



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

SCENARIUSZ 10. SERWOMECHANIZM

scenariusz zajęć pozalekcyjnych

autor: Wojciech Karcz

redakcja: Andrzej Zawieracz

SŁOWA KLUCZOWE:

Arduino, serwomechanizm, PWM, silnik na prąd stały

KRÓTKI OPIS ZAJĘĆ:

Podczas warsztatów uczestnicy dowiedzą się jak działa **serwomechanizm** i w jaki sposób możemy go połączyć z Arduino. Jest to niezwykle przydatna część pozwalająca bardzo precyzyjnie poruszać jakimś elementem (np. ramieniem robota).

WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIA / 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,
- wie, czym jest serwomechanizm i na jakiej zasadzie działa,
- potrafi sterować serwomechanizmem i wykorzystywać go w różnych projektach.
- 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.

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”):

2

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 stykowa, przewody połączeniowe,
- oporniki 220 omów,
- serwomechanizm (micro servo),
- potencjometr 10K 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.

Poznajemy serwomechanizm – 10 minut

Celem zajęć jest zapoznanie się uczestników z kolejnym urządzeniem elektronicznym – serwomechanizmem. Jest to niewielki silniczek, którym możemy sterować w bardzo precyzyjny sposób. Serwomechanizmy są stosowane głównie w modelarstwie (ze względu na niewielkie rozmiary i małą masę) i robotyce. Mogą się obracać głównie w zakresie od 0 do 180 stopni i tym samym nie wykonują pełnego obrotu. Dzięki serwomechanizmom możemy poruszać różnymi elementami bardzo precyzyjnie.

Więcej informacji na temat serwomechanizmów znajduje się w materiałach dodatkowych.

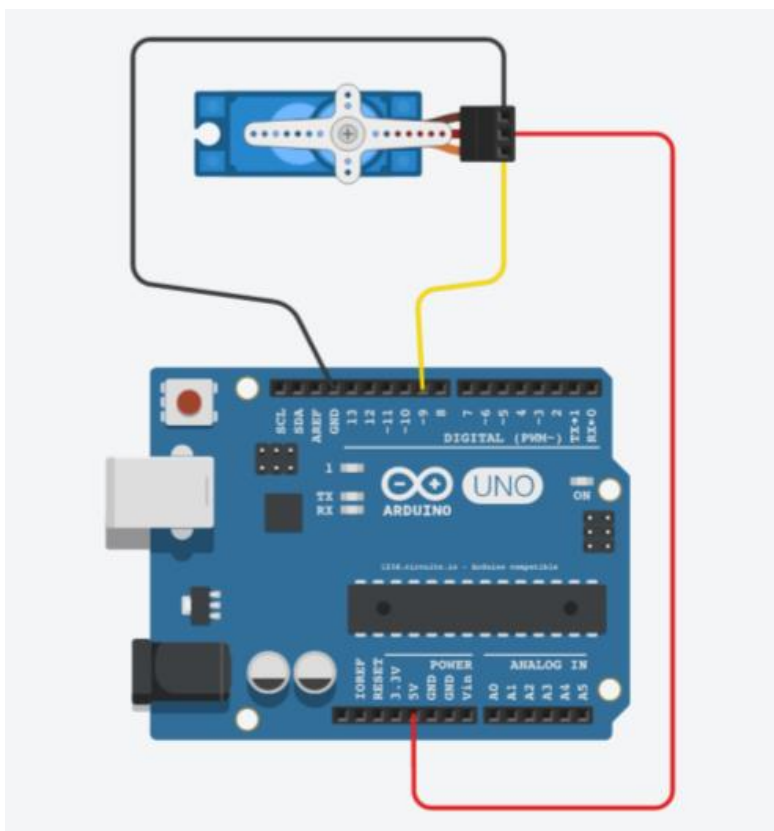
Zajęcia rozpoczynamy od dyskusji z uczniami na temat silników oraz robotów, zadajemy takie pytania, jak:

- *Jak działa silnik elektryczny (np. w wentylatorze)?*
- *Czy możemy sterować dokładną liczbą obrotów wentylatora?*
- *Gdzie przydaje się precyzyjny silnik, jakie może mieć zastosowania?* (mogą to być np. roboty medyczne lub przemysłowe)

Zbieramy i podsumowujemy odpowiedzi. Tłumaczymy uczniom, jak działają serwomechanizmy oraz czym różnią się od prostego silnika na prąd stały.

