

# Mikrokontroler z czujnikami i akcesoriami

**VR 0003** Zestaw startowy UNO XXL-1. Starter Kit UNO R3, zgodny z Arduino

**VR 0004** Zestaw RFID. Kompletny zestaw startowy z samouczkiem video

**VR 0006** Największy zestaw startowy do nauki Arduino

## OSTRZEŻENIA!

1. Nie nadaje się dla dzieci w wieku poniżej 8 lat – małe części – ryzyko zadławienia
2. Do użytku pod nadzorem osoby dorosłej
3. Nie wyrzucać do śmieci
4. Chronić włosy, dłonie, twarz, ubrania, przewody z dala od ruchomych części zestawu.
5. Używać urządzenia ostrożnie i tylko zgodnie z przeznaczeniem.
6. Konstrukcje mogą uszkodzić meble lub ściany a przy nieumiejętnym używaniu, powodować kontuzję ludzi i zwierząt domowych.
7. Elementy elektryczne nie powinny mieć kontaktu z wodą.
8. Nie zanurzać w wodzie elementów zestawu
9. Trzymaj baterie z dala od ognia
10. Jeśli elementy elektryczne uległy zamoczeniu, wytrzyj je jak najszybciej i pozostaw do całkowitego wyschnięcia przed ponownym złożeniem.
11. Uważaj, by nie potknąć się o przewód.
12. Zalecamy użycie baterie alkalicznych. Nie używaj w tym zestawie baterii akumulatorowych
13. Baterie powinny być poprawnie umieszczone w przedziale baterijnym.
14. Umieść baterie zgodnie z oznaczeniem +/- . Nigdy nie powodować zwarc biegunów baterii.
15. Baterie, które nie są do tego przeznaczone nie powinny być doładowywane.
16. Włóż nowe baterie na wskazane miejsca upewniając się, że bieguny (+/-) baterii są umieszczone w odpowiednich kierunkach.
17. Tylko baterie jednego typu mogą być jednocześnie używane.
18. Nie łącz ze sobą baterii alkalicznych, akumulatorowych i standardowych.
19. Nie używać jednocześnie starych baterii z nowymi.
20. Wyczerpane baterie powinny być natychmiast usunięte.
21. Usuń baterie, jeśli nie używasz urządzenia przez dłuższy czas (okres dłuższy niż 2 tygodnie)
22. Należy zachować opakowanie lub/i instrukcję. Zawierają one ważne informacje mogące być przydatne w przyszłości.
23. Użytkowanie niezgodne z zaleceniami zwalnia producenta od odpowiedzialności za ewentualne szkody.



**nowa szkoła**  
ul. POW 25, 90-248 Łódź,  
[www.nowaszkoła.com](http://www.nowaszkoła.com)  
tel. (42) 630 17 28,  
(42) 630 04 88, fax: (42) 632 73 28



SIN VR 0003, 4, 6 01/22

**Wiek**

■ 8+

## Spis treści

Nasz Samouczek .....	4
Lekcja 0. Instalacja Arduino IDE .....	4
Lekcja 1. Dodaj Biblioteki oraz Monitor Szeregowy .....	7
Lekcja 2. Migająca dioda .....	10
Lekcja 3. Diody LED .....	13
Lekcja 4. Diody RGB LED .....	15
Lekcja 5. Wejścia cyfrowe .....	17
Lekcja 6. Brzęczyk aktywny .....	18
Lekcja 7. Brzęczyk pasywny .....	19
Lekcja 8. Czujnik przechyłu .....	20
Lekcja 9. Serwo .....	21
Lekcja 10. DHT11 . Czujnik temperatury i wilgotności .....	22
Lekcja 11. Moduł joystick analogowy .....	23
Lekcja 12. Moduł czujnik wykrywania poziomu wody .....	24
Lekcja 13. Moduł zegar czasu rzeczywistego .....	26
Lekcja 14. Moduł czujnik dźwięku .....	27
Lekcja 15. Moduł RC522 RFID .....	28
Lekcja 16. Wyświetlacz LCD .....	29
Lekcja 17. Termometr .....	31
Lekcja 18. Osiem diod z rejestrem przesuwным 74HC595 .....	32
Lekcja 19. Monitor Portu Szeregowego .....	34
Lekcja 20. Fotokomórka .....	36

### Animacje

Przed wprowadzeniem animacji, należy wspomnieć o dwóch operatorach bitowych, prawym (<<) i lewym (>>).

Poniżej pokazano przykład:

```

Left shift operator <<
  Bit  7 6 5 4 3 2 1 0
      | | | | | | | |
byte a= B 1 0 0 1 1 0 1 0
      | | | | | | | |
      B 1 0 0 1 1 0 1 0
byte b=a<<4= B 1 0 1 0 0 0 0 0
  
```

Bit przesuwa się w lewo, a prawa strona wypełnia się zerami.

Podobnie dzieje się z prawą stroną, prawy operator przesuwa bit na prawą stronę, a lewa wypełnia się zerami,

Byte a=B11110001;

Byte b=a>>4;

B stanie się B00001111.

### Program Arduino:

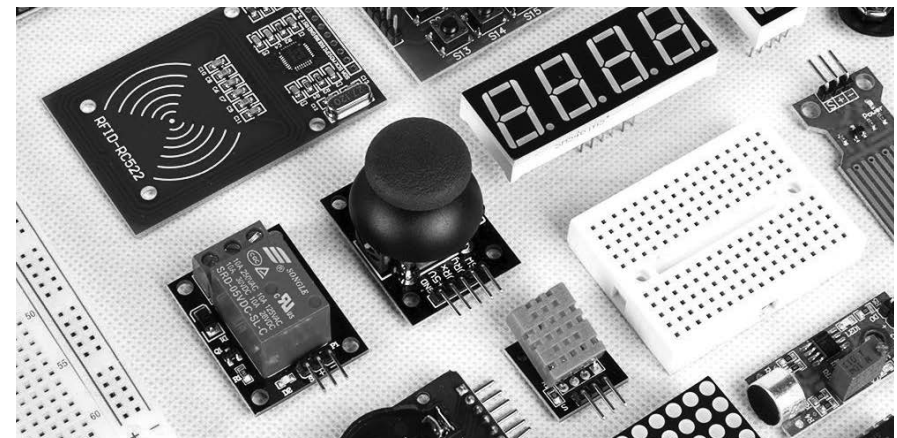
```

#define DATA 6
#define LATCH 8
#define CLOCK 10
void setup() {
  // put your setup code here, to run once:
  pinMode(DATA,OUTPUT);
  pinMode(LATCH,OUTPUT);
  pinMode(CLOCK,OUTPUT);
}
int binary[]={1,2,4,8,16,32,64,128};
byte smile[]={
  B00011000,
  B00011000,

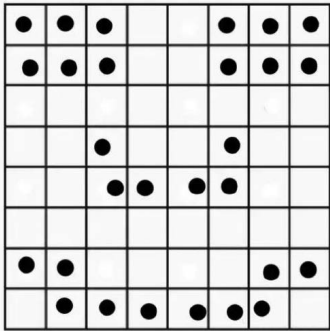
  B11111111,
  B11011011,
  B11000011,
  B11111111,
  B00111100,
  B10000001
};
void loop()
{
  for(int shift=0;shift<9;shift++)
  {
    for(int stay=0;stay<30;stay++)
    {
      for(int i=0;i<8;i++)
      {
        digitalWrite(LATCH,LOW);
        shiftOut(DATA,CLOCK,MSBFIRST,smile[i]<<shift);
        shiftOut(DATA,CLOCK,MSBFIRST,binary[i]);
        digitalWrite(LATCH,HIGH);

        //end i
        delay(1);
      }
      //end stay
    }
    //end loop
  }
}
  
```

Dziękujemy za przeczytanie samouczka.



## Generowanie grafiki



Grafika, którą chcemy wygenerować

Program Arduino:

```
#define DATA 6
#define LATCH 8
#define CLOCK 10
void setup() {
  // put your setup code here, to run once:
  pinMode(DATA,OUTPUT);
  pinMode(LATCH,OUTPUT);
  pinMode(CLOCK,OUTPUT);
}
int binary[]={1,2,4,8,16,32,64,128};
int r,c=0;
void setLed(int row,int column,int del)
{
  digitalWrite(LATCH,LOW);
  shiftOut(DATA,CLOCK,MSBFIRST,"binary[column]");//列
}

shiftOut(DATA,CLOCK,MSBFIRST,binary[row]);//行
digitalWrite(LATCH,HIGH);
delay(del);
}
void loop()
{
  setLed(0,0,0);
  setLed(0,1,0);
  setLed(0,2,0);
  setLed(0,5,0);
  setLed(0,6,0);
  setLed(0,7,0);
  setLed(1,0,0);
  setLed(1,1,0);
  setLed(1,2,0);
  setLed(1,5,0);
  setLed(1,6,0);
  setLed(1,7,0);
  setLed(3,2,0);
  setLed(3,5,0);
  setLed(4,2,0);
  setLed(4,3,0);
  setLed(4,4,0);
  setLed(4,5,0);
  setLed(6,0,0);
  setLed(6,1,0);
  setLed(6,6,0);
  setLed(6,7,0);
  setLed(7,1,0);
  setLed(7,2,0);
  setLed(7,3,0);
  setLed(7,4,0);
  setLed(7,5,0);
  setLed(7,6,0);
}
```

Osiągniemy poniższy efekt:



Uśmiechnięta grafika

Ale ta metoda może być uciążliwa, dlatego następny przykład zaprezentuje bardziej intuicyjną metodę.

Wygeneruj dwie grafiki:

Program Arduino:

```
#define DATA 6
#define LATCH 8
#define CLOCK 10
void setup() {
  // put your setup code here, to run once:
  pinMode(DATA,OUTPUT);
  pinMode(LATCH,OUTPUT);
  pinMode(CLOCK,OUTPUT);
}

int binary[]={1,2,4,8,16,32,64,128};
byte smile[]={
  B00011000,
  B00011000,
  B11111111,
  B11011011,
  B11000011,
  B11111111,
  B00111100,
  B10000001
};
};
void loop()
{
  for(int i=0;i<8;i++)
  {
    digitalWrite(LATCH,LOW);
    shiftOut(DATA,CLOCK,MSBFIRST,smile[i]);//列
    shiftOut(DATA,CLOCK,MSBFIRST,binary[i]);//行
    digitalWrite(LATCH,HIGH);
  }
  delay(1);
}
```

Uwagi:

W tym układzie, 0 w każdym bicie reprezentuje diodę matrycy, którą chcemy włączyć, co sprawia, że grafiki są bardziej intuicyjne. Spróbuj wprowadzić własne zmiany.

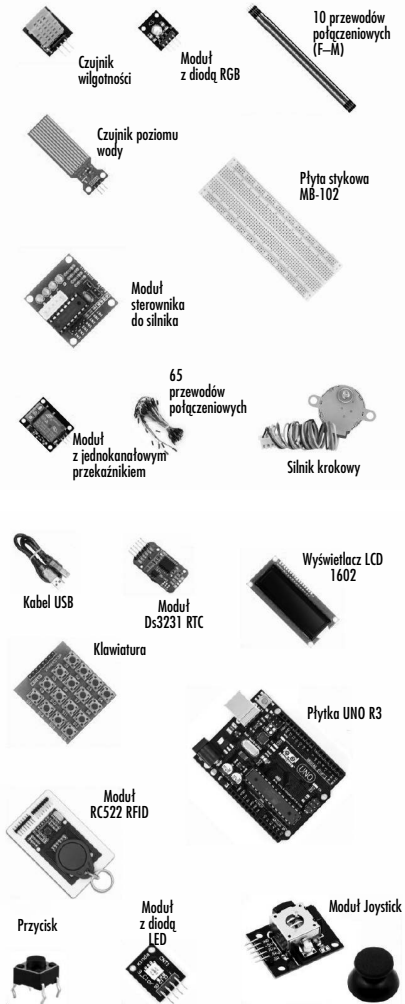


Lekcja 21. Rejestr przesuwany i wyświetlacz segmentowy .....	38
Lekcja 22. 4-cyfrowy wyświetlacz siedmiosegmentowy .....	39
Lekcja 23. Silnik krokowy .....	40
Lekcja 24. Moduł SMD RGB i moduł RGB .....	42
Lekcja 25. Moduł przekaźnika jednokanałowego .....	42
Lekcja 26. Arduino-klawiatura matrycowa 4x4 .....	43
Lekcja 27. Arduino-wyświetlacz matrycowy LED 8x8.	
Szczegółowe zastosowanie .....	43

## Nasz Samouczek

Ten samouczek dedykowany jest osobom początkującym. Przedstawione są w nim podstawowe informacje, które wyjaśniają jak korzystać z minikontrolera Arduino, czujników i pozostałych elementów. Aby lepiej poznać Arduino, polecamy zapoznać się z pozycją „Arduino Cookbook” autorstwa Michael’a Margolis’a. Niektóre kody w samouczku zostały opracowane przez Simon’a Monk’a, który napisał wiele książek poświęconych urzędziom open-source.

## Wykaz elementów zestawu



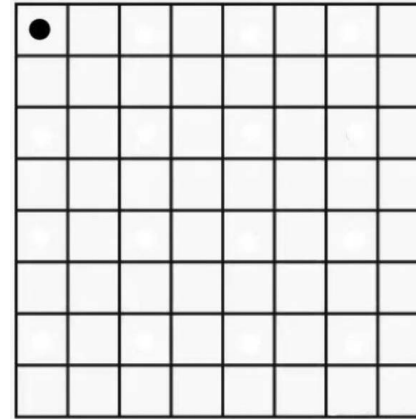
## Lekcja 0. Instalacja Arduino IDE

### Wprowadzenie

Arduino IDE (Integrated Development Environment) to aplikacja służąca do programowania platformy Arduino. W tej lekcji pokażemy, jak skonfigurować komputer, aby skorzystać z aplikacji Arduino i zacząć naukę. Oprogramowanie Arduino jest dostępne dla systemów operacyjnych Windows, Mac i Linux. Proces instalacji jest inny dla każdego z tych systemów i niestety częściowo wymaga ręcznej instalacji.

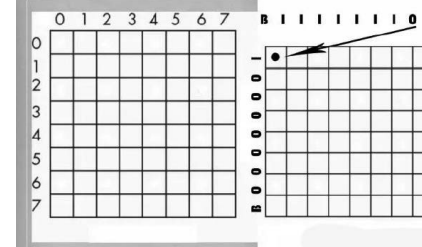
Jak zapalić tylko jedną diodę LED

Teraz spróbuj uzyskać poniższy efekt:



Dioda LED w pierwszym wierszu pierwszego wiersza

Taki układ można otrzymać poprzez porównanie mapy bitowej:



Mapa bitowa diod LED na pierwszym wierszu pierwszego wiersza

### Program Arduino

```
#define DATA 6
#define LATCH 8
#define CLOCK 10
void setup() {
  // put your setup code here, to run once:
  pinMode(DATA,OUTPUT);
  pinMode(LATCH,OUTPUT);
  pinMode(CLOCK,OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LATCH,LOW);
  shiftOut(DATA,CLOCK,MSBFIRST,"B00000001");//1
  shiftOut(DATA,CLOCK,MSBFIRST,B00000001);//1
  digitalWrite(LATCH,HIGH);
  do{}while(1);
}
```

Zauważ, że w domyślnej kolumnie bity LED są ustawione na 0:

Aby dioda na początku wiersza zapaliła się, ciąg bitów pierwszego wiersza prezentuje się następująco: B00000001, a odpowiada mu cyfra 1.

Jeśli chcesz zapalić diodę LED na początku drugiego wiersza, ciąg bitów drugiego wiersza będzie wyglądał w ten sposób: B00000010, a odpowiada mu cyfra 2.

...

Aby zapalić diody siódmego wiersza, ciąg bitów będzie wyglądał w ten sposób: B10000000, a odpowiada mu liczba 128.

Kolumny należy „odwrócić”.

Poniższy przykład pozwoli napisać funkcję określającą odpowiedni wiersz i kolumnę, na których chcemy włączyć analogiczne diody LED, bez potrzeby porównywania map bitowych.

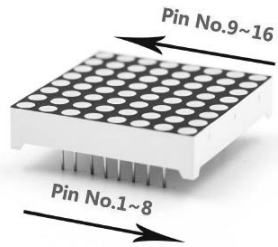
Funkcja włącz jedną diodę LED

Program Arduino:

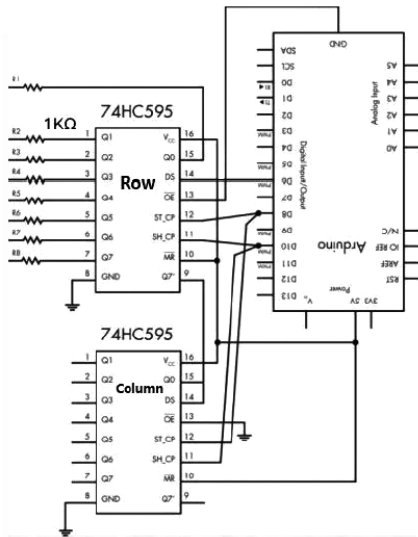
```
#define DATA 6
#define LATCH 8
#define CLOCK 10
void setup() {
  // put your setup code here, to run once:
  pinMode(DATA,OUTPUT);
  pinMode(LATCH,OUTPUT);
  pinMode(CLOCK,OUTPUT);
}
int binary[]={1,2,4,8,16,32,64,128};
int r,c=0;
void setLed(int row,int column,int del)
{
  digitalWrite(LATCH,LOW);
  shiftOut(DATA,CLOCK,MSBFIRST,"binary[column]");//1
  shiftOut(DATA,CLOCK,MSBFIRST,binary[row]);//1
  digitalWrite(LATCH,HIGH);
  delay(del);
}
void loop()
{
  for(int i=0;i<8;i++)
  {
    for(int j=0;j<8;j++)
    {
      setLed(i,j,100);
    }
  }
}
```

Funkcja setLed określa odpowiedni wiersz lub odpowiednią kolumnę na których chcemy włączyć analogiczne diody LED. Dwie funkcje pętli „For” w programie sprawiają, że diody LED przesuwają się z pierwszego wiersza i pierwszej kolumny na ósmą kolumnę pierwszego wiersza.

Odwolując się do cechy wzorku jaką jest bezwładność, można stworzyć różne grafiki, jak to na kolejnym przykładzie.

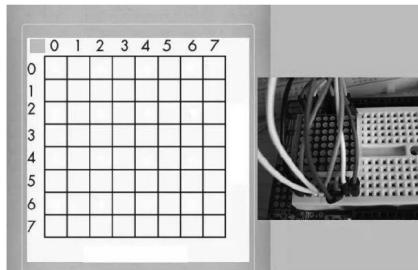


Schemat połączenia Arduino z dwoma rejestrami przesuwными



Piny numer 9 na rejestrach przesuwanych (Row) łączą się z pinami numer 14 (kolumna).

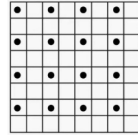
Dwa rejestry przesuwne podłączone do Arduino – schemat



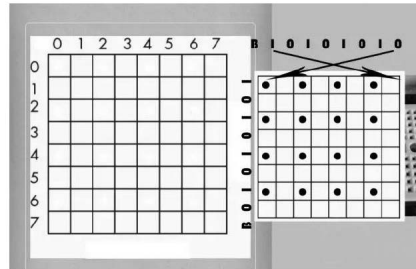
Wiersze i kolumny. Mapa bitowa

Jeśli kolumna (column) jest ustawiona na 0 (LOW) i jeśli wiersz (row) jest ustawiony na 1 (HIGH), odpowiednia dioda LED zapali się.

Na przykład, każda matryca wyświetli wzór pokazany poniżej (czarne kropki oznaczają podświetlone diody LED):



Odwołując się do wyznacznika bitmapowego, można uzyskać poniższą mapę bitową:



Mapa bitowa

(PROGRAM)

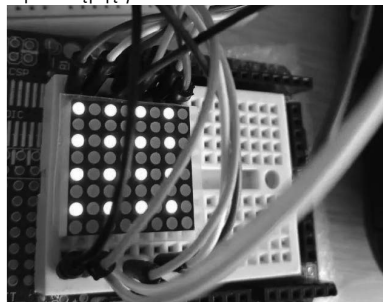
```
#define DATA 6
#define LATCH 8
#define CLOCK 10
void setup() {
  // put your setup code here, to run once:
  pinMode(DATA,OUTPUT);
  pinMode(LATCH,OUTPUT);
  pinMode(CLOCK,OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LATCH,LOW);
  shiftOut(DATA,CLOCK,MSBFIRST,"B01010101");//列
  shiftOut(DATA,CLOCK,MSBFIRST,"B01010101");//行
  digitalWrite(LATCH,HIGH);
  do{}while(1);
}
```

Konieczne będą operacje na bitach:

```
~B01010101==B01010101
```

Operacja odwrotna

Efekt jest następujący:



**Krok 1:** Wejdź na <https://www.arduino.cc/en/Main/Software> i odzyskaj stronę pokazaną poniżej. Wersja dostępna na stronie jest zazwyczaj najnowszą wersją i może różnić się od wersji pokazanej na zdjęciu.



**Krok 2:** Pobierz oprogramowanie, które jest kompatybilne z twoim systemem operacyjnym. System Windows jest tutaj tylko przykładem.

Kliknij Windows Installer (Instalator Windows).



