

# Wprowadzenie do programowania urządzeń Arduino

(Arduino dla Informatyków)



Zajęcia pilotażowe z Arduino  
**Systemy operacyjne 2**

Autor: **Piotr Duch, Tomasz Jaworski**  
Instytut Informatyki Stosowanej  
Politechnika Łódzka

Kontakt: [pduch@is.p.lodz.pl](mailto:pduch@is.p.lodz.pl), [tjaworski@iis.p.lodz.pl](mailto:tjaworski@iis.p.lodz.pl)

# Zadania projektowe

## Gra na wyświetlaczu OLED

### Zadanie 9.1

Napisz program, który wyświetli czas jaki minął od uruchomienia programu na wyświetlaczu LCD.

### Zadanie 9.2

Napisz program, który narysuje prostokąt, o wymiarach szerokość - 16 pikseli, wysokość - piksele, na dole wyświetlacza OLED. Prostokąt ma być sterowany za pomocą joysticka (prawy / lewy). Prostokąt nie może nigdy zniknąć z wyświetlacza. Odświeżanie pozycji prostokąta ma następować co 100 ms.

### Zadanie 9.3

Dodaj do swojego programu funkcję losującą pozycję okręgu (numer kolumny z przedziału 0 - 128) i rysującą go w pierwszym wierszu. Okrąg ma co 50 ms spadać o jeden wiersz.

Przydatne funkcje:

- *random(max)* - funkcja generująca liczby pseudolosowe, zwraca liczbę z przedziału  $<0 - max$ ). *max* - górna granica przedziału, z którego będą losowane liczby.
- *random(min, max)* - funkcja generująca liczby pseudolosowe, zwraca liczbę z przedziału  $<min - max$ ). *min* - dolna granic przedziału, z którego będą losowane liczby, *max* - górna granica przedziału, z którego będą losowane liczby.

*randomSeed(seed)* -funkcja inicjalizująca generator liczb pseudolosowych, wywołanie tej funkcji z losową wartością *seed* spowoduje, że za każdym uruchomieniem programu losowana będzie inna liczba. *seed* - jeżeli konieczne jest zagwarantowanie losowania różnych liczb przy każdym kolejnym uruchomieniu programu zaleca się zainicjalizowanie generatora z losową liczbą, na przykład poprzez wykorzystanie funkcji *analogRead()* z numerem dowolnego, niepodłączonego pinu.

Przykład - program losujący liczbę z przedziału  $<0 - 300$ ):

```
long randomNumber;
void setup() {
  long seed = 0;
  for (int i = 0; i < 12; i++)
    seed += analogRead(i);
  randomSeed(seed);
  randomNumber = random(300);
}
```

```
}  
void loop() {}
```

## Zadanie 9.4

Dodaj do programu obsługę punktów, które będą wyświetlane na wyświetlaczu siedmiosegmentowym. Jeżeli spadająca okrąg, po osiągnięciu ostatniego wiersza, jest w tym samym miejscu co prostokąt, to liczba punktów ma się zwiększyć o jeden.

## Zadanie 9.5

Dodaj do programu obsługę brzęczyka za każdym razem, gdy spadający okrąg będzie znajdował się w tym samym miejscu co prostokąt sterowany joystickiem. W takim wypadku program będzie uruchamiał brzęczyk na 50 ms.

## Zadanie 9.6

Dodaj do programu obsługę żyć. Na początku gracz ma posiadać 3 życia a aktualnie posiadana liczba żyć ma być prezentowana za pomocą zapalonych na wyświetlaczu matrycowym diod. Za każdym razem, kiedy spadający okrąg po osiągnięciu ostatniego wiersza nie będzie się znajdował w tym samym miejscu co okrąg sterowany joystickiem, liczba żyć ma się zmniejszyć o jeden.

## Zadanie 9.7

Dodaj do swojej gry zakończenie: kiedy liczba żyć spadnie do 0 to program ma wyczyścić matrycę diod 8x8 oraz wyświetlić napis "Game Over" na wyświetlaczu OLED.

## Zadanie 9.8

Dodaj do gry obsługę poziomów trudności. Zmiana poziomu trudności odbywać się będzie za pomocą potencjometru, a aktualny poziom trudności ma być prezentowany za pomocą zapalonych diod LED (liczba zapalonych diod oznacza aktualny poziom trudności). Poziom trudności wpływa na szerokość prostokąta, każda zmiana zmniejsza lub zwiększa szerokość o 4.