

Wydanie II

# Arduino dla początkujących Podstawy i szkice



Mc  
Graw  
Hill  
Education

Simon Monk

Helion

Tytuł oryginału: Programming Arduino: Getting Started with Sketches, Second Edition

Tłumaczenie: Konrad Matuk

ISBN: 978-83-289-4054-3

Original edition copyright © 2016, 2012 by McGraw-Hill Education.  
All rights reserved.

Polish edition copyright © 2019, 2023, 2026 by Helion S.A.  
All rights reserved.

Arduino is a trademark of Arduino LLC.

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 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.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: [helion.pl](http://helion.pl) (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

[helion.pl/user/opinie/ard2vv](http://helion.pl/user/opinie/ard2vv)

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

# Spis treści

O autorze .....	9
Podziękowania .....	10
Przedmowa .....	11
Wstęp .....	13
Czym jest Arduino? .....	13
Co będzie potrzebne? .....	14
Korzystanie z niniejszej książki .....	14
Przykłady kodu .....	15
Rozdział 1. Oto Arduino .....	17
Mikrokontrolery .....	17
<i>Płyty rozwojowe</i> .....	18
Płyta Arduino .....	19
<i>Zasilanie</i> .....	19
<i>Złącza zasilania</i> .....	20
<i>Wejścia analogowe</i> .....	20
<i>Złącza cyfrowe</i> .....	20
<i>Mikrokontroler</i> .....	20
<i>Pozostałe podzespoły</i> .....	21
Początki Arduino .....	22
Rodzina płyt Arduino .....	22
<i>Uno, Duemilanove i Diecimila</i> .....	23
<i>Mega i Due</i> .....	24
<i>Micro i Small</i> .....	24
<i>Yun</i> .....	25
<i>Lilypad</i> .....	26
<i>Inne „oficjalne” płytki</i> .....	26
Inne klony i odmiany Arduino .....	27
Podsumowanie .....	28

Rozdział 2. Rozpoczynamy przygodę z Arduino .....	29
Zasilanie .....	29
Instalacja oprogramowania .....	30
Ładowanie pierwszego szkicu .....	30
Aplikacja Arduino .....	35
Podsumowanie .....	36
Rozdział 3. Podstawy języka C .....	39
Programowanie .....	39
Czym jest język programowania? .....	40
Blink po raz kolejny .....	44
Zmienne .....	46
Eksperymentowanie w języku C .....	48
<i>Zmienne numeryczne i arytmetyka</i> .....	49
Polecenia .....	51
<i>if</i> .....	52
<i>for</i> .....	53
<i>while</i> .....	56
Stałe .....	56
Podsumowanie .....	57
Rozdział 4. Funkcje .....	59
Czym jest funkcja? .....	59
Parametry .....	60
Zmienne globalne, lokalne i statyczne .....	61
Zwracanie wartości .....	64
Zmienne innych typów .....	65
<i>float</i> .....	65
<i>boolean</i> .....	66
<i>Inne typy danych</i> .....	67
Styl zapisu kodu .....	68
<i>Wcięcia</i> .....	68
<i>Nawiasy klamrowe otwierające</i> .....	69
<i>Białe znaki</i> .....	69
<i>Komentarze</i> .....	70
Podsumowanie .....	71
Rozdział 5. Tablice i łańcuchy .....	73
Tablice .....	73
<i>Zastosowanie tablic do alfabetu Morse'a i sygnału SOS</i> .....	76
Tablice łańcuchów .....	77
<i>Literaly łańcuchowe</i> .....	77
<i>Zmienne łańcuchowe</i> .....	78
Tłumacz alfabetu Morse'a .....	79
<i>Dane</i> .....	80
<i>Zmienne globalne i funkcja setup</i> .....	81
<i>Funkcja loop</i> .....	81

<i>Funkcja flashSequence</i> .....	83
<i>Funkcja flashDotOrDash</i> .....	84
<i>Składanie całości programu</i> .....	85
Podsumowanie .....	86
Rozdział 6. Wejścia i wyjścia .....	87
Wyjścia cyfrowe .....	87
Wejścia cyfrowe .....	90
<i>Rezystor podwyższający</i> .....	91
<i>Wewnętrzny rezystor podwyższający</i> .....	94
<i>Usuwanie stuków</i> .....	94
Wyjścia analogowe .....	99
Wejścia analogowe .....	101
Podsumowanie .....	102
Rozdział 7. Standardowa biblioteka Arduino .....	103
Liczby losowe .....	103
Funkcje matematyczne .....	105
Manipulacja bitami .....	106
Zaawansowane funkcje wejścia i wyjścia .....	108
<i>Generowanie tonów</i> .....	108
<i>Wprowadzanie rejestru przesuwnego</i> .....	109
Przerwania .....	110
Podsumowanie .....	112
Rozdział 8. Zapisywanie danych .....	113
Stałe .....	113
Przechowywanie danych w pamięci flash .....	114
EEPROM .....	116
<i>Przechowywanie wartości zmiennej typu int w pamięci EEPROM</i> .....	117
<i>Korzystanie z biblioteki AVR EEPROM</i> .....	117
<i>Przechowywanie wartości typu float w pamięci EEPROM</i> .....	118
<i>Przechowywanie łańcucha w pamięci EEPROM</i> .....	119
<i>Wymazywanie zawartości pamięci EEPROM</i> .....	120
Kompresja .....	121
<i>Kompresja zakresu</i> .....	121
Podsumowanie .....	122
Rozdział 9. Wyświetlacze .....	123
Alfanumeryczne wyświetlacze LCD .....	124
Tablica wyświetlająca komunikaty za pośrednictwem interfejsu USB .....	124
Korzystanie z wyświetlacza .....	127
Inne funkcje biblioteki wyświetlacza LCD .....	127
Graficzne wyświetlacze OLED .....	128
Podłączanie wyświetlacza OLED .....	128
Szkielet .....	129
Podsumowanie .....	131

