

Konstruktorzy gier

Autorzy: Grzegorz Zawistowski, Maciej Wojnicki

Lekcja 9 i 10:

Sterowanie ekranem - lampa z różnymi funkcjami

W czasie tej lekcji uczniowie poznają zasad działania najbardziej popularnego modelu przestrzeni barw: RGB. Poznają także funkcję rysującą prostokąt.

Cele lekcji:

Uczeń porafi:

- Podłączyć wyświetlacz RGB 8x8 do adaptera LOFI Brain (gniazdo OUTPUT3).
- Wyjaśnić zasadę pisania szkiców służących do wyświetlania grafiki na wyświetlaczu RGB 8x8.
- Napisać program wyświetlający na wyświetlaczu dowolny numer z zakresu od 0 do 99 za pomocą funkcji `displayNumber(int NUMER), display.show()` i `displayClear()`
- Wyjaśnić, na czym polega model przestrzeni barw RGB.
- Uzyskać kilka podstawowych barw z modelu RGB.
- Pisać programy wyświetlające prostokąty konkretnego koloru, wielkości i o konkretnym położeniu.
- Napisać program wyświetlający prostokąt, którego wielkość lub położenie może być sterowana za pomocą 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 → RGB (model przestrzeni barw) → wyświetlacz RGB 8x8 → funkcje (`setup`, `loop`, `write`, `delay`, `read`, `buzzer`, `distance`, `servo`, `display.begin`, `display.show`, `displayClear`, `displayNumber`, `rectangle`, `color`, `rainbowColor`)

Czas realizacji: 90 min.

Metody pracy:

- wykład problemowy,
- pogadanka,
- dyskusja dydaktyczna związana z wykładem,
- ćwiczenia laboratoryjne,
- projekt.

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?

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

Dzisiaj poznacie funkcję rysującą prostokąt oraz omówimy model przestrzeni barw RGB.

Część zasadnicza (80 min.)

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

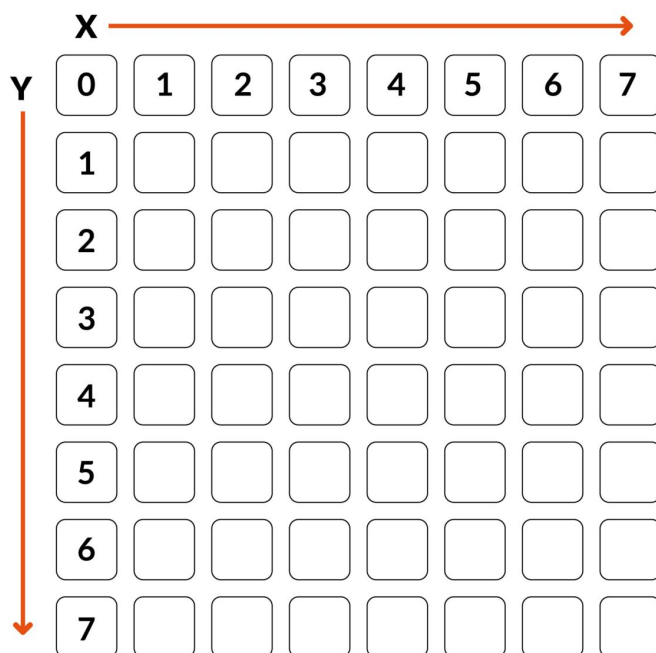
Nowymi funkcjami, jakie dzisiaj poznamy będą:

- funkcja odpowiedzialna za rysowanie prostokąta:
`rectangle(int X, int Y, int SZEROKOŚĆ, int WYSOKOŚĆ);`
- funkcja odpowiedzialna za określenie koloru:
`color(int RED, int GREEN, int BLUE);`
- funkcja do określania koloru pojedynczym parametrem
`rainbowColor(int VALUE);`

Funkcja **`rectangle(int X, int Y, int SZEROKOŚĆ, int WYSOKOŚĆ);`** powoduje wyświetlenie prostokąta o współrzędnych lewego górnego rogu o wartości X i Y oraz określonej szerokości i wysokości.

