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

SCENARIUSZ 3. SYRENA ALARMOWA

scenariusz zajęć pozalekcyjnych

autor: Wojciech Karcz

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

Arduino, programowanie, mikrokontroler, dioda LED, PWM, Pulse Width Modulation, Modulacja Szerokości Impulsu

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczniowie i uczennice budują prosty model syreny alarmowej, wykorzystujący **buzzer (brzęczyk)** i wyjścia **Arduino** oznaczone symbolem **PWM** (Modulacja Szerokości Impulsu, ang. „Pulse Width Modulation”). Poznają i/lub utrwalają podstawowe pojęcia programistyczne (skrypt, program, algorytm, sterowanie, warunek, pętla). Korzystając z pętli **for** oraz właściwości PWM regulują jasność diody i głośność buzzera. Wykonują zadanie: budują syrenę alarmową z Arduino, czyli układ, w którym **buzzer** w płynny sposób zwiększa głośność wydawanych dźwiękow – od ciszy do maksymalnej wartości.

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 wyjście PWM w Arduino,
- wie, co to jest pętla „for”, potrafi zbudować taką pętlę,
- zna różnicę pomiędzy urządzeniami cyfrowymi a analogowymi,

- zna podstawowe elementy interfejsu środowiska programistycznego Arduino IDE i podstawowe komendy języka Arduino IDE: pinMode(), digitalWrite(), delay(),
- potrafi wykorzystywać funkcję **analogWrite()**,
- 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”):

1

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/),
- płytka Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytka stykowa,
- oporniki 220 omów,
- przewody połączeniowe,
- diody LED w różnych kolorach, buzzer (brzęczyk),
- 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.

Omawiamy i testujemy wyjścia typu PWM – 30 minut

Podczas zajęć pokażemy uczniom, w jaki sposób działają w Arduino złącza typu PWM (Pulse Width Modulation). Dzięki wykorzystaniu złącz PWM możemy symulować w mikrokontrolerach wyjścia analogowe. Podczas zajęć uczniowie nauczą się wykorzystywać tę funkcję do kontrolowania różnych urządzeń.

Rozpoczynamy zajęcia od dyskusji na temat urządzeń analogowych oraz cyfrowych. Zadajemy takie pytania jak:

- *Czym są urządzenia cyfrowe?*
- *Co to znaczy, że jakieś urządzenie jest cyfrowe?*
- *Jakie znacie urządzenia analogowe? Podajcie przykłady.*

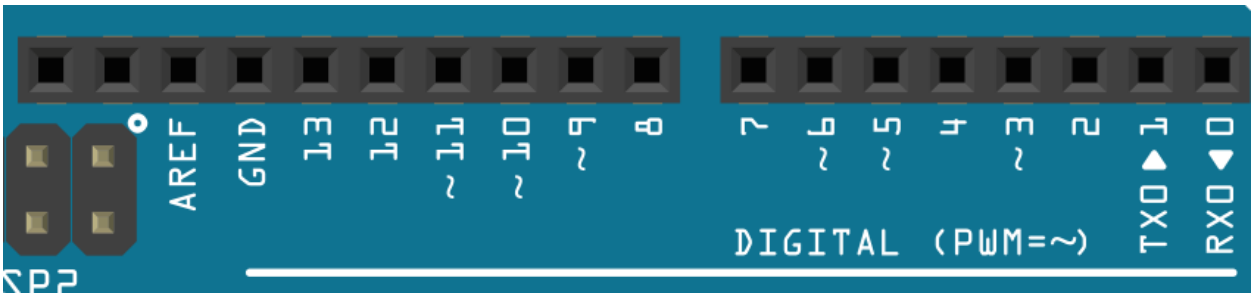
Następnie wyjaśniamy różnicę pomiędzy urządzeniami cyfrowymi (np. komputer, telefon komórkowy), a analogowymi (zegarek mechaniczny, stare radio). Tłumaczymy też, że wykorzystując w Arduino funkcję **digitalWrite()**, możemy nadać pinom dwa stany: wysoki (oznaczony liczbą 1 – wtedy płynie prąd) lub niski (oznaczonmy liczbą 0 – wtedy prąd nie płynie). W podobny sposób działają komputery, które odczytują informacje zapisanie w systemie binarnym składającym się tylko z 0 i 1. W przypadku sygnału analogowego możemy przypisać różne wartości z zakresu od 0 do 255 za pomocą funkcji **analogWrite()**, które odpowiadają wartościom z zakresu między 0V, a 5V. Więcej informacji na ten temat znajduje się w materiałach dodatkowych.

