



**Szalony
Geniusz**

SIMON MONK

**ARDUINO
I ANDROID
NIESAMOWITE
PROJEKTY**

Poznaj fascynujący
świat elektroniki!

Tytuł oryginału: Arduino™ + Android™ for the Evil Genius™:
Control Arduino with Your Smartphone or Tablet

Tłumaczenie: Andrzej Watrak
Projekt okładki: Studio Gravite/Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

ISBN: 978-83-246-8708-4

Original edition copyright © 2012 by The McGraw-Hill Companies.
All rights reserved.

Polish edition copyright © 2014 by HELION SA.
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 bierze jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Wydawnictwo HELION nie ponosi również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock Images LLC.

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/nparan.zip>

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
<http://helion.pl/user/opinie/nparan>
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	7
Podziękowania	7
Wprowadzenie	9

CZĘŚĆ I URZĄDZENIA PERYFERYJNE ANDROID

1. Robot sterowany sygnałem Bluetooth	15
Budowa	15
Teoria	24
Podsumowanie	26
2. Licznik Geigera	27
Otwarte akcesorium Google	27
Budowa	28
Teoria	38
Podsumowanie	44
3. Świetlny pokaz	45
Budowa: bazowe akcesorium Droid	45
Wykorzystanie projektu	55
Teoria	56
Podsumowanie	60
4. Pilot telewizyjny	61
Budowa	61
Wykorzystanie projektu	65
Teoria	65
Podsumowanie	66
5. Rejestrator temperatury	67
Budowa	67
Wykorzystanie projektu	71
Teoria	71
Podsumowanie	73
6. Dalmierz ultradźwiękowy	75
Budowa	76
Wykorzystanie projektu	79
Teoria	80
Podsumowanie	81

CZĘŚĆ II AUTOMATYKA DOMOWA

7. Sterownik automatyki domowej	85
Moduł interfejsu dźwiękowego	87
Oprogramowanie Android	97
Dostęp przez internet	101
Teoria	104
Podsumowanie	110

8.	Sterowanie zasilaniem	111
	Układ sterowania zasilaniem	111
	Budowa modułu sterowania zasilaniem	112
	Dołączanie modułu do sterownika automatyki domowej	119
	Skonfiguruj swój dom	121
	Teoria	122
	Podsumowanie	127
9.	Inteligentny termostat	129
	Budowa	129
	Korzystanie z systemu	140
	Teoria	140
	Podsumowanie	144
10.	Zamek otwierany identyfikatorem RFID	145
	Budowa	145
	Użytkowanie systemu	155
	Teoria	155
	Podsumowanie	161
11.	Flagi sygnalizacyjne	163
	Budowa	163
	Teoria	168
	Podsumowanie	169
12.	Wyłącznik czasowy	171
	Budowa	171
	Teoria	177
	Podsumowanie	181
A	Podręcznik Open Accessory	183
	Nauka programowania w systemie Android	183
	Programowanie Arduino	183
	Przykład	183
	O płycie Arduino	184
	Android	186
	Wnioski	192
	Skorowidz	193

Inteligentny termostat

Służący Szalonego Geniusza są bardzo wrażliwi na temperaturę. Gdy jest im zbyt zimno, przeziębają się, sinieją i nie są w stanie spełniać szalonych zachcianek swojego pana. Jeżeli znów jest za gorąco, bardzo się pocą i chlapią na jego plany dominacji nad światem.

Ten projekt w dziedzinie regulacji temperatury nie ma sobie równych. Nie tylko działa jako niezależny termostat, ale posiada również łączność radiową ze sterownikiem automatyki domowej, który może ustawiać różną temperaturę w różnych porach dnia.

Sterownik będzie co 10 sekund przysyłał do termostatu drogą radiową nowe dane o temperaturze. Abyś miał pewność, że wszystko działa poprawnie, dioda LED będzie dwukrotnie migać po odebraniu każdego komunikatu.

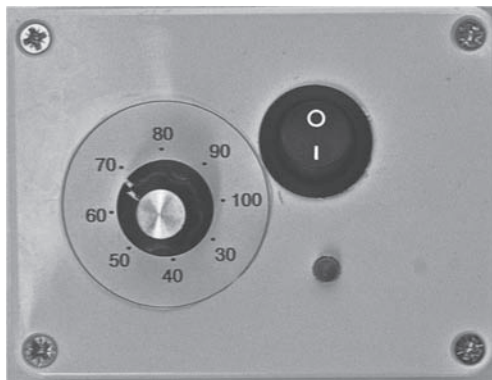
W celu zmniejszenia wymiarów i kosztów projektu zastosowany jest w nim mikrokontroler wyjęty z Arduino i umieszczony na płycie prototypowej. Płyta Arduino będzie użyta do zaprogramowania mikrokontrolera, po czym zostanie on zastąpiony innym układem. Układ ATmega328 zastosowany w Arduino jest powszechnie dostępny i kosztuje kilkanaście złotych.

Rysunek 9.1 przedstawia termostat wyposażony w wyłącznik trybu nadrzędnego, pokrętkę regulacyjną i diodę LED.

Główna część oprogramowania termostatu domowego znajduje się w sterowniku automatyki domowej, który udostępni przyjemny dotykowy interfejs użytkownika, umożliwiając ustawienie profilu temperatury (patrz rysunek 9.2). Dostępne są dwie strony, na których ustawia się temperaturę, jedna dla dni roboczych i druga dla weekendów. Dzień jest podzielony na przedziały czasu.

Na przykład 01-04 oznacza okres czasu od godziny 1:00 do 4:00. Temperatura dla każdego okresu jest ustawiana przy użyciu rozwijanej listy.

Jeżeli włączysz główny wyłącznik na termostacie, wówczas ustawiona będzie nie temperatura określona przez sterownik, ale ta ustawiona za pomocą pokrętki.



RYСУNEK 9.1. Inteligentny termostat systemu ogrzewania (tutaj temperatura w skali Fahrenheita)

Budowa

Musisz zbudować sam termostat, jak również zmodyfikować sterownik automatyki domowej, wyposażając go w nadajnik radiowy i aktualizując oprogramowanie w Arduino.

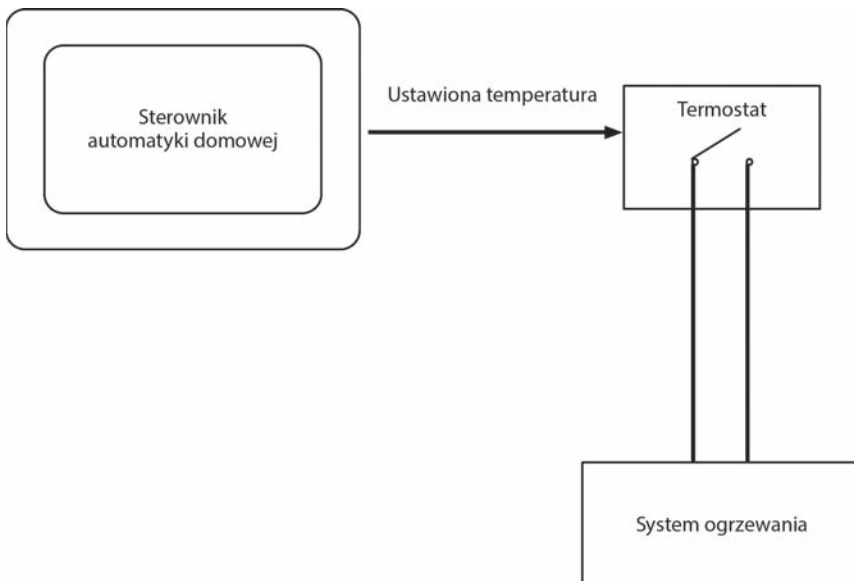
Najlepiej jest przygotować w pierwszej kolejności termostat, dzięki czemu będziesz mógł go użyć do sprawdzenia, czy zmiany w sterowniku są wprowadzone poprawnie.

Oprócz mikrokontrolera i kilku dodatkowych elementów na płycie prototypowej zostanie umieszczony tranzystor, przekaźnik i odbiornik radiowy. Do zasilania zostanie użyty stary zasilacz 5 V z ładowarki do telefonu komórkowego. Jeżeli nie masz takiej ładowarki, możesz użyć zwykłego zasilacza.



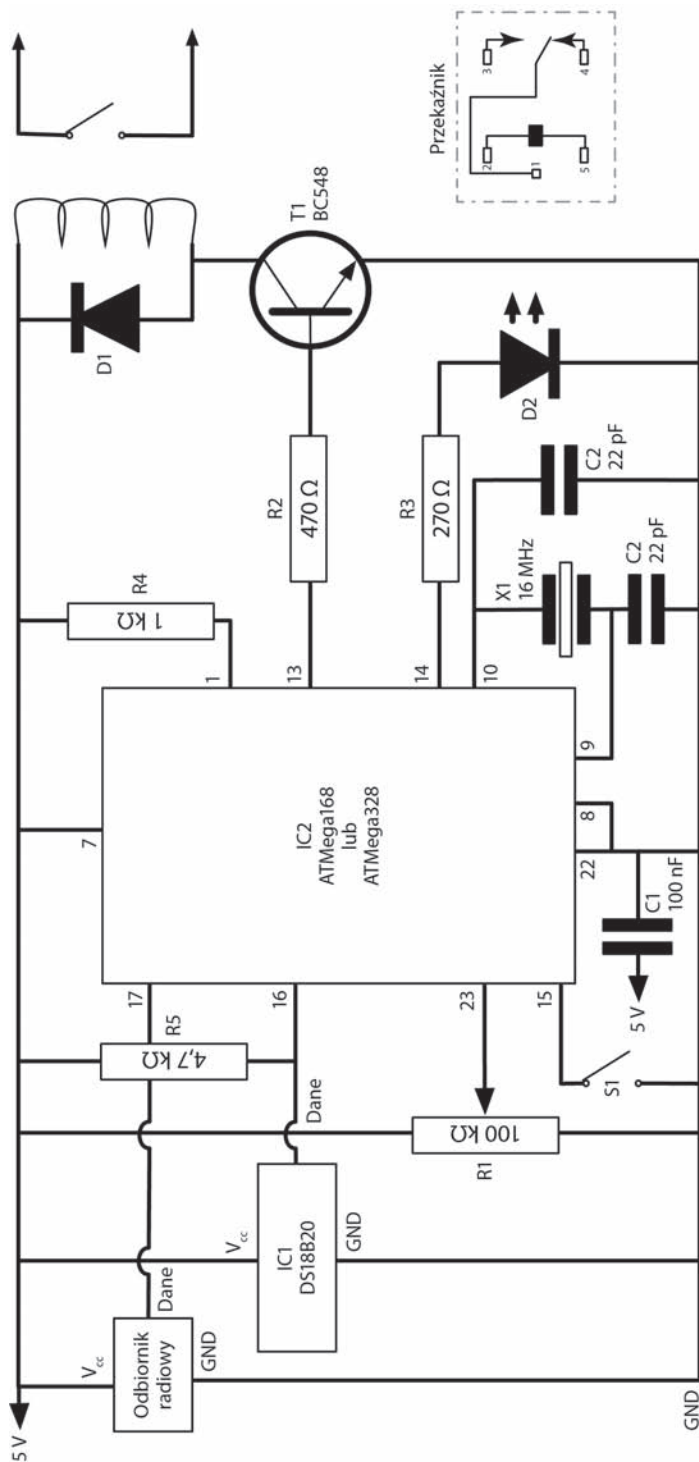
RYСУNEK 9.2. Sterownik automatyki domowej — regulacja temperatury

Rysunek 9.3 przedstawia zależności pomiędzy termostatem, systemem centralnego ogrzewania i sterownikiem automatyki domowej.



