



Technologia i rozwiązania

Arduino

Automatyka domowa dla każdego

Inteligentne budynki dla pasjonatów elektroniki!



Marco Schwartz

[PACKT] open source*
PUBLISHING community experience distilled

Tytuł oryginału: Arduino Home Automation Projects

Tłumaczenie: Zbigniew Waśko

ISBN: 978-83-283-0936-4

Copyright © 2014 Packt Publishing

First published in the English language under the title 'Arduino Home Automation Projects – (9781783986064)'.

Polish edition copyright © 2015 by Helion S.A.

All rights reserved.

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 Wydawnictwo HELION 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 Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

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)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/ardaud.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/ardaud>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

Wprowadzenie	9
Rozdział 1. Budowanie bezprzewodowych detektorów ruchu	13
Niezbędny sprzęt i oprogramowanie	14
Montaż sprzętu	15
Sprzęganie czujnika PIR z Arduino	17
Programowanie przesyłania danych	18
Tworzenie interfejsu graficznego dla detektorów ruchu	21
Podsumowanie	24
 Rozdział 2. Sterowanie oświetleniem za pomocą smartfona lub tabletu	 25
Niezbędny sprzęt i oprogramowanie	26
Montaż sprzętu	28
Testowanie przekaźników i łącza Wi-Fi	31
Tworzenie interfejsu graficznego do sterowania przekaźnikami	34
Testowanie interfejsu graficznego	36
Podsumowanie	37
 Rozdział 3. Bezprzewodowy pomiar temperatury	 39
Niezbędny sprzęt i oprogramowanie	40
Montaż sprzętu	42
Pisanie szkicu	45
Testowanie czujnika temperatury i wilgotności	46
Zdalny pomiar temperatury i wilgotności	48
Podsumowanie	50

Rozdział 4. Stacja pogodowa w chmurze	53
Niezbędny sprzęt i oprogramowanie	54
Łączenie podzespołów	55
Testowanie czujników	56
Zakładanie konta w serwisie ThingSpeak	58
Pisanie szkicu	60
Logowanie w serwisie ThingSpeak i wyświetlanie danych	62
Podsumowanie	64
Rozdział 5. Chmurowy monitoring zużycia energii	65
Niezbędny sprzęt i oprogramowanie	66
Łączenie podzespołów	68
Testowanie układu	70
Konfigurowanie konta w serwisie ThingSpeak	73
Wysyłanie danych do serwisu ThingSpeak	75
Podsumowanie	78
Rozdział 6. Przerabianie urządzeń fabrycznych	79
Niezbędny sprzęt i oprogramowanie	80
Łączenie podzespołów	82
Sterowanie urządzeniem za pomocą komputera	84
Tworzenie interfejsu graficznego	86
Podsumowanie	88
Rozdział 7. Zbuduj własny system automatyki domowej	91
Niezbędny sprzęt i oprogramowanie	92
Budowanie systemu Arduino od podstaw	93
Testowanie systemu Arduino	95
Projektowanie płytki drukowanej dla własnego systemu automatyki	97
Wykonanie płytki	101
Projektowanie i drukowanie 3D obudowy dla własnego systemu automatyki domowej	103
Podsumowanie	106
Skorowidz	109

Sterowanie oświetleniem za pomocą smartfona lub tabletu

W tym rozdziale zajmiemy się inną, równie popularną dziedziną automatyki domowej — sterowaniem oświetleniem. Na półkach sklepowych znajdziesz mnóstwo gotowych systemów służących do sterowania urządzeniami domowymi, ale zazwyczaj są one bardzo drogie. I jeśli nawet są bezprzewodowe, to na ogół wymagają od nas używania specjalnego pilota.

My jednak zastosujemy inne podejście i zbudujemy system, dzięki któremu do sterowania oświetleniem będzie można użyć komputera, smartfona lub tabletu. Oto krótki opis tego, co nas czeka:

- Najpierw zmontujemy sprzętową część projektu, czyli połączymy chip Wi-Fi z płytką Arduino i urządzenie sterowane (w tym przypadku zwykłą lampę) z modulem przekaźnika. W naszym projekcie zastosujemy dwa takie moduły, ale oczywiście może być ich więcej.
- Następnie napiszemy kod testujący poszczególne podzespoły. Sprawdzimy więc, czy przekaźnik działa poprawnie i czy chip Wi-Fi rzeczywiście łączy się z siecią domową.
- Na koniec opracujemy graficzny interfejs, który umożliwi sterowanie urządzeniem nie tylko z poziomu komputera, ale również smartfona lub tabletu.

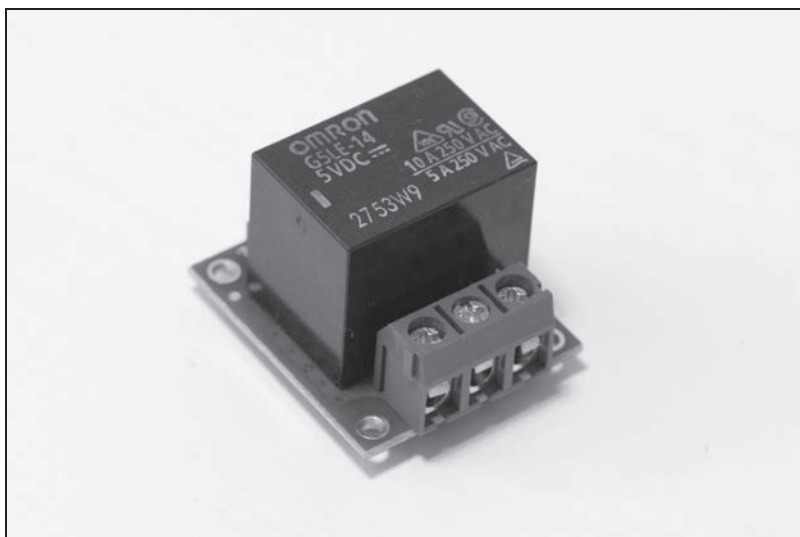
Niezbędny sprzęt i oprogramowanie

Zobaczmy, co konkretnie będzie potrzebne do realizacji tego projektu. Tak jak we wszystkich projektach opisanych w książce, podstawą będzie płytka Arduino Uno (może być też Arduino Mega).

