

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

**SCENARIUSZ 9. CZUJNIK IMPEDANCYJNY**

*scenariusz zajęć pozalekcyjnych*

autor: Wojciech Karcz

redakcja: Andrzej Zawieracz

**SŁOWA KLUCZOWE:**

Arduino, wykrywanie ruchu, czujnik impedancyjny, biblioteki

**KRÓTKI OPIS ZAJĘĆ:**

Podczas zajęć uczniowie i uczennice poznają zasadę działania **czujnika impedancyjnego**. Jest to bardzo prosty czujnik składający się wyłącznie z pojedynczej anteny, który umożliwia wykrywanie poruszających się obiektów posiadających ładunek elektryczny (takich jak np. człowiek, zwierzęta, naelektryzowane przedmioty itp.). W trakcie zajęć uczniowie testują działanie takiego czujnika i sprawdzają, które przedmioty są naładowane elektrycznie, a które nie.

**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.

### **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”):**

3

### **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/>),
- płytko Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytko stykowa,
- oporniki 220 omów,
- przewody połączeniowe,
- buzzer,
- diody LED w różnych kolorach,
- 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 czujnik impedancyjny – 10 minut**

Celem zajęć jest zapoznanie uczniów z bardzo ciekawym rodzajem czujnika, który posiada zdolność wykrywania ładunków elektrycznych. Jest to tzw. czujnik impedancyjny. Choć sądząc po nazwie urządzenie może wydawać się skomplikowane, w rzeczywistości czujnik ten wymaga podłączenia do Arduino wyłącznie jednego

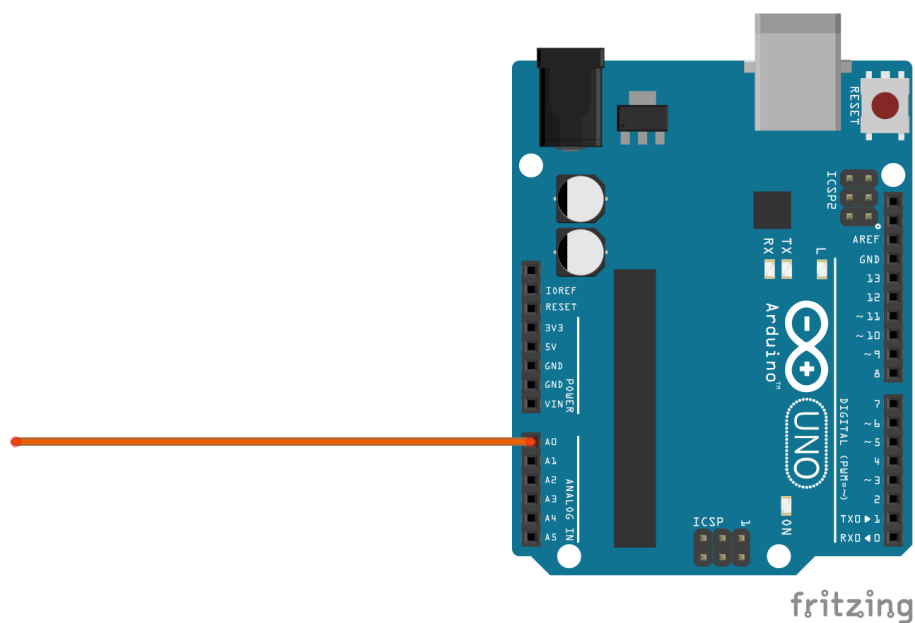
kabelka oraz wgrania odpowiedniej biblioteki.

Zajęcia rozpoczynamy od dyskusji z uczniami, zadajemy takie pytania, jak:

- *Jakie znacie czujniki ruchu?*
- *Jak można wykryć ruch człowieka oraz innych żywych istot?*

Zbieramy i systematyzujemy wszystkie odpowiedzi. Jeśli wcześniej był realizowany scenariusz z czujnikiem PIR, to można nawiązać do tych zajęć.

