

# Politechnika Świętokrzyska Wydział Elektrotechniki, Automatyki i Informatyki

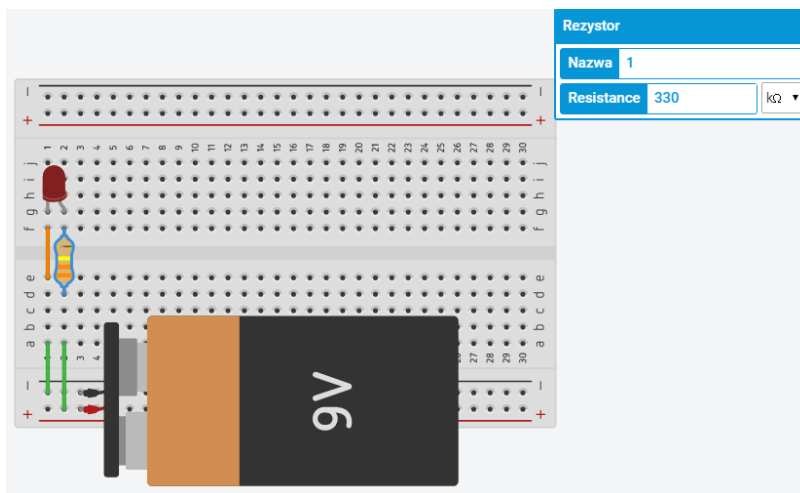
Technologie IoT rozproszone sieci sensoryczne