Pamiętajmy, że nasz wyświetlacz składa się z 64 diod RGB ułożonych w 8 wierszy, czyli jego rozdzielczość wynosi zaledwie 8x8 pikseli. To mało w porównaniu z Full HD gdzie rozdzielczość wynosi 1920x1080 pikseli.

Warto wiedzieć, że współrzędne wyświetlacza liczone są od 0 do 7. Tj. pierwsza od lewej kolumna diod ma wartość X=0, druga X=1 itd. aż do ósmej gdzie X=7. To samo dotyczy wierszy: pierwszy wiersz od góry to Y=0, drugi wiersz licząc od góry to Y=1 itd. Najniżej położony, ósmy rząd, licząc od góry, to Y=7.

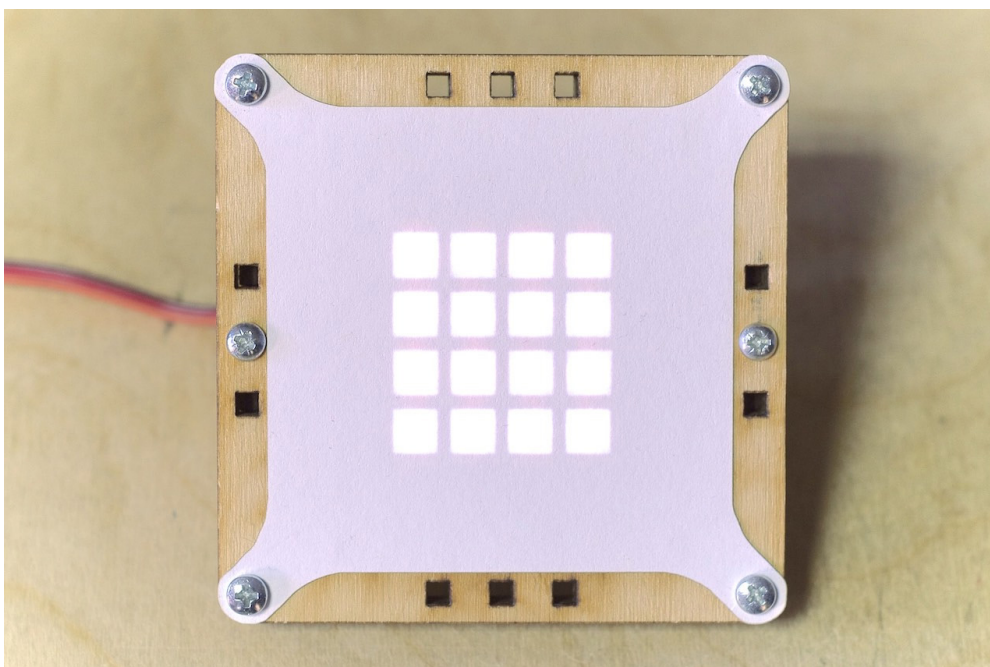


Przykład / Zadanie 1

Zacznijmy od narysowania kwadracika. Niej jego lewy górny róg ma współrzędne X=2, Y=2, a wysokość i szerokość wynosi 4. Szkic zapisz jako "lofi_kwadrat", zweryfikuj program i wgraj na płytkę.

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.rectangle(2,2,4,4);
11 |  robot.displayShow();
12 | }
```



Zadanie 2

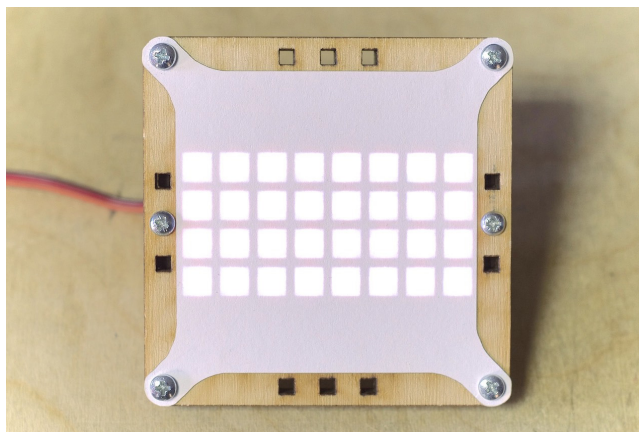
Zmodyfikuj powyższy szkic, aby uzyskać:

1. "wyśrodkowany" na ekranie kwadrat o boku 6x6. Gdzie ma znajdować się jego lewy górny róg?
2. "wyśrodkowany" prostokąt o szerokości 2 i wysokości 8 pikseli.
3. "wyśrodkowany" prostokąt o szerokości 8 i wysokości 4 pikseli.
4. prostokąt wg własnego pomysłu

Każdy szkic przed wgraniem na płytkę Arduino najpierw zapisz pod inną nazwą (np. lofi_rys_2a, lofi_rys_2b, itd), zweryfikuj program i wgraj na płytkę.

Przykładowe rozwiązanie do Zadania 2 c)

