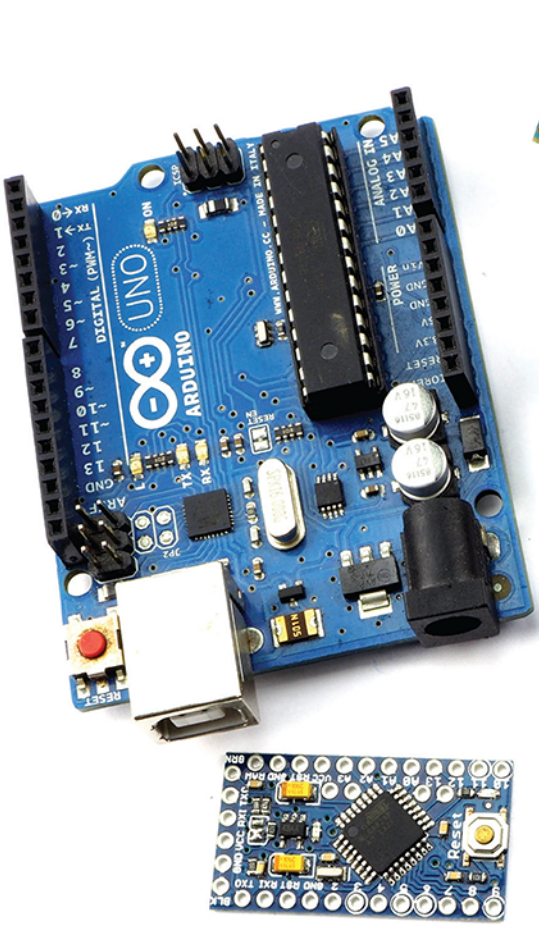
Wydanie II

Arduino® dla początkujących

Kolejny krok



Simon Monk

TAB Make Great Stuff!
Mc Graw Hill Education
Helion

Tytuł oryginału: Programming Arduino Next Steps: Going Further with Sketches, Second Edition

Tłumaczenie: Anna Mizerska
z wykorzystaniem fragmentów Arduino dla początkujących. Kolejny krok
w przekładzie Konrada Matuka

ISBN: 978-83-283-7548-2

Helion SA
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)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/arpok2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

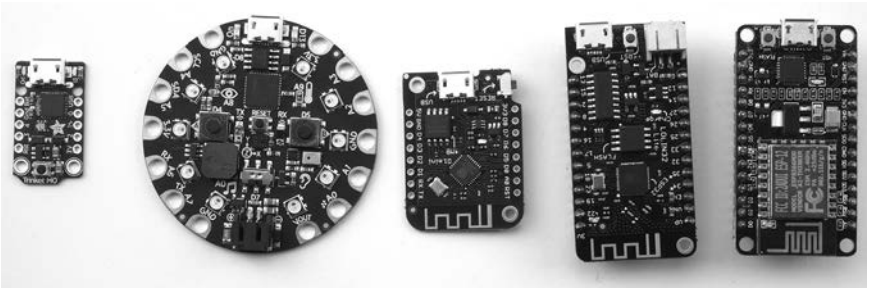
Spis treści

Rozdział 3.

Kiedy Arduino to nie Arduino?

Zespół Arduino wykonał świetną pracę, przez co środowisko programistyczne Arduino jest modułowe i można łatwo dodawać do niego nowe płytki. Co za tym idzie, możliwe jest dodanie nowego oprogramowania wspierającego płytki kompatybilne z Arduino i programowanie ich za pomocą IDE z użyciem tych samych poleceń.

Możesz programować nie tylko gotowe płytki, które widać na rysunku 3.1, lecz możesz również użyć Arduino IDE do programowania samych mikrokontrolerów (innych niż te zastosowane w płytkach Arduino).



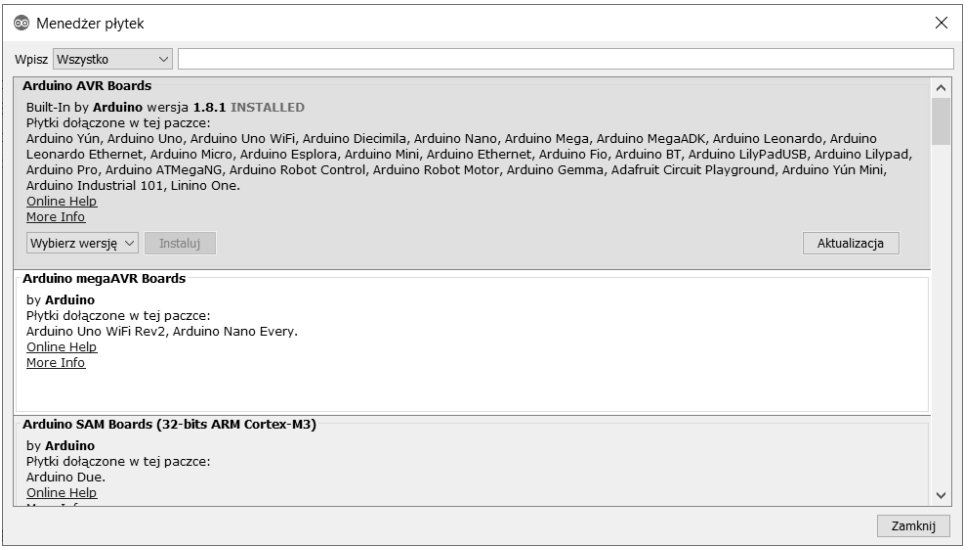
Rysunek 3.1. *Od lewej do prawej: Trinket m0, Circuit Playground Express, Wemos D1 Mini, LOLIN32, NodeMCU*

