



MoboLab – roboty i tablety w Twojej szkole

Obszar II. „Stwórz własnego robota”

Scenariusze lekcji i zajęć pozalekcyjnych

SCENARIUSZ 24. CZUJNIK TEMPERATURY – TERMOMETR Z PREZENTACJĄ DIODY RGB

scenariusz zajęć pozalekcyjnych

autor: Michał Kłosiński

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

termometr, Arduino, pomiary, jednostki, dioda RGB

KRÓTKI OPIS ZAJĘĆ:

Celem zajęć jest budowa elektronicznego **termometru**, którego funkcje uczniowie mogą samodzielnie modyfikować dzięki zdobytym umiejętnościom. Zajęcia mogą być realizowane w grupach lub samodzielnie, ale zakładają komunikację i współpracę między uczniami.

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”):

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/>),
- płytko Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytko stykowa,
- oporniki 220 omów,
- czujniki temperatury LM35DZ,
- diody RGB,
- przewody połączeniowe,
- 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 budowę i zastosowania elektronicznego termometru – 15 minut

Zapowiadamy, że na tych zajęciach uczniowie i uczennice będą budować elektroniczne termometry zaopatrzone w świecące diody. Opowiadamy o zastosowaniach takiego termometru oraz o tym, czym różni się taki termometr od

gotowych produktów. Podkreślamy, że posiadając odpowiednie umiejętności można stworzyć własne unikalne urządzenie. Jeżeli uczniowie wykażą się wolą dyskusji i/lub będą chcieli podzielić się swoimi pomysłami na wykorzystanie urządzenia, zachęcamy ich do tego. Zadaniem nauczyciela jest moderowanie dyskusji i podsuwanie pomysłów na kreatywne zastosowanie urządzenia.

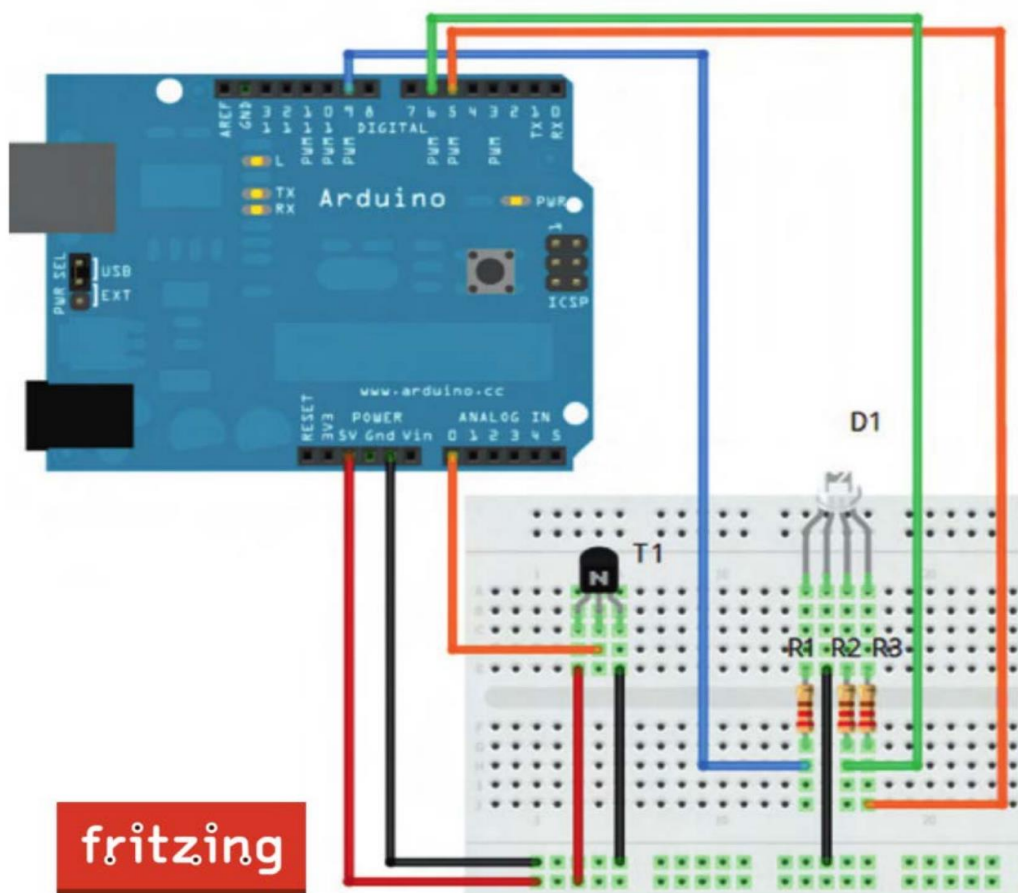
Uwaga! W zależności od liczby zestawów Arduino dzielimy uczniów na takie zespoły, aby każdy mógł z bliska przyglądać się całemu procesowi, a następnie samemu go przeprowadzić. Jeżeli wywiązała się dyskusja na temat zastosowań termometru, może ona być podstawą do podzielenia uczniów na zespoły o podobnych zainteresowaniach i motywacjach.



Montaż układu – 15 minut

Jeśli uczniowie i uczennice posiadają już dostateczną wiedzę na temat Arduino, mogą przygotować propozycję projektu układu samodzielnie lub w swoich zespołach. W tej sytuacji moderujemy prace zespołów i pomagamy uczniom w trudnych momentach. Sugerowanym sposobem działania jest zadawanie pytań

mających na celu naprowadzenie młodzieży na właściwe rozwiązania i uczulenie na potencjalne problemy.



Pokazujemy i omawiamy schemat połączeń. Jeżeli schemat został przygotowany przez uczniów, podsumowujemy go i wyjaśniamy, rozrysowując go na tablicy. Uczniowie i uczennice wspólnie łączą elementy, z których powstanie układ elektronicznego termometru. Monitorujemy postępy, upewniając się, że wszyscy podopieczni nadążają.

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

Przypomnienie materiału, odtworzenie układu – 10 minut

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

Programowanie układu – 20 minut

Omawiamy lub przypominamy interfejs programu Arduino IDE. Upewniamy się, że wszystkie moduły nawiązały komunikację z oprogramowaniem i działają. 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 oraz kilkakrotnie załadowywać kod do modułu. Gdy natrafiają na błąd kompilacji programu, spowodowany np. przez literówkę lub brak średnika na końcu linii, zaznajamiają 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 – wykonuje kod w niej zawarty tylko wtedy, jeśli zostanie spełniony pewien określony w niej warunek.

Jest to dobry moment do upewnienia się, czy wszystkie układy zostały odpowiednio złożone. Jeżeli doświadczenie pozwala nauczycielowi na poprowadzenie modułu tak, by przy okazji omawiania funkcji przeprowadzić podstawowe testy, należy to zrobić. Możemy zadać uczniom takie pytania, jak:

- *Czy pamiętacie, do których numerków na płytce podłączaliśmy przewody?*
- *Jak byście opisali to w programie na podstawie zadanego przykładu?*
- *Co się stanie, gdy załadujecie kod na moduły klikając w strzałkę?*

Objaśniamy kod i zadajemy pytania, by upewnić się, że grupa rozumie poszczególne linijki i moduły kodu. W miarę potrzeb dokonujemy dodatkowych wyjaśnień. Jest to właściwy moment, by wytypować uczniów i uczennice, którzy mają problem z nadążeniem z materiałem. Jest to moduł pracy samodzielnej, który da nauczycielowi chwilę czasu na pomoc uczniom o największych problemach z przyswojeniem materiału.

Warto również wytypować uczniów, którzy dobrze opanowali temat – przyda się to w przypadku, gdy część grup nie będzie mogła ukończyć swoich modułów, lub pojawią się nieprzewidziane problemy. W tej sytuacji osoby wyprzedzające grupę mają szansę natrafić na problemy odpowiadające ich umiejętnościom, przy czym nauczyciel powinien osądzić czy skala problemów odpowiada umiejętnościom wybranych osób, komunikować się z nimi, regularnie monitorować postępy i być gotowym do podjęcia stosownych działań, jeżeli problem przerośnie ucznia.

Uczniowie wpisują przygotowany wcześniej kod i wgrywają go na Arduino. Kod jest w stanie się kompilować w swojej niepełnej formie. Zachęcamy uczniów do eksperymentowania i testowego ładowania kodu przez zakończeniem jego wpisywania. To pozwoli na:

- *szybkie wykluczenie błędów, np. literówek,*
- *zrozumienie i utrwalenie struktury programu i istniejących w niej powiązań („Czemu się nie kompiluje? Co jeszcze jest potrzebne?”).*

Gdy kod zacznie działać na modułach, uczniowie powinni wymyśleć sposób na przetestowanie dokładności i prędkości działania układu (np. poprzez przyłożenie do niego źródła ciepła o stałej znanej temperaturze). Mogą testować układ w kontakcie z elementami pomieszczenia po odpowiednim zmodyfikowaniu układu (podpięcie sensora przez przewody, pozwalające na jego przemieszczanie względem płytki prototypowej).

```
scenariusz_arduino_dioda | Arduino 2.1.0.5+dfsg2-4
Plik Edycja Szukaj Narzędzia Pomoc
scenariusz_arduino_dioda s

int temperatura; //deklarujemy zmienne, mówimy arduino iż będziemy używać zmiennych, czyli różnych wartości
int odczyt; //deklarujemy kolejną zmienną w której będziemy przechowywać aktualną wartość napięcia wychodzącą z czujnika które przerabiamy aby wynik był w stopniach celciusza
int tempPin = A0; //Pin do który łączymy z nóżką z czujnika temperatury.
int czerwonaPin = 5; // Pin który łączymy z nóżką diody RGB, możemy sprawdzić wcześniej, która nóżka odpowiada za jaki kolor}
int zielonaPin = 6;
int niebieskaPin = 9;

void setup() // ustawienia pętli startowej
{
  Serial.begin(9600); //otwarcie portu i ustawienie prędkości na 9600 bodów
  pinMode(czerwonaPin, OUTPUT); //ustawiamy pinout diody, informujemy tutaj, że dioda jest urządzeniem wyjścia, w sensie iż informacja ( napięcie) wychodzi z arduino
  pinMode(zielonaPin, OUTPUT);
  pinMode(niebieskaPin, OUTPUT);
}
void loop()
{
  odczyt = analogRead(tempPin); // odczytujemy napięcie które dociera z czujnika do pinu analogowego w arduino
  temperatura = odczyt / 9.31; // dzielimy przez 9.31 wg dokumentacji czujnika
  Serial.println(temperatura); // serial print nl (nowa linia)to funkcja która w serial monitorze wyświetla zmienną z nawiasu, w tym przypadku (temperaturę)
  delay(200); // czekamy 200 milisekund na kolejny przebieg pętli // sprawdzamy czy warunek jest spełniony
  if (temperatura <25) { // jeśli temperatura jest mniejsza niż 25 stopni
    digitalWrite(zielonaPin, HIGH); // to zapala się zielona dioda
    delay(400); // czekamy 400 milisekund
    digitalWrite(zielonaPin, LOW); // funkcja digitalWrite na pinie zielonaPin ustawia wartość 0 i napięcie przestaje płynąć i dioda gaśnie.
  }
  if (temperatura <=25 && temperatura >28) { // sprawdzamy czy warunek jest spełniony
    digitalWrite(czerwonaPin, HIGH); // jeśli tak to dioda czerwona zapala się
    delay(400);
    digitalWrite(czerwonaPin, LOW); // dioda gaśnie
  }
  if (zielonaPin >=28) { // sprawdzamy czy warunek jest spełniony
    digitalWrite(niebieskaPin, HIGH); // jeśli tak to dioda niebieska zapala się
    delay(400);
    digitalWrite(niebieskaPin, LOW); // dioda gaśnie i pętla uruchamia się kolejny raz
  }
}
```

Uwaga! W przypadku modułów o nieprawidłowym działaniu, konstrukcji lub oprogramowaniu mamy kilka możliwości:

- wymieszanie pojedynczych osób z grup, u których moduły działają, do grup z modułami niedziałającymi, w nadziei że znajdą problemy,
- ponowne podłączenie i oprogramowanie płytki,
- pomoc osoby prowadzącej,
- w przypadku problemu, którego nie można rozwiązać, wymieszanie grup z wadliwymi modułami z grupami gdzie moduły działają – w tej sytuacji moduły niedziałające pozostają dostępne dla uczniów chętnych, chcących się wykazać umiejętnościami rozwiązywania trudnych problemów.

Zadania dodatkowe - modyfikowanie układu i oprogramowania

Zadanie dodatkowe nr 1

W jaki sposób można zmniejszyć lub zwiększyć temperaturę, aby sprawdzić, czy termometr działa poprawnie?

Zadanie dodatkowe nr 2

Uczniowie mogą zmodyfikować kod i układ połączeń, zamieniając diodę RGB na diodę LED.

Zadanie dodatkowe nr 3

Uczniowie mogą zmodyfikować kod tak, aby na ekranie komputera pojawiły się stopnie Celsjusza lub Fahrenheita.

Zadanie dodatkowe nr 4

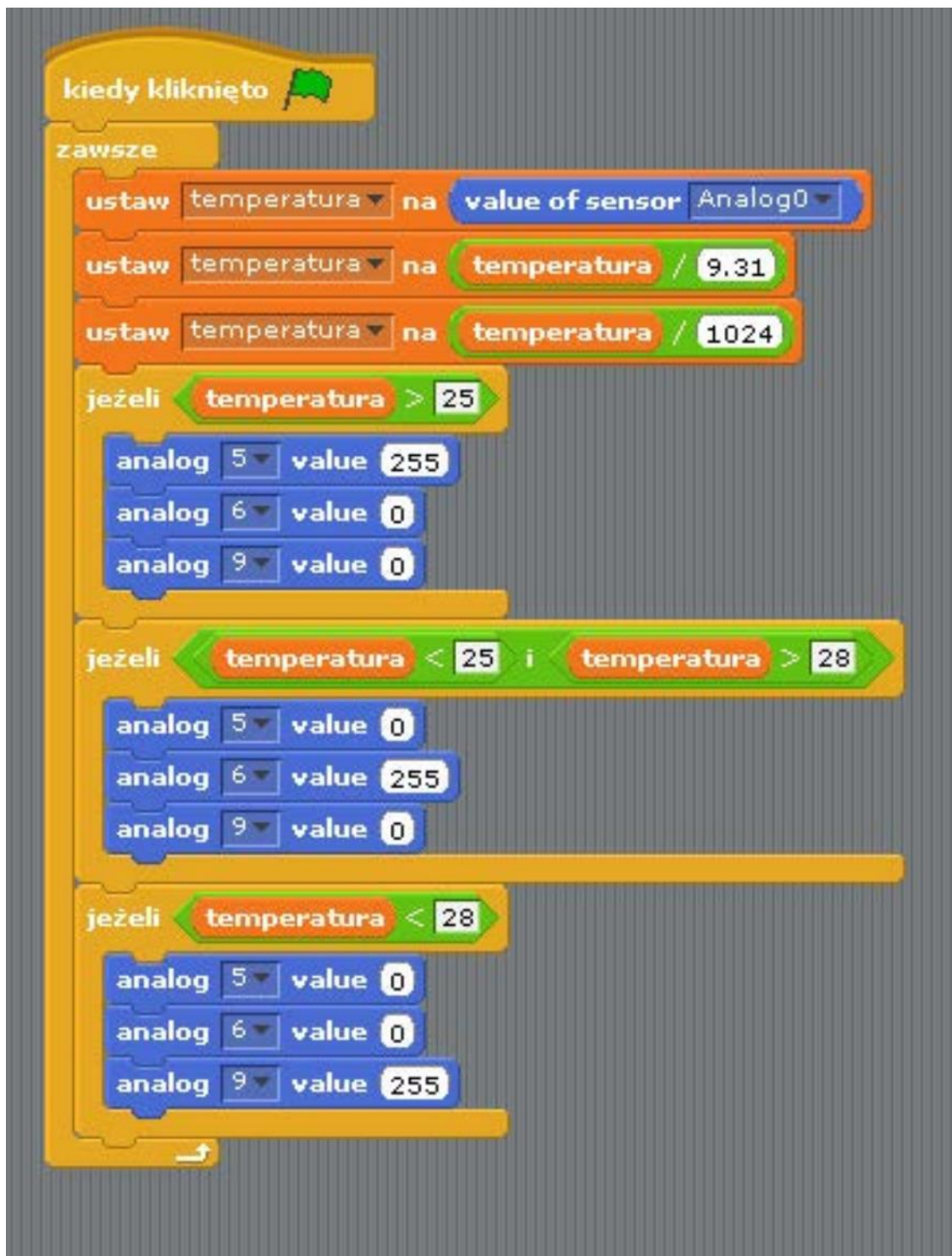
Uczniowie mogą zmodyfikować układ tak, aby pokazać wysokość temperatury za pomocą częstotliwości migającej diody, lub zmiany koloru diody RGB.

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 klas młodszych zajęcia można przeprowadzić za pomocą Arduino4Scratch (S4A). W tym celu, zamiast kodu do programowania zostaną użyte puzzle z instrukcją. Nauczyciel tłumaczy, w jaki sposób działają zmienne oraz w jaki sposób działają warunki.

Poniższy schemat pokazuje rozwiązanie kodu, który działa przy czujniku temperatury LM35DZ:



ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

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

LM35DZ, diody LED i oporników. Zadanie polega na zaprogramowaniu układu i wyjaśnieniu zasady jego działania.

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>

Informacja o diodach LED:

https://pl.wikipedia.org/wiki/Dioda_elektroluminescencyjna

Jak działa płytka stykowa (prototypowa):

https://pl.wikipedia.org/wiki/P%C5%82ytka_prototypowa

Film twórcy Arduino:

www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination

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 ponadgimnazjalnych 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/).