Dwie podstawowe funkcje kontrolujące piny wyjściowe w Arduino to:

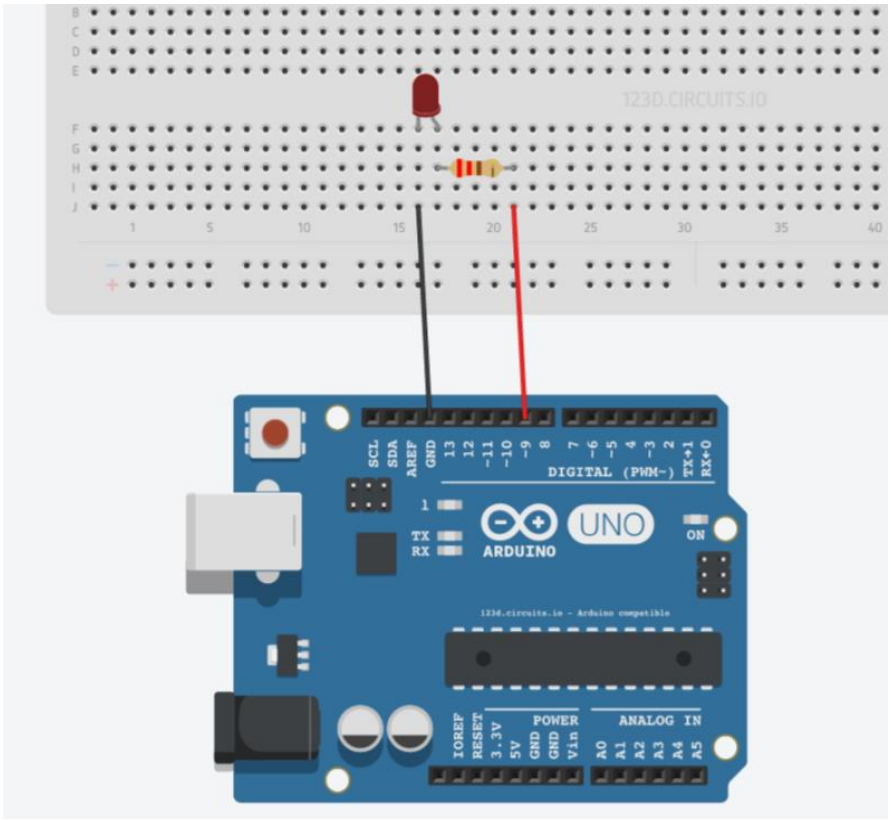
digitalWrite(0 lub 1, HIGH lub LOW) – za pomocą tej funkcji nadajemy stan niski lub wysoki dla pinów 0-13,

analogWrite(0-255) - za pomocą tej funkcji nadajemy wartości od 0 do 255, które odpowiadają napięciu na pinie wyjściowym w zakresie 0V (0) - 5V (255).

Warto zwrócić uwagę na oznaczenia pinów. W Arduino piny działające w trybie PWM są oznaczone **znakiem ~**.



Wspólnie z uczniami montujemy układ przedstawiony na schemacie poniżej.