RYСУNEK 9.3. Sterownik automatyki domowej i termostat

Na rysunku 9.4 pokazany jest schemat ideowy termostatu.



RYСУNEK 9.4. Schemat ideowy termostatu

UWAGA: 

W tym projekcie przełączane jest napięcie domowej instalacji elektrycznej. Wymagana jest również modyfikacja układu elektrycznego w systemie ogrzewania. Napięcie w sieci energetycznej jest niebezpieczne dla życia i zdrowia, może być również przyczyną pożaru. Możesz samodzielnie zainstalować ten system, jeżeli posiadasz odpowiednie kwalifikacje i jesteś świadomy wpływu zmian wprowadzonych w instalacji na bezpieczeństwo mieszkańców.

CO BĘDZIE CI POTRZEBNE

Do wykonania termostatu będziesz potrzebować elementów wymienionych na liście materiałów.

LISTA MATERIAŁÓW

Element	Ilość	Opis	Zakup
Mikrokontroler	1	Układ ATmega328 z programem rozruchowym	Sparkfun: DEV-10524
Odbiornik radiowy	1	Moduł odbiornika radiowego 433 MHz	Farnell: 1304026
Nadajnik radiowy	1	Moduł nadajnika radiowego 433 MHz	Farnell: 1304024
Przełącznik	1	Przełącznik 5 V lub 6 V	Farnell: 1455502
R1	1	Potencjometr 100 kΩ	
Pokrętko	1	Pokrętko do potencjometru	Farnell: 1282303
R2	1	Rezystor metalizowany 470 Ω, 0,5 W	Farnell: 1099883
R3	1	Rezystor metalizowany 270 Ω, 0,5 W	Farnell: 9340300
R4	1	Rezystor metalizowany 1 kΩ, 0,5 W	Farnell: 9339779
R5	1	Rezystor metalizowany 4,7 kΩ, 0,5 W	Farnell: 9340629
C1	1	Kondensator ceramiczny 100 nF	Farnell: 1200414
C2, C3	2	Kondensator ceramiczny 22 pF	Farnell: 1600966
X1	1	Oscylator kryształowy 16 MHz	Farnell: 1611761
D1	1	Dioda 1N4001	Farnell: 1458986
D2	1	Czerwona dioda LED 5 mm	Farnell: 1712786
IC1	1	Układ scalony DS18B20	Sparkfun: SEN-00245
T1	1	Tranzystor BC548	Farnell: 1467872
S1	1	Wyłącznik kołkowy	Farnell: 1634645
Gniazdo	1	Gniazdo układu scalonego DIP na 28 pinów	Farnell: 1824463
Zasilacz	1	Zasilacz 5 V	Patrz opis
Zaciski	1	Dwustronne zaciski elektryczne 2 A	Sklep z częściami zamiennymi
Płyta prototypowa	1	Płyta prototypowa o wymiarach 29 pasków po 24 otwory	Farnell: 1201473
Obudowa	1	Mała obudowa projektowa	Sklep z elementami elektronicznymi

Do zaprogramowania układu ATmega328 lub ATmega168 będzie również potrzebna płyta Arduino Uno lub Duemilanove. Przy zakupie płyty upewnij się, że jest to wersja Arduino z zainstalowanym programem rozruchowym (bootloaderem).

W sprzedaży jest dostępnych wiele modułów radiowych pracujących w paśmie 433 MHz i większość z nich ma taki sam prosty w użyciu rozkład pinów. Porównaj jednak dane z dokumentacji modułu, którego zamierzasz użyć, z rysunkiem 9.4 i upewnij się, że układ pinów jest taki sam.

Przełącznik powinien być dostosowany do zasilania napięciem 5 V lub 6 V. Styki muszą mieć podobną specyfikację jak w oryginalnym termostacie, który zostanie zamieniony. Jeżeli nie znasz parametrów napięcia i natężenia prądu przełącznika, zapytaj o to fachowca od systemu ogrzewania. Zwróć uwagę, że wymieniony przełącznik jest przystosowany do systemu ogrzewania używanego przez autora. Autor nie dysponuje informacjami, czy przełącznik nadaje się dla Twojego systemu ogrzewania, dlatego musisz sam to sprawdzić.

Służący Szalonego Geniusza zużywają telefony komórkowe w takim tempie, jakby świat miał się zaraz skończyć. W kryjówce zawsze znajdzie się stary zasilacz do nieużywanego od dawna telefonu. Oczywiście Szalony Geniusz chce mieć swój udział w ochronie planety, tuż przed jej podbojem.

Do naszych celów zostanie użyta taka właśnie ładowarka, dostarczająca napięcie 5 V do termostatu. Jeżeli jednak takiej nie posiadasz, możesz kupić zwykły zasilacz 5 V.

Oprócz powyższych elementów będziesz potrzebować narzędzi wymienionych w tabeli.

☰ NARZĘDZIA ☰

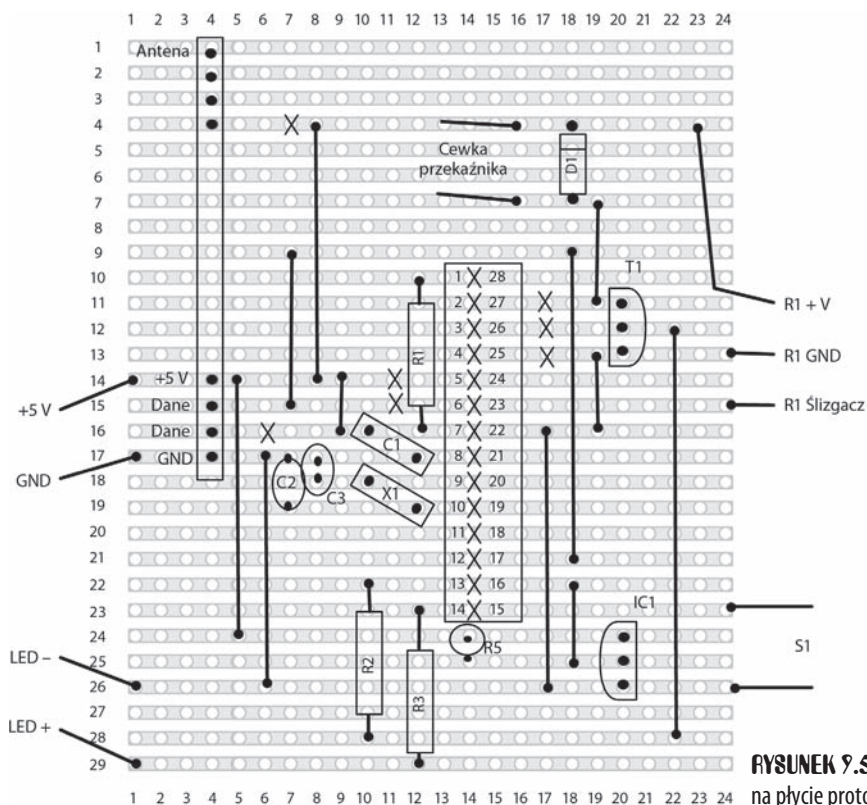
- zestaw do lutowania,
- woltomierz,
- nóż z wymiennym ostrzem i linijka,
- multimetr.
- pistolet do klejenia na gorąco lub klej epoksydowy,

➤ KROK 1. PRZYGOTUJ PŁYTKĘ PROTOTYPOWĄ

Rysunek 9.5 przedstawia układ elementów termostatu na płycie prototypowej.

Wytnij odpowiedniej wielkości fragment płyty prototypowej. Najlepiej jest naciąć mocno płytę nożem z wymiennym ostrzem i przełamać ją na krawędzi stołu wzdłuż nacięcia. Możesz również użyć solidnych nożyczek, ale efekt nie będzie równie estetyczny.

Jak pokazuje rysunek, na płycie trzeba zrobić kilka przerw. Podobnie jak w przypadku interfejsu dźwiękowego z rozdziału 7., do zrobienia przerw w oznaczonych miejscach użyj wiertła. Zwróć uwagę, że rysunek 9.5 przedstawia wierzchnią stronę płyty.



Po wykonaniu 19 przerw płyta powinna wyglądać jak na rysunku 9.6.

◆ KROK 2. PRZYLUTUJ POŁĄCZENIA I GNIAZDO UKŁADU SCALONEGO

Następnym krokiem jest przylutowanie połączeń do płyty (patrz rysunek 9.7). Tutaj również jest ich sporo (jedenaście), dlatego sprawdź dokładnie, czy wykonałeś wszystkie i we właściwych miejscach.

◆ KROK 3. PRZYLUTUJ REZYSTORY I DIODĘ

Teraz możesz umieścić na płycie właściwe elementy. Jak zwykle rozpocznij od najniższych, czyli rezystorów i diody. Następnie przylutuj kondensatory. Podczas lutowania diody zwróć uwagę, czy umieściłeś ją we właściwym położeniu. Końcówka oznaczona paskiem powinna być skierowana ku dolnemu brzegowi płyty.

Rezystor R5 będzie zamontowany pionowo, dlatego pozostaw go do następnego kroku, gdy będziesz lutować kondensatory i wyższe elementy.

Po przylutowaniu wymienionych elementów możesz zająć się gniazdem układu scalonego. Dobrze jest umieścić je tak, aby niewielkie nacięcie oznaczające pin nr 1 było skierowane ku górnemu brzegowi płyty.

Rysunek 9.8 przedstawia płytę z umieszczonymi na niej elementami.

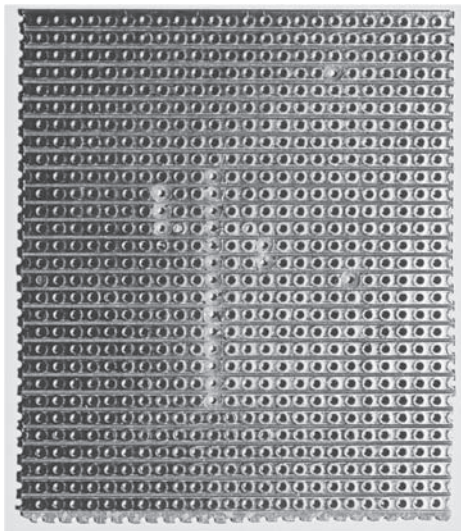
◆ KROK 4. PRZYLUTUJ POZOSTAŁE ELEMENTY

Teraz możesz umieścić na płycie pozostałe elementy. Ostrożnie obchodź się z układem scalonym IC1 i tranzystorem T1, nie podgrzewaj ich końcówek zbyt długo podczas lutowania, jak również upewnij się, że umieściłeś je we właściwym położeniu.