Rozdział 10. Arduino i programowanie aplikacji internetu rzeczy .....	133
Komunikacja z serwerami sieciowymi .....	134
HTTP .....	134
HTML .....	134
Arduino w roli serwera sieci Web .....	136
Sterowanie pracą Arduino za pośrednictwem sieci .....	138
Serwer sieciowy Node MCU .....	143
Sterowanie pracą płytki Node MCU za pośrednictwem sieci .....	147
Wywoływanie usług sieciowych .....	150
Współpraca płytki Arduino Uno z usługą IFTTT .....	152
Płytki Node MCU ESP8266 i usługa IFTTT .....	154
Inne opcje projektów internetu rzeczy .....	155
<i>Arduino Yun</i> .....	156
<i>Particle Photon</i> .....	156
Podsumowanie .....	156
Rozdział 11. C++ i biblioteki .....	157
Mechanizmy obiektowe .....	157
<i>Klasy i metody</i> .....	157
Przykład wbudowanej biblioteki .....	158
Tworzenie bibliotek .....	158
<i>Plik nagłówkowy</i> .....	158
<i>Plik implementacji</i> .....	160
<i>Uzupełnianie swojej biblioteki</i> .....	160
Podsumowanie .....	162
Skorowidz .....	165

## Rozdział 1.

# Oto Arduino

Arduino jest mikrokontrolerem. Platforma Arduino została bardzo ciepło przyjęta przez miłośników elektroniki. Jest ona otwarta i łatwa w obsłudze, a więc może się przydać każdej osobie, która chce zrealizować swój własny projekt związany z elektroniką.

Arduino kontroluje urządzenia podłączone do swoich złączy. Na przykład włącza i wyłącza światła lub silniki. Może być również stosowane do odczytu temperatury lub ilości padającego światła. Z tego powodu Arduino określane jest czasem mianem **fizycznego komputera**. Wszystkie elementy podłączone do Arduino mogą być również sterowane z poziomu komputera, do którego podłączony jest ten mikrokontroler. W tym wypadku Arduino odgrywa rolę interfejsu.

Niniejszy rozdział stanowi wstęp do Arduino. Opisano tutaj rys historyczny tego urządzenia, a także omówiono jego budowę.

---

## Mikrokontrolery

Sercem Arduino jest mikrokontroler. Wszystkie pozostałe elementy na płytce układu zajmują się zasilaniem układu lub pozwalają na komunikację układu z komputerem.

Mikrokontroler jest tak naprawdę małym komputerem zamkniętym w pojedynczym chipie. Ma on wszystko to, co miały pierwsze komputery przeznaczone do użytku domowego, a nawet więcej. Arduino posiada procesor, kilobajt lub dwa kilobajty pamięci o dostępie bezpośrednim (RAM) służącej do przechowywania danych, kilka kilobajtów pamięci stałej programowalnej elektrycznie (EEPROM) lub pamięci flash służącej do przechowywania programów, a także złącza wejścia i wyjścia. Złącza te służą do łączenia mikrokontrolera z innymi elementami elektronicznymi. Wejścia mogą odbierać zarówno sygnał cyfrowy (stan zwarcia i rozwarca), jak i analogowy (napięcie na złączu). Pozwala to między innymi na podłączenie różnorodnych czujników temperatury, czujników ciśnienia akustycznego oraz światłomierzy.

Wyjścia mogą działać również w trybie cyfrowym lub analogowym. Możesz skonfigurować dany pin tak, aby działał w trybie zwarcia lub rozwarcia (0 woltów lub 5 woltów). A więc można w ten sposób bezpośrednio włączać lub wyłączać diodę elektroluminescencyjną (LED). Wyjścia mogą być również stosowane do sterowania urządzeniami większej mocy, takimi jak silniki. Złącza wyjściowe mogą również działać w trybie analogowym. A więc można ustalić jakąś wartość napięcia na danym złączu, a tym samym sterować prędkością obrotową silnika albo jasnością diody.

Mikrokontroler zainstalowany na płytce Arduino jest dwudziestoosmiostykowym chipem umieszczonym w gnieździe znajdującym się na środku płytki. Ten pojedynczy chip posiada wbudowane układy procesora oraz pamięci. Wbudowano w nim również całą elektronikę odpowiedzialną za styki wejścia i wyjścia. Firma Atmel — jeden z największych producentów mikrokontrolerów — jest odpowiedzialna za produkcję tego układu. Każdy producent mikrokontrolerów produkuje różne układy zgrupowane w rodziny. Nie wszystkie mikrokontrolery są tworzone z myślą o elektronikach hobbystach. Jesteśmy tylko niewielką częścią tego ogromnego rynku. Mikrokontrolery są tworzone w celu stosowania ich w takich produktach konsumpcyjnych jak samochody, pralki, odtwarzacze DVD, zabawki, a nawet odświeżacze powietrza.

Dużą zaletą Arduino jest to, że ogranicza on tę zatrważającą liczbę kontrolerów możliwych do wyboru do jednego wystandaryzowanego układu. Później dowiesz się, że zdanie to nie jest do końca prawdziwe, ale na chwilę obecną możemy je za takie uznać.

Oznacza to, że rozpoczynając pracę nad nowym projektem, nie musisz zastanawiać się nad zaletami i wadami mikrokontrolerów różnych typów.