Na zajęciach będziemy budować czujnik ruchu wykorzystujący niezwykle ciekawy mechanizm. Cały czujnik składa się wyłącznie z anteny, która jest zwykłym kabelkiem wtykowym:



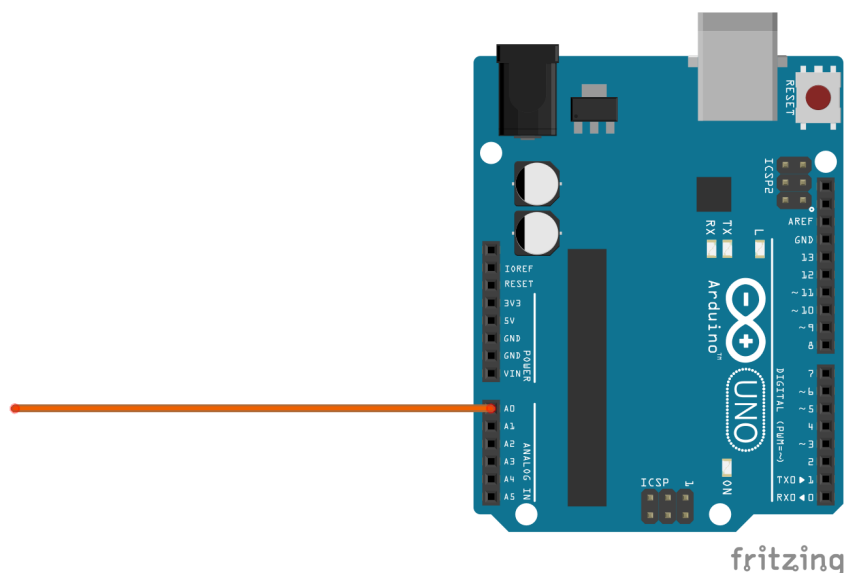
Każdy z nas praktycznie zawsze jest naelektryzowany i posiada na sobie pewien ładunek elektryczny. Czasem ten ładunek jest na tyle wysoki, że gdy dotkniemy jakiegoś uziemionego przedmiotu, to zobaczymy małą iskrę (i poczujemy niewielkie „kopnięcie” prądu). Niemal każdy z nas doświadczył tego przynajmniej raz w życiu. Można też zapytać uczniów o podobne doświadczenia z elektryzowaniem się różnych przedmiotów.

W ramach eksperymentu można nadmuchać gumowy balon, a następnie pocierać nim o sweter. Przykładając tak naelektryzowany balon w pobliżu włosów zauważymy, że będą się one kierować w stronę balona.

Na podobnej zasadzie działa czujnik impedancyjny. Każdy z nas ma jakiś ładunek, którego ruch powoduje wytworzenie niewielkiego prądu w antenie dołączonej do pinu analogowego Arduino. Za pomocą specjalnej biblioteki Buzz możemy przetworzyć ten sygnał i wykorzystać go do wykrywania ruchu.

## Montaż układu i programowanie – 20 minut

Wspólnie z uczniami montujemy układ przedstawiony na poniższym schemacie:



W przeciwieństwie do wielu popularnych czujników analogowych, gdzie odcytujemy wartość sygnału za pomocą wejść analogowych i przeliczamy ją na konkretną wielkość fizyczną, w tym przypadku musimy wykorzystać bibliotekę.

## Biblioteki w Arduino

Oprócz wbudowanych podstawowych funkcji możemy również zastosować w Arduino tzw. biblioteki. Biblioteki zawierają fragmenty kodu, które są odpowiedzialne za komunikację z różnymi urządzeniami czy sensorami. Dzięki temu nie musimy przepisywać konkretnego kodu, lecz wystarczy zastosować wygodne funkcje, jak np. **buzz.begin()**. Ta funkcja uruchamia czujnik impedancyjny.

Do bibliotek musimy odwołać się na początku naszego programu stosując komendę **#include**. Biblioteki możemy instalować za pomocą **Menedżera bibliotek** (Szkic > Dołącz bibliotekę > Zarządzaj bibliotekami). Drugim sposobem jest instalacja

biblioteki bezpośrednio z pliku .zip. Po ściągnięciu biblioteki na dysk instalujemy ją za pomocą funkcji **Dodaj bibliotekę .ZIP** (Szkic > Dołącz bibliotekę > Dodaj bibliotekę .ZIP).