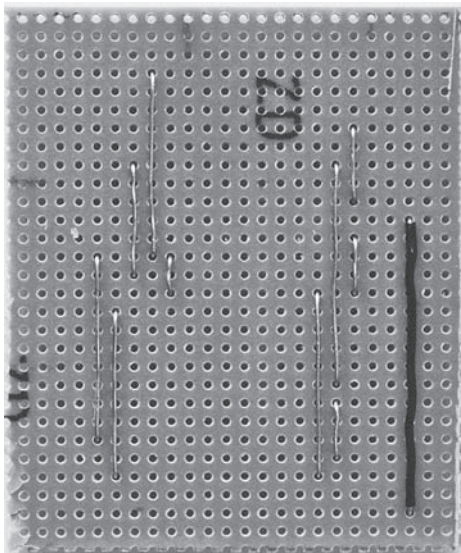
W module odbiornika radiowego wykorzystane są tylko cztery dolne piny, ale jeżeli termostat będzie bardzo oddalony od sterownika automatyki domowej, powinieneś przylutować do pinu ANT przewód o długości ok 100 mm, który będzie odgrywał rolę anteny.

Po połączeniu wszystkich elementów płyta powinna wyglądać jak na rysunku 9.9.

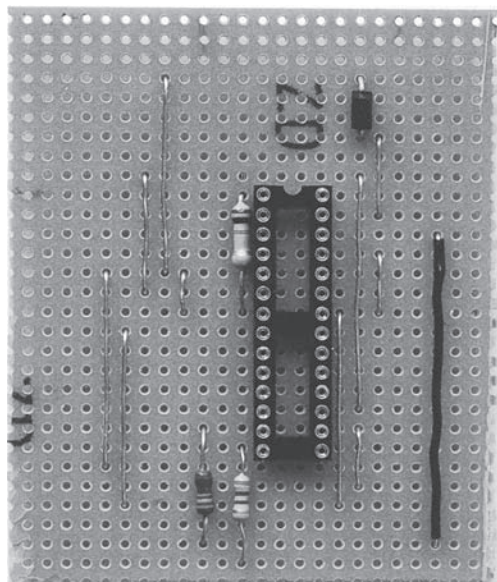
W tym momencie dobrze jest sprawdzić całą płytę. Obejrzyj jej wierzchnią stronę i upewnij się, że wszystkie elementy i połączenia znajdują się na swoich miejscach, zgodnie z rysunkiem 9.5. Sprawdź spodnią stronę płyty, w szczególności czy przecięcia są wykonane we właściwych miejscach i czy nie ma przypadkowych mostków lutowniczych pomiędzy ścieżkami.



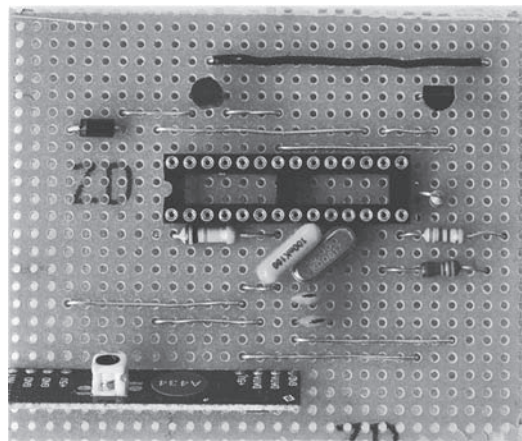
RYСУNEK 9.6. Przygotowana płyta prototypowa



RYСУNEK 9.7. Płyta prototypowa z połączeniami



RYСУNEK 9.8. Płyta prototypowa z umieszczonymi na niej rezystorami i kondensatorami



RYСУNEK 9.9. Gotowa płyta prototypowa

▶ KROK 5. ZAPROGRAMUJ I ZAINSTALUJ MIKROKONTROLER

W mikrokontrolerze sterującym termostatem musisz umieścić szkielet Arduino. Najłatwiej jest użyć w tym celu płyty Arduino.

Zazwyczaj programuje się mikrokontroler umieszczony na płycie Arduino i po przeniesieniu go na płytę prototypową zastępuje go kupionym nowym układem.

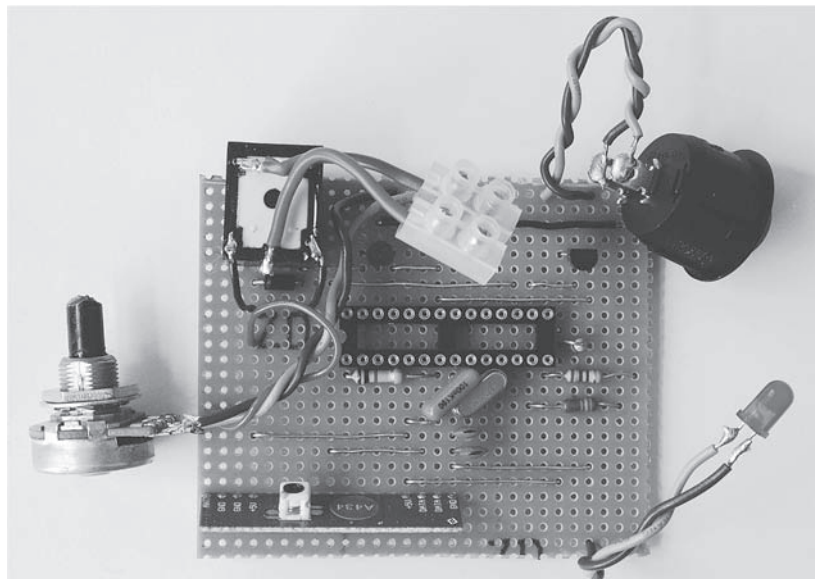
Szkielet wykorzystuje trzy biblioteki, dwie do czujnika temperatury (*OneWire* oraz *DallasTemperature*) oraz *VirtualWire* do odbiornika radiowego. Tę ostatnią bibliotekę prawdopodobnie już zainstalowałeś w rozdziale 8. Musisz jednak w środowisku Arduino na swoim komputerze zainstalować dwie pierwsze biblioteki. Procedura jest taka sama jak dla każdej innej biblioteki (patrz rozdział 1.), a potrzebne pliki możesz pobrać pod adresami http://www.pjrc.com/teensy/td_libs_OneWire.html oraz http://www.milesburton.com/?title=Dallas_Temperature_Control_Library.

Zainstaluj szkielet *ch09_thermostat* w Arduino. Następnie odłącz zasilanie i wyjmij ostrożnie mikrokontroler z płyty i włóż do gniazda na płycie prototypowej. Upewnij się, że umieściłeś go we właściwym położeniu.

Bądź ostrożny podczas wyjmowania mikrokontrolera, ponieważ łatwo można zgąć i złamać piny. Używając noża, podważ nieznacznie układ kilka razy z obu stron, aż wyjdzie całkowicie z gniazda. Pamiętaj też o zaleceniach antystatycznych. Zanim weźmiesz układ w dłoń, dotknij uziemionego metalowego przedmiotu, aby spłynął z Ciebie ładunek elektryczny. Inną opcją jest założenie na rękę uziemionej opaski.

▶ KROK 6. POŁĄCZ WSZYSTKIE KOMPONENTY

Płyta jest gotowa, więc po szczegółowym sprawdzeniu, czy nie powstały niepożądane mostki lutownicze pomiędzy ścieżkami, możesz połączyć wszystkie komponenty (patrz rysunek 9.10). Nie umieszczaj ich jednak w obudowie, dopóki nie sprawdzisz, czy wszystko działa prawidłowo.

**RYСУNEK 9.10.**

Połączone komponenty na płycie

Jako zasilacza użyj niepotrzebnej ładowarki do telefonu komórkowego. Nie ma potrzeby stosowania porządnego zasilacza. Wystarczy, że będzie dostarczał napięcie 5 V i prąd o natężeniu 100 mA. Przy odłączonej ładowarce odetnij wtyk na końcu przewodu i zdejmij z ich końców izolację, abyś mógł podłączyć multimetr i sprawdzić napięcie. Najpierw sprawdź, czy jest to rzeczywiście 5 V, a ponadto która końcówka jest dodatnia. Potem możesz przylutować przewody bezpośrednio do płyty.

Następnie przymocuj przewody do potencjometru. Zwróć uwagę, że kolejność przewodów na płycie łączących potencjometr jest inna, niż się spodziewałeś. Patrząc na potencjometr od dołu z końcówkami skierowanymi w dół, dodatnia końcówka znajduje się z lewej strony, ujemna z prawej, a ślizgacz w środku.

Wyłącznik i dioda LED powinny być podłączone za pomocą krótkich przewodów wielożyłowych. Aby uniknąć przypadkowej pomyłki w polaryzacji diody, do jej podłączenia dobrze jest użyć przewodów w różnych kolorach.

Do cewki przekaźnika przewody mogą być dowolnie podłączone. Przyklej przekaźnik do płyty i dołącz przewody zasilające cewkę. Przylutuj następnie przewody do rozwartych w normalnym stanie styków i dołącz je z drugiej strony do zacisków elektrycznych. Wiele przekaźników ma układ końcówek przedstawiony na rysunku 9.4, niemniej jednak porównaj go ze swoim przekaźnikiem.

Ustaw wyłącznik w nadrzędnym trybie regulacji temperatury (załączenia) i podłącz zasilanie. Podczas obracania potencjometru zgodnie z ruchem wskazówek zegara powinna zapalić się dioda i załączyć przekaźnik. W trybie nadrzędnym temperatura jest ustawiana za pomocą potencjometru, a więc ustawienie wyższej temperatury niż mierzona przez czujnik powinno skutkować włączeniem ogrzewania.

Możesz zrobić eksperyment i ustawić potencjometr w położeniu, w którym zapali się dioda, a następnie dotknąć palcem czujnika temperatury, aby go podgrzać. Gdy temperatura przekroczy ustawioną wartość, dioda powinna zgasnąć.

To jest wszystko, co możemy sprawdzić przed modyfikacją sterownika automatyki domowej.

◆KROK 7. ZMODYFIKUJ STEROWNIK AUTOMATYKI DOMOWEJ

Do płyty perforowanej sterownika automatyki domowej musisz dodać nadajnik radiowy. Rysunek 9.11 przedstawia nowy układ elementów na płycie.

Moduł nadajnika radiowego jest mniejszy niż moduł odbiornika i jest umieszczony obok płyty interfejsu dźwiękowego. Pin danych w nadajniku jest dołączony do pinu D12 Arduino. Połączone są ze sobą piny masy (GND), a zasilanie +5 V z pinem Vcc. Podobnie jak w odbiorniku możesz do końcówki ANT nadajnika podłączyć przewód, który będzie odgrywał rolę anteny, aczkolwiek w przypadku niewielkich odległości nie jest to konieczne.

Rysunki 9.12 i 9.13 przedstawiają odpowiednio wierzchnią i spodnią stronę płyty perforowanej.

Szkic załadowany na początku do Arduino zawiera kod obsługujący nadajnik radiowy, nie trzeba więc go aktualizować.

◆KROK 8. TEST

Teraz możesz przetestować układ razem ze sterownikiem automatyki domowej. Włącz zasilanie termostatu i kontrolera, przy czym wyłącznik na termostacie ustaw w pozycji rozłączonej. Po około 10 sekundach dwukrotnie powinna zamigać dioda LED, co będzie oznaczać, że sterownik automatyki domowej wysłał do termostatu ustawienie temperatury.

Jeżeli nie jest wyjątkowo zimno, dioda LED i przekaźnik powinny być wyłączone, ponieważ domyślnie będzie ustawiona temperatura 0°C.