Laboratorium 2

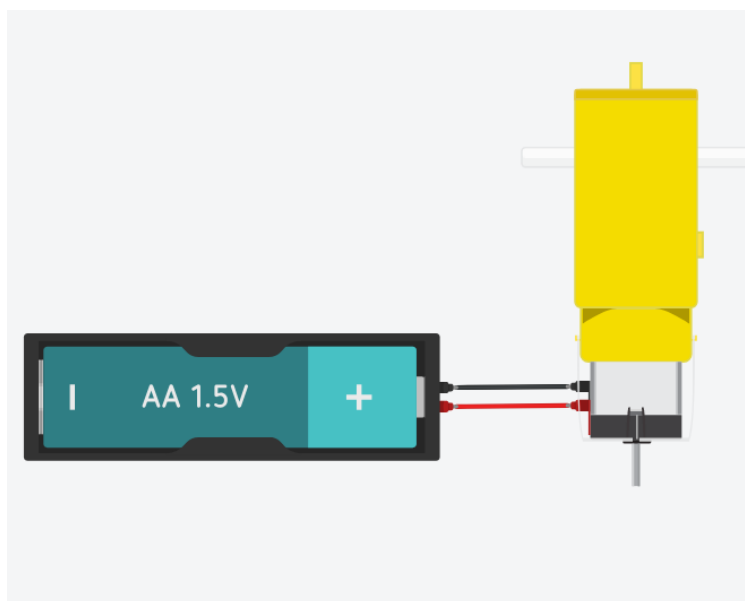
Adrian Dorota  
3ID15B

## 1. Rozdział 2 instrukcja 2.1.1.5

Część 1: Stwórz prosty obwód, wykorzystując baterię 9V jako źródło zasilania

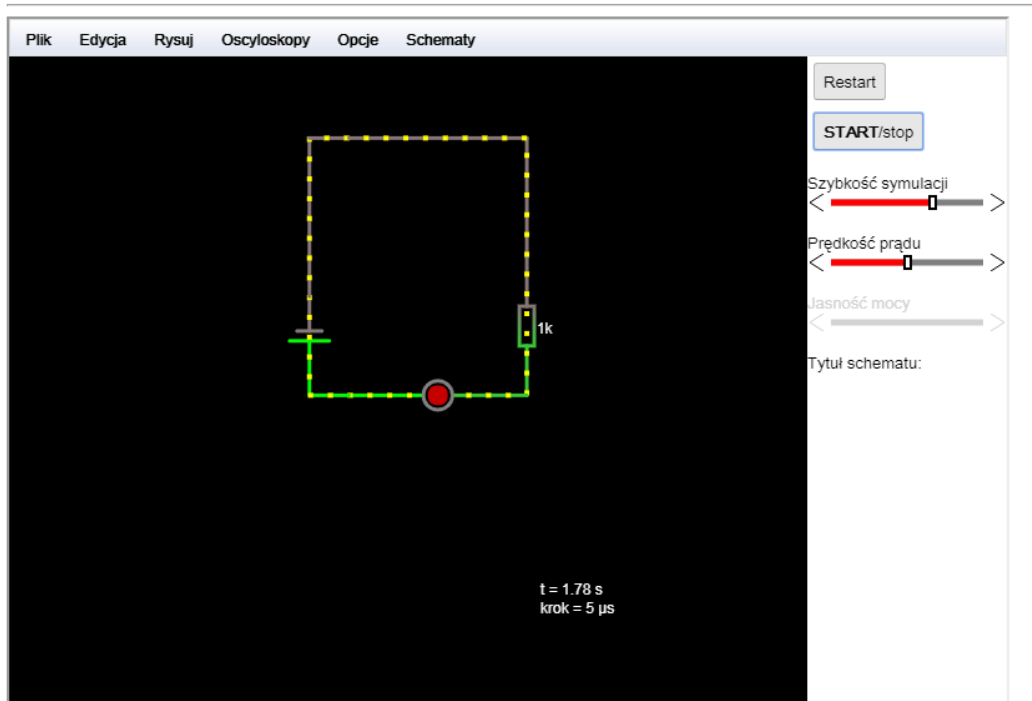


Część 2: Utwórz prosty obwód, wykorzystując jako źródło zasilania baterię 1,5 V



