

**MoboLab – roboty i tablety w Twojej szkole**  
**Obszar II. „Stwórz własnego robota”**  
Scenariusze lekcji i zajęć pozalekcyjnych

**SCENARIUSZ 12. SILNIK NA PRĄD STAŁY**

*scenariusz zajęć pozalekcyjnych*

autor: Wojciech Karcz

redakcja: Agnieszka Koszowska

**SŁOWA KLUCZOWE:**

Arduino, programowanie, mikrokontroler, dioda LED, Arduino IDE

**KRÓTKI OPIS ZAJĘĆ:**

Jednym z przykładów urządzeń wyjściowych jest **silnik działający na prąd stały**. Takie silniki możemy również podłączyć do Arduino. Podczas zajęć uczniowie i uczennice uczą się, jak działa taki silnik oraz jak można zbudować prosty **wentylator**. Poznają i/lub utrwalają podstawowe pojęcia programistyczne (skrypt, program, algorytm, sterowanie, warunek, pętla). Budują układ elektroniczny i tworzą program dla silnika. Na zakończenie wykonują zadanie polegające na odwróceniu kierunku poruszania się wału silnika, tak aby kręcił się w przeciwną stronę.

**WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIĄ / UCZENNICĘ:**

- wie, czym są mikrokontrolery i do czego służą,
- zna pojęcia: mikrokontroler, skrypt, program, algorytm, sterowanie, warunek, pętla,
- zna projekt Arduino, wie, czym jest platforma Arduino, z jakich części się składa,
- potrafi w podstawowym stopniu samodzielnie obsługiwać Arduino (podłączyć płytkę do komputera, wgrać prosty program),
- wie, co to jest dioda LED,
- potrafi poprawnie podłączyć diodę LED do Arduino,
- zna podstawowe elementy interfejsu środowiska programistycznego Arduino IDE i podstawowe komendy języka Arduino IDE: **pinMode()**,

### **digitalWrite(), delay(),**

- rozumie zasadę działania funkcji **digitalWrite()** i potrafi wykorzystać ją w praktyce,
- zna podstawowe elementy języka **Scratch**, potrafi stworzyć prosty skrypt w tym języku,
- wie, co to jest silnik elektryczny na prąd stały i jak działa,
- wie, jak działają układy sterowane za pomocą tranzystorów,
- potrafi sterować silnikiem elektrycznym i regulować jego prędkość obrotową za pomocą Arduino.

### **GRUPA DOCELOWA:**

Starsze klasy szkoły podstawowej (VII-) i klasy gimnazjalne (po dostosowaniu: możliwość realizacji w młodszych klasach: I-III i IV-VI szkoły podstawowej). W młodszych klasach – możliwość wykorzystania programu mBlock (po przejściu scenariusza nr 18. *Programowanie Arduino z wykorzystaniem programu mBlock*) lub Scratch for Arduino (po przejściu scenariusza nr 1. *Wprowadzenie do Arduino*).

### **LICZBA UCZNIÓW/UCZENNIC W GRUPIE:**

Liczba optymalna: 12, liczba maksymalna: 16

### **CZAS TRWANIA ZAJĘĆ:**

90 min (lub 2 x 45 minut)

### **STOPIEŃ TRUDNOŚCI/SKOMPLIKOWANIA**

**(w skali od 1 do 5 dla obszaru II. „Stwórz własnego robota”):**

4

### **POTRZEBNY SPRZĘT I OPROGRAMOWANIE:**

- komputer (przenośny lub stacjonarny),
- program Arduino IDE (do pobrania ze strony: <http://www.arduino.org/downloads>),
- (opcjonalnie) program mBlock (do pobrania ze strony: <http://www.mblock.cc/download/>) lub Scratch for Arduino (do pobrania ze strony: <http://s4a.cat/>),
- płytki Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytki stykowe,
- przewody połączeniowe,

- silnik DC, tranzystor NPN,
- dioda 1N4001,
- potencjometr 10K omów,
- oporniki 270 omów,
- projektor i laptop (w części teoretycznej).