Otwórz w sterowniku stronę *Ogrzewanie pon.-pt.* i ustaw dla bieżącego przedziału czasu maksymalną możliwą temperaturę (patrz rysunek 9.17). Po chwili powinna zacząć migać dioda na termostacie i powinien się włączyć przekaźnik.

Jeżeli dioda nie miga, sprawdź zmiany wprowadzone w połączeniach w sterowniku i termostacie, szczególnie w pobliżu modułów radiowych. Upewnij się, że napięcie na pinach zasilania jest równe 5 V.

Aby sprawdzić, czy przekaźnik działa poprawnie, przełącz multimetr w tryb przewodzenia i dotknij zacisków jego końcówkami. Gdy przekaźnik załączy się i zapali się dioda LED, wówczas multimetr powinien wydać sygnał dźwiękowy.

◆KROK 9. UMIEŚĆ UKŁAD W OBUDOWIE

Twój termostat zostanie użyty w miejscu wbudowanego termostatu. Dlatego idealna będzie obudowa tej samej wielkości lub nieco większa, która pomieści płytę prototypową i inne elementy.

Zasilanie można zrealizować na kilka sposobów. Jeżeli obecny termostat posiada przewody zerowy i gorący, możesz je doprowadzić do obudowy. W przeciwnym wypadku będzie potrzebne gniazdko elektryczne w pobliżu termostatu.

Autor użył obudowy na tyle dużej, aby można było łatwo i dogodnie rozmieścić w niej wszystkie komponenty.

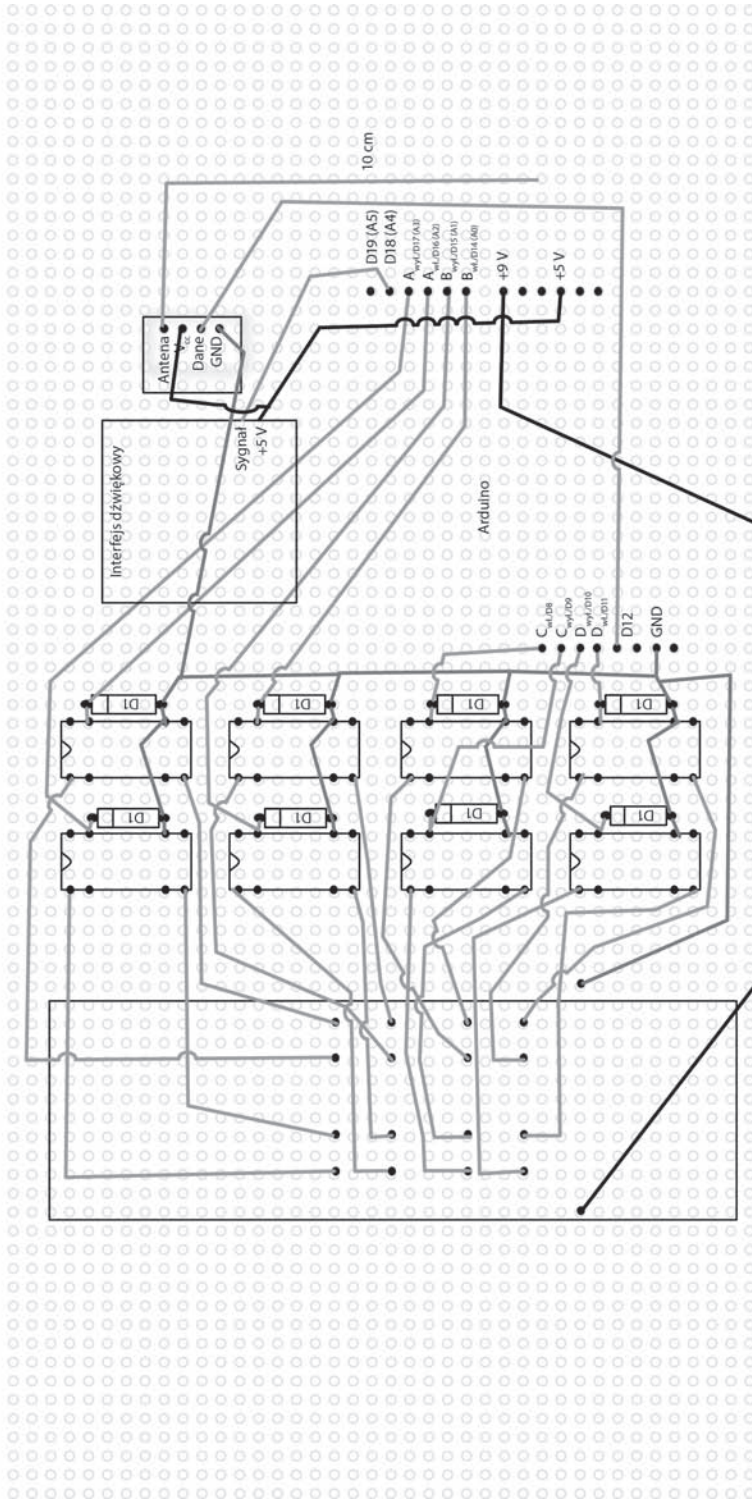
Gdy będziesz pewien, gdzie będzie umieszczony każdy komponent, zaznacz na obudowie miejsca na otwory.

Rysunek 9.14 przedstawia komponenty rozmieszczone w obudowie, natomiast na rysunku 9.15 pokazany jest przedni panel z wykonanymi otworami.

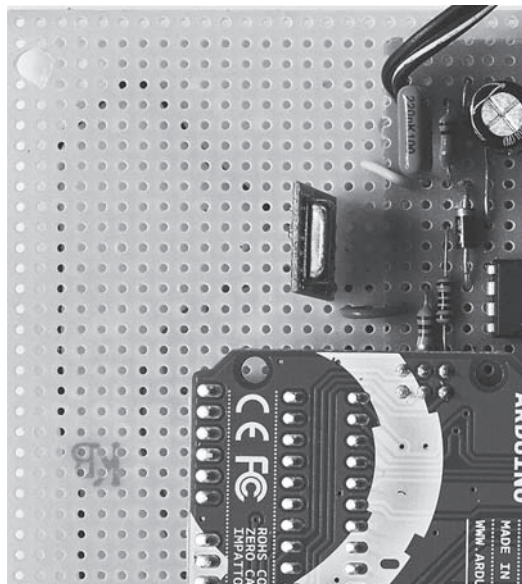
Dioda LED jest umieszczona na wcisk w otworze o średnicy 5 mm, który będzie ją utrzymywał we właściwym położeniu.

Musisz również wywiercić otwór na przewód zasilający oraz kilka dodatkowych otworów w tylnej ścianie obudowy do doprowadzenia przewodów z systemu ogrzewania i do przytwierdzenia obudowy do ściany.

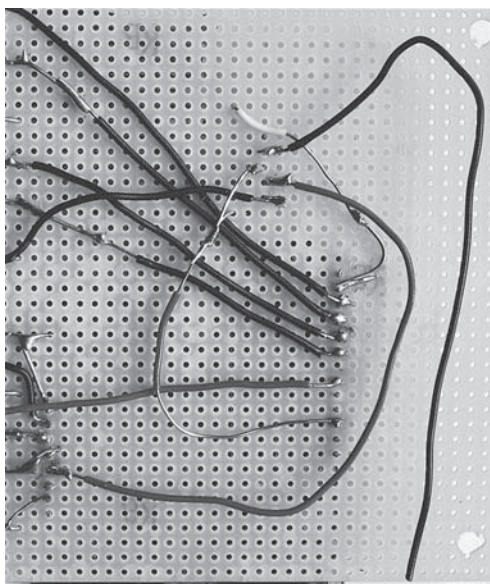
Widok spodniej strony płyty



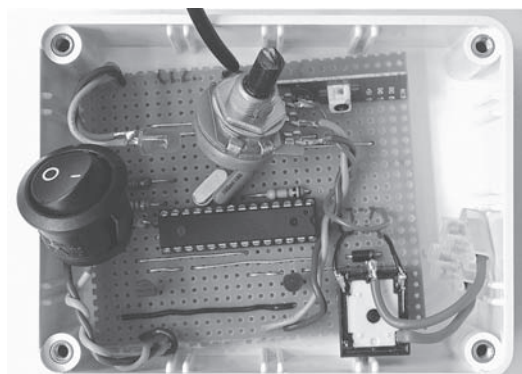
RYСУNEK 9.11. Zmodyfikowany układ elementów na płycie perforowanej



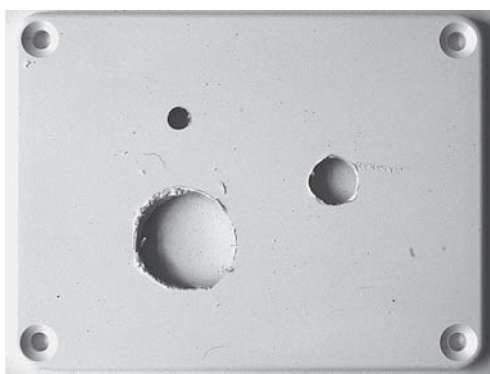
RYSUNEK 9.12. Wierzchnia strona płyty perforowanej



RYSUNEK 9.13. Spodnia strona płyty perforowanej



RYSUNEK 9.14. Rozmieszczenie komponentów



RYSUNEK 9.15. Przedni panel z wykonanymi otworami

Otwory do zamocowania obudowy powinny dokładnie zgadzać się z otworami w ścianie mocującymi obecny termostat.

Na swoim termostacie powinieneś również umieścić kilka oznaczeń temperatury. Minimalna temperatura wynosi 0°C , a maksymalna 40°C . Zaznacz te pozycje i dodaj pomiędzy nimi jeszcze siedem równomiernie rozmieszczonych punktów oznaczających temperaturę 5, 10, 15, 20, 25, 30 i 35 stopni. Możesz te liczby zapisać flamastrem na obudowie lub nakleić gotowe cyfry, albo wydrukować na kartce papieru, wyciąć i przykleić.

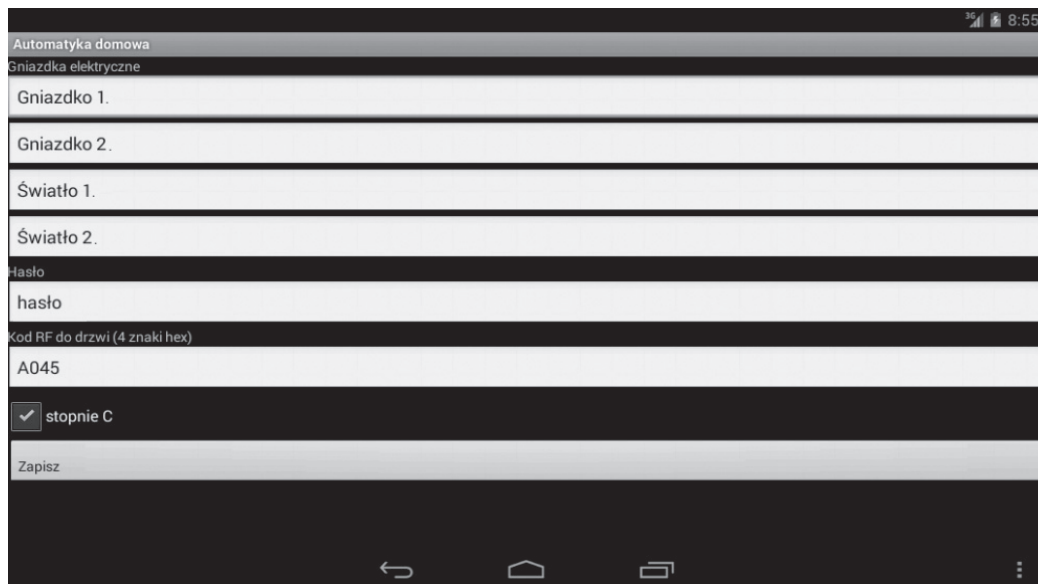
► KROK 10. INSTALACJA

Cały projekt sprowadza się do sterowania wyłącznikiem, który będzie zamykał obwód, gdy wymagane będzie włączenie ogrzewania. Tym wyłącznikiem jest przekaźnik, którego styki w normalnym stanie są rozłączone. Sposób podłączenia przekaźnika do instalacji centralnego ogrzewania zależy od Ciebie.