## 2. Rozdział 2 instrukcja 2.1.2.8

Część 1: Należało wykonać prosty układ składający się z rezystora, diody, źródła napięcia, oraz odczytania wartości na poszczególnych komponentach.



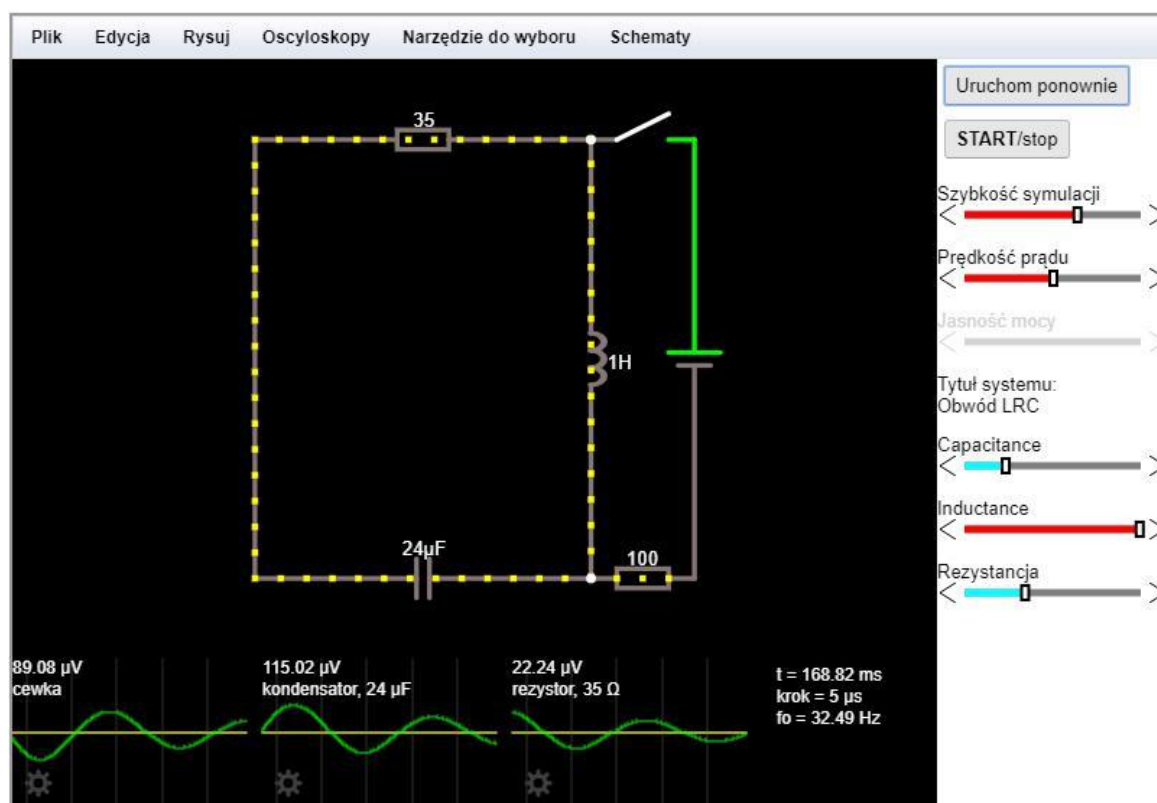
rezystor	LED (default-led)	źródło napięcia
I = 9.75 mA	I = 9.75 mA	I = 9.75 mA
Vd = 3.22 V	Vd = 1.78 V	Vd = 5 V
R = 330 $\Omega$		(R = 512.73 $\Omega$ )
P = 31.38 mW	P = 17.38 mW	P = -48.76 mW