Potrzebne będą również moduły przekaźnikowe jako elementy bezpośrednio sterujące lampami. Przekaźnik to taki elektromechaniczny wyłącznik. Zwykle używa się go, gdy obwód niskonapięciowy (np. płytka Arduino pracująca przy napięciu 5 V) ma wywołać zmianę w obwodzie o znacznie wyższym napięciu (np. w lampie podłączonej do gniazdka w ścianie). Więcej informacji o przekaźnikach znajdziesz pod adresem:

<http://pl.wikipedia.org/wiki/Przekaźnik>.

Ja zastosowałem moduły przekaźnikowe marki Pololu przystosowane do przełączania obwodów o napięciu do 250 V (mogą więc obsługiwać urządzenia zasilane z sieci zarówno 110 V, jak 230 V) i natężeniu prądu do 10 A. Jeden z nich jest pokazany na rysunku poniżej.



Jako że chciałem sterować dwiema lampami, użyłem dwóch takich modułów. Oczywiście nie stoi na przeszkodzie, abyś zastosował ich więcej albo poprzestał tylko na jednym.

Nie musisz też szukać dokładnie takich samych przekaźników jak moje. Mogą być dowolnej innej marki, byleby tylko miały podobne parametry — po stronie wejściowej muszą być kompatybilne z obwodami 5 V (jak płytka Arduino), a po stronie wyjściowej muszą być przystosowane do napięcia i natężenia prądu zasilającego urządzenie sterowane. Jak już podawałem, przekaźniki użyte przeze mnie mogą sterować urządzeniami zasilanymi prądem o napięciu nawet 230 V i natężeniu 10 A, czyli mającymi moc aż 2300 W.

Ponieważ będziemy mieć do czynienia z urządzeniami działającymi przy napięciu 110 V lub 230 V, musisz zachować szczególną ostrożność. Nigdy nie dotykaj przełącznika podłączonego kablem do gniazdka, a nawet nie zbliżaj się do niego. Jeśli to możliwe, umieść przełącznik w jakiejś plastikowej obudowie, aby nikt nie miał do niego dostępu.

Kolejna rzecz, jakiej będziesz potrzebował, to moduł Wi-Fi. Kod napisany na potrzeby tego projektu uwzględnia parametry modelu CC3000, popularnego wśród użytkowników platformy Arduino, więc bez trudu znajdziesz w internecie informacje na jego temat. Jest to **plytka drukowana (PCB)** z interesującym nas chipem i kilkoma elementami niezbędnymi dla jego właściwego funkcjonowania. Poniższy rysunek przedstawia płytkę, której użyłem.



Podobnie jak w przypadku przełączników, płytka Wi-Fi może pochodzić od dowolnego producenta. Musi tylko zawierać chip CC3000 i być dostosowana do zasilania 5 V (większość płytek z chipem CC3000 spełnia ten warunek). Możesz użyć również nakładki Arduino zawierającej wspomniany chip, chociażby tej, którą oferuje firma Adafruit.

Aby dowiedzieć się czegoś więcej na temat tej płytki i działania układu CC3000, przejrzyj dokumentację i przykłady zastosowań zamieszczone w internecie pod adresem:

https://github.com/adafruit/Adafruit_CC3000_Library.

Musisz zdobyć także płytkę stykową i przewody montażowe, aby te wszystkie podzespoły połączyć.

Oto pełna lista części potrzebnych do wykonania systemu sterowania oświetleniem:

- Arduino Uno R3 (<http://www.adafruit.com/products/50>)¹,
- przekaźnik Pololu 5V (<http://www.pololu.com/product/2480>)²,
- moduł Wi-Fi CC3000 (<http://www.adafruit.com/products/1469>)³,
- płytka stykowa (<http://www.adafruit.com/product/64>)⁴,
- przewody połączeniowe (<http://www.adafruit.com/product/759>)⁵.

Do połączenia lamp z przekaźnikami będziesz potrzebował paru przewodów z gniaздkami i wtyczkami. Więcej szczegółów podam przy omawianiu montażu. Jeśli chodzi o oprogramowanie, to będziesz musiał zainstalować w swoim komputerze najnowszą wersję środowiska programistycznego Arduino. Potrzebna będzie również biblioteka aREST. Znajdziesz ją pod adresem:

<https://github.com/marcoschwartz/aREST>.

Aby pobrać repozytorium z serwisu GitHub, po prostu kliknij przycisk *Download Zip*. Potrzebna będzie też biblioteka chipu CC3000. Znajdziesz ją pod adresem:

https://github.com/adafruit/Adafruit_CC3000_Library.

Z kolei bibliotekę CC3000_MDNS znajdziesz pod adresem:

https://github.com/adafruit/CC3000_MDNS.