## Płyty rozwojowe

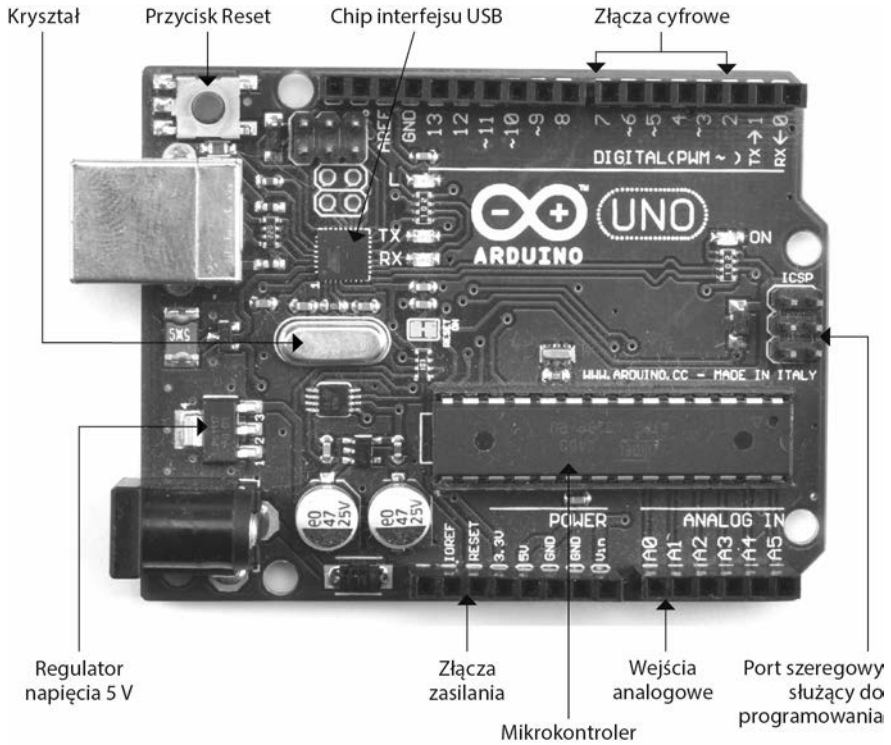
Powiedzieliśmy, że mikrokontroler to tak naprawdę tylko układ scalony. Pojedynczy chip nie jest jednakże w stanie działać bez elektroniki odpowiednio go zasilającej (mikrokontrolery są wrażliwe na problemy związane z zasilaniem) oraz bez elektroniki pozwalającej na komunikację z komputerem programującym mikrokontroler.

Wszystkie te funkcje zapewniają płyty rozwojowe. Płyta Arduino Uno jest tak naprawdę płytą rozwojową mikrokontrolera, której projekt jest niezależny i otwarty. Oznacza to, że pliki projektowe płytki drukowanej układu, a także jej schematy są ogólnodostępne i każdy może na ich podstawie tworzyć i sprzedawać swoje własne płytki Arduino.

Każdy producent mikrokontrolerów — w tym również Atmel produkujący mikrokontroler ATmega328 stosowany w płytach Arduino — dostarcza własne płyty rozwojowe i oprogramowanie służące do programowania układu. Co prawda rzeczy te zwykle nie są drogie, jednakże ich docelowym odbiorcą są inżynierowie, a nie elektronicy hobbysci. Takie płyty i oprogramowanie są trudniejsze w użyciu i wymagają dłuższej nauki obsługi przed uzyskaniem jakichkolwiek użytecznych rezultatów.

# Płyta Arduino

Na rysunku 1.1 przedstawiono płytę Arduino Uno. Przyjrzyjmy się elementom na niej umieszczonym.



Rysunek 1.1. Płyta Arduino Uno

## Zasilanie

Popatrz na rysunek 1.1. Bezpośrednio pod złączem USB znajduje się regulator napięcia 5 V. Układ ten generuje stałe napięcie 5 V niezależnie od podanego mu napięcia (w zakresie od 7 V do 12 V).

Układ regulatora napięcia jest dość duży jak na element montowany powierzchniowo. Układ ten ma taki rozmiar, ponieważ ułatwia to rozproszenie ciepła powstającego podczas regulacji napięcia przy dużych prądach. Jest to przydatne podczas zasilania urządzeń zewnętrznych.

Płytę Arduino można zasiląć za pośrednictwem gniazda zasilającego przy użyciu ogniw lub zasilacza. Dodatkowo można w tym celu wykorzystać gniazdo USB używane do jej programowania.

## Złącza zasilania

Następnie przyjrzyjmy się złączom zasilania znajdującym się na dole rysunku 1.1. Z rysunku możesz odczytać nazwy złączy. Pierwsze złącze jest oznaczone jako „Reset” — ma ono taką samą funkcję jak przycisk *Reset*. Podobnie jak ponowne uruchomienie komputera, skorzystanie z przycisku *Reset* powoduje rozpoczęcie wykonywania programu przez mikrokontroler od początku. Aby zresetować mikrokontroler, należy połączyć na chwilę pin Reset z pinem masy.

Pozostałe piny w tej sekcji dostarczają, zgodnie z oznaczeniami, prąd o odpowiednim napięciu (3,3 V, 5 V, GND i 9 V). Złącze GND, zwane również masą, dostarcza napięcie 0 V. Jest to napięcie, do którego odnoszą się wszystkie pozostałe napięcia na płytce układu.

## Wejścia analogowe

Sześć złączy opisanych jako *Analog In* o oznaczeniach od A0 do A5 można stosować do pomiaru przyłożonego do nich napięcia. Wynik pomiaru może zostać wykorzystany w szkicu. Zauważ, że mierzone jest napięcie, a nie prąd. Pomiedzy tymi złączami a masą mogą płynąć tylko niewielkie prądy. Dzieje się tak, ponieważ złącza te posiadają duży opór wewnętrzny.

Złącza te są oznaczone jako złącza analogowe, jednakże są one po prostu złączami działającymi domyślnie w trybie analogowym. Złącza te mogą również pracować jako cyfrowe wejścia lub wyjścia.