### CO NALEŻY PRZYGOTOWAĆ PRZED ZAJĘCIAMI:

- zainstalować program Arduino IDE,
- (opcjonalnie): zainstalować program **mBlock** lub **Scratch for Arduino**,
- sprawdzić, czy wszystkie komputery wykrywają podłączone Arduino,
- przeczytać dokładnie scenariusz,
- zapoznać się z materiałami dodatkowymi (w części „Pigułka wiedzy i inspiracji”),
- wykonać samodzielnie zadania zawarte w scenariuszu,
- przy każdym stanowisku komputerowym rozłożyć elementy zestawu Arduino, które będą wykorzystywane na tych zajęciach,
- dopasować stopień trudności zadania do potrzeb i możliwości klasy, dla której organizowane są zajęcia według wskazówek zawartych w scenariuszu.

### KOMPETENCJE OSOBY PROWADZĄCEJ:

- wie, czym jest projekt Arduino, zna podstawowe informacje o projekcie,
- potrafi przynajmniej w stopniu podstawowym obsługiwać Arduino,
- zna podstawowe pojęcia z zakresu elektroniki,
- zna podstawowe pojęcia programistyczne,
- wie, dlaczego warto uczyć się programowania i jakie korzyści daje posiadanie umiejętności programistycznych,
- potrafi zachęcić do nauki programowania zarówno chłopców, jak i dziewczynki.

### PRZEBIEG ZAJĘĆ:

#### **Podłączenie Arduino, uruchomienie programu Arduino IDE i przypomnienie podstawowych informacji – ok. 15 minut**

**Uwaga!** Informacje o tym, jak podłączyć Arduino, uruchomić program Arduino IDE i Scratch for Arduino, a także podstawowe informacje niezbędne przy rozpoczynaniu pracy z Arduino zawierają scenariusze 1 i 2. Tę część zajęć warto powtarzać za

każdym razem w takim zakresie, jaki jest potrzebny, do czasu aż podstawowy materiał zostanie utrwalony.

### **Jak działa silnik na prąd stały? – 15 minut**

Celem zajęć jest pokazanie uczniom, jak działa silnik na prąd stały oraz w jaki sposób nim sterować za pomocą Arduino.

Silniki na prąd stały są stosowane w wielu urządzeniach, z których korzystamy na co dzień (np. wentylator w komputerze, odtwarzacz CD, elektryczna szczoteczka do zębów, silnik wibracyjny w telefonie, itp.). Silniki prądu stałego są stosowane od bardzo dawna. Zasada ich działania opiera się na ruchu elektromagnesu pomiędzy dwoma różnoimiennymi magnesami. Za pomocą komutatora zmienia się cyklicznie biegunowość elektromagnesu w taki sposób, żeby była taka sama jak sąsiadujących z nim magnesów stałych. W ten sposób powstaje moment obrotowy, który wprawia w ruch wał silnika.