Aby zainstalować bibliotekę, wypakuj wszystkie pliki do folderu */libraries*, znajdującego się w głównym folderze instalacyjnym aplikacji Arduino. Pod koniec rozdziału będziemy tworzyć interfejs sieciowy dla naszego systemu i wtedy niezbędny będzie działający na komputerze serwer internetowy.

Montaż sprzętu

Zobaczymy teraz, jak należy te wszystkie komponenty połączyć. Zaczniemy od połączenia modułu przekaźnika i chipu Wi-Fi z płytką Arduino. Cały zmontowany układ powinien ostatecznie wyglądać mniej więcej tak jak na poniższym rysunku.

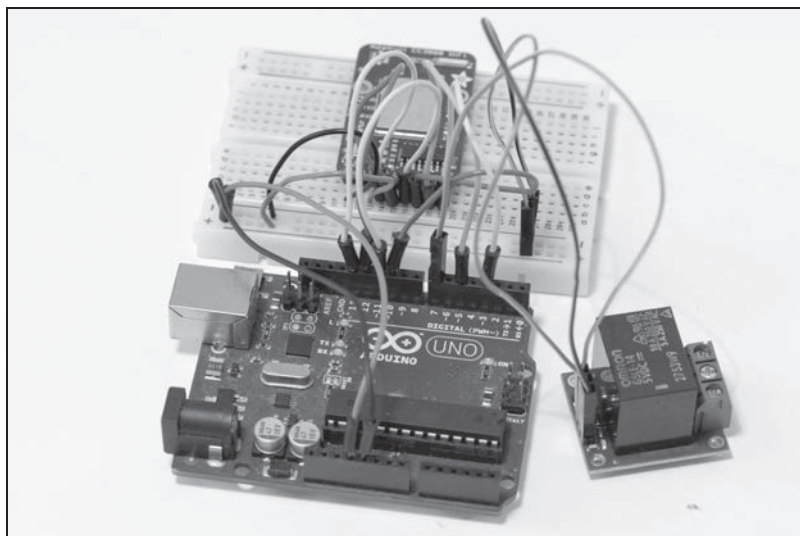
¹ <http://botland.com.pl/arduino-moduly-glowne/1060-arduino-uno-r3.html> — przyp. tłum.

² <http://www.gotronik.pl/modul-przekaznika-5v-optoizolacja-p-2366.html> — przyp. tłum.

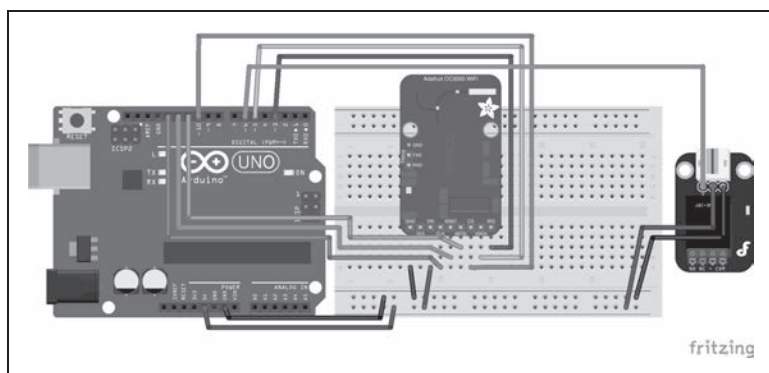
³ <http://botland.com.pl/moduly-wifi/2276-cc3000-modul-wifi-sparkfun.html> (współpracuje z biblioteką Adafruit, ale dedykowana biblioteka jest dostępna pod adresem http://github.com/sparkfun/SFE_CC3000_Library/archive/master.zip) — przyp. tłum.

⁴ <http://botland.com.pl/plytki-stykowe/56-plytki-stykowa-400-otworow.html> — przyp. tłum.

⁵ <http://botland.com.pl/przewody-polaczeniowe/1223-przewody-polaczeniowe-mesko-meskie-20cm-40szt.html> — przyp. tłum.



Na następnym rysunku pokazałem dokładnie, co z czym należy połączyć.



Schemat ten przedstawia połączenia dla systemu w wersji z jednym przełącznikiem. Zobaczmy, jak to wygląda w szczegółach. Zaczynij od umieszczenia wszystkich komponentów jeden blisko drugiego, po czym umieść moduł CC3000 na płytce uniwersalnej. Następnie wykonaj połączenia zasilające. Pin masy (GND) płytki Arduino połącz z zaciskiem masy płytki CC3000, a zacisk 5V płytki Arduino — z zaciskiem VIN płytki CC3000.

Przejdźmy teraz do samego chipu CC3000. Jak widać, jego moduł ma całkiem sporo wyprowadzeń, więc postępuj dokładnie według moich wskazówek, bo jeśli się pomylisz, moduł może nie zadziałać lub będzie działał niewłaściwie. Pin IRQ modułu CC3000 połącz z pinem 3 płytki Arduino, VBEN z pinem 5 i CS z pinem 10. Następnie wykonaj połączenie linii interfejsu szeregowego SPI: piny MOSI, MISO i CLK modułu CC3000 połącz z pinami, odpowiednio, 11, 12 i 13 płytki Arduino.

Teraz podłącz moduł przekaźnika. Moduł ten ma trzy piny wejściowe: dwa zasilające (VCC i GND) i jeden sygnałowy (SIG). Łącząc go z płytką Arduino, połącz SIG z pinem 6, GND z GND i VCC z pinem 5V.

