

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

SCENARIUSZ 4. SYGNALIZACJA ŚWIETLNA

scenariusz zajęć pozalekcyjnych

autor: Wojciech Karcz

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

Arduino, programowanie, mikrokontroler, dioda RGB, dioda LED, analogWrite()

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczniowie i uczennice budują prosty model sygnalizacji świetlnej, wykorzystując **diodę RGB**, która może świecić w dowolnym kolorze. Poznają i/lub utrwalają podstawowe pojęcia programistyczne (skrypt, program, algorytm, sterowanie, warunek, pętla). Uczą się kontrolować diodę RGB za pomocą Arduino. Wykonują zadanie: konstruują „system” sygnalizacji świetlnej z wykorzystaniem diody RGB programowanej za pomocą Arduino.

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 system RGB, zna pojęcie „paleta kolorów RGB”,
- wie, co to jest dioda RGB i potrafi ją zaprogramować za pomocą Arduino,
- 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/>),
- płytko Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytko stykowa,
- oporniki 220 ohm,
- przewody połączeniowe,
- dioda RGB,
- 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 diodę RGB – 30 minut

Zapowiadamy, że podczas zajęć uczniowie poznają diodę RGB, która może świecić w dowolnym kolorze, gdyż działa zgodnie z systemem RGB, pozwalającym na mieszanie i dobieranie różnych kolorów.

Dioda RGB to rodzaj diody LED, składający się z trzech mniejszych diod: czerwonej (R – ang. „red”), zielonej (G – ang. „green”) i niebieskiej (B – ang. „blue”). Stąd też

pochodzi jej nazwa „RGB” – od angielskich nazw kolorów. Dioda posiada cztery nóżki, spośród których trzy odpowiadają za poszczególne kolory, a czwarta stanowi wspólną anodę lub katodę. Diody RGB występują w tych dwóch wersjach, lecz w tym przypadku będziemy omawiać diody RGB ze wspólną katodą.

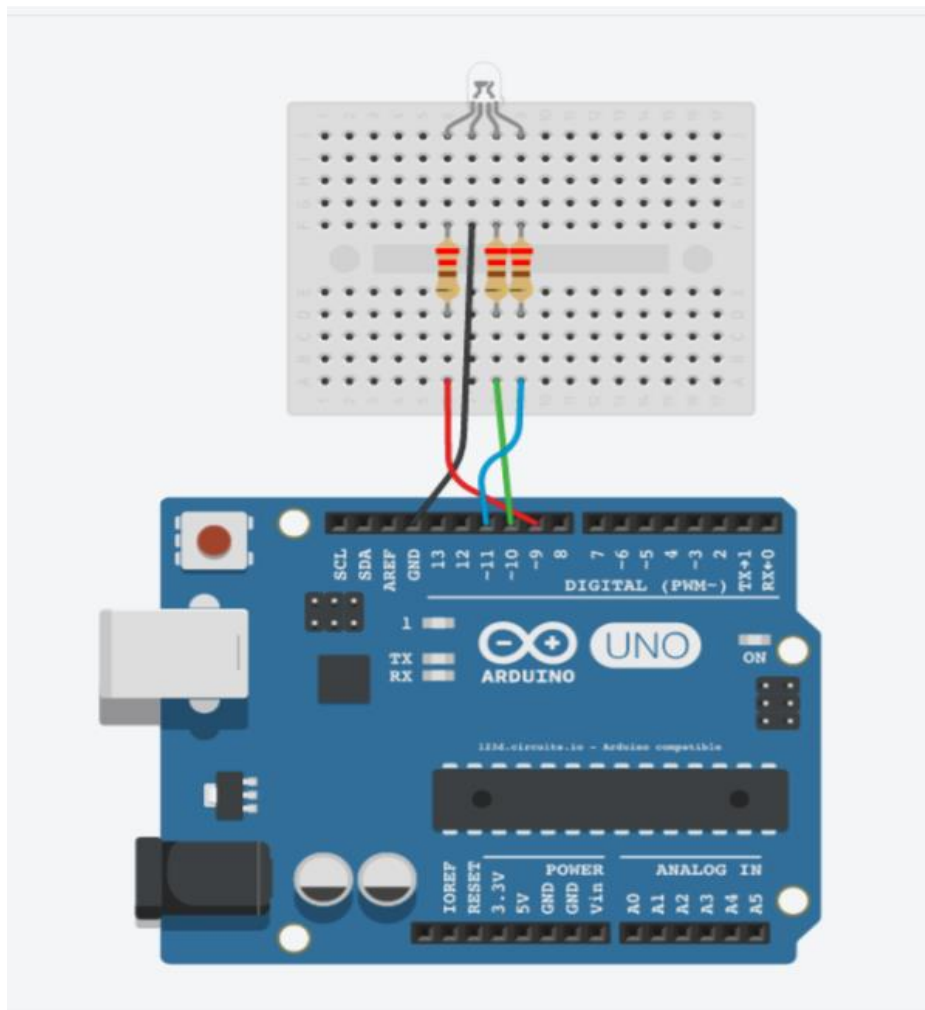
Podając napięcie do konkretnej nóżki uruchamiamy odpowiadający jej kolor. Korzystając ze złącz PWM i funkcji `analogWrite()` możemy sterować jasnością każdego kanału. Ustalając różne wartości napięcia na wszystkich trzech kanałach możemy dowolnie mieszać barwy składowe, otrzymując tym samym pełną paletę kolorów RGB.