Kliknij JUST DOWNLOAD (tylko pobierz).

- arduino-1.8.0-linux32.tar.gz
- arduino-1.8.0-linux64.tar.gz
- arduino-1.8.0-macosx.zip
- arduino-1.8.0-windows.exe
- arduino-1.8.0-windows.zip

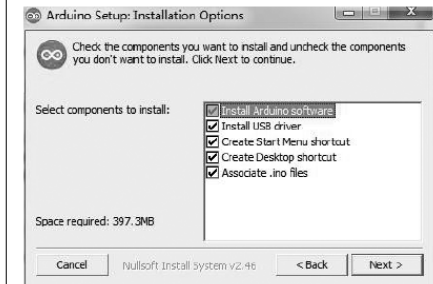
Wersja 1.8.0 pokazana w samouczku w czasie jego opracowywania była wersją najnowszą.

Instalacja Arduino (Windows)

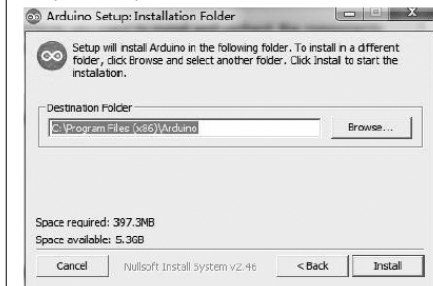
Zainstaluj Arduino przy użyciu pakietu instalacyjnego EXE.



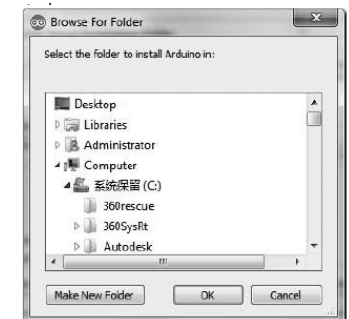
Kliknij **I Agree** (wyrzuję zgodę), aby zobaczyć następujący interfejs:



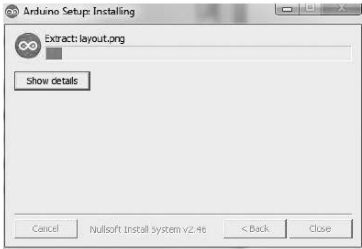
Kliknij **Next** (dalej).



Kliknij **Browse** (szukaj), aby wybrać ścieżkę instalacji lub wpisz



Kliknij **Install** (zainstaluj), aby rozpocząć instalację.



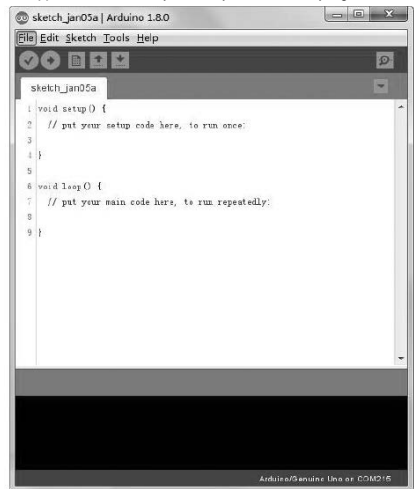
Kiedy pojawi się interfejs pokazany poniżej, kliknij **Install** (zainstaluj), aby zakończyć proces instalacji.



Na pulpicie pojawi się ikona Arduino.



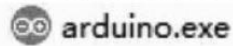
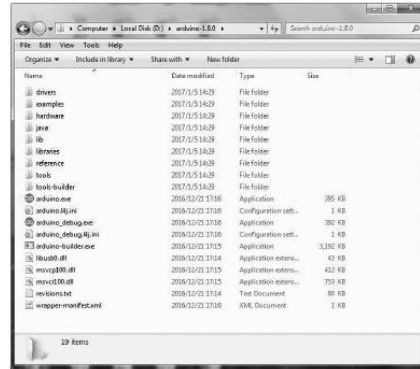
Kliknij ją dwukrotnie, aby otworzyć środowisko programowania.



Wybierając pakiet instalacyjny możesz pominąć poniższe kroki i przejść od razu do następnej sekcji. Jednak, jeśli chcesz poznać inne metody instalacji niż pakiet, kontynuuj czytanie.

Rozpakuj pobrany plik ZIP, kliknij dwukrotnie, aby go otworzyć i zobaczyć środowisko programowania.

[arduino-1.8.0-windows.zip](#)



Inne metody instalacji wymagają pobrania sterowników.

Folder Arduino zawiera oprogramowania oraz sterowniki, które pozwalają na połączenie platformy Arduino z komputerem za pomocą kabla USB. Przed uruchomieniem oprogramowania Arduino, należy zainstalować sterowniki USB. Podłącz jeden koniec kabla USB do Arduino, a drugi do komputera. Dioda zasilania zaświeci się. Może pojawić się komunikat „Found New Hardware” / Znalaziono nowy sprzęt. Zignoruj go oraz przerwij próby automatycznej instalacji sterowników przez system Windows. Najpewniejszą metodą jest instalacja sterowników USB za pomocą Menedżera urządzeń (Device Manager). Dostęp do niego zależy od posiadanej wersji systemu operacyjnego. W przypadku wersji Windows 7, najpierw należy otworzyć panel sterowania, następnie kliknąć Właściwości i odszukać opcję Menedżer

**Krok 2:** Program (Skorzystaj z przykładowego programu na płycie CD lub pobierz z oficjalnej strony).

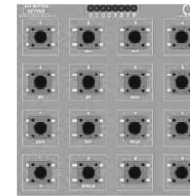
**Krok 3:** Skompiluj program.

**Krok 4:** Wgraj program na płytkę UNO.

Podsumowanie

Moduł przekaźnika jedнокanałowego jest bardzo prostą i praktyczną technologią, którą łatwo opanować.

## Lekcja 26. Arduino-klawiatura matrycowa 4x4



Ten samouczek Robo India wyjaśnia, jak podłączyć klawiaturę matrycową 4x4 do płytki Arduino.

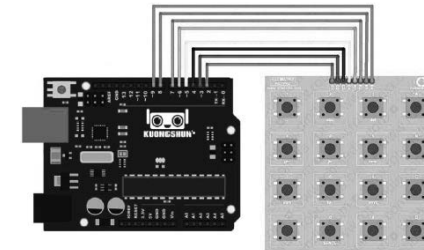
Szczegółowy samouczek

**1. Wprowadzenie:**

Ilustrowany samouczek, który krok po kroku pokazuje, jak podłączyć 16 klawiszy klawiatury 4x4 do Arduino. Wszystkie przyciski są ze sobą połączone w formie matrycy posiadającej cztery wiersze i cztery kolumny. Klawiatura wyposażona jest w 8 pinów, po cztery na wiersz i kolumnę. Aby ją uruchomić, wyjście jest przypisywane do kolumny lub wiersza. Proces ten odbywa się na wysokiej częstotliwości dla wszystkich kolumn i wierszy.

**2. Obwód:**

Stwórz poniższy obwód korzystając z wcześniej wspomnianych komponentów. Możesz wykorzystać oryginalną płytkę UNO.



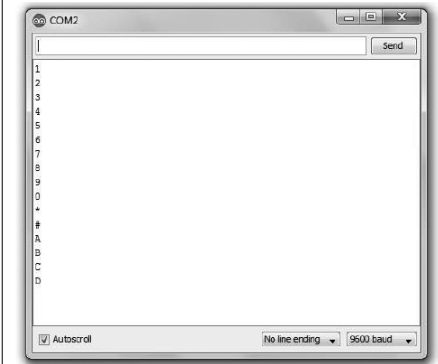
**3. Programowanie:**

Poniżej prezentujemy program kompatybilny z tym obwodem. Każda komenda jest opisana w komentarzu.

Program można pobrać (Szkic Arduino) z tego miejsca.

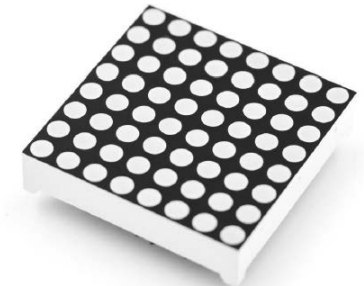
<https://playground.arduino.cc/Code/Keypad/>

**4. Wyjście:** Po wgraniu szkicu, otwórz monitor szeregowy, aby zobaczyć poniższe okno.



Obwód wygląda tak, kiedy jest połączony z oryginalną płytką UNO.

## Lekcja 27. Arduino-wyświetlacz matrycowy LED 8x8-szczegółowe zastosowanie



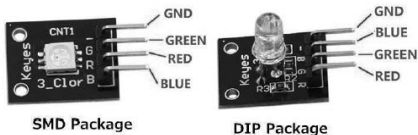
Niezbędne komponenty

- 1 x matryca LED 8x8 (czerwona, wspólna anoda)
- 8 x rezystorów 1K ohm
- 2 x rejestr przesuwany 74HC595
- 1 x płytka stykowa
- 1 x płytka Arduino UNO
- Kilka przewodów
- Podstawowy schemat okablowania

## Lekcja 24. Moduł SMD RGB i moduł RGB

### Wprowadzenie

Ten projekt nauczy nas korzystania z modułów SMD RGB i RGB. Funkcje obu modułów są praktycznie takie same. Możemy jednak wybrać kształt, który nam odpowiada lub który jest nam potrzebny w danym momencie. Moduły SMD RGB i RGB składają się z wielokolorowej diody LED. Dzięki podłączeniu napięcia do pinów R (czerwony), G (zielony), B (niebieski) możemy dostosować jasność trzech głównych kolorów (czerwonego, niebieskiego, zielonego) lub uzyskać efekt melanżu.



### Komponenty

- (2) 1\* płytko UNO
- (3) 1\* kabel USB
- (4) 1\* Moduł SMD RGB
- (5) 1\* Moduł RGB
- (6) damsko-męskie przewody połączeniowe DuPont

### Zasady

Moduł powinien być podłączony do rezystora, aby zapobiec przepaleniu.

PWM: kontroluje jasność barw podstawowych, abyśmy mogli uzyskać różne kolory.

Napięcie: 5V

Moduł ma wyprowadzoną wspólną katodę

Doświadczenie krok po kroku:

**Krok 1:** Podłącz obwód tak jak pokazano poniżej:



**Krok 2:** Programowanie (zaleca się wykorzystanie kodu z płyty CD lub strony internetowej producenta)

**Krok 3:** Kompilowanie programu

**Krok 4:** Wgranie programu na płytkę UNO.

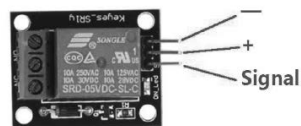
### Podsumowanie

Oba moduły są bardzo proste i praktyczne. Co więcej, są bardzo łatwe do opanowania i późniejszego wykorzystania.

## Lekcja 25. Moduł przekaźnika jednocanałowego

### Wprowadzenie

Ten projekt pomoże nam zrozumieć, jak korzystać z przekaźnika jednocanałowego. Przekaznik jest elektrycznie sterowanym komponentem, który pracuje z napięciem od 28V do 250, przez co pozwala nam kontrolować inne elementy pobierające prąd. Może być wykorzystany jako włącznik do urządzeń elektrycznych, w alarmach, np. przeciwwłamaniowych, zabawkach, itp.



### Niezbędne komponenty

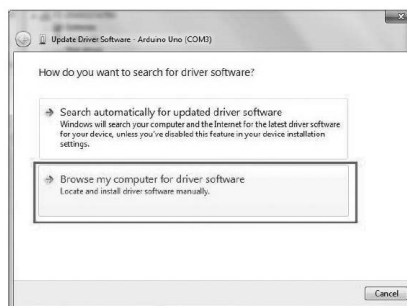
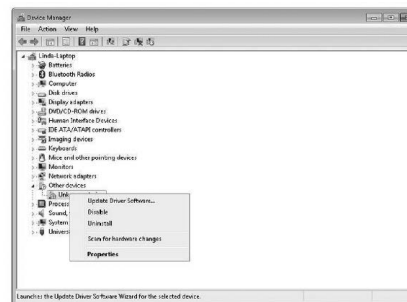
- 1 x płytko UNO
- 1 x kabel USB
- 1 x moduł przekaźnika jednocanałowego

### Procedury realizacji projektu

**Krok 1:** podłącz obwód:

- Uno R3 masa → wyprowadzenie wejściowe modułu „-”
- Uno R3 zasilanie +5V → wyprowadzenie wejściowe modułu „+”
- Uno R3 pin cyfrowy 2 → Rezystor lub nie → wyprowadzenie wejściowe modułu „S”

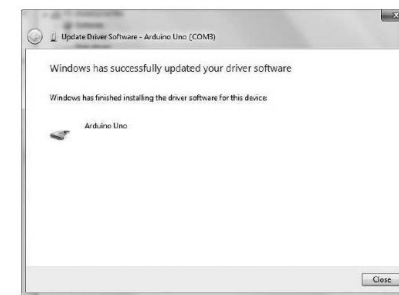
urządzeń na liście. Po kliknięciu opcji inne urządzenia, powinna pojawić się kolejna opcja, nieznanne urządzenie, oznaczona małym, żółtym trójkątem ostrzegawczym. To właśnie Arduino.



Kliknij prawy przycisk myszy i wybierz pierwszą opcję z menu (Update Driver Software/Aktualizuj oprogramowanie sterownika). Następnie zostaniesz poproszony o wybranie opcji „Wyszukaj automatycznie zaktualizowane oprogramowanie sterownika” (Search automatically for updated driver software) lub „Przeglądaj mój komputer w poszukiwaniu sterownika” (Browse my computer for driver software). Wybierz opcję Przeglądaj/Browse i podaj ścieżkę do katalogu X\arduino1.8.0\drivers.



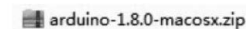
Kliknij Dalej/Next. W tym momencie może pojawić się komunikat bezpieczeństwa. Jeśli tak się stanie, zezwól na instalację oprogramowania. Po zakończeniu instalacji pojawi się wiadomość potwierdzająca.



Użytkownicy systemu Windows mogą pominąć następane kroki, które dotyczą procesu instalacji oprogramowania na systemach Linux i Mac.

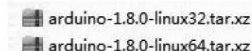
### Instalacja Arduino (Mac OS X)

Pobierz i rozpakuj plik ZIP, kliknij dwukrotnie aplikację Arduino, aby uruchomić środowisko Arduino IDE. Możesz zostać poproszony o zainstalowanie biblioteki uruchomieniowej Java, jeśli nie zostanie wykryta. Kiedy instalacja zakończy się sukcesem, możesz otworzyć Arduino IDE.

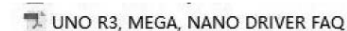


### Instalacja Arduino (Linux)

Aby rozpocząć proces instalacji, otwórz terminal i wydaj polecenie instalacji. Jeśli korzystasz z systemu Ubuntu, należy pobrać Arduino IDE za pomocą Centrum oprogramowania Ubuntu.



Wskazówki: Jeśli masz problem z instalacją sterowników, skorzystaj z sekcji Często Zadawane Pytania w pliku UNO R3, MEGA, NANO DRIVER FAQ.



## Lekcja 1. Dodaj Biblioteki i Monitor Szeregowy

### Instalacja dodatkowych bibliotek Arduino

Kiedy zapoznasz się z oprogramowaniem Arduino i jego wbudowanymi funkcjami, będziesz chciał rozszerzyć możliwości Arduino dzięki dodatkowym bibliotekom.

### Czym są biblioteki?

Biblioteki to zbiory kodu, które ułatwiają połączenie się z czujnikiem, wyświetlaczem oraz modułami. Na przykład, wbudowana

biblioteka LiquidCrystal pozwala na łatwą komunikację z wyświetlaczem LCD. Korzystając z Internetu, znajdziesz setki innych dodatkowych bibliotek do pobrania. Wbudowane biblioteki, a także kilka innych, są wymienione w samouczku. Aby móc korzystać z dodatkowych bibliotek, trzeba je wcześniej zainstalować.

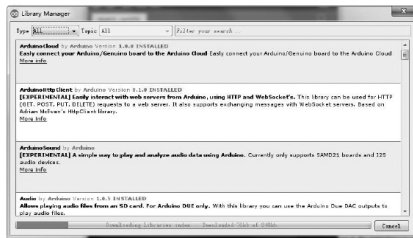
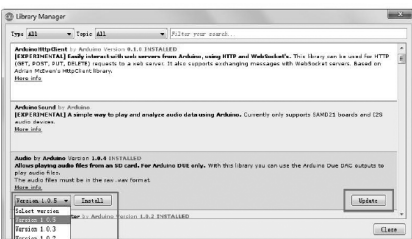
## Jak zainstalować bibliotekę?

Z wykorzystaniem Menadżera Bibliotek:

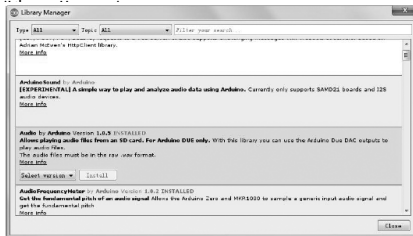
Aby zainstalować bibliotekę w Arduino IDE, możesz użyć Menadżera Bibliotek (dostępny od wersji 1.8.0 IDE). Otwórz środowisko IDE, następnie menu i kliknij Sketch/Szkic, Include Library/Dodaj bibliotekę -> Manage libraries /Zarządzaj bibliotekami.



Menadżer bibliotek uruchomi się i zobaczysz listę bibliotek, które zostały już zainstalowane lub są gotowe do instalacji. Dla przykładu, zainstalujemy bibliotekę Bridge. Przewiń listę, aby ją odnaleźć, następnie wybierz wersję biblioteki, którą chcesz zainstalować. Czasem tylko jedna wersja danej biblioteki jest dostępna. Jeśli nie pojawia się menu z wyborem wersji, nie przejmuj się, to normalne.



Kliknij przycisk Install/Zainstaluj i poczekaj aż IDE zainstaluje nową bibliotekę. Na pobieranie bibliotek z Internetu może mieć wpływ łączność internetowa. Kiedy proces instalacji dobiegnie końca, znacznik (tag) Installed/Zainstalowano powinien pojawić się obok biblioteki Bridge. Możesz zamknąć menadżera bibliotek



Teraz nową bibliotekę możesz znaleźć w menu Dodaj bibliotekę (Include Library). Jeśli chcesz dodać własną bibliotekę, skorzystaj z serwisu internetowego GitHub.

Możesz teraz znaleźć nową, dostępną bibliotekę w menu Dodaj Bibliotekę (Include Library)

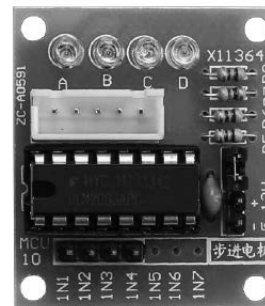
## Importowanie biblioteki .zip

Biblioteki są często rozpowszechniane jako pliki lub foldery .zip. Nazwa folderu jest również nazwą biblioteki. W folderze znajdują się pliki .cpp, plik .h, plik .txt ze słowami kluczowymi, folder z przykładami oraz inne niezbędne pliki dla funkcjonowania biblioteki. Począwszy od wersji 1.0.5, możliwe jest instalowanie bibliotek zewnętrznych w IDE. Nie należy rozpakowywać pliku zip z biblioteką. W Arduino IDE, wybierz opcję Szkic > Dodaj bibliotekę. W górnej części menu rozwijanego, wybierz opcję Add .ZIP Library/Dodaj bibliotekę .ZIP.



między przewodami. W ten sposób powinieneś znaleźć dwa przewody z taką samą rezystancją. Jeśli przewody twojego miernika są podłączone do przewodów, które nie są połączone z tą samą cewką, wskaże on nieskończenie dużą rezystancję.

## Układ sterujący ULN2003



### Opis

- Rozmiar: 42mm x 30mm
- Zastosowany układ: ULN2003, 500mA
- Diody LED A. B. C. D. wskazujące aktualną fazę silnika.
- Wtyczka zgodna z dwubiegunowym silnikiem krokowym.
- Oddzielne piny zasilania
- Pozostałe piny układu przeznaczone na inne indywidualne projekty.

Najprostszym sposobem, aby podłączyć jednobiegunowy silnik krokowy z Arduino, jest użycie płytki sterującej ULN2003. Układ posiada siedem sterowników tranzystorów Darlington, co równa się siedmiu tranzystorom TIP120. Moduł ULN2003 może przestrać do 500 mA na kanał, a jego spadek napięcia na oporze wewnętrznym wynosi 1V podczas pracy. Wyposażony jest także w wewnętrzne diody zaciskowe, które rozpraszają niepożądane skoki napięcia przy obciążeniu indukcyjnym. Aby sterować silnikiem krokowym, podłącz do napięcia każdą z cewek w odpowiedniej kolejności.

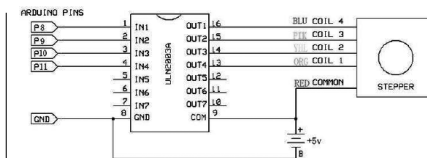
### Kolejność:

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-	-	-	-	-	-	-
3 YEL	-	-	-	-	-	-	-	-
2 PIK	-	-	-	-	-	-	-	-
1 BLU	-	-	-	-	-	-	-	-

Kolory przewodów ---> ruch zgodny z ruchem wskazówek zegara (Faza 1-2)

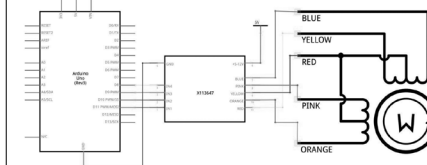
- 4 ORG (orange, pomarańczowy)
- 3 YEL (yellow, żółty)
- 2 PIK (pink, różowy)
- 1 BLU (blue, niebieski)

Poniżej pokazano, jak podłączyć jednobiegunowy silnik krokowy do czterech pinów sterujących za pomocą układu ULN2003.

