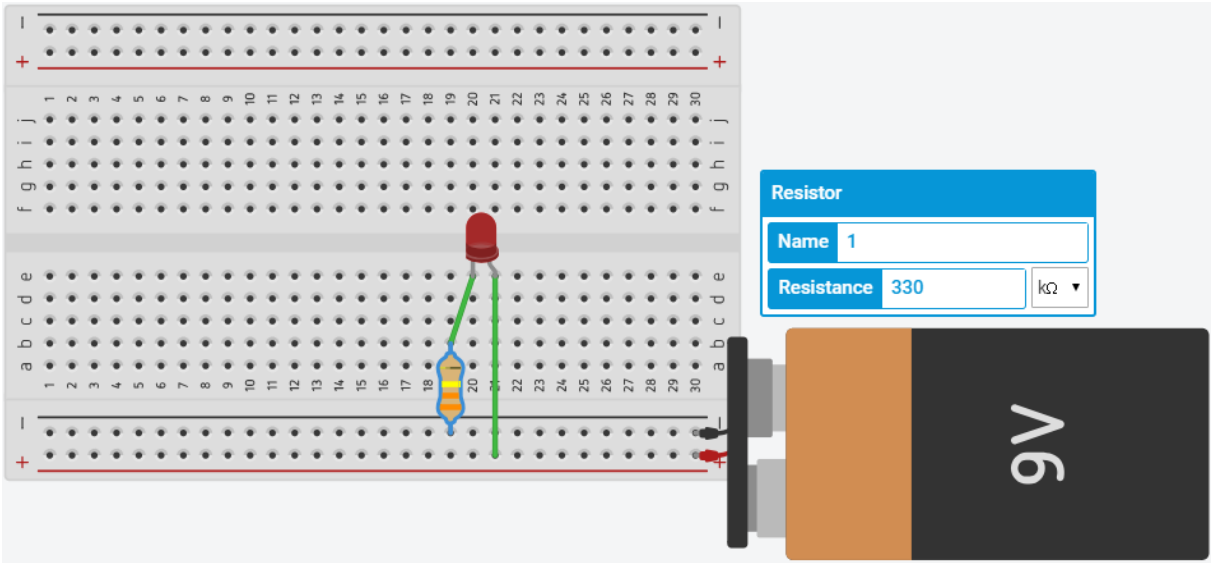
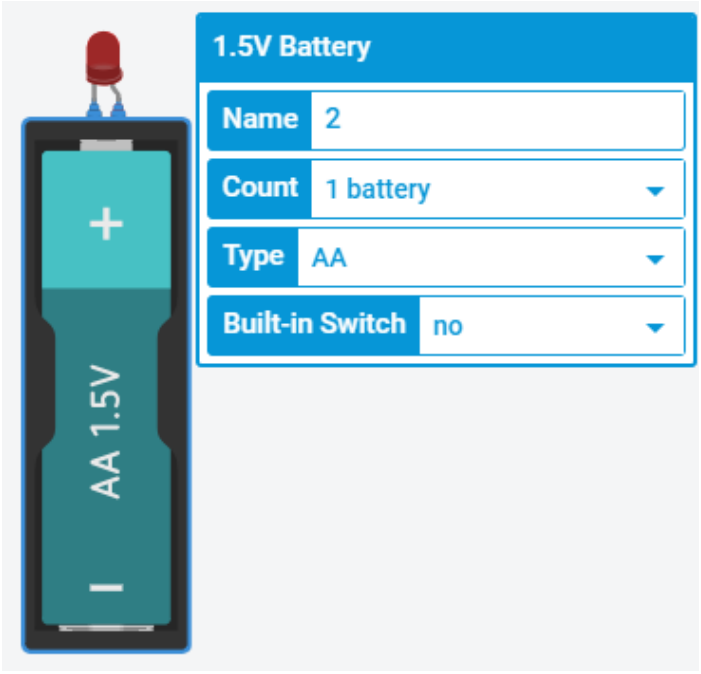
Politechnika Świętokrzyska w Kielcach **Wydział Elektrotechniki, Automatyki i Informatyki**	
Technologie IoT rozproszone sieci sensoryczne	
Czujniki, elementy wykonawcze i mikrokontrolery	Wykonał: **Wojciech Jabłoński, Joanna Gmyr** Grupa: **3ID15B**
Numer laboratorium: **2**	Data wykonania: **15.11.2018**