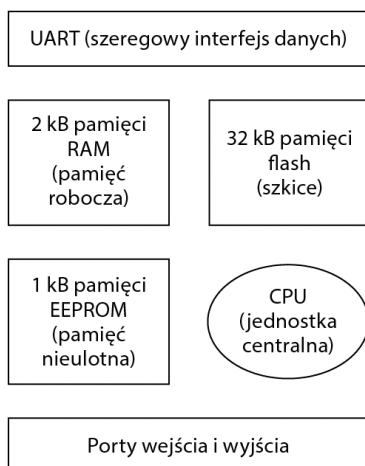
## Złącza cyfrowe

Teraz zajmiemy się omówieniem złączy pokazanych w prawym górnym rogu rysunku 1.1. Znajdują się tam piny o numerach od 0 do 13. Złącza te mogą służyć zarówno jako wejścia, jak i wyjścia. Działając jako wyjścia, zachowują się podobnie do opisanych wcześniej źródeł napięcia; jedyną różnicą jest to, że podają napięcie 5 V i można je włączać lub wyłączać z poziomu szkicu. Złącza te po wyłączeniu będą mieć potencjał 0 V. Stosując te złącza jako źródła prądowe, należy uważać na to, aby ich zbyt nie obciążać. Pierwsze dwa złącza (oznaczone 0 i 1) posiadają również oznaczenia RX (odbiór) i TX (transmisja). Złącza te są stosowane do transmisji danych. Są one pośrednio połączone ze złączem USB służącym do komunikacji z komputerem.

Omawiane złącza mogą zapewnić prąd o maksymalnym natężeniu 40 mA (miliamperów) przy napięciu 5 V. Jest to wystarczająca wartość do zasilania standardowej diody LED, jednakże jest to zbyt mały prąd, by bezpośrednio zasilac silnik elektryczny.

## Mikrokontroler

Opisując kolejne elementy znajdujące się na płytce Arduino, dochodzimy do samego chipu mikrokontrolera. Jest to prostokątny układ scalony posiadający 28 styków. Układ ten znajduje się w dwurzędowym podłużnym gnieździe. Można go z łatwością wymienić. Mikrokontroler umieszczony na płytce Arduino Uno to ATmega328. Na rysunku 1.2 przedstawiono diagram blokowy ilustrujący budowę tego układu.



Rysunek 1.2. Schemat blokowy układu ATmega328

Sercem, a właściwie mózgiem tego układu jest jednostka centralna (CPU). Kontroluje ona wszystko to, co dzieje się w układzie. Jednostka ta pobiera i wykonuje instrukcje zawarte w pamięci flash. Takie operacje mogą wymagać pobrania danych z pamięci roboczej (RAM), modyfikacji tych danych, a następnie zapisania ich z powrotem w pamięci roboczej. Instrukcje mogą również włączać lub wyłączać napięcie podawane na złącza cyfrowe.

Pamięć EEPROM jest nieulotna, podobnie jak pamięć flash. A więc po wyłączeniu zasilania urządzenia dane w niej zapisane nie zostaną utracone. Pamięć flash ma przechowywać instrukcje programu (szkice), a pamięć EEPROM ma przechowywać dane, których nie chcesz utracić w wyniku wciśnięcia przycisku *Reset* lub wyłączenia zasilania.

## Pozostałe podzespoły

Nad procesorem znajduje się mały, srebrny, prostokątny element. Jest to kwarcowy generator drgań. Generuje on 16 milionów impulsów na sekundę. Z każdym impulsem mikrokontroler wykonuje jedną operację matematyczną (np. dodawania lub odejmowania).

W lewym górnym rogu płytki znajduje się przycisk *Reset*. Wciśnięcie tego przycisku spowoduje wysłanie sygnału logicznego sprawiającego, że mikrokontroler wyczyści pamięć operacyjną i uruchomi od początku wykonywany program. Warto zwrócić uwagę na to, że program jest przechowywany w pamięci nieulotnej — zawartość tej pamięci nie jest kasowana w wyniku wciśnięcia przycisku *Reset* lub odłączenia zasilania.

Po prawej stronie płytki znajduje się port szeregowy służący do programowania. Pozwala on na programowanie Arduino bez potrzeby korzystania z interfejsu USB. Nie będziemy poruszać kwestii związanych z możliwością programowania Arduino poprzez port szeregowy, ponieważ Arduino posiada złącze USB i odpowiednie oprogramowanie, co pozwala na wygodną pracę.

W lewym górnym rogu płytki (obok gniazda USB) znajduje się chip interfejsu USB. Układ ten przetwarza standardowy sygnał USB na sygnał o poziomie akceptowanym przez płytkę Arduino.

---

## Początki Arduino

Arduino powstało jako pomoc mająca służyć studentom. Następnie (w 2005 r.) Massimo Banz i David Cuartielles zaczęli komercyjnie rozwijać ten projekt. Z powodu łatwej obsługi i wytrzymałości Arduino jest szczególnie popularne wśród konstruktorów, studentów i artystów.

Kolejnym czynnikiem mającym wpływ na sukces tego mikrokontrolera jest to, że jego dokumentacja jest ogólnodostępna na zasadach licencji Creative Commons. Pozwoliło to na powstanie wielu tańszych odpowiedników płyty. Nazwa Arduino jest chroniona, a więc takie klony mają często nazwy utrzymane w stylistyce „\*duino” (np. Boarduino, Seeeduino i Freeduino). W 2014 r. doszło do podziału pomiędzy twórcami platformy Arduino a głównym

producentem płytek. Jednym ze skutków tego podziału jest to, że poza USA płytka Arduino Uno jest sprzedawana pod nazwą Genuino Uno. Wiele sklepów zajmuje się dystrybucją tylko oficjalnych płytek, których jakość wykonania nie budzi żadnych zastrzeżeń, a dodatkowo są one sprzedawane w eleganckich opakowaniach.

