



# PRACOWNIA MIERNICTWA ELEKTRONICZNEGO/ INTERNETU RZECZY

## ĆWICZENIE NR 4

**Temat:**

### System sterowania diodami

Opracowanie wstępne:  
Rozwiązania mobilne w Internecie Rzeczy

**Komponenty:**

Arduino  
Rezystor 10 k $\Omega$  2 szt.

Dioda LED 8 szt.  
Rezystor 220  $\Omega$  8 szt.

Tact switch 2 szt.

**Cel ćwiczenia:**

Zbudowanie układu, który będzie wyświetlał przy pomocy szeregu 8 diod LED zadaną liczbę w systemie binarnym. Następnie należy zaprogramować kilka programów świecenia (mrużania) diod i dodać możliwość przełączania pomiędzy programami przy pomocy dwóch przycisków typu Tact switch.

# INSTRUKCJA

## Krok 0.

Należy podłączyć diodę LED w sposób pokazany w materiale „Podstawy budowy i programowania Arduino oraz budowania obwodów elektronicznych”

## Krok 1.

W podobny sposób co powyżej należy podłączyć sumarycznie 8 diod – każdą do osobnego pinu cyfrowego – tak by utworzyły prosty „wyświetlacz”. Następnie należy stworzyć kod, który wyświetli na diodach zadaną w programie liczbę w systemie binarnym. Przed blokiem `setup` należy zdefiniować:

a) Listę przechowującą numery pinów, do których podłączone są kolejne LEDy:

```
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9};
```

W tym przykładzie diody podłączone zostały do pinów 2, 3, 4, 5, 6, 7, 8 i 9.

b) Funkcję konwertującą liczbę dziesiętną na postać binarną (w naszym przykładzie funkcja będzie od razu zapalała odpowiednie diody):

```
void convertToBin(int number) {  
    for (int i=0; i<8; i++){  
        digitalWrite(ledPins[i], number>>i & 1);  
    }  
}
```

W powyższej funkcji użyto operacji przesunięcia bitowego w prawo (>>) oraz porównania bitowego (&). Kryje się za tym następująca logika:

Przykładowa liczba 6 ma binarną postać 0b00000110. W pierwszym kroku przesuwamy ją o 0 miejsc i porównujemy z wartością 0b00000001. Uzyskujemy wartość 0b00000000 i przypisujemy ją do pierwszej diody (pierwsza dioda zgaszona). Następnie przesuwamy liczbę o jedno miejsce w prawo, otrzymując 0b00000011. Teraz z porównania z jedyneką wyjdzie nam już 0b00000001 (druga dioda zapalona). Liczba 6 przesunięta o dwa miejsca będzie miała postać 0b00000001 (trzecia dioda również będzie zapalona). Po sprawdzeniu wszystkich diod powinniśmy otrzymać schemat xxxoox, który faktycznie odpowiada liczbie 6.

W bloku `setup` powinniśmy ustawić piny przypisane do LEDów jako wyjście. Możemy znów wykorzystać pętlę `for`:

```
for (int i = 0; i < 8; i++) {  
    pinMode(ledPins[i], OUTPUT);  
}
```

W bloku `loop` wystarczy wówczas, że wywołamy funkcję `convertToBin`, podając za argument odpowiednią liczbę:

```
convertToBin(6);
```

## Krok 2a.

Należy zaprogramować trzy przykładowe programy świecenia diod (mrganie wszystkich, licznik binarny odliczający w górę oraz licznik binarny odliczający w dół) oraz dodać przycisk przełączający program na następny.

```
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9};
int buttonPin = 13;
int program = 1;

void convertToBin(int number) {
  for (int i=0; i<8; i++){
    digitalWrite(ledPins[i], number>>i & 1);
  }
}

void blink() {
  while (true) {
    convertToBin(255);
    delay(1000);
    convertToBin(0);
    delay(1000);
    if (digitalRead(buttonPin) == HIGH) {break;}
  }
}

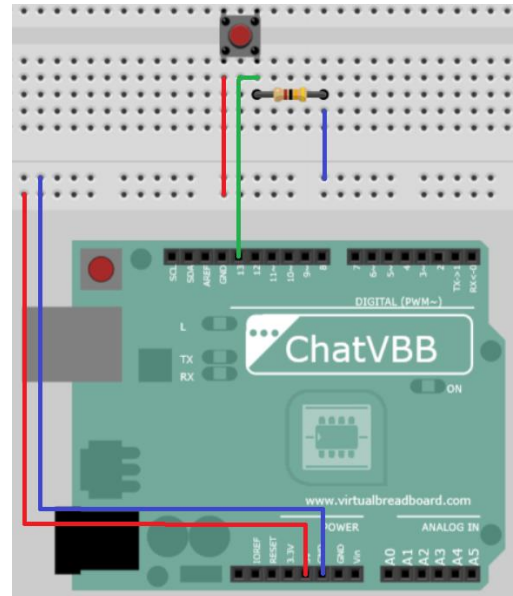
void countUp() {
  int counter = 0;
  while (true) {
    convertToBin(counter);
    counter++;
    if (counter == 256) {counter = 0;}
    delay(500);
    if (digitalRead(buttonPin) == HIGH) {break;}
  }
}

void countDown() {
  int counter = 255;
  while (true) {
    convertToBin(counter);
    counter--;
    if (counter == 0) {counter = 255;}
    delay(500);
    if (digitalRead(buttonPin) == HIGH) {break;}
  }
}

void setup() {
  pinMode(buttonPin, INPUT);
  for (int i = 0; i < 8; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}

void loop() {
  if (program == 1) {
    blink();
    program++;
  }
  if (program == 2) {
    countUp();
    program++;
  }
  if (program == 3) {
```

### Podłączenie przycisku:



Dla sprawniejszego przełączania programów, można funkcję `delay` zamienić poniższą, która częściej sprawdza wciśnięcie przycisku:

```
void breakDelay(long ms) {
  for (int i = 1; i < ms/20; i++){
    delay(20);
    if (digitalRead(buttonPin) == HIGH)
    {break;}
  }
}
```

```
    countdown();  
    program++;  
}  
if (program > 3) {  
    program = 1;  
}  
}
```

### **Krok 2b.**

Należy podłączyć drugi przycisk i dodać odpowiedni kod do przełączania programu na poprzedni.

Opracowanie końcowe:

**Jakie mogą być przykładowe zastosowania podobnych systemów (sygnały świetlne o zaprogramowanej konfiguracji) w rozwiązaniach Internetu Rzeczy**

### **ZADANIE DODATKOWE**

Należy skonstruować dwa dodatkowe programy świecenia:

1. Trzybitowy (trzy diody) wąż poruszający się w prawo i w lewo. Tutaj przydatne okazać się mogą operacje przesunięć bitowych w prawo i w lewo omówione w kroku 1.
2. Dowolny program inny od omówionych wcześniej