Rozpoczynamy od dyskusji dotyczącej kolorów oraz ekranów/wyświetlaczy. Zadajemy uczniom pytania:

- *Jakie znacie typy wyświetlaczy i/ lub ekranów? Gdzie się je stosuje? W jakich urządzeniach?*
- *Jak działa wyświetlacz?*
- *Co to jest piksel?*
- *Co się dzieje, kiedy mieszamy różne kolory, np. farb plakatowych?*

Zbieramy i podsumowujemy wszystkie odpowiedzi. Wprowadzamy pojęcie „palety kolorów RGB”. Porządkujemy wiedzę związaną z powstawaniem kolorów oraz mieszaniem barw. Tłumaczymy, że prawie każdy ekran (telewizor, monitor, smartfon itp.) zawiera w sobie miniaturowe piksele. Każdy z tych pikseli może świecić w trzech kolorach (czerwony, zielony, niebieski) i dzięki zmieszaniu tych barw możemy otrzymać dowolny kolor. Więcej informacji na ten temat znajduje się w materiałach dodatkowych.

W tej części wspólnie podłączamy diodę RGB do Arduino wg poniższego schematu:



UWAGA: do każdej nóżki diody odpowiadającej za dany kolor podpinamy rezystor (może być 330 ohm). Tak jak wcześniej wspominaliśmy, dioda RGB jest zbudowana z trzech mniejszych diod i każdą z nich sterujemy niezależnie. Jednak ta dioda RGB ma wspólną katodę i dzięki temu wystarczy tylko jeden kabelek podłączony do naszego minusa w Arduino (pin GND). Warto pomóc uczniom w przypinaniu, ponieważ mogą wystąpić problemy z rozłożeniem nóżek diody RGB i wetknięciem ich w otwory płytki stykowej.

Po sprawdzeniu wszystkich połączeń wgrujemy na Arduino poniższy program:

**definiowanie
pinów
wyjściowych**

```
const int RED_PIN = 9;  
const int GREEN_PIN = 10;  
const int BLUE_PIN = 11;
```

```
void setup()  
{  
  
pinMode(RED_PIN, OUTPUT);  
pinMode(GREEN_PIN, OUTPUT);  
pinMode(BLUE_PIN, OUTPUT);  
}
```

**wszystkie kolory
wyłączone**

```
void loop()  
{  
  
digitalWrite(RED_PIN, LOW);  
digitalWrite(GREEN_PIN, LOW);  
digitalWrite(BLUE_PIN, LOW);  
  
delay(1000);
```

kolor: czerwony

```
digitalWrite(RED_PIN, HIGH);  
digitalWrite(GREEN_PIN, LOW);  
digitalWrite(BLUE_PIN, LOW);  
  
delay(1000);
```

kolor: zielony

```
digitalWrite(RED_PIN, LOW);  
digitalWrite(GREEN_PIN, HIGH);  
digitalWrite(BLUE_PIN, LOW);  
  
delay(1000);
```

kolor: niebieski

```
digitalWrite(RED_PIN, LOW);  
digitalWrite(GREEN_PIN, LOW);  
digitalWrite(BLUE_PIN, HIGH);  
  
delay(1000);
```

kolor: żółty

```
digitalWrite(RED_PIN, HIGH);  
digitalWrite(GREEN_PIN, HIGH);  
digitalWrite(BLUE_PIN, LOW);  
  
delay(1000);
```

kolor: cyjan

```
digitalWrite(RED_PIN, LOW);  
digitalWrite(GREEN_PIN, HIGH);  
digitalWrite(BLUE_PIN, HIGH);  
  
delay(1000);
```

kolor: purpura

```
digitalWrite(RED_PIN, HIGH);  
digitalWrite(GREEN_PIN, LOW);  
digitalWrite(BLUE_PIN, HIGH);  
  
delay(1000);
```

kolor: biały

```
digitalWrite(RED_PIN, HIGH);  
digitalWrite(GREEN_PIN, HIGH);  
digitalWrite(BLUE_PIN, HIGH);  
  
delay(1000);  
}
```

Możemy wspólnie z uczniami wpisać ten kod ręcznie lub przygotować go wcześniej i rozesłać na ich komputery. Decyzja zależy od osoby prowadzącej oraz czasu na zajęciach.

