Konstruktorzy gier

Autorzy: Grzegorz Zawistowski, Maciej Wojnicki

Lekcja 4:
Poznajemy funkcję READ sterownika i moduły elektroniczne

W pierwszej części lekcji uczniowie nauczą się deklarować zmienne. Następnie poznają nowe funkcje sterownika LOFI.

Cele lekcji:
Uczeń porafi:
- wyjaśnić podstawowe zasady przy pisaniu kodu w Arduino IDE,
- analizować szkice,
- dokonywać poprawek, zapisywać, weryfikować i wgrywać na płytkę,
- deklarować zmienną liczbową i przypisać jej wartość początkową,
- modyfikować istniejące szkice zmieniając wartości liczbowe na zmienne,
- posługiwać się funkcją write() i read(),
- przypisać wartość z funkcji read(); zmiennej i wykorzystać ją jako wartość funkcji write, np. sterować jasnością diody za pomocą potencjometru.

Materiały pomocnicze:
- zestaw LOFI Robot CODEBOX
- komputery stacjonarne lub przenośne z zainstalowanym Arduino IDE
- komputer nauczyciela z zainstalowanym Arduino IDE, projektor, tablica projekcyjna

Pojęcia kluczowe:
→ Arduino IDE → szkic/program → otwórz, zapisz, zweryfikuj, wgraj → funkcje (setup, loop, write, delay, read) → zmienna → sterownik Arduino → adapter LOFI Brain → dioda → potencjometr

Czas realizacji: 45 min.

Metody pracy:
- wykład problemowy,
- dyskusja dydaktyczna związana z wykładem,
- pokaz,
- ćwiczenia laboratoryjne.

Treści programowe:
Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy VII-VIII, informatyka:
I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:
1) formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) i wyróżnia kroki w algorytmicznym rozwiązywaniu problemów. Stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków;
2) stosuje przy rozwiązywaniu problemów podstawowe algorytmy:
a) na liczbach naturalnych: bada podzielność liczb, wyodrębnia cyfry danej liczby, przedstawia działanie algorytmu Euklidesa w obu wersjach iteracyjnych (z odejmowaniem i z resztą z dzielenia),
4) rozwija znajomość algorytmów i wykonuje eksperymenty z algorytmami, korzystając z pomocy dydaktycznych lub dostępnego oprogramowania do demonstracji działania algorytmów;
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
1) projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice.
2) projektuje, tworzy i testuje oprogramowanie sterujące robotem lub innym obiektem na ekranie lub w rzeczywistości;
5) wyszukuje w sieci informacje potrzebne do realizacji wykonywanego zadania,stosując złożone postaci zapytań i korzysta z zaawansowanych możliwości wyszukiwarek.
III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi. Uczeń:
3) poprawnie posługuje się terminologią związaną z informatyką i technologią.
IV. Rozwijanie kompetencji społecznych. Uczeń:
1) bierze udział w różnych formach współpracy, jak: programowanie w parach lub w zespole, realizacja projektów, uczestnictwo w zorganizowanej grupie uczących się, projektuje, tworzy i prezentuje efekty wspólnej pracy;

Wprowadzenie w tematykę i integracja grupy (5 min.)

Pytamy uczniów, co robiliśmy podczas ostatniej lekcji?

- umiemy uruchomić Arduino IDE,
- znamy składnię kodu w środowisku Arduino IDE,
- wykorzystujemy funkcję write z biblioteki LOFI,
- piszemy proste programy sterujące miganiem diod,
- umiemy wgrywać je na płytkę Arduino UNO,
- podłączamy do płytki urządzenia wyjścia.

Dziś będziemy deklarować zmienne, poznamy kolejne funkcje z biblioteki LOFI oraz nauczymy się definiować własne funkcje. Będziemy pisać proste programy, wgrywać ja na płytkę Arduino i testować ich działanie, podłączając różne urządzenia wejścia i wyjścia.

Część zasadnicza (35 min.)

Przedstawiciel każdej grupy uczniów bierze od nauczyciela przypisany danej grupie zestaw. Uczniowie siadają przy komputerach. Prosimy uczniów, aby włączyli komputery, zalogowali się i uruchomili Arduino IDE. Prosimy, aby otworzyli szkic “zadanie_3” z poprzedniej lekcji, podłączyli płytkę Arduino do komputera oraz dwie diody do OUTPUT1 i OUTPUT2. Prosimy, aby wgrali program na płytkę.

Zadajemy uczniom pytania problemowe:

- Co trzeba zrobić, aby zmienić prędkość migania diod?

Odp: Zmienić wartość liczbową przy poleceniach **delay.**

- W ilu miejscach trzeba zmienić tę wartość?

Odp: W kilku lub kilkunastu, to zależy od stopnia skomplikowania programu.

- Jak myślicie, czy jest sposób na to, aby w szybszy sposób zmieniać prędkość migania diod?

Odp: Jest.

Wyjaśniamy:

- Zamiast zmieniać wartość liczbową w kilkunastu miejscach, możemy na początku programu zdefiniować własną zmienną.
- Sami możemy wymyślić dowolną nazwę tej zmiennej.
- Zmiennej możemy przypisać dowolną wartość liczbową.
- Następnie jako parametr w poleceniu **delay**(TU) zamiast liczby wpisujemy nazwę naszej zmiennej.
- Dzięki temu, jeżeli zechcemy zmienić prędkość migania diod, zamiast zmieniać wartość w kilkunastu miejscach, zmieniamy ją tylko raz, przy definiowaniu zmiennej.
- Poleceniem, które służy do deklarowania nowej zmiennej, jest **int**.
- Po nim: spacja i dowolna nazwa zmiennej (bez polskich znaków).
- Tak zdefiniowanej zmiennej możemy przypisać wartość liczbową:

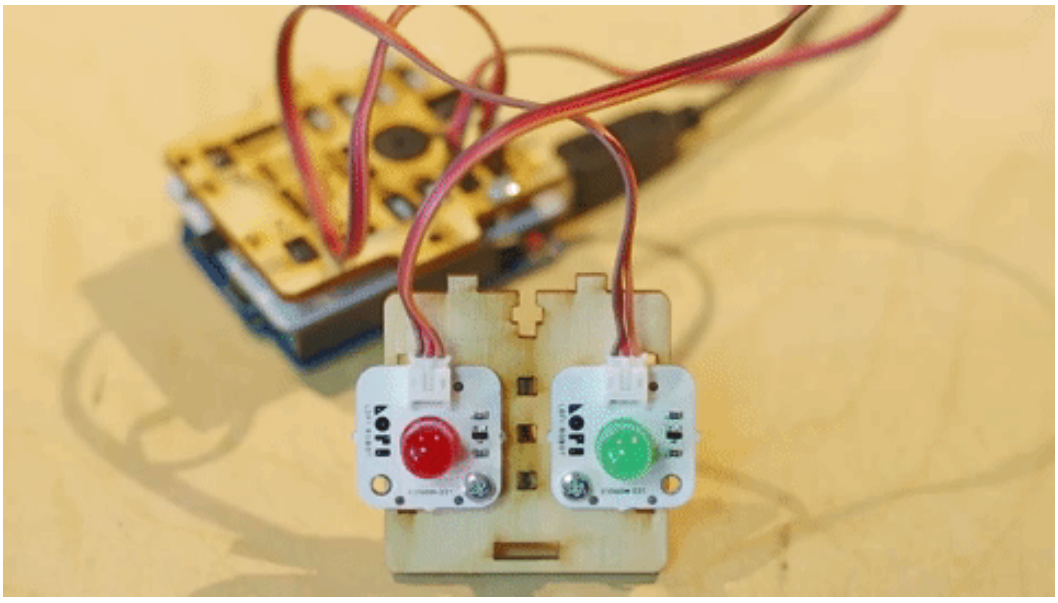
int szybkosc = 200;

- Następnie w treści programu wszędzie tam, gdzie była wartość np. 200, wstawiany nazwę zmiennej np. **szybkosc**.

Przyjrzyjmy się przykładowi z Zadania 3, do którego wprowadzono zmienną **szybkosc**:

```
#include <LOFI.h>
LOFI robot;
//definicja nowej zmiennej szybkosc i jednoczesne przypisanie jej wartości 200
int szybkosc = 200;
void setup() {
}

void loop() {
robot.write(OUTPUT1, 100);
// w poleceniu delay jako parametr zamiast dotychczasowej wartości 200,
// wpisujemy nazwę zmiennej - szybkosc
delay(szybkosc);
robot.write(OUTPUT1, 0);
delay(szybkosc); //tu również zamiast wartości 200 wstawiamy nazwę szybkosc
robot.write(OUTPUT2, 100);
delay(szybkosc); //tu również zamiast wartości 200 wstawiamy nazwę szybkosc
robot.write(OUTPUT2, 0);
delay(szybkosc); //tu również zamiast wartości 200 wstawiamy nazwę szybkosc
}
```



Zadanie 5

Polecenie:

Zmodyfikuj plik z Zadania 3 tak, jak na powyższym przykładzie. Następnie zapisz jako “zadanie_5”. Wgraj go na płytkę Arduino. Następnie spróbuj zmienić wartości deklarowanej zmiennej i sprawdź, czy rzeczywiście zmiana jednej wartość wpływa na działanie całego programu. Ćwiczenie powtórz przynajmniej 3 razy, za każdym razem wpisując inną wartość i sprawdzając działanie programu.

Zadanie 6

Polecenie:

Zmodyfikuj plik z Zadania 4, deklarując dwie nowe zmienne. Przypisz tym zmiennym dwie różne wartości, a następnie w programie przy każdym poleceniu **delay** zamiast parametru liczbowego użyj odpowiednich nazw zmiennych. Zweryfikuj i wgraj program na płytkę Arduino.

Czas na wykonanie zadania: 10 minut. Jeżeli wykonasz zadanie przed czasem, spróbuj zmieniać wartości deklarowanych zmiennych i sprawdź, jak działa Twój program.

Przykładowe rozwiązanie:

```
#include <LOFI.h>
LOFI robot;
int czasswiecenia = 500; //definicja nowej zmiennej czasswiecenia i przypisanie jej wartości 500
int czasprzerwy = 200;  //definicja nowej zmiennej czasprzerwy i przypisanie jej wartości 200

void setup() {
}

void loop() {
robot.write(OUTPUT1, 100);
delay(czasswiecenia); //użycie nowej zmiennej zamiast dotychczasowej wartości liczbowej
robot.write(OUTPUT1, 0);
delay(czasprzerwy); //użycie nowej zmiennej zamiast dotychczasowej wartości liczbowej
robot.write(OUTPUT1, 100);
delay(czasswiecenia);
robot.write(OUTPUT1, 0);
delay(czasprzerwy);
robot.write(OUTPUT2, 100);
delay(czasswiecenia);
robot.write(OUTPUT2, 0);
delay(czasprzerwy);
robot.write(OUTPUT2, 100);
delay(czasswiecenia);
robot.write(OUTPUT2, 0);
delay(czasprzerwy);
}
```

Znamy już funkcję **write(wyjście, wartość)** oraz umiemy deklarować własne zmienne. Czas poznać kolejne funkcje z biblioteki LOFI.

Funkcja **read(wejście);** służy do odczytywania wartości z wejść. Parametrem funkcji read jest wejście, czyli informacja z którego wejścia (**INPUT1, INPUT2, INPUT3** czy **INPUT4**) ma zostać odczytana wartość. Wartości można odczytywać z urządzeń wejścia takich jak: potencjometr, czujnik natężenia światła czy przycisk, podłączonych do gniazd INPUT1-4.

Zadanie 7

Polecenie:

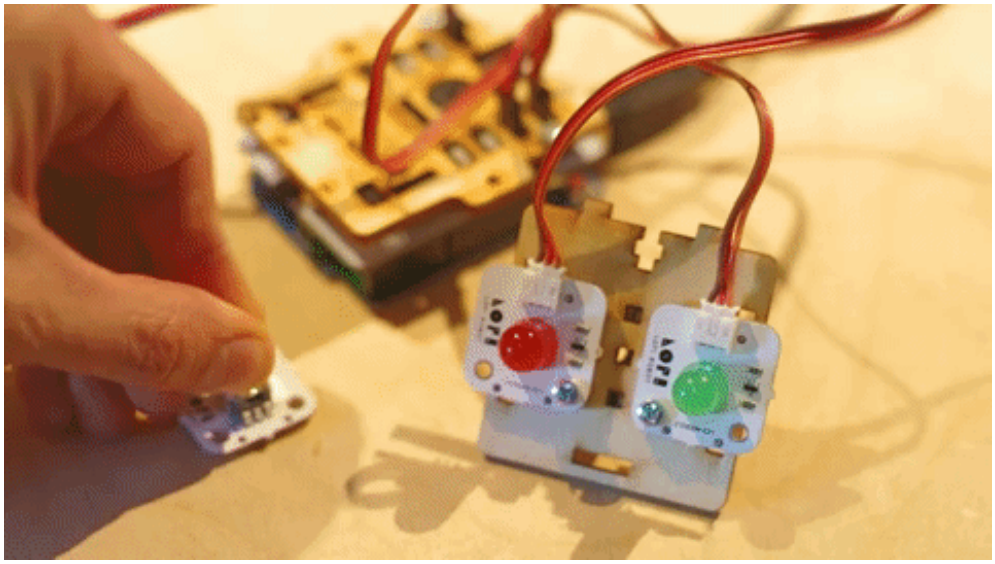
Przepisz poniższy kod. Do wejścia **INPUT1** podłącz potencjometr, a do **OUTPUT1** – diodę. Zapisz plik jako “zadanie_7”, zweryfikuj program i wgraj na płytkę.

Przykładowe rozwiązanie:

```
#include <LOFI.h>
LOFI robot;

void setup() {
}

void loop() {
int potencjometr = robot.read(INPUT1);
// definiujemy nową zmienną potencjometr, której przypisujemy wartość
// z urządzenia podłączonego o INPUT1
// zamiast słowa “potencjometr” możemy użyć dowolnego innego słowa lub litery
// definicja nowej zmiennej jest wewnątrz funkcji loop(), a nie na górze programu,
// aby była odczytywana cały czas, a nie tylko jeden raz
robot.write(OUTPUT1, potencjometr);
// wykorzystujemy znaną nam funkcję write, aby zaświeciła się dioda podłączona do OUTPUT1
// z mocą taką samą, jak wartość odczytana z urządzenia połączonego do INPUT1,
// zapamiętana jako zmienna “potencjometr”
}
```



Podsumowanie i ewaluacja (5 min.)

Prosimy, aby uczniowie ostrożnie spakowali zestawy. Jeden przedstawiciel każdej grupy przynosi zestaw na wyznaczone przez nauczyciela miejsce w klasie.

Zadajemy uczniom pytanie: Czego nauczyliśmy się na dzisiejszej lekcji?

- wykorzystujemy różne funkcje z biblioteki LOFI,
- deklarujemy własne zmienne,
- piszemy proste programy i wgrywamy ja na płytkę Arduino,
- podłączamy do płytki urządzenia wejścia (potencjometr, czujnik odległości, czujnik natężenia światła) i wyjścia (diody, buzzer).

Na zakończenie opowiadamy uczniom, co będziemy robić i czego się nauczymy podczas kolejnej lekcji: będziemy w dalszym ciągu pisać proste programy sterujące pracą diod i buzzera. Nauczymy się również definiować własne funkcje.