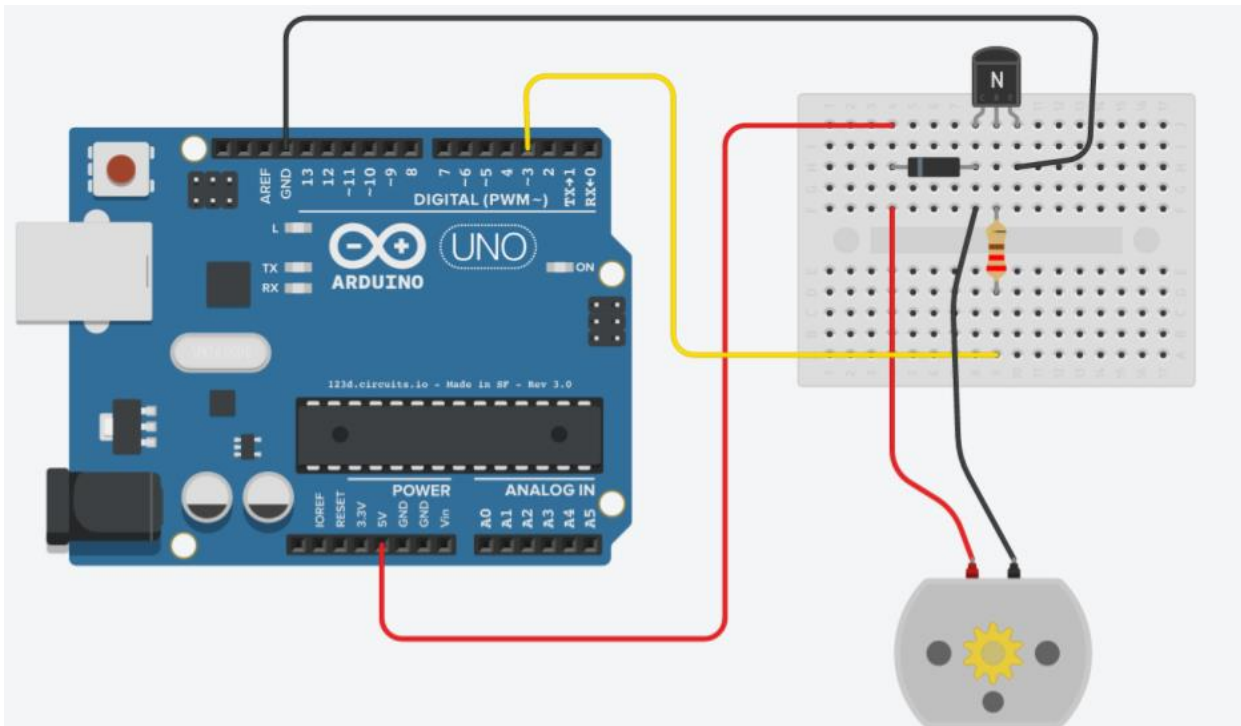
Zajęcia rozpoczynamy od dyskusji z uczniami na temat silników, zadajemy np. takie pytania, jak:

- *Jakie znacie rodzaje silników?*
- *Czym różni się silnik elektryczny od spalinowego?*
- *Jak zasilić silnik elektryczny?*
- *W jaki sposób możemy (i czy w ogóle możemy) sterować prędkością obrotową takiego silnika?*

Podsumowujemy odpowiedzi i opowiadamy uczniom, jak działa silnik elektryczny na prąd stały. Zapowiadamy, że podczas zajęć będą się uczyć sterować silnikiem za pomocą Arduino. Jest to kolejny rodzaj aktuatora, które może dodać elementy ruchu do naszego projektu.

### **Montaż układu i sterowanie silnikiem – 15 minut**

Wspólnie z uczniami montujemy układ widoczny poniżej:



Uwaga! Zdecydowana większość silników nie nadaje się do zasilania bezpośrednio z Arduino, ponieważ wymagają do rozruchu stosunkowo dużych prądów. Dlatego też w tym projekcie wykorzystujemy bardzo słaby silniczek, który bez problemu możemy zasilać ze złącza 5V Arduino. Jednak przy stosowaniu innych silników warto dokładnie zapoznać się z dokumentacją i sprawdzić, jaką moc ma konkretny silnik.

Mocniejsze silniki zasilamy z osobnego zasilacza prądu stałego o odpowiedniej mocy lub z baterii. Jedyna różnica w montażu układu polega na tym, że zamiast pinów 5V i GND, podłączamy silnik do plusa i minusa zewnętrznego źródła zasilania. W zależności od mocy silnika trzeba też zwrócić uwagę na dobór odpowiednio mocnego tranzystora. Przy mocniejszych silnikach warto zastosować układ MOSFET lub tzw. mostek H, co zabezpieczy Arduino przed przepaleniem.

**W tym miejscu możliwy jest podział zajęć na dwie części (kolejna część scenariusza będzie realizowana na następnych zajęciach).**

### **Przypomnienie materiału, odtworzenie układu – 10 minut**

Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i odtworzenia układu zbudowanego na poprzednich zajęciach.