```
1 | #include <LOFI.h>
2 | LOFI robot;
3 |
4 | void setup() {
5 |     robot.displayBegin();
6 | }
7 |
8 | void loop() {
9 |     robot.displayClear();
10 |    robot.rectangle(0,2,8,4);
11 |    robot.displayShow();
12 | }
```



Zadanie 3

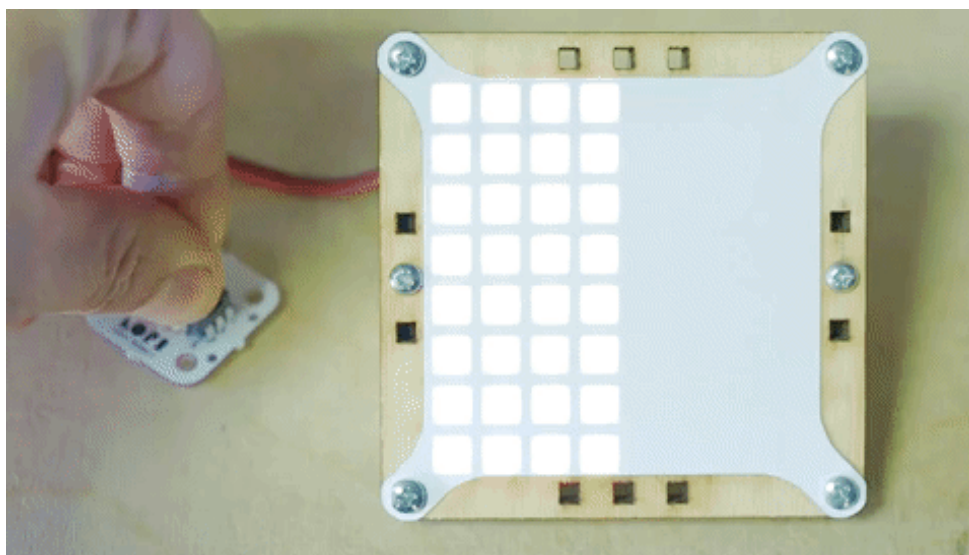
Napisz skrypt umożliwiający sterowanie szerokością prostokąta przy pomocy potencjometru przypiętego do **INPUT2**. Użyj do tego celu funkcji **read(INPUT2)**. Pamiętaj jednak, że wartości jakie zwraca ta funkcja, mieszczą się w przedziale od 0 do 100, podczas gdy dysponujesz matrycą o szerokości do 8 pikseli. Jak więc dostosować tę zmienną?

Podpowiedź: do programu wprowadź zmienną potencjometr której przypiszesz wartość z funkcji **read(INPUT2)**; podzieloną przez 12. Dzięki temu zmienna będzie przyjmowała wartości od 0 do 8, bo $100/12=8,33$.

Szkic zapisz jako **“lofi_prostokat_potencjometr”**, zweryfikuj program i wgraj na płytkę.

Przykładowe rozwiązanie:

```
1 | #include <LOFI.h>
2 | LOFI robot;
3 |
4 | void setup() {
5 |     robot.displayBegin();
6 |     robot.displayShow();
7 | }
8 |
9 | void loop() {
10 |    int potencjometr = robot.read(INPUT2)/12;
11 |    robot.displayClear();
12 |    robot.rectangle(0,0,potencjometr,8);
13 |    robot.displayShow();
14 | }
```

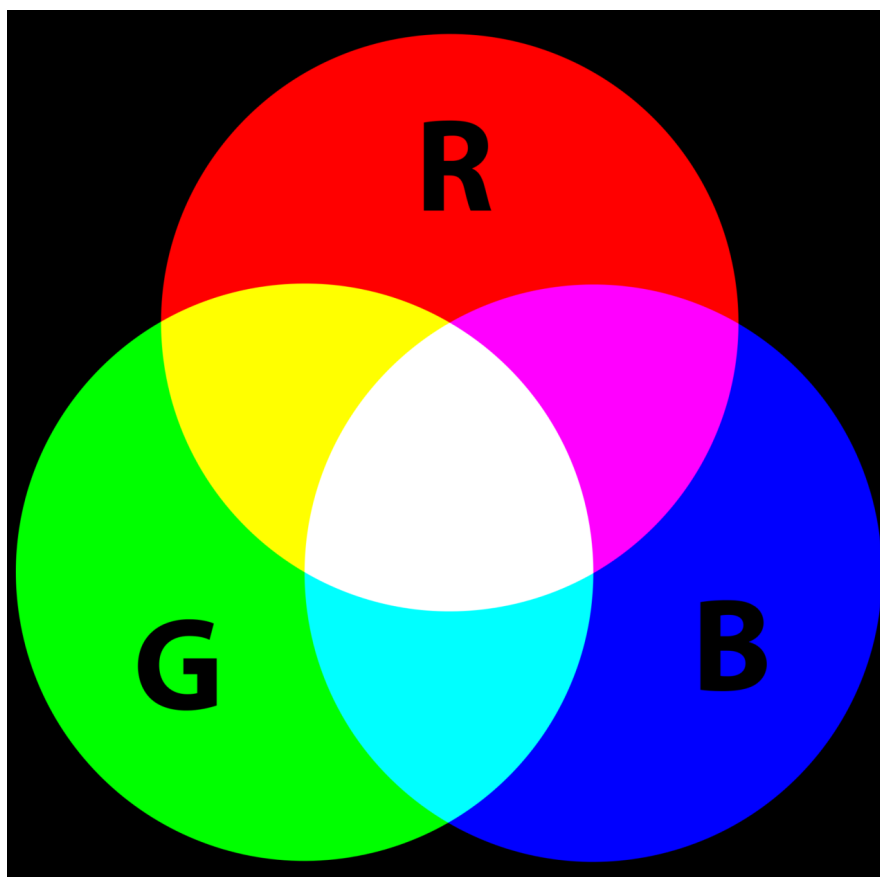


Po wykonaniu tego ćwiczenia mówimy uczniom, że zapewne zauważyli, że wszystkie uzyskiwane do tej pory efekty graficzne wykonywane były za pomocą światła białego.

Teraz nauczymy się zmieniać kolor rysowanych efektów. Najpierw zajmiemy się funkcją odpowiedzialną za określenie koloru w systemie RGB:

color(int RED, int GREEN, int BLUE);

Dyskusja problemowa na temat "Czym jest RGB?"



- RGB to jeden z najpopularniejszych modeli przestrzeni barw.
- Nazwa pochodzi od pierwszych liter angielskich nazw barw: **R – red (czerwonej)**, **G – green (zielonej)** i **B – blue (niebieskiej)**, z których model ten się składa.
- Jest to model wynikający z właściwości naszych oczu, w których wrażenie widzenia dowolnej barwy można wywołać przez zmieszanie w ustalonych proporcjach trzech wiązek światła o barwach: czerwonej, zielonej i niebieskiej.
- Model RGB obecnie wykorzystywany jest w urządzeniach analizujących obraz (aparatach cyfrowych, skanerach) oraz w urządzeniach wyświetlających obraz (telewizorach, monitorach, projektorach).
- Wprowadzenie wartości R=0, G=0, B=0 daje kolor czarny (to tak, jakbyśmy byli zamknięci w ciemnym pomieszczeniu, do dyspozycji mielibyśmy 3 latarki: czerwoną, zieloną i niebieską. Jeśli nie włączymy żadnej latarki w pomieszczeniu będzie panowała ciemność – kolor czarny).
- Wprowadzenie wartości R=255, G=255, B=255 daje kolor biały (3 latarki włączone na 100% świecące w jeden punkt dadzą w tym punkcie kolor biały).
- Ustawienie wartości tylko jednego z kolorów na 255, a dwóch pozostałych na 0, da nam dany kolor (np. gdy włączymy tylko latarkę czerwoną na 100%, a zielona i niebieska będą wyłączone tj. = 0, to w pomieszczeniu będzie kolor czerwony) itd.
- Mieszając w różnych proporcjach te 3 kolory światła, można uzyskać wszystkie kolory, jakimi dysponują monitory i telewizory. W języku C jest to ponad 16 mln kombinacji.

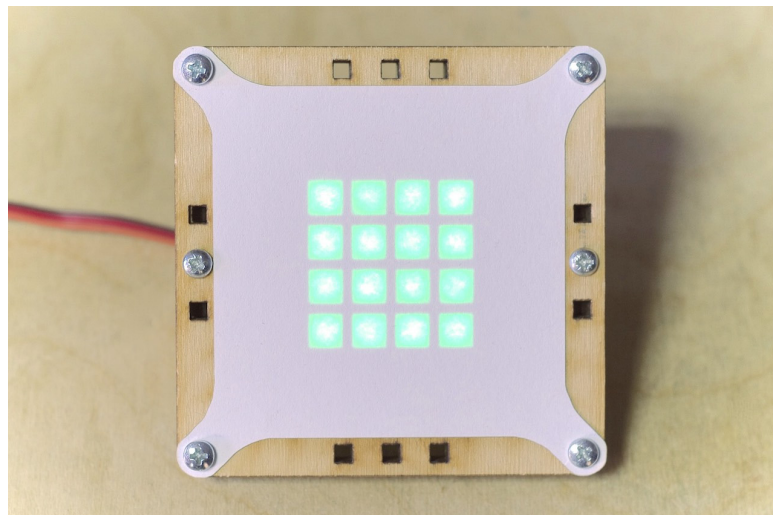
Przykład / Zadanie 4

Narysuj kwadrat, którego lewy górny róg ma współrzędne **X=2, Y=2**, a wysokość i szerokość wynosi 4 – taki sam jak w zadaniu 1, jednak jego kolor ma być zielony. Szkic zapisz jako **“lofi_kwadrat_kolor”**, zweryfikuj program i wgraj na płytkę.

Podpowiedź: funkcję **color(int RED, int GREEN, int BLUE)** należy wstawić przed funkcją rysującą kwadrat.

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.color(0,255,0);
11   robot.rectangle(2,2,4,4);
12   robot.displayShow();
13 }
```