Jeszcze raz zwrócę uwagę, że podczas podłączania powinieneś zachować szczególną ostrożność. Jeżeli nie masz pewności, czy potrafiśz to zrobić sam, poproś o pomoc wykwalifikowanego fachowca od systemów centralnego ogrzewania.

Korzystanie z systemu

Na stronie ustawień sterownika automatyki domowej znajduje się opcja wyboru jednostek temperatury (patrz rysunek 9.16). Jeżeli zaznaczysz *stopnie C*, wówczas wartości temperatury pokazywane na sterowniku będą wyrażone w stopniach Celsjusza.



RYСУNEK 9.16. Wybór skali temperatury

Musisz zaznaczyć tę opcję, zanim zaczniesz definiować profil temperatury. Jeżeli nie zrobisz tego teraz, będziesz musiał później powtórzyć wszystkie czynności.

Szalony Geniusz obniża opłaty za ogrzewanie, ustawiając profil temperatury odpowiadający jego planom dnia. Nie ma więc potrzeby ogrzewania kryjówki, kiedy służący Szalonego Geniusza i on sam są poza domem i robią nieczne czynki. Ponieważ Szalony Geniusz w weekendy jest domatorem, na te dni potrzebny jest osobny harmonogram. Dlatego dostępne są osobne opcje od poniedziałku do piątku oraz w weekend.

Rysunek 9.17 przedstawia typowy profil temperatury dla dni powszednich.

Na noc jest ustawiona minimalna temperatura (0°C), następnie w godzinach od 4:00 do 6:00 rośnie do 10°C, a dalej między 6:00 a 7:00 do 15°C. W godzinach 7:00 – 9:00, kiedy Szalony Geniusz dokonuje porannej toalety, temperatura jest ustawiona na 20°C.

Między godzinami 9:00 a 16:00, kiedy wszyscy są poza domem, temperatura jest zmniejszana do minimum. Następnie po południu rośnie ponownie, aż do końca wieczoru, gdy czas iść spać.

Teoria

Jest to standardowy projekt Arduino. Użytych jest w nim jednak kilka ciekawych elementów, czyli czujnik temperatury i moduły radiowe do komunikacji między dwiema płytami Arduino.



RYSUNEK 9.17. Profil temperatury

Omówimy teraz oba elementy i zobaczymy, jak korzysta z nich szkic Arduino.

GZUJNIKI ONE-WIRE

Czujnik temperatury zastosowany w tym projekcie jest bardzo interesującym urządzeniem. Łączy w sobie sam czujnik, jak również procesor wysyłający pomiar w formacie cyfrowym. Dzięki temu można go umieścić w pewnej odległości od Arduino bez obawy, że długość przewodów wpłynie na dokładność pomiaru temperatury.

Artykuł na stronie www.arduino.cc/playground/Learning/OneWire zawiera opis tych interesujących czujników.

SZKIC OBSŁUGUJĄCY TERMOSTAT

Poniżej przedstawiony jest szkic obsługujący termostat.

```
#define THERMOSTAT_ID 0x41
#define MIN_ON_TIME 120L // sekundy
#define RADIO_CHECK_PERIOD 10000L
#define MIN_TEMP 30
#define MAX_TEMP 100

#define rFRxPin 11
#define tempRxFPin 10
#define potPin 0
#define ledPin 8
#define relayPin 7
#define overrideSwitchPin 9

OneWire oneWire(tempRxFPin);
DallasTemperature sensors(&oneWire);
```

```

DeviceAddress thermometer;

int setTemp = 0;
int actualTemp = 0;
long lastCheckedRadio = 0;

void setup(void)
{
  pinMode(ledPin, OUTPUT);
  pinMode(relayPin, OUTPUT);
  pinMode(overrideSwitchPin, INPUT);
  digitalWrite(overrideSwitchPin, HIGH); // zasilenie rezystora podwyższającego R
  Serial.begin(9600);
  vw_set_ptt_pin(5); // poza zakresem
  vw_setup(2000);
  vw_rx_start();
  sensors.getAddress(thermometer, 0);
  sensors.begin();
  sensors.setResolution(thermometer, 10);
  Serial.println("Gotowy");
}

void loop()
{
  if (digitalRead(overrideSwitchPin) == LOW)
  {
    // tryb nadrz. wyłączony, odbierz temperaturę ze sterownika drogą radiową
    if (millis() > (lastCheckedRadio + RADIO_CHECK_PERIOD))
    {
      checkForMessage();
      lastCheckedRadio = millis();
    }
  }
  else
  {
    setTemp = readSetTemperature();
  }
  actualTemp = readTemperature();
  setPower();
  delay(500);
}

void checkForMessage()
{
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
  if (vw_get_message(buf, &buflen))
  {
    // Jest komunikat do odebrania
    // Ważne są tylko dwa pierwsze bajty
    // Pierwszy oznacza odbiorcę komunikatu,
    // a drugi temperaturę w stopniach F
    byte receiver = buf[0];
    byte payload = buf[1];
    if (receiver == THERMOSTAT_ID) // komunikat A
    {
      // ten komunikat jest dla mnie
      setTemp = payload;
      Serial.print("Temp. ustawiona radiem:: "); Serial.println(setTemp);
    }
  }
}

```

```

        flash(2);
    }
}

int readTemperature()
{
    sensors.requestTemperatures();
    float tempC = sensors.getTempC(thermometer);
    return (int)(DallasTemperature::toFahrenheit(tempC));
}

int readSetTemperature()
{
    int raw = analogRead(potPin);
    int t = map(raw, 0, 1023, MIN_TEMP, MAX_TEMP);
    return t;
}

void setPower()
{
    static boolean lastOnOff = false;
    static long powerLastChanged = 0;
    static int lastSetTemp = 0;
    boolean onOff = (actualTemp < setTemp);
    long t = millis();
    long t2 = powerLastChanged + MIN_ON_TIME * 1000L;
    boolean enoughTimeElapsed = (t > t2);
    boolean tempSettingChanged = (abs(setTemp - lastSetTemp) > 1);
    digitalWrite(ledPin, onOff);
    if ((onOff != lastOnOff) && (enoughTimeElapsed || tempSettingChanged))
    {
        Serial.print("ustawiona temp.: "); Serial.println(setTemp);
        Serial.print("zmierzona temp.: "); Serial.println(actualTemp);
        digitalWrite(relayPin, onOff);
        powerLastChanged = t;
        lastOnOff = onOff;
        lastSetTemp = setTemp;
    }
}

void flash(int n)
{
    for (int i = 0; i <= n; i++)
    {
        digitalWrite(ledPin, HIGH);
        delay(100);
        digitalWrite(ledPin, LOW);
        delay(100);
    }
}

```

Trzy instrukcje `include` na początku ładują biblioteki wykorzystywane w szkicu.

Stała `THERMOSTAT_ID` odzwierciedla kod aplikacji Android dla sterownika automatyki domowej i służy do rozpoznawania komunikatów wysyłanych do termostatu.

Dalej zdefiniowane są cztery stałe, które możesz zmienić odpowiednio do swoich potrzeb.

Stała `MIN_ON_TIME` oznacza minimalny czas, przez który będzie włączone lub wyłączone ogrzewanie. W ten sposób unika się niekorzystnego dla systemu ogrzewania „kołysania”, czyli częstego

włączania i wyłączania ogrzewania w momencie osiągnięcia zadanej temperatury. Czas jest ustawiony na dwie minuty.

Stała `RADIO_CHECK_PERIOD` oznacza czas pomiędzy sprawdzeniami, czy został wysłany sygnał ze sterownika automatyki domowej. Jest on ustawiony na 10 000 milisekund, czyli 10 sekund.

Stałe `MIN_TEMP` oraz `MAX_TEMP` określają zakres temperatur (w skali Fahrenheita). Zwróć uwagę, że skala Fahrenheita jest przyjęta w całym szkicu. Jeżeli używasz skali Celsjusza, nie modyfikuj kodu, zmień jedynie podziałkę na regulatorze.

Dalej zdefiniowane są piny wykorzystywane przez różne urządzenia dołączone do mikrokontrolera. Inicjowane są również biblioteki *OneWire* oraz *DallasTemperature*.

Funkcja `setup` przeprowadza standardową konfigurację pinów. Ciekawa sztuczka jest zrobiona z pinem `overrideSwitchPin`, który jest ustawiany w trybie `INPUT` (jako wejście), a następnie funkcja `digitalWrite` zmienia jego stan na `HIGH` (wysoki). W efekcie zasilony zostaje wewnętrzny rezystor podwyższający, dzięki czemu nie jest potrzebny osobny rezystor. Funkcja włącza również nasłuch sygnału radiowego i konfiguruje czujnik temperatury.

Funkcja `loop` zawiera dwa wątki kodu. Pierwszy jest wykonywany, gdy załączony jest wyłącznik nadrzędnej regulacji temperatury, a drugi, gdy wyłącznik jest rozłączony. Jeżeli wyłącznik **nie** jest załączony, sprawdzany jest sygnał z odbiornika radiowego, o ile upłynął ustawiony przedział czasu od chwili ostatniego sprawdzenia. W przeciwnym wypadku gdy wyłącznik jest załączony, ustawiona temperatura jest odczytywana z potencjometru.

W głównej pętli odczytywana jest rzeczywista temperatura i wywoływana jest funkcja `setPower` decydująca, czy przekaźnik ma być włączony, czy wyłączony.

Funkcja `checkForMessage` sprawdza, czy w buforze oczekuje komunikat z odbiornika radiowego. Jeżeli tak, funkcja dalej sprawdza, czy pierwszy bajt zawiera liczbę `0x41`. Oznacza ona, że komunikat faktycznie jest przeznaczony dla termostatu i kolejny bajt zawiera zadaną temperaturę wyrażoną w stopniach Fahrenheita. Przy odbiorze każdego komunikatu dwukrotnie miga dioda.

Funkcja `readTemperature` odczytuje temperaturę z czujnika i zamienia dane w skali Celsjusza na stopnie Fahrenheita.