Montaż układu i programowanie serwomechanizmu – 20 minut

Budujemy układ – taki jak ten przedstawiony na poniższym schemacie:



Serwomechanizmy posiadają zazwyczaj trzy kabelki:

- **Czerwony** – zasilanie (5V)
- **Żółty/pomarańczowy** – sterowanie
- **Czarny/brązowy** – masa (GND)

Serwomechanizmy są zasilane napięciem 5V i muszą być podłączone do masy (GND). Dodatkowo podłączamy do nich przewód sterujący. Sterujemy nimi za pomocą sygnału PWM (więcej informacji w scenariuszu nr 3 o syrenie alarmowej). Dla minimalnej wartości sygnału PWM równej 0 – serwo przyjmuje położenie 0 stopni. Dla maksymalnego sygnału 255 przyjmuje położenie 180 stopni.

Uwaga! Na tych zajęciach stosujemy tzw. micro servo, które pobiera niewielki prąd, dzięki czemu może być zasilane bezpośrednio z Arduino poprzez kabel USB. Nie powinno się podłączać mocniejszych serwomechanizmów i silników na prąd stały bezpośrednio do Arduino, gdyż grozi to uszkodzeniem Arduino.

Po podłączeniu serwomechanizmu do Arduino otwieramy program o nazwie Sweep, który znajduje się w przykładach (Plik > Przykłady > Servo > Sweep). Program powinien wyglądać następująco:

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);                // tell servo to go to position in variable 'pos'
    delay(15);                          // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);                // tell servo to go to position in variable 'pos'
    delay(15);                          // waits 15ms for the servo to reach the position
  }
}
```

Jest to program typu „hello world” dla serwomechanizmu. Wgrywamy program na Arduino i obserwujemy zachowanie serwomechanizmu. Pytamy uczniów, w jaki sposób zachowuje się silniczek i czy widzą jakiś wzór, według którego się porusza. Dla lepszego efektu najlepiej od razu zamocować na wale serwomechanizmu jeden z plastikowych orczyków.

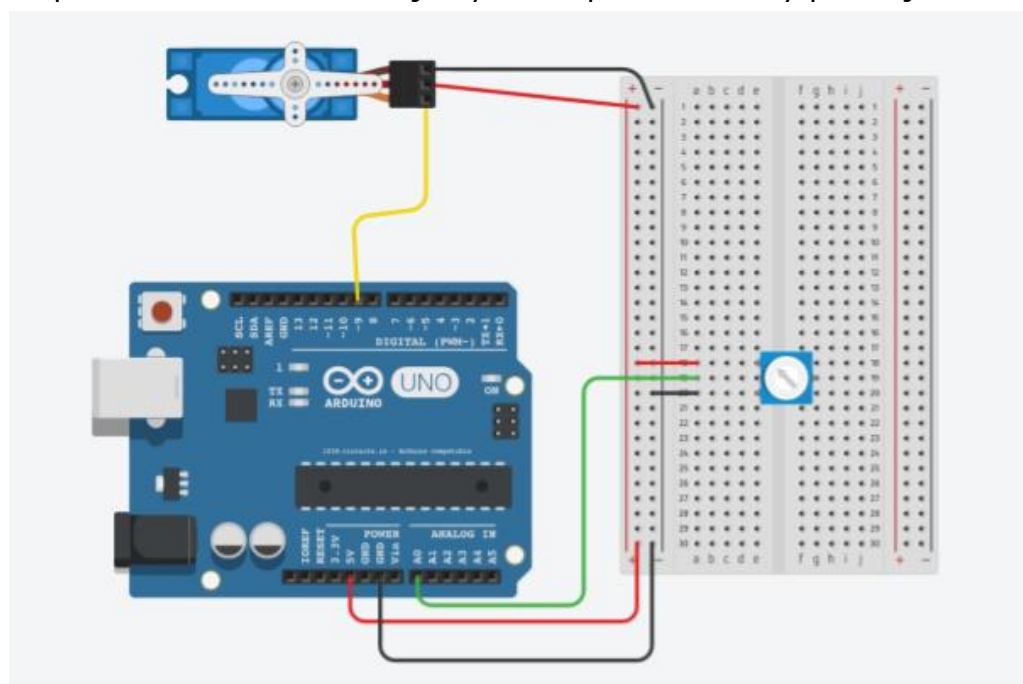
W tym miejscu możliwy jest podział zajęć na dwie części (kolejna część scenariusza będzie realizowana na następnych zajęciach).

Przypomnienie materiału, odtworzenie układu i analiza programu – 20 minut

Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i odtworzenia układu zbudowanego na poprzednich zajęciach. Następnie analizujemy z uczniami kod programu i tłumaczymy, w jaki sposób program wpływa na ruch serwomechanizmu. Możemy wraz z uczniami modyfikować kod programu i np. skrócić zakres obrotu serwa do 90 stopni (trzeba zmienić wartości ze 180 na 90 w obydwu pętlach „for”).

Precyzyjne sterowanie serwomechanizmem – 25 minut

Po przedstawieniu uczniom podstawowego sposobu działania serwomechanizmu można przejść do precyzyjnego sterowania serwem za pomocą potencjometru. Wspólnie z uczniami montujemy układ przedstawiony poniżej:



W porównaniu z poprzednim przykładem tutaj dokładamy jedynie potencjometr (nie ma potrzeby demontować poprzedniego układu, wystarczy tylko dołożyć potencjometr).

Następnie otwieramy program **Knob**, który znajduje się w przykładach (Plik > Przykłady > Servo > Knob). Wgrywamy program na Arduino. Zachęcamy uczniów, żeby pokręcili potencjometrem i sprawdzili jak zachowuje się serwomechanizm. Powinien on wychylać ramię wraz ze wzrastającą wartością na potencjometrze.

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); // waits for the servo to get there
}
```

Wspólnie z uczniami analizujemy kod programu. Staramy się zachęcić ich do samodzielnego zrozumienia kodu. Nową funkcją jest **myservo.attach(9)**, która oznacza przypisanie sygnału sterującego serwo do pinu nr 9 na Arduino. Podobnie przypisujemy pin wyjściowy, kiedy chcemy sterować diodą LED.

Funkcja **analogRead()** odczytuje wartości z potencjometru (podobnie jak w fotorezystorze).

Funkcja **map()** przekształca wartości analogowe z zakresu 0-1023 na 0-180, czyli wartości kąta, o który może obrócić się serwo.

Funkcja **myservo.write()** przypisuje konkretne wartości kąta (wyrażonego w stopniach), o który ma się obrócić serwo.

W tym momencie możemy dać uczniom chwilę na samodzielne eksperymentowanie z kodem i obserwowanie zachowania serwomechanizmu.

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 podłączaniem przewodów.

ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole buduje układ z wykorzystaniem Arduino, płytki stykowej, przewodów połączeniowych, micro servo i potencjometru. Zadanie końcowe polega na napisaniu prostego programu, który będzie obracać serwo z położenia zerowego (0 stopni) o kąt 45 stopni, czekać 3 s, wracać z powrotem do położenia zerowego i czekać 3 s.

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

Co to jest serwomechanizm:

<http://forbot.pl/blog/artykuly/programowanie/kurs-arduino-cz-5-silnikipwm-zewnetrzne-biblioteki-id3913>

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/).