Kolejnym czynnikiem mającym wpływ na sukces Arduino jest fakt, że Arduino nie jest ograniczone tylko do płytki mikrokontrolera. Istnieje wiele płyt stykowych kompatybilnych z Arduino. Płytki takie są wpinane bezpośrednio na płytę Arduino. Dzięki temu, że istnieje wiele takich płytek stykowych (zwanых również potocznie „shieldami”), możliwe jest uniknięcie korzystania z lutownicy. Wystarczy tylko wpiąć w siebie płytki. Poniżej wymieniono najpopularniejsze płytki stykowe:

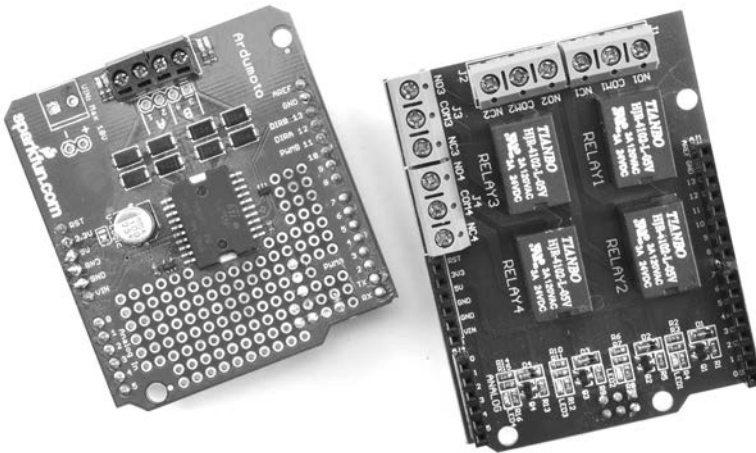
- Ethernet — zapewnia obsługę sieci.
- Motor — pozwala obsługiwać silniki elektryczne.
- Host USB — pozwala na obsługę urządzeń wyposażonych w interfejs USB.
- Moduł przekaźników — pozwala na obsługę przekaźników za pośrednictwem Arduino.

Na rysunku 1.3 pokazano płytkę sterownika silników (strona lewa) i płytkę przekaźników (strona prawa).

---

## Rodzina płyt Arduino

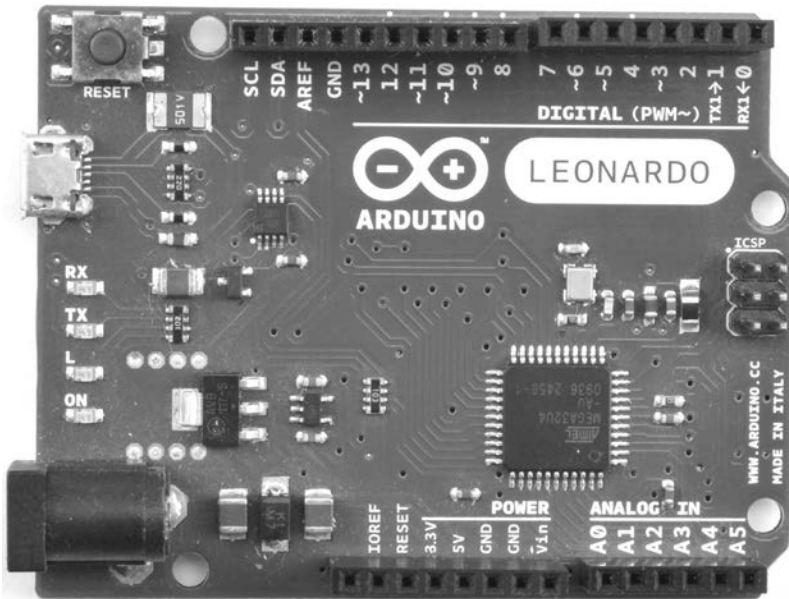
Podstawowa wiedza na temat różnych płyt Arduino może okazać się przydatna. Płyty Arduino Uno będziemy używać jako standardowego urządzenia — będzie to model Arduino Uno R3 (wersja 3). Jest to najczęściej stosowana płyta Arduino. Wszystkie płyty Arduino są programowane w tym samym języku i mają podobną konfigurację wejść i wyjść, a więc ewentualne zastosowanie innej płyty nie stanowi większego problemu.



Rysunek 1.3. Płytkę sterownika silników i płytkę przekaźników

## Uno, Duemilanove i Diecimila

Arduino Uno jest jedną z wersji najbardziej popularnej serii płyt Arduino. Do serii tych płyt zaliczają się również włoskie konstrukcje Diecimila (z włoskiego 10 000) i Duemilanove (z włoskiego 2009). Na rysunku 1.4 pokazano płytkę Arduino Leonardo. Jak już pewnie zauważyłeś, Arduino jest włoskim wynalazkiem.



Rysunek 1.4. Płytkę Arduino Leonardo

Płytką Arduino Leonardo (patrz rysunek 1.4) jest kolejną popularną wersją Arduino. W większości zastosowań może ona zastąpić płytkę Arduino Uno. Jest nieco tańsza od modelu Uno. Płytki Uno i Leonardo są wyposażone w te same złącza, ale chip modelu Leonardo jest przylutowany do powierzchni płytki, a więc nie można go z niej wymontować (można to zrobić w przypadku płytki Uno). Niższa cena płytki Leonardo wynika z tego, że znajdujący się na niej chip jest zintegrowany ze sterownikiem interfejsu USB i w przeciwieństwie do płytki Uno na płytce Leonardo nie trzeba umieszczać dodatkowego układu zarządzającego interfejsem USB.

## Mega i Due

Arduino Mega to prawdziwe Ferrari wśród płyt Arduino. Płyta ta została wyposażona w wiele dodatkowych złączy wejścia i wyjścia. Złącza te zostały jednakże dodane na skraju płytki, a więc Arduino Mega jest w pełni kompatybilne pod względem układu złączy z Arduino Uno i Leonardo, w związku z czym możliwe jest stosowanie płytek stykowych zgodnych z Arduino Uno.