## Programowanie układu – 20 minut

Następnie piszemy z uczniami podstawowy program na Arduino sterujący silnikiem. Przykładowy kod znajduje się poniżej:

```
int motorPin = 3;

void setup()
{
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
  while (!Serial);
  Serial.println("Speed 0 to 255");
}

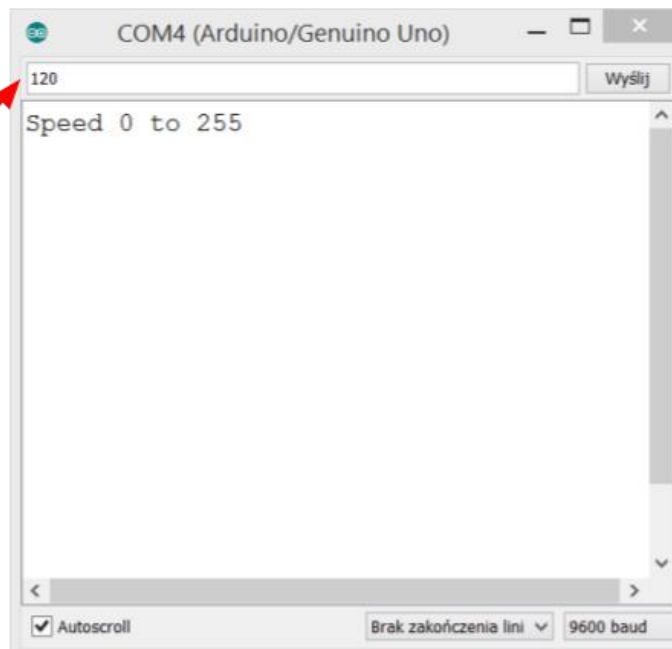
void loop()
{
  if (Serial.available())
  {
    int speed = Serial.parseInt();
    if (speed >= 0 && speed <= 255)
    {
      analogWrite(motorPin, speed);
    }
  }
}
```

Jak widać, kod nie jest zbyt skomplikowany. Wgrywamy kod na Arduino. Aby wprawić silnik w ruch, musimy uruchomić Szeregowy Monitor (Narzędzia > Szeregowy Monitor). W okienku na samej górze wpisujemy wartości w zakresie 0-255 i naciskamy ENTER. Wspólnie z uczniami obserwujemy, co się dzieje i jak zachowuje się silnik.

Zadajemy takie pytania, jak:

- *Czy widzicie zależność pomiędzy wpisywanymi liczbami?*
- *Jak wpływają one na zachowanie silnika?*
- *Kiedy silnik obraca się najszybciej, a kiedy najwolniej?*

**Tutaj wpisujemy  
wartości z zakresu  
0-255**



Dajemy uczniom chwilę czasu, aby samodzielnie poeksperymentowali z silnikiem.

Dodatkowo poza montażem układu, można z kartonu lub papieru wyciąć proste skrzydło i umieścić na wale silnika. W ten sposób znacznie łatwiej będzie obserwować zmianę obrotów.

Następnie omawiamy kod programu i działanie całego układu. Bardzo ważnym elementem jest tranzystor, który pełni rolę przełącznika, włączającego i wyłączającego prąd w obwodzie silnika. Zwracamy uwagę uczniów, że silnik jest tak naprawdę podpięty do osobnego obwodu elektrycznego (może być również zasilany z baterii; w ramach dodatkowej modyfikacji można dla przykładu zasilić silnik z baterii np. 9V).