Następna funkcja, `readSetTemperature`, zamienia odczyt z pinu analogowego, do którego dołączony jest potencjometr, na temperaturę w zakresie od `30°F` do `100°F` (tj. od ok. -1°C do 37°C), wykorzystując funkcję `map`. Ta wbudowana funkcja jest bardzo przydatna do konwersji odczytów z jednego zakresu na inny. Jej argumenty to liczba, która ma zostać zamieniona, pierwotny zakres wartości (w tym przypadku odczyt o wartości od 0 do 1023 z wejścia analogowego) i zakres wyjściowy (tutaj od 30 do 100).

Na końcu funkcja `setPower` zmienia stan przekaźnika, jeżeli zachodzi taka potrzeba. Wykorzystane są trzy statyczne zmienne, które zachowują swoje wartości po zakończeniu wykonywania funkcji. Zmienne pozwalają określić, czy stan przekaźnika został zmieniony, ile czasu upłynęło od ostatniej zmiany oraz jaka była ostatnio ustawiona temperatura. Stan przekaźnika jest zmieniany tylko wtedy, gdy jego poprzedni stan był inny, upłynął zadany okres czasu i ustawiona jest inna temperatura. Sprawdzenie ostatniego warunku dotyczącego zmiany temperatury umożliwia szybszą reakcję układu, gdy użytkownik zmieni ustawienia za pomocą potencjometru.

Podsumowanie

Teraz Szalony Geniusz może leżeć w łóżku i za pomocą telefonu Android włączać i wyłączać oświetlenie i gniazdka w swojej kryjówce, jak również sterować temperaturą.

W następnym i ostatnim poświęconym automatyce domowej rozdziale Szalony Geniusz unieвозмоли służącym opuszczanie domu. Zbuduje system zamka elektrycznego, sterowany zarówno sygnałem radiowym, jak i sterownikiem automatyki domowej.

Skorowidz

A

- acc.isConnected, 42, 59
 - acc.powerOn, 42
 - acc.read, 59
 - Accessory Development Kit, 27
 - action, 127
 - ACTION_USB_ACCESSORY_DETACHED, 188
 - activity, 186
 - addCode, 160
 - ADK, 27
 - program instalacyjny, 64, 78
 - współpraca Androida i Arduino, 27
 - ADK_release_0512, 184
 - alert, 192
 - Amarino, 10, 22, 23
 - analogWrite, 57
 - Android, 9, 10, 186
 - Amarino, 22
 - aplikacja testowa na urządzenie Android, 92
 - automatyczne uruchamianie i pobieranie aplikacji, 186
 - cykl życia aplikacji, 187
 - klient USB, 28
 - manifest projektu, 186
 - nauka programowania w systemie, 183
 - Nieznane źródła, 22
 - odbieranie danych, 191
 - odbiornik komunikatów rozgłoszeniowych, 189
 - otwarcie akcesorium, 188
 - tablet, 10
 - Ustawienia, 22
 - wysyłanie danych, 190
 - ądanie uruchomienia aplikacji, 187
 - Android Open Accessory, 10, 45
 - AndroidAccessory, 39, 59
 - AndroidManifest.xml, 42
 - API Keys, 71
 - aplikacje
 - Amarino, 23
 - Add BT Device, 23
 - Connect, 23
 - Ustawienia, 98, 99
 - Zamek w drzwiach, 100, 102
 - zasilanie i oświetlenie, 100
 - Dalmierz ultradźwiękowy, 78
 - do zdalnego sterowania robotem, 15
 - DroidDroid, 23
 - Licznik Geigera, 28, 36
 - Pilot telewizyjny, 64
 - Rejestrator temperatury, 70
 - Świetlny pokaz, 55
 - Test dźwięku, 93, 105
- application, 187
- Arduino, 9, 10, 16
 - bootloader, 132, 146
 - Duemilanove, 146
 - host USB, 28
 - instalacja biblioteki, 184
 - instalacja środowiska, 20
 - interfejs USB, 28
 - konfiguracja typu płyty, 21
 - moduł Bluetooth, 18
 - nakładki, 9
 - pin PWM, 57
 - pliki i foldery, 34
 - płyta, 184
 - programowanie, 183
 - przykładowa aplikacja, 184
 - szkic, 20, 185
 - wyjścia cyfrowe, 112
 - zalety, 15
 - zasilanie telefonu, 28
 - złącze hosta, 28
 - złącze klienta, 28
 - Arduino Mega, 183
 - Arduino Uno, 16, 30, 88, 146
 - robot sterowany sygnałem Bluetooth, 20
 - szkice, 93
 - ATMega168, 146
 - ATMega328, 45, 47, 62, 129, 146
- ## B
- baudRate, 25
 - baza Droid Duino, 10

- bazowe akcesorium Droid, 46
 - budowa, 45
 - instalacja szkieletu Droid Geiger, 50
 - lista materiałów, 47
 - lutowanie
 - gniazda, 47
 - oscylatora i innych elementów, 48
 - piny gniazda, 47
 - nakładka
 - z podłączonymi przewodami, 48
 - z zamocowanym gniazdem, 48
 - narzędzia, 47
 - piny
 - D2, 61, 68
 - D3, 50, 61
 - D5, 50, 61
 - D6, 50, 61, 68
 - IO, 45, 68
 - PWM, 45
 - plyta prototypowa, 49
 - podłączenie
 - przewodów do nakładki USB, 47
 - przewodów zasilających, 49
 - wyłącznika, 49
 - zasilania 5 V, 49, 50
 - potrzebne elementy, 47
 - pozostałe połączenia, 49
 - rozmieszczenie połączeń przewodowych, 49
 - schemat ideowy, 46
 - test, 50
 - beep, 106
 - Beeper, 105
 - bezpieczeństwo poprzez niezrozumiałość, 145
 - biblioteki
 - Amarino, 22
 - Android, 22
 - AndroidAccessory, 34, 39, 184
 - Arduino
 - DallasTemperature, 69, 135
 - EEPROM, 159
 - GPL, 60
 - hosta USB, 34, 59, 184
 - JPachube, 73
 - Max3124e, 41, 59
 - MeetAndroid, 22, 25
 - OneWire, 69, 135
 - Open Accessory, 72
 - USB, 41
 - USB_Host_Shield, 185
 - VirtualWire, 135, 151, 159, 160
 - bitNo, 110
 - BLANK, 181
 - Blinkendroid, 60
 - Bluesmirf, 16
 - bootloader, 132, 146
 - ButtonPressPeriod, 124
- ## C
- C1, 30, 47, 88, 132, 146, 173
 - C2, 30, 47, 88, 132, 146, 173
 - C3, 88, 132, 146, 173
 - C4, 88, 146
 - C5, 88, 146
 - checkCode, 160
 - checkForMessage, 144, 160
 - clearAllCodes, 161
 - clearLastCode, 160
 - closeAccessory, 188, 190
 - code, 160
 - CODE_1, 151, 160
 - CODE_2, 151, 160
 - codesEqual, 161
 - composeInt, 191
 - Creative Commons, 16, 30
 - cycleTime, 59
 - cylinder Geigera-Müllera, 27, 39
 - umieszczenie w obudowie, 37
 - częstotliwość
 - próbki, 106
 - sygnału nośnego, 106
 - czujnik
 - One-Wire, 141
 - temperatury, 68, 140
 - czytniki kodów RFID, 160
- ## D
- D1, 30, 51, 62, 76, 88, 132, 146, 173
 - D1-8, 114
 - D2, 51, 61, 68, 132, 146, 173
 - D3, 50, 51, 61
 - D5, 50, 61
 - D6, 50, 61, 68
 - DallasTemperature, 69, 135
 - dalmierz ultradźwiękowy, 11, 75
 - automatyczne wykrywanie ruchu, 79
 - bazowe akcesorium Droid, 77
 - budowa, 76
 - echo, 76
 - laser, 75
 - lista materiałów, 76
 - lutowanie lasera i rezystora, 77
 - moduł, 77
 - narzędzia, 77
 - niezasilający przewód USB, 77
 - obliczanie powierzchni pomieszczenia, 79
 - potrzebne elementy, 76
 - przetwornik dalmierza, 79
 - schemat ideowy, 76
 - sterowanie mocą lasera, 76
 - sygnał wyzwalacza, 76
 - szkielet, 78
 - teoria, 80
 - testowanie projektu, 78
 - umieszczenie projektu w obudowie, 79
 - wykorzystanie projektu, 79
 - zasilanie modułu ultradźwiękowego, 76
 - device, 127
 - diagram przejść między stanami, 177
 - digitalWrite, 144
 - digitalWrite HIGH, 181
 - diody LED, 51
 - regulacja jasności, 56, 57
 - direction, 25
 - doorOpenTime, 160
 - Dremel, 96
 - Droid, 61, 62
 - Droid Duino Base, 10
 - Droid Geiger, 50
 - DroidDroid, 23
 - DroidGeigerActivity, 43
 - DroidSoundDisplayActivity, 60
 - DS18B20, 68
 - Duemilanove, 132
 - Duino Droid, 64, 70

E

eBay, 12
 Eclipse, 183
 EEPROM, 159
 elementy, 12
 ENC28J60, 164
 establishPermissionsAndOpen
 ↳ Accessory, 189
 eventInterrupt, 42
 eXtreme Programming, 163

F

f, 106
 Farnell, 12
 Feed, 73
 fillBuffer, 106
 findCodePosition, 160
 flagi sygnalizacyjne, 11, 163
 budowa, 163
 drewniana konstrukcja, 165
 lista materiałów, 164
 mocowanie flag, 168
 narzędzia, 164
 schemat, 166
 serwomechanizmy, 168
 sterowanie, 163
 strona do sterowania
 systemem, 164
 szkic, 167
 teoria, 168
 test, 168
 flash, 160, 181
 funkcje
 acc.isConnected, 42, 59
 acc.powerOn, 42
 acc.read, 59
 addCode, 160
 alert, 192
 analogWrite, 57
 checkCode, 160
 checkForMessage, 144, 160
 clearAllCodes, 161
 clearLastCode, 160
 codesEqual, 161
 composeInt, 191
 digitalWrite, 144
 digitalWrite HIGH, 181
 eventInterrupt, 42
 findCodePosition, 160
 flash, 160, 181
 getInt, 25

isConnected, 186
 isValidCode, 160
 log, 192
 loop, 25, 42, 59, 110, 124,
 144, 160, 181
 map, 144
 microsecondsToCentimeters,
 81
 millis(), 41
 powerOn, 186
 pressButton, 124, 127
 processRadio, 127
 processWord, 127
 pulseIn, 80, 110
 readCard, 160
 readSetTemperature, 144
 readTemperature, 144
 receive, 25
 sendMessage, 42, 186
 Serial.available, 124, 160
 setLeft, 25
 setMotors, 26
 setPower, 144
 setRight, 25
 setup, 25, 41, 59, 110, 124,
 144, 160, 181
 setupAccessory, 188
 show, 181
 takeSounding, 80
 toggleRelay, 181
 unlockDoor, 160
 writeCode, 160

G

getInt, 25
 getLastNonConfiguration
 ↳ Instance, 188
 Google Open Accessory, 34
 Development Kit, 27
 wymagane biblioteki, 184