Odpowiedzi do pytań z instrukcji.

- Jakie jest napięcie na Diodzie Led
  - Napięcie Diodzie led wynosi 1.78 V.
- Jakie jest napięcie na rezystorze.
  - -Napięcie na rezystorze wynosi 3.22 V.
- Jakie napięcie jest na baterii.
  - Napięcie na baterii wynosi 5 V.

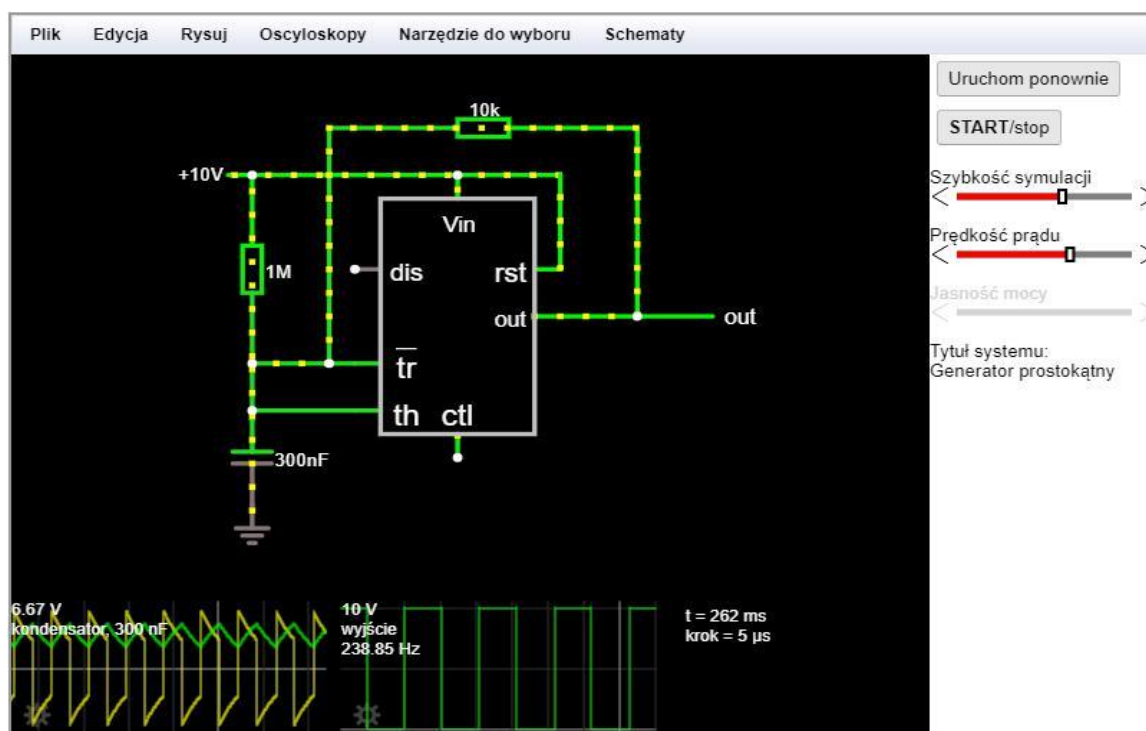
## Część 2:symulacje na dwóch obwodach.