Wszystkie poprawnie zainstalowane biblioteki znajdują się w głównym folderze Arduino (przeważnie w Moich Dokumentach) w folderze **libraries**. Więcej informacji o instalowaniu bibliotek w Arduino znajduje się w materiałach dodatkowych.

Zanim zaczniemy programować Arduino, musimy ściągnąć odpowiednią bibliotekę, która będzie obsługiwała nasz czujnik. Biblioteka nosi nazwę **Buzz** i znajdziemy ją pod następującym adresem:

<https://github.com/connornishijima/arduino-buzz/archive/master.zip>

Po ściągnięciu biblioteki uczniowie instalują ją na swoich komputerach. Informacje z dokładnym opisem, jak zainstalować bibliotekę z pliku .ZIP znajdują się w materiałach dodatkowych. Następnie piszemy kod przedstawiony na poniższym rysunku:

importujemy  
bibliotekę Buzz

uruchamiamy antenę  
przy pinie A0

wysyłamy dane do  
Arduino przez port  
szeregowy

```
#include "Buzz.h"
Buzz buzz;

void setup() {
  Serial.begin(115200);
  buzz.begin(A0, 50, 3000);
}

void loop() {
  Serial.println(buzz.level());
  delay(25);
}
```

Wgrywamy program na Arduino i włączamy Monitor portu szeregowego (Narzędzia >

Monitor portu szeregowego). Powinniśmy zaobserwować zmianę wartości na wykresie. Gdy czujnik nie wykrywa ruchu, to wartości niewiele się zmieniają. Kiedy poruszymy ręką w pobliżu czujnika, to powinniśmy zauważyć nagły wzrost lub spadek wartości. Poniżej przedstawiony jest przykładowy wykres:



Możemy dać uczniom chwilę, żeby poeksperymentowali z czujnikiem i sami poszukali zależności pomiędzy ruchem swojej ręki, a wartościami odczytywanymi w Arduino. Można również polecić uczniom, żeby sprawdzili, jaki zasięg ma ich czujnik i z jakiej odległości reaguje na ruch.

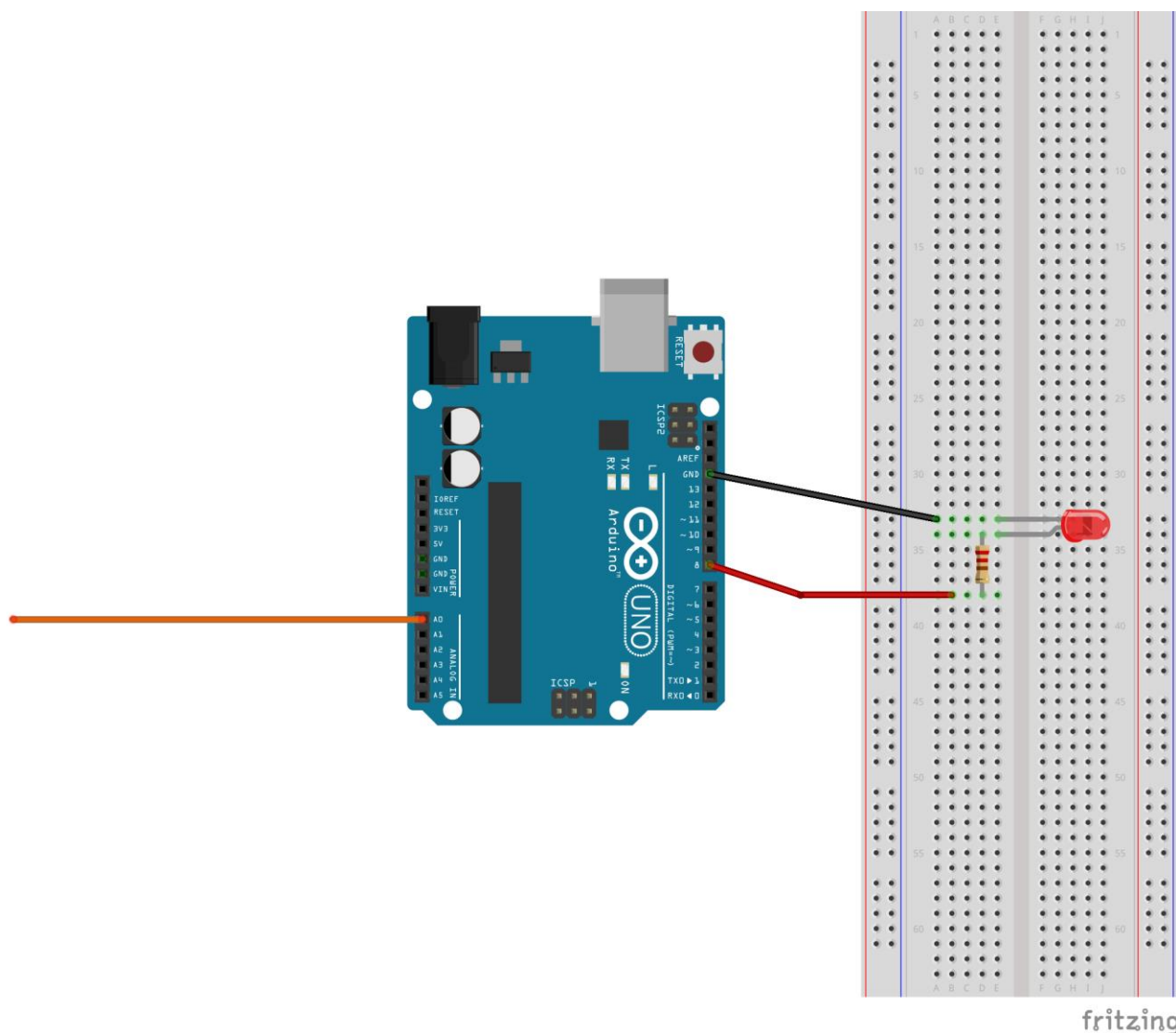
**W tym miejscu możliwy jest podział zajęć na dwie części (kolejna część scenariusza będzie realizowana na następnych zajęciach).**