Rozszerzalna architektura Arduino IDE

Arduino IDE, a tak naprawdę cały projekt Arduino, ma otwarte źródło (licencja *open source*). Oznacza to, że możesz przeanalizować i skopiować na własne potrzeby cały kod Arduino IDE i kod, który uruchamiany jest na płytce Arduino. Mimo że formaliści kłóciliby się ze stwierdzeniem, że sprzęt Arduino całkowicie spełnia założenia licencji *open source*, faktem jest, że wszystkie schematy i informacje potrzebne do zrobienia własnego Arduino są ogólnie dostępne.

Ponadto Arduino IDE jest elastyczne i nie ma tak naprawdę potrzeby zmiany jego kodu źródłowego. Jednym z kluczowych powodów takiego rozwiązania jest możliwość łatwego i szybkiego podłączenia innych platform lub jednostek bazowych do Arduino IDE.

Najprostszym sposobem na dodanie sterownika innej płytki jest skorzystanie z menedżera płytek (rysunek 3.2), który znajduje się w *Narzędzia/Płytka/Menedżer płytek*.



Rysunek 3.2. *Menedżer płytek*

Domyślnie nie znajdziesz tam wszystkich płytek, a szczególnie tych mniej popularnych, pochodzących z mniej oficjalnych źródeł. Możesz je jednak dodać poprzez wprowadzenie adresu URL w polu *Dodatkowe adresy URL do menedżera płytek* w oknie preferencji, które otworzysz z poziomu menu *Plik* (rysunek 3.3).

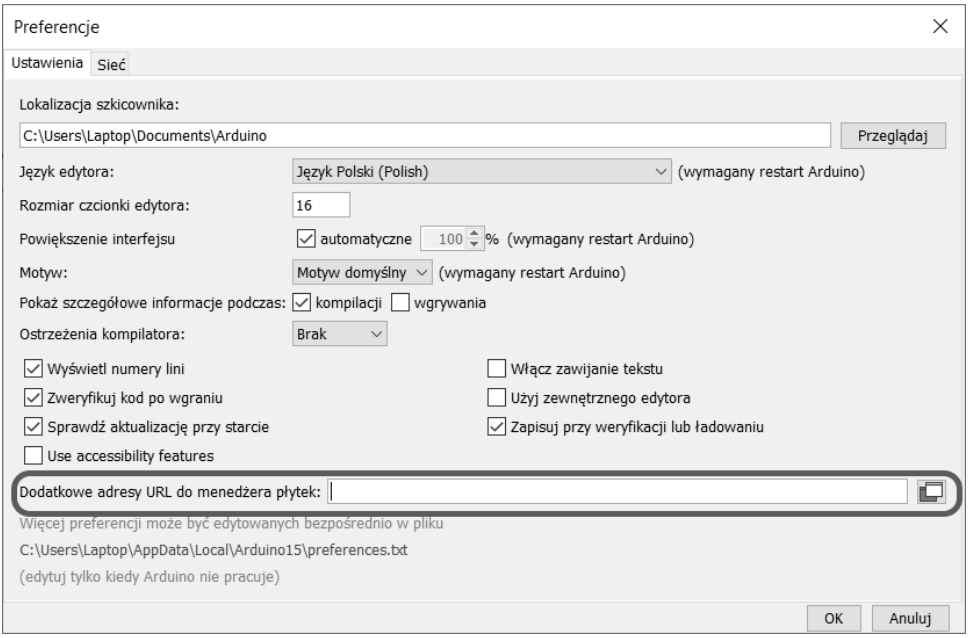
W kolejnym podrozdziale dodamy wsparcie dla popularnej płytki Adafruit Circuit Playground.

Adafruit Circuit Playground Express

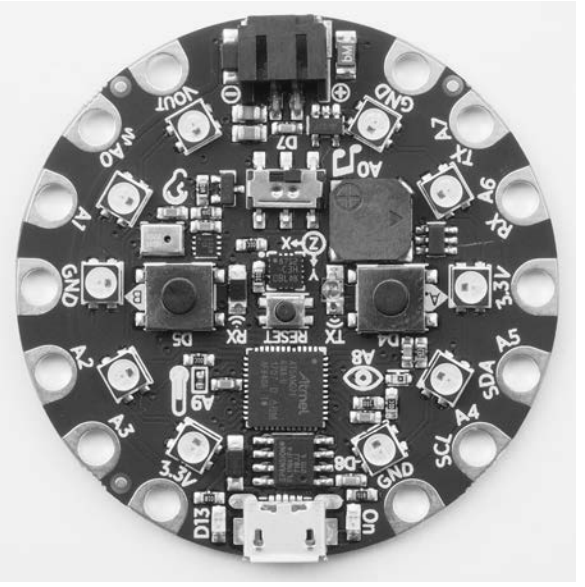
Płytka Adafruit Circuit Playground (rysunek 3.4) może być programowana za pomocą wielu różnych języków i środowisk, w tym Arduino. Po części płytka została zaprojektowana do celów edukacyjnych, więc ma już wbudowanych kilka urządzeń peryferyjnych, między innymi: 10 diod LED RGB „Neopixel”, brzęczyk, dwa przyciski, akcelerometr i mikrofon.