Zadanie 5

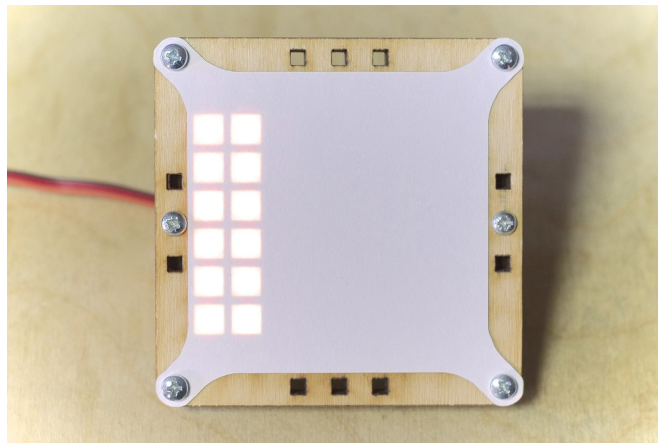
Zmodyfikuj powyższy szkic, aby uzyskać:

- czerwony kwadrat o współrzędnych X=1, Y=3 i wymiarach 5×5,
- niebieski prostokąt o współrzędnych X=2, Y=0, szerokości 3 i wysokości 2,
- żółty prostokąt o współrzędnych X=0, Y=1, szerokości 2 i wysokości 6 (podpowiedź kolor żółty uzyskujemy ze zmieszania dwóch barw: czerwonej i zielonej).