Arduino Mega wyposażono w procesor obsługujący dodatkowe złącza — ATmega1280. Mikrokontroler ten jest montowany powierzchniowo, a więc jest przymocowany do płyty na stałe. W przypadku uszkodzenia procesora jego wymiana nie jest możliwa. Omówiona wcześniej płytką Arduino Uno umożliwia wymianę uszkodzonego mikroukładu.

Dodatkowe złącza umieszczono na skraju płytki. Dodatkowe funkcje tej wersji Arduino to:

- 54 piny wejścia i wyjścia;
- 128 kB pamięci flash służącej do przechowywania szkiców i stałych danych (Arduino Uno posiadało tylko 32 kB tej pamięci);
- 8 kB pamięci RAM i 4 kB pamięci EEPROM.

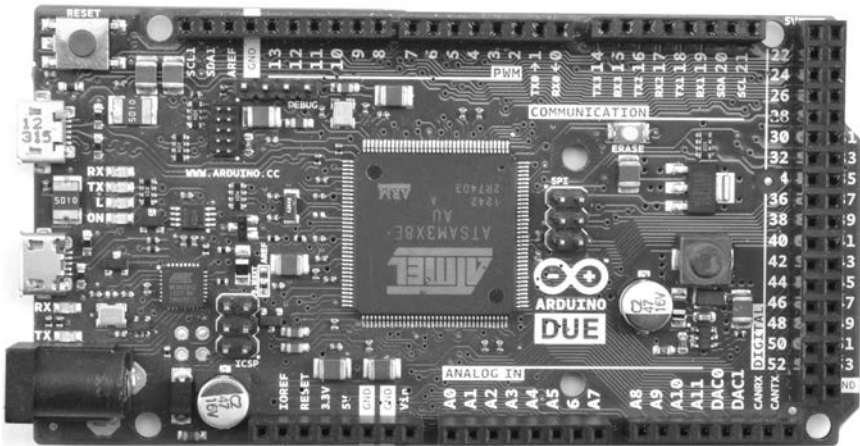
Płytką Arduino Due (patrz rysunek 1.5) jest wyposażona w te same złącza co płytką Mega i ma identyczne wymiary, ale zainstalowano na niej 32-bitowy procesor ARM taktowany zegarem o częstotliwości 84 MHz. Ponadto płytką ta jest zasilana prądem o napięciu 3,3 V, a dla większości płytek Arduino napięcie zasilania to 5 V. W związku z tym część płytek stykowych może z nią nie współpracować.

## Micro i Small

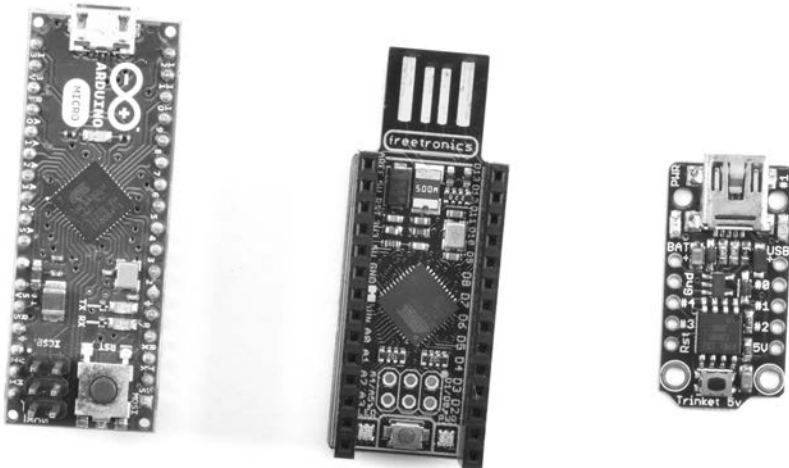
Płytką Uno jest zbyt duża, aby można ją było stosować w niektórych projektach. Wówczas można skorzystać z wielu mniejszych klonów Arduino i płytek kompatybilnych z platformą Arduino. Na rysunku 1.6 pokazano przykłady tego typu płytek.

Na płytce Arduino Micro zamontowano ten sam mikrokontroler co na płytce Leonardo, ale płytką ma mniejsze wymiary. Alternatywą dla tej płytki są produkty oferowane przez niezależnych producentów, takie jak LeoStick i Trinket.

Wadą takich małych płytek jest to, że z powodu niewielkich wymiarów nie da się na nie założyć płytek rozszerzeń kompatybilnych z Arduino Uno.



Rysunek 1.5. Płytki Arduino Due



Rysunek 1.6. Płytki Arduino Micro (po lewej), Freetronics LeoStick (na środku) i Adafruit Trinket (po prawej)

## Yun

Płytki Arduino Yun (patrz rysunek 1.7) jest połączeniem modelu Arduino Leonardo z miniaturowym modułem wi-fi pracującym pod kontrolą systemu Linux. Urządzenie to jest przeznaczone do projektów wymagających połączenia z internetem. Chip Arduino komunikuje się z systemem Linux modułu sieci bezprzewodowej za pośrednictwem programu określanego mianem mostka. Płytki Yun jest programowana standardowo przy użyciu środowiska programistycznego Arduino, ale po skonfigurowaniu połączenia z siecią lokalną można ją również programować bezprzewodowo.