W przeciwieństwie do Arduino ta płytka nie ma procesora z rodziny Mega i jej mikrokontroler oparty jest na architekturze Arm. Teraz użyjemy tej płytki, aby pokazać zastosowanie menedżera płytek.

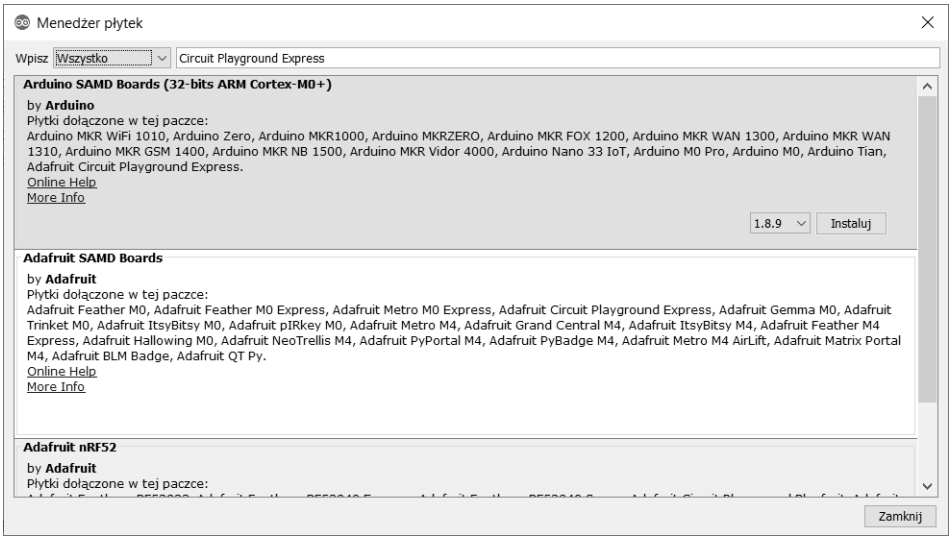
Otwórz menedżera płytek, w polu wyszukiwania wpisz **`Circuit Playground Express`** (rysunek 3.5) i wybierz pierwszą pozycję z listy. Zapewne zauważysz, że ten sterownik daje wsparcie dla dużej liczby płytek, w tym Circuit Playground Express. Naciśnij przycisk *Instaluj*, który znajduje się obok informacji o wersji.



Rysunek 3.3. *Dodawanie adresów URL do menedżera płytek*

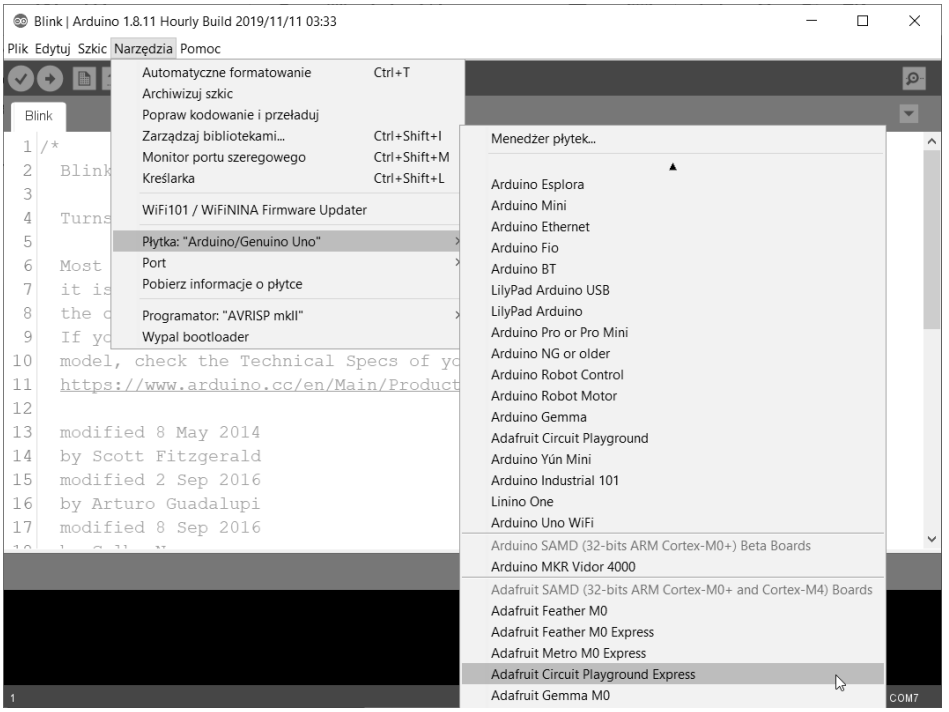


Rysunek 3.4. *Płytka Adafruit Circuit Playground*



Rysunek 3.5. *Dodawanie płytki za pomocą menedżera płytek*

Po zainstalowaniu sterownika na liście dostępnych płytek powinna pojawić się płytka Circuit Playground Express (rysunek 3.6).



