

Politechnika Świętokrzyska w Kielcach
Wydział Elektrotechniki, Automatyki i Informatyki

Laboratorium 3
The Digital Oscilloscope
Packet Tracer - Sensors and the PT Microcontroller

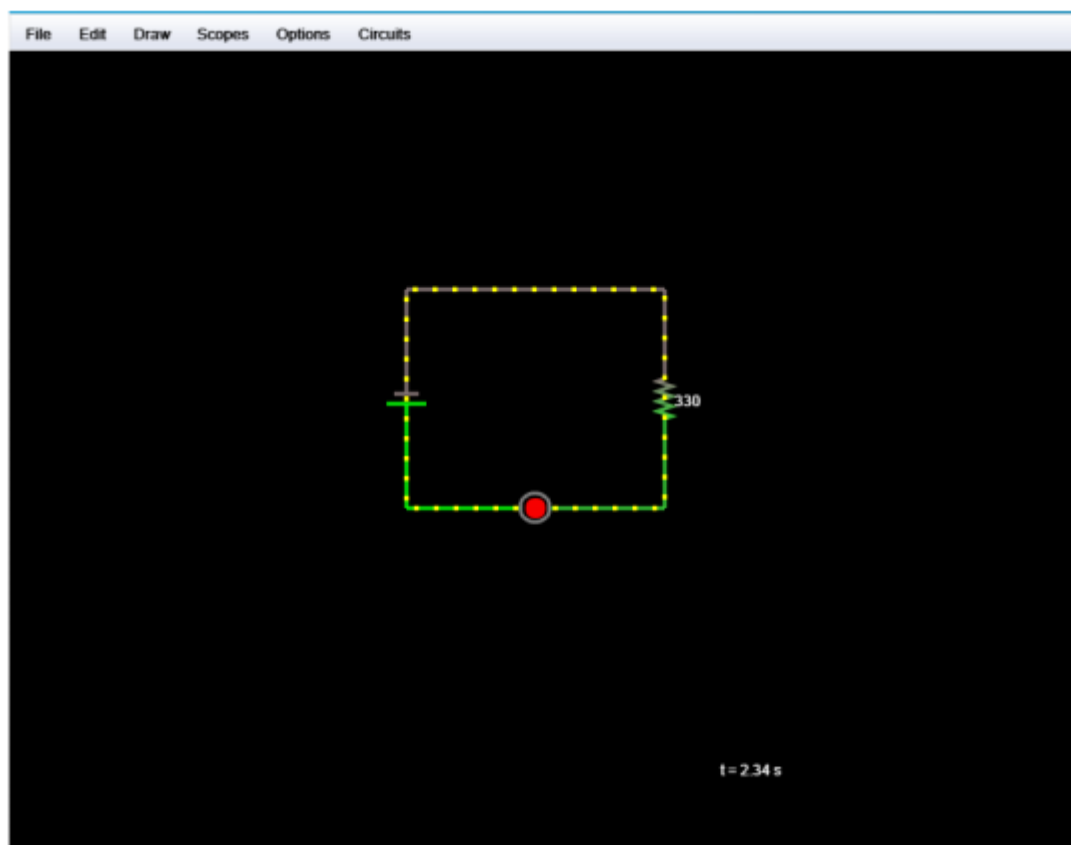
Technologie IoT rozproszone sieci sensoryczne

Autorzy:
Adrian Śmigłarski
Patryk Tracz
Grupa: 3ID15A

1. Lab - The Digital Oscilloscope

a) The Circuit Simulator and Basic Circuits

Topologia



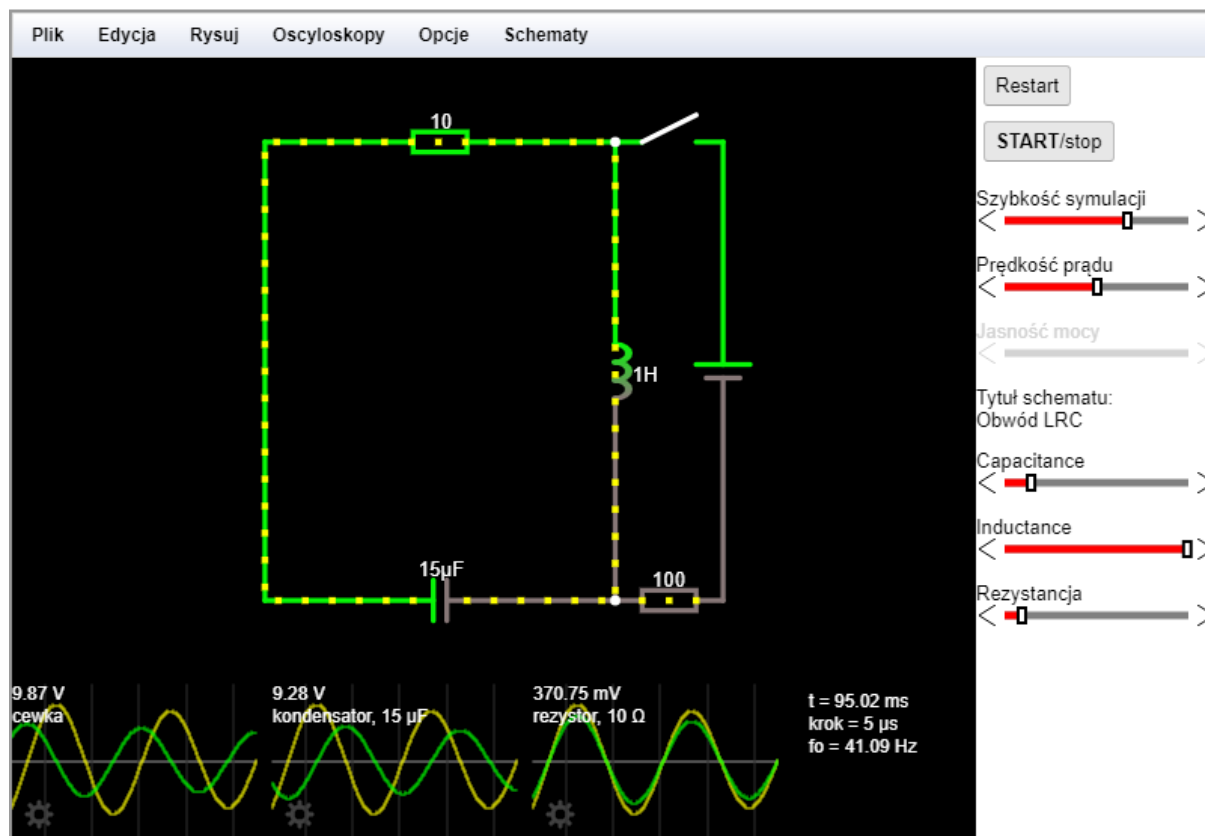
Wartości napięcia na poszczególnych elementach obwodu:

LED- 1.78V

rezystor- 3.22V

bateria- 5V

b) Visualizing Signals