- Obwód LCR:



Schemat ten przedstawia prąd zmienny (sinusoidalny), jest on używany przez wiele krajów transportu energii na dużą odległość.

- Generator sygnałów prostokątnych:



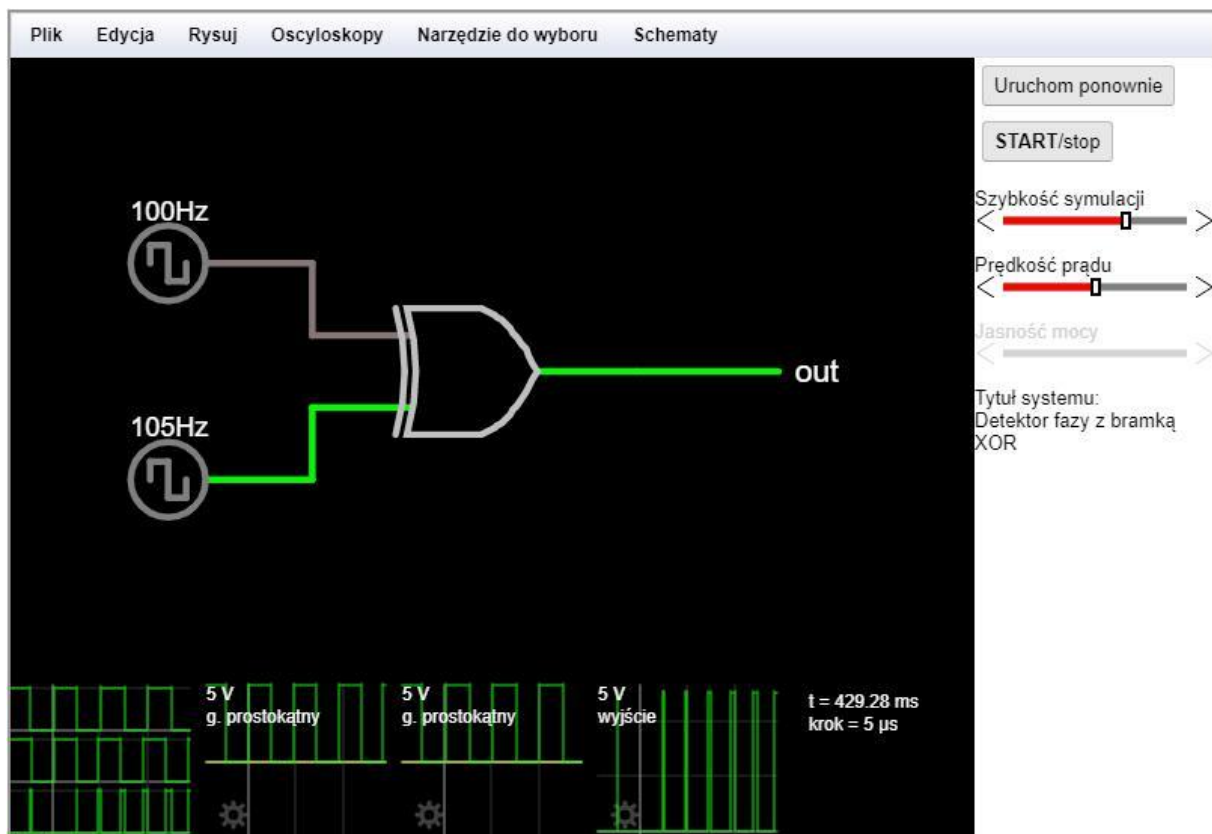
Fale kwadratowe są najczęściej widoczne są w obwodach cyfrowych.

