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

SCENARIUSZ 28. POTENCJOMETR

scenariusz zajęć pozalekcyjnych

autor: Kamil Kociszewski

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

Arduino, potencjometr, dioda LED, pomiar, napięcie

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczestnicy uczą się praktycznego wykorzystania **potencjometru**. W pierwszej części poznają sposób badania wartości na wejściu analogowym przy pomocy funkcji **Serial.println**. W drugiej części tworzą wskaźnik ostrzegający o dużym napięciu wykorzystując 3 diody LED.

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),
- zna pojęcie rezystora, potencjometru,
- potrafi wykorzystać możliwości potencjometru w zastosowaniach praktycznych,
- Wykorzystuje praktycznie funkcję If/Else, Serial.println, digitalWrite, delay zna podstawowe elementy interfejsu środowiska programistycznego Arduino IDE i podstawowe komendy języka Arduino IDE: pinMode(), digitalWrite(), delay(),
- 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”):

4

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,
- potencjometr,
- rezystory 220 omów,
- 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.

Na tych zajęciach uczniowie będą wykorzystywali wyjścia cyfrowe, piny zasilające oraz wejścia analogowe. Przedstawiamy wejścia analogowe jako elementy umożliwiające pomiar napięcia, a także omawiamy sposób pomiaru napięcia. Można nawiązać do wiedzy nabytej na lekcjach fizyki, jeśli uczniowie już posiadają tę wiedzę.

Omawiamy zasadę działania rezystora, montujemy układ – 15 minut

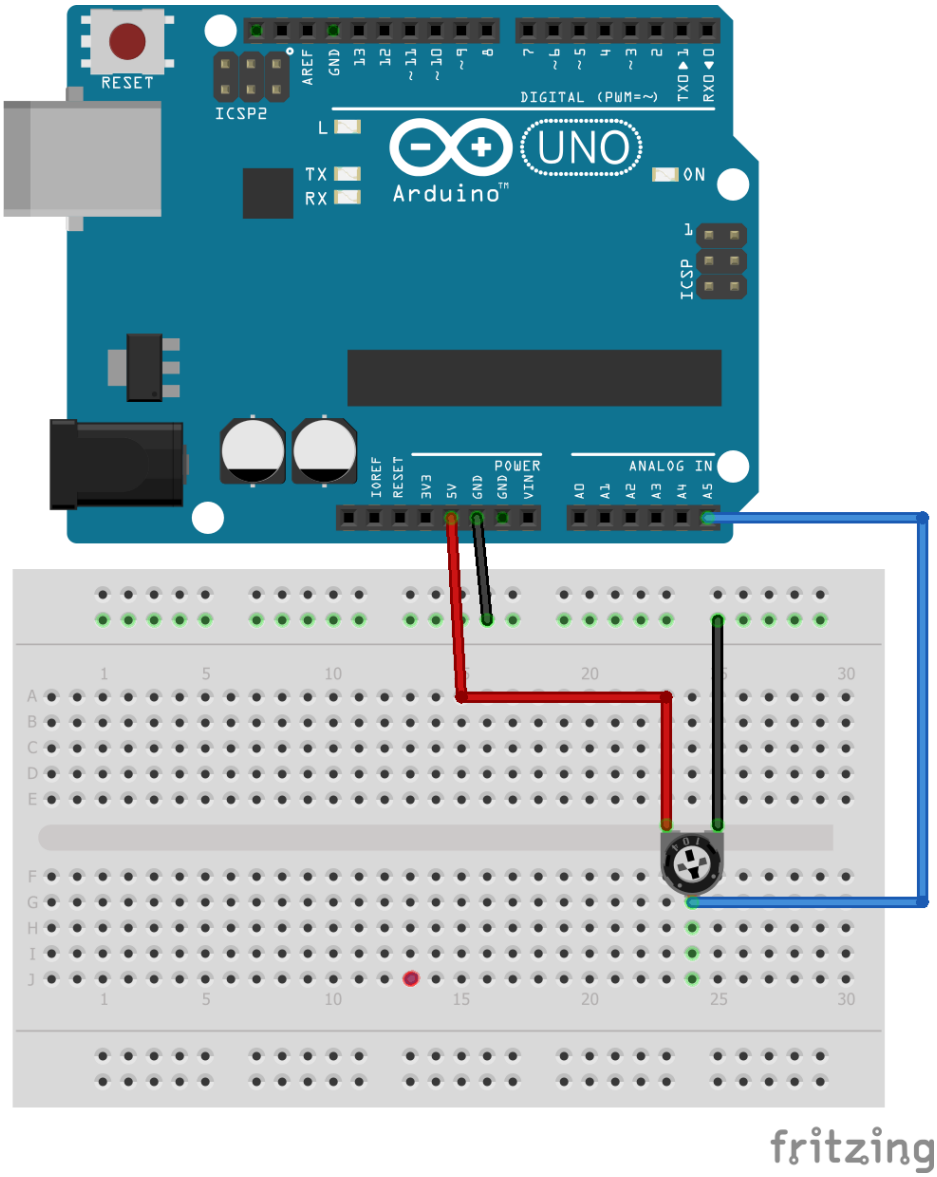
Pytamy uczniów, czym jest rezystor. Wspólnie dochodzimy do wniosku, że jest to element, który stanowi przeszkodę dla przepływu prądu. Im rezystor ma większą wartość w omach, tym większy opór stawia prądowi. Zmienia on energię elektryczną w ciepło. Potencjometr jest wyjątkowym rezystorem, ponieważ możemy w sposób płynny zmieniać jego rezystancję. Pozwala nam to zmieniać napięcie w płynącym obwodzie.