Rysunek 3.6. *Circuit Playground Express na liście płytek*

Aby w pełni wykorzystać wbudowane urządzenia peryferyjne płytki Circuit Playground Express, dodatkowo musisz zainstalować odpowiednie biblioteki Arduino. W tym celu otwórz menedżera bibliotek i wyszukaj *Playground Express*. Przetestuj jeden z przykładowych szkiców

dołączonych do biblioteki, który znajdziesz w menu *Plik/Przykłady*, choćby szkic *HelloNeoPixels* z kategorii *Hello Circuit Playground*. Przejrzyj pozostałe przykładowe szkice biblioteki *Circuit Playground*, aby zobaczyć, co możesz jeszcze zrobić z płytką, programując ją tak jak standardową płytkę Arduino.

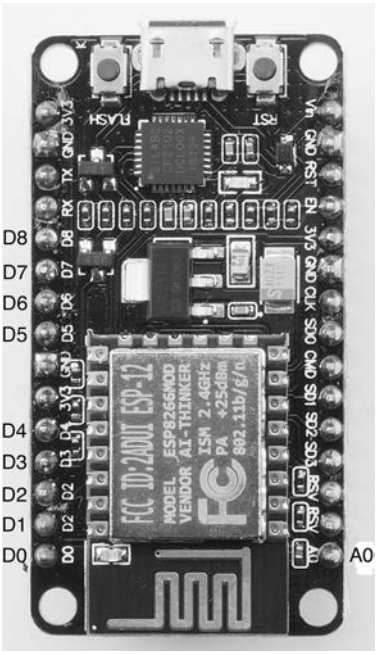
Uwaga: jeśli korzystasz z systemu Windows, to aby móc używać płytki, musisz zainstalować sterownik dla tego systemu (zobacz *https://learn.adafruit.com/introducing-circuit-playground/windows-driver-installation*). Użytkownicy systemów Linux i macOS nie muszą tego robić.

Wszystkie informacje na temat płytki Playground Express znajdziesz na stronie: *https://learn.adafruit.com/introducing-circuit-playground/*.

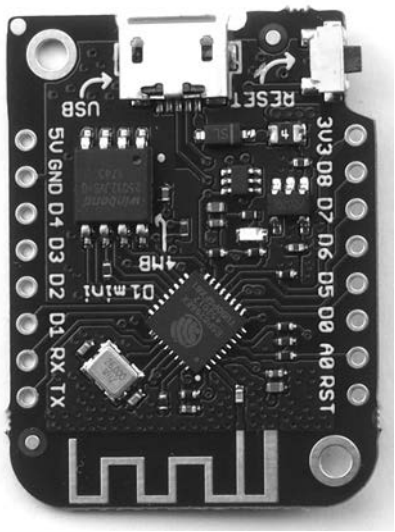
Firma Adafruit ma w swojej ofercie także duży wybór płytek kompatybilnych z Arduino. Odwiedź ich stronę i wyszukaj „feather", „trinket" i „flora", aby sprawdzić inne produkty tej firmy. Niektóre z nich mają specjalne funkcjonalności, na przykład wbudowaną antenę radiową, Wi-Fi czy ładowarkę baterii LiPo.

NodeMCU

Płytka NodeMCU i płytka z tej samej rodziny, Wemos D1 Mini (rysunki 3.7 i 3.8), to bardzo tanie płytki kompatybilne z Arduino z wbudowanym modułem Wi-Fi. Obie są oparte na module mikrokontrolera ESP8266 WiFi. Można go programować za pomocą wielu języków programowania, w tym Arduino.



Rysunek 3.7. *Płytka NodeMCU*



Rysunek 3.8. *Płytka Wemos D1 Mini*

Pomimo że na pierwszy rzut oka wydaje się, że płytka NodeMCU ma tyle samo pinów wejścia/wyjścia ogólnego przeznaczenia (GPIO) co Arduino Pro Mini, w rzeczywistości ma ich dużo mniej — tylko 1 wejście analogowe i 9 cyfrowych wejść/wyjść. Płytka Wemos D1 Mini korzysta z tego faktu i dzięki temu jest mniejsza.

Odwołując się do tych pinów, należy dodawać literę *A* lub *D*, na przykład A0, D4. To podejście różni się od standardowej płytki Arduino, gdzie litera *D* może być pominięta. Zauważ, że płytka NodeMCU, pokazana na rysunku 3.7, ma błąd w oznaczeniach pinów — D2 występuje dwukrotnie.

