

25.10.2022

## **INŻYNIERIA SYSTEMÓW MIKROINFORMATYCZNYCH**

### **SPRAWOZDANIE 2**

**Temat: Projektowanie i programowanie podstawowych układów  
sterowania i wizualizacji stanów w architekturze Arduino.**

**Lab 4, grupa 2**

Dominik Stycuła

Michał Szczupak

## Zadanie 1:

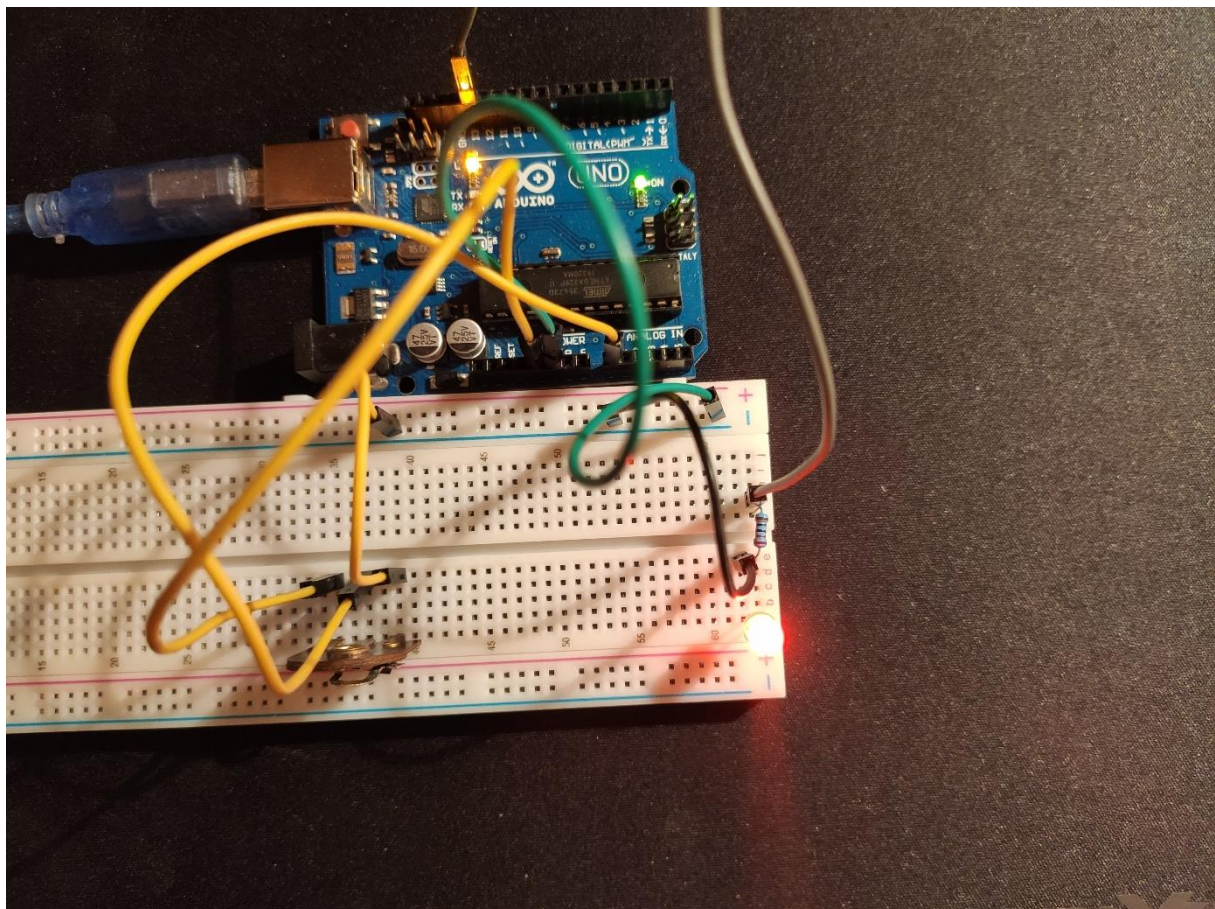
a) Uruchom gotowy szkic i przeanalizuj jego szczegółowe parametry.

```
sketch_oct25a.ino
1 int diodaPin = 13;
2 int potPin = 0;
3 int pot;
4 int przerwa;
5 void setup() {
6   pinMode(diodaPin, OUTPUT);
7 }
8 void loop() {
9   pot = analogRead(potPin);
10
11   przerwa = map(pot, 0, 1023, 50, 3000);
12   digitalWrite(diodaPin, HIGH);
13   delay(przerwa);
14   digitalWrite(diodaPin, LOW);
15   delay(przerwa);
16 }
```

Output

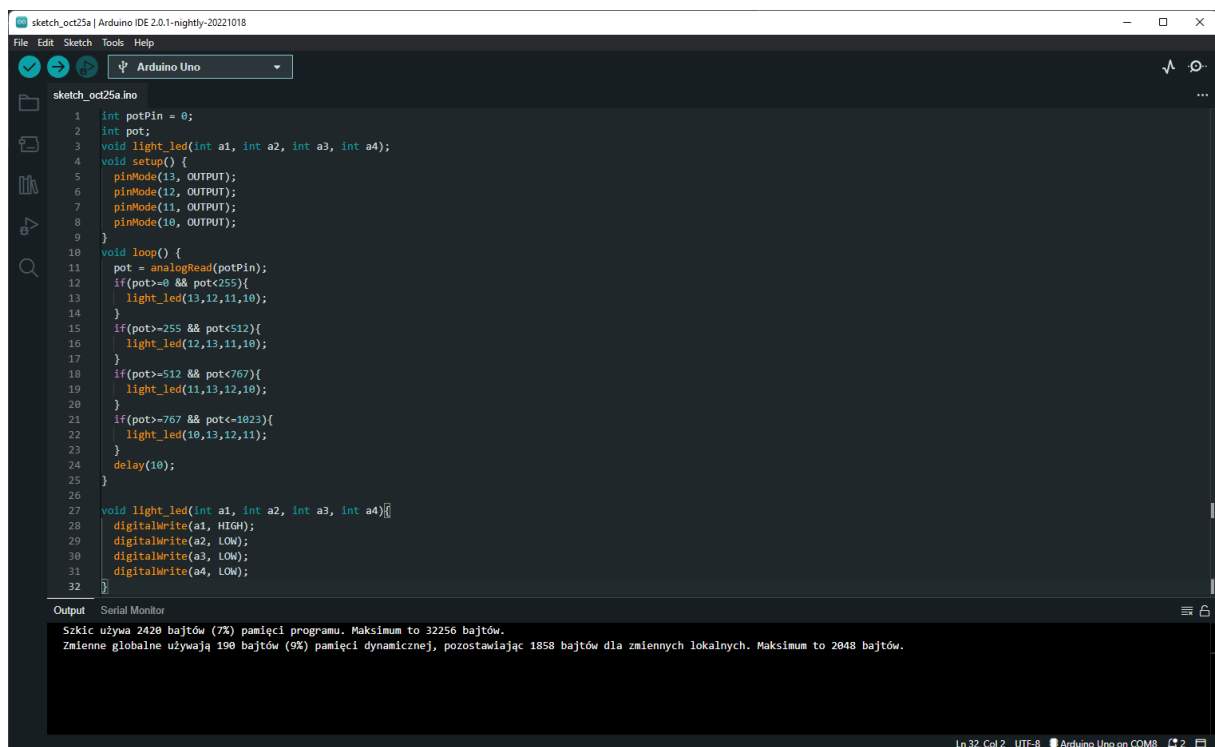
Szkic używa 1224 bajtów (3%) pamięci programu. Maksimum to 32256 bajtów.  
Zmienne globalne używają 11 bajtów (0%) pamięci dynamicznej, pozostawiając 2037 bajtów dla zmiennych lokalnych. Maksimum to 2048 bajtów.

Ln 15, Col 18 UTF-8 Arduino Uno on COM8 2



Algorytm nadaje czas napięcia wyjściowego na pinie nr 13 (zintegrowanego z diodą) w stanie wysokim oraz czas napięcia w stanie niskim zapętlonego w funkcji loop(). Jako czas program pobiera wartość z zakresu 0-1023 odbieranego na pinie wejściowym analogowym o numerze A0, do którego podpięty jest potencjometr. Dzięki temu w zależności jak ustawiony jest potencjometr dioda zaświeca się oraz gaśnie z ustaloną prędkością.

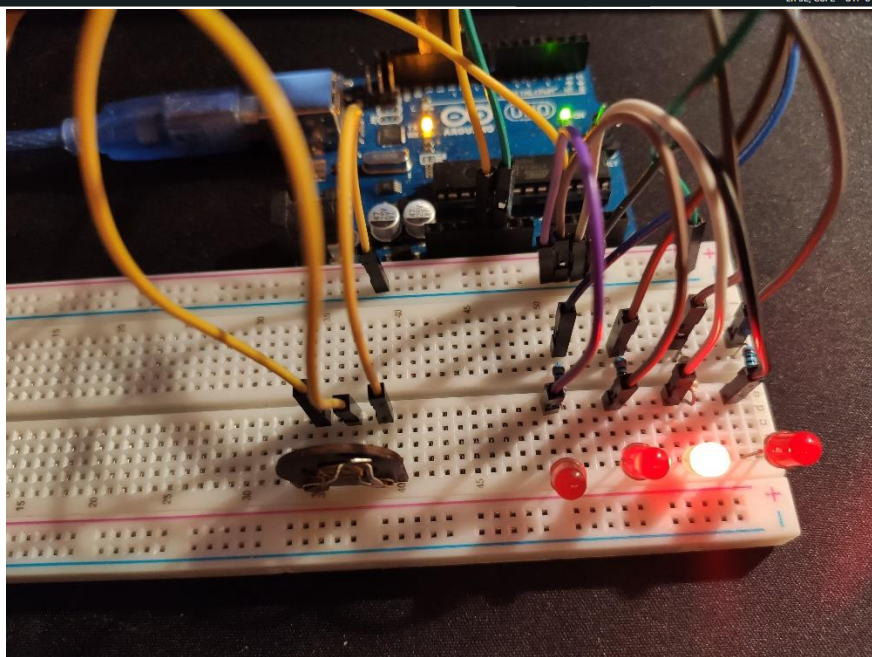
- b) Zaproponuj modyfikację przykładu a) tak, aby można wyznaczyć pozycję potencjometru za pomocą czterech diod. Wykorzystaj cztery diody LED do wizualizacji wartości napięcia wejścia analogowego. Diody powinny sygnalizować cztery stany nastaw potencjometru, 0%...25 %, 26%... 50%, 51%... 75% oraz 76%...100%.



```
sketch_oct25a.ino
1  int potPin = 0;
2  int pot;
3  void light_led(int a1, int a2, int a3, int a4);
4  void setup() {
5    pinMode(13, OUTPUT);
6    pinMode(12, OUTPUT);
7    pinMode(11, OUTPUT);
8    pinMode(10, OUTPUT);
9  }
10 void loop() {
11   pot = analogRead(potPin);
12   if(pot>=0 && pot<255){
13     light_led(13,12,11,10);
14   }
15   if(pot>=255 && pot<512){
16     light_led(12,13,11,10);
17   }
18   if(pot>=512 && pot<767){
19     light_led(11,13,12,10);
20   }
21   if(pot>=767 && pot<=1023){
22     light_led(10,13,12,11);
23   }
24   delay(10);
25 }
26
27 void light_led(int a1, int a2, int a3, int a4){
28   digitalWrite(a1, HIGH);
29   digitalWrite(a2, LOW);
30   digitalWrite(a3, LOW);
31   digitalWrite(a4, LOW);
32 }
```

Output Serial Monitor

Szkic używa 2420 bajtów (7%) pamięci programu. Maksimum to 32256 bajtów.  
Zmienne globalne używają 190 bajtów (9%) pamięci dynamicznej, pozostawiając 1858 bajtów dla zmiennych lokalnych. Maksimum to 2848 bajtów.



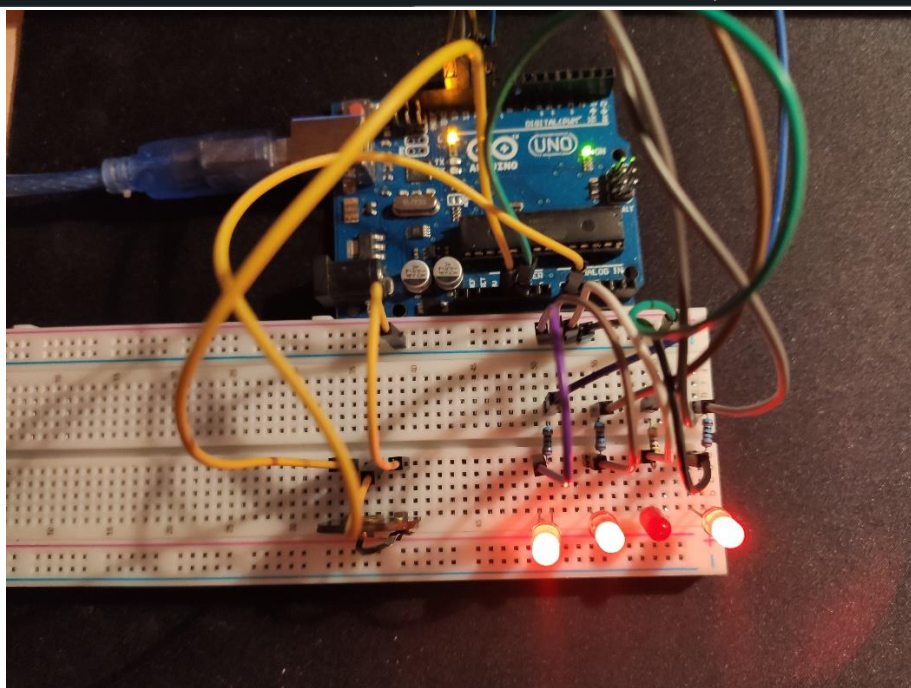
- c) Zbuduj i zaprogramuj licznik czterobitowy, którego wartość ma być wyświetlana na czterech diodach LED umieszczonych na płytce montażowej. Licznik ma pracować w pętli głównej programu i zliczać liczby od 0 do 15. Prędkość zliczania ma być ustawiana za pomocą potencjometru. Realizacja zadania wymaga zastosowania konwersji zmiennej typu integer (zmienna całkowita) na odpowiednią liczbę binarną. W języku C zmienne integer oznaczane są słowem kluczowym `int` i w systemie Arduino są zapisywane jako 16-bitowa liczba binarna.

```
sketch_oct25a | Arduino IDE 2.0.1-nightly-20221018
File Edit Sketch Tools Help
Arduino Uno
sketch_oct25a.ino
1 #define BIT(n,i) (n>>i&1)
2 int number, przerwa;
3 void setup(){
4   for(int i = 0; i<4; i++){
5     pinMode(10+i, OUTPUT);
6   }
7 }
8 void loop(){
9   for(int i = 0; i<16; i++){
10    for(int x = 0; x<4; x++){
11      if(BIT(i,x) == 0){
12        digitalWrite(13-x,LOW);
13      }else{
14        digitalWrite(13-x,HIGH);
15      }
16    }
17    delay(map(analogRead(0),0,1023,50,3000));
18  }
19 }
```

Output Serial Monitor

Szkic używa 1224 bajtów (3%) pamięci programu. Maksimum to 32256 bajtów.  
Zmienne globalne używają 9 bajtów (0%) pamięci dynamicznej, pozostawiając 2039 bajtów dla zmiennych lokalnych. Maksimu

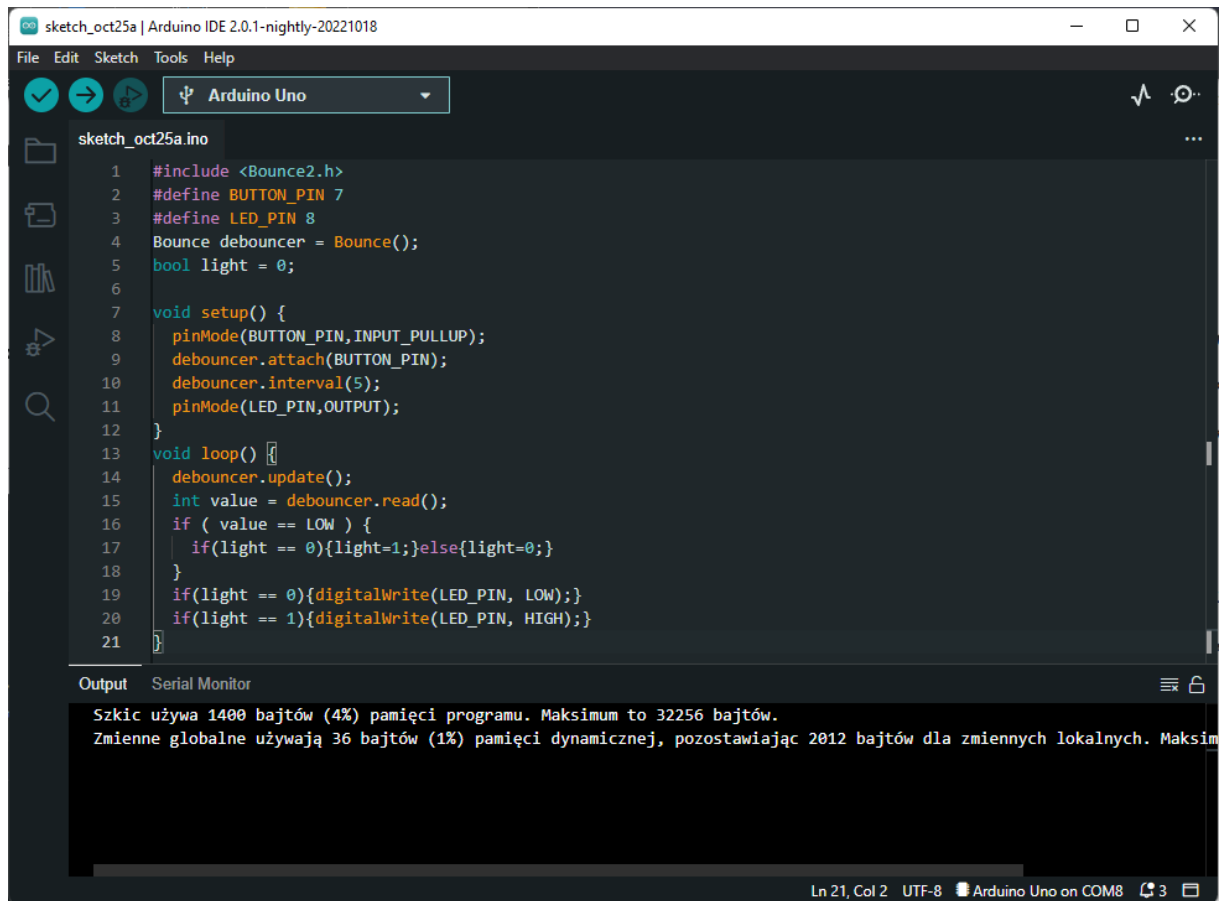
Ln 2, Col 21 UTF-8 Arduino Uno on COM8





## Zadanie 2:

- a) Zmodyfikuj szkic, tak aby dioda zaświeciła się po sygnale startowym, którym jest wciśnięcie przycisku, a przestała świecić po kolejnym naciśnięciu przycisku itd. Użyj biblioteki Bounce (ogólny przykład poniżej) do poprawnej obsługi mikroprzełącznika.

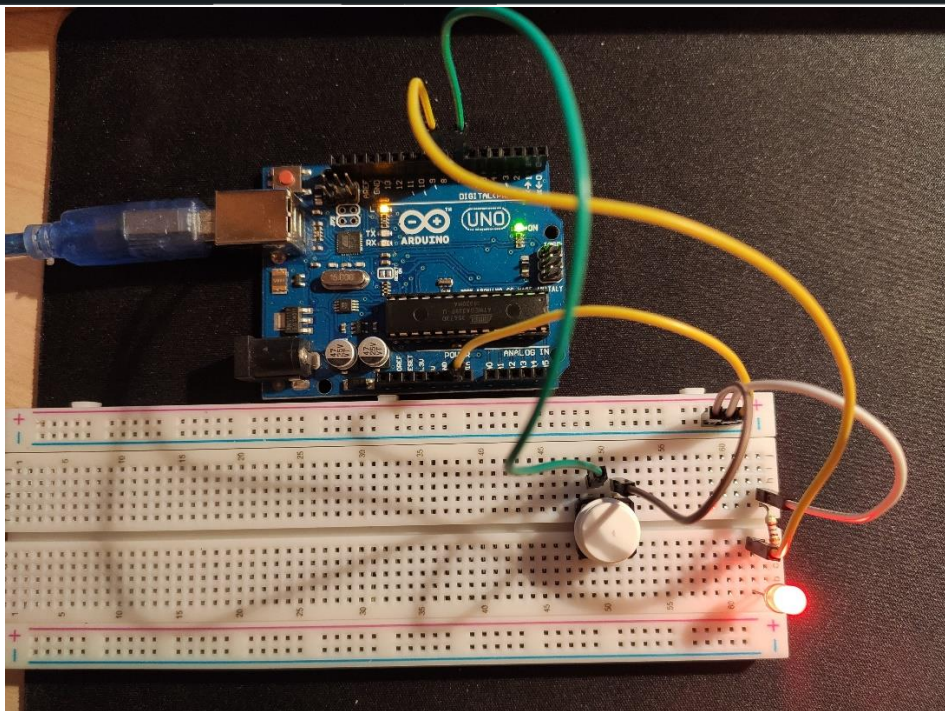


```
1 #include <Bounce2.h>
2 #define BUTTON_PIN 7
3 #define LED_PIN 8
4 Bounce debouncer = Bounce();
5 bool light = 0;
6
7 void setup() {
8   pinMode(BUTTON_PIN, INPUT_PULLUP);
9   debouncer.attach(BUTTON_PIN);
10  debouncer.interval(5);
11  pinMode(LED_PIN, OUTPUT);
12 }
13 void loop() {
14   debouncer.update();
15   int value = debouncer.read();
16   if ( value == LOW ) {
17     if(light == 0){light=1;}else{light=0;}
18   }
19   if(light == 0){digitalWrite(LED_PIN, LOW);}
20   if(light == 1){digitalWrite(LED_PIN, HIGH);}
21 }
```

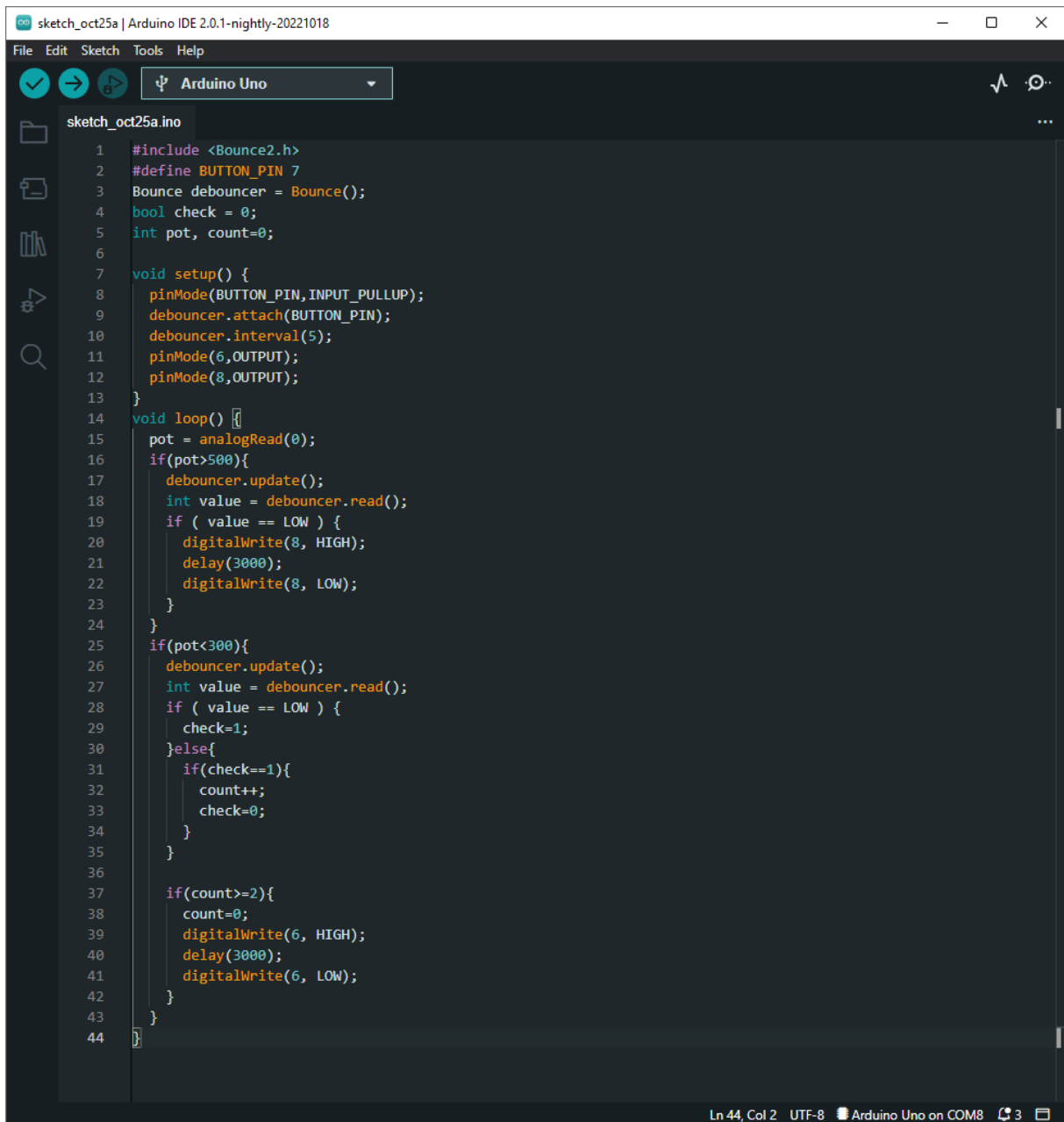
Output Serial Monitor

Szkic używa 1400 bajtów (4%) pamięci programu. Maksimum to 32256 bajtów.  
Zmienne globalne używają 36 bajtów (1%) pamięci dynamicznej, pozostawiając 2012 bajtów dla zmiennych lokalnych. Maksimum to 32768 bajtów.

Ln 21, Col 2 UTF-8 Arduino Uno on COM8

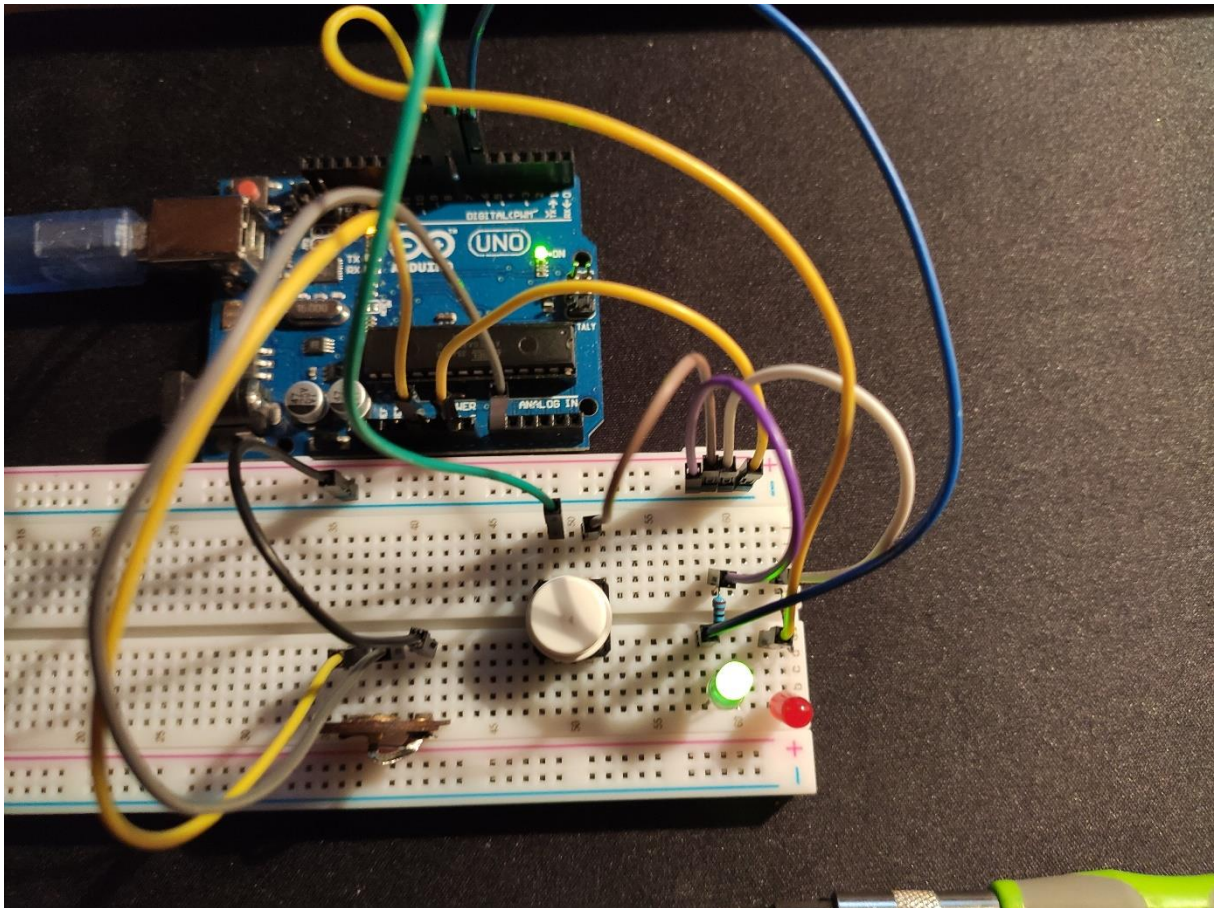


- b) Korzystając z rozwiązań poprzednich zadań przygotuj projekt pozwalający sterować diodami LED w następujący sposób (zakres potencjometru: 0-1000):
- 1) Dioda czerwona zaświeci gdy temperatura powietrza (ustawienie potencjometru) jest większa od 500 (skala umowna), a naciśnięcie przycisku włączy obwód świecenia na czas 3 sekund.
  - 2) Dioda zielona zaświeci gdy temperatura powietrza (ustawienie potencjometru) jest w przedziale 0 do 300 (skala umowna), a po dwukrotnym kliknięciu przycisku włączy świecenie na czas 3 sekund.



```
sketch_oct25a | Arduino IDE 2.0.1-nightly-20221018
File Edit Sketch Tools Help
[Icons] Arduino Uno
sketch_oct25a.ino
1 #include <Bounce2.h>
2 #define BUTTON_PIN 7
3 Bounce debouncer = Bounce();
4 bool check = 0;
5 int pot, count=0;
6
7 void setup() {
8   pinMode(BUTTON_PIN, INPUT_PULLUP);
9   debouncer.attach(BUTTON_PIN);
10  debouncer.interval(5);
11  pinMode(6, OUTPUT);
12  pinMode(8, OUTPUT);
13 }
14 void loop() {
15   pot = analogRead(0);
16   if(pot>500){
17     debouncer.update();
18     int value = debouncer.read();
19     if ( value == LOW ) {
20       digitalWrite(8, HIGH);
21       delay(3000);
22       digitalWrite(8, LOW);
23     }
24   }
25   if(pot<300){
26     debouncer.update();
27     int value = debouncer.read();
28     if ( value == LOW ) {
29       check=1;
30     }else{
31       if(check==1){
32         count++;
33         check=0;
34       }
35     }
36   }
37   if(count>=2){
38     count=0;
39     digitalWrite(6, HIGH);
40     delay(3000);
41     digitalWrite(6, LOW);
42   }
43 }
44 }
```

Ln 44, Col 2 UTF-8 Arduino Uno on COM8 3



- c) Prosta gra zręcznościowa: zbudowana z dwóch elementów (mikroprzełącznik), które będą testowały reakcję graczy oraz trzech sygnalizatorów (diody LED): dwóch informujących kto pierwszy wcisnął przycisk, oraz jednego informującego graczy o tym, kiedy mogą rozpocząć rywalizację (Start/Stop). Lista założeń działania gry:
- 1) Sygnalizator świeci przez losowy czas (dioda zielona).
  - 2) Sygnalizator gaśnie aktywując elementy testujące refleks graczy.
  - 3) Sprawdzenie refleksu graczy (kto pierwszy naciśnie swój przycisk).
  - 4) Zapalenie sygnalizatora gracza (dioda czerwona), który miał szybszy refleks.
  - 5) Powrót do punktu pierwszego.



```
sketch_oct25a | Arduino IDE 2.0.1-nightly-20221018
File Edit Sketch Tools Help
[Icons] Arduino Uno

sketch_oct25a.ino
1 #define BUTTON_PIN_2 6
2 #define BUTTON_PIN_1 7
3 #define LED_1_PIN 4
4 #define LED_2_PIN 2
5 #define LED_PIN 3
6 #include <Bounce2.h>
7 Bounce debouncer1 = Bounce(); Bounce debouncer2 = Bounce();
8 bool check = 0;
9 int value1 = 0;
10 int value2 = 0;
11 void setup() {
12   pinMode(BUTTON_PIN_1, INPUT_PULLUP); pinMode(BUTTON_PIN_2, INPUT_PULLUP);
13   debouncer1.attach(BUTTON_PIN_1); debouncer2.attach(BUTTON_PIN_2);
14   debouncer1.interval(5); debouncer2.interval(5);
15   pinMode(LED_PIN, OUTPUT); pinMode(LED_1_PIN, OUTPUT); pinMode(LED_2_PIN, OUTPUT);
16 }
17 void loop() {
18   if(check == 0){
19     digitalWrite(LED_PIN, LOW);
20     digitalWrite(LED_PIN, HIGH);
21     delay(2000);
22     digitalWrite(LED_PIN, LOW);
23     check = 1;
24   }
25   if(check == 1){
26
27     debouncer1.update();
28     value1 = debouncer1.read();
29     if ( value1 == LOW ) {
30       digitalWrite(LED_1_PIN, HIGH);
31       delay(500);
32       digitalWrite(LED_1_PIN, LOW);
33       check = 0;
34     }
35     debouncer2.update();
36     value2 = debouncer2.read();
37     if ( value2 == LOW ) {
38       digitalWrite(LED_2_PIN, HIGH);
39       delay(500);
40       digitalWrite(LED_2_PIN, HIGH);
41       check = 0;
42     }
43   }
44 }
```

Building sketch Ln 44, Col 2 UTF-8 Arduino Uno on COM8 3