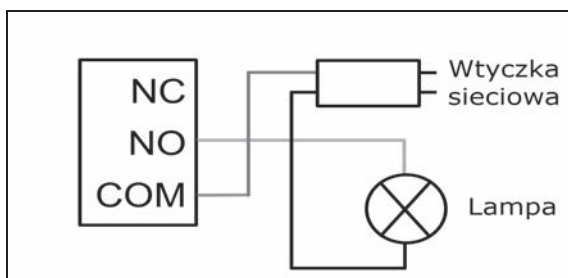
Na koniec wykonaj podłączenie lampy do modułu przekaźnika. Zrób to za pomocą dwóch przewodów: jednego z wtyczką i drugiego z gniazdkiem. Zmontowane połączenie jest pokazane na rysunku poniżej.



Zaciski wyjściowe przekaźnika to zazwyczaj: COM (wspólny), NC (normalnie zwarty) i NO (normalnie otwarty). Jeden koniec przewodu zasilającego ma być podłączony do zacisku COM, drugi do zacisku NO, a zacisk NC powinien zostać wolny.

Aby prawidłowo połączyć kable zasilające z przekaźnikiem, wykonaj następujące czynności:

1. Jeden koniec przewodu z gniazdkiem połącz z zaciskiem COM.
2. Jeden koniec przewodu z wtyczką połącz z zaciskiem ON.
3. Pozostałe końce przewodów złóż za pomocą zwykłej kostki elektrycznej.
4. Teraz możesz podłączyć lampę. Połączenia między lampą a przekaźnikiem powinny być zgodne z poniższym schematem.



5. Przewód z wtyczką włącz do gniazdka w ścianie, a do przewodu z gniazdkiem podłącz lampę. Oczywiście całą procedurę musisz powtórzyć tyle razy, ile przekaźników zastosujesz. Zasilania kolejnych przekaźników podłączaj do linii +5V i GND na płytce stykowej, a końcówki SIG — do niezajętych pinów płytki Arduino.

Testowanie przekaźników i łącza Wi-Fi

Po zmontowaniu całego układu możemy przystąpić do pisania kodu, który posłuży do przetestowania naszego projektu. Zaczniemy od przygotowania szkicu testującego działanie przekaźnika. Otwórz więc środowisko programistyczne Arduino i przygotuj się do pisania. Szczegółowo omówię tylko najważniejsze partie kodu, a całość możesz pobrać z serwera FTP zawierającego materiały dodatkowe do książki.

Najpierw deklarujemy, do którego pinu jest podłączony testowany przekaźnik:

```
const int relay_pin = 6;
```

Używamy tutaj zmiennej `const int` — choć moglibyśmy użyć dyrektywy `#define` — ponieważ dzięki temu mamy pewność, że stała jest odpowiedniego typu. W funkcji `setup()` musimy zadeklarować, że jest to wyjściowy pin cyfrowy. Posłużymy się w tym celu funkcją `pinMode()`:

```
pinMode(relay_pin, OUTPUT);
```

W funkcji `loop()` włączamy przekaźnik, czekamy 5 sekund, wyłączamy go, czekamy kolejne 5 sekund i powtarzamy procedurę od początku. Zrobimy to za pomocą następującego kodu:

```
// Włącz przekaźnik
digitalWrite(relay_pin, HIGH);
// Czekaj 5 sekund
delay(5000);
// Wyłącz przekaźnik
digitalWrite(relay_pin, LOW);
// Czekaj 5 sekund
delay(5000);
```

Teraz prześlij ten kod do płytki Arduino. Jeśli wszystkie połączenia wykonałeś prawidłowo, lampa podłączona do przekaźnika powinna zapalać się i gasnąć w odstępach 5-sekundowych, a każdej takiej zmianie powinno towarzyszyć kliknięcie styków przekaźnika. Jeśli tak się nie dzieje, sprawdź połączenia, zwłaszcza te między modulem przekaźnika a płytką Arduino.

Zanim dotkniesz czegokolwiek w zmontowanym układzie, sprawdź, czy przewód zasilający jest odłączony od gniazdka w ścianie. Nie ryzykuj przypadkowego dotknięcia odizolowanego kabelka, jeśli urządzenie jest pod napięciem, bo to może być niebezpieczne.

Kod tego szkicu jest dostępny w materiałach dodatkowych do książki zamieszczonych na serwerze FTP: <ftp://ftp.helion.pl/przyklady/ardaud.zip> — znajdziesz go w folderze *Rozdział_02/relay_test*.

Jeśli test przekaźnika wypadł pomyślnie, możesz przejść do następnego etapu, polegającego na napisaniu szkicu sprawdzającego, czy chip CC3000 działa poprawnie i czy możliwe jest sterowanie przekaźnikiem w sposób bezprzewodowy.