H

handleGeigerMessage, 43
 Handler, 43, 191
 host USB, 28
 hours, 178, 181

I

IC1, 62, 68, 88, 132, 146
 IC2, 88

identyfikatory
 AndroidAccessory, 41
 E, 42
 Event, 42
 L, 42
 Log, 42
 modułu Bluetooth, 23, 26
 oscPin, 41
 R, 42
 Reading, 42
 RFID, 145
 include, 143, 185
 InputController, 43, 73
 instantaneousCPM, 42
 inteligentny termostat, 11, 129
 budowa, 129
 instalacja, 139
 korzystanie z systemu, 140
 lista materiałów, 132
 lutowanie
 gniazda układu scalonego,
 134
 połączenia, 134
 pozostałych elementów,
 134
 rezystorów i diody, 134
 łączenie komponentów, 135
 mikrokontroler,
 programowanie
 i instalacja, 135
 moduł radiowy, 132
 nadajnik radiowy, 137
 napięcie domowej instalacji
 elektrycznej, 132
 narzędzia, 133
 oznaczenia temperatury, 139
 płyta prototypowa, 133
 profil temperatury, 140, 141
 przedni panel, 139
 przekaźnik, 132
 przewody, 136
 schemat ideowy, 131
 sterownik automatyki
 domowej, 130
 szkic, 135
 teoria, 140
 test, 137
 umieszczanie układu
 w obudowie, 137
 wyłącznik, 136, 144
 zasilanie, 136, 137

- interfejs dźwiękowy, 10, 87
 - dekodowanie sygnału
 - w Arduino, 108
 - kondensatory
 - elektrolityczne, 91
 - konfiguracja środowiska Arduino, 93
 - lista materiałów, 88
 - lutowanie
 - połączeń, 90
 - pozostałych elementów, 91
 - rezystorów i diody, 91
 - układu scalonego, 91
 - moduł, 87
 - narzędzia, 89
 - obudowa, 89
 - płyta prototypowa, 89
 - port szeregowy, 94
 - monitor portu, 95
 - prostowanie i filtrowanie sygnału, 108
 - przewody, 89
 - podłączenie, 91
 - schemat ideowy, 87, 107
 - tablet Android, 88
 - test, 92
 - testowy szkic projektu, 94
 - transmisja szeregową
 - odbierana z Arduino, 94
 - układ elektroniczny, 106
 - filtr dolnoprzepustowy, 108
 - proces przetwarzania, 107
 - schemat blokowy, 107
 - sposób zasilania, 108
 - wtyki prostokątne, 89
 - wybór typu płyty Arduino, 94, 95
 - zamocowanie tabletu, 97
 - isConnected, 186
 - isValidCode, 160
- J**
- JPachube, 73
- K**
- klasy
 - AndroidAccessory, 185, 186
 - Beeper, 105
 - DroidGeigerActivity, 43
 - DroidSoundDisplayActivity, 60
 - Feed, 73
 - Handler, 191
 - InputController, 43, 73
 - org.cbse.blinkendroid.audio, 60
 - Pachube, 73
 - ValueMsg, 192
 - Visualizer, 60
 - klient USB, 28
 - kodowanie danych za pomocą dźwięku, 104
- L**
- L1, 30
 - lastPulseTime, 110
 - left, 25
 - licencja Creative Commons, 16, 30
 - licznik Geigera, 11, 27, 42
 - budowa, 28
 - cylinder Geigera-Müllera, 27, 30
 - anoda, 39
 - katoda, 39
 - parametry, 30
 - podłączenie, 36
 - gniazda bezpiecznika, 33
 - końcówki elementów użyte do połączeń, 33
 - lista materiałów, 30
 - lutowanie
 - końcówek do pinów Arduino, 32
 - pozostałych elementów, 31
 - mocowanie
 - łączówek do nakładki, 31
 - płaskich elementów, 31
 - moduł połączeniowy, 28
 - nakładka USB
 - wierzchnia strona, 31
 - z przymocowanymi łączówkami, 31
 - narzędzia, 31
 - nieładujący przewód USB, 28, 37
 - obudowa, 37
 - ostatnie połączenia, 33
 - obudowa, 89
 - płyta prototypowa, 32
 - schemat ideowy, 29
 - styk PP3, 33
 - teoria, 38
 - test, 36
 - umieszczanie projektu
 - w obudowie, 37
 - ustawianie właściwego napięcia, 35
 - zasilanie, 33
- M**
- log, 192
 - logPeriod, 41
 - loop, 25, 42, 59, 110, 124, 144, 160, 181
- mAccessory, 188
 - map, 144
 - MAX_TEMP, 144
 - Max3124e, 41, 59
 - Metal Oxide Semiconductor Field Effect Transistor, 56
 - metody
 - beep, 106
 - closeAccessory, 188, 190
 - establishPermissionsAnd
 - ↳OpenAccessory, 189
 - fillBuffer, 106
 - getLastNonConfiguration
 - ↳Instance, 188
 - handleGeigerMessage, 43
 - onCreate, 188
 - onResume, 188, 189
 - openAccessory, 188, 191
 - read, 186
 - run, 43
 - sendCommand, 190
 - sendDataToArduino, 26
 - sendMessageToArduino, 190
 - setupAccessory, 188, 190
 - toneRequired, 106
 - write, 186
 - microsecondsToCentimeters, 81
 - mikrokontroler ATmega328, 45
 - millis(), 41
 - MIN_ON_TIME, 143
 - MIN_TEMP, 144
 - modem Bluesmirf, 16

modulacja szerokości impulsu, 57

moduł

- Bluetooth, 15, 16
- identyfikator, 23, 26
- podłączenie, 18
- połączenie z płytą
 - Arduino, 21
- prędkość transmisji, 22, 25
- interfejsu dźwiękowego, 87, 119
- LED, 51
 - sterowanie jasnością, 57
- radiowy, 137, 140, 146
- RFID, 146
- serwomechanizmu, 168
- zegara, 171

monitor portu szeregowego, 95, 110, 118

MOSFET, 56

motorVolts, 21, 22, 25

multimetr, 35

mUsbReceiver, 188, 189

N

nakładki

- Ethernet, 9, 163
- hosta USB, 10, 28, 45, 62
 - mocowanie łączówek, 31
 - płyta prototypowa, 28
 - podłączenie przewodów, 48
 - zamocowanie gniazda, 48
- silnikowe, 9, 17
 - mocowanie łączówki, 17
 - mocowanie zacisków, 17
- piny, 25
- USB, 9

niezasilający przewód USB, 62

O

obiekty

- mAccessory, 188
- mUsbReceiver, 188

offPins, 124

onCreate, 188

OneWire, 69, 135

onPins, 124

onResume, 188, 189

op, 41

Open Accessory, 10, 64, 78

openAccessory, 188, 191

oprogramowanie

- Android
 - Open Accessory, 45
 - sterownik automatyki domowej, 97
- Arduino, 34
 - Arduino Android, 176
 - biblioteka
 - AndroidAccessory, 34
 - biblioteka hosta USB, 34
 - menu Plik, 176
 - monitor portu szeregowego, 118
 - pliki i foldery, 94
 - Załaduj, 94, 176

org.cbse.blinkendroid.audio, 60

oscPin, 41

otwarcie akcesorium, 188

otwarte akcesorium Google, 27

otwarty sprzęt, 16

overrideSwitchPin, 144

P

Pachube, 73

pamięć EEPROM, 159

phone, 25

pierwsze kroki, 12

pilot na podczerwień, 65

pilot telewizyjny, 11, 61

- bazowe akcesorium Droid, 61
- budowa, 61
- lutowanie, 63
- szkic, 64
- lista materiałów, 62
- łączenie komponentów przewodami, 63
- narzędzia, 62
- płyta prototypowa, 63
- schemat ideowy, 62
- teoria, 65
- test, 64
- trening pilota, 64
- układ komend, 64
- umieszczenie projektu w obudowie, 65
- wykorzystanie projektu, 65
- zapis kodów, 64
- zmiana nazwy przycisku, 65

piny

- +5 V, 94
- A4, 94, 110
- analogowe jako piny
 - cyfrowe, 124
- czujnika temperatury, 69
- D2, 61, 68
- D3, 50, 61
- D5, 50, 61
- D6, 50, 61, 68
- direction, 25
- echo, 77
- GND, 33, 35, 77, 137
- IO, 45, 68
- masy, 137
- nr 1, 150
- nr 7, 49
- overrideSwitchPin, 144
- PWM, 25, 50
 - Arduino, 57
 - bazowe akcesorium Droid, 45
 - sygnał, 57
- skojarzone z przyciskami, 124
- Vcc, 137
- Vin, 33
- wyzwalacza, 80
- zasilania, 77

płyta

- perforowana, 52
- prototypowa, 89

Podręcznik Open Accessory, 11, 183

- Android, 186
 - automatyczne uruchamianie i pobieranie aplikacji, 186
- cykl życia, 187
- nauka programowania w systemie, 183
- odbieranie danych, 191
- odbiornik komunikatów rozgłoszeniowych, 189
- otwarcie akcesorium, 188
- wysyłanie danych, 190

Arduino

- instalacja biblioteki, 184
- płyta, 184
- programowanie, 183
- szkic, 185
- wnioski, 192

Pololu, 16
 powerOn, 186
 pressButton, 124, 127
 processRadio, 127
 processWord, 127
 projekty
 Blinkendroid, 60
 dalmierz ultradźwiękowy, 11, 75
 flagi sygnalizacyjne, 11, 163
 inteligentny termostat, 11, 129
 licznik Geigera, 11, 27
 pilot telewizyjny, 11, 61
 podsumowanie, 11
 rejestrator temperatury, 11, 67
 robot sterowany sygnałem
 Bluetooth, 11, 15
 sterowanie zasilaniem, 11, 111
 sterownik automatyki domowej, 11, 85
 światlny pokaz, 11, 45
 wyłącznik czasowy, 11, 171
 zamek otwierany identyfikatorem RFID, 11, 145
 protokół X10, 111
 przekaźniki, 123
 I-8, 114
 kontaktronowe, 123
 przyciski
 Laser, 79
 Menu, 65
 Odblokuj, 153
 opcji, 79
 Włącz, 121
 wyboru, 79
 Wyłącz, 119
 Zasilanie i oświetlenie, 119
 Pulse Width Modulation, 57
 pulseIn, 80, 110
 PWM, 25, 57

R

R, 62, 68
 R1, 30, 47, 62, 68, 88, 132, 146
 R1 – 7, 173
 R1-3, 51
 R2, 30, 88, 132, 146
 R3, 30, 88, 132, 146
 R4, 30, 88, 132, 146