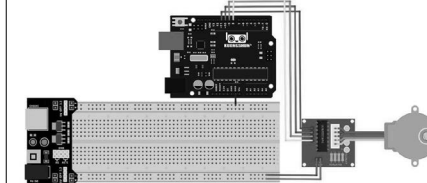


## Połączenia

### Schemat



### Schemat połączeń



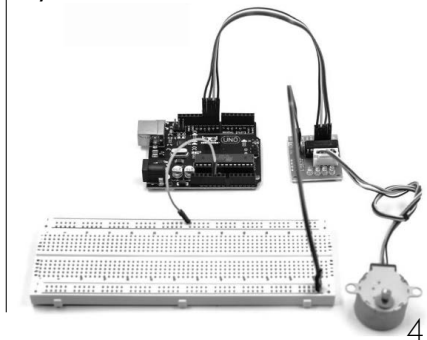
Wykorzystujemy 4 piny do sterowania silnikiem krokowym.

Piny 8-11 kontrolują silnik. Masę z płytki UNO podłączamy do silnika krokowego.

## Program

Po połączeniu według schematu, uruchom program – Lesson 23 Stepper Motor (Lekcja 23 – Silnik krokowy) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <Stepper> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.

### Przykład

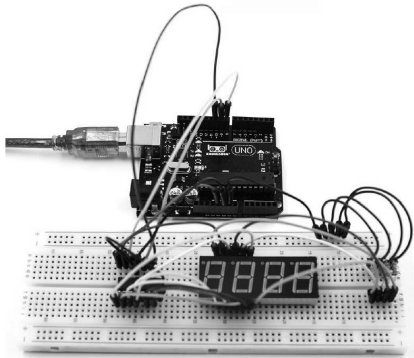




## Program

Po połączeniu według schematu, uruchom program – Lesson 22 Four Digital Seven Segment Display (Lekcja 22 – 4-cyfrowy wyświetlacz siedmiosegmentowy) z folderu ze szkicami i kliknij wgrać. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrania, jeśli wystąpią błędy.

## Przykład



## Lekcja 23. Silnik krokowy

### Opis

W tej lekcji dowiesz się, jak w łatwy i przyjemny sposób napędzić silnik krokowy. Silnik, z którego będziemy korzystać, posiada własny układ sterujący, co sprawia, że połączenie z płytką UNO staje się niezwykle proste.

### Niezbędne komponenty

- (1) x płytka UNO R3
- (1) x płytka stykowa 830
- (1) x układ sterujący ULN2003
- (1) x silnik krokowy
- (6) x żeńsko-męskie przewody połączeniowe
- (1) x męsko-męski przewód połączeniowy

### Opis komponentów

#### Silnik krokowy



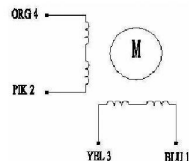
Silnik krokowy jest urządzeniem elektromechanicznym, które konwertuje impulsy elektryczne na ruchy mechaniczne. Wyposażony jest w wał silnikowy, który nie obraca się w sposób ciągły, z racji tego, że jego dyskretny ruch obrotowy sterowany jest za pomocą impulsów elektrycznych wysyłanych w odpowiedniej kolejności. Obrót silnika zależy od zastosowanych impulsów wejściowych. Kolejność w jakiej je wysłano jest bezpośrednio związana z kierunkiem obrotu wału silnikowego, a prędkość obrotu powiązana jest z kolejnością sygnali wejściowych. Na długość obrotu wpływa liczba wysłanych impulsów. Jedną z niepodważalnych zalet silnika krokowego jest łatwość z jaką można nim precyzyjnie sterować w pętli otwartej. Oznacza to, że informacja zwrotna o położeniu nie jest konieczna, więc nie ma potrzeby montowania czujników lub innych urządzeń, na przykład enkoderów optycznych. Położenie sledzimy na podstawie wysłanych impulsów wyjściowych.

Silnik krokowy 28BYJ-48. Specyfikacja:

- model: 28BYJ-48
- napięcie pracy: 5V
- Liczba faz: 4
- Przełożenie: 1/64
- Kąt obrotu kroku:  $5.625^\circ$
- Częstotliwość: 100Hz
- Rezystancja prądu stałego:  $50\Omega \pm 7\%$  (25 °C)
- Maksymalna częstotliwość biegu jałowego:  $> 600\text{Hz}$
- Maksymalna prędkość silnika na biegu jałowym:  $> 1000\text{Hz}$
- Moment obrotowy:  $> 34.3\text{mNm}$
- Moment sił tarcia: 600-1200 gf.cm
- Moment obrotowy, przy którym silnik może przesunąć obciążenie bez przyspieszania: 300 gf.cm
- Rezystancja izolowana:  $> 10\text{M}\Omega$  (500V)
- Izolacja elektryczna: 600VAC/1mA/1s
- Klasa izolacji: A
- Wzrost temperatury:  $< 40\text{K}$  (120Hz)
- Hałas:  $< 35\text{dB}$  (120Hz, bez obciążenia, 10 cm)

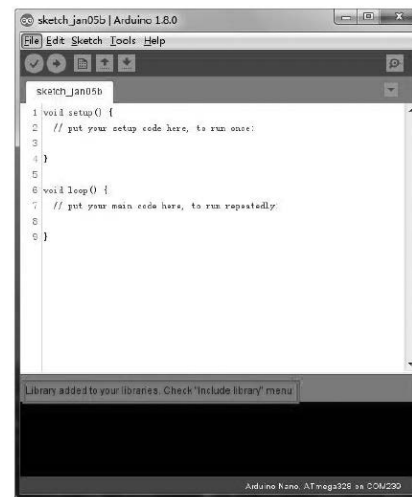
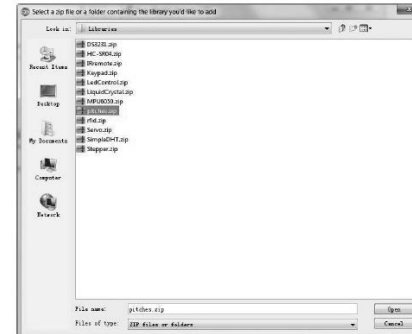
### Obwody połączeniowe:

#### Schemat połączeń



Dwubiegunowe silniki krokowe posiadają cztery przewody. W odróżnieniu od jednobiegunowych silników, nie mają wspólnych przewodów, ale są za to wyposażone w dwie niezależne grupy cewek. Silniki można również rozróżnić mierząc rezystancję

Zostaniesz poproszony o wybranie biblioteki, którą chcesz dołączyć. Wyszukaj plik i otwórz go.



Wróć do menu Szukaj > Dołącz bibliotekę. Powinieneś zobaczyć bibliotekę na dole menu rozwijanego. Plik jest gotowy do użycia w szkicu. Plik zip został rozszerzony w folderze biblioteki, w katalogu szkiców Arduino. **NB: Biblioteka będzie dostępna w szkicach, ale nie jej przykłady. Zobacysz je w Plik > Przykłady dopiero kiedy uruchomisz IDE ponownie.**

Te sposoby są stosowane najczęściej. Podobnie można postąpić w przypadku systemów MAC i Linux. Poniższa metoda, która służy jako alternatywa do wcześniejszych propozycji, jest rzadko stosowana i można ją pominąć, jeśli nie ma takiej potrzeby.

### Instalacja ręczna

Aby zainstalować bibliotekę ręcznie, po pierwsze, należy zamknąć aplikację Arduino. Następnie, trzeba rozpakować plik ZIP zawierający bibliotekę. Przykład: Jeśli instalujesz bibliotekę ArduinoParty,

rozpakuj plik ArduinoParty.zip. Powinien zawierać folder o tej samej nazwie, w którym znajdują się pliki ArduinoParty.cpp i ArduinoParty.h. (Jeśli plików o rozszerzeniach .cpp i .h nie ma w folderze, musisz stworzyć nowy. W tym przypadku, utwórz taki o nazwie „ArduinoParty” i przenieś do niego wszystko, co znajdowało się w pliku ZIP.) Przenieś folder ArduinoParty do tego folderu (plik z twoimi bibliotekami). Prawdopodobnie, w systemie Windows, ścieżka dostępu będzie wyglądała następująco: Moje dokumenty\Arduino\biblioteki. Dla użytkowników systemu Mac, ścieżka dostępu prawdopodobnie będzie wyglądała tak: Dokumenty/Arduino/biblioteki. W systemie Linux, folder biblioteki będzie znajdował się w szkieletniku.

Twój plik bibliotek Arduino powinien wyglądać następująco (Windows):

My documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp  
My documents\Arduino\libraries\ArduinoParty\ArduinoParty.h  
My documents\Arduino\libraries\ArduinoParty\examples

Lub (Mac i Lunux):

Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp  
Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h  
Documents/Arduino/libraries/ArduinoParty/examples

...

W folderze, oprócz plików .cpp i .h, możesz zobaczyć również inne pliki. Upewnij się, że tam pozostały. (Biblioteka nie zadziała, jeśli zamieścisz pliki .cpp i .h bezpośrednio w folderze bibliotek lub innym miejscu. Przykład: Dokumenty\Arduino\biblioteki\ArduinoParty\ArduinoParty\ArduinoParty.cpp nie będą działały).

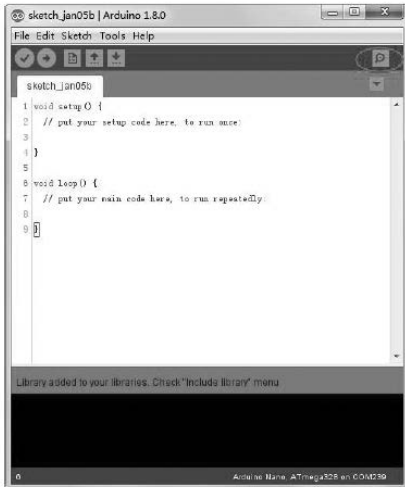
Uruchom ponownie aplikację Arduino. Upewnij się, że nowa biblioteka pojawiła się w menu Szukaj > Importuj bibliotekę. To wszystko! Udało Ci się zainstalować bibliotekę!

### Monitor Szeregowy Arduino (Windows, Mac, Linux)

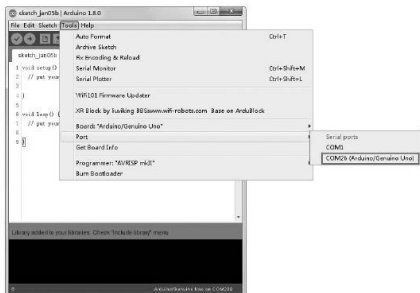
Zintegrowane Środowisko Programistyczne (IDE) to aplikacja służąca do programowania platformy Arduino. Ponieważ praca z terminalem jest nieodłączną częścią Arduino i innych mikrokontrolerów, firma zdecydowała się na włączenie do zestawu terminalu wraz z oprogramowaniem. W środowisku Arduino nazywany jest Monitorem Szeregowym.

### Tworzenie połączenia.

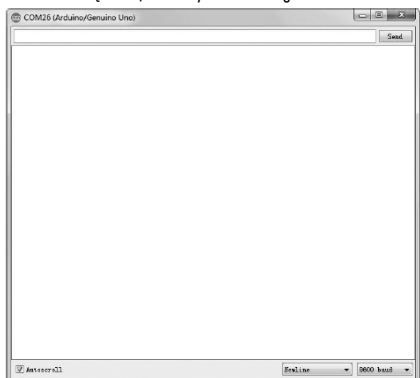
Monitor szeregowy jest kompatybilny z każdą wersją Arduino IDE. Aby go otworzyć, należy kliknąć jego ikonę.



Sposób w jaki wybieramy port, który chcemy uruchomić, jest taki sam, jak w przypadku portu do pobrania kodu Arduino. Wybierz opcję Narzędzia/Tools-> Monitor portu szeregowego/Serial Port, a następnie wybierz właściwy port. **Wskazówka: wybierz ten sam port COM, który wybrałeś w menadżerze urządzeń.**

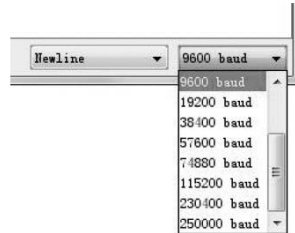


Po otwarciu się okna, zobaczysz coś takiego:



## Ustawienia

Monitor szeregowy ma ograniczone opcje, ale są wystarczające by sprostać twoim potrzebom w zakresie komunikacji szeregowej. Pierwszą opcją, którą możesz zmienić, jest szybkość transmisji.



Na samym końcu, możesz włączyć lub wyłączyć auto przewijanie (Autoscroll) poprzez zaznaczenie lub odznaczenie okienka w lewym dolnym rogu.



## Zalety

Monitor portu szeregowego jest świetnym i szybkim sposobem na stworzenie połączenia z Arduino. Jeśli już pracujesz z Arduino IDE, nie ma potrzeby otwierania oddzielnego terminalu w celu wyświetlenia danych.

## Wady

Mala ilość opcji nie spełnia wszystkich oczekiwań wobec monitora szeregowego, a w przypadku zaawansowanej komunikacji szeregowej, dostępne opcje mogą być niewystarczające.

## Lekcja 2. Migająca dioda

### Opis

W tej lekcji wyjaśnimy, jak zaprogramować mikrokontroler UNO R3, tak aby znajdująca się na nim dioda LED zaczęła migać, a także jak pobrać programy w kilku prostych krokach.

### Potrzebne komponenty:

- (1) x kuangshun UNO R3

### Zasada

Płytką z mikrokontrolerem posiada rzędy łączników wzdłuż obu stron. Wykorzystuje się je do podłączania urządzeń elektronicznych oraz rozszerzeń, które zwiększają możliwości płytki. Wyposażona jest również w jedną diodę LED, którą można kontrolować za pomocą odpowiedniego kodu, zwanego szkicem. Dioda jest wbudowana w płytkę UNO R3 i często oznaczana się ją jako „L”.

74HC595 pin	Seven shows remarkable control pin (stroke)
Q0	7 (A)
Q1	6 (B)
Q2	4 (C)
Q3	2 (D)
Q4	1 (E)
Q5	9 (F)
Q6	10 (G)
Q7	5 (DP)

**Krok pierwszy:** Podłącz rejestr przesuwny 74HC595.

Najpierw podłącz zasilanie i masę: VCC (pin16) oraz MR (pin 10) podłącz do 5V GND (pin8) i OE (pin 13) do masy. Połączenia pinów DS<sub>+</sub>, ST\_CP oraz SH\_CP: DS (pin 14) podłącz do pinu 2 na płytce UNO R3 (poniżej żółtej linii), ST\_CP (pin 12, latch pin) podłącz do pinu 3 na płytce UNO R3 (poniżej niebieskiej linii), SH\_CP (pin 11, clock pin) podłącz do pinu 4 na płytce UNO R3 (poniżej białej linii).

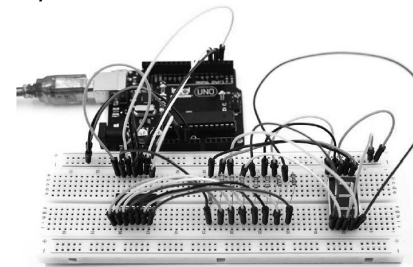
**Krok drugi:** Podłącz wyświetlacz siedmiosegmentowy.

Piny 3,8 wyświetlacza podłącz do GND na płytce UNO R3 (Tutaj wykorzystujemy wspólną katodę, jeśli korzystasz ze wspólnej anody, podłącz piny 3,8 do zasilania +5V na płytce UNO R3). Odnosząc się do tabeli powyżej, podłącz rejestr przesuwny 74HC595 Q0 ~ Q7 do odpowiedniego pinu (A ~ G oraz DP) siedmiosegmentowego wyświetlacza, a następnie każde jego wyprowadzenie do rezystora 220 omów.

### Program

Po połączeniu według schematu, uruchom program – Lesson 21 74HC595 And Segment Display (Lekcja 21 – Rejestr przesuwny 74HC595 i wyświetlacz segmentowy) z folderu ze szkicami i kliknij wgrać. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy.

### Przykład



## Lekcja 22. 4-cyfrowy wyświetlacz siedmiosegmentowy

### Opis

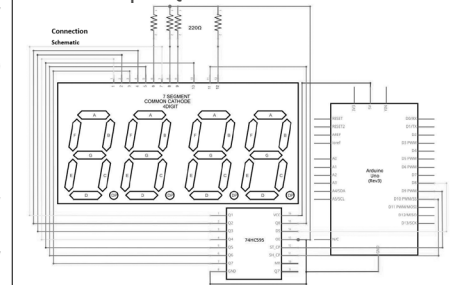
W tej lekcji nauczysz się, jak korzystać z 4-cyfrowego wyświetlacza siedmiosegmentowego. Kiedy używasz 1-cyfrowego wyświetlacza siedmiosegmentowego, zwróć uwagę na fakt, że jeśli opiera się na wspólnej anodzie, jej pin jest podłączony do źródła zasilania, a jeśli opiera się na wspólnej katodzie, jej pin podłączony jest do masy (GND). Kiedy pracujesz z 4-cyfrowym wyświetlaczem, wspólna anoda lub katoda jest potrzebna do kontroli nad tym, która cyfra jest wyświetlana. Mimo że działa tylko jedna cyfra, cecha wzroku, określaną również mianem bezwładności wzroku, sprawia, że nie dostrzegamy prędkości z jaką zmieniają się interwały.

### Niezbędne komponenty

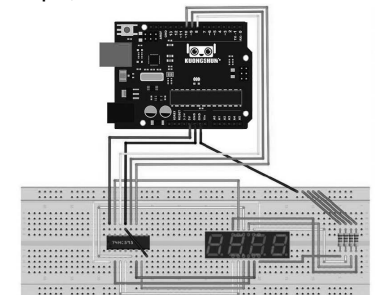
- (1) x płytka UNO R3
- (1) x płytka stykowa 830
- (1) x rejestr przesuwny 74HC595
- (1) x 4-cyfrowy wyświetlacz siedmiosegmentowy
- (4) x rezystor 220 ohm
- (23) x męsko-męskie przewody połączeniowe



### Schemat połączeń



### Schemat połączeń



# Lekcja 21. Rejestr przesuwny 74HC595 i wyświetlacz segmentowy.

## Opis

Tym razem wykorzystamy wiedzę z lekcji 18, 25 oraz 20 i użyjemy rejestru przesuwnego do sterowania wyświetlaczem segmentowym, który wyświetla cyfry od 9 do 0.

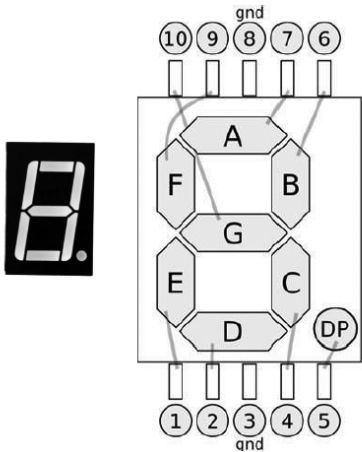
## Niezbędne komponenty

- (1) x płytka UNO R3
- (1) x płytka stykowa 830
- (1) x rejestr przesuwny 74HC595
- (8) x rezystor 220 ohm
- (26) x męsko-męskie przewody połączeniowe

## Opis komponentów

### Wyświetlacz siedmiosegmentowy

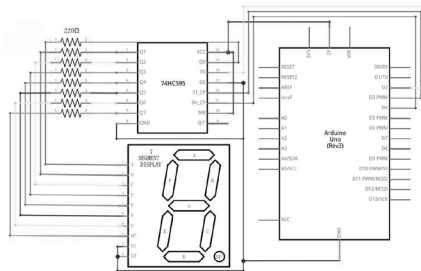
Poniżej pokazano siedmiosegmentowy schemat pinów.



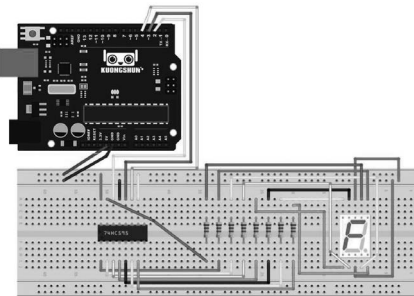
Każda z dziesięciu cyfr, od 0 do 9, odpowiada jednemu segmentowi (poniższa tabela odpowiada standardowemu urządzeniu z siedmiosegmentowym wyświetlaczem ze wspólną katodą. Jeśli korzystasz z urządzenia o wspólnej anodzie, każde 1 0 0 powinno być zastąpione przez 1):

Display/digital	dp	a	b	c	d	e	f	g
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

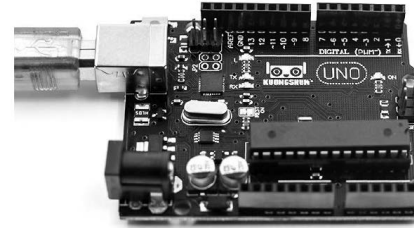
### Schemat połączeń



### Schemat połączeń



Poniższa tabela pokazuje, które wyprowadzenia wyświetlacza korespondują z tymi, które znajdują się na rejestrze przesuwym 74HC595:

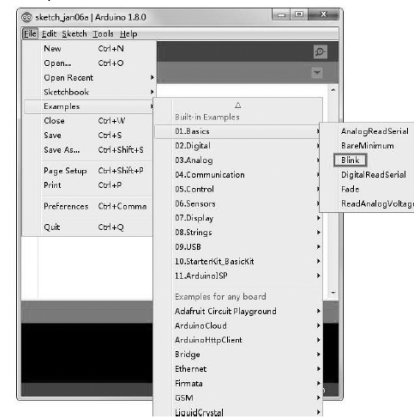


Zdarza się, że dioda na płytce UNO R3, oznaczona jako „L”, miga po podłączeniu jej do wtyczki USB. Dzieje się tak, ponieważ płytki zazwyczaj wysyłane są z już uprzednio zainstalowanym szkicem „Migaj”. W tej lekcji przeprogramujemy płytkę UNO R3, pisząc własny program „Migaj”, i zmienimy częstotliwość migania.