Zacznij od zadeklarowania odpowiednich bibliotek:

```
#include <Adafruit_CC3000.h>
#include <SPI.h>
#include <CC3000_MDNS.h>
#include <aREST.h>
```

Zadeklaruj też, do których pinów jest przyłączony chip CC3000. Jeśli zmontowałeś układ zgodnie z moimi wskazówkami, nie zmieniaj niczego w poniższym zapisie:

```
#define ADAFRUIT_CC3000_IRQ 3
#define ADAFRUIT_CC3000_VBAT 5
#define ADAFRUIT_CC3000_CS 10
```

Następnie korzystając z biblioteki CC3000, utwórz instancję chipu CC3000 z podanymi wcześniej pinami:

```
Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS,
    ADAFRUIT_CC3000_IRQ, ADAFRUIT_CC3000_VBAT, SPI_CLOCK_DIV2);
```

Aby chip „wiedział”, z którą siecią ma się łączyć, trzeba odpowiednio ustawić pewne parametry. Przede wszystkim musisz podać identyfikator swojej sieci Wi-Fi (SSID) i hasło:

```
#define WLAN_SSID "twójSSID"
#define WLAN_PASS "twojehaśło"
#define WLAN_SECURITY WLAN_SEC_WPA2
```

Jeśli nie używasz standardu WPA2, będziesz musiał zmienić ustawienie związane z zabezpieczeniem sieci. Pełną listę możliwych ustawień bezpieczeństwa znajdziesz w przykładowych szkicach ilustrujących stosowanie biblioteki CC3000:

```
// Ustawieniem bezpieczeństwa może być: WLAN_SEC_UNSEC, WLAN_SEC_WEP,
WLAN_SEC_WPA lub WLAN_SEC_WPA2
```

Podaj też numer portu, na którym komputer będzie odbierał połączenia przychodzące. Ja wybrałem port 80 ze względu na wygodę — można sterować płytką przez wpisywanie poleceń bezpośrednio w przeglądarce:

```
#define LISTEN_PORT 80
```

Kolejna rzecz, jaką musisz podać, to serwer działający na płytce Arduino i przyjmujący tam połączenia przychodzące:

```
Adafruit_CC3000_Server restServer(LISTEN_PORT);
```

Następnie utwórz instancję usługi mDNS, która umożliwi dostęp do płytki z pominięciem jej adresu IP:

```
MDNSResponder mdns;
```

Nie zapomnij też o utworzeniu instancji biblioteki aREST, aby możliwe było przyjmowanie żądań ze świata zewnętrznego:

```
aREST rest = aREST();
```

W funkcji `setup()` po zainicjalizowaniu ustawień płytki i połączenia z siecią Wi-Fi możesz uruchomić usługę mDNS. Jest to też moment, kiedy musisz przypisać płytce Arduino jakąś nazwę. Ja nazwałem ją po prostu `arduino`, a to oznacza, że później aby uzyskać do niej dostęp, będę musiał wpisywać w przeglądarce nazwę `arduino.local`:

```
if (!mdns.begin("arduino", cc3000)) {
    while(1); }
```

Może się jednak zdarzyć, że serwer mDNS nie zadziała na Twoim komputerze — więcej informacji znajdziesz na stronie biblioteki CC3000 mDNS. Jeśli tak się stanie, będziesz musiał uzyskać dokładny adres IP płytki CC3000. Możesz się w tym celu posłużyć podaną niżej funkcją, która będzie wyświetlała żądany adres przy każdym uruchomieniu płytki:

```
while (!displayConnectionDetails()) {
    delay(1000); }
```

Aby zakończyć fazę ustawiania parametrów, musisz jeszcze uruchomić serwer na płytce Arduino. Zrobisz to za pomocą następującego kodu:

```
restServer.begin();
Serial.println(F("Nasłuchiwanie połączen..."));
```

Teraz w funkcji `loop()` musisz przede wszystkim zaktualizować usługę mDNS:

```
mdns.update();
```

Sprawdź też, czy jakiś klient nie chce się połączyć z internetowym serwerem Arduino, a jeśli coś takiego ma miejsce, obsłuż klienta, używając biblioteki aREST:

```
Adafruit_CC3000_ClientRef client = restServer.available();
rest.handle(client);
```

Dzięki bibliotece aREST nie musisz w tej sprawie robić nic więcej! Teraz możesz przesłać kod na płytkę (upewnij się wcześniej, że wpisałeś właściwy identyfikator SSID i hasło), a gdy już to zrobisz, otwórz monitor portu szeregowego. Po kilku sekundach powinieneś ujrzeć napis:

```
Nasłuchiwanie połączen...
```

Jeśli go widzisz, to znaczy, że płytka jest połączona z siecią Wi-Fi. Powinien być widoczny również adres IP płytki. Zanotuj go, gdyż wkrótce będzie Ci potrzebny.

Zamknij monitor portu szeregowego i otwórz przeglądarkę internetową. Na początek musisz określić jako wyjściowe piny, do których podłączone są przekaźniki. Zaczynając od pinu nr 6, wpisz w polu adresowym przeglądarki następujący tekst:

```
http://arduino.local/mode/6/o
```

Jeśli wszystko zadziała jak należy, powinieneś ujrzeć potwierdzenie w postaci następującego komunikatu:

Setting pin D6 to output

Jeśli jest inaczej, nie wpadaj w panikę. Może po prostu biblioteka mDNS jest niekompatybilna z Twoim komputerem. Spróbuj jeszcze raz, ale zamiast aliasu `arduino.local` użyj adresu IP płytki CC3000. Wpisz np. coś takiego:

```
http://192.168.0.103/mode/6/o
```

To samo powtórz dla pinu nr 7, jeśli do niego podłączyłeś kolejny przekaźnik. A teraz spróbuj uaktywnić za pośrednictwem łącza Wi-Fi pierwszy przekaźnik. Wpisz w przeglądarce:

```
http://arduino.local/digital/6/1
```

Natychmiast powinieneś usłyszeć kliknięcie przekaźnika i lampa powinna się zaświecić, jeśli tylko jest podłączona do układu. Natomiast w przeglądarce powinien się ukazać komunikat potwierdzający ustawienie wysokiego poziomu napięcia na wskazanym pinie:

Pin D6 set to 1

To samo możesz zrobić z pinem nr 7, po prostu żeby sprawdzić, czy on też daje się sterować za pośrednictwem Wi-Fi. Oczywiście całą tę procedurę testową powinieneś dostosować do liczby użytych przekaźników i do sposobu, w jaki je połączyłeś z płytką.