## Przypomnienie materiału, odtworzenie układu i programu – 15 minut

Rozpoczynamy od przypomnienia materiału z poprzedniej części zajęć i odtworzenia układu oraz programu napisanego na poprzednich zajęciach.

## Budujemy prosty alarm – 30 minut

Po pierwszych eksperymentach przystępujemy do napisania programu, który będzie włączać jakiś akuator, kiedy czujnik impedancyjny wykryje ruch. Budujemy bardzo podobne urządzenie do tego, które powstawało w ramach scenariuszy o czujniku ruchu PIR czy fotokomórce. Efekt jest podobny, a uzyskujemy go stosując trzy skrajnie różne sposoby wykrywania ruchu.





Piszemy wspólnie z uczniami poniższy program:

ta część jest podobna do poprzedniego programu

odczytujemy wartość z pinu A0

przekształcamy wartość pomiaru na wartość bezwzględną za pomocą funkcji abs()

tutaj ustalamy próg czułości

```
#include "Buzz.h"
Buzz buzz;
int val;
int ledPin = 6;

void setup() {
  Serial.begin(115200);
  buzz.begin(A0, 50, 3000);

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void loop() {
  val = abs(buzz.level());
  Serial.println(buzz.level());

  if (val > 100) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
  delay(25);
}
```

Próg czułości czujnika ustalamy indywidualnie i możemy go dowolnie zmieniać. Często wartość tego progu trzeba dostosować do warunków w których wykonywany jest pomiar.

Kod działa bardzo podobnie jak w przypadku fotokomórki. Po przekroczeniu określonego progu uruchamiamy jakiś aktuator. W tym wypadku jest to dioda LED.

### 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 z dokładnym podłączaniem przewodów.

### ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole buduje czujnik impedancyjny i testuje jego działanie. Zadanie końcowe polega na

zrobieniu prostego modelu syreny alarmowej. Po wykryciu ruchu Arduino powinno włączyć buzzer, który będzie piszczeć w odstępach 0,3 sekundy.

### **FIGUŁKA 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 płytki stykowa (prototypowa):

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

Jak działa biblioteka Buzz:

<https://github.com/connornishijima/arduino-buzz>

Jak instalować biblioteki w Arduino:

<http://mikrokontroler.pl/2014/04/01/krok-po-kroku-arduino-jak-zaimportowac-biblioteki/>

*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/).