Układ jest bardzo podobny do zastosowanego w scenariuszu o diodzie LED, tylko warto zwrócić uwagę, że pinem wyjściowym jest tutaj pin nr 9 (można oczywiście dowolnie wybrać jeden z sześciu pinów PWM).

Dioda LED ma dwie nóżki o różnej długości. Dłuższa jest anodą i do niej podpinamy opornik.

Następnie uruchamiamy program o nazwie Fading (Plik > Przykłady > 03.Analog > Fading). Kompilujemy szkic oraz wgrywamy program na Arduino. Pytamy uczniów,

w jaki sposób zachowuje się dioda LED. Czy jest to zachowanie cyfrowe, czy analogowe? (Poprawna odpowiedź to: analogowe, ponieważ dioda płynnie się rozjaśnia i gaśnie). Wyjaśniamy uczniom sposób działania diody w tym przypadku i możemy to odnieść do przykładu ze scenariusza o diodzie LED, gdzie sterowaliśmy diodą w sposób cyfrowy.

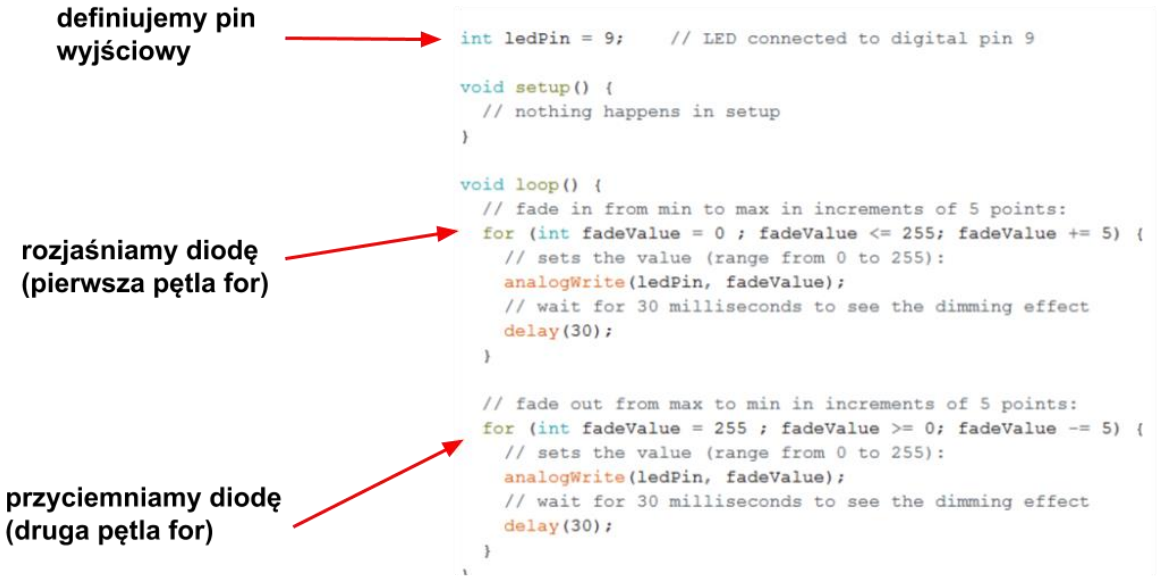
W tym miejscu możliwy jest podział zajęć na dwie części (kolejna część scenariusza będzie realizowana na następnych zajęciach).

Montujemy układ z diodą LED – przypomnienie – 15 minut

Rozpoczynamy zajęcia od krótkiego przypomnienia materiału z poprzednich zajęć i odtworzenia zbudowanego wcześniej układu. Ponownie uruchamiamy program Fading.