Pełny kod potrzebny do przetestowania przekaźników i połączenia Wi-Fi można pobrać z serwera FTP: <ftp://ftp.helion.pl/przyklady/ardaud.zip>.

Tworzenie interfejsu graficznego do sterowania przekaźnikami

Do samego testowania układu można posłużyć się przeglądarką internetową, ale używanie jej na co dzień do sterowania lampami byłoby raczej niewygodne. Dlatego zbudujemy specjalny interfejs, który umożliwi włączanie i wyłączanie poszczególnych źródeł światła przez zwykłe klikanie przycisków.

Dzięki temu, że korzystamy z technologii internetowych, interfejs ten będzie dostępny również na urządzeniach mobilnych, takich jak smartfon czy tablet, jeśli tylko będą połączone z tą samą siecią Wi-Fi co system sterujący.

Podobnie jak w pierwszym rozdziale, przy opracowywaniu interfejsu będziemy używać języków HTML, CSS, JavaScript i PHP. Zaczniemy od tego, co trzeba napisać w HTML-u. Będą to bloki kodu odnoszące się do przekaźników. Każdemu z tych urządzeń przypiszemy dwa przyciski: jeden do załączania i drugi do wyłączania. Oto kod realizujący tę część interfejsu:

```
<div class="relayBlock"><span class="relayTitle">Przekaźnik 1</span>
  <button class="btn btn-block btn-lg btn-primary" type="button"
    id="1" onClick="relayClick(this.id)">Włącz</button>
  <button class="btn btn-block btn-lg btn-danger" type="button"
    id="2" onClick="relayClick(this.id)">Wyłącz</button>
</div>
```

Żeby przyciski mogły robić cokolwiek użytecznego, trzeba je powiązać z odpowiednim kodem w języku JavaScript. Wszystko to jest zakodowane w pliku JavaScript. Pobieramy identyfikator (ID) przycisku, który został kliknięty, i wysyłamy do układu sterującego odpowiednie dla tego przycisku polecenie. Przykładowo, jeśli przycisk o identyfikatorze równym 1 jest przypisany do przekaźnika połączonego z pinem nr 6 i służy do włączania tego przekaźnika, to za obsługę jego kliknięcia będzie odpowiedzialny następujący fragment kodu:

```
if (clicked_id == "1"){
  $.get( "update_relay.php", {
    command: "/digital/6/1" } );
}
```

To samo zostało zrobione dla pozostałych przekaźników. Oczywiście zawartość tego pliku musisz odpowiednio zmodyfikować w zależności od sprzętowej konfiguracji swojego układu.

Spójrzmy teraz na skrypt PHP, który będzie wywoływany przy każdym kliknięciu przycisku. Wiemy już, że za każdym razem jest wtedy przekazywana do skryptu zmienna o nazwie `command` (polecenie). Wewnątrz skryptu musimy najpierw pobrać jej wartość:

```
$command = $_GET['command'];
```

Skoro mamy już polecenie, które należy wysłać, możemy przystąpić do tworzenia łańcucha znakowego zawierającego właściwy adres URL. Taki właśnie łańcuch wpisywaliśmy w przeglądarce podczas testowania szkicu realizującego łączność Wi-Fi. Przypominam, że jeśli usługa mDNS nie działa na Twoim komputerze, musisz w tym łańcuchu umieścić adres IP modułu CC3000. A oto jak należy utworzyć zmienną zawierającą stosowny łańcuch z adresem URL i poleceniem:

```
$service_url = 'http://arduino.local'. $command;
```

Teraz możemy zainicjować usługę cURL. Robimy to w sposób następujący:

```
$curl = curl_init($service_url);
```

Potem zlecamy wykonanie polecenia:

```
$curl_response = curl_exec($curl);
```

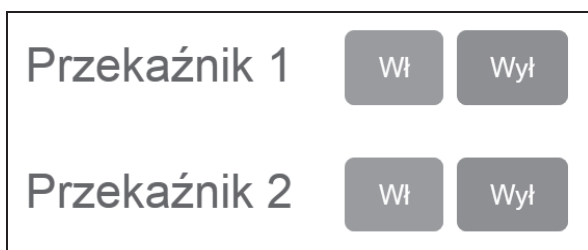
A następnie zamykamy usługę cURL:

```
$curl_close($curl);
```

Zauważ, że tym razem nie interesuje nas odpowiedź płytki Arduino, więc jej nie odbieramy i nigdzie nie wysyłamy.

Testowanie interfejsu graficznego

Wszystko zmontowane, kod interfejsu napisany, więc czas na generalny sprawdzian. Upewnij się, że Twój serwer internetowy działa i że w jego głównym folderze znajdują się wszystkie pliki interfejsu. Aby to sprawdzić, otwórz ten folder (przez wpisanie w przeglądarce internetowej adresu localhost). Gdy już się tam dostaniesz, otwórz plik HTML. Powinieneś ujrzeć coś zbliżonego do rysunku zamieszczonego poniżej.



Aby przetestować interfejs, po prostu klikaj widoczne przyciski. Każde kliknięcie powinno spowodować zmianę stanu odpowiedniej lampy. Jeśli jest inaczej, musisz sprawdzić kilka rzeczy.