Jeśli używasz systemu Windows lub Linux, możliwe, że aby móc stosować te płytki, będziesz musiał zainstalować sterowniki dla CP2102 (dla NodeMCU) lub interfejsu USB CH340 (dla Wemos D1 Mini) (zobacz, odpowiednio, *www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers* lub *www.arduined.eu/ch340-windows-8-driver-download*).

Żeby dodać wsparcie dla tych płytek do Arduino IDE, otwórz menedżera płytek, wyszukaj *ESP8266* i zainstaluj *esp8266 by ESP8266 Community*. Jeśli wgrasz szkic o nazwie *03_01_node_mcu_blink*, zobaczysz, jak miga wbudowana dioda LED (która w przypadku NodeMCU zazwyczaj podłączona jest do pinu D0, ale czasami do pinu D1 lub D2, a na Wemos D1 Mino — do pinu D4). Wszystkie przytoczone w tej książce szkice znajdziesz na stronie *http://www.helion.pl/ksiazki/arpok2.htm*.

Należy pamiętać, że wejścia i wyjścia tych płytek działają pod napięciem 3,3 V, dlatego nie podłączaj urządzeń wyjścia 5 V do wejść 3,3 V, bo zniszczysz płytkę. Każde cyfrowe wyjście może dostarczyć 12 mA. Jest to mniej niż w przypadku Arduino Uno, które może dostarczyć 40 mA, ale jest to wciąż wystarczająca ilość prądu, aby dioda LED jasno świeciła.

W rozdziale 14. znajdziesz przykłady wykorzystania wbudowanej funkcji Wi-Fi tych płytek oraz płytki ESP32 z kolejnej części.

ESP32

Często ograniczona liczba pinów GPIO płytki NodeMCU i innych płytek kompatybilnych z Arduino opartych na ESP8266 nie stanowi problemu. Jednak czasami możesz potrzebować ich więcej lub będziesz chciał użyć Bluetootha. W takich przypadkach wybierz płytkę z ESP32.

Firmy SparkFun i Adafruit produkują wysokiej jakości płytki oparte na ESP32, dla których stworzona jest rzetelna dokumentacja (na przykład SparkFun ESP32 Thing i Adafruit Huzzah32 Feather Board). Obie płytki mają wbudowaną możliwość ładowania baterii LiPo, co sprawia, że są doskonałym wyborem dla projektów mobilnych. Jeśli chcesz wydać mniej, w internecie możesz znaleźć tańsze alternatywy, między innymi pokazaną na rysunku 3.1 płytkę LOLIN32.

Aby móc dodać płytki oparte na ESP32 z poziomu menedżera płytek, musisz najpierw dodać adres URL: *https://dl.espressif.com/dl/package_esp32_index.json* w oknie preferencji, w polu *Dodatkowe adresy URL do menedżera płytek* (zobacz rysunek 3.3). Dzięki temu w menedżerze płytek znajdziesz płytkę *esp32 by Espresif Systems*.

Więcej informacji na temat projektu ESP32 jest dostępnych na stronie: *https://github.com/espressif/arduino-esp32*. Znajdziesz tam również obecny status projektu, ponieważ podczas pisania tej książki niektóre funkcjonalności nie zostały jeszcze wprowadzone, na przykład wyjście PWM (`analogWrite`).

ESP32 jest godnym uwagi urządzeniem z podwójnym procesorem, dużą ilością pamięci i naprawdę niskim zużyciem mocy, dzięki czemu doskonale sprawdzi się w projektach związanych z internetem rzeczy. Do tego zagadnienia wrócimy w rozdziale 14.

Mikrokontrolery ATtiny

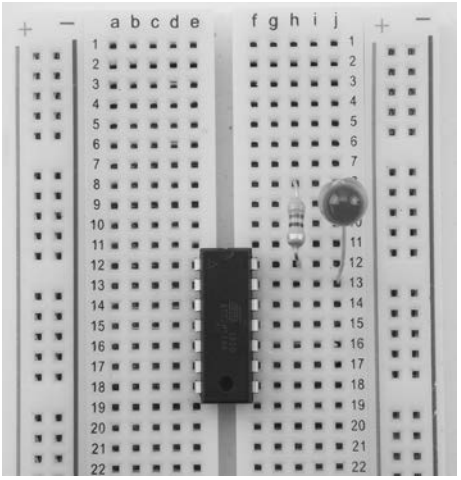
Niektóre projekty wymagają niewiele, jeśli chodzi o piny wejścia i wyjścia i zużywają zaledwie niewielki procent dostępnej 32-kilobajtowej pamięci flash mikrokontrolera ATmega328. W przypadku takich projektów możesz rozważyć zastosowanie płytki opartej na pokrewnym mikrokontrolerze z rodziny ATtiny. Te mikrokontrolery mają wiele wspólnych cech z ATmega, ale jak sama nazwa wskazuje (ang. *tiny* — malutki), ogólnie mają wszystkiego mniej, włączając w to piny i ilość pamięci każdego typu. Zaletami oczywiście są niska cena i mały rozmiar. ATtiny jest doskonałym kolejnym krokiem w przygodzie z Arduino Uno. Możesz zrezygnować z większości płytek Arduino i opierać swoje projekty tylko na jednym mikrokontrolerze.