Pętla „for" - 15 minut

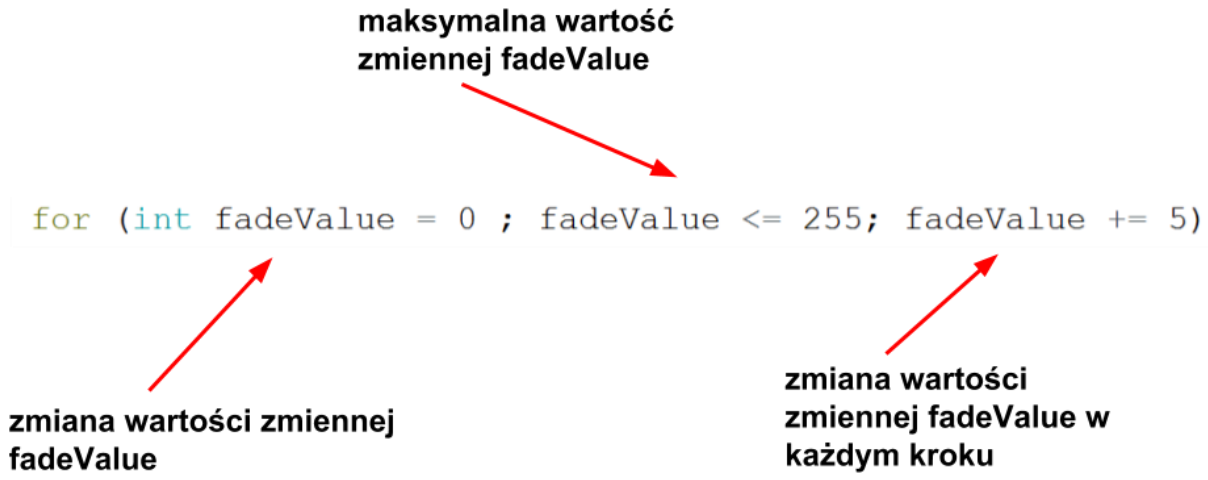
W tej części zajęć analizujemy z uczniami sposób działania programu Fading.



Omawiamy po kolei wszystkie elementy kodu. Zwracamy uwagę na pętlę „for” oraz funkcję analogWrite() zawartą wewnątrz tej pętli. Tłumaczymy, w jaki sposób działają pętle – tu wprowadzamy pojęcie pętli „for”. Warto zwrócić uwagę, że w głównej pętli Arduino znajdują się dwie pętle „for” następujące po sobie: pierwsza odpowiada za rozjaśnienie diody, a droga za jej przyciemnienie. Więcej informacji na temat pętli „for” można znaleźć w materiałach dodatkowych.