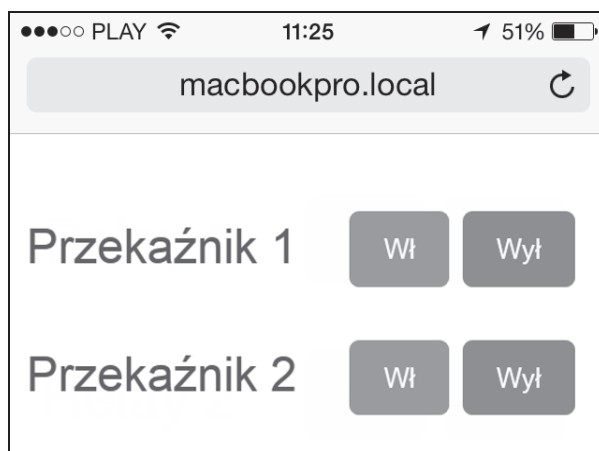
Przed wszystkim upewnij się, że serwer internetowy działa poprawnie. Sprawdź też, czy w skrypcie PHP wpisałeś właściwą nazwę płytki Arduino (lub adres IP). Na koniec przetestuj jeszcze raz szkic sterujący działaniem płytki Arduino, wpisując polecenia bezpośrednio w przeglądarce internetowej. Upewnij się też, czy przekaźniki są nadal podłączone do płytki, tak samo jak były podczas pisania wspomnianego szkicu.

Pełny kod interfejsu można pobrać z serwera FTP zawierającego materiały dodatkowe do książki.

Opisanego tu interfejsu możesz używać również za pośrednictwem telefonu komórkowego lub tabletu — wystarczy, że znajdziesz w sieci nazwę swojego komputera. Dużo zależy od systemu operacyjnego, ale zwykle znajdziesz ją w ustawieniach systemowych pod nagłówkiem *Ustawienia sieci* lub podobnym. Przykładowo w systemie OS X mój komputer był dostępny pod nazwą macbookpro.local. W systemie Windows znajdziesz odpowiednią opcję,

wybierając kolejno *Panel sterowania/Sieć i Internet/Wyświetl stan sieci i zadania*. Jeśli używasz Linuksa, możesz po prostu wpisać w terminalu nazwę hosta.

Po wskazaniu nazwy komputera w urządzeniu przenośnym będziesz mógł przejść bezpośrednio do folderu z plikami interfejsu. Samo posługiwanie się nim będzie wyglądało tak samo jak na komputerze. Poniżej zamieściłem rysunek pokazujący, jak ten interfejs wygląda w smartfonie.



Podsumowanie

Podsumujmy teraz to, co zrobiliśmy w ramach drugiego projektu. Otóż zbudowaliśmy system bezprzewodowego sterowania przełącznikami za pomocą komputera lub urządzenia mobilnego. System działa tak samo jak te kupowane w sklepach, ale jego zaletą jest to, że w każdej chwili możesz go łatwo dostosować do swoich potrzeb. Ważne jest również to, że do sterowania tym systemem nie musisz mieć kolejnego pilota — wystarczy, że sięgniesz po swój smartfon.

Zobaczymy jeszcze raz w skrócie, jak przebiegała realizacja tego projektu. Najpierw zgromadziliśmy komponenty niezbędne do zbudowania systemu, w tym moduły przełącznikowe potrzebne do włączania i wyłączania sterowanych urządzeń. Przygotowaliśmy również chip Wi-Fi, aby móc połączyć wszystko z lokalną siecią internetową.

Następnie zmontowaliśmy cały układ sterujący i podłączyliśmy do niego lampę biurkową, aby sterować nią poprzez łącze Wi-Fi.

Potem napisaliśmy odpowiednie szkice dla platformy Arduino, które umożliwiły nam sprawdzenie poprawności działania poszczególnych modułów przełącznikowych oraz skuteczności połączenia systemu z siecią Wi-Fi.

Na koniec zbudowaliśmy interfejs internetowy w celu ułatwienia sterowania urządzeniami podłączonymi do systemu. Interfejs został zbudowany w oparciu o standardy internetowe, takie jak HTML czy JavaScript, a zatem można go używać na każdym urządzeniu mogącym się połączyć z siecią Wi-Fi i umożliwiającym skorzystanie z przeglądarki internetowej.

Jeśli chcesz sterować oświetleniem w kilku pokojach, bez problemu możesz dołożyć kolejne płytki Arduino, ale każdej z nich będziesz musiał przypisać inną, unikatową nazwę, np. `arduino1.local`, `arduino2.local` itd. Oczywiście będziesz też musiał odpowiednio przerobić kod po stronie serwera.