R5, 88, 132
 R8, 173
 R9, 173
 RADIO_CHECK_PERIOD, 144
 read, 186
 readCard, 160
 readSetTemperature, 144
 readTemperature, 144
 receive, 25
 rejestrator temperatury, 11, 67
 bazowe akcesorium Droid, 69
 budowa, 67
 czujnik temperatury, 68
 lista materiałów, 68
 mocowanie elementów do zacisku elektrycznego, 69
 narzędzia, 69
 rezystor podwyższający, 68
 schemat, 68, 69
 serwis Xively, 67, 71
 szkic, 69
 teoria, 71
 test, 70
 umieszczenie projektu w obudowie, 70
 wykorzystanie projektu, 71
 result, 110
 rezystancja, 56
 rezystor podwyższający, 68
 right, 25
 robot sterowany sygnałem
 Bluetooth, 11, 15
 aplikacja, 15
 budowa, 15
 instalacja
 aplikacji Android, 22
 modułu Bluetooth, 18
 testowego szkicu projektu, 20
 właściwego szkicu
 Arduino, 22
 konfiguracja środowiska
 Arduino, 20
 lista materiałów, 17
 mocowanie
 komory na baterie, 18
 łązówek do nakładki, 17
 rolki, 19
 silników, 18
 zacisków do nakładki, 17
 nakładka silnikowa, 18
 narzędzia, 17
 oprogramowanie, 15
 ostatnie połączenia, 19
 podłączenie przewodów, 19
 potrzebne elementy, 16
 próba, 23
 przycinanie pudełka, 19
 schemat ideowy projektu, 16
 silniki i komora na baterie, 19
 sprawdzenie silników, 20
 teoria, 24
 ustawianie identyfikatora urządzenia, 23
 wewnątrz robota, 20
 wybór typu płyty Arduino, 21
 załadowanie szkicu do płyty, 22
 zmiana ustawień telefonu Android umożliwiających bezpieczne pobieranie danych, 23
 rosyjski wojskowy cylinder Geigera CI-1, 30
 run, 43
 Russian Military GEIGER TUBE COUNTER CI-1, 30

S

S1, 132, 146
 S2, 146
 SAMPLE_RATE, 106
 sendCommand, 190
 sendDataToArduino, 26
 sendMessage, 42, 186
 sendMessageToArduino, 190
 Serial.available, 124, 160
 Serial.flush(), 160
 serwis Xively, 67, 71
 serwomechanizmy, 168
 przykładowy przebieg impulsu, 168
 sterowanie, 168
 wybór, 164
 setLeft, 25
 setMotors, 26
 setPower, 144
 setRight, 25
 SETTING_DELAY, 181
 setup, 25, 41, 59, 110, 124, 144, 160, 181
 setupAccessory, 188, 190
 show, 181

- silnik
 - z przekładniami, 16
 - zasilany sygnałem PWM, 57
 - Sparkfun, 17
 - SparkFun, 12
 - stałe
 - BLANK, 181
 - ButtonPressPeriod, 124
 - CODE_1, 151, 160
 - CODE_2, 151, 160
 - cycleTime, 59
 - doorOpenTime, 160
 - MAX_TEMP, 144
 - MIN_ON_TIME, 143
 - MIN_TEMP, 144
 - motorVolts, 22, 25
 - RADIO_CHECK_PERIOD, 144
 - SETTING_DELAY, 181
 - STANDBY, 181
 - supplyVolts, 25
 - THERMOSTAT_ID, 143
 - WAITING, 181
 - standard Open Accessory, 10, 183
 - funkcjonalności, 186
 - natężenie prądu ładowania, 37
 - STANDBY, 181
 - state, 181
 - sterowanie zasilaniem, 11, 111
 - budowa modułu, 112
 - konfiguracja, 121
 - lista materiałów, 114
 - lutowanie połączeń, 116
 - mocowanie płyty Arduino, 116
 - moduł interfejsu
 - dźwiękowego, 119
 - montaż układu, 116
 - narzędzia, 114
 - płyta perforowana, 112, 116
 - podłączanie przewodów do płyty pilota, 114
 - potrzebne elementy, 112
 - procedura odnajdywania połączeń, 115
 - przełącznik kontaktronowy, 111, 112
 - przewody przylutowane do przycisków, 116
 - rozmontowanie pilota, 114
 - schemat
 - sterowanie oświetleniem, 122
 - sterownik automatyki domowej, 119
 - szkic docelowy, 125
 - szkic testowy, 118, 123
 - teoria, 122
 - testowanie, 118, 119
 - wnętrze obudowy, 122
 - zacisk elektryczny, 122
 - zdalnie sterowane gniazdko elektryczne, 111
 - sterownik automatyki domowej, 11, 85, 119
 - dostęp przez internet, 101
 - interfejs dźwiękowy, 87
 - jednostka sterująca, 85
 - Odblokuj, 153
 - oprogramowanie Android, 97
 - przykład konfiguracji, 122
 - regulacja temperatury, 129, 130
 - sterowanie zasilaniem i oświetleniem, 98
 - teoria, 104
 - ustawianie kodu do drzwi, 154
 - współpraca z innymi projektami, 86
 - wybór jednostek temperatury, 140
 - Zamek w drzwiach, 153
 - sterownik ENC28J60, 164
 - supplyVolts, 25
 - systemy
 - automatyki domowej, 86
 - Automatyki Kryjówki, 111
 - szkic, 20
 - dalmierz ultradźwiękowy, 78
 - dekodowanie sygnału w Arduino, 109
 - do testowania
 - dźwięku, 94
 - silników, 20
 - Droid Geiger, 50
 - flagi sygnalizacyjne, 167
 - inteligentny termostat, 141
 - licznik Geigera, 39
 - pilot telewizyjny, 64
 - rejestrator temperatury, 69
 - rfid_door, 151
 - robot sterowany sygnałem Bluetooth, 22, 24
 - sterowanie zasilaniem, 125
 - odbierający komendy Bluetooth, 22
 - sterujący zamkiem, 155
 - światlny pokaz, 57
 - wyłącznik czasowy, 176
- ## §
- środowisko Eclipse, 183
 - światlny pokaz, 11, 45
 - bazowe akcesorium Droid, 45
 - budowa, 50
 - czułość układu, 55
 - instalacja
 - aplikacji Android, 54
 - szkicu Arduino, 54
 - lista materiałów, 51
 - lutowanie elementów, 52
 - mikrofon, 55
 - mocowanie
 - łącówek, 52, 53
 - tranzystora, 52
 - zacisków, 52
 - moduły LED, 51
 - narzędzia, 52
 - obudowa z wywierconymi otworami, 55
 - panele LED, 56
 - płyta perforowana, 52
 - podłączenie
 - części projektu, 54
 - zasilacza prądu stałego 12 V do płyty, 54
 - potrzebne elementy, 51
 - schemat ideowy, 50
 - teoria, 56
 - test, 55
 - tranzystor MOSFET, 50
 - tryb
 - Słupki, 55
 - testowy, 55, 59
 - Uderzenie, 55
 - umieszczenie projektu w obudowie, 55
 - włącznik i gniazdo zasilające, 55
 - wykorzystanie projektu, 55

T

T1, 30, 132, 146, 173
 T1-3, 51
 tabele

- offPins, 124
- onPins, 124

 tablet Android, 10

- przesyłanie poleceń do Arduino, 86
- sterowanie zasilaniem, 111
- sterownik automatyki domowej, 85

 tagi

- activity, 186
- application, 187

 takeSounding, 80
 technika zwinnego wytwarzania oprogramowania, 163
 telefon Android, 28

- pilot telewizyjny, 64

 terminal portu szeregowego, 94
 termostat, 129
 test

- generatora wysokiego napięcia, 35
- licznika Geigera, 36

 THERMOSTAT_ID, 143
 timerPeriod, 41
 toggleRelay, 181
 toneRequired, 106
 tranzystor

- MOSFET, 50, 56
- sterowany poziomem logicznym, 56

U

układ

- Arduino Mega, 183
- ATMega328, 129

 ultradźwiękowy pomiar odległości, 80
 unlockDoor, 160
 urządzenie Android

- Ustawienia, 93

 USB_Host_Shield, 185

V

ValueMsg, 192
 VirtualWire, 135, 151, 159
 Visualizer, 60

W

WAITING, 181
 wejścia

- wyzwalacz, 77

 write, 186
 writeCode, 160
 wyjścia

- cyfrowe Arduino, 112
- echo, 77, 80
- PWM, 38

 wyłącznik czasowy, 11, 171

- budowa, 171
- instalacja szkicu Arduino, 176
- lista materiałów, 173
- lutowanie
 - gniazda i łączówki, 175
 - połączeń, 175
 - pozostałych elementów, 175
- narzędzia, 173
- plyta prototypowa, 173
- schemat ideowy, 172
- siedmiosegmentowy
 - wyświetlacz o wspólnej katodzie, 171
- teoria, 177
- test, 176
- zasilanie, 173

X

X1, 47, 132, 146, 173
 Xively, 66, 71

- Add Channel, 71
- Add Device, 71
- identyfikator kanału, 71
- osobisty klucz API konta, 71
- podsumowanie utworzonego kanału, 72

Private Device, 71
 Save Channel, 71
 tworzenie kanału, 72
 wykres temperatury, 72

Z

zamek otwierany

- identyfikatorem RFID, 11, 145
- ATMega168, 146
- ATMega328, 146
- budowa, 145
- czytnik kodu RFID, 146
- elektromagnes zamka, 146
- instalacja, 155
- jednostka sterująca, 155
- lista materiałów, 146
- lutowanie
 - gniazda układu scalonego i wyłączników, 150
 - połączeń, 148
 - pozostałych elementów, 150
 - rezystorów i diody, 149
- mikrokontroler
- moduł
 - radiowy, 146, 151
 - RFID, 146
- narzędzia, 148
- plyta prototypowa, 148, 149, 150, 151, 152
- dłaczanie komponentów, 152
- teoria, 155
- test, 152
- tranzystor MOSFET, 145
- układ scalony stabilizatora napięcia, 146
- umieszczanie układu w obudowie, 154
- użytkowanie systemu, 155
- zamek, 148
- zdalne otwieranie drzwi, 153

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄZKI**
- 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**

**Szalony
Geniusz**

ARDUINO I ANDROID NIESAMOWITE PROJEKTY

Elektroniczny ninja na start! Wyobraź sobie zdalnie sterowane urządzenie, które w krótkim czasie stworzysz za pomocą systemu Android i platformy Arduino. Ten najpopularniejszy mobilny system operacyjny i platforma, która szturmem zdobyła świat elektronicznych maniaków, to niesamowity duet — z nim zbudujesz wszystko, co sobie zamarysz. Koniecznie zajrzyj do tej książki i zrealizuj niezwykle projekty, które napelnia Cię dumą i zaskoczą niejednego znajomego!

Znajdziesz tu instrukcje pozwalające na wykonanie między innymi pilota telewizyjnego, licznika Geigera i termometru. Każdy gadżet jest szczegółowo opisany — dowiesz się, jakich części potrzebujesz, jak je ze sobą połączyć i jak uruchomić urządzenie. Zbuduj samodzielnie dalmierz i inteligentny termostat. Już teraz połącz działanie lutownicy z możliwościami platformy Android i zaprojektuj swój własny zdalnie sterowany świat!

Dzięki tej książce:

- poznasz tajniki platformy Arduino
- stworzysz aplikację dla systemu Android
- zrealizujesz wymarzony projekt
- wkroczysz w świat praktycznej elektroniki

Twoja przepustka do świata elektroniki!

Patronat medialny:

mikrokontrolery.blogspot.com



helion.pl
księgarnia
internetowa

Nr katalogowy: 19869



Księgarnia internetowa:

<http://helion.pl>



Zamówienia telefoniczne:

0 801 339900



0 601 339900



Helion

Sprawdź najnowsze promocje:

• <http://helion.pl/promocje>

Książki najchętniej czytane:

• <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

• <http://helion.pl/newsy>

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 KORZYŚCI

ISBN: 978-83-246-8708-4



9 788324 687084

Cena: 39,90 zł

Informatyka w najlepszym wydaniu