

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

**SCENARIUSZ 7. PRZYCISK**

*scenariusz zajęć pozalekcyjnych*

autor: Wojciech Karcz

redakcja: Agnieszka Koszowska

**SŁOWA KLUCZOWE:**

Arduino, programowanie, przycisk, button, digitalRead(), dioda LED, Arduino IDE

**KRÓTKI OPIS ZAJĘĆ:**

Przyciski są nieodłącznymi elementami różnych urządzeń elektronicznych. Podczas zajęć uczniowie i uczennice uczą się, jak działa prosty przycisk i jak za pomocą przycisku kontrolować Arduino. Montują układ elektroniczny wykorzystując Arduino, płytkę stykową, **diodę LED**, **oporniki** i **przycisk THT**. Wykonują zadanie: tworzą taki układ i program, który odwraca działanie przycisku: dioda świeci się cały czas, gdy przycisk nie jest naciśnięty, a po naciśnięciu gaśnie.

**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,
- wie, czym jest przycisk i na jakiej zasadzie działa,
- potrafi wykorzystać funkcję digitalRead() w programowaniu 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”):**

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 stykowe,
- oporniki 220 omów i 10K omów,
- przewody połączeniowe,
- przyciski (tzw. przycisk THT),
- 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.

## **Jak działają przyciski? – 10 min**

Podczas tych zajęć skupiamy się na elemencie elektronicznym, z którego większość z nas korzysta na co dzień, czyli przycisku. Zasada jego działania jest bardzo prosta. Jest to element, który po wciśnięciu lub przełączeniu zamyka albo przerywa obwód elektryczny. Prostą ilustracją działania przycisku jest zapalenie i gaszenie lampki biurkowej. Po wciśnięciu przycisku lampki zwieramy obwód elektryczny, przez który może popłynąć prąd elektryczny i rozświecić żarówkę. Kiedy przełączymy przycisk, rozłączymy obwód, wówczas prąd nie będzie płynąć i lampka zgaśnie.

Wykorzystanie przycisków jest najprostszą metodą komunikowania się człowieka z urządzeniami elektronicznymi. Oprócz prostego przerywania i łączenia obwodu, naciśnięcie przycisku może spowodować wykonanie różnych skomplikowanych zadań. Na przykład naciskamy przycisk windy, który daje sygnał odpowiednim sterownikom, a one z kolei uruchamiają szereg procesów sprowadzając windę na piętro.

Podczas zajęć zajmiemy się wykrywaniem przez Arduino momentu, w którym zostanie wciśnięty przycisk. Umożliwi nam to różnych przeróżnych operacji, jednak na początek skupimy się na zapaleniu diody LED.

Zadajemy uczniom np. takie pytania:

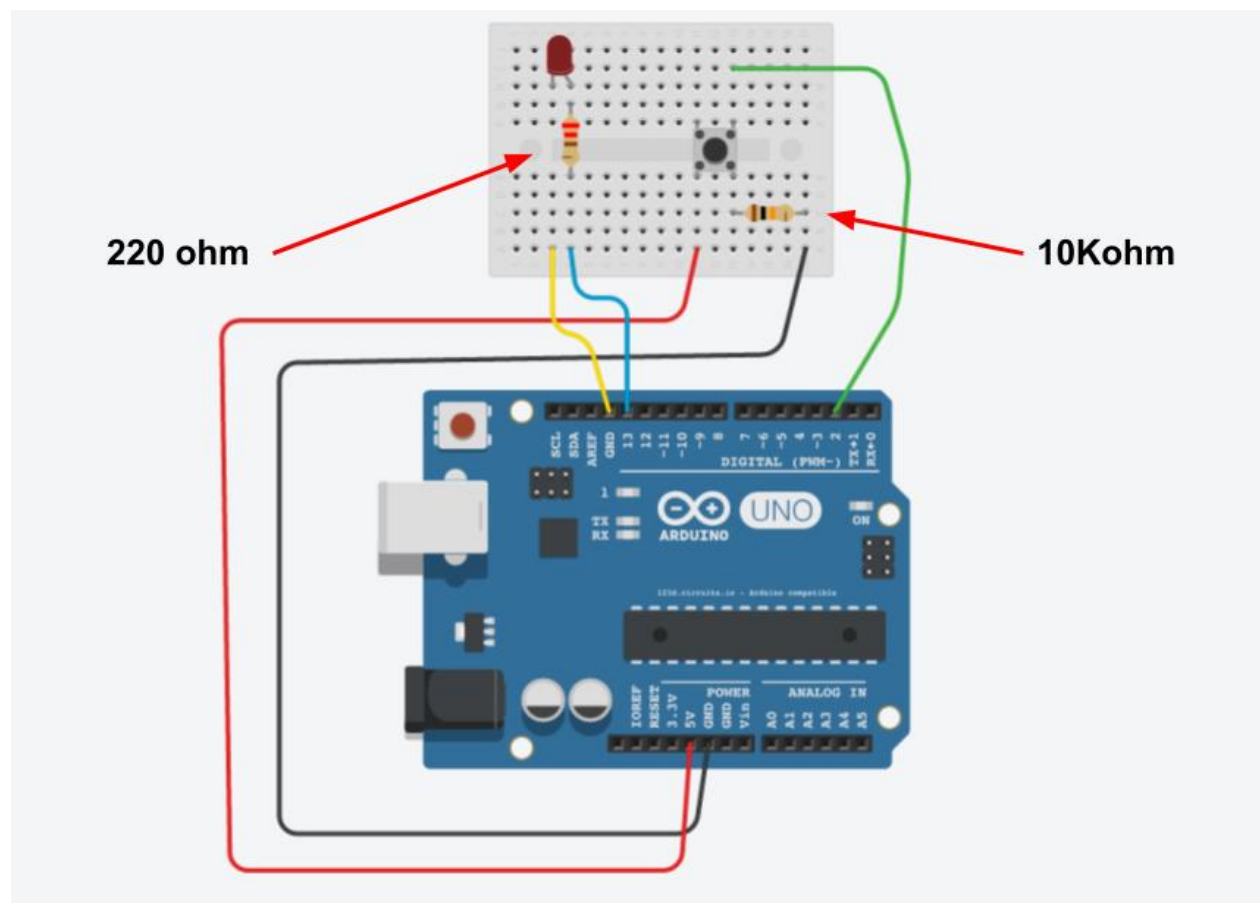
- *Czy wykorzystujecie przyciski?*
- *Do czego według Was służy przycisk?*
- *Jakie przyciski znacie?*
- *Jakie urządzenia posiadają przyciski?*

Podsumowujemy dyskusję. Nie powinno być tutaj problemu z odpowiedziami, ponieważ przyciski są dosłownie wszędzie i nawet młodsze dzieci powinny bez problemu wskazać przykłady ich zastosowania w różnych urządzeniach.

Następnie tłumaczymy, w jaki sposób działa przycisk. Warto też przypomnieć podstawy związane z elektroniką, obwodami elektronicznymi, przewodnikami, ponieważ te tematy są ściśle ze sobą powiązane. Więcej informacji na ten temat można znaleźć w scenariuszu poświęconym diodzie LED.

## Montujemy układ – 20 minut

Wspólnie z uczniami montujemy układ przedstawiony na schemacie poniżej:



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

## Przypomnienie materiału, odtworzenie układu z przyciskiem – 10 minut

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

## Programowanie przycisku THT – 20 minut

Wspólnie z uczniami piszemy program wykrywający działanie przycisku. Kod programu:

```

int buttonPin = 2;    // definiujemy pin przycisku
int ledPin = 13;     // definiujemy pin diody LED

int buttonState = 0; // zmienna przechowująca stan przycisku

void setup() {
  // definiujemy pin ledPin jako pin wyjściowy
  pinMode(ledPin, OUTPUT);
  // definiujemy pin buttonPin jako pin wejściowy
  pinMode(buttonPin, INPUT);
}

void loop() {
  // odcytujemy wartość stanu pinu buttonPin
  buttonState = digitalRead(buttonPin);

  // sprawdzamy czy przycisk jest wciśnięty, jeśli tak, to wartość
  // zmiennej buttonState powinna wynosić HIGH
  if (buttonState == HIGH) {
    // włączamy diodę LED:
    digitalWrite(ledPin, HIGH);
  } else {
    // wyłączamy diodę LED
    digitalWrite(ledPin, LOW);
  }
}

```

definiujemy piny wejściowe/wyjściowe

odczytujemy stan przycisku za pomocą funkcji digitalRead()

sprawdzamy, czy przycisk jest włączony

Po napisaniu programu, przesyłamy go na Arduino i sprawdzamy, czy działa. Po wciśnięciu przycisku i przytrzymaniu dioda powinna się zapalić. Kiedy puścimy przycisk, dioda LED gaśnie.