ATtiny44

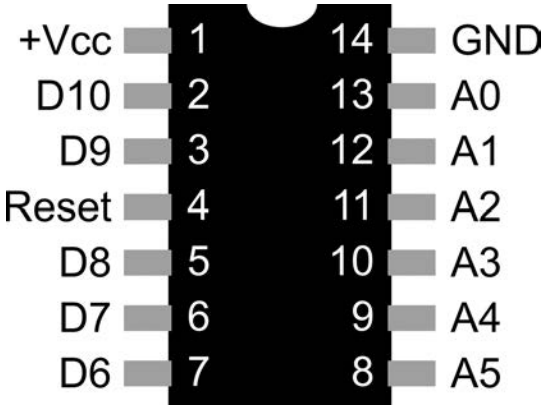
Dla przykładu przyjrzyjmy się, jak używa się mikrokontrolera ATtiny44. Ten chip plasuje się między pełnowymiarowymi mikrokontrolerami i naprawdę małymi 8-bitowymi kontrolerami ATtiny85.

ATtiny44 ma tylko 4 kB pamięci flash i 256 bajtów pamięci RAM. Jeśli zabraknie Ci miejsca na program, możesz zawsze użyć mikrokontrolera ATtiny84 z kompatybilnymi pinami i 8-kilobajtową pamięcią flash.

Na rysunku 3.9 widać mikrokontroler ATtiny44 z podłączoną diodą LED (sprawimy, że będzie migać) i opornikami umieszczonymi na płytce stykowej. Chip ma 14 pinów: 2 zasilania, 1 resetu, a pozostałe 11 to piny GPIO (zobacz rysunek 3.10).



Rysunek 3.9. *Mikrokontroler ATtiny44 na płytce stykowej*



Rysunek 3.10. *Układ pinów ATtiny44*

Arduino jako programator

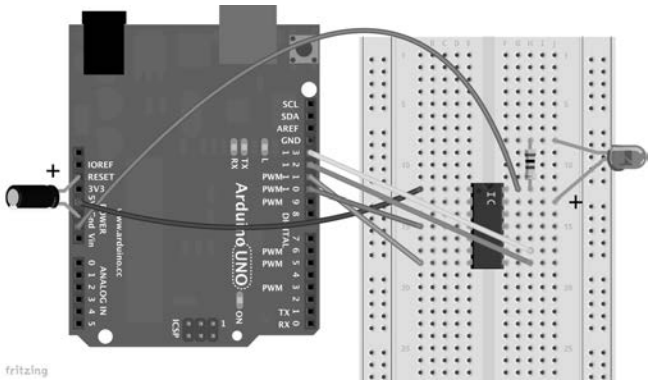
Może pamiętasz, że w rozdziale 2. Arduino Uno pełnił funkcję programatora w celu wgrania programu rozruchowego na inną płytkę Arduino. Arduino Uno możemy również użyć do zaprogramowania mikrokontrolera ATtiny, lecz zamiast wgrywać program rozruchowy, na mikrokontrolerze umieścimy cały szkic (w tym przypadku program sprawiający, że dioda LED miga).

Zanim zrobisz cokolwiek innego, musisz wgrać szkic *ArduinoISP* na Arduino Uno. Znajdziesz go w menu *Plik*, w przykładach. Następnie możesz podłączyć Arduino Uno do ATtiny w opisany niżej sposób. Ważne jest, aby najpierw wgrać szkic *ArduinoISP*, ponieważ po podłączeniu kondensatora 10 µF do pinu Reset Arduino, nie będziesz mógł wgrać żadnego szkicu, dopóki kondensator nie zostanie usunięty.

Do wykonania pokazanych w tabeli 3.1 połączeń między Arduino Uno i ATtiny można użyć przewodów podłączeniowych zakończonych obustronnie końcówką męską. Połączenia pokazano także na rysunku 3.11.

Tabela 3.1. Połączenie Arduino Uno z ATtiny44 w celu programowania

Funkcja	Arduino Uno	ATtiny44
SCK (zegar)	13	9
MISO (wejście urządzenia nadrzędnego, wyjście urządzenia podrzędnego)	12	8
MOSI (wyjście urządzenia nadrzędnego, wejście urządzenia podrzędnego)	11	7
Reset	10	4



Rysunek 3.11. *Połączenie Arduino Uno z ATtiny44 w celu programowania*

Można łatwo określić, w która stronę umieścić chip na płytce stykowej, gdyż ATtiny44 ma małą dziurkę na górnej krawędzi, obok pinu 1 (zobacz rysunek 3.10). Możliwe, że aby dopasować mikrokontroler do otworów płytki stykowej, będziesz musiał delikatnie wygiąć piny po obu stronach. Zazwyczaj sprawdza się lekkie przyciśnięcie obu stron do blatu stołu.

Kondensator ma jedną nóżkę dłuższą. Jest to przewód dodatni i należy podłączyć go do pinu Reset na Arduino. Podobnie jest w przypadku diody LED — dłuższa nóżka jest dodatnia.