Skorowidz

A

ACS712, 67, 72
Adafruit, 27, 40
Adafruit CC3000, 54
Arduino, 13, 14, 17
 IDE, 55
 Mega, 40, 54
 Micro, 81
 Mini, 81
 Uno, 23, 26, 92
 Uno R3, 28, 41, 54, 67
aREST, 15, 19, 28
ATmega328, 94
Autorouter, 100

B

biblioteka
 aREST, 15, 19, 28, 82
 DHT, 42, 45, 55
 JeeLib, 93, 96
 jQuery, 21
Bluefruit EZ-Link, 40
Bluetooth, 39
Bluetooth Bluefruit EZ-Link, 41

C

CC3000, 27, 32, 54
CLK, 29

COM (wspólny), 83
cURL, 36
czujnik
 ACS712, 67, 72
 DHT11 z rezystorem, 41, 54
 DHT22, 45
 PIR, 14, 17
 pomiarowy prądu, 67
 ruchu, 13, 14

D

delay(), 96
DHT, 42
DHT11, 41, 54
DHT22, 45, 54
dioda LED, 93, 95

E

EAGLE, 97

F

Fab Lab, 106
fotokomórka, 55
funkcja
 setup(), 18, 19
 delay(), 96
 parseJSON(), 21

G

graficzny interfejs, 13

H

HTML, 86

I

instrukcja `digitalRead()`, 18
internet przedmiotów, 53

J

JavaScript, 86
JeeLib, 93, 96
języku Python, 39
jQuery, 21

K

klucz API, 58
kondensator, 41, 92
konwerter USB dla modułu XBee, 14

L

`loop()`, 18

Ł

łączenie podzespółów, 55

M

mDNS, 33
MISO, 29
moduł
 Bluetooth Bluefruit EZ-Link, 41
 radiowy, 13
 Wi-Fi CC3000, 28, 54, 67
 XBee, 13, 14, 17
MOSI, 29
MySQL, 51

N

nakładka XBee, 14
NC (normalnie zwarty), 83
NO (normalnie otwarty), 83

O

OpenSCAD, 103
OSH Park, 101

P

`parseJSON()`, 21
PCB — printed circuit board, 93
Personal Area Network, 20
pętla `loop()`, 18
PHP, 86
pin
 CLK, 29
 GND, 29, 43
 IRQ, 29
 MISO, 29
 MOSI, 29
 SIG, 30
płytki
 Arduino Uno R3, 14, 41, 54, 67
 Arduino Uno, 14, 23, 81, 92
 Arduino, 17, 18
 drukowana, 27, 93
 stykowa, 28, 41, 55
 TinyDuino, 82
 TinyShield USB & ICP, 82
Pololu, 26, 81, 82
położenie
 DLINE, 16
 UART, 20
program EAGLE, 97
program OpenSCAD, 103
protokół ZigBee, 13
przełącznik Pololu 5V, 28
przewody połączeniowe, 28, 41
pull up, 41
Python, 39, 42, 48

R

REpresentational State Transfer, 19
 rezonator kwarcowy, 93
 rezystor, 93
 podciągający, 41

S

SaaS, 19
 Sculpteo, 106
 Seeedstudio, 101
 setup(), 18, 19
 Shapeways, 105
 SIG, 30
 Software as a Service, 19
 Sparkfun, 14
 stacja pogodowa, 53

T

testowanie działania modułu, 13
 ThingSpeak, 53, 54, 58
 TinyCircuits, 81
 TinyDuino, 82
 TinyShield, 82
 transfer stanu reprezentacji, 19

U

UART, 20
 Uteadstudio, 101

V

VBEN, 29
 VIN, 29

W

Wi-Fi CC3000, 67

X

XBee, 13, 14, 17, 92

Z

zacisk
 COM (wspólny), 30
 NC (normalnie zwarty), 30
 NO (normalnie otwarty), 30
 ZigBee, 13

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ 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 WYDAWNICZA



Helion SA

Arduino

Automatyka domowa dla każdego

Arduino to platforma, dzięki której budowa zaawansowanych układów elektronicznych stała się bardzo prosta i przyjemna. Już nie musisz tworzyć skomplikowanych projektów układów, wykonywać mnóstwa obliczeń ani rozwiązywać problemów. Wystarczy, że zdobędziesz płytkę Arduino i zaczniesz tworzyć układ Twoich marzeń. Jeżeli pragniesz mieszkać w inteligentnym domu i masz ochotę osiągnąć ten cel we własnym zakresie, wiedz, że dzięki tej książce i platformie Arduino jest to możliwe!

Sięgnij po ten poradnik i zobacz, jak zrealizować najróżniejsze praktyczne projekty. Bezprzewodowy detektor ruchu, termometr lub licznik zużytej energii to tylko niektóre z nich. Jeżeli chciałbyś sterować oświetleniem budynku za pomocą urządzeń mobilnych lub zbudować własną stację pogodową, także znajdziesz tu coś dla siebie, ponieważ książka zawiera przejrzyste opisy realizacji takich projektów. Na sam koniec dowiesz się, jak przerabiać urządzenia fabryczne, testować system oparty na Arduino oraz przygotować obudowę Twojego urządzenia. Książka ta jest doskonałą lekturą dla wszystkich pasjonatów elektroniki chcących zrealizować nowe, fascynujące projekty!

Twoje mieszkanie też może być inteligentne!



Dzięki tej książce:

- poznasz niesamowite zastosowania platformy Arduino
- zbudujesz bezprzewodowy detektor ruchu
- wykonasz pomiar temperatury
- zbudujesz własną stację pogodową
- opracujesz budowę dla stworzonego układu

Marco Schwartz — jest pisarzem i przedsiębiorcą. Pasjonują go elektronika i komputery. Ma kilkuletnie doświadczenie w branży elektrycznej. Interesuje się automatyką domową, platformą Arduino oraz Raspberry Pi, a także drukowaniem 3D.

[PACKT] open source
PUBLISHING community experience distilled

Helion

35062 numer katalogowy

księgarnia internetowa

<http://helion.pl>

zamówienia telefoniczne



0 801 339900



0 601 339900

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/novosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYSCI

ISBN 978-83-283-0936-4



Informatyka w najlepszym wydaniu

cena: 32,90 zł