1. **Rozdział 2 instrukcja 2.1.1.5**

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



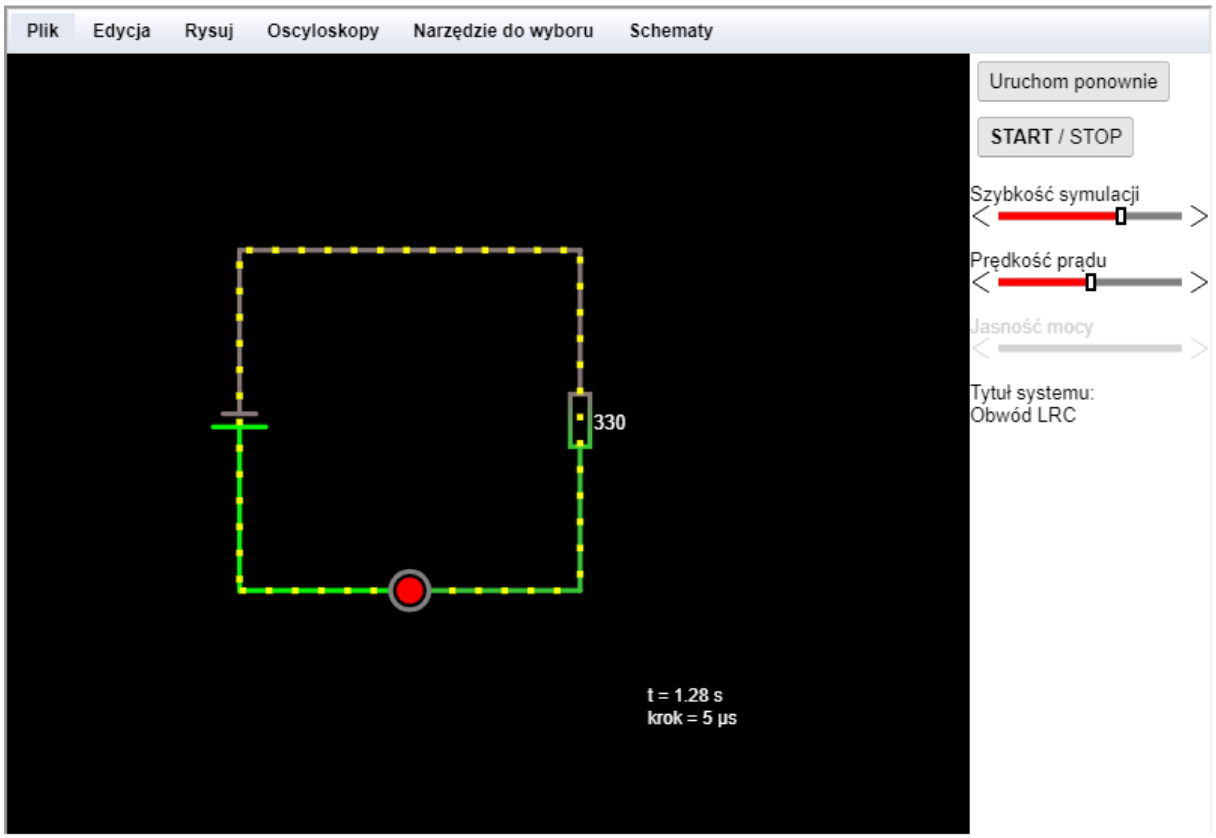
Część 2: Utwórz prosty obwód, wykorzystując jako źródło zasilania baterię 1,5 V



2. Rozdział 2 instrukcja 2.1.2.8

Część 1:

Do wykonania na zajęciach był prosty układ składający się z rezystora, diody, źródła napięcia oraz odczytania wartości na poszczególnych komponentach.



Odczytane wartości:

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 Ω	P = 17.38 mW	(R = 512.73 Ω)
P = 31.38 mW		P = -48.76 mW

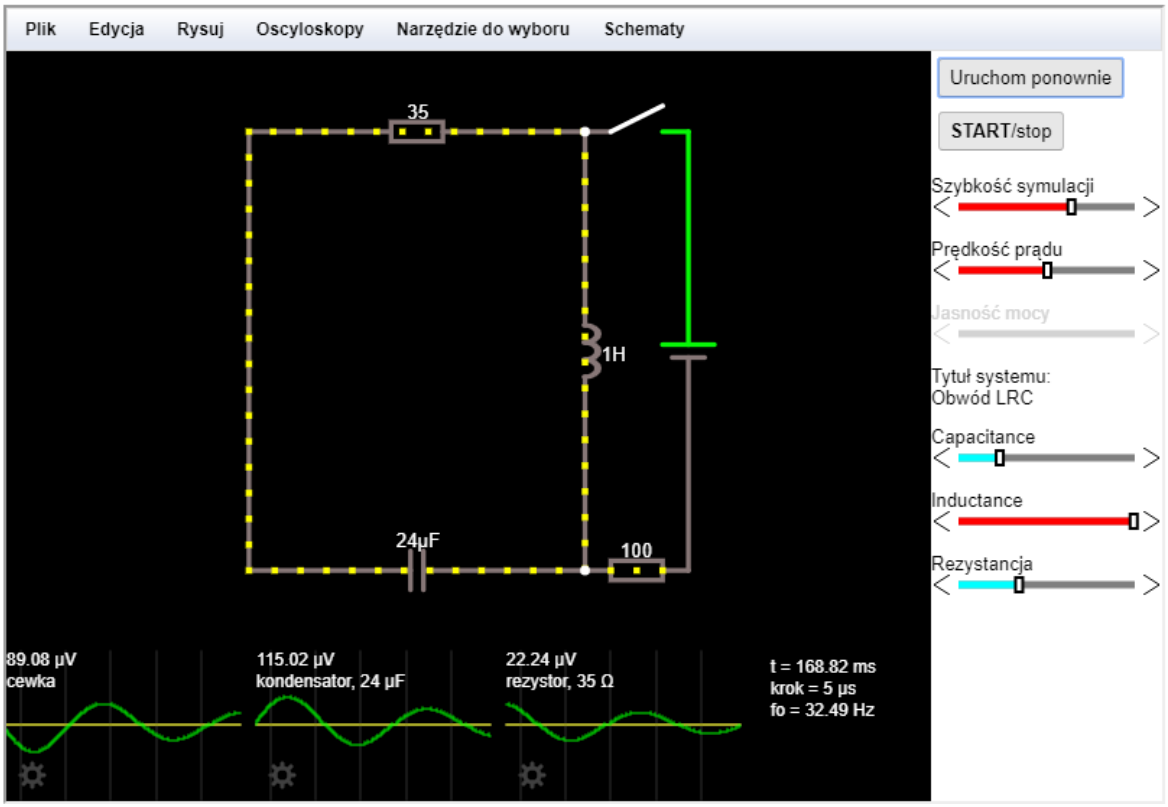
Odpowiedzi do pytań z instrukcji.

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

Część 2:

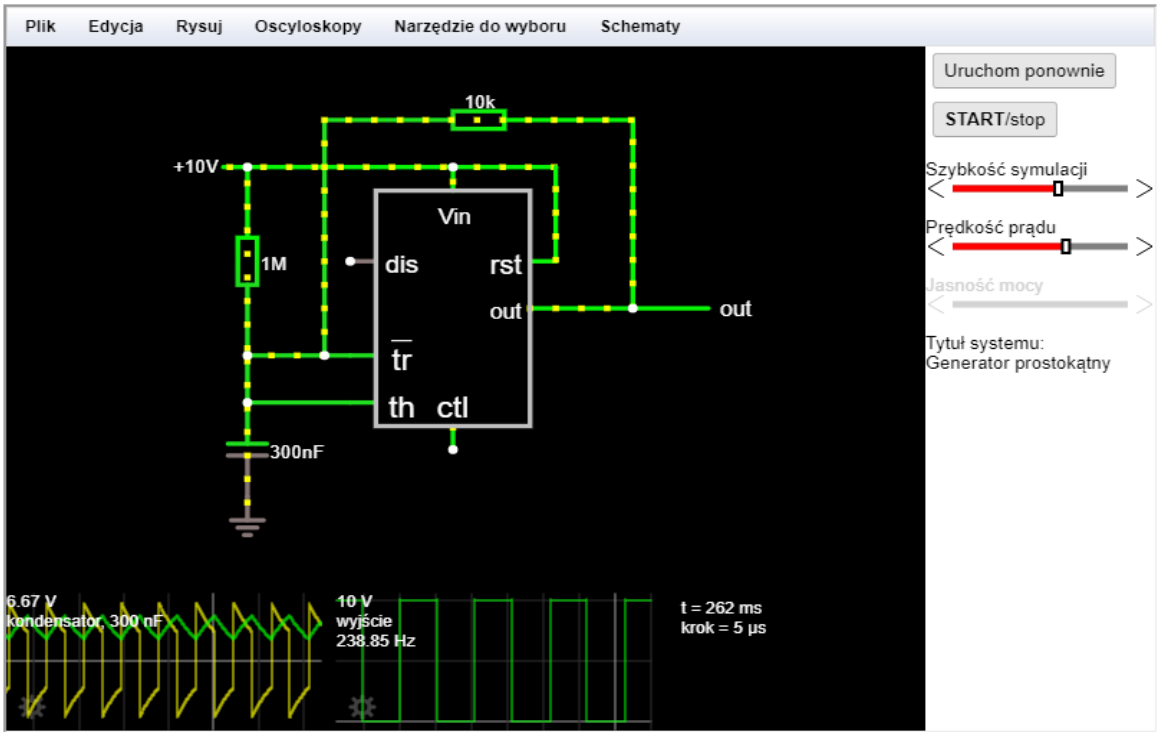
W kolejnej części ćwiczenia mieliśmy za zadanie przeprowadzenie symulacji na dwóch obwodach.

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:

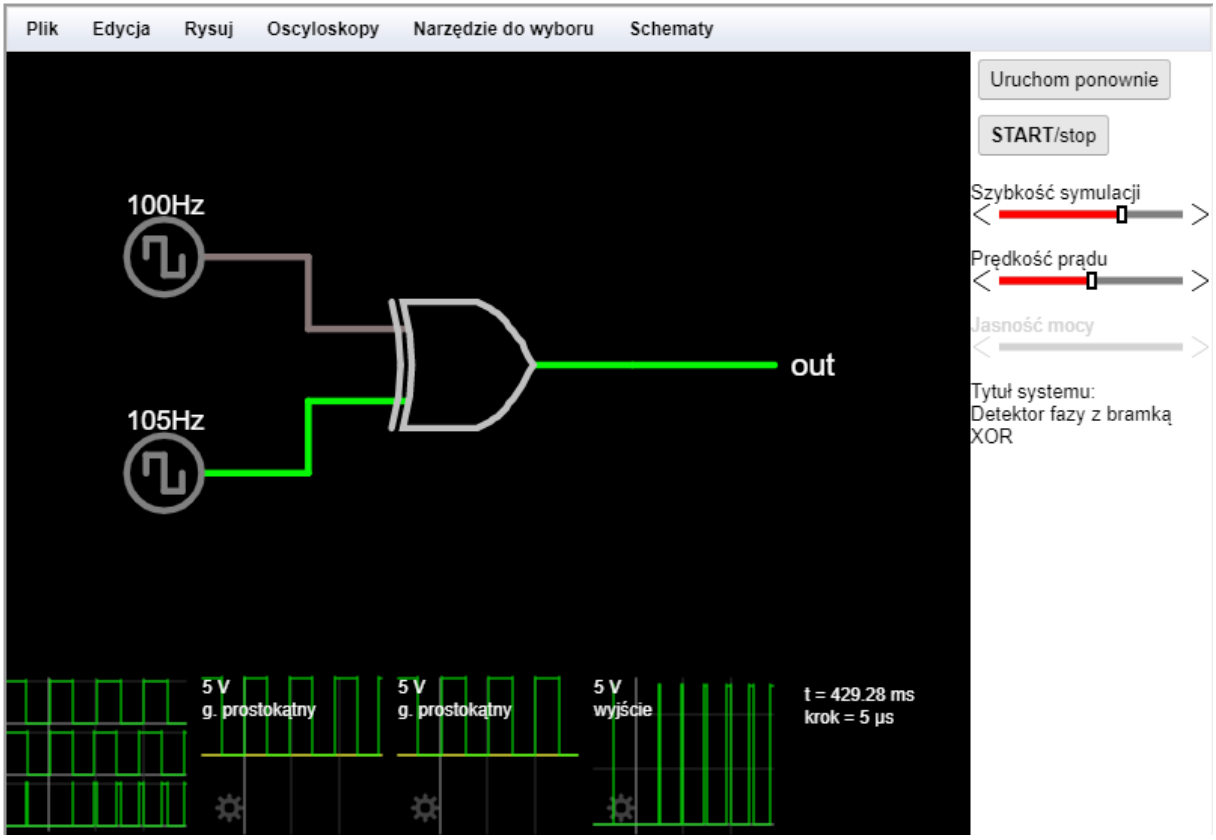
Oscyloskop.

Oscyloskop – przyrząd służący do obserwowania, obrazowania i badania przebiegów pomiędzy dwiema wielkościami elektrycznymi.

Poniżej przedstawiony jest oscyloskop.

Podpięcie oscyloskopu do różnych schematów.

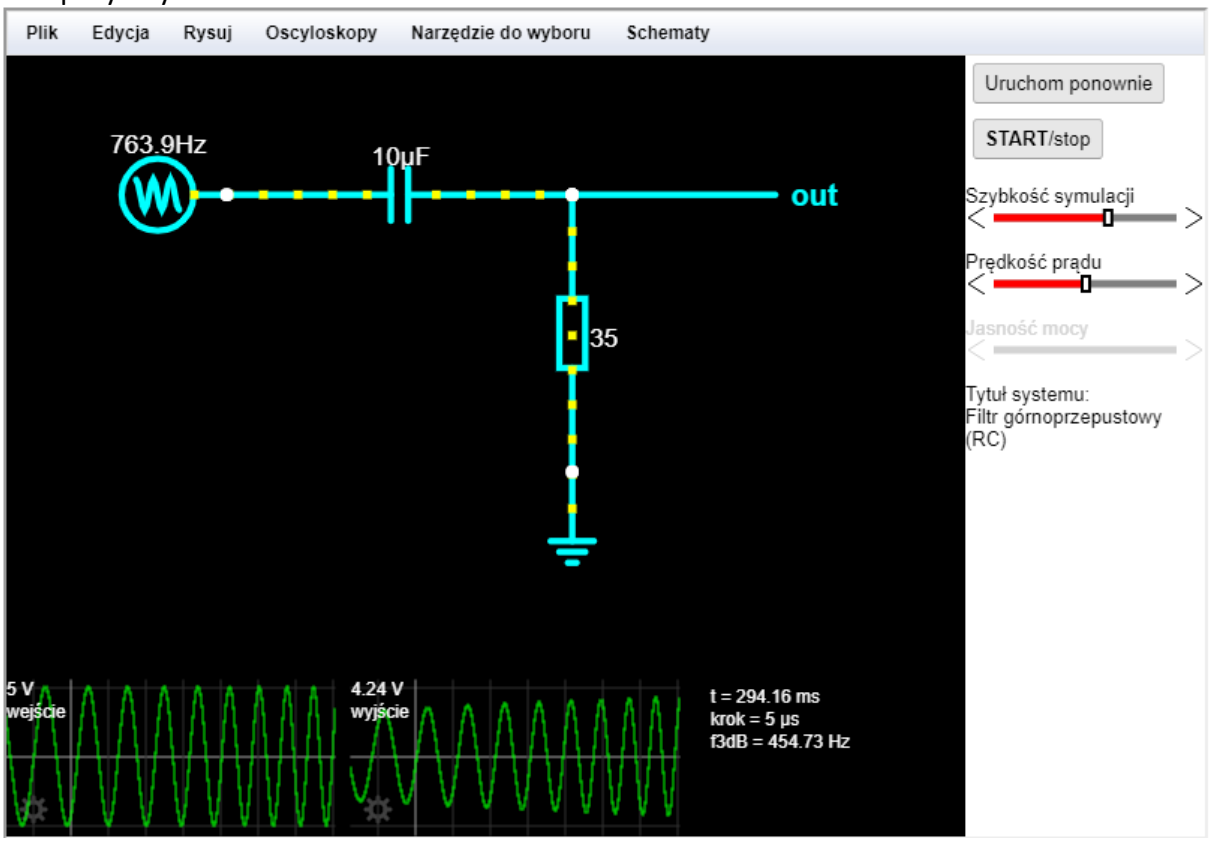
Detektor faz x bramką XOR:



Obserwacje:

Po podpięciu oscyloskopu możemy zobaczyć jaki przebieg ma wykres. W tym przypadku jest on prostokątny oraz możemy zobaczyć jakie napięcie jest na wejściu czy wyjściu, oczywiście nie są to wszystkie funkcje oscyloskopu.

Filtr pasywny:



Obserwacje:
Na tym obrazku możemy zaobserwować przebieg sinusoidalny, dzięki temu możemy poznać wartość napięcia w danej chwili.

3. Packet Tracer - Symulowanie urządzeń IoT

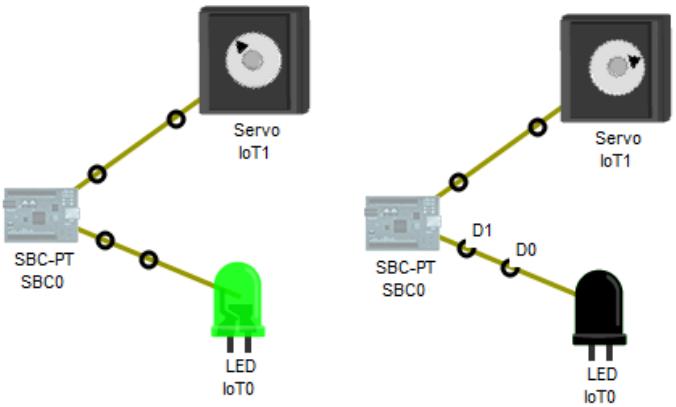
Instrukcja 2.2.1.4 Packet Tracer

Część 1:

Kod w Python z kontrolera.

```
from gpio import *
from time import *

def main():
    pinMode(1, OUT)
    print("Blinking")
    while True:
        digitalWrite(1, HIGH);
        customWrite(0, 50); #<- tutaj dokonaliśmy zmian
        delay(1000)
        digitalWrite(1, LOW);
        customWrite(0, 255); #<- tutaj dokonaliśmy zmian
        delay(500)

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

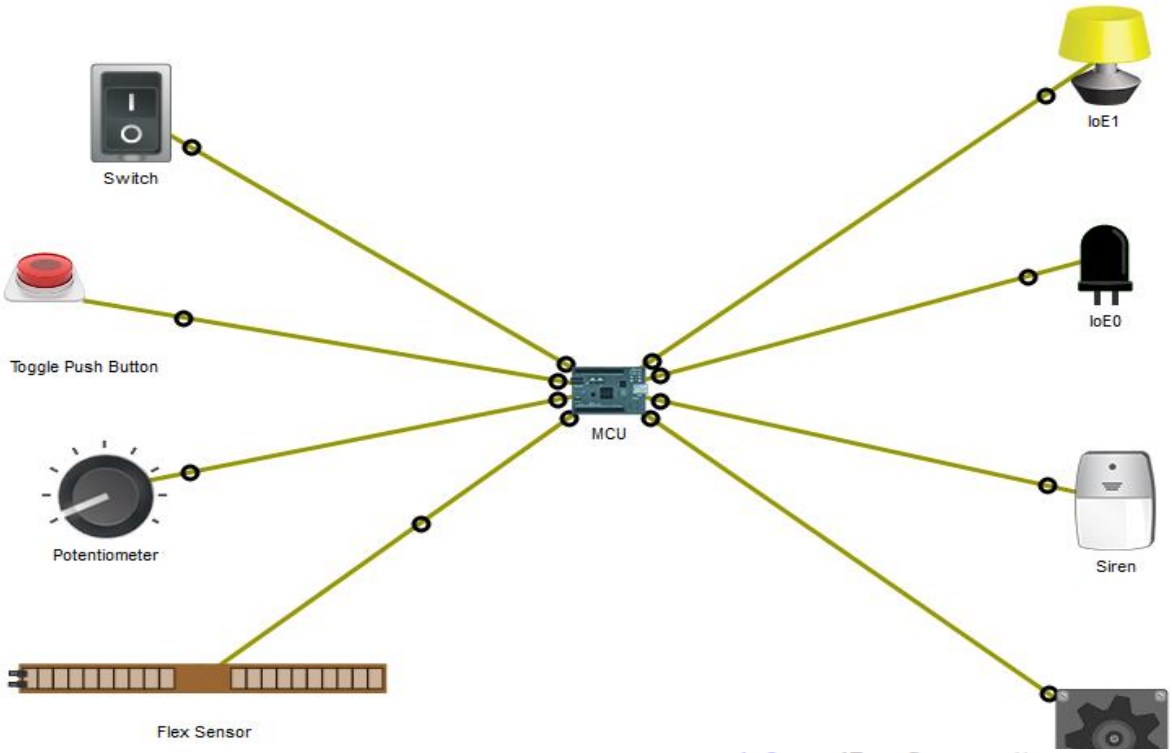
Zrzut ekrany z PT.



Co można zmienić, aby serwo obróciło się w przeciwnym kierunku, gdy dioda miga?
- Żeby serwo obracało się w drugą stronę wystarczy zmienić zakres jego pracy.

4. Packet Tracer - czujniki i mikrokontroler PT

Instrukcja 2.3.1.2 Packet Tracer
Część 1:

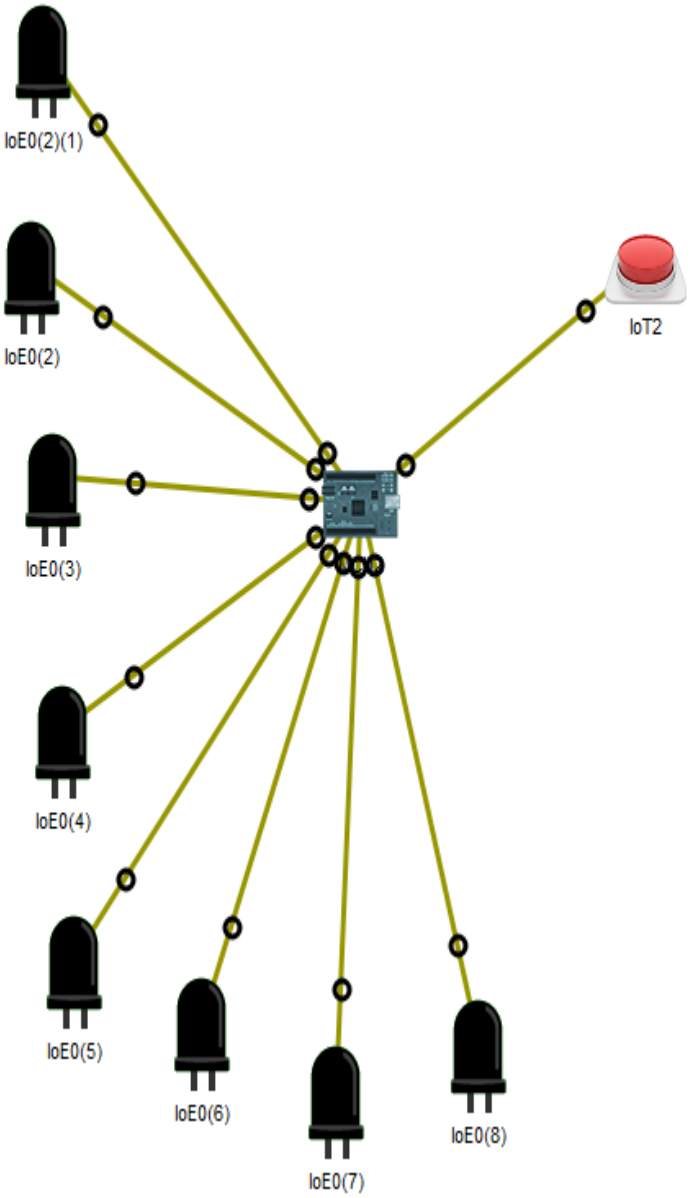


Ćwiczenie to polegało na tym, że bazowo switch włączał lampkę, a toggle push button diodę LED. Mieliśmy zmienić to tak, żeby toggle push button włączał lampkę, a switch diodę LED.

Kod w Python z kontrolera --->

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

    togglePushButtonValue = digitalRead(0)
    switchValue = digitalRead(1) # read To
    potentiometerValue = analogRead(A0) #
    flexSensorValue = analogRead(A1) # rea

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

    if (togglePushButtonValue == HIGH): #
        digitalWrite(3, HIGH) # turn on th
    else:
        digitalWrite(3, LOW) # turn off th

    if (potentiometerValue > 512): # evalu
        customWrite(4, HIGH) # turn on the
    else:
        customWrite(4, LOW) # turn off the

    if (flexSensorValue > 0): # evaluates
        analogWrite(5, flexSensorValue) #
    else:
        analogWrite(5, 0) # turn off the m

def main(): # defines the main function
    pinMode(0, IN) # sets digital slot 0 (
    pinMode(1, IN) # sets digital slot 1 (
    pinMode(2, OUT) # sets digital slot 2
```

Część 2:
Kolejną częścią było, aby do mikrokontrolera podłączyć 8 diod, a następnie aby te diody zapalały się jedna po drugiej podczas naciśnięcia przycisku.



Kod w Python z kontrolera:

```
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()
```