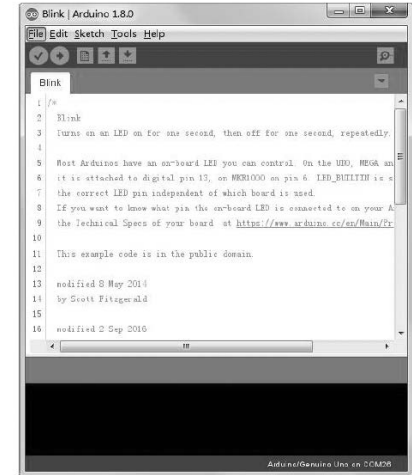
W Lekcji 0 pokazaliśmy, jak zainstalować Arduino IDE oraz jak znaleźć odpowiedni port szeregowy w celu podłączenia płytki z mikrokontrolerem UNO R3. Teraz nadszedł czas, aby przetestować to połączenie i zaprogramować płytkę.

Arduino IDE posiada znaczną kolekcję przykładowych szkiców do wykorzystania, włączając w to szkic, który sprawia, że dioda „L” miga.

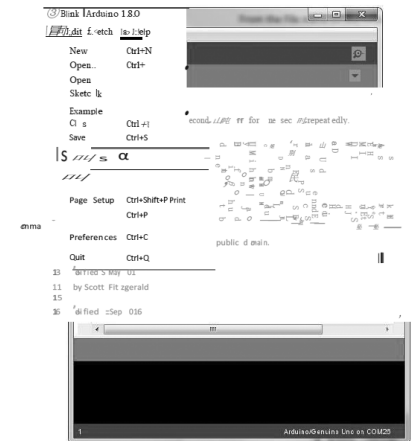
załaduj szkic Migaj/Blink, który znajduje się w menu systemu IDE, klikając File/Plik > Examples/Przykłady > 01. Basics/0.1.Podstawowe.

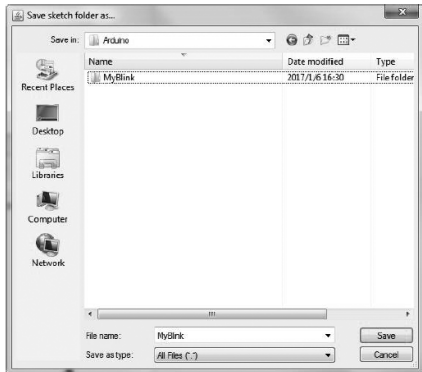


Kiedy otworzy się okno poglądowe, powiększ je, tak abyś mógł zobaczyć cały szkic.



Przykładowe szkice, które znajdują się w Arduino IDE, są tylko do odczytu. To oznacza, że możesz je wgrać na płytkę UNO R3, ale jeśli je edytujesz, nie będziesz mógł ich zapisać jako ten sam plik. W tej lekcji wprowadzimy zmiany w przykładowym programie, więc pierwszy krok, to zapisanie własnej kopii. W menu File/Plik wybierz opcję Save as/Zapisz jako i zapisz plik pod nazwą „MyBlink”.





Zapisałeś swoją kopię w szkiełkowniku. To oznacza, że jeśli kiedyś będziesz chciał ją odnaleźć, wystarczy, że skorzystasz opcji File/ Plik > Sketchbook/Szkiełkownik.



Podłącz płytke Arduino do komputera za pomocą kabla USB i sprawdź, czy typ płytki (Board type) i port szeregowy (Serial port) są ustawione prawidłowo.

Uwaga: Typy płytek lub portów szeregowych mogą się różnić od tych pokazanych na zdjęciu. Jeśli korzystasz z Arduino Mega2560, wybierz typ Mega 2560, na tej postawie można dokonać także innych wyborów. Port szeregowy również może się różnić, poza COM26 pokazanym na zdjęciu, prawidłowymi portami będą również COM3 i COM4. Prawidłowy wzór portu szeregowego to COMX (Arduino XXX), który odpowiada kryteriom certyfikacji.

Arduino IDE pokazuje aktualne ustawienia płytki w dolnym rogu okna.



Kliknij przycisk Upload/Prześlij, czyli drugi przycisk od lewej na pasku narzędzi.



Jeśli spójrz na obszar statusu IDE, gdzie zobaczysz pasek postępu oraz serię wiadomości. Na początku pojawi się informacja Compiling Sketch/Kompilowanie szkicu. Oznacza to, że szkic konwertowany jest na format odpowiedni do wgrania na płytkę.



Następnie, status zmieni się na Uploading/Wgrywanie. Na tym etapie, diody LED na płytce powinny zacząć migać.



Na końcu status zmieni się na Done/Zakończono.



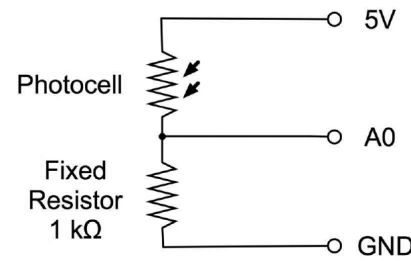
Kolejna wiadomość informuje nas, że szkic używa 928 bajtów z dostępnych 32,256. Po wiadomości Compiling Sketch/Kompilowanie szkicu może pojawić się komunikat o błędzie:



To oznacza, że płytka jest niepołączona lub że nie zainstalowano sterowników lub że wybrano zły port szeregowy. Wtedy należy wrócić do Lekcji 0 i upewnić się, że wszystko zostało poprawnie zainstalowane lub/i podłączone. Kiedy wgrywanie zakończy się sukcesem, płytka powinna się zresetować i zacząć migać. Uruchom program.

Zwróć uwagę, że większa część szkicu składa się z komentarzy. To nie są rzeczywiste instrukcje kodu, a jedynie wyjaśnienia, jak program działa. Są przeznaczone tylko dla użytkownika.

Wszystko między /\* i \*/ jest komentarzem, który informuje nas o przeznaczeniu danego szkicu.

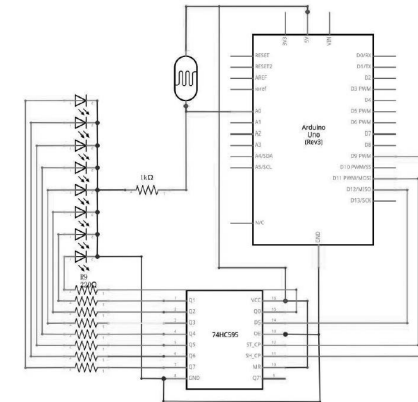


Rezystor i fotokomórka, kiedy są połączone razem, działają jak potencjometr. Gdy światło jest bardzo mocne, opór fotokomórki jest niższy w porównaniu do oporu rezystora. Dzieje się tak, gdy potencjometr jest podkręcony do maksimum. Kiedy fotokomórka znajduje się w słabym świetle, jej rezystancja rośnie ponad poziom rezystora 1000Ω, którego użyliśmy podczas tej lekcji. Co sprawia wrażenie, że potencjometr jest zwrócony w stronę masy.

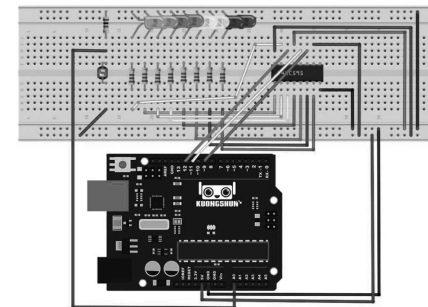
Wgraj poniższy szkic i spróbuj zakryć fotokomórkę palcem, a następnie wystawić ją na działanie mocnego światła.

## Podłączenia

### Schemat



### Schemat połączenia



## Program

Po połączeniu według schematu, uruchom program – Lesson 20 Photocell (Lekcja 20 – Fotokomórka) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrwania, jeśli wystąpią błędy.

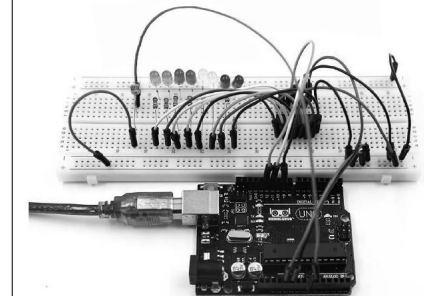
Pierwszą rzeczą, którą musimy zmienić, jest nazwa analogowego pinu. Zmieniamy jej nazwę z 'potPin' na 'lightPin', ponieważ potencjometr nie jest już podłączony.

Drugą istotną zmianą, którą musimy wykonać, jest zmiana linii szkicu, która przelicza, ile diod LED włączy.

`int numLEDSlit=reading/ 57; //all LEDs lit at 1k`  
(wszystkie LEDy świecą przy oporze 1k)

Tym razem, podzieliliśmy odczyt przez 57, a nie 114. Innymi słowy, podzieliliśmy go o połowę mniej w porównaniu do potencjometru, z którego wydzieliśmy 9 zakresów, od wszystkich diod zgaszonych po wszystkie 8 włączone. Uwzględnia to działanie rezystora 1000 Ω. Kiedy wartość rezystancji fotokomórki wyniesie tyle samo, co rezystora (1000 Ω), odczyt wyniesie 1023/2 = 511. Będzie to równoznaczne z włączeniem wszystkich diod, a bit (numLEDSlit) wyniesie 8.

### Przykład



Kiedy wiadomość została prawidłowo odebrana, widzimy to w następnej linii programu: `char ch = Serial.read();`

Jej zadaniem jest odczytanie znaków z bufora, a następnie usunięcie ich z niego i przypisanie do zmiennej „ch”. Zmienna „ch” określa „char” (z ang. „character”), ponieważ posiada tylko jeden znak przypisany do siebie.

Jeśli uważnie śledziłeś instrukcje znajdujące się w komunikacie w górnej części monitora, to pamiętasz, że ten znak przyjmie formę pojedynczej cyfry, od 0 do 7, lub litery „x”.

Instrukcja warunkowa „if” w następnym wierszu weryfikuje, czy jest to pojedyncza cyfra, sprawdzając, czy wartość „ch” jest większa lub równa 0 i mniejsza bądź równa 7. Porównywanie znaków w ten sposób może wydawać się dziwne, ale jest to w pełni akceptowalna metoda. Każdy znak reprezentowany jest przez niepowtarzalną liczbę, zwaną kodem ASCII. Oznacza to, że kiedy porównujemy znaki używając `<=` i `=>`, tak naprawdę porównujemy kody ASCII.

Po przeprowadzeniu testu, przejdźmy do kolejnej linii:

```
int led = ch - '0';
```

Teraz pora na arytmetykę! Odejmuje cyfrę '0' od każdej wprowadzonej cyfry, więc jeśli ta cyfra jest '0', wtedy '0' - '0' jest równe 0. Jeśli jest nia '7', wtedy '7' - '0' równa się 7. Dzieje się tak, ponieważ w rzeczywistości to działanie przeprowadzamy na kodzie ASCII.

Skoro już znamy numer diody LED, którą chcemy włączyć, musimy ustawić odpowiedni bit w zmiennej „leds” i zaktualizować rejestr przesuwny.

```
bitSet(leds, led);
```

```
updateShiftRegister();
```

Dwa następne wiersze wysyłają wiadomość potwierdzającą do Monitora Szeregowego.

```
Serial.print("Turned on LED ");
```

```
Serial.println(led);
```

W pierwszej linii pojawia się `Serial.print`, a nie `Serial.println`. Wynika to z tego, że `Serial.print` nie zaczyna nowej linii po wydrukowaniu czegokolwiek, co znajduje się w jej parametrze. Korzystamy z tej funkcji w pierwszym wierszu, ponieważ drukujemy wiadomość w dwóch częściach, najpierw tę ogólną, „Turned on LED”, a następnie numer diody LED.

Numer diody znajduje się w wartości zmiennej „int”, a nie stanowi ciągu tekstowego. `Serial.print` może przesyłać ciągi tekstowe w cudzysłowie lub wartości zmiennej „int”, lub jakkolwiek inną formę zmiennej.

Po instrukcji warunkowej „if”, która zajmowała się kwestią odczytania pojedynczej cyfry, czas na drugi warunek, który sprawdzi, czy „ch” jest literą „x”.

```
if (ch == 'x')
```

```
{
```

```
leds = 0; updateShiftRegister();
```

```
Serial.println("Cleared");
```

```
}
```

Jeśli jest litera „x”, wtedy wszystkie diody zostają wyłączone i wysyłana jest wiadomość potwierdzająca.

## Lekcja 20. Fotokomórka

### Opis

W tej lekcji dowiesz się, jak zmierzyć intensywność światła używając wejścia analogowego, które stworzyłeś podczas lekcji 18. Poziom natężenia światła będzie wpływał na ilość diod LED, które w danym momencie będą świecić. Fotokomórka znajduje się w dolnej części płytki stykowej, potencjometr natomiast znajduje się powyżej.

### Niezbędne komponenty:

- (1) x płytko UNO R3
- (1) x płytko stykowa z 830 otworami
- (8) x dioda LED
- (8) x rezystor 220 Ω
- (1) x rezystor 1000 Ω
- (1) x rejestr przesuwny 74hc595
- (1) x fotorezystor (fotokomórka)
- (16) x męsko-męskie przewody połączeniowe



### Opis komponentów

#### Fotokomórka

Fotokomórka jest rodzajem fotorezystora, czasem nazywanym LDR. Jak sama nazwa wskazuje, fotokomórka zachowuje się jak rezystor z tą różnicą, że opór zmienia się zależności od ilości światła padającego na fotokomórkę. Rezystor dołączony do zestawu stawia opór o wartości 50 Ω w bardzo słabym świetle, a 500 Ω w intensywnym. Aby przekształcić te różnice się wartości w coś łatwiejszego do zmierzenia dla UNO, musimy zmienić je w napięcie elektryczne. Możemy tego dokonać poprzez połączenie fotokomórki z rezystorem.

Komentarze jednowierszowe zaczynają się od `//`. Wszystko, co napisane po tym znaku, jest traktowane jako komentarz.

```
Pierwszą linią kodu jest: int led = 13;
```

Jak wyjaśnia komentarz nad nim, kod nadaje nazwę złączu, do którego przymocowana jest dioda LED. W przypadku Arduino UNO i Leonardo jest to 13.

Potem widzimy funkcję konfiguracji (setup function). Komentarz informuje nas, że ona uruchamia się, gdy wciskamy przycisk reset, a także kiedy płytka zresetuje się z jakiegokolwiek innego powodu, np. po wgraniu szkicu.

```
void setup() {
```

```
// initialize the digital pin as an output. pinMode(led, OUTPUT);
```

```
}
```

Każdy szkic musi mieć funkcję konfiguracji oraz miejsce, gdzie można dodać własne instrukcje między znakami `{}` oraz `}`.

Tutaj widzimy tylko jedno polecenie, które, jak wyjaśnia nam komentarz, przekazuje płytce Arduino, że użyjemy pin LED jako wyjścia.

Szkic zawsze posiada również funkcję `loop`/pętla. W odróżnieniu od funkcji konfiguracji, która uruchamia się tylko raz, po resetcie, funkcja `loop` po wykonaniu polecenia uruchomi się ponownie.

```
void loop() {
```

```
digitalWrite(led,HIGH); // włączenie diody (WYSOKI to poziom napięcia) delay(1000);
```

```
// odczekaj sekunde
```

```
digitalWrite(led,LOW); // wyłączenie diody (NISKI poziom napięcia) delay (1000);
```

```
//odczekaj sekunde
```

```
}
```

Funkcja `loop` najpierw włącza pin LED (HIGH), odczekuje 1000 milisekund (1 sekundę), potem wyłącza pin LED i odczekuje sekundę. Teraz sprawimy, że dioda będzie migotała szybciej. Kluczem do tego jest zmiana parametrów między `()` przy poleceniu `delay`/odczekaj.

```
20 // the loop function runs over and over again forever
21 void loop() {
22   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the volt
23   delay(500) // wait for a second
24   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the vo
25   delay(500) // wait for a second
26 }
```

Czas oczekiwania podany jest w milisekundach, więc aby dioda migotała dwa razy szybciej, należy zmienić wartość z 1000 na 500. To sprawi, że okres czas zmieni się z sekundy na pół sekundy.

Wgraj szkic jeszcze raz i sprawdź, czy dioda miga szybciej.

## Lekcja 3. Diody LED

### Opis

W tej lekcji nauczysz się, jak zmienić jasność diod LED za pomocą rezystorów o różnych wartościach.

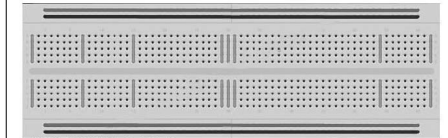
### Niezbędne komponenty:

- (1) x płytko UNO R3
- (1) x dioda LED 5mm – czerwona
- (1) x rezystor 220 ohm
- (1) x rezystor 1k ohm
- (1) x rezystor 10k ohm
- (2) x przewody połączeniowe męsko-męskie (M-M)

### Opis komponentów

#### Płyta Stykowa MB-102

Płyta stykowa umożliwiła szybkie tworzenie obwodów bez konieczności lutowania połączeń. Poniżej pokazano przykład jednego z takich obwodów.



Płyty stykowe mają różne rozmiary i konfiguracje. Najprostsze z nich, to po prostu siatka otworów w plastikowym bloku. W środku znajdują się metalowe blaszki, które tworzą przyłącze elektryczne pomiędzy otworami w płytkich rowkach. Wpięcie nóżek dwóch różnych komponentów w te same rowki, łączy je ze sobą tworząc obwód elektryczny. Głęboki kanał, biegnący przez środek, informuje nas, gdzie jest przerwa w połączeniu, co oznacza, że można podłączyć układ scalony z obu stron, nie tworząc połączenia między nimi. Niektóre płyty stykowe posiadają dwa rzędy otworów biegnących wzdłuż dłuższych boków płyty, które są odseparowane od głównej obudowy z otworami. Posiadają paski biegnące przez długość płytki, które dają możliwość podłączenia napięcia wspólnego.

Zwykle są w parach dla napięcia +5V i masy. Te paski nazywamy szynami i umożliwiają one podłączenie zasilania do wielu komponentów lub punktów na płycie.

Płytki stykowe idealnie nadają się do prototypowania, ale mają swoje ograniczenia, ponieważ połączenia odbywają się za pomocą ctyczek i są tymczasowe, nie są tak niezawodne jak te lutowane. Jeśli masz problemy z obwodami, przyczyną może być słabe połączenie na płycie stykowej.

## Diody LED

Diody LED są świetnymi wskaźnikami. Zużywają bardzo mało energii, co czyni je niezwykle trwałymi. Podczas tej lekcji, wykorzystasz najbardziej powszechną, czerwoną diodę LED o średnicy 5mm. Inne powszechne diody to te o średnicy 3mm i 10 mm. Uwaga, nie można podłączyć diody LED bezpośrednio do baterii lub źródła zasilania, ponieważ:

1. Dioda LED posiada dodatni i ujemny przewód, więc nie będzie świecić, jeśli zostanie umieszczona niewłaściwie.
2. Dioda LED musi być użyta z rezystorem, aby ograniczyć ilość dostarczanej do niej energii, inaczej się przepali!



Jeśli nie użyjesz rezystora wraz z diodą LED, ulegnie ona natychmiastowemu zniszczeniu, ponieważ będzie przez nią przepływać zbyt duża dawka energii, która spowoduje przegrzanie i zniszczenie złącza, w którym powstaje światło. Są dwa sposoby na określenie, które wyprowadzenie diody LED jest dodatnie, a które ujemne. Pierwszy sposób: wyprowadzenie dodatnie jest dłuższe. Drugi sposób: w miejscu, gdzie ujemne wyprowadzenie łączy się z korpusem, rant soczewki jest ścięty na płasko. Jeżeli zdarzy się, że posiadasz diody LED, które mają płaską powierzchnię obok dłuższego przewodu, możesz założyć, że dłuższe wyprowadzenie jest dodatnie.