Rysunek 1.7. Płytko Arduino Yun

## Lilypad

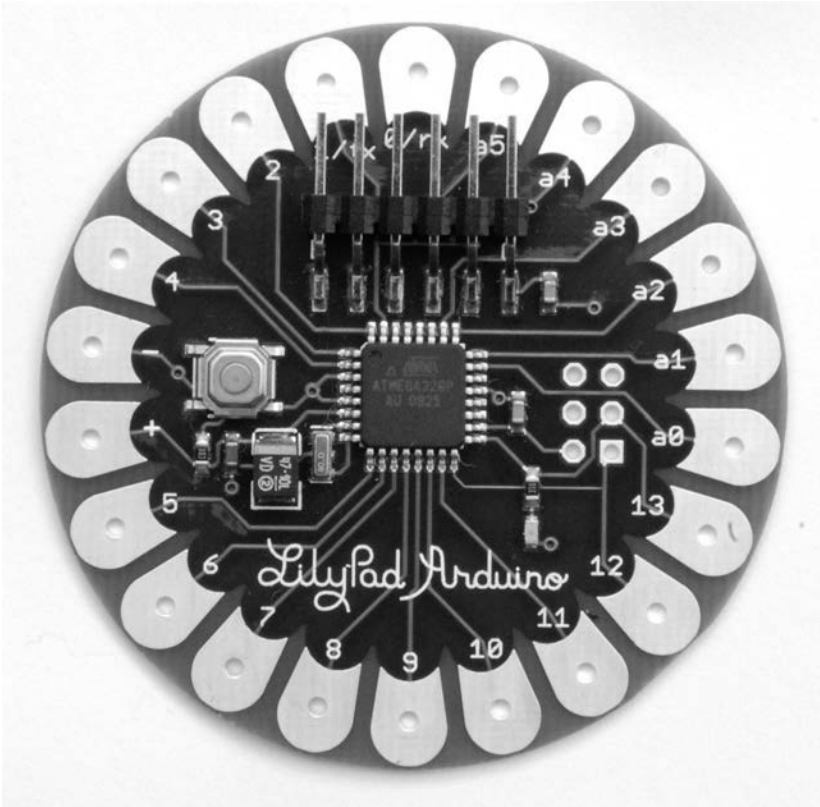
Lilypad (patrz rysunek 1.8) jest niewielką, cienką płytką Arduino, którą można przyczepić do ubrania. Płytkę tę można stosować w przypadku tworzenia układów elektronicznych noszonych pod ubraniem.

Lilypad nie posiada złącza USB. Do zaprogramowania tej płytki niezbędny jest zewnętrzny programator. Projekt tej płytki jest wyjątkowo estetyczny.

Firma Adafruit oferuje płytkę Flora, która wygląda podobnie do modelu Lilypad.

## Inne „oficjalne” płytki

Przedstawione wcześniej płytki Arduino są najbardziej przydatne i najczęściej stosowane. Rodzina płytek Arduino ciągle się jednak powiększa. Aby uzyskać najświeższe informacje dotyczące płytek z rodziny Arduino, zajrzyj na oficjalną witrynę internetową Arduino pod adresem: <http://www.arduino.cc/en/Main/Hardware>.



Rysunek 1.8. Płytko Arduino Lilypad

## Inne klony i odmiany Arduino

Nieoficjalne płytki można podzielić na dwie kategorie. Niektóre płytki są po prostu tańszymi, standardowymi kopiami Arduino. Do tej kategorii można zaliczyć następujące płytki:

- Sparkfun RedBoard,
- Adafruit Metro,
- Olimexino.

Niektóre płytki kompatybilne z Arduino mają na celu usprawnienie działania Arduino lub rozszerzenie jego możliwości. Ciągłe powstają nowe odmiany tego typu płytek. Nie-możliwe jest wyliczenie ich wszystkich. Poniższa lista zawiera bardziej interesujące i częściej spotykane odmiany Arduino:

- Płytko Node MCU zbudowana na bazie chipu ESP8266 WiFi System. Model ten doskonale sprawdza się w projektach wymagających połączenia z siecią wi-fi. Więcej informacji na ten temat znajdziesz w rozdziale 10.

- Adafruit Trinket — bardzo mały klon Arduino.
- Freetronics EtherTen — Arduino zintegrowane z kontrolerem sieci Ethernet.
- Particle Photon — tania płytką z obsługą sieci wi-fi. Programuje się ją za pośrednictwem internetu przy użyciu języka C platformy Arduino za pomocą specjalnego sieciowego środowiska programistycznego zastępującego standardowe zintegrowane środowisko programistyczne Arduino.

---

## Podsumowanie

Gdy już poznałeś Arduino od strony sprzętowej, nastał czas, aby zacząć tworzyć oprogramowanie na to urządzenie.

# Skorowidz

## A

alfabet Morse'a, 79  
alfanumeryczne wyświetlacze LCD, 124  
aplikacja Arduino, 13, 17, 35

- Diecimila, 23
- Due, 24
- Duemilanove, 23
- Lilypad, 26
- Mega, 24
- Micro, 24
- Small, 24
- Uno, 23
- Yun, 25

Arduino Uno

- usługa IFTTT, 152

Arduino Yun, 156  
argument funkcji, 40  
arytmetyka, 49

## B

białe znaki, 69  
biblioteka

- AVR EEPROM, 117
- Bounce, 97
- Flasher, 162
- MCU, 145
- PROGMEM, 114
- standardowa, 103

biblioteki

- tworzenie, 158
- wbudowane, 158

bit, 106  
blok kodu, 44

## D

dokument HTML, 135  
dyrektywa PROGMEM, 121

## E

EEPROM, 116

- przechowywanie wartości zmiennej, 117
- wartości typu float, 118
- wymazywanie zawartości pamięci, 120

## F

formularz akcji, 151  
funkcja, 59

- EEPROM.write, 116
- flashDotOrDash, 84
- flashSequence, 83
- loop, 44, 55, 81
- noInterrupts, 111
- setup, 44, 48, 81

funkcje

- biblioteki wyświetlacza LCD, 127
- matematyczne, 105
- wejścia i wyjścia, 108
- złączy płytki wyświetlacza, 126

