

Politechnika Świętokrzyska w Kielcach

Wydział Elektrotechniki, Automatyki i Informatyki

Laboratorium: IoT

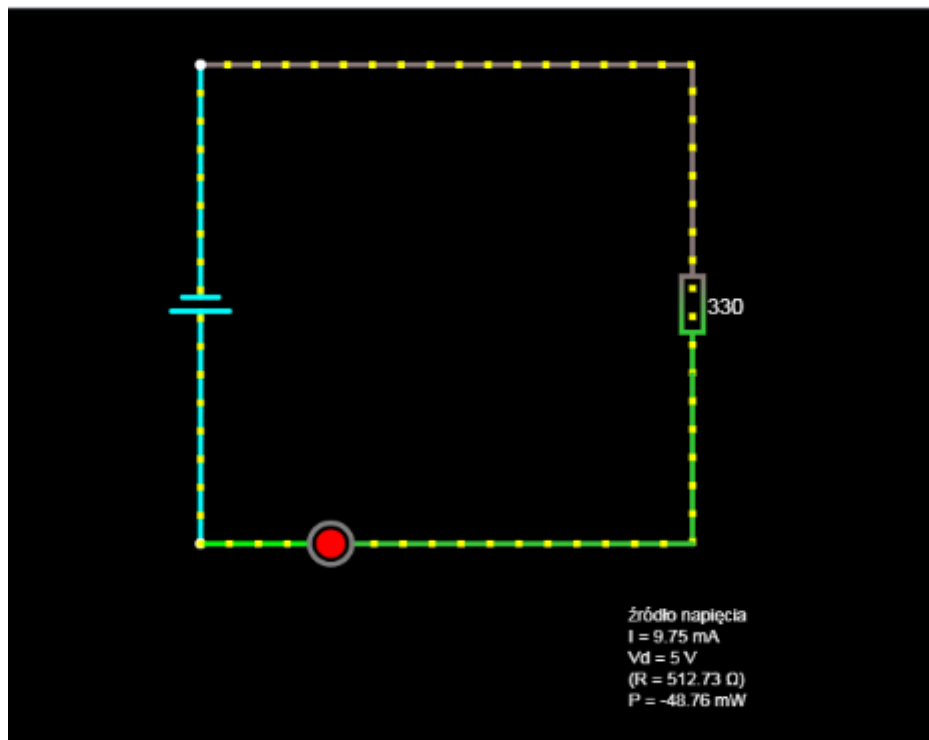
Numer ćwiczenia:3	Temat: The Digital Oscilloscope oraz Sensors and the PT Microcontroller	Grupa: 3ID15A Michał Ortyl Konrad Nowakowski
Ocena:		Data wykonania ćwiczenia: 20.11.2018

1. The Digital Oscilloscope

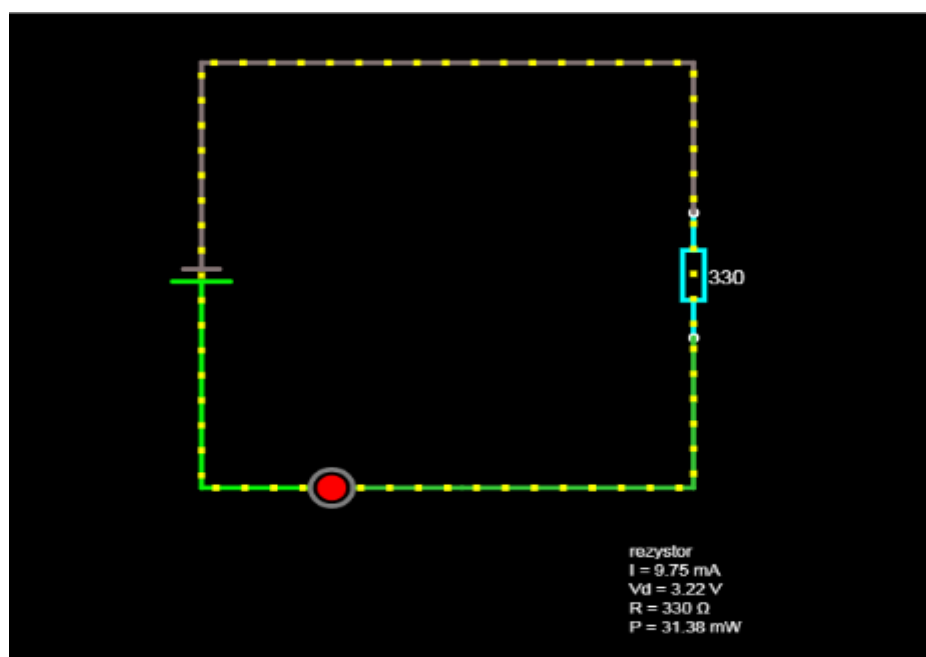
a) Cel ćwiczenia.

Należy użyć narzędzia Web Circuit Simulation Tool, zapoznać się z obwodem i podstawowymi prawami i elementami elektorniki.

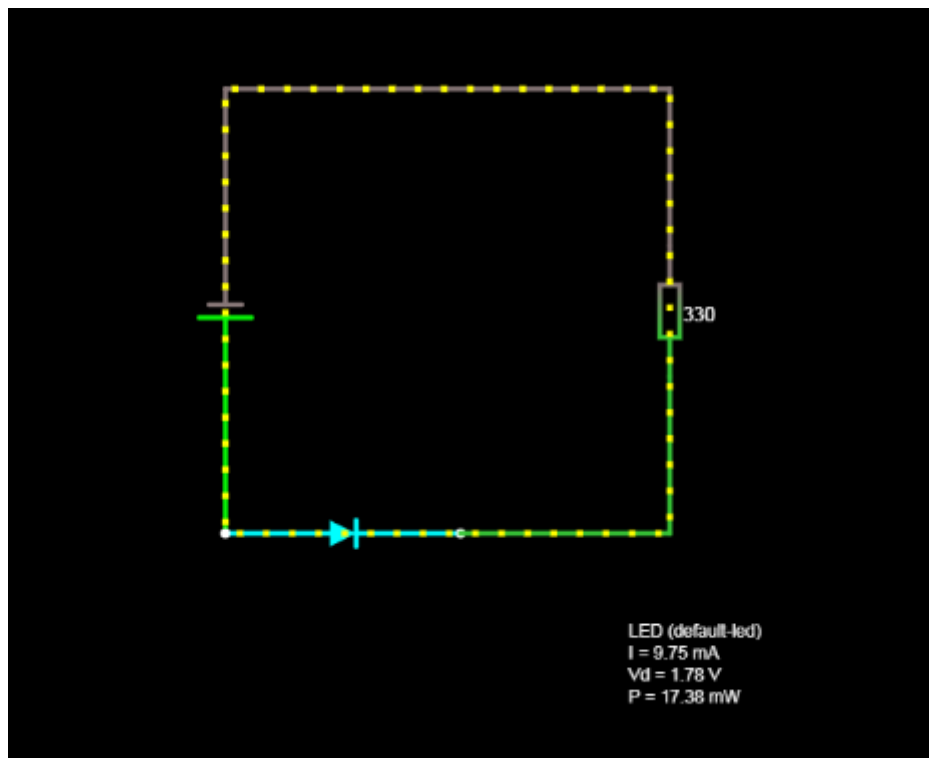
b) Topologia



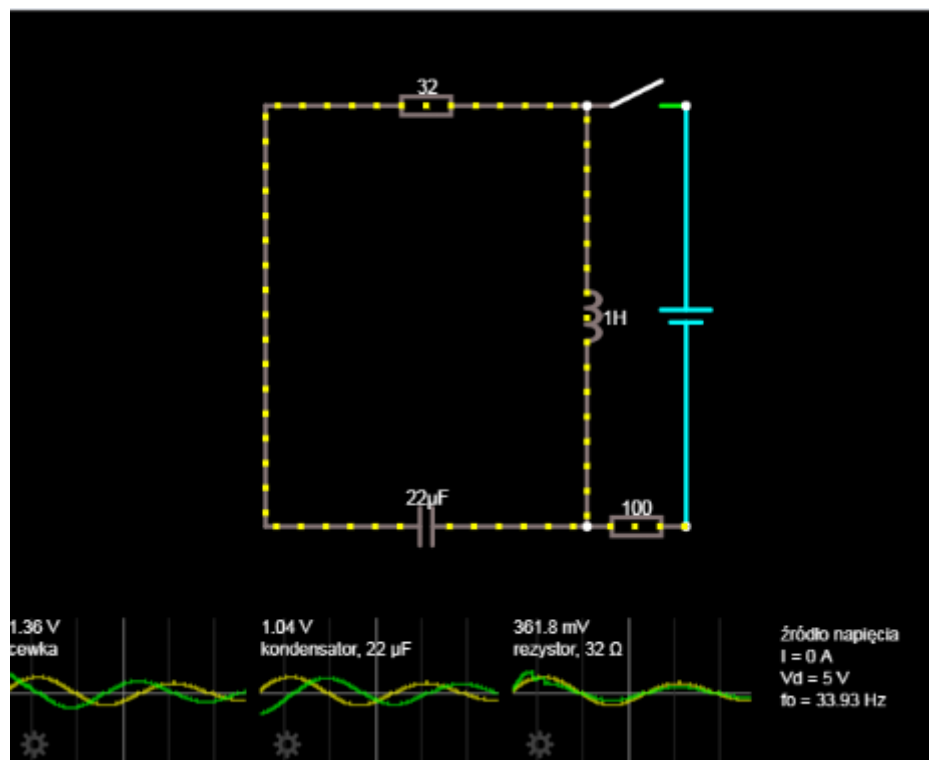
c) Rezystor i jego wartości.



d) Dioda LED i jej wartości.



e) Obwód RLC.

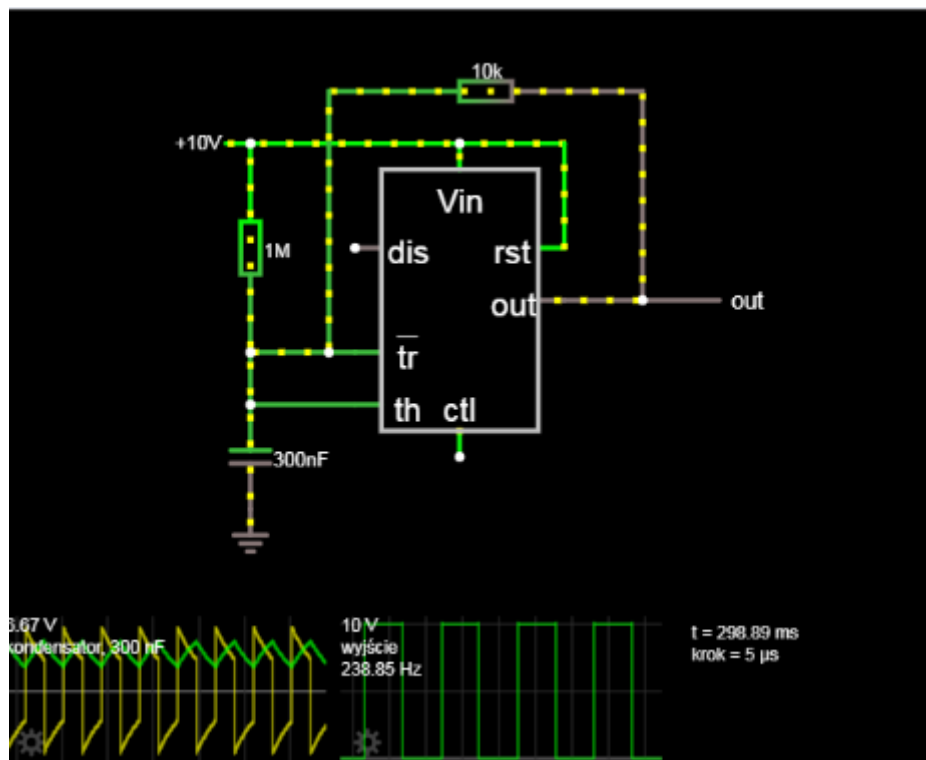


RLC – skrótowe oznaczenie dla obwodów elektrycznych (w tym elektronicznych) składających się tylko z trzech podstawowych elementów pasywnych:

- rezystora, oznaczanego przez R (rezystancja)
- cewki, oznaczanej przez L (indukcyjność)
- kondensatorów, oznaczanych przez C (pojemność)

Na wykresach pod obwodem możemy zauważyć przebiegi sinusoidalne, powstają one ponieważ, obwód jest tak skonstruowany i zmusza elektrony do zmiany kierunku przepływu tworząc jednocześnie naprzemienne poziomy sygnałów.

d) Square Wave(generator fal prostokątnych)



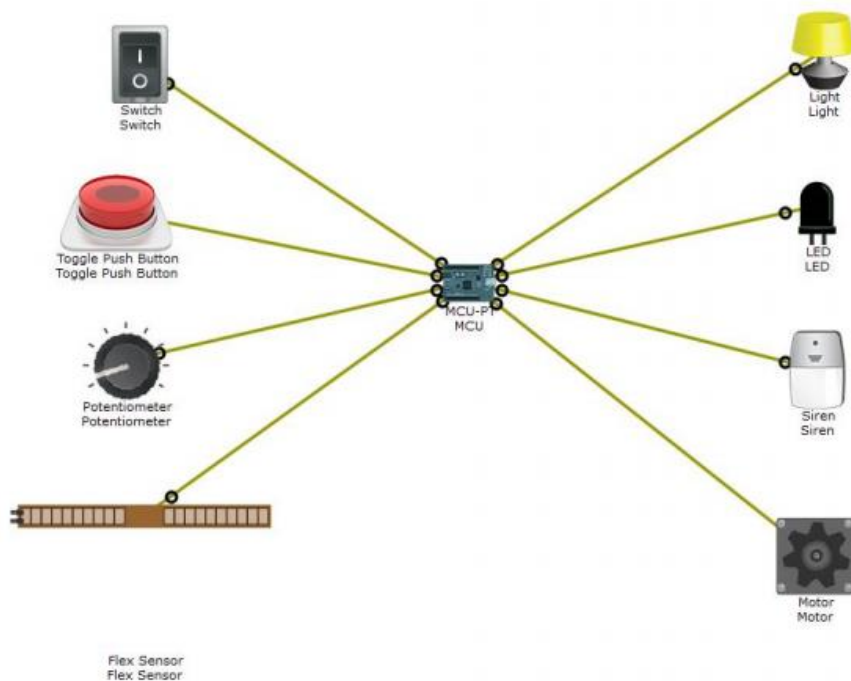
Takie przebiegi kwadratowe najczęściej widzimy w obwodach cyfrowych, w których występują dwie wartości (0,1). Takie układy nazywamy układami dyskretnymi.

2. Sensors and the PT Microcontroller

a) Cel ćwiczenia.

Poznania działania mikrokontrolera, zapoznanie się z topologia i zmiana kodu źródłowego.

b) Topologia



Topologia jest zbudowana w taki sposób, że na środku znajduje się MCU, po lewej stronie mamy urządzenia wejścia, po prawej urządzenia wyjścia które są sterowane/włączane poprzez urządzenia wejścia.

Określenie mikrokontroler pochodzi od głównego obszaru jego zastosowań, jakim jest sterowanie urządzeniami elektronicznymi, takimi jak: urządzenia biurowe, urządzenia medyczne (w tym implanty), zdalnego sterowania, elektronarzędzia, systemy sterowania silnikami samochodowymi, a nawet zabawki i inne systemy wbudowane.

c) Zmiany w kodzie źródłowym.

W dostarczonym kodzie lampa jest wyłączana i wyłączana poprzez przełącznik(off/on), dioda LED jest kontrolowana poprzez guzik. Naszym zadaniem jest zmiana, dioda LED ma być kontrolowana poprzez przełącznik a lampka poprzez guzik.

Funkcja zarządzająca przed zmianą:

```
20 def writeToActuators():
21     if (switchValue == HIGH): # evaluates to True if the Switch sensor value is digital HIGH, otherwise false
22         customWrite(2, "2") # turn on the Light
23     else:
24         customWrite(2, "0") # turn off the Light
25
26     if (togglePushButtonValue == HIGH): # evaluates to True if the Toggle Push Button sensor value is digital HIGH, otherwise false
27         digitalWrite(3, HIGH) # turn on the LED
28     else:
29         digitalWrite(3, LOW) # turn off the LED
30
31     if (potentiometerValue > 512): # evaluates to True if the Potentiometer is turned at least half way
32         customWrite(4, HIGH) # turn on the Siren
33     else:
34         customWrite(4, LOW) # turn off the Siren
35
36     if (flexSensorValue > 0): # evaluates to True if the Flex Sensor is bent, otherwise false
37         analogWrite(5, flexSensorValue) # turn on the motor with speed equal to the Flex Sensor value
38     else:
39         analogWrite(5, 0) # turn off the motor
40
```

Funkcja zarządzająca po zmianie:

```
20 def writeToActuators():
21     if (switchValue == HIGH): # evaluates to True if the Switch sensor value is digital HIGH, otherwise false
22         digitalWrite(3, HIGH) # turn on the Light
23     else:
24         digitalWrite(3, LOW) # turn off the Light
25
26     if (togglePushButtonValue == HIGH): # evaluates to True if the Toggle Push Button sensor value is digital HIGH, otherwise false
27         customWrite(2, "2") # turn on the LED
28     else:
29         customWrite(2, "0") # turn off the LED
30
31     if (potentiometerValue > 512): # evaluates to True if the Potentiometer is turned at least half way
32         customWrite(4, HIGH) # turn on the Siren
33     else:
34         customWrite(4, LOW) # turn off the Siren
35
36     if (flexSensorValue > 0): # evaluates to True if the Flex Sensor is bent, otherwise false
37         analogWrite(5, flexSensorValue) # turn on the motor with speed equal to the Flex Sensor value
38     else:
39         analogWrite(5, 0) # turn off the motor
40
```