Omawiamy z uczniami cały kod programu krok po kroku. Na początku głównej pętli void loop() odcytujemy stan pinu, do którego podpieliśmy przycisk za pomocą funkcji **digitalRead()**. Dzięki tej funkcji otrzymujemy informację, czy na pinie jest stan niski (LOW, nie płynie prąd), czy wysoki (HIGH, płynie prąd). Możemy odczytać tylko te dwie wartości. Warto też tutaj zapytać uczniów, jaką widzą różnicę w odczytywaniu wartości za pomocą funkcji **digitalRead()**, a **analogRead()**, którą stosowaliśmy np, w scenariuszu poświęconym fotokomórce.

Następnie zwracamy uwagę na instrukcję warunkową „if”. W tym przypadku stosujemy operator logiczny „==” oznaczający równoważność. Jeżeli zmienna „**buttonState**” ma tą samą wartość (czyli stan wysoki - HIGH), to program wykonuje kod zawarty w pierwszym nawiasie klamrowym. Jeśli nie, to program wykonuje kod w nawiasie klamrowym po słowie „else”. Przekładając to na działanie przycisku: jeśli wciśniemy przycisk, to zwieramy obwód, wykrywamy na pinie wejściowym stan wysoki (HIGH, ponieważ płynie prąd), porównujemy tę wartość w instrukcji warunkowej i zapalamy diodę LED za pomocą funkcji digitalWrite(), ponieważ warunek jest spełniony. W innym przypadku program wykonuje drugą część kodu i wyłącza diodę LED.

Kod programu można jeszcze dodatkowo rozbudować (w miarę możliwości czasowych oraz umiejętności uczniów) o dodanie funkcji przesyłających dane o stanie pinu wejściowego do Monitora Portu Szeregowego. Wystarczy dodać dwie funkcje w miejscach zaznaczonych poniżej:

tutaj dodajemy funkcje przesyłające dane o stanie przycisku przez port szeregowy

```
int buttonPin = 2;
int ledPin = 13;

int buttonState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  Serial.println(buttonState);

  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Źródło: Materiały własne

Następnie wgrujemy program, włączamy Monitor Portu Szeregowego (Narzędzia > Monitor Portu szeregowego) i obserwujemy zmiany wartości po wciśnięciu przycisku. Pytamy uczniów, czy widzą jakieś zmiany?

Po wciśnięciu przycisku powinniśmy widzieć **1**, a kiedy przycisk jest nieużywany **0**, co odpowiada kolejno stanowi wysokiemu (HIGH, 1) oraz niskiemu (LOW, 0).

## 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 układ z wykorzystaniem Arduino, płytki stykowej, diody LED i oporników. Zadanie polega na odwróceniu działania przycisku: dioda świeci się cały czas, gdy przycisk nie jest naciśnięty, a po naciśnięciu gaśnie.

Bardziej ambitnym uczniom można dać trudniejsze zadanie polegające na stworzeniu programu, który będzie symulował działanie zwykłej lampki nocnej. Kiedy wciśniemy raz przycisk, to dioda LED będzie świecić się cały czas, a kiedy kolejny raz naciśniemy przycisk, to zgaśnie. **Podpowiedź:** trzeba wprowadzić dodatkową zmienną, która będzie nam przechowywać informację o tym, czy dioda jest włączona, czy nie.

## 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>

Informacja o diodach LED:

[https://pl.wikipedia.org/wiki/Dioda\\_elektroluminescencyjna](https://pl.wikipedia.org/wiki/Dioda_elektroluminescencyjna)

Jak działa płytka stykowa (prototypowa):

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

Podstawowe informacje na temat prądu elektrycznego:

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

Czym są operatory logiczne w programowaniu:

[https://pl.wikipedia.org/wiki/Operator\\_logiczny](https://pl.wikipedia.org/wiki/Operator_logiczny)



*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/).