Instalacja ATtinyCore w IDE

Dla ATtiny dostępnych jest wiele oprogramowań, które możesz dodać do swojego Arduino IDE. W niniejszej książce wykorzystano oprogramowanie o nazwie ATtinyCore, a jego dokumentację można znaleźć pod adresem: *https://github.com/SpenceKonde/ATtinyCore*.

Na tej stronie znajdziesz również pełną listę obsługiwanych przez ATtinyCore płytek wraz z bardzo przydatnymi schematami układu ich pinów.

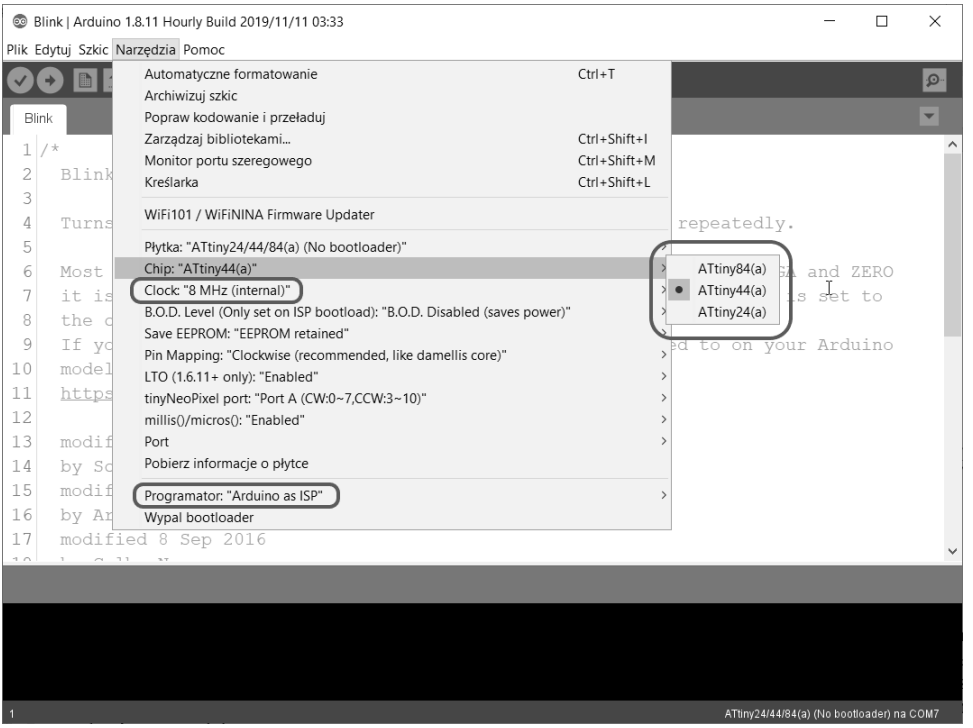
Oprogramowanie jest dostępne z poziomu menedżera płytek, jednak najpierw musisz dodać URL (*http://drazzy.com/package_drazzy.com_index.json*) w oknie preferencji (zobacz rysunek 3.4).

Po dodaniu adresu URL otwórz menedżera płytek, wyszukaj *attinycore* i zainstaluj najnowszą wersję.

Zegary, kryształy i bezpieczniki

Już wszystkie przewody są podłączone i Arduino Uno może pełnić funkcję programatora dla ATtiny. Jesteśmy prawie gotowi do wgrania szkicu na mikrokontroler, ale uprzednio musimy nieco skonfigurować Arduino IDE.

Zacznij od wyboru płytki *ATtiny24/44/84* z sekcji *ATtinyCore* (zobacz rysunek 3.12).



Rysunek 3.12. *Wybór płytki ATtiny24/44/84*

Spowoduje to dodanie kilku opcji dotyczących ATtiny w menu *Narzędzia*. Dla większości z nich można zostawić ustawienia domyślne. Powinieneś zmienić opcję *Chip* na *ATtiny44* (zobacz rysunek 3.12). Upewnij się również, że opcja *Clock* (zegar) ma wartość *8 MHz (internal)*, a w menu *Narzędzia* ustaw *Programator* (druga pozycja od końca) na *Arduino as ISP*. Jeśli masz inny programator, to możesz go również wybrać w tym miejscu.

Niektóre opcje tego menu mają wpływ na „bezpieczniki", które określają zachowanie mikrokontrolera, a nawet konfigurację danych pinów. Na przykład jeśli ustawisz opcję *Clock* na *8 MHz (internal)*, to informujesz Arduino IDE, że piny 2 i 3 ATtiny nie będą potrzebne rezonatorowi kwarcowemu, więc mogą być używane jako piny GPIO. Tutaj musimy być ostrożni, bo jeśli na przykład ustawimy opcję *Clock* na *16 MHz (external)*, to gdy będziemy chcieli zaprogramować ATtiny, okaże się, że nie jest to możliwe bez rezonatora kwarcowego. W rezultacie „zabetonowalibyśmy" nasz chip. Na szczęście wprowadzone w tych opcjach zmiany nie dają żadnego efektu, dopóki nie naciśniesz funkcji *Wypal bootloader*, która znajduje się na końcu menu *Narzędzia*. W rzeczywistości ta funkcja nie wgrywa programu rozruchowego na ATtiny, a jedynie ustawia bezpieczniki.