Przedstawiamy uczniom problem, który będą rozwiązywać podczas zajęć. Celem jest stworzenie systemu informującego o odpowiednim napięciu, ostrzegającego o dużym i o zbyt dużym napięciu. Zadajemy uczniom pytania:

- *Czy duże napięcie jest niebezpieczne?*
- *Jaki system ostrzegania możemy stworzyć, mając do dyspozycji 3 kolorowe diody i potencjometr?*

Jeśli uczniowie i uczennice posiadają już dostateczną wiedzę na temat Arduino, mogą w grupie przygotować propozycję projektu układu. W tej sytuacji nauczyciel moderuje grupę oraz pomaga rozwiązać ewentualne trudności. Sugerowanym trybem działania jest zadawanie pytań mających na celu naprowadzenie młodzieży na właściwe rozwiązania i sygnalizowanie potencjalnych problemów.

Uczniowie i uczennice wspólnie łączą elementy, z których powstanie układ elektronicznego woltomierza. W tym czasie zadaniem nauczyciela jest monitorowanie postępów i upewnianie się, że wszyscy nadążają.



Programowanie układu – 15 minut

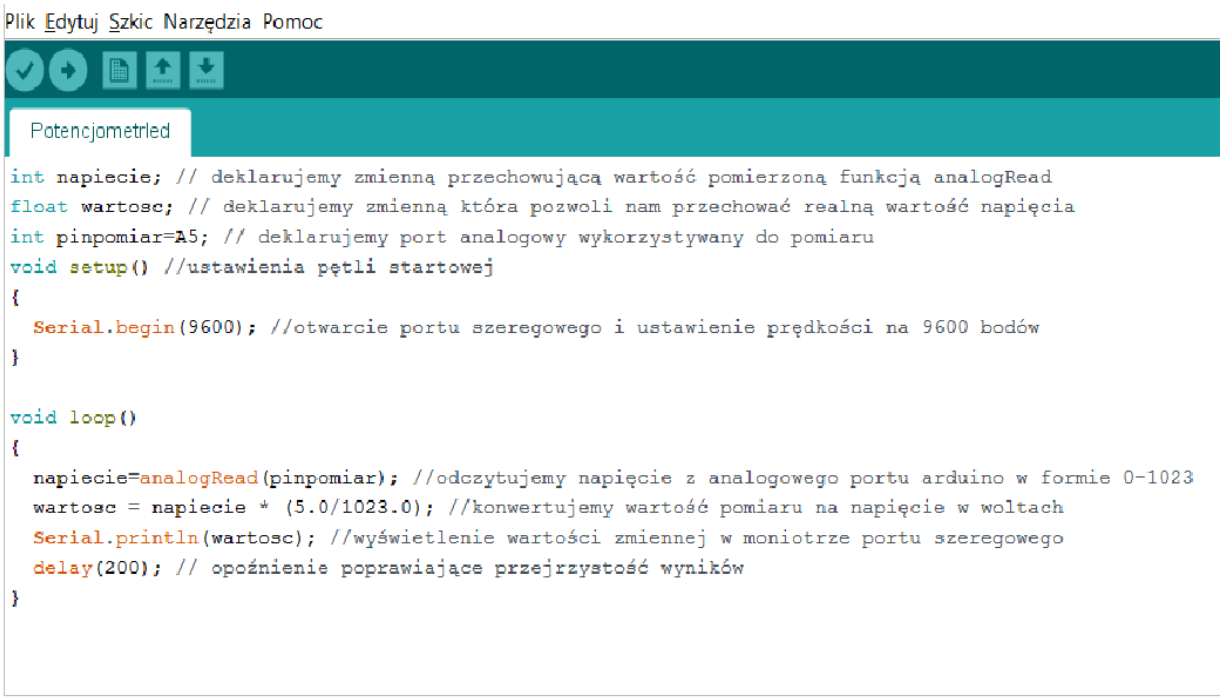
Uczniowie uruchamiają program Arduino IDE i sprawdzają poprawność połączenia. Ponieważ programy Arduino mają stałą, dobrze określoną strukturę składającą się z dwóch funkcji setup() i loop(), uczniowie powinni uczyć się ich, wpisując je bezpośrednio w Arduino IDE na podstawie objaśnień nauczyciela, oraz kilkukrotnie załadowywać kod do modułu. Powinni natrafić na błąd kompilacji programu spowodowany np. przez literówkę lub brak średnika na końcu linii, by zaznajomić się z reakcją oprogramowania i sposobem identyfikacji błędu.

Funkcje które powinny zostać omówione na potrzeby tego modułu to:

- **void setup()** - początkowa konfiguracja – część przygotowująca;
- **void loop()** - główna pętla – wpisany program będzie wykonywany cyklicznie, wciśnięcie przycisku **reset** spowoduje uruchomienie kodu od początku;