Po wgraniu programu obserwujemy, w jaki sposób zachowuje się dioda. Zadajemy uczniom pytania:

- *Jakie kolory widzicie?*
- *W jakiej sekwencji zmieniają się kolory?*
- *Czy widzicie, jak świecą się trzy mniejsze diody dając składowe barwy?*

Następnie wspólnie z uczniami analizujemy kod programu. Tłumaczymy, że każdy pin odpowiada za inny kolor. Pokazujemy, w którym miejscu włączamy, a w którym wyłączamy poszczególne kanały (miejsca, gdzie jest stan niski lub wysoki). Możemy uczniom dać zadanie, żeby rozszyfrowali, które miejsce kodu odpowiada za konkretny kolor, wyświetlany przez diodę RGB. Dzięki temu znacznie łatwiej rozumieją działanie kodu oraz całego układu.

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

Przypomnienie materiału, odtworzenie układu z diodą RGB – 10 minut

Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i odtworzenia układu zbudowanego na poprzednich zajęciach. Uczniowie podłączają diodę RGB i wgrywają program.

Sterowanie diodą za pomocą złącz PWM – 25 minut

W tej części zajęć będziemy sterować diodą RGB za pomocą funkcji `analogWrite()`, na temat której więcej informacji znajduje się w scenariuszu poświęconym złączom PWM (Scenariusz 3. „Syrena alarmowa”). Podobnie jak w tamtym przypadku, tutaj również możemy przypisywać wartości z zakresu 0-255 i tym samym symulować wyjście analogowe. Warto zwrócić uwagę na odpowiednie podpięcie przewodów do pinów wyjściowych Arduino. Cały czas nie zmieniamy układu i zostawiamy diodę podpiętą do pinów 9, 10, 11 (złącza PWM).

Wspólnie z uczniami piszemy nowy prosty program pozwalający regulować jasność poszczególnych kanałów. Warto zwrócić uwagę, żeby polecenie digitalWrite() zmienić na analogWrite(), ponieważ w innym wypadku nie będziemy mogli przypisywać wartości analogowych w zakresie 0-255. Poniżej znajduje się przykładowy program:

```
const int RED_PIN = 9;
const int GREEN_PIN = 10;
const int BLUE_PIN = 11;

void setup()
{

pinMode(RED_PIN, OUTPUT);
pinMode(GREEN_PIN, OUTPUT);
pinMode(BLUE_PIN, OUTPUT);
}

void loop(){

analogWrite(RED_PIN, 0);
analogWrite(GREEN_PIN, 0);
analogWrite(BLUE_PIN, 0);

delay(1000);

analogWrite(RED_PIN, 255);
analogWrite(GREEN_PIN, 255);
analogWrite(BLUE_PIN, 255);

delay(1000);

}
```

tutaj zmieniamy wartości i możemy kontrolować intensywność każdej barwy

Omawiamy pokrótce z uczniami kod programu i pokazujemy, gdzie mogą samodzielnie zmienić parametry i tym samym mieszać barwy. Po wgraniu programu, pytamy uczniów, czy widzą różnicę oraz jaki kolor udało im się uzyskać. Możemy dać im ok. 10 min na samodzielne eksperymentowanie z tworzeniem różnych kolorów.

Konstruujemy prosty system sygnalizacji świetlnej - 10 minut

Zadaniem uczniów jest stworzenie prostego symulatora drogowej sygnalizacji świetlnej. Dioda RGB powinna świecić tak samo jak światło drogowe w następującej sekwencji: światło zielone, żółte, czerwone, czerwone, żółte, zielone.

Warto najpierw wspólnie z uczniami zastanowić się, jak działa takie światło? Które kolory świecą krótko, a które dłużej? Poniżej znajduje się przykładowy algorytm – sekwencja świateł, na której można się wzorować tworząc program:

światło zielone – 15 s

światło żółte – 3 s

światło czerwone – 15 s

światło czerwone i żółte (jednocześnie) – 2 s

Wszystko zawieramy w głównej pętli i dzięki temu otrzymujemy działającą w sposób ciągły sygnalizację świetlną.

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 model sygnalizacji świetlnej, wzorowanej na światłach drogowych, z wykorzystaniem Arduino i diody RGB. Dioda RGB w zbudowanym układzie powinna się świecić w różnych kolorach w następującej sekwencji: światło zielone, żółte, czerwone, czerwone, żółte, zielone.

FIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH

Kurs programowania Arduino Forbot:

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

Informacje o systemie RGB:

<https://pl.wikipedia.org/wiki/RGB>

Przewodnik pokazujący, jak podłączyć do Arduino diodę RGB:

<https://majsterkowo.pl/podlaczymy-diode-rgb-ze-wspolna-anoda-arduino/>

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/).