W końcu możesz wgrać szkic na mikrokontroler. Możesz użyć szkicu o nazwie *03_02_attiny44_blink*, który znajduje się na stronie *http://www.helion.pl/ksiazki/arpok2.htm*. Aby wgrać szkic na ATtiny, wystarczy, że naciśniesz przycisk *Wgraj*, tak jak w przypadku standardowego Arduino. Po zakończonym wgrywaniu dioda LED powinna zacząć migać.

Minimalne Arduino

Jak pewnie zauważyłeś, większą część Arduino Uno zastąpiliśmy chipem umieszczonym na płytce stykowej. Zatem jak udało nam się pozbyć tak wielu elementów i co poświęciliśmy w zamian?

Przede wszystkim utraciliśmy źródło zasilania. Nie mamy już stabilizatora napięcia. W tym momencie ATtiny korzysta z zasilania Arduino Uno. Jednak ATtiny zadowoli się każdym źródłem zasilania o napięciu z zakresu od 2,7 V do 5,5 V. Więc gdy zaprogramujesz już ATtiny, możesz zasilać go napięciem 3 V z dwóch baterii AA.

Z powodu tego, że użyliśmy Arduino Uno jako programatora, nie potrzebowaliśmy chipu interfejsu USB i powiązanych z nim komponentów. Nie mamy również oscylatora kwarcowego 16 MHz, małego srebrnego komponentu, który możesz znaleźć na Arduino Uno. ATtiny ma swój własny wbudowany oscylator, który działa o połowę wolniej. Jest to rekompensowane przez oprogramowanie i tak funkcja `delay` wciąż zatrzymuje program na zadany czas. Wbudowany oscylator nie jest tak dokładny jak zewnętrzny kryształ. Według dokumentacji dokładność oscylatora ATtiny44 wynosi ±2 procent, natomiast margines błędu zewnętrznego oscylatora to ±0,003 procent. Jednak w rzeczywistości dokładność oscylatora ATtiny jest dużo lepsza od tej podanej w dokumentacji.

Podsumowanie

W tym rozdziale poznaliśmy kilka płytek, które stanowią alternatywę dla oficjalnych płytek Arduino, włączając w to możliwość stworzenia własnego Arduino za pomocą mikrokontrolera i płytki stykowej. Niektórych z tych urządzeń będziemy używać w kolejnych rozdziałach.

W następnym rozdziale przyjrzymy się przerwaniom i zegarom, dzięki którym Arduino może reagować na określone zdarzenia zewnętrzne i na te zaplanowane.

Zostań mistrzem Arduino!

W pewnych kwestiach nic się nie zmienia — Arduino pozostaje praktycznym, tanim mikrokontrolerem, który może komunikować się ze światem zewnętrznym i obsługiwać przeróżne urządzenia elektroniczne. Daje też pole do popisu licznym amatorom i hobbystom. Środowisko programistyczne wciąż ma prosty interfejs, a programowanie płytki samo w sobie jest łatwym zadaniem nawet dla ludzi bez doświadczenia w pisaniu kodu. Zmiany w świecie Arduino dotyczą większych możliwości oraz sukcesywnie dodawanych funkcjonalności, takich jak obsługa bibliotek, czy kolejnych interfejsów.

To drugie, starannie zaktualizowane wydanie przewodnika po programowaniu Arduino, przeznaczonego dla osób, które opanowały podstawy i chcą zdobyć umiejętności na wyższym, profesjonalnym poziomie. W książce dodatkowo ujęto zagadnienia wykorzystania Arduino do komunikacji z urządzeniami szeregowymi oraz z internetem rzeczy (IoT). Dowiesz się również, jak używać środowiska programistycznego Arduino do programowania kompatybilnych płytek. Prezentowane treści zostały zilustrowane praktycznymi przykładami stosowania omówionych technik. Nie zabrakło licznych kodów do pobrania, które ułatwią Ci pracę z nawet bardzo ambitnymi projektami.

Dzięki książce dowiesz się, jak:

- konfigurować Arduino IDE i tworzyć efektywne szkice
- poprawić wydajność pracy przy zmniejszeniu natężenia prądu pobieranego przez Arduino
- pracować z różnymi interfejsami: I2C, 1-Wire, SPI, a także z układem TTL, USB i UART
- korzystać z Ethernetu, Bluetootha i DSP oraz z zasobów internetu
- tworzyć i udostępniać własne biblioteki

Dr Simon Monk jest inżynierem cybernetykiem, informatykiem i uzdolnionym hakerem. Po kilku latach pracy na uczelni został przedsiębiorcą. Obecnie dzieli swój czas pomiędzy pisanie rozchwytywanych książek i projektowanie produktów dla MonkMakes — firmy, którą prowadzi wraz z żoną Lindą. Mieszka w Preston w Wielkiej Brytanii.

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

ISBN 978-83-283-7548-2
9 788328 375482

INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 54,90 zł