

Konstruktorzy gier

Autorzy: Grzegorz Zawistowski, Maciej Wojnicki

Lekcja 8:

Sterowanie ekranem - wyświetlanie liczb

W czasie tej lekcji nauczymy się implementować działanie wyświetlacza LED 8 x 8 pikseli RGB. Poznamy zasadę sterowania nim i nauczymy się wyświetlać liczby.

Cele lekcji:

Uczeń porafi:

- podłączyć wyświetlacz RGB 8x8 do adaptera LOFI Brain,
- wyjaśnić zasadę pisania szkiców służących do wyświetlania grafiki na wyświetlaczu RGB 8x8,
- zaimplementować działanie wyświetlacza przy użyciu funkcji `display.begin()` w pętli `setup()`,
- napisać program wyświetlający na wyświetlaczu dowolny numer z zakresu od 0 do 99 za pomocą funkcji `displayNumber(int NUMER)`, `display.show()` i `displayClear()`,
- napisać program, który będzie wyświetlał liczby "z potencjometru".

Materiały pomocnicze:

- zestaw LOFI Robot CODEBOX Starter z rozszerzeniem CODEBOX Tv,
- komputery stacjonarne lub przenośne z zainstalowanym Arduino IDE
- komputer nauczyciela z zainstalowanym Arduino IDE, projektor, tablica projekcyjna

Pojęcia kluczowe:

→ dioda RGB → wyświetlacz RGB 8x8 → funkcje (`setup`, `loop`, `write`, `delay`, `read`, `buzzer`, `distance`, `servo`, `display.begin`, `display.show`, `displayClear`, `displayNumber`)

Czas realizacji: 45 min.

Metody pracy:

- wykład problemowy,
- pogadanka,
- 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?

- o serwomotorach,
- nauczyliśmy się sterować serwomotorem.

Dzisiaj pierwsza lekcja z wyświetlaczem LED 8x8 pikseli RGB. Nauczymy się implementować jego działanie. Poznamy zasadę sterowania i nauczymy się wyświetlać liczby.

Część zasadnicza (35 min.)

Mówimy uczniom, że do sterowania wyświetlaczem będziemy wykorzystywać kilka gotowych funkcji z biblioteki LOFI:

- funkcja odpowiedzialna za uruchomienie wyświetlacza – rozpoczyna nadawanie sygnału cyfrowego sterującego wyświetlaczem, musi znajdować się zawsze wewnątrz głównej funkcji `setup()`

`display.begin()`

- funkcja odpowiedzialna za wyświetlanie danych na wyświetlaczu – powoduje odświeżenie wyświetlacza i wyświetlenie wszystkich danych zdefiniowanych wcześniej przez funkcje rysujące (o których za chwilę), powinna być użyta tylko raz wewnątrz pętli `loop()`, wówczas wyświetlanie obrazu jest najflakidniejsze:

`display.show()`

- Warto zwrócić uwagę, że funkcja ta będzie włączala konkretne diody i podobnie jak funkcja `write(OUTPUT1, 100)` włączająca diodę, potrzebujemy kolejnej funkcji, która wyłączy wcześniej wyświetlany efekt, zanim zechcemy wyświetlić nowy. W przypadku diody po poleceniu `delay(czas)` używaliśmy ponownie funkcji `write(OUTPUT1, 0)`. W przypadku wyświetlacza potrzebujemy do tego nowej funkcji.
- funkcja odpowiedzialna za wyczyszczenie wyświetlacza powoduje ustawienie wszystkich pikseli wyświetlacza na wartość zero (czyli jako wyłączenie). Powinna być ona użyta tylko raz jako pierwsza funkcja wewnątrz pętli `loop()`, aby niżej zamieszczona funkcja `display.show()`; włączyła odpowiednie diody:

`displayClear()`

- pomiędzy `displayClear()` a `display.show()` będziemy zamieszczać funkcje rysujące.

Zasada działania wyświetlacza jest następująca:

Krok 1 – wyłączamy wszystkie diody – `displayClear()`

Krok 2 – używamy funkcji rysującej

Krok 3 – używając funkcji wyświetlającej włączamy świecenie konkretnych diod – `display.show()`

Rozdajemy zestawy LOFI Robot CODEBOX. Uczniowie siadają przy komputerach. Prosimy aby włączyli komputery, uruchomili Arduino IDE oraz wyjęli z zestawów płytkę Arduino, wyświetlacz i potencjometr:

- podłączamy wyświetlacz LED do **OUTPUT3**
- podłączamy potencjometr do **INPUT2**.

Uwaga: wyświetlacz zawsze musi być podłączony do OUTPUT3

Prezentujemy uczniom podstawowy szkielet szkicu.

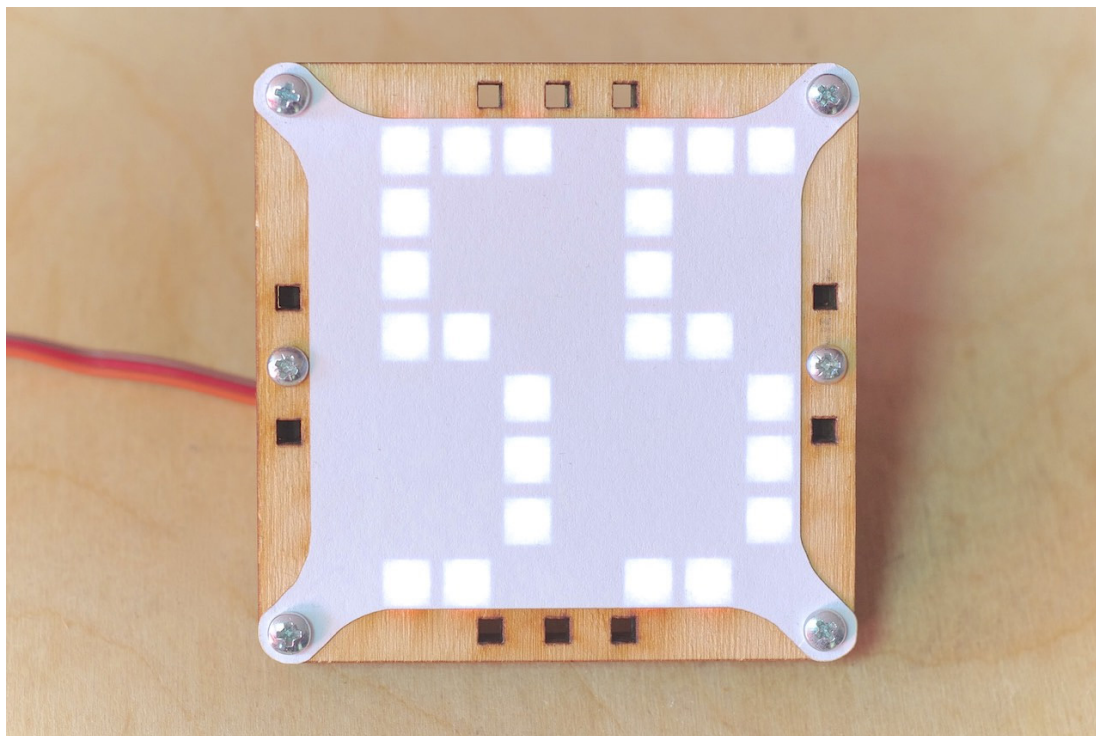
Prosimy aby przepisali go w Arduino IDE:

```
1  #include <LOFI.h>
2  LOFI robot;
3
4  void setup() {
5    robot.displayBegin();
6    robot.displayShow();
7  }
8
9  void loop() {
10   robot.displayClear(); //funkcja wyłączająca poprzedni efekt graficzny
11   // tutaj będziemy wpisywać instrukcje do rysowania
12   robot.displayShow(); // funkcja wywołująca pokazanie danych na wyświetlaczu
13 }
```

Pierwszą funkcją do rysowania, którą poznamy jest funkcja wyświetlająca liczby: **displayNumber(int NUMBER)**; Funkcja ta wyświetla liczby od 0 do 99.