**G**

generowanie tonów, 108

**H**

hermetyzacja, 62, 157

HTML, 134

HTTP, 134

**I**

importowanie biblioteki, 162

instalacja

oprogramowania, 30

biblioteki Bounce, 98

interfejs USB, 124

internet rzeczy, 133, 155

**J**

język

C, 39

C++, 157

HTML, 134

**K**

klasy, 157

klony, 27

komentarze, 70

kompilacja, 41

kompresja, 121

zakresu, 121

komunikacja z serwerami sieciowymi, 134

konstruktor, 159

**L**

LCD, 124

liczby losowe, 103

literały łańcuchowe, 77

**Ł**

ładowanie pierwszego szkicu, 30

łańcuch, 77

**M**

mechanizmy obiektowe, 157

metody, 157

mikrokontrolery, 17, 20

monitor portu szeregowego, 75, 89

**N**

nawiasy klamrowe, 69

**O**

odmiany, 27

OLED, 128

opcje projektów internetu rzeczy, 155

operator

iloczynu logicznego, 66

sumy logicznej, 66

operatory porównywania, 53

otwieranie szkicu, 163

**P**

pamięć

EEPROM, 116

flash, 114

parametry, 60

Particle Photon, 156

pętla

for, 55

while, 56

pierwszy szkic, 30

plik

implementacji, 160

nagłówkowy, 158

plytka

Arduino, 19

Node MCU, 147

Node MCU ESP8266, 154

podłączanie wyświetlacza OLED, 128

podzespoły, 21

polecenie

for, 53

if, 52

while, 56

pomiar napięcia, 101  
 programowanie, 39  
   aplikacji internetu rzeczy, 133  
   intencyjne, 81  
 protokół  
   HTTP, 134  
 przechowywanie  
   łańcucha, 119  
   wartości zmiennej, 117, 118  
 przełącznik, 96  
 przerwania, 110

## R

regulacja napięcia, 100  
 rejestr przesuwny, 109  
 rezystor podwyższający, 91, 94  
 rzutowanie, 121

## S

sieci  
   serwer, 134  
     Arduino, 136  
     Node MCU, 143  
     Web, 136  
   sterowanie pracą Arduino, 138, 147  
   usługa IFTTT, 152, 154  
   usługi, 150  
 słowa kluczowe, 161  
 stałe, 56, 113  
 standardowa biblioteka Arduino, 103  
 sterowanie pracą Arduino, 138, 147  
 stuki, 94  
 styl zapisu kodu, 68  
 suma logiczna, 66  
 sygnał  
   o fali kwadratowej, 109  
   przerwania, 111  
 symbol negacji, 67  
 system dwójkowy, 106  
 szkic, 129

## T

tablica wyświetlająca komunikaty, 124  
 tablice, 73  
   łańcuchów, 77

tłumacz alfabetu Morse'a, 79  
 tworzenie bibliotek, 158  
 typ danych, 67  
   boolean, 66  
   float, 65

## U

USB, 124  
 usługa IFTTT, 152, 154  
 usługi sieciowe, 150  
 usuwanie stuków, 94

## W

wejścia, 87  
   analogowe, 20, 101  
   cyfrowe, 90  
 wewnętrzny rezystor podwyższający, 94  
 wskaźniki, 76  
 wyjścia, 87  
   analogowe, 99  
   cyfrowe, 87  
 wyświetlacze, 123  
   LCD, 124  
   OLED, 128


## Z

zapis kodu  
   białe znaki, 69  
   komentarze, 70  
   nawiasy klamrowe, 69  
   wcięcia, 68  
 zapisywanie danych, 113  
 zasilanie, 19, 29  
 złącza  
   cyfrowe, 20  
   zasilania, 20  
 zmienne, 46  
   globalne, 61, 81  
   lokalne, 61  
   łańcuchowe, 78  
   numeryczne, 49  
   składowe, 157  
   statyczne, 61  
 znaczniki, 135  
 zwracanie wartości, 64



# 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**

# Arduino — fascynujący mikrokontroler!

Arduino to prosty komputer, wykorzystywany w najróżniejszych domowych gadżetach. Potrafi komunikować się z innymi urządzeniami, posiada także złącza umożliwiające współpracę z silnikami, przekaźnikami, fotodiodami, diodami laserowymi, głośnikami, mikrofonami itp. Oprogramowanie dla Arduino jest darmowe, łatwe w użyciu i dostępne na platformy Windows, macOS i Linux. Przy tym Arduino jest bardzo prosty w rozbudowie o kolejne elementy. Świetnie nadaje się do budowy przeróżnych robotów, sterowników, czujników czy interfejsów do komunikacji z innymi urządzeniami. Dodatkowo Arduino jest ciekawym i bardzo przyjemnym sposobem na nauczenie się podstaw elektroniki, informatyki bądź programowania.

Oto zaktualizowane i uzupełnione wydanie popularnego podręcznika programowania dla osób, które chcą zacząć przygodę z Arduino. Jest odpowiednie także dla początkujących. Wystarczy zestaw Arduino i dobre chęci, aby wykonać urządzenie do sterowania oświetleniem, zarządzania energią wytwarzaną przez panele słoneczne, tablicę do wyświetlania komunikatów i wiele innych ciekawych projektów. Dzięki przejrzystym wskazówkom i wyjaśnieniom unikniesz podstawowych błędów w kodzie. Gruntowna znajomość podstaw języka C będzie dodatkową korzyścią, którą zyskasz dzięki tej książce.

## W tej książce:

- budowa płytki Arduino i przygotowanie środowiska pracy
- programowanie Arduino w języku C
- odbieranie i wysyłanie danych przez interfejs wejścia-wyjścia
- wskazówki dotyczące najczęstszych błędów

**Dr Simon Monk** jest cybernetykiem, informatykiem, autorem książek i konstruktorem. Przez kilka lat był pracownikiem akademickim, później zajął się rozwijaniem firmy programistycznej Momote Ltd. Napisał około dwudziestu książek na temat swoich największych pasji: elektroniki i majsterkowania. Zachęcony ich popularnością, w 2015 roku wraz z żoną Lindą założył spółkę MonkMakes Ltd., w której zajmuje się projektowaniem nowych produktów i... parzeniem świetnej herbaty dla swoich gości.

**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 978-83-289-4054-3



Cena: 54,90 zł