## Oporniki

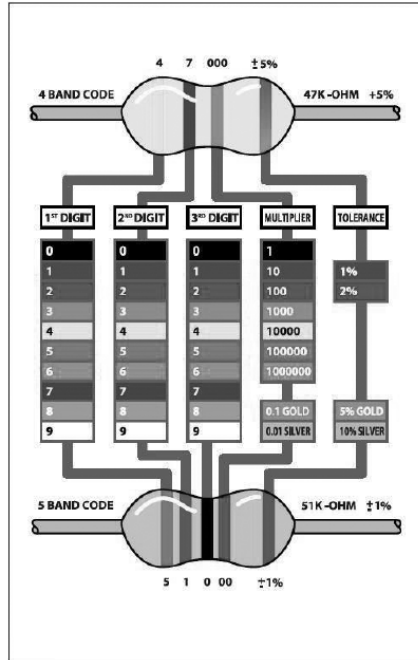
Jak sama nazwa wskazuje, oporniki stawiają opór przepływającej energii. Im wyższa wartość opornika, tym większy stawia opór i tym mniejsza ilość energii przez niego przepływa. Wykorzystamy to, aby kontrolować ilość energii elektrycznej przepływającej przez diodę LED, a co za tym idzie, jak jasno świeci.



Ale najpierw, kilka słów o rezystorach...

Jednostką oporu elektrycznego jest Om, zazwyczaj zapisywany jako grecka litera Omega ( $\Omega$ ). Ponieważ jeden Om ma bardzo małą wartość oporu (praktycznie nie stanowi żadnego oporu), opór oznaczamy również w k $\Omega$  (1,000  $\Omega$ ) i M $\Omega$  (1,000,000  $\Omega$ ). Nazywamy je kilo-omami i mega-omami.

Podczas tej lekcji, użyjemy rezystorów o różnych wartościach: 220 $\Omega$ , 1k $\Omega$  i 10k  $\Omega$ . Wyglądają tak samo, lecz mają paski w różnych kolorach, które określają wartość rezystora. Kod kolorystyczny składa się z 3 kolorowych pasków oraz jednego złotego paska na jednym z końców opornika.

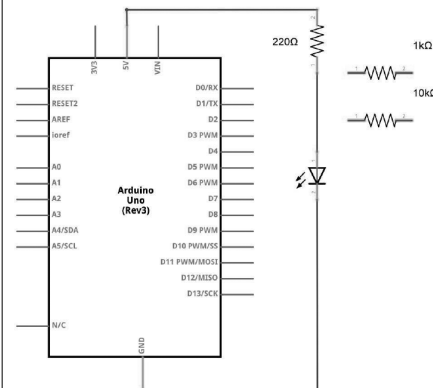


W przeciwieństwie do diod LED, rezystory nie mają dodatniego i ujemnego przewodu, więc można je podłączyć w dowolny sposób.

Jeśli ta metoda wydaje się zbyt skomplikowana, możesz odczytać kolory znajdujące się na rezystorach, aby określić ich wartość. Możesz również użyć cyfrowego multimetru.

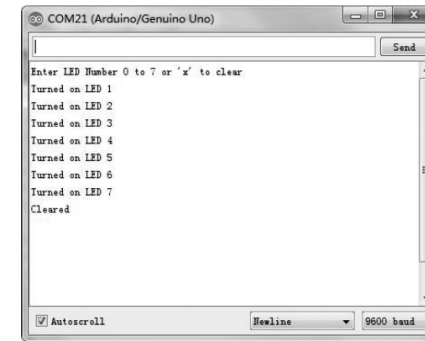
## Połączenie

### Schemat



lania wiadomości z komputera do płytki UNO (poprzez USB) oraz odbierania wiadomości z płytki. Wiadomość „Enter LED Number 0 to 7 or 'x' to clear” została wysłana przez Arduino i informuje nas, jakie komendy możemy wysłać: x, aby wyłączyć wszystkie diody LED lub liczbę diod, które chcemy włączyć (gdzie 0 to ostatnia dioda, 1 to następna dioda w kolejności, i tak aż do 7).

Spróbuj wpisać te komendy w górnej części Monitora Szeregowego, na tym samym poziomie co przycisk „send”. Kliknij przycisk po wpisaniu następującej treści: x 0 3 5. Wpisanie x nie będzie miało żadnego efektu, jeśli wszystkie diody LED są wyłączone, ale wraz z wpisaniem cyfry, odpowiednia dioda powinna się zaświecić. W ten sposób dowiesz się, że komenda dotarła do płytki. Okno powinno wyglądać tak, jak pokazano poniżej.



Wpisz x ponownie i kliknij „send”, aby wszystkie diody zgasty.

## Program

Po połączeniu według schematu, uruchom program – Lesson 19 The Serial Monitor (Lekcja 19 – Monitor Szeregowy) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy. Szkic opiera się na tym z którego korzystaliśmy w lekcji 18, dlatego wyjaśnimy tylko nowe fragmenty. Odwołanie się do całego programu w Arduino IDE może okazać się pomocne. W funkcji setup znalazły się trzy nowe wiersze na końcu: **void setup()**

```

{
  pinMode(latchPin, OUTPUT); pinMode(dataPin,
  OUTPUT); pinMode(clockPin, OUTPUT);
  updateShiftRegister(); Serial.begin(9600);
  (str 125)
  While (! Serial); // Wait until Serial is ready – Leonardo Serial.
  println("Enter LED
  Number 0 to 7 or 'x' to clear");
}

```

Na początku mamy komendę „Serial.begin(9600)”. Rozpoczyna ona komunikację szeregową z UNO poprzez połączenie USB. Wartość 9600 nazywa się „szybkością transmisji” połączenia i oznacza prędkość z jaką dane są wysyłane. Możesz ją zmienić na wyższą, ale pamiętaj, że wtedy trzeba też ustawić wartość monitara szeregowego na taką samą. Omówimy to w następnych krokach, więc na razie radzimy pozostać przy wartości 9600.

Wiersz zaczynający się od „while” upewnia się, że na drugim końcu połączenia USB znajduje się odbiorca, zanim Arduino rozpocznie komunikację, aby zapobiec wysłaniu wiadomości, której nie można wyświetlić. Ten fragment programu jest niezbędny tylko jeśli korzystasz z Arduino Leonardo, ponieważ Arduino UNO automatycznie resetuje płytkę po uruchomieniu Monitara Szeregowego, co nie dzieje się w przypadku Leonardo.

Ostatnia z nowych linii kodu w funkcji setup wysyła wiadomość, którą widzimy w górnej części Monitara Szeregowego. Funkcja pętla jest odpowiedzialna za realizację komend: **void loop()**

```

{
  if(Serial.available())
  {
    Char ch = Serial.read();
    if (ch >= '0' && ch <= '7')
    {
      Int led = ch - '0'; bitSet(leds, led);
      updateShiftRegister();
      Serial.print("Turned on LED "); Serial.println(led);
    }
    if (ch == 'x')
    {
      Leds = 0; updateShiftRegister();
      (str 126)
      Serial.println("Cleared");
    }
  }
}

```

Wszystko co zawiera funkcja pętla zawarte jest w instrukcji warunkowej – „if”. Jeśli dane sprawdzone poprzez wywołanie funkcji Serial.available nie będą „prawdziwe”, nic się nie stanie. Komunikat „true” / prawdziwe pojawi się, gdy dane zostały już wgrane do UNO i są gotowe do przetworzenia. Wiadomości przychodzące są przechowywane w buforze, więc funkcja Serial.available() wraca z informacją „true”, gdy bufor nie jest pusty.

znajduje się w zmiennej, po czym następuje półsekundy przerwy zanim „i” zwiększy wartość argumentu o jeden w celu włączenia diod. void loop]

```
{
  leds = 0; updateShiftRegister();
  delay(500);
  for (int i = 0; i < 8; i++)
  {
    bitSet(leds, i); updateShiftRegister();
    delay(500);
  }
}
```

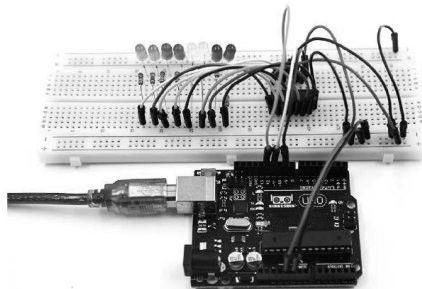
Funkcja „updateShiftRegister” ustawia stan pinu latch na niski, a potem przywołuje funkcję „shiftOut”, zanim przywróci stan wysoki. Wymaga to czterech parametrów, między innymi dwóch pinów, które posłużą kolejno dla „Data” i „Clock”. Trzeci parametr precyzuje od którego końca danych chcesz zacząć. My zaczniemy od najmniej znaczącego bitu określanego jako „Least Significant Bit” (LSB).

Ostatnim parametrem są rzeczywiste dane, które zostaną przesłane do rejestru przesuwającego. W tym wypadku dotyczy to zmiennej „leds”.

```
void updateShiftRegister()
{
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin,
  LSBFIRST, leds);
  digitalWrite(latchPin, HIGH);
}
```

Jeśli chciałbyś wyłączyć jedną z diod LED zamiast włączyć, mógłbyś przywołać podobną funkcję Arduino (bitClear) ze zmienną „leds”. Ustawia ona bity „leds” na 0, a potem wysyła komunikat „updateShiftRegister”.

#### Przykład



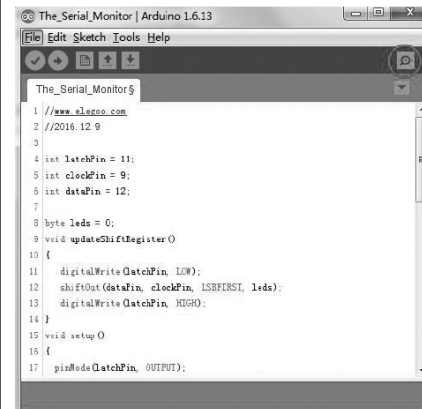
## Lekcja 19. Monitor Szeregowy

### Opis

Ta lekcja będzie uzupełnieniem lekcji 18, ponieważ wykorzystamy Monitor Szeregowy Arduino do kontrolowania diod LED. Monitor gra rolę łącznika między twoim komputerem a płytką UNO. Dzięki niemu można wysyłać wiadomości tekstowe, usuwać błędy z programu lub kontrolować UNO za pomocą klawiatury. Będziesz mógł na przykład wysyłać komendy z komputera, aby włączyć diody. Potrzebne będą dokładnie te same części co w poprzedniej lekcji. Schemat połączeń również będzie podobny.

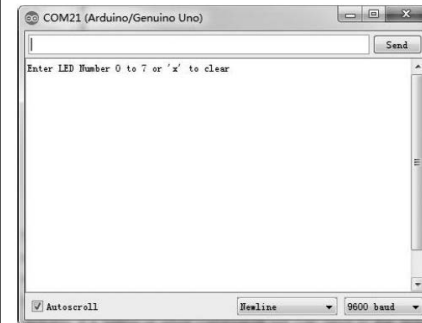
### Kroki

Po wgraniu poniższego szkicu do płytki UNO, kliknij przycisk położony najbardziej na prawo na pasku narzędzi Arduino IDE. Zakreślono go na zdjęciu.



Otwórz się poniższe okno.

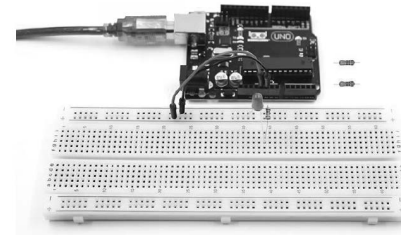
Kliknij przycisk Monitor Szeregowy, aby go włączyć. Szczegóły dotyczące korzystania z monitora szeregowego znajdziesz w lekcji 1.



To okno nazywamy Monitorem Szeregowym, który jest częścią środowiska Arduino IDE. Jego zadaniem jest umożliwienie przesy-

UNO jest pewnym źródłem napięcia o wartości 5V, którego użyjemy do zapewnienia energii diodom i rezystorowi. Po podłączeniu płytki przez kabel USB, nie musisz już nic robić. Dzięki rezystorowi 220Ω, dioda będzie się świeciła dosyć intensywnie. Kiedy jednak zamienimy 220Ω na 1kΩ, jasność zmniejszy się. Przy rezystorze 10kΩ, światło diody będzie ledwo widoczne. Wypnij czerwony przewód połączony z płyty stykowej, a następnie wykorzystaj go jako przełącznik, wpinając i wypinając go z otworu. Na tym etapie powinieneś zauważyć różnicę. W tej chwili, napięcie 5V przepływa przez jedną nóżkę rezystora, druga podpięta jest do anody, a kolejne wyprowadzenie diody podłączone jest do masy. Jeśli jednak przesuniemy rezystor, tak aby znalazł się za diodą (pokazano na obrazku poniżej), to dioda nadal będzie się świecić. Kiedy zechcesz umieścić rezystor 220Ω na swoim miejscu, nie musisz zwracać szczególnej uwagi na to, po której stronie diody to zrobić, najważniejszy jest fakt, że jest on gdziekolwiek podpięty.

#### Przykład



## Lekcja 4. Diody RGB LED

### Opis

Diody RGB LED są łatwym i przyjemnym sposobem na wprowadzenie odrobiny koloru do różnych projektów. Z racji tego, że budowa przypomina 3 zwykłe diody LED, korzysta się z nich w podobny sposób. Występują w dwóch wersjach: ze wspólną anodą lub wspólną katodą. Wspólna anoda o napięciu 5V znajduje się na wspólnym wyprowadzeniu, podczas gdy wspólna katoda przyłączona jest do masy. Podobnie jak w przypadku diod LED, trzeba będzie podłączyć kilka rezystorów w linii (łącznie 3), aby ograniczyć przepływ prądu. Szkic zaczniemy od diody LED w czerwonym kolorze, który następnie przejdzie w kolor zielony, potem niebieski, a następnie ponownie w czerwony. W ten sposób otrzymamy większość kolorów możliwych do uzyskania.

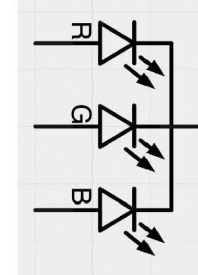
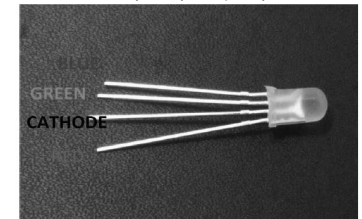
### Niezbędne komponenty

- (1) x płytka UNO R3
- (1) x płytka stykowa 830
- (4) x męsko-męskie przewody połączeniowe
- (1) x dioda RGB LED
- (3) x rezystory 220 ohm

## Opis komponentów

### RGB

Na pierwszy rzut oka, diody RGB (Red, Green, Blue) wyglądają jak zwykłe diody LED. Jednak standardowy zestaw diod LED składa się z trzech diod, jednej czerwonej, jednej zielonej i jednej niebieskiej. Kontrolując ich jasność, możemy uzyskać różne kolory. Kolory mieszają się podobnie jak farby na paletce, dostosowując jasność każdej z trzech diod. Bardziej skomplikowanym sposobem byłoby wykorzystanie rezystorów o różnej wartości, podobnie jak w lekcji 2. Na szczęście, płytka UNO R3 posiada analogową funkcję Write, której można użyć z pinami oznaczonymi jako ~, aby przesyłać daną ilość mocy do wybranej diody LED.



Na zdjęciu pokazano 4 wyprowadzenia diody. Każda osobna nóżka, odpowiadająca za kolor zielony, niebieski lub czerwony, nazywana jest anodą i łączy się zawsze z „+”. Katody podpinają się do „-” (masy). Jeśli odwrócisz kolejność, dioda nie zaświeci się.

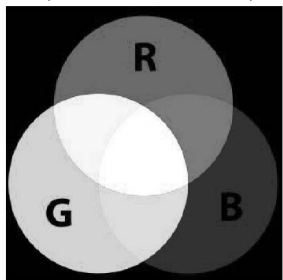
Wspólną katodą diody LED jest druga nóżka od płaskiej strony, jest ona również najdłuższa. Podłącza się ją do masy.

Każda dioda LED, która znajduje się wewnątrz, wymaga podłączenia do rezystora 220 ohm, aby zapobiec zbyt dużemu przepływowi prądu przez nią. Trzy anody (czerwona, zielona, niebieska) połączone są z pinem wyjścia płytki UNO poprzez rezystory.

### Kolor

Możemy dowolnie mieszać kolory poprzez różnicowanie stopnia jasności koloru czerwonego, zielonego i niebieskiego. Te trzy barwy postrzega ludzkie oko dzięki receptorom, a następnie mózg przetwarza informację o długości fal i sprawia, że widzimy dany kolor. Używając diod w tych kolorach oszukujemy ludzkie oko.

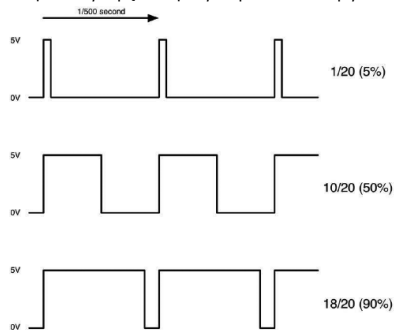
Tak jak dzieje się to w przypadku telewizorów, których wyświetlacze LCD składają się z czerwonych, zielonych oraz niebieskich kropek położonych obok siebie, które tworzą piksel.



Jeśli ustawimy jasność każdej z trzech diod LED na taki sam poziom, uzyskamy biały kolor. Jeśli wyłączymy niebieską diodę LED, ale nie zmienimy jasności pozostałych diod, uzyskamy żółty. Możemy kontrolować jasność każdej z trzech diod LED oddzielnie, aby otrzymać pożądany kolor. Czarny nie jest kolorem, a jedynie brakiem światła, wobec tego należałoby po prostu wyłączyć wszystkie diody.

## PWM w teorii

Skrót PWM oznacza Modułację Szerokości Impulsu (z ang. *Pulse Width Modulation*). Jest to technika, która pozwala kontrolować ilość dostarczanej energii elektrycznej. W tym wypadku, technika ta pozwala ustawić jasność każdej diody LED. Poniższy wykres pokazuje zmiany napięcia na jednym z pinów PWM na płytce UNO.

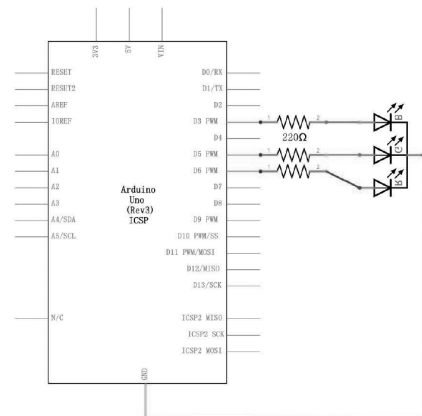


Mniej więcej co 1/500 sekundy wyjście PWM wytwarza impuls. Długość tego pulsu jest kontrolowana za pomocą funkcji `analogWrite`. Jeśli funkcja ustawiona jest na 0, `analogWrite(0)` nie wytworzy żadnego impulsu, a jeśli ustawiona jest na 255, funkcja `analogWrite(255)` wytworzy impuls, który trwa aż do pojawienia się następnego. Jeśli zatem w funkcji `analogWrite` ustawimy wartość między 0 a 255, wytworzymy impuls. Gdyby była to wartość 5, nasz impuls wysłałby jedynie 5% z całej energii jaką możemy dostarczyć. Jednak, jeśli amplituda wyjścia jest ustawiona na 5V przez 90% czasu; wtedy wypełnienie wyniesie

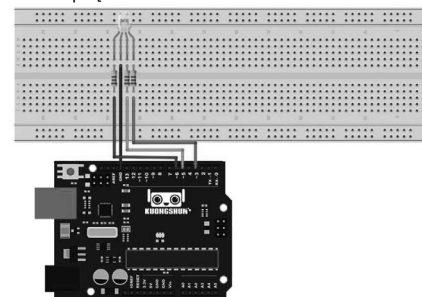
około 90% dostarczonej energii, a my nie będziemy w stanie dostrzec jak diody LED włączają się i wyłączają.

## Połączenie

### Schemat



### Schemat połączeń



## Program

Po połączeniu według schematu, uruchom program – Lesson 4 RGB LED (Lekcja 4 Diody LED RGB) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy.

Nasz program wykorzysta funkcję `FOR loops` (pętle `FOR`) do regulowania przejść między kolorami. Pierwsza pętla będzie regulowała przejście z czerwonego (RED) na zielony (GREEN).

Druga pętla będzie kontrolowała przejście z zielonego (GREEN) na niebieski (BLUE).

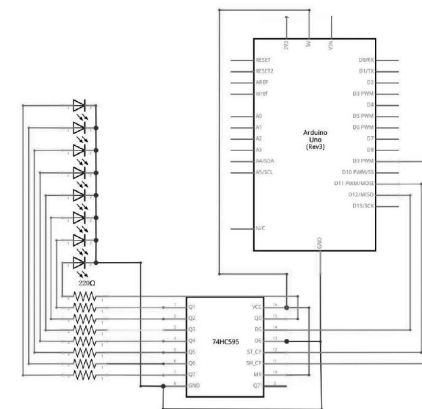
Wypróbuj ten szkic, a następnie zajmiemy się szczegółowymi kwestiami.

Szkic zaczyna od określenia, które piny zostaną wykorzystane do poszczególnych kolorów:

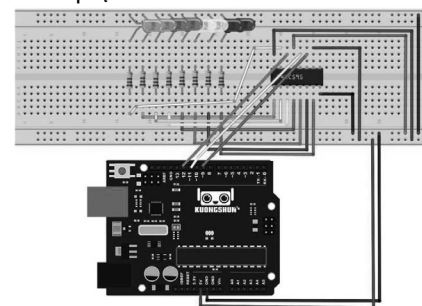
Szerokości Impulsu), aby za pomocą funkcji `analogWrite` kontrolować jasność diod LED. Pin ten posiada stan niski, więc podłączamy go do masy.