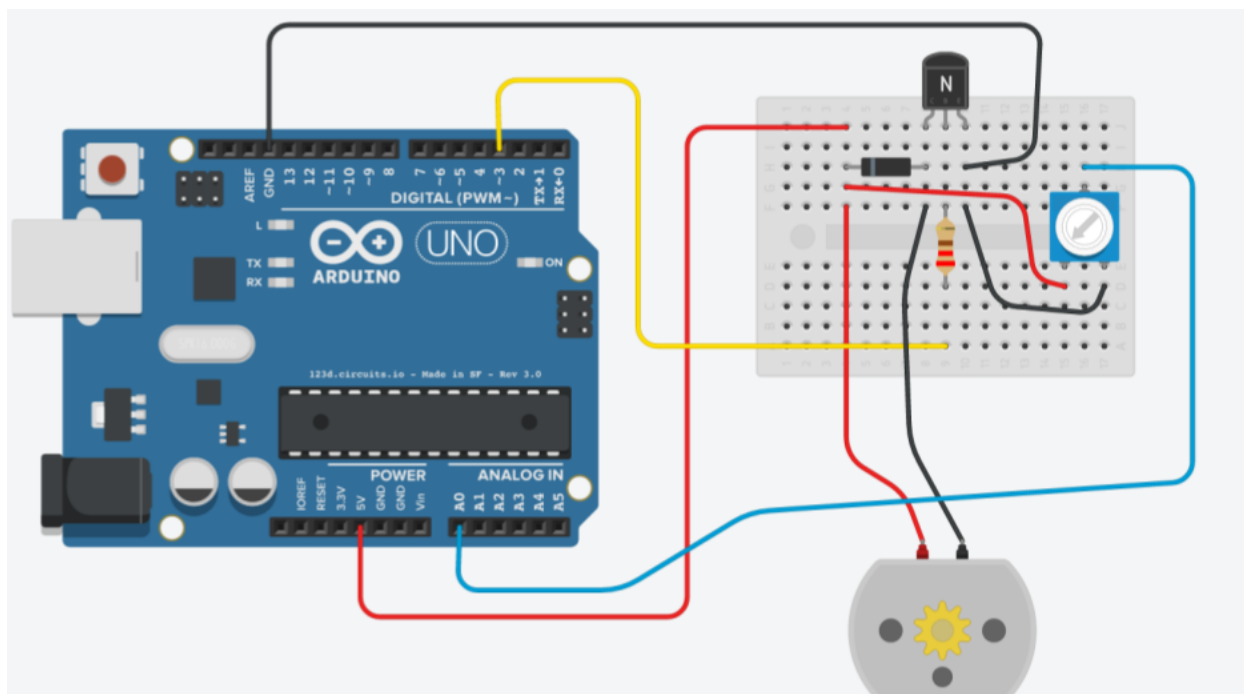
Dlatego rozdzielamy te dwa obwody (obwód silnika oraz obwód sterujący), żeby nie przepalić Arduino. Podczas pracy silnik pobiera bardzo duży prąd, który może zniszczyć Arduino. Podobna idea stoi za stosowaniem przekaźników (jak zostało to opisane w scenariuszu nr 11 o przekaźniku). Jednak zasada działania przełącznika jest trochę inna. Tranzystor jest elementem półprzewodnikowym, a przekaźnik w sposób mechaniczny łączy i rozdziela obwód.

Program, który steruje silnikiem, działa w następujący sposób:

- definiujemy pin 3, jako pin wyjściowy i uruchamiamy komunikację przez port szeregowy;
- w głównej pętli przesyłamy dane do Arduino z komputera za pomocą funkcji `Serial.parseInt()`;
- następnie sprawdzamy, czy wartość znajduje się we właściwym przedziale 0-255. Jeśli tak to przesyłamy tę wartość dalej poprzez pin sterujący do tranzystora za pomocą funkcji `analogWrite()`. Dlatego też pin sterujący w tym wypadku powinien być pinem PWM.

### **Sterowanie obrotami silnika za pomocą potencjometru – 15 minut**

Jeśli starczy czasu, to możemy z uczniami zmodyfikować układ z silnikiem i podłączyć potencjometr, który będzie sterował obrotami silnika. Montujemy poniższy układ:



Tak naprawdę nie musimy demontować w całości poprzedniego układu. Wystarczy tylko dodać obwód z potencjometrem.