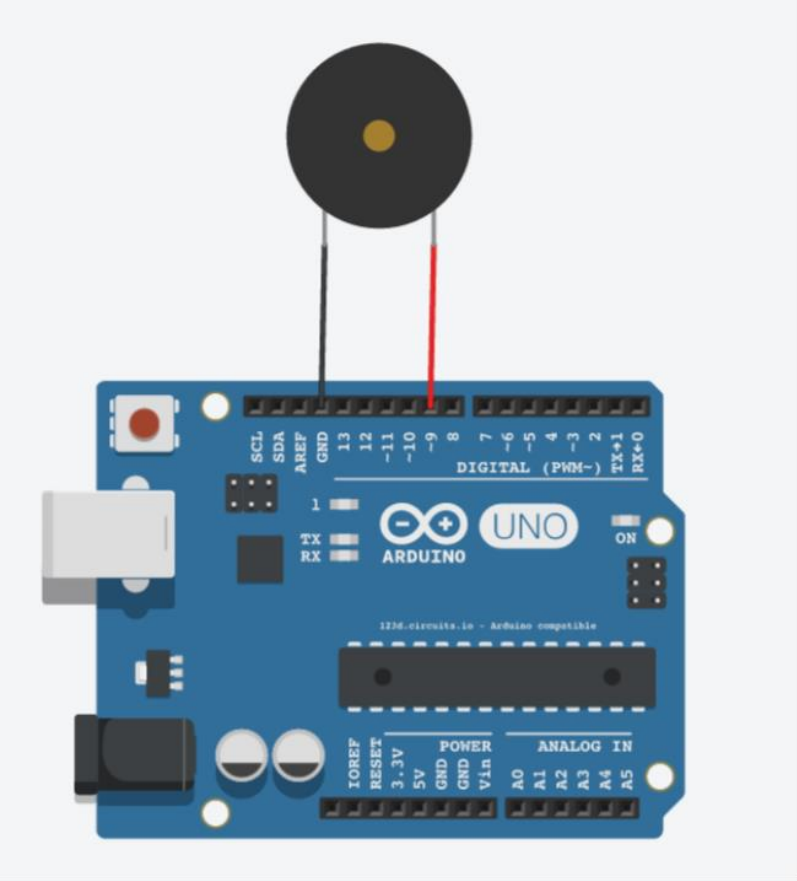
Pętla „for” działa w ten sposób, że operację zawartą wewnątrz pętli program wykonuje określoną przez nas liczbę razy. W tym przypadku mamy trzy parametry. Pierwszy parametr jest zmienną fadeValue, która na początku ma wartość 0, ponieważ zaczynamy rozjaśniać diodę od 0. Drugi parametr to warunek określający, do kiedy pętla będzie działać. W tym przypadku przerywamy działanie pętli, kiedy osiągniemy maksymalną wartość 255. Trzeci parametr to liczba, o którą podnosimy lub obniżamy wartość zmiennej fadeValue przy kolejnych cyklach pętli „for”. Tutaj jest to wyrażone symbolem +=, który oznacza wzrost o 5 przy każdym przebiegu pętli (tj. kolejno: 0, 5, 10, 15, 20, etc.). Symbol -= (w drugiej pętli „for”) oznacza spadek o 5 (tj. kolejno: 255, 250, 245, 240, etc.).

Następnie dajemy uczniom zadanie: należy zmodyfikować parametry dwóch pętli „for” i sprawdzić, jak wpływa to na zachowanie diody. Poniżej znajduje się podpowiedź, które parametry można zmieniać i na co wpływają:



Montujemy układ z brzęczykiem – 15 minut

Kiedy poznaliśmy już działanie złącz PWM i przetestowaliśmy ich działanie na diodzie LED, to teraz wspólnie z uczniami sprawdzimy, jak będą w tym trybie działać inne aktuatory (urządzenia wyjściowe). Wspólnie z uczniami montujemy poniższy prosty układ na płytce stykowej z wykorzystaniem buzzera (brzęczyka):



Jeszcze raz wgrywamy program Fading z poprzedniego punktu i sprawdzamy, w jaki sposób zachowuje się buzzer. Zadajemy uczniom pytanie, czy widzą różnicę w zachowaniu buzzera i diody LED. A może nie ma takiej różnicy? Buzzer powinien zachowywać się tak samo, czyli wydawać – na przemian – cichszy i głośniejszy dźwięk.

Podobnie jak w poprzedniej części zajęć dajemy uczniom chwilę na samodzielne manipulowanie parametrami pętli „for” i testowanie, jak to wpływa na zachowanie buzzera.

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, w którym buzzer w płynny sposób zwiększa głośność wydawanych dźwięków – od ciszy do maksymalnej wartości, następnie piszczy przez 3 sekundy z najwyższą głośnością, a potem cykl powtarza się.

PIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Kurs programowania Arduino Forbot:
http://forbot.pl/blog/artykuly/programowanie/kurs-arduino-w-robotyce-1-wstep-id936

Czym jest PWM:
https://pl.wikipedia.org/wiki/Modulacja_szeroko%C5%9Bci_impuls%C3%B3w

Różnica pomiędzy sygnałem cyfrowym, a analogowym:
http://livesound.pl/tutoriale/kursy/3938-technika-cyfrowa-podstawy-zacznijmy-od-pocztku

Pojęcie pętli w programowaniu:
https://pl.wikipedia.org/wiki/P%C4%99tla_(informatyka)

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).