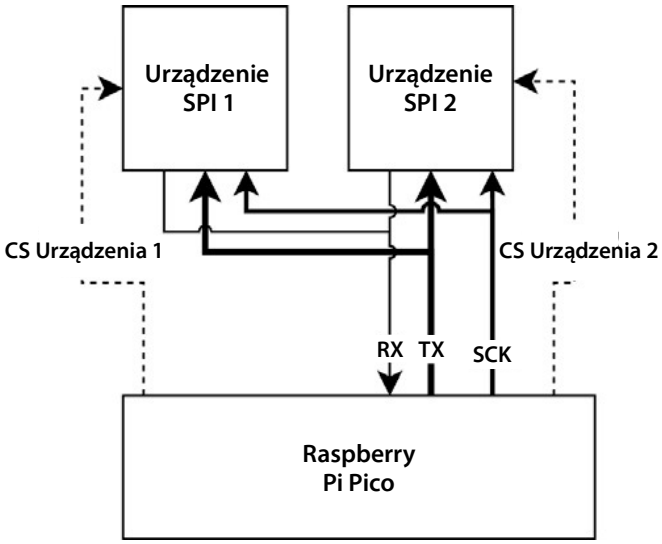
nie można kontrolować ich jasności. W analogiczny sposób wejście może również wykrywać stany włączenia lub wyłączenia. Raspberry Pi Pico ma 26 takich pinów.

Modulacja szerokości impulsu (ang. pulse-width modulation, **PWM**) jest wypełnieniem sygnału w celu sterowania wyjściami, takimi jak diody LED i silniki – w tym silniki prądu stałego, silniki krokowe i silniki serwo. Piny PWM wysyłają impulsy fali prostokątnej ze zmiennym (modulowanym) współczynnikiem włączenia-wyłączenia (szerokością impulsów). Zmiana szerokości impulsu powoduje zmianę jasności diody LED, prędkości silnika lub pozycji silnika serwo. Raspberry Pi Pico posiada 16 kanałów PWM, dzięki czemu może sterować wieloma takimi urządzeniami jednocześnie. Tego typu piny PWM do napędzania silników wymagają dodatkowego urządzenia dostarczającego moc.

Piny wejść **analogowych** wykrywają wartości napięcia między **masą** (**GND**) a 3,3V. Jest to dobre do współpracy z prostymi czujnikami, takimi jak czujniki światła, joysticki, suwaki/pokrętła, czujniki temperatury i mierzącymi natężenia prądów (przy użyciu nieco bardziej rozbudowanych obwodów). Raspberry Pi Pico ma trzy takie wejścia.

Uniwersalny asynchroniczny nadajnik-odbiornik (ang. *universal asynchronous receiver-transmitter*, **UART**) steruje portem szeregowym. Może wysyłać strumienie danych do i z urządzeń za pomocą dwóch pinów: pinu **nadawczego TX** i pinu **odbiorczego RX**. Dzięki temu jest w stanie wysyłać/odbierać dane, które są bardziej skomplikowane niż tylko zmieniający się poziom napięcia. Raspberry Pi Pico posiada dwa niezależne interfejsy UART.

Pico ma dwa kontrolery magistrali **Serial Peripheral Interface** (**SPI**). SPI wykorzystuje cztery piny, jak pokazano na poniższym schemacie:



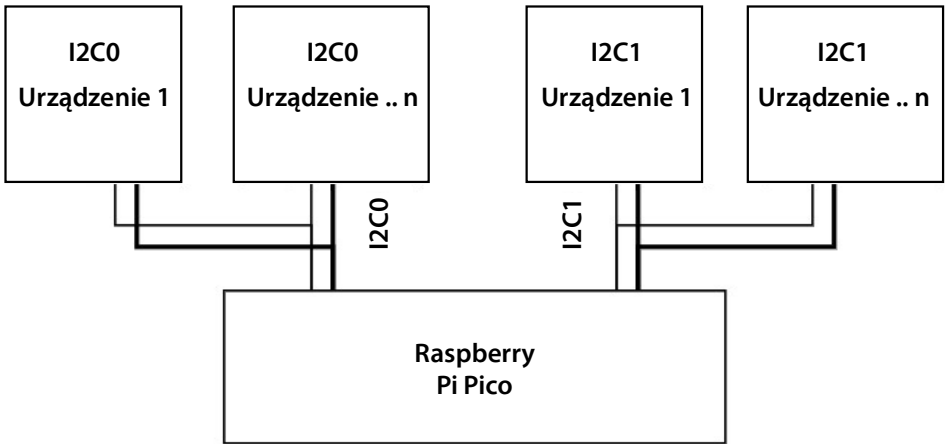
Rysunek 1.4 – Użycie magistrali SPI w Raspberry Pi Pico

Powyższy diagram przedstawia użycie magistrali SPI Raspberry Pi Pico do podłączenia dwóch urządzeń – na przykład wyświetlaczy lub czujników. Magistrala zawiera linię **nadawczą** (**TX**), znaną również jako **Controller Out/Peripheral In** (**COPI**) lub **Microcontroller Out/Sensor In** (**MOSI**) do przesyłania danych z kontrolera, linię **odbiorczą** (**RX**), znaną

Kup książkę

również jako **Controller In/Peripheral Out** (**CIPO**) lub **Microcontroller In/Sensor Out** (**MISO**) do odbierania danych z powrotem przez kontroler, **SCK** (zegar do synchronizacji sygnału) i **Chip Select** (**CSEL/CS**) – linia wyboru układu podrzędnego. SPI wykorzystuje wskazywanie układu podrzędnego, aby umożliwić komunikację z wieloma urządzeniami, jak pokazano przerywanymi liniami dla **CS Urządzenia 1** i **CS Urządzenia 2**. Szczegółowe informacje na temat aktualnych akronimów SPI znajdują się w `https://makezine.com/article/maker-news/mosi-miso-and-140-years-of-wrong/`.

Inter-Integrated Circuit (**I2C**) to magistrala danych przeznaczona do komunikacji między układami scalonymi, takimi jak czujniki, urządzenia pamięci i urządzenia wyjściowe. Magistrala I2C ma linię danych (która jest często nazywana *SDA – Serial Data*) i linię zegara (która jest często nazywana *SCL – Serial Clock*), utrzymując w ten sposób synchronizację. Kilka urządzeń może współdzielić magistralę I2C, wysyłając/odbierając dane, posługując się adresami, tak jak te na poniższym diagramie:



Rysunek 1.5 – Magistrale I2C w Raspberry Pi Pico

Rysunek 1.5 pokazuje Pico oraz kilka podrzędnych urządzeń peryferyjnych połączone przez dwie niezależne magistrale I2C, które można przypisać do różnych konfiguracji pinów, przy czym niektóre urządzenia mają ten sam adres, ale są połączone do różnych magistral I2C. Ponadto, I2C może adresować rejestry (takie jak lokalizacje pamięci) w urządzeniach. Później użyjemy I2C do komunikacji z czujnikami.

Oprócz tego Raspberry Pi Pico posiada PIO. PIO to funkcja unikalna dla Pico. PIO składa się z dwóch bloków z czterema *maszynami stanowymi*. W każdym z bloków można wykonywać prosty kod niezależnie od głównego procesora i kontrolować jeden lub więcej pinów w celu wysyłania lub odbierania przez nie danych. Maszyna jednostanowa może kontrolować wszystkie piny, jeśli byłoby to użyteczne dla kodu. Dodatkowo każda maszyna stanowa jest wyposażona w bufory do przechowywania danych, dopóki nie będzie można ich przetransferować. Poniżej przedstawiono przykładowy schemat blokowy systemu PIO:

Kup książkę