Następnie piszemy i wgrywamy na Arduino poniższy kod:



```

int potPin = A0;
int motorPin = 3;
int potValue = 0;
int motorValue = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  potValue = analogRead(potPin);
  motorValue = map(potValue, 0, 1023, 0, 255);
  analogWrite(motorPin, motorValue);
  Serial.print("potentiometer = " );
  Serial.print(potValue);
  Serial.print("\t motor = ");
  Serial.println(motorValue);
  delay(2);
}

```

Sprawdzamy wspólnie z uczniami, jak działa silnik pod wpływem obrotów potencjometru. Pytamy uczniów, czy widzą podobieństwa z poprzednim programie, a może są jakieś różnice?

Tak naprawdę program działa bardzo podobnie. Zamiast ręcznego wpisywania wartości dla pinu sterującego przez port szeregowy, odczytujemy wartości analogowe z potencjometru. Następnie za pomocą funkcji `map()` przekształcamy wartości analogowe z zakresu 0-1023 otrzymywane od potencjometru na wartości przesyłane przez pin sterujący z zakresu 0-255 (takie wartości możemy przesyłać poprzez funkcję `analogWrite()`). Wartość 0 odczytaną z wejścia analogowego konwertujemy na 0 wysyłane przez złącze PWM. Tak samo odpowiednio 1023 będzie przekształcone na 255.

Dokładne wartości odczytywane z potencjometru i przesyłane przez pin sterujący. Możemy śledzić na bieżąco poprzez Szeregowy Monitor (Narzędzia > Szeregowy Monitor).

## MOŻLIWE MODYFIKACJE DLA MŁODSZYCH KLAS:

Pracując z uczniami w młodszych klasach można wykorzystać zamiast Arduino IDE program S4A (Arduino for Scratch). W przypadku zajęć z młodszymi dziećmi warto zwrócić uwagę na ewentualne problemy z dokładnym podłączeniem przewodów.

## ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole buduje prosty wentylator z wykorzystaniem Arduino i silnika DC. Zadanie polega na odwróceniu kierunku poruszania się wału silnika. Prosimy uczniów, żeby sprawili, żeby wał silnika kręcił się w przeciwną stronę. Można na wał nabić np. proste śmigło z kartonu, czy z papieru, żeby łatwiej było stwierdzić różnicę.

## PIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Kurs programowania Arduino Forbot:

<http://forbot.pl/blog/artykuly/programowanie/kurs-arduino-w-robotyce-1-wstep-id936>

Podstawowe informacje na temat prądu elektrycznego:

<http://forbot.pl/blog/artykuly/podstawy/podstawy-elektroniki-1-napiecie-prad-opor-zasilanie-id3947>

Jak działa płytka stykowa (prototypowa):

[https://pl.wikipedia.org/wiki/P%C5%82ytka\\_prototypowa](https://pl.wikipedia.org/wiki/P%C5%82ytka_prototypowa)

Silnik na prąd stały:

[https://pl.wikipedia.org/wiki/Silnik\\_pr%C4%85du\\_sta%C5%82ego](https://pl.wikipedia.org/wiki/Silnik_pr%C4%85du_sta%C5%82ego)

<https://www.youtube.com/watch?v=LAtPHANefQo>

*Scenariusz został opracowany na potrzeby projektu „MoboLab – roboty i tablety w Twojej szkole”. Celem projektu jest zwiększenie kompetencji informatycznych z zakresu programowania i wykorzystywania technologii mobilnych w uczeniu się, a także kreatywności, innowacyjności i umiejętności współpracy w zespole z wykorzystaniem TIK, uczniów / uczennic z (UCZ) z 6 szkół podnadgimnazjalnych i 4 gimnazjów Wołomina i Zielonki. Projekt dofinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego (Regionalny Program Operacyjny Województwa Mazowieckiego na lata 2014-2020, Oś Priorytetowa X. Edukacja dla rozwoju regionu, Działanie 10.1. Edukacja ogólna i przedszkolna, Poddziałanie 10.1.2. Edukacja ogólna w ramach ZIT).*



Ten utwór jest dostępny na licencji [Creative Commons Uznanie autorstwa 4.0 Międzynarodowe](https://creativecommons.org/licenses/by/4.0/).