Każdy szkic przed wgraniem na płytkę Arduino najpierw zapisz pod inną nazwą (np. lofi_rys_5a, lofi_rys_5b, itd), zweryfikuj program i wgraj na płytkę.

Przykładowe rozwiązanie do Zadania 5c):

```
1 | #include <LOFI.h>
2 | LOFI robot;
3 |
4 | void setup() {
5 |   robot.displayBegin();
6 | }
7 |
8 | void loop() {
9 |   robot.displayClear();
10 |  robot.color(255,255,0);
11 |  robot.rectangle(0,1,2,6);
12 |  robot.displayShow();
13 | }
```

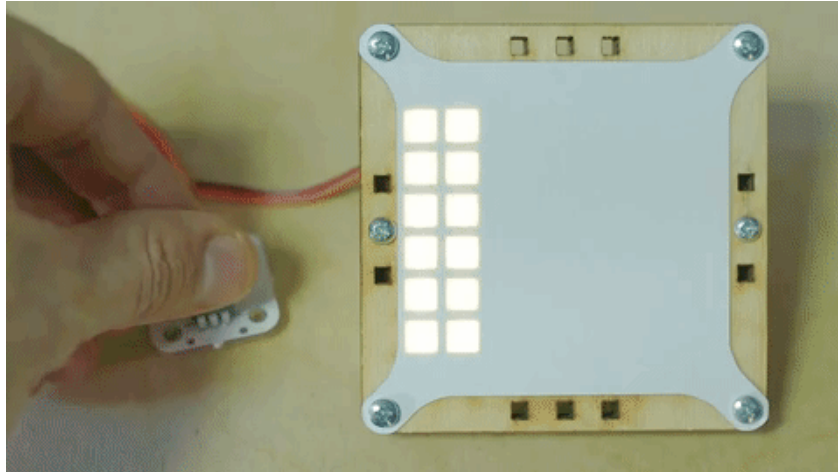


Zadanie 6

Zmodyfikuj szkic z zadania 5c tak, aby położeniem (lewa-prawa) żółtego prostokąta można było sterować za pomocą potencjometru podpiętego pod **INPUT2**.

Przykładowe rozwiązanie:

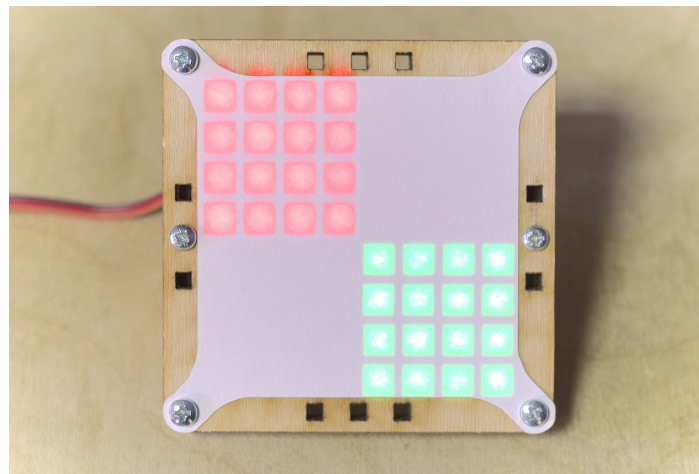
```
1 | #include <LOFI.h>
2 | LOFI robot;
3 |
4 | void setup() {
5 |   robot.displayBegin();
6 | }
7 |
8 |
9 | void loop() {
10 |  int potencjometr = robot.read(INPUT2)/14;
11 |  robot.displayClear();
12 |  robot.color(255,255,0);
13 |  robot.rectangle(potencjometr,1,2,6);
14 |  robot.displayShow();
15 | }
```



Zadanie 6

Napisz program rysujący dwa kwadraty jak na poniższym obrazku.