- funkcja **pinMode** ustawia kierunek sygnału na pinie modułu (INPUT, OUTPUT);
- funkcja **digitalWrite** ustawia cyfrowy stan sygnału na pinie modułu (HIGH, LOW);
- funkcja **delay** odlicza ustaloną w milisekundach długość czasu (1000 ms = 1 s);
- funkcja **if/else** wykonuje kod w niej zawarty, zostanie spełniony pewien określony w niej warunek, w przeciwnym wypadku realizowany jest kod zawarty w **else**;
- funkcja **analogRead(zmienna)** która zwraca wartość wejścia analogowego (należy pamiętać o nazewnictwie np. A5);
- funkcja **Serial.println(np. zmienna)** pozwala na wysłanie informacji do komputera, który wyświetla ją przy pomocy Monitora portu szeregowego dostępnego w zakładce narzędzia.

Wyjaśniamy uczniom, że porty analogowe są tak skonstruowane, że widzą napięcie jako część z 1024 możliwych punktów. Zeru odpowiada napięcie 0 V, 1023 odpowiada napięcie 5V. Nie można podłączyć do tego portu większego napięcia.

Pytamy uczniów, czy mają pomysł na kod realizujący ten pomiar z wykorzystaniem funkcji **Serial.println**. Po krótkiej dyskusji prezentujemy kod i omawiamy go.



Uczniowie wprowadzają kod do programu, a następnie wgrywają go na Arduino. Jeśli nie popełnili żadnego błędu, program potwierdzi udane wgranie.

Uczniowie uruchamiają Monitor portu szeregowego i przy pomocy potencjometru sprawdzają, jaki wpływ na wynik ma regulacja elementu (regulacja poprzez przekręcanie).

Pytamy uczniów o wnioski dotyczące wyników oraz prosimy, aby uczniowie wspólnie podjęli decyzję dotyczącą progów, które przyjmą jako: **odpowiednie napięcie** (można przyjąć od 1V), **duże napięcie** (można przyjąć 3V) i **zbyt duże napięcie** (można przyjąć 4V). Należy tu wspomnieć, że nie są to niebezpieczne zajęcia, a nazwy są wykorzystane w odniesieniu do dostępnego zakresu napięć.