## Część 3: Oscyloskop.

**Oscyloskop** przyrząd elektroniczny służący do obserwowania, obrazowania i badania przebiegów zależności pomiędzy dwiema wielkościami elektrycznymi, bądź innymi wielkościami fizycznymi reprezentowanymi w postaci elektrycznej.

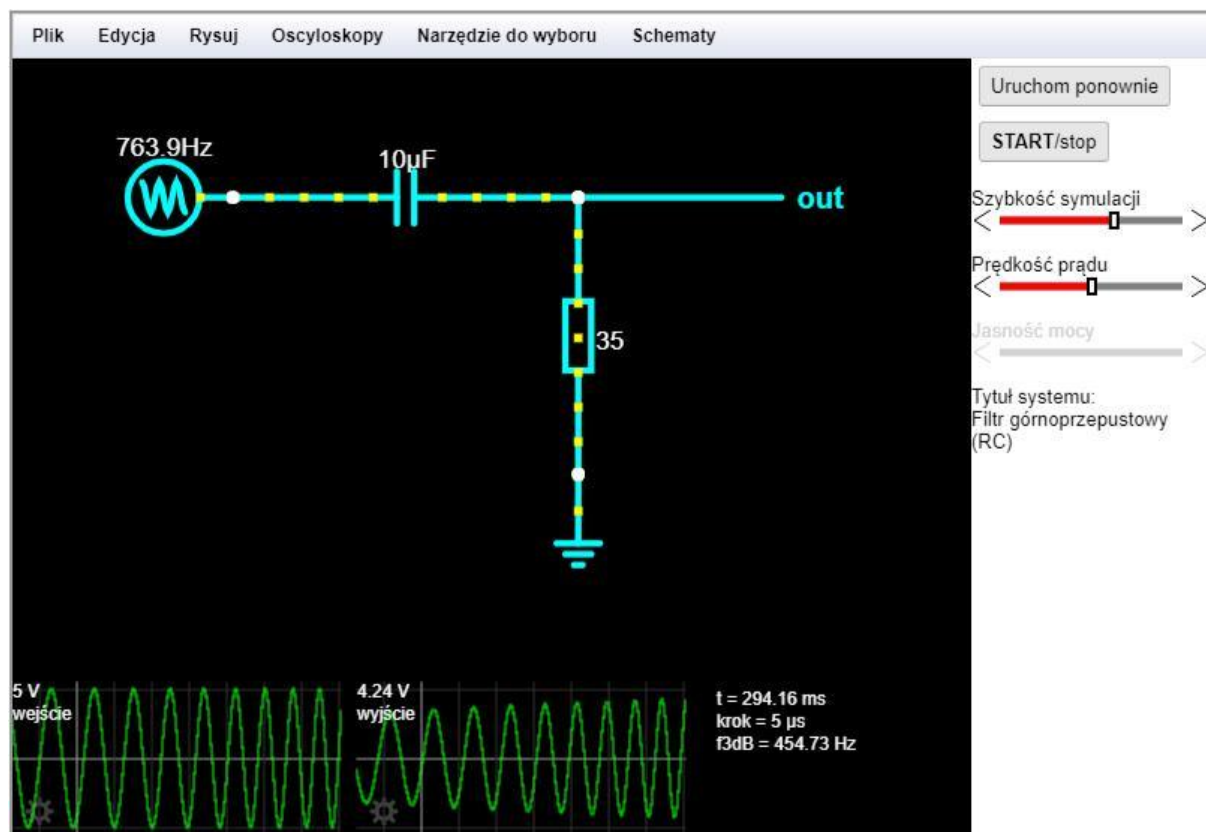


Detektor faz x bramką XOR:



W tym przypadku po podpięciu oscyloskopu, możemy zobaczyć jaki przebieg ma wykres. Jest on prostokątny oraz widoczna jest wartość napięcia na wejściu czy wyjściu.

Filtr pasywny:



Wykres przedstawia przebieg sinusoidalny napięcia, z którego możemy poznać wartość napięcia w danej chwili.

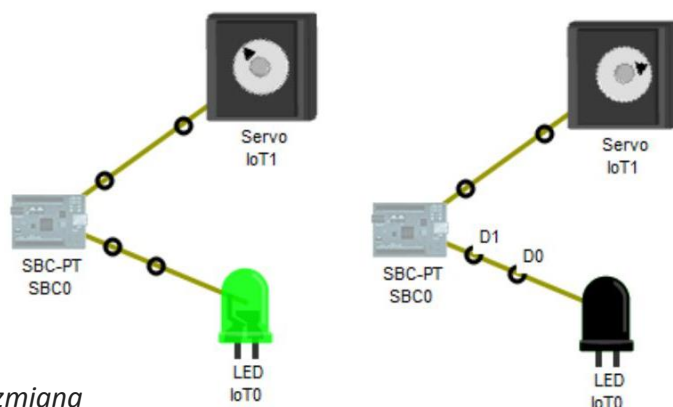
## Część 4: Packet Tracer - Symulowanie urządzeń IoT

### Instrukcja 2.2.1.4 Packet Tracer

```
from gpio import *
from time import *
```

```
def main():
    ping mode(1, OUT)
    print("blinking")
    while true:
        digitalwrite(1, HIGH);
        customwrite(0,50); # ← zmiana
        delay(1000)
        digitalwrite(1, LOW);
        customwrite(0,255); # ← zmiana
        delay(500)
```

```
if __name__ == "__main__":
    main()
```



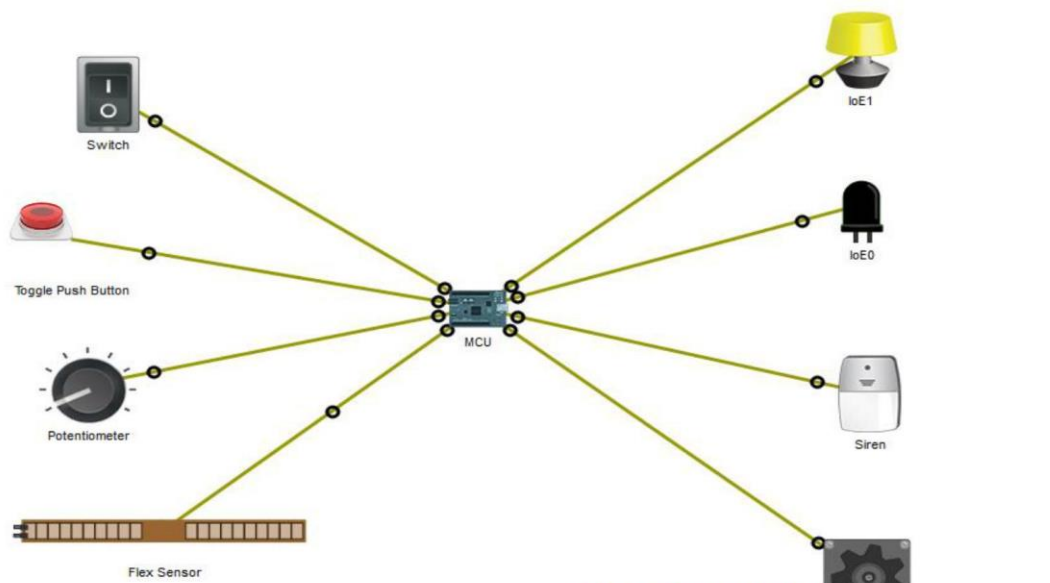
Co można zmienić, aby serwo obróciło się w przeciwnym kierunku, gdy dioda miga?