## Połączenia

### Schemat



### Schemat połączenia



Z racji tego, że mamy osiem rezystorów i osiem diod LED, trzeba będzie wykonać sporo połączeń. Najlepiej będzie zacząć od układu scalonego 74HC595, ponieważ z nim będzie najwięcej połączeń. Umieść go w taki sposób, żeby małe wycięcie w kształcie litery U skierowane jest w stronę górnej części płytki stykowej. Pin 1 tego układu mieści się po lewej stronie wycięcia. Wyjście cyfrowe 12 płytki UNO podłączamy do pinu 14 na rejestrze przesuwnym. Wyjście 11 Uno do pinu 12 rejestru przesuwnego, wyjście 9 Uno do pinu 11. Większość wyjść układu scalonego, z wyjątkiem jednego, znajduje się po lewej stronie. Dlatego, aby ułatwić pracę, umieszczone są tam też diody.

Po podłączeniu rejestru przesuwnego, pora na rezystory. Zwróć szczególną uwagę na fakt, że wyprowadzenia rezystorów nie mogą się ze sobą stykać. Pamiętaj o tym przed podłączeniem płytki Uno do zasilania. Jeśli trudno jest rozmieścić rezystory

w odpowiedniej odległości od siebie, możesz przyciąć ich nożki, tak aby leżały bliżej powierzchni płytki stykowej. Następnie umieść diody na płycie stykowej. Dłuższe nożki diod, które są dodatnie, należy umieścić w kierunku układu. Podłącz przewody połączeniowe według pokazanego wyżej schematu. Nie zapomnij o tym, który łączy pin 8 układu scalonego z GND na płycie stykowej. Wgraj szkic, który pojawi się w następnych krokach, aby sprawdzić, czy wszystko działa poprawnie. Każda dioda LED powinna się zaświecić po kolei, tak aby wszystkie były włączone, a następnie powinny zgasnąć i potem znowu zaświecić się.

## Program

Po połączeniu według schematu, uruchom program – Lesson 18 Eight LED with 74HC595 (Lekcja 18 – Ośmiem diod z rejestrze przesuwym 74HC595.) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy.

Najpierw musimy zdefiniować trzy piny z których będziemy korzystać. To właśnie te cyfrowe wejścia UNO podłączymy do pinów Latch, Clock i Data na układzie 74HC595.

`int latchPin = 11; int clockPin =`

`9; int dataPin = 12;`

Następnie definiujemy zmienną pod nazwą „leds”, aby ustalić kolejność z jaką diody będą się włączały lub wyłączały. Dane typu bajt reprezentują liczby używając ośmiu bitów. Każdy z bitów może być włączony lub nie, więc idealnie nadaje się do śledzenia, które diody są włączone, a które wyłączone.

`Byte leds = 0;`

Funkcja `setup` ustawia zdefiniowane przez nas piny na wyjścia cyfrowe. `void setup ()`

```
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}
```

Funkcja pętla początkowo wyłącza wszystkie diody nadając zmiennej „leds” wartość 0. Następnie nadaje komunikat „updateShiftRegister”, który wysyła informację do rejestru przesuwnego, aby wyłączył wszystkie diody. Wyjaśnimy działanie komendy „updateShiftRegister” w późniejszych krokach. Funkcja pętla zatrzymuje się na pół sekundy i zaczyna odliczać od 0 do 7, używając pętli „for” i zmiennej „i”. Za każdym razem wykorzystuje funkcję `Arduino „bitSet”`, która ustawia bity odpowiedzialne za daną diodę w zmiennej „leds”. Potem wysyła komunikat „updateShiftRegister”, aby diody zareagowały na to, co

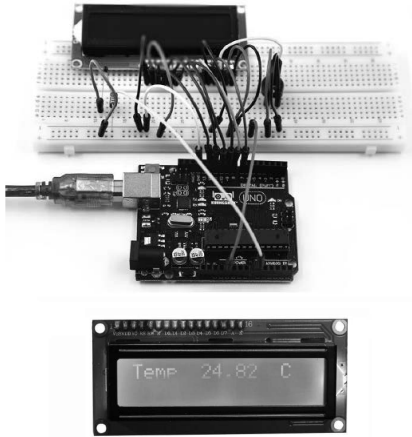


Wyświetlanie zmieniających się pomiarów może być trudne. Wynika to z tego, że odczyt nie zawsze składa się z tej samej liczby cyfr. Jeśli wartość trzycyfrowa zmieni się na dwucyfrową, to pierwsza liczba pierwszego pomiaru może pozostać na wyświetlaczu podczas wyświetlania następnego. Aby tego uniknąć, za każdym razem wpisz całą linię komendy LCD przy pętli.

```
lcd.setCursor(0, 0);
Temp C    "); lcd.setCursor(6, 0); lcd.print(TempF);
```

Ten dziwny komentarz przypomina o fackie, że na wyświetlaczu znajduje się 16 kolumn. Pozwoli to wydrukować ciąg o tej długości ze specjalnymi miejscami na pomiar rzeczywisty. Aby wypełnić te miejsca, ustaw kursor tam, gdzie pomiar powinien się znaleźć, a następnie wydrukuj go.

### Przykład



## Lekcja 18. Osiem diod z rejestrem przesuwным 74HC595.

### Opis

W tej lekcji dowiesz się, jak używać ośmiu dużych, czerwonych diod LED z płytką UNO, bez potrzeby korzystania z ośmiu wyjść. Mimo że możliwe teoretycznie byłoby podłączenie każdej diody LED z rezystorem do płytki Uno, w praktyce zabrakłoby na niej wyjść. Aby to się udało, trzeba by zrezygnować z podpięcia przycisków, czujników, serwo, itp. Na szczęście, istnieje inne rozwiązanie w postaci rejestru przesuwного 74HC595. Jest on wyposażony w osiem wyjść oraz trzy wejścia, które pozwolą stopniowo wprowadzać do niego dane. Rejestr sprawia, że diody

LED działają nieco wolniej (mogą się zmieniać 500,000 razy na sekundę zamiast 8,000,000 razy na sekundę), ale to nadal szybkość, której ludzkie oko nie dostrzega.

### Niezbędne komponenty

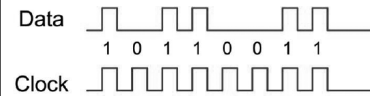
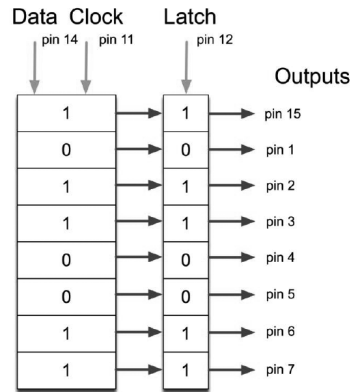
- (1) x płytka UNO R3
- (1) x płytka stykowa 830
- (8) x diod LED
- (8) x rezystorów 220 ohm
- (1) x rejestr przesuwny 74HC595.
- (14) x męsko-męskie przewody połączeniowe



### Opis komponentów

#### Rejestr przesuwny

8-bitowy układ scalony, który odbiera wysłaną szeregowo informację i przesyła ją do pinów. Stan wyjścia może być ustawiony jako 0 lub 1, aby włączyć lub wyłączyć bity, użyj pinów „Data” i „Clock” na układzie.



Pin „Clock” musi otrzymać osiem impulsów. Każdy z nich wysyła do wejścia rejestru, które ma ustawiony stan wysoki, wartość 1, w innym przypadku będzie to 0. Po otrzymaniu 8 impulsów, pin „Latch” kopiuje te wartości do rejestru zatraskowego. Jest to konieczny krok, aby diody LED migły prawidłowo po wprowadzeniu danych do rejestru przesuwного.

Układ scalony posiada także pin OE (Output Enable), który umożliwia jednoczesne włączenie lub wyłączenie wszystkich wyjść. Należy go podłączyć do Arduino, do pinu PWM (Modulacja

```
// Define Pins (określ piny)
```

```
#define BLUE 3
```

```
#define GREEN 5
```

```
#define RED 6
```

Następnie trzeba napisać funkcję setup. Jak już wiemy z poprzednich lekcji, funkcja setup uruchamia się tylko raz, po zresetowaniu Arduino. W przypadku tego szkicu, ta funkcja określi trzy piny, które posłużą jako wyjścia.

```
void setup()
{
  pinMode(RED, OUTPUT); pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH); digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
}
```

Zanim przyjrzymy się funkcji loop, zerknijmy na ostatnią funkcję w szkicu. Zmienne wartości.

```
redValue = 255; // choose a value between 1 and 255 to change the colour (tłum.: wybierz wartość między 1 a 255, aby zmienić kolor). greenValue = 0; blueValue = 0;
```

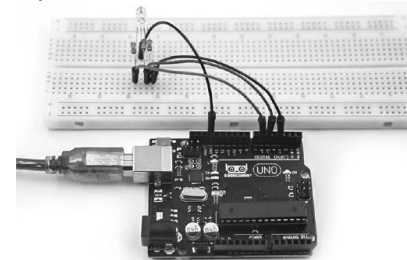
Ta funkcja przyjmuje trzy warunki, z czego każdy określa jasność jednej diody LED, czerwonej, zielonej lub niebieskiej. Za każdym razem wartość będzie między 0 a 255, gdzie 0 oznacza wyłączenie diody, a 255 maksymalną jasność. Następnie funkcja analogWrite dostaje komendę, aby ustawić jasność każdej diody. Funkcja loop pozwala nam ustawić jasność czerwonego, zielonego i niebieskiego koloru, a następnie odczekać sekundę przed zmianą koloru.

```
#define delayTime10 // fading time between colors
```

```
Delay(delayTime);
```

Spróbuj dodać własne kolory i podziwiaj efekt.

### Przykład



## Lekcja 5. Wejścia cyfrowe

### Opis

Podczas tej lekcji, nauczysz się, jak używać przycisków z wejściami cyfrowymi w celu włączenia lub wyłączenia diod LED.

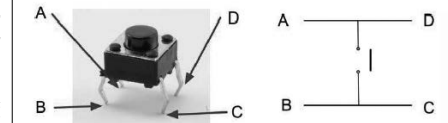
### Niezbędne komponenty:

- (1) x płytka UNO R3
- (1) x 830 płytka stykowa
- (1) x czerwona dioda LED 5mm
- (1) x rezystor 220Ω
- (2) x przyciski
- (7) x męsko-męskie przewody połączeniowe

### Opis komponentów

#### Przyciski

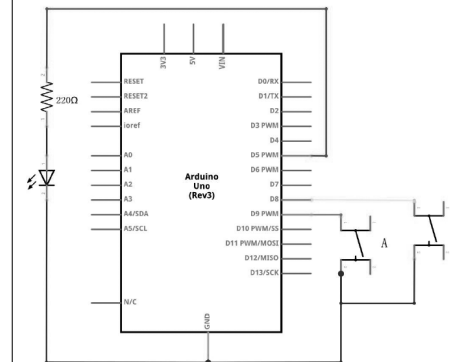
To bardzo proste komponenty. Kiedy naciskasz przycisk lub przesuwasz dźwignię, łączysz ze sobą dwa styki, a dzięki temu może przez nie przepłynąć prąd. Małe przełączniki dotykowe, które wykorzystamy w tej lekcji, posiadają cztery styki, co może być odrobinię dezorientujące.



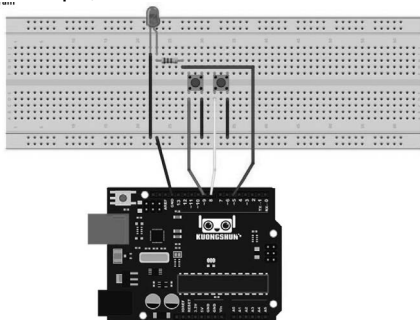
Tak naprawdę przełączniki posiadają tylko dwa połączenia elektryczne. Wewnątrz obudowy przycisków, wyprowadzenia B i C oraz A i D są ze sobą połączone.

### Połączenie

#### Schemat



## Schemat połączeń



Wyprowadzenia wychodzą z przeciwległych stron przycisku, mimo że ich obudowy są kwadratowe. Oznacza to, że piny będą w odpowiedniej odległości od siebie, tylko kiedy zostaną prawidłowo umieszczone na płytce stykowej. Pamiętaj, aby krótsza nóżka diody LED znajdowała się po lewej stronie.

## Program

Po połączeniu według schematu, uruchom program – Lesson 5 Digital Inputs (Lekcja 5 - Wejścia cyfrowe) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy.

Wgraj szkic do swojego UNO. Naciśnięcie lewego przycisku włączy diodę, natomiast wciśnięcie przycisku prawego, spowoduje jej wyłączenie. Pierwsza część szkicu definiuje trzy zmienne dla trzech pinów, których chcemy użyć. ledPin określa wyjście, a buttonApin odnosi się do przycisku w górnej części płyty stykowej. Funkcja 'buttonBpin' odnosi się do drugiego przycisku. Funkcja setup definiuje ledPin jako normalne wyjście, lecz teraz posiadamy dwa wejścia. W tym przypadku, ustawimy tryb pinMode, aby komenda 'INPUT\_PULLUP' wyglądała w ten sposób:

```
pinMode(buttonApin,INPUT_PULLUP); pinMode(buttonBpin,
INPUT_PULLUP);
```

Tryb INPUT\_PULLUP oznacza, że pin zostanie wykorzystany jako wejście, jeśli jednak nic innego nie jest do niego podłączone, to powinno zostać 'podciągnięte' do wartości HIGH. Innymi słowy, wartością wyjściową dla wejścia będzie HIGH, jeśli nie zostanie obniżona do wartości LOW poprzez użycie przycisku. To właśnie dlatego przyciski są uzziemione. Kiedy je naciskamy, łączymy pin wejściowy z masą (GND), powodując spadek wartości HIGH.

Wartość wejścia jest zazwyczaj HIGH, a schodzi do poziomu LOW tylko wtedy, kiedy wciskamy przycisk, więc trochę wbrew logice. Z tym problemem poradzimy sobie korzystając z funkcji loop.

```
void loop()
{
if (digitalRead(buttonApin) == LOW)
(
digitalWrite(ledPin, HIGH);
)
if (digitalRead(buttonBpin) == LOW)
```

## Lekcja 6. Aktywny brzęczyk

### Opis

W tej lekcji nauczysz się, jak generować dźwięk z wykorzystaniem aktywnego brzęczyka.

### Niezbędne komponenty:

- (1) x płytka UNO R3
- (1) x Aktywny brzęczyk
- (2) x damsko-męskie przewody DuPont

### Opis komponentów

#### Brzęczyk

Elektryczne brzęczki są zasilane prądem stałym i są wyposażone w wbudowany obwód. Są często stosowane w komputerach, drukarkach, kopiarkach, alarmach, zabawkach, elektronicznych częściach samochodowych, telefonach, zegarach, czasomierzach i innych produktach. Brzęczki dzielą się na aktywne i pasywne. Odwróć piny obu brzęczyków, tak abyś mógł je zobaczyć. Brzęczyk z zielonym obwodem, to brzęczyk pasywny, natomiast brzęczyk aktywny, to ten osłonięty czarną taśmą.

Różnica pomiędzy nimi polega na tym, że brzęczyk aktywny posiada wbudowaną wiązkę, która generuje dźwięk przy dopływie energii elektrycznej. Brzęczyk pasywny jej nie posiada, więc nie będzie wydawał dźwięku po przepuszczeniu przez niego prądu stałego. Brzęczki aktywne są dużo droższe od pasywnych z uwagi na liczne, wbudowane obwody oscylacyjne.



## Lekcja 17. Termometr

### Opis

W tej lekcji wykorzystamy wyświetlacz LCD do pokazania temperatury.

### Niezbędne komponenty

- (1) x płytka UNO R3
- (1) x moduł LCD1602
- (1) x rezystor 10k om
- (1) x termistor
- (1) x potencjometr
- (1) x płyta stykowa 830
- (18) x męsko-męskie przewody połączeniowe.

### Opis komponentów

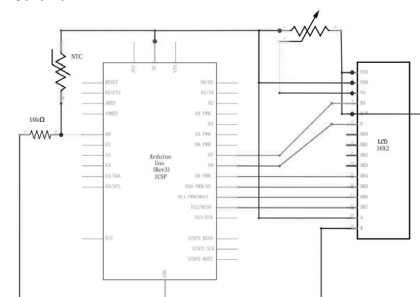
#### Termistor

Termistor to inaczej rezystor termiczny, czyli taki, którego rezystencja zmienia się wraz z temperaturą. Formalnie rzecz biorąc, każdy rezystor jest termistorem, ponieważ temperatura zawsze ma wpływ na rezystencję, ale często te zmiany są zbyt małe i trudne do zmierzenia. Rezystory termiczne zaprojektowane są tak, aby rezystencja była bardzo czuła na jej zmiany, np. w obrębie 100 omów na każdy stopień.

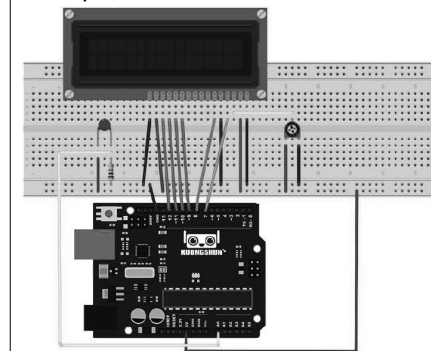
Wyróżniamy dwa typy: NTC, które służą do odczytywania temperatury, i PTC, które najczęściej są wykorzystywane jako bezpieczniki wielokrotnego użytku. Wraz ze wzrostem temperatury ich rezystencja rośnie, co powoduje zwiększony przepływ prądu, ale bezpieczniki nagrzewają się i przerywają obwód.

### Połączenia

#### Schemat



## Schemat połączeń



Układ połączeń opiera się na schemacie z lekcji 16, z tą różnicą, że kilka przewodów niedaleko potencjometru zostało przesuniętych. Rezystor 10kΩ oraz termistor są nowymi komponentami w tym projekcie.

## Program

Po połączeniu według schematu, uruchom program – Lesson 17 Thermometer (Lekcja 17 – Termometr) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <LiquidCrystal> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.

Szkic również opiera się na tym z lekcji 16. Wgraj program do Arduino. Następnie, odszukaj palcem czujnik ciepła i zaobserwuj, jak zmienia się pomiar temperatury. Nad komendą „lcd” można umieścić komentarz, co będzie przydatne, jeśli użyjemy innych pinów niż poprzednio:

```
// BS E D4 D5 D6 D7
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Funkcja loop jest tym razem bardziej interesująca, posłuży do przekonwertowania analogowego odczytu z czujnika temperatury na rzeczywistą temperaturę, a potem do ustalenia, jak pokazać pomiar na wyświetlaczu. Na początku skupimy się na obliczeniu temperatury:

```
int.
```

```
tempReading = analogRead(tempPin);
```

```
(...)
```

```
Float tempF = (tempC * 9.0)/ 5.0 + 32.0;
```

oraz rejestr instrukcji/komend, który przekazuje informacje dotyczące następných kroków.

**R/W:** tryb Odczytu / Zapisu.

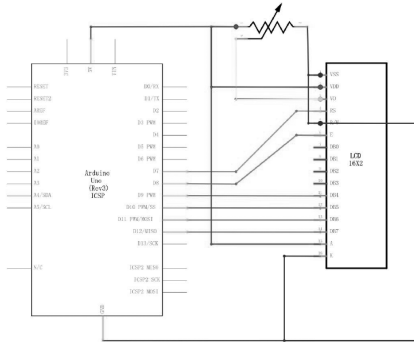
**E:** pin, który zezwala modułowi wyświetlacza na wykonanie odpowiednich instrukcji, po ustawieniu energii na niskim poziomie.

**D0-D7:** dane.

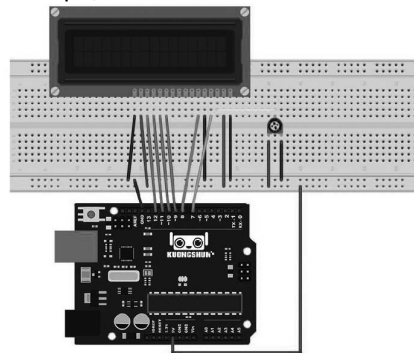
**A i K:** piny kontrolujące podświetlenie.

## Połączenia

### Schemat



### Schemat połączenia



Moduł LCD potrzebuje sześciu pinów Arduino, podłączonych do wejść cyfrowych, a także podłączenia do zasilania i masy. Wyrównanie wyświetlacza z górną częścią płytki stykowej pozwala w łatwy sposób odnaleźć piny, bez zbędnego liczenia, zwłaszcza gdy pierwszy wiersz płytki również znajduje się w jej górnej części. Nie zapomnij o dzielniku napięcia, długi żółty przewód podłączony do pinu 3 na module, który pozwala regulować kontrast.

## Program

Po połączeniu według schematu, uruchom program — Lesson 16 LCD Display (Lekcja 16 — Wyświetlacz LCD) z folderu ze szkicami

i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <LiquidCrystal> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.

Wgraj program Arduino, aby na wyświetlaczu pokazała się wiadomość „Hello, world”, a po niej liczba powyżej zera.

Pierwszą linią kodu jest:

```
#include<LiquidCrystal.h>
```

Dzięki temu Arduino wie, że chcemy skorzystać z biblioteki LiquidCrystal. Następna linijka wymaga naszej ingerencji, ponieważ musimy określić, które piny Arduino będą podłączone do wyświetlacza.

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Po wgraniu programu, upewnij się, że podświetlenie jest włączone. Obracaj pokrętkę potencjometru, dopóki nie zobaczysz wiadomości na wyświetlaczu.

Do funkcji setup przypisane są dwie komendy:

```
lcd.begin(16, 2);
```

```
lcd.print(„Hello, World!”);
```

Pierwsza mówi bibliotece LiquidCrystal, ile kolumn i wierszy ma pojawić się na wyświetlaczu, a druga wyświetla wiadomość tekstową.

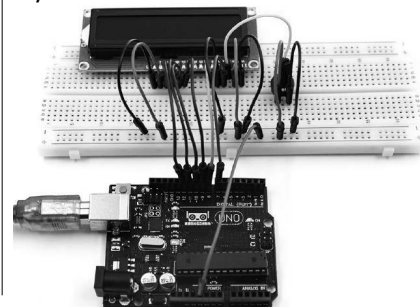
Funkcja loop/pętla również posiada dwie komendy:

```
lcd.setCursor(0, 1);
```

```
lcd.print(millis()/1000);
```

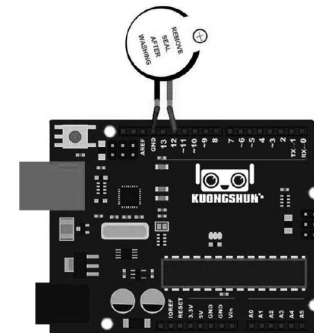
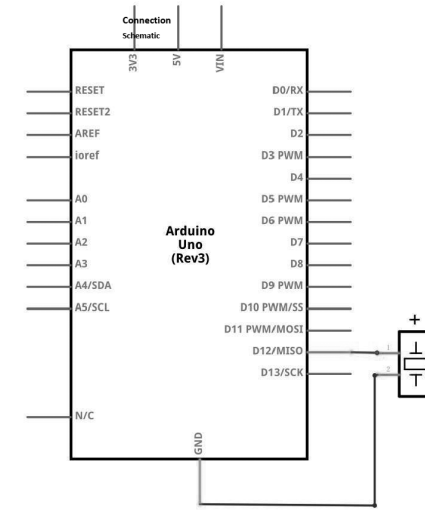
Pierwsza komenda ustawia pozycję kursora na kolumnę 0 i wiersz 1 (gdzie pojawi się kolejny tekst). Zarówno kolumny, jak i wiersze zaczynają się od 0, a nie 1. Druga komenda wyświetla liczbę milisekund od zresetowania Arduino.

### Przykład



## Połączenia

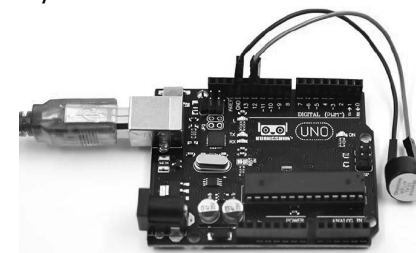
### Schemat



### Program

Po połączeniu według schematu, uruchom program — Lesson 6 Making Sounds (Lekcja 6 brzęczyk aktywny) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy.

### Przykład



## Lekcja 7. Pasywny brzęczyk

### Opis

W tej lekcji dowiesz się, jak używać pasywnego brzęczyka. Celem tego eksperymentu jest wytworzenie ośmiu różnych dźwięków, z czego każdy ma trwać 0,5 sekundy: Alt Do (532Hz), Re (587Hz), Mi (659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz), Do (1047Hz) (sopran).

### Niezbędne komponenty:

- (1) x płytka Uno R3
- (1) x pasywny brzęczyk
- (2) x żeńsko-męskie przewody połączeniowe DuPont

### Opis komponentów

#### Pasywny brzęczyk

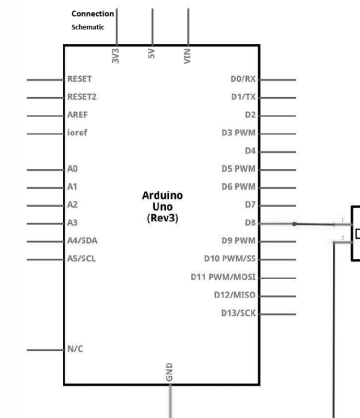
Brzęczyk działa za sprawą technologii PWM, która generuje dźwięk wprawiając w ruch powietrze, a dzięki zmianom częstotliwości wibracji, możliwe jest wyprodukowanie różnych dźwięków. Na przykład, wysyłając impuls 532Hz, otrzymujemy dźwięk Alt Do, wysyłając impuls 587Hz, otrzymujemy dźwięk w zakresie średnich tonów Re, a impuls 659Hz, generuje dźwięk Mi. Można również zagrać melodię.

Uwaga, upewnij się, że nie używasz funkcji analogWrite() do wygenerowania impulsu dla brzęczyka, ponieważ wyjście impulsu funkcji analogWrite() jest już ustawione (500Hz).

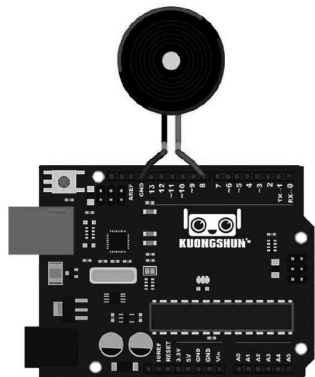


## Połączenia

### Schemat



## Schemat połączenia

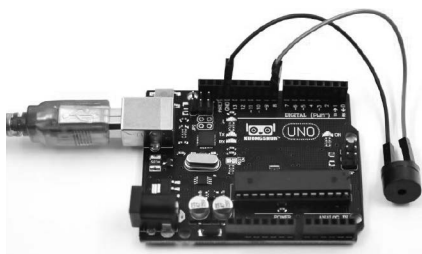


Połączenie z płytką UNO R3: czerwony przewód (dodatni) łączymy z pinem 8, a czarny przewód (ujemny) z masą (GND).

## Program

Po połączeniu według schematu, uruchom program – Lesson 7 Passive Buzzer (Lekcja 7 pasywny brzęczek) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <pitches> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.

## Przykład



## Lekcja 8. Czujnik przechyłu

### Opis

W tej lekcji dowiesz się, jak używać czujnika przechyłu w celu wykrycia niewielkiego przechylenia.

### Wymagane komponenty

- (1) x płytko UNO R3
- (1) x czujnik przechyłu
- (2) x żeńsko-męskie przewody połączeniowe



## Opis komponentów

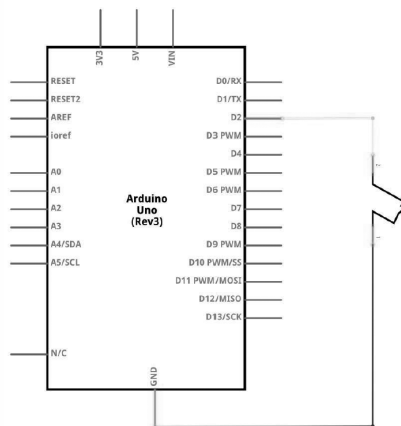
### Czujnik przechyłu

Czujniki przechyłu są małe, niedrogie, energooszczędne i łatwe w użyciu. Jeśli są używane prawidłowo, nie zużyją się. Dzięki tym właściwościom, często stosuje się je przy produkcji zabawek, różnych gadżetów czy urządzeń.

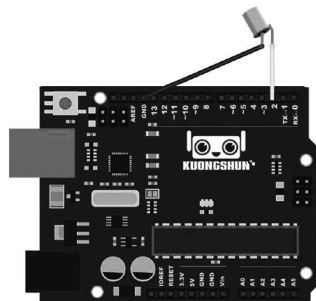
Zazwyczaj składają się z niewielkiej wgnęki (najczęściej o walcowatym kształcie, choć nie zawsze), wewnątrz której znajduje się obwód w formie kulek lub kropeł rtęci. Jeden koniec wgnęki posiada dwa elementy przewodzące (bieguny). Kiedy czujnik zostaje przechylony w tę stronę, kulki lub krople zwiernają obwód. Choć nie tak elastyczny i dokładny jak akcelerometr, ten czujnik pozwala wykryć ruch i przechylenia. Kolejną zaletą jest fakt, że większe czujniki mogą samodzielnie włączyć zasilanie. Z kolei akcelerometry wyprowadzają napięcie cyfrowe lub analogowe, które należy następnie przeanalizować za pomocą dodatkowych obwodów.

## Połączenia

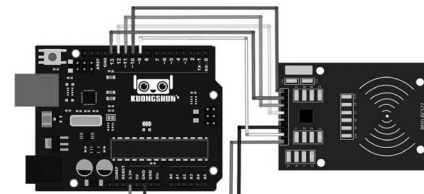
### Schemat



### Schemat połączenia



## Schemat połączeń



## Program

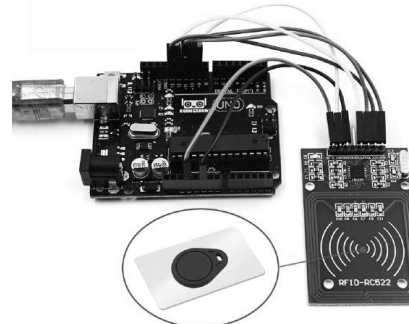
Po połączeniu według schematu, uruchom program. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <rfid> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.

Signal	W5C522 Reader/PCD Pin	Arduino Uno Pin	Arduino Mega Pin	Arduino Nano v3 Pin	Arduino Leonardo/Micro Pin	Arduino Pro Micro Pin
RST/Reset	RST	9	5	D0	RESET/ICSP-5	RST
SPI SS	SMA (CS)	10	S3	D10	10	10
SPI MISO	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15

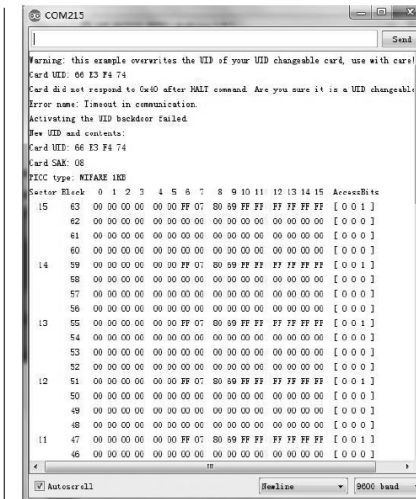
```
#define RST_PIN 9 // Configurable, see typical pin layout above
#define SS_PIN 10 // Configurable, see typical pin layout above
```

Lokalizacja pinów SPI może się różnić w zależności od chipów. Być może będziesz musiał dokonać niewielkich modyfikacji układu.

## Przykład



Po uruchomieniu monitora portu szeregowego, zauważysz następujące dane: Kliknij przycisk Monitor Szeregowy, aby go włączyć. Szczegóły dotyczące korzystania z monitora szeregowego znajdziesz w lekcji 1.



## Lekcja 16. Wyświetlacz LCD

W tej lekcji dowiesz się, jak podłączyć wyświetlacz LCD oraz jak nim sterować. Wyświetlacz posiada podświetlenie LED i może wyświetlać dwa wiersze, po 16 liter w każdym z nich. Widoczne są też na nim prostokąty, w których znajdują się litery złożone z pikseli, tak aby spełniał swoją podstawową funkcję pokazowania białego tekstu na niebieskim tle. Tym razem skorzystamy z programu, który znajduje się w bibliotece, a w następnej lekcji posłużymy się czujnikami, które mierzą temperaturę.

### Niezbędne komponenty

- (1) x płytko UNO R3
- (1) x moduł LCD1602
- (1) x potencjometr (10k)
- (1) x płyta stykowa 830
- (16) x żeńsko-męskie przewody połączeniowych.



### Opis komponentów

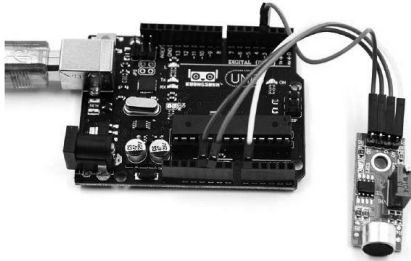
#### Wyświetlacz LCD1602

#### Piny:

- VSS:** pin podłączony do masy.
- VDD:** pin podłączony do zasilania +5V.
- VO:** dostosowanie kontrast.
- RS:** wybór rejestrów, który pozwala kontrolować, gdzie w pamięci wyświetlacza zapisywane są dane. Dostępne opcje: rejestr danych, który odpowiada za to, co pokazuje wyświetlacz

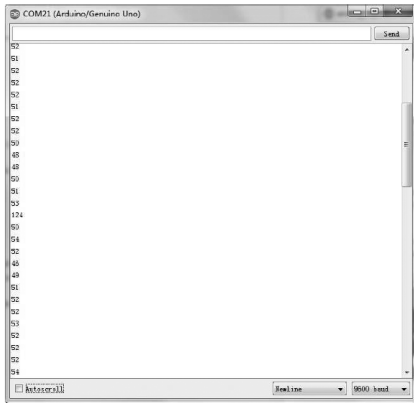
Ten moduł posiada dwa tryby wyjść sygnałowych, dla których napisano dwa kody: `digital_signal_output` i `analog_signal_output`. Program `digital_signal_output` działa, kiedy dźwięk osiągnie pewną wartość. Wtedy wysyła sygnał cyfrowy i ustawia pin `dig#11` na płytce na stan wysoki, a wskaźnik L podświetla się. Wartość wywołującą sygnał można zmienić regulując wrażliwość mikrofonu. Program `analog_signal_output` odczyta analogową wartość jaką przekaże moduł i wyświetli ją bezpośrednio na monitorze szeregowym.

### Przykład



Uruchom monitor, aby zobaczyć poniższe dane:

Kliknij przycisk monitor szeregowy, aby go włączyć. Więcej szczegółów dotyczących monitora szeregowego możesz znaleźć w Lekcji 1.



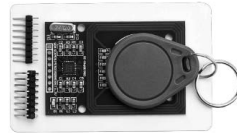
## Lekcja 15: Moduł RFID RC522

### Opis

Podczas tej lekcji nauczysz się, jak podłączyć moduł czytnika RC522 RFID do UNO R3. Moduł ten wykorzystuje szeregowy interfejs urządzeń peryferyjnych (SPI) do komunikacji z mikrokomputerami takimi jak Arduino, Raspberry Pi, beagle board, itd.

### Niezbędne komponenty

- (1) x płytka UNO R3
- (1) x Moduł RFID RC522
- (7) x damsko-męskie przewody połączeniowe DuPont (grafika)



### Opis komponentu

MFRC522 to wysoce zintegrowany czytnik, umożliwiający bezkontaktowy odczyt i zapis danych z urządzeń RFID na częstotliwości 13,56 MHz. Czytnik obsługuje standardy ISO 14443A/MIFARE®. Wbudowany przełącznik pozwala na sterowanie anteną czytnika, która pozwala na komunikację z kartami i przełącznikami w standardzie ISO/IEC 14443A/MIFARE®, które nie posiadają aktywnych obwodów. Odbiornik zapewnia stabilną oraz wydajną implementację obwodów demodulacji i dekodowania sygnałów kart lub przełączników kompatybilnych z ISO/IEC 14443A/MIFARE®. Część cyfrowa przejmuje zadania związanych z framingiem ISO/IEC 14443A oraz wykrywaniem błędów (Parity & CRC). MFRC522 obsługuje produkty MIFARE®Classic (np. MIFARE®Standard). Obsługuje również bezprzewodową komunikację przy wykorzystaniu wyższych prędkości transferu MIFARE®, do 848 kbit/s w obu kierunkach.

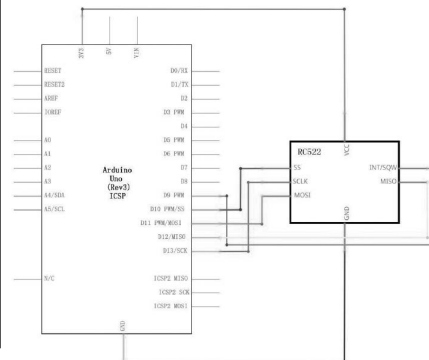
Zastosowano różne interfejsy hosta:

- Interface SPI
- Układ scalony UART (podobny do RS232, poziomy napięcia są zgodnie z napięciem zasilacza)
- Interface I2C

Poniższa grafika przedstawia schemat obwodu, który wykorzystuje do połączenia dodatkową antenę MFRC522.

### Połączenia

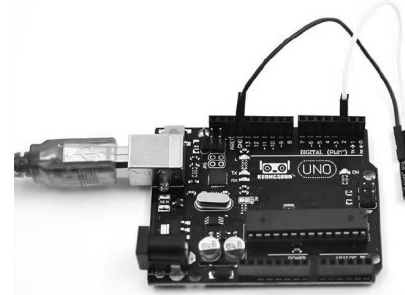
#### Schemat



### Program

Po połączeniu według schematu, uruchom program — Lesson 8 Ball Switch Tilt (Lekcja 8 czujnik przechyłu) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy.

### Przykład



## Lekcja 9. Serwo

### Opis

Serwo jest rodzajem motoru, który może obracać się o 180 stopni. Jest kontrolowany przez impulsy elektryczne wysyłane z płytki UNO R3. Impulsy te przekazują Serwo informację, w której pozycji ma się znaleźć. Serwo posiada 3 przewody: brązowy, przewód uziemiający, który powinien być podpięty do wejścia GND, czerwony, który jest przewodem zasilającym i powinien być podłączony do portu 5V oraz pomarańczowy, będący przewodem sygnałowym, który podłączamy do portu Dig#9.

### Wymagane komponenty

- (1)x kuongshun UNO R3
- (2) x Serwo (SG90)
- (3) x przewody M-M (Męsko-męskie przewody połączeniowe)

### Opis komponentów

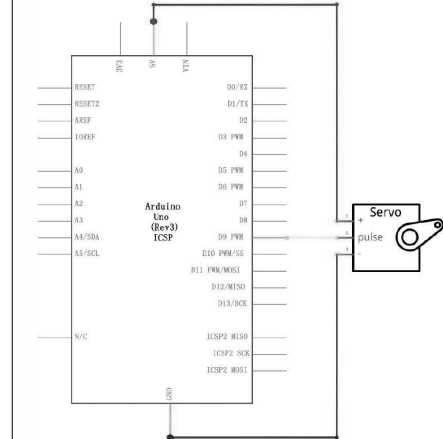
#### SG90

- kompatybilny ze złączami JR i FP
- długość przewodów: 25 cm
- bez obciążenia; szybkość: 0.12 s / kąt działania: 60 stopni (4.8V), 0.10s / 60 stopni (6.0V)
- moment (4.8V): 1.6 kg/cm
- temperatura pracy: -30–60°C
- szerokość martwej strefy: 5us
- napięcie zasilania: 3.5 ~6 V
- wymiary: 3,2 cm x 3 cm x 1.2 cm
- waga: 134g

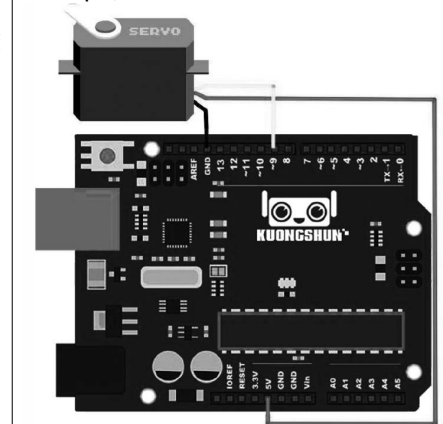


### Połączenie

#### Schemat



#### Schemat połączenia

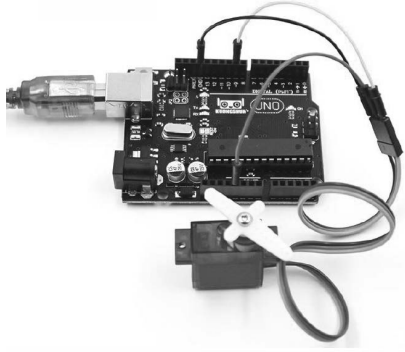


### Program

Po połączeniu według schematu, uruchom program — Lesson 9 Servo (Lekcja 9 Serwo) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <Servo> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ

bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.

### Przykład



Na załączonym obrazku, brązowy przewód jest podłączony przez czarny przewód M-M, czerwony podłączony przez czerwony przewód M-M a pomarańczowy jest podłączony przez żółty przewód M-M do UNO.

## Lekcja 10. DHT11. Czujnik temperatury i wilgotności

### Opis

W tym samouczku dowiemy się, jak korzystać z czujnika temperatury i wilgotności (DHT11). Jest on wystarczająco dokładny dla większości projektów, które wymagają śledzenia odczytów temperatury lub wilgotności. Podczas tej lekcji skorzystamy również ze specjalnej biblioteki przeznaczonej dla tego czujnika, aby nasz kod był krótki i łatwy do napisania.

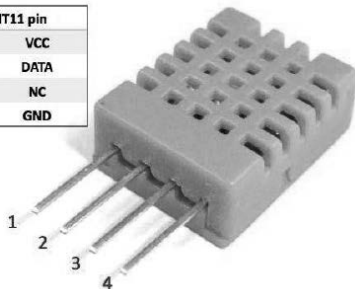
### Niezbędne komponenty

- (1) x płytko UNO R3
- (1) x czujnik temperatury i wilgotności DHT11
- (3) x żeńsko-męskie przewody połączeniowe

### Opis komponentów

#### Czujnik temperatury i wilgotności

DHT11 pin	
1	VCC
2	DATA
3	NC
4	GND



DHT11 to kompozytowy, cyfrowy czujnik temperatury i wilgotności, który posiada skalibrowane wyjście cyfrowego sygnału. Składa się z rezystancyjnego czujnika wilgotności, termistora (NTC) i łączy się z wydajnym 8-bitowym mikrokontrolerem.

Zastosowanie: COWiG (z ang. HVAC), osuszacz powietrza, sprzęt do testowania lub kontrolowania, produkty konsumenckie, pojazdy, automatyczne sterowanie, rejestratory danych, stacje pogodowe, urządzenia domowe, regulator wilgotności, sprzęty medyczne i inne urządzenia służące do pomiarów.

### Parametry produktu:

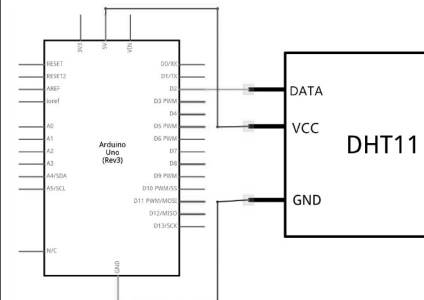
- Wilgotność względna
- Rozdzielczość: 16 bit
- Powtarzalność:  $\pm 1\%$  RH
- Dokładność: przy  $25^{\circ}\text{C} \pm 5\%$  RH
- Wymiennosc: w pełni wymienny
- Czas odpowiedzi:  $1/e$  (63%) z  $25^{\circ}\text{C}$ , 6s
- Powietrze -  $1\text{m}^3/\text{s}$ , 6s
- Histeresa:  $< \pm 0,3\%$  RH
- Stabilność długoterminowa:  $< \pm 0,5\%$  RH / rok
- Temperatura
- Rozdzielczość: 16 bit
- Powtarzalność:  $\pm 0,2^{\circ}\text{C}$
- Zakres: przy  $25^{\circ}\text{C} \pm 2^{\circ}\text{C}$
- Czas odpowiedzi:  $1/e$  (63%), 10s
- Parametry elektryczne
- Napięcie zasilania: DC 3,5–5,5V
- Pobór prądu: pomiar 0,3mA, czuwanie 60uA, czas pomiaru: dłużej niż 2 sekundy.

### Opis pinów

1. VDD zasilany napięciem w zakresie 3.3V do 5.5V.
2. DATA, dane
3. NC, brak funkcji, niepodłączony
4. GND, masa

### Połączenia

#### Schemat



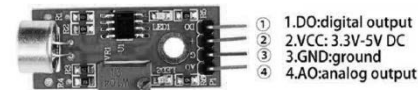
## Lekcja 14. Moduł czujnik dźwięku

### Opis

W tej lekcji dowiesz się, jak korzystać z czujnika dźwięku. Ten moduł ma dwa wyjścia:

AO: wyjście analogowe, sygnał napięcia wyjściowego mikrofonu w czasie rzeczywistym;

DO: kiedy natężenie dźwięku osiąga pewien próg, wyjście ma niski lub wysoki poziom sygnału. Wspomniany próg można regulować ustawieniami potencjometru. Aby mieć pewność, że mikrofon wykrywa głos, zmień ustawienia wrażliwości operując niebieskim potencjometrem umieszczonym na module. Zazwyczaj udaje się prawidłowo ustawić mikrofon po dziesiątym przekręceniu.



### Niezbędne komponenty

- (1) x płytko UNO R3
- (1) x czujnik dźwięku
- (4) x żeńsko-męskie przewody połączeniowe

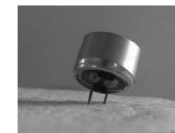
### Opis komponentów

#### Mikrofon

Przetwornik konwertuje jedną postać energii na inną. Mikrofon również jest przetwornikiem i zmienia energię fal dźwiękowych na sygnały elektryczne. Można je kupić w wielu kształtach i rozmiarach. W zależności od jego przeznaczenia, mikrofon wykorzystuje różne technologie do konwertowania dźwięku na sygnały elektryczne. Tym razem omówimy mikrofon pojemnościowy, który znajduje swoje zastosowanie w telefonach, laptopach, itp.

Jak sugeruje nazwa, mikrofon pojemnościowy jest zbudowany z równolegle połączonych kondensatorów, które opierają się na zasadzie zmiennej pojemności.

Składa się z dwóch płytek, jednej nieruchomej i drugiej ruchomej, zwanej membraną, z niewielkim odstępem między nimi. Potencjał elektryczny ładuje nieruchomą płytkę, kiedy fale dźwiękowe uderzają w membraną, która zaczyna drgać, zmieniając tym samym pojemność między płytkami, co prowadzi do przepływu zmiennego prądu elektrycznego.



Tego typu mikrofony są wykorzystywane w obwodach elektrycznych do wykrywania dźwięków lub nieznacznych drgań powietrza,

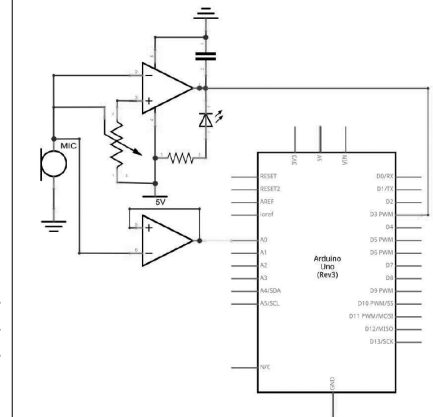
które następnie są konwertowane na sygnały elektryczne. Dwie metalowe nóżki widoczne na zdjęciu ustanawiają połączenie elektryczne z obwodem.



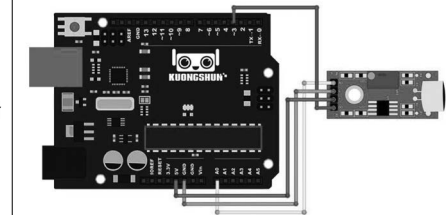
Solidny metalowy korpus mieści w sobie różne części mikrofonu. Wierzch zakryty jest porowatym materiałem, który przymocowano za pomocą kleju. Pełni rolę filtra przeciwpływowego. Fale dźwiękowe/wibracje powietrza przedostają się do wnętrza przez materiał i uderzają w membraną przez dziurę widoczną na zdjęciu.

### Połączenia

#### Schemat



#### Schemat połączenia



### Program

Po połączeniu według schematu, uruchom program – Lesson 14 Sound Sensor Module (Lekcja 14 – Moduł czujnik dźwięku) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrzywania, jeśli wystąpią błędy.

## Lekcja 13. Moduł zegar czasu rzeczywistego.

### Opis

W tej lekcji dowiesz się, jak korzystać z DS3231, czyli modułu, który wyświetla lata, miesiące, tygodnie, dni, godziny, minuty i sekundy. Wyposażony jest również w zapasową ładowarkę podtrzymującą akumulator, z której można skorzystać, kiedy moduł nie jest podłączony do płytki UNO R3.

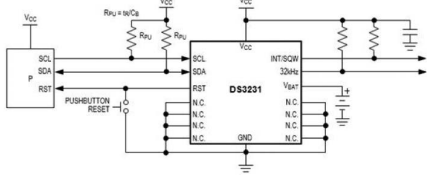


### Niezbędne komponenty:

- (1) x płytka UNO R3
- (1) x moduł DS3231 RTC
- (4) x żeńsko-męskie przewody połączeniowe

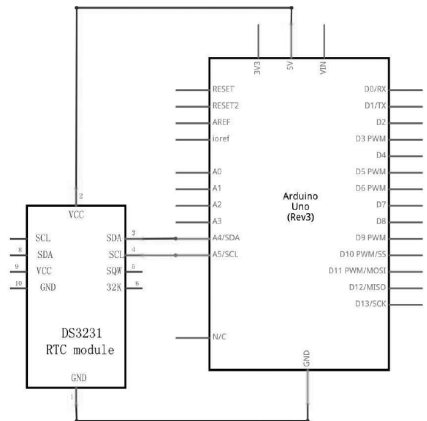
### Opis komponentów

Moduł DS3231 jest prostym układem, którego zadaniem jest odczytywanie czasu. Posiada zintegrowaną baterię, co pozwala mu pełnić swoją funkcję nawet gdy jest niepodłączony do płytki.



### Połączenia

#### Schemat



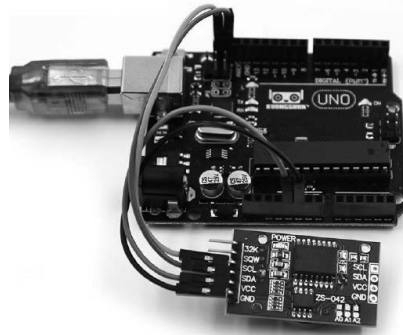
### Schemat połączenia

Pomiędzy piny 32K i SQW; tym razem nie będą potrzebne. Podłącz pin SCL do pola SCL na płytce UNO R3, to samo zrób z pinem SDA. Następnie pin VCC wepnij do pola 5V, a pin GND, do pola GND.

### Program

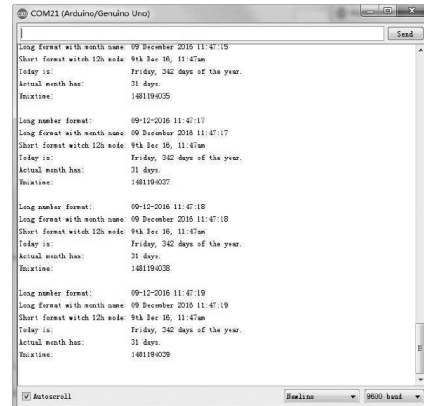
Po połączeniu według schematu, uruchom program – Lesson 13 Real Time Clock Module (Lekcja 13 - Moduł zegar czasu rzeczywistego) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <SimpleDHT> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.

### Przykład

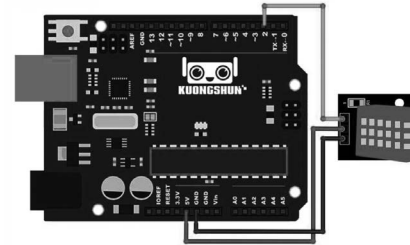


Uruchom monitor, aby zobaczyć poniższe dane:

Kliknij przycisk monitor szeregowy, aby go włączyć. Więcej szczegółów dotyczących monitora szeregowego możesz znaleźć w Lekcji 1.



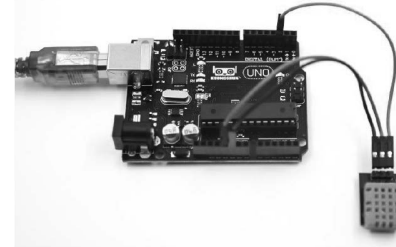
### Schemat połączenia



Jak widać na schemacie, potrzebujemy tylko trzech połączeń, ponieważ 4 pin nie pełni żadnej funkcji. Nasze połączenia to: Napięcie, Masa i Sygnał, które można podłączyć do każdego pinu na płytce UNO.

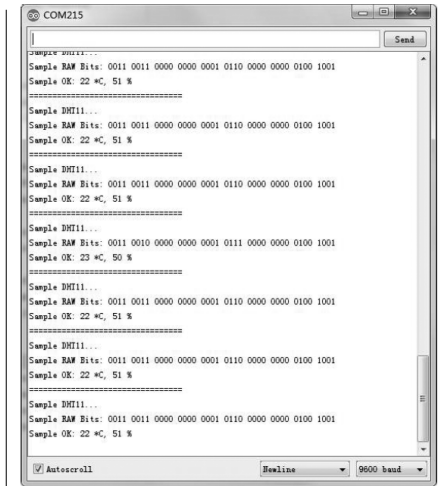
### Program

Po połączeniu według schematu, uruchom program – Lesson 10 DHT11 Temperature and Humidity Sensor (Lekcja 10 DHT11 - Czujnik temperatury i wilgotności) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy. Ponadto, upewnij się przed uruchomieniem programu, że biblioteka <SimpleDHT> została zainstalowana. W razie potrzeby możesz zainstalować ją ponownie, ponieważ bez niej program nie będzie działał. Więcej szczegółów dotyczących bibliotek znajduje się w Lekcji 1.



Wgraj program, a następnie otwórz monitor, aby zobaczyć dane pokazane obok (pokazano temperaturę otoczenia, która wynosi 22 stopnie).

Kliknij przycisk monitor szeregowy, aby go włączyć. Więcej szczegółów dotyczących monitora szeregowego znajduje się w Lekcji 1.



## Lekcja 11. Moduł joystick analogowy

### Opis

Analogowe joysticki to świetny sposób, aby wprowadzić trochę kontroli do projektów. W tej lekcji nauczymy się, jak korzystać z modułu analogowego joysticka.

### Niezbędne komponenty

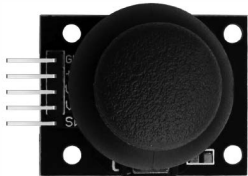
- (1) x płytka UNO R3
- (1) x moduł joysticka
- (5) x żeńsko-męskie przewody połączeniowe

### Opis komponentów

#### Joystick

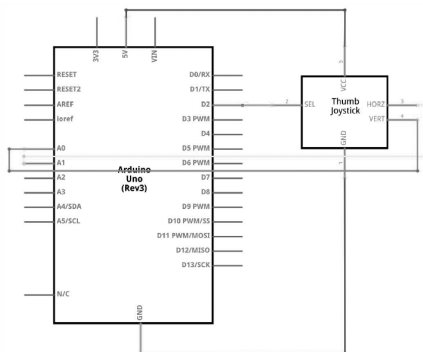
Moduł posiada pięć wyprowadzeń: zasilanie (VCC), Masa, odczyt potencjometru osi X i Y, cyfrowy sygnał przycisku. Pamiętaj, że te nazwy mogą się nieznacznie różnić między modułami. Przycisk joysticka jest analogowy, co czyni go bardziej precyzyjnym od joysticków kierunkowych z przyciskami lub mechanicznymi przełącznikami. Dodatkowo, wciskając przycisk joysticka w dół, można aktywować opcję „naciśnij, aby wybrać”.

Aby odczytać dane z pinów X/Y, należy użyć analogowych pinów Arduino, i osobnego pinu, aby podłączyć przycisk. Cyfrowy sygnał przycisku jest podłączony do masy, kiedy nacisniemy gałkę. Kiedy znajduje się w stanie „floating”, nie jest podłączony. Wtedy należy wspomóc się zasilaniem za pośrednictwem rezystora. W tym celu można użyć wbudowanych rezystorów na analogowych pinach Arduino.

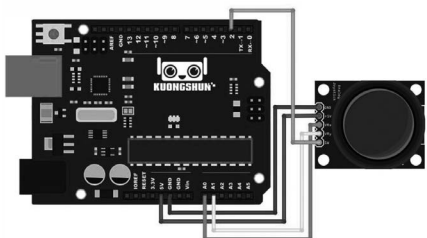


## Połączenia

### Schemat



### Schemat połączenia



Potrzebnych jest 5 połączeń z joystickiem; cyfrowy sygnał przycisku, potencjometr osi X i Y, napięcie oraz masa. Osie X i Y to połączenia analogowe, a cyfrowy sygnał przycisku, to połączenie cyfrowe. Jeśli nie ma potrzeby użycia galki, wystarczą 4 połączenia.

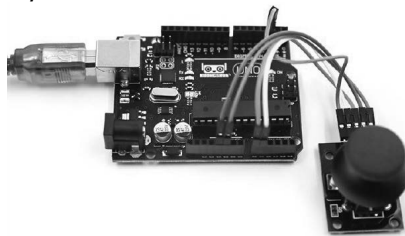
### Program

Po połączeniu według schematu, uruchom program – Lesson 11 Analog Joystick Module (Lekcja 11 Moduł joystick analogowy) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy.

Analogowe joysticki działają jak potencjometry, więc wysyłają wartości analogowe.

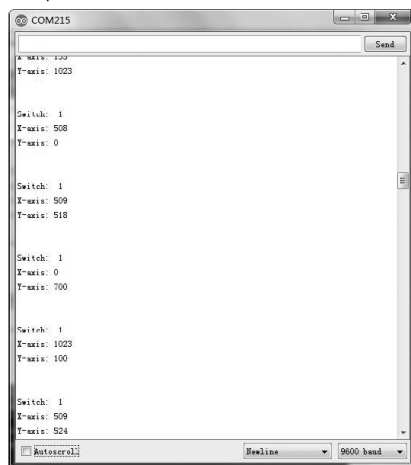
Jeśli joystick jest w stanie spoczynku, powinien wysłać wartość 512. Zakres wartości waha się od 0 do 1024.

## Przykład



Uruchom monitor, aby zobaczyć poniższe dane:

Kliknij przycisk monitor szeregowy, aby go włączyć. Więcej szczegółów dotyczących monitora szeregowego możesz znaleźć w Lekcji 1.



## Lekcja 12. Moduł czujnik wykrywania poziomu cieczy

### Opis

W tej lekcji wyjaśnimy, jak korzystać z modułu czujnika wykrywania poziomu wody. Czujnik ten mierzy głębokość wody, a jego główną częścią jest układ wzmacniający, który składa się z tranzystora i płytki drukowanej ze ścieżkami. Po włożeniu do wody, ścieżki płytki pozwalają wykryć rezystencję wody, która zmienia się w zależności od głębokości cieczy. Sygnał jaki wysłał czujnik na określonej głębokości jest następnie konwertowany na sygnał analogowy. Dzięki temu wykrywamy zmianę poziomu cieczy poprzez funkcję analogRead (ADC).

## Niezbędne komponenty

- (1) x płytki UNO R3
- (3) x żeńsko-męskie przewody połączeniowe
- (1) x czujnik poziomu cieczy

## Opis komponentów

### Czujnik cieczy

Moduł może znaleźć zastosowanie przy pomiarze opadów deszczu, poziomu wody, a nawet przy przeciekach. Czujnik scala się z następujących elementów: elektronicznego złącza, rezystora 1MΩ oraz kilku odsonionych przewodów i działa dzięki serii (odkrytych) ścieżek podłączonych do masy. Między uziemione ścieżki wpleciono ścieżki czujnika. Obwód będzie funkcjonował po podłączeniu do cyfrowych lub analogowych pinów wejścia i wyjścia na płycie UNO R3. Czujnik rozpoznaje poziom cieczy /kropli za pośrednictwem szeregu równoległych przewodów odsonionych, które zanurza się w danej cieczy, aby określić jej objętość. Ostatnim etapem jest konwersja poziomu cieczy na sygnał analogowy i wyjściowe wartości analogowe, które potem można wykorzystać przy pisaniu programu w celu stworzenia alarmu. Moduł jest energooszczędny i bardzo czuły.

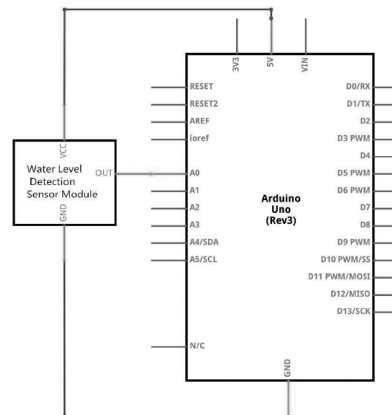
### Dane techniczne

- Napięcie zasilania: 5V
- Prąd roboczy: <20 mA
- Interfejs: analogowy
- Zakres wykrywalności: 40 mm x 16 mm
- Temperatura robocza: 10°C do 30°C
- Sygnał napięcia wyjściowego: od 0 do 4.2V

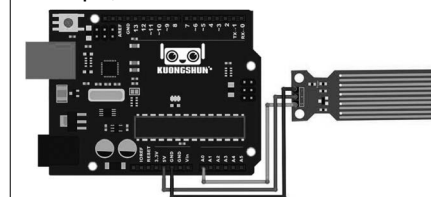


## Połączenia

### Schemat



## Schemat połączenia

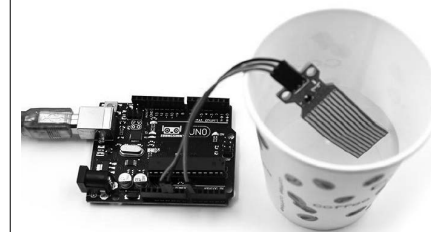


Wskazówki: Zasilanie (+) jest podłączone do płytki UNO R3, elektroda uziemiająca (-) podłączona jest do masy. Wyjście sygnału (S) podłączone jest do portów (A0-A5), które wprowadzają sygnał analogowy do płytki UNO R3.

### Program

Po połączeniu według schematu, uruchom program – Lesson 12 Water Level Detection Sensor Module (Lekcja 12 Moduł czujnik poziomu wykrywania cieczy) z folderu ze szkicami i kliknij wgraj. Wróć do lekcji 2, aby poznać więcej szczegółów dotyczących wgrывania, jeśli wystąpią błędy.

### Przykład



Uruchom monitor, aby zobaczyć poniższe dane:

Kliknij przycisk monitor szeregowy, aby go włączyć. Więcej szczegółów dotyczących monitora szeregowego możesz znaleźć w Lekcji 1.