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

Przypomnienie materiału, odtworzenie układu i programu – 10 minut

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

Tworzenie systemu ostrzegania o wysokim napięciu – 30 minut

Omawiamy sposób realizacji zadania. Do wykorzystania są 3 kolorowe diody oraz potencjometr. Uczniowie powinni zauważyć, że mogą wykorzystać poprzedni program, dodając do niego świetlną sygnalizację. Proponujemy rozwiązanie:

- zapalona dioda zielona – odpowiednie napięcie,
- zapalona dioda żółta – ostrzeżenie o wysokim napięciu,
- migająca dioda czerwona – ostrzeżenie o zbyt dużym napięciu.

Można to zrealizować za pomocą warunków **if/else**.

Rysujemy na tablicy algorytm postępowania. Może on mieć postać: „jeśli napięcie (próg ustalony wcześniej) to zapala się dioda żółta, jeśli nie, dioda pozostaje zgaszona". Analogicznie rozwiązanie mamy dla pozostałych diód.

Miganie czerwonej diody możemy zrealizować z pomocą funkcji **delay()**.

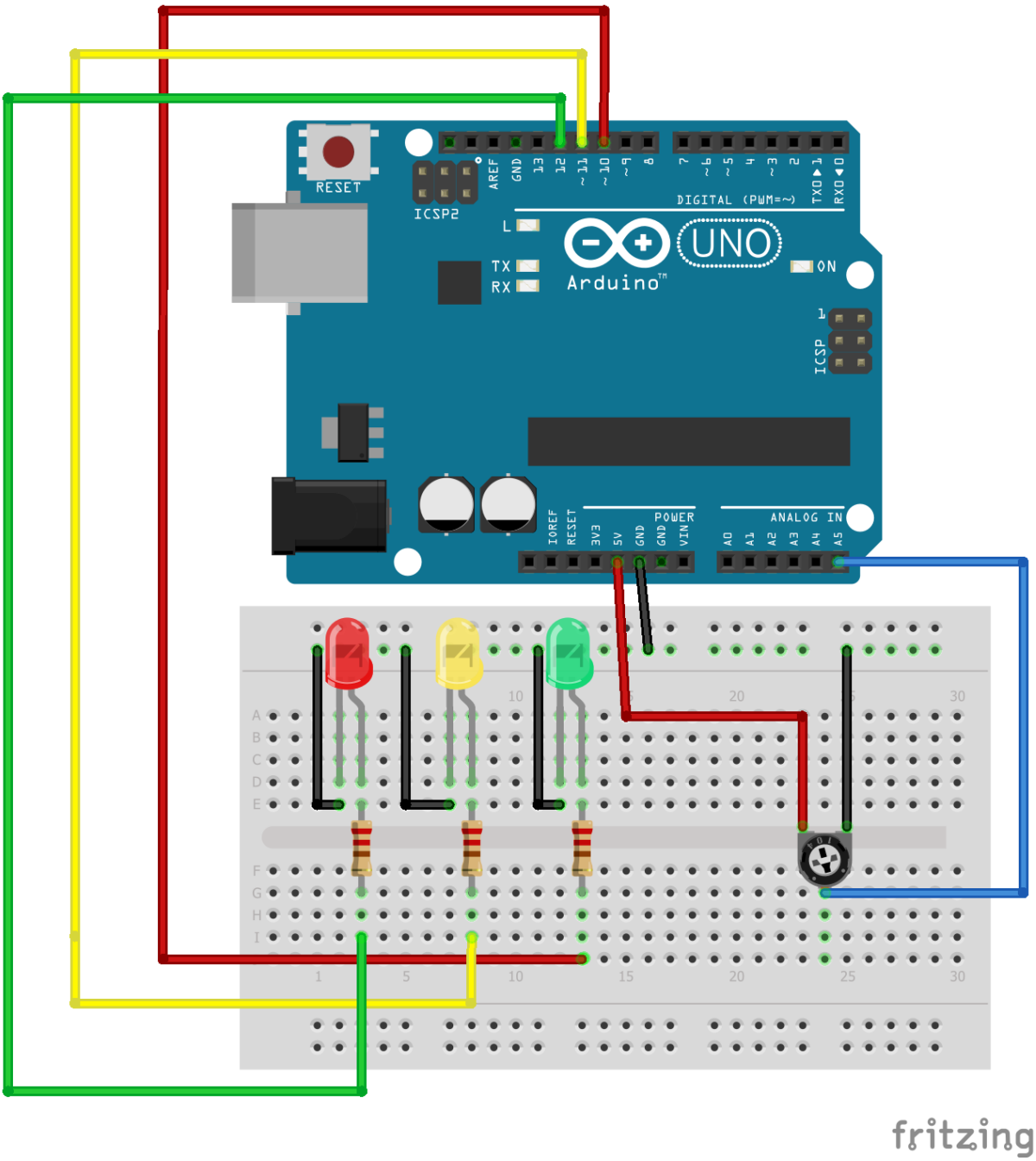
„Jeśli zbyt duże napięcie, zapal diodę, poczekaj, zgaś diodę".

Można wspomnieć, że dla ułatwienia uczniowie mogą powielić funkcje warunkowe i tylko zmienić ich warunki.

Uczniowie realizują samodzielnie zadanie. Rozpoczynają od wykonania wymaganych połączeń na płytce. Osoba prowadząca powinnna przypomnieć uczniom o zastosowaniu rezystorów przed diodami LED, aby uniknąć ich uszkodzenia.

Zespoły, które zakończyły zadanie, zgłaszają gotowość, sprawdzamy sposób realizacji i pozwalamy na przejście do pisania kodu programu.

Jeśli jakieś zespoły nie będą potrafiły stworzyć obwodu, pokazujemy im obrazek z gotowym schematem.



Uczniowie, którzy skończą, zgłaszają kod do sprawdzenia. Mogą uruchomić program bez podłączenia płytki, aby uzyskać weryfikacje poprawności programu. W razie ewentualnych błędów mogą spróbować sami je poprawić na podstawie informacji zwróconych przez program.

W przypadku drobnych błędów, które pozwalają na działanie programu, ale nie realizują zadania, można pozwolić na wgranie na płytkę programu. Następnie powinno się poprosić uczniów o wyciągnięcie wniosków i poprawienie kodu.

Zespoły, które zakończą zadanie przed czasem, mogą spróbować zaimplementować do swojego projektu buzzer, aby stworzyć ostrzeżenie dźwiękowe. Jeśli skończą z mniejszym zapasem czasu, można wymieszać ich ze słabiej radzącymi sobie grupami.

```
void setup()
{
  Serial.begin(9600); //otwarcie portu szeregowego i ustawienie prędkości na 9600 bodów
  pinMode(czerwonaPin, OUTPUT); //ustalamy że wybrany port jest portem wyjścia, tzn że informacja będzie z niego wychodzić
  pinMode(zoltaPin, OUTPUT);
  pinMode(zielonaPin, OUTPUT);
}

void loop()
{
  napiecie=analogRead(pinpomiar);  //odczytujemy napięcie z analogowego portu arduino w formie 0-1023
  wartosc = napiecie * (5.0/1023.0);  //konwertujemy wartość pomiaru na napięcie w woltach
  Serial.println(wartosc);  //wyświetlenie wartości zmiennej w moniotrze portu szeregowego
  if (wartosc > 1) // funkcja warunkowa, "Jeśli wartość napięcia jest większa od jednego"
  {
    digitalWrite(zielonaPin,HIGH);} // Jeśli tak, to zapal diodę
  else
  {
    digitalWrite(zielonaPin,LOW); // Jeśli nie, zgaś diodę
    }
  if (wartosc > 3)
  {
    digitalWrite(zoltaPin,HIGH);}
  else
  {
    digitalWrite(zoltaPin,LOW);
    }
  if (wartosc > 4)
  {
  digitalWrite(czerwonaPin,HIGH);
  delay(200); // opóźnienie które pozwala osiągnąć lepszy efekt migania diody
  digitalWrite(czerwonaPin,LOW);
   delay(200); //opóźnienie które pozwala osiągnąć lepszy efekt migania diody
  }
}
```

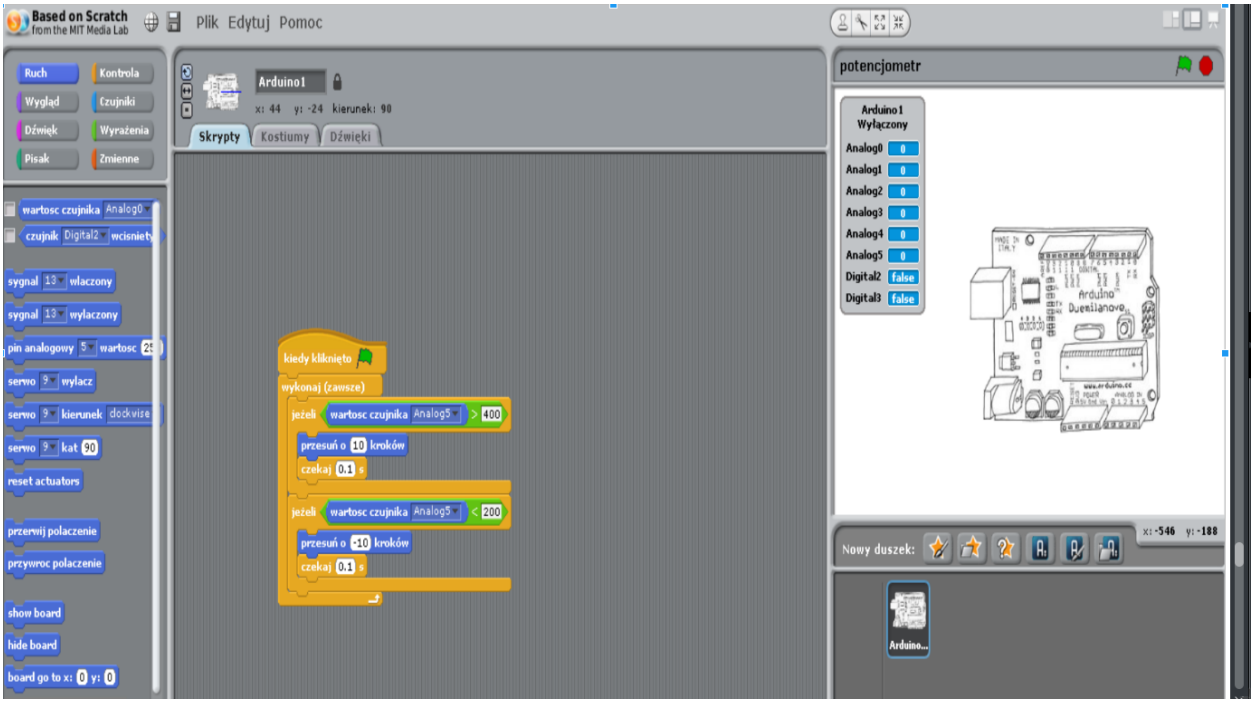
Podsumowanie zadania – 5 minut

Krótko omawiamy napotkane przez uczniów problemy. Podsumowujemy działanie potencjometru i funkcji **Serial.println**. Możemy zadać uczniom pytania o ewentualne modyfikacje i rozszerzenia programu.

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.

Dla młodszych klas w programie S4A można zrealizować prostą grę używającą potencjometru jako kontrolera ruchu „duszka". Dla bardziej zaawansowanych grup można zaproponować budowę identycznego systemu ostrzegania, jak w zadaniu powyżej, ale przy użyciu S4A oraz przygotowanego już obwodu.



ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole buduje układ z wykorzystaniem Arduino, płytki stykowej i potencjometru.

Uczniowie po zajęciach powinni znać zastosowanie funkcji Serial.println oraz budowę funkcji warunkowej „if/else". Powinni także potrafić korzystać z wejść analogowych oraz znać zasadę budowy woltomierza w oparciu o Arduino. Wiedzę można zweryfikować zadając zadanie stworzenia na kartce blokowego algorytmu realizującego funkcję woltomierza.

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

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).