- wystarczy zmienić zakres jego pracy

## Część 5 : Packet Tracer - czujniki i mikrokontroler PT

### Instrukcja 2.3.1.2 Packet Tracer

A.



Należało zmienić sposób włączania elementów z:

- Switch włącza lampkę
- Toggle push Button, diode LED

Na:

- Switch włącza diode LED
- Toggle Push Button, Lampkę

```
def readFromSensors():
    global switchValue # declare switchValue
    global togglePushButtonValue # declare togglePushButtonValue
    global potentiometerValue # declare potentiometerValue
    global flexSensorValue # declare flexSensorValue

    togglePushButtonValue = digitalRead(0)
    switchValue = digitalRead(1) # read Toggle Push Button
    potentiometerValue = analogRead(A0) # read Potentiometer
    flexSensorValue = analogRead(A1) # read Flex Sensor

def writeToActuators():
    if (switchValue == HIGH): # evaluates if switch is on
        customWrite(2, "2") # turn on the light
    else:
        customWrite(2, "0") # turn off the light

    if (togglePushButtonValue == HIGH): # evaluates if toggle push button is pressed
        digitalWrite(3, HIGH) # turn on the LED
    else:
        digitalWrite(3, LOW) # turn off the LED

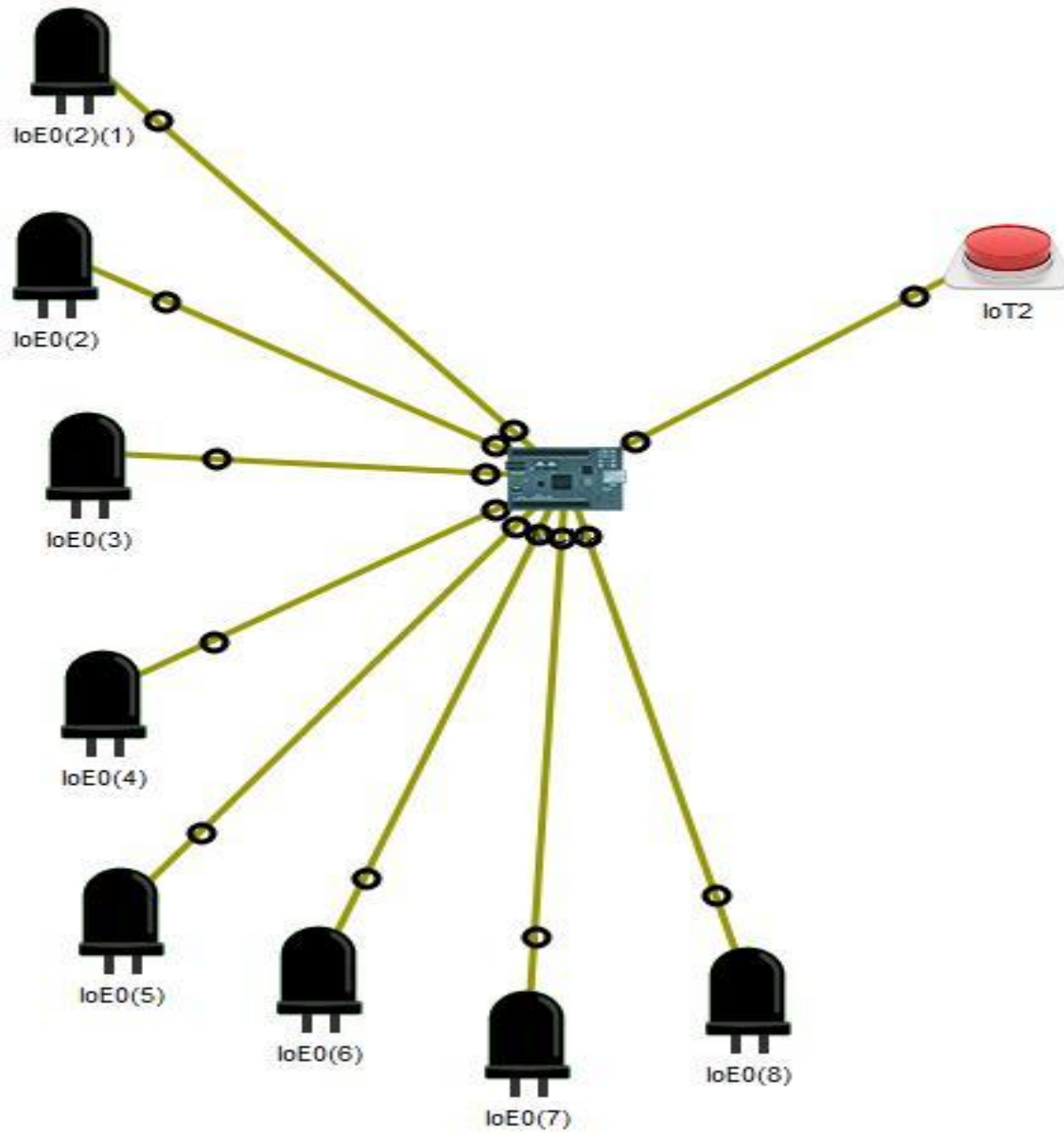
    if (potentiometerValue > 512): # evaluates if potentiometer is turned right
        customWrite(4, HIGH) # turn on the siren
    else:
        customWrite(4, LOW) # turn off the siren

    if (flexSensorValue > 0): # evaluates if flex sensor is bent
        analogWrite(5, flexSensorValue) # turn on the servo motor
    else:
        analogWrite(5, 0) # turn off the servo motor

def main(): # defines the main function
    pinMode(0, IN) # sets digital slot 0 (Toggle Push Button) to input
    pinMode(1, IN) # sets digital slot 1 (Switch) to input
    pinMode(2, OUT) # sets digital slot 2 (Light) to output
    pinMode(3, OUT) # sets digital slot 3 (LED) to output
    pinMode(4, OUT) # sets digital slot 4 (Siren) to output
    pinMode(5, OUT) # sets digital slot 5 (Servo Motor) to output

    while True:
        readFromSensors()
        writeToActuators()
```

- B. Kolejnym zadaniem było, by do mikrokontrolera podłączyć 8 diod, które zapalały by się jedna po drugiej przy kolejnych naciśnięciach przycisku.





```

from gpio import *
from time import *

def SwitchAllLeds(leds,LH):
    for i in range(1,leds-1):
        digitalWrite(i,LH)

def main():
    pinMode(1, OUT)
    pinMode(0, IN)

    initial=1
    last=8

    buttonPressed=False
    totalLeds=8
    SwitchAllLeds(totalLeds,LOW)

    while True:
        valueRead=digitalRead(0)
        if valueRead>0 and buttonPressed==False:
            digitalWrite(initial, HIGH)
            digitalWrite(last, LOW)
            buttonPressed=True
        elif valueRead==0 and buttonPressed==True:
            SwitchAllLeds(totalLeds,LOW)
            buttonPressed=False
            last=initial
            initial=initial%8+1
        delay(50)

if __name__ == "__main__":
    main()

```