Szkic zapisz jako "lofi_szachownica", zweryfikuj program 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.color(255, 0, 0);
11 |   robot.rectangle(0,0,4,4);
12 |   robot.color(0,255, 0);
13 |   robot.rectangle(4,4,4,4);
14 |   robot.displayShow();
15 | }
```

Po wykonaniu tego zadania mówimy uczniom, że istnieje jeszcze możliwość określania koloru pojedynczym parametrem. Służy do tego funkcja

rainbowColor(int VALUE);

Funkcja ta umożliwi wybranie dowolnego koloru z pełnej palety barw przy pomocy pojedynczej wartości liczbowej (nie trzeba osobno określać składowych RGB). Funkcja jest szczególnie przydatna gdy chcemy uzyskać na wyświetlaczu efekt płynnie zmieniających się barw np. efekt tęczy.

Przykład:

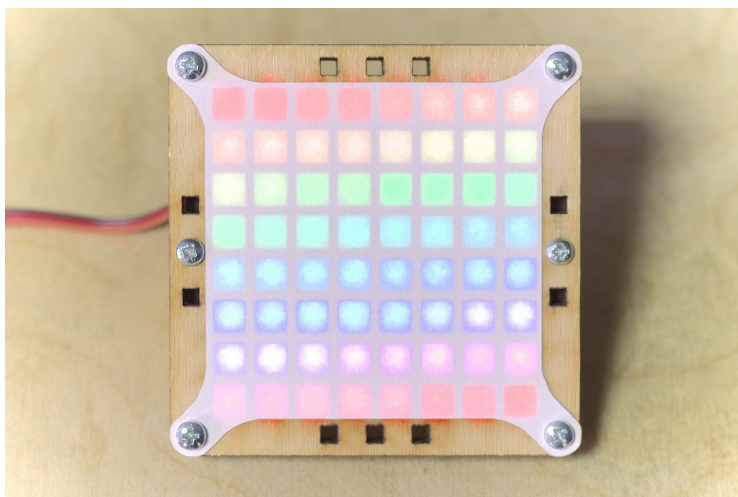
Poniżej zaprezentowano szkic, w którym dla wszystkich 64 diod RGB na wyświetlaczu przypisywany będzie kolejny kolor, gdzie **zmienna i** to kolejny numer diody / piksela na naszym wyświetlaczu.

Za pomocą funkcji **display.setPixelColor(dioda, kolor);** ustawiamy konkretnej diodzie (od 0 do 63, bo pierwsza dioda ma nr 0) konkretny kolor.

W tym przypadku zamiast definicji koloru za pomocą RGB użyto funkcji określającej kolor pojedynczym parametrem **rainbowColor(int VALUE);**

Przykład ten zawiera również iterację – pętlę powtarzającą działanie od $i = 0$, do i mniejszego niż 64, dodając za każdym razem do zmiennej i wartość 1 ($i++$).

```
1 | #include <LOFI.h>
2 | LOFI robot;
3 |
4 | void setup() {
5 |     robot.displayBegin();
6 | }
7 |
8 | void loop() {
9 |     for (int i=0;i<64;i++) {
10 |         robot.display.setPixelColor(i, robot.rainbowColor(i));
11 |     }
12 |     robot.displayShow();
13 | }
```



W zależności od sprawności wykonywanych zadań nauczyciel może zrezygnować z realizacji wybranych zadań. Jeżeli natomiast zostanie trochę wolnego czasu – można pozwolić uczniom na eksperymenty graficzne lub zlecać do uzyskania konkretne efekty graficzne.

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?

- umiemy rysować prostokąty
- wiemy na czym polega model przestrzeni barw RGB,
- posługujemy się funkcją odpowiedzialną za określenie koloru w systemie RGB,
- poznaliśmy funkcję do określania koloru pojedynczym parametrem.

Na zakończenie mówimy uczniom, że na kolejnej lekcji będziemy wykorzystywać zdobytą do tej pory wiedzę i umiejętności przy realizacji projektów – własnych urządzeń i gier. Pierwszym z nich będzie “strzelnica”.