Przykład / Ćwiczenie 1

W powyższym szkicu zadeklaruj nową zmienną liczbę i przypisz jej wartość 55 oraz, dodając funkcję wyświetlającą liczby **displayNumber(liczba)**, wyświetl tę wartość na wyświetlaczu. Szkic zapisz jako **“lofi_wyświetlacz_1”**, zweryfikuj i wgraj na płytce.



Przykładowe rozwiązanie:

```
1  #include <LOFI.h>
2  LOFI robot;
3  int liczba = 55;
4
5  void setup() {
6    robot.displayBegin();
7  }
8
9  void loop() {
10   robot.displayClear();
11   robot.displayNumber(liczba);
12   robot.displayShow();
13 }
```

Ćwiczenie 2

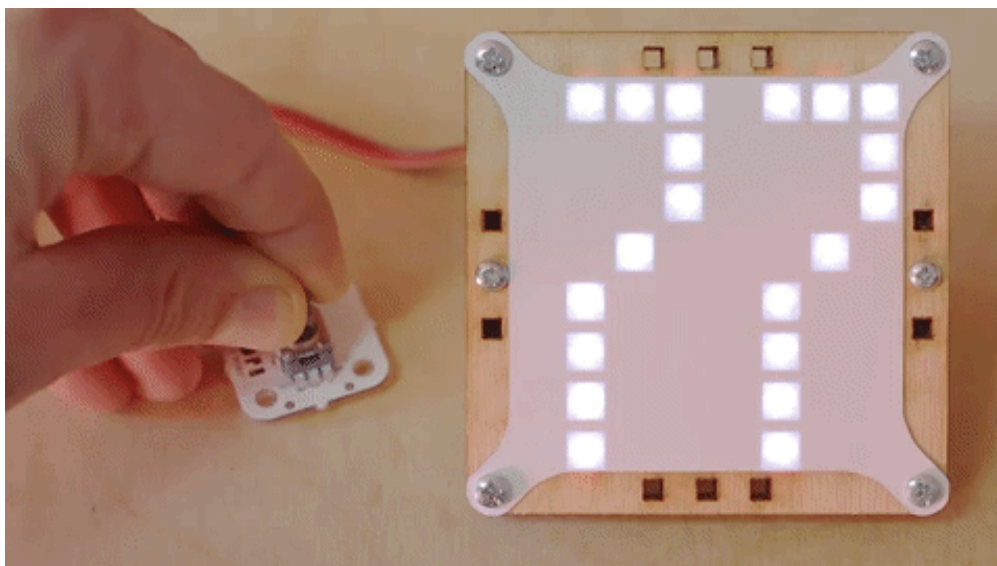
Zmodyfikuj szkic tak, aby na wyświetlaczu została wyświetlona inna liczba. Ćwiczenie powtórz dla 2-3 różnych liczb.

Przykład / Ćwiczenie 3

Napisz program który będzie wyświetlał liczby “z potencjometru” podłączonego do wejścia **INPUT2**. Szkic zapisz jako “**lofi_wyswietlacz_2**”, zweryfikuj i wgraj na płytkę.

Przykładowe rozwiązanie:

```
1  #include <LOFI.h>
2  LOFI robot;
3
4  void setup() {
5  robot.displayBegin();
6  }
7
8  void loop() {
9  robot.displayClear();
10 robot.displayNumber(robot.read(INPUT2));
11 robot.displayShow();
12 }
```



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?

- poznaliśmy funkcję **display.begin()** inicjującą działanie wyświetlacza,
- omówiliśmy zasadę działania wyświetlacza – wiemy do czego służą funkcje **display.show()** oraz **displayClear()**
- znamy funkcję rysującą **displayNumber(liczba)** odpowiedzialną za wyświetlanie liczb od 0 do 99,

Na zakończenie mówimy uczniom, że na kolejnej lekcji nadal będziemy uczyć się korzystać z wyświetlacza. Poznamy m.in. nową funkcję **rectangle** odpowiedzialną za rysowanie prostokąta. Sporo czasu poświęcimy również omówienie systemu RGB, o którym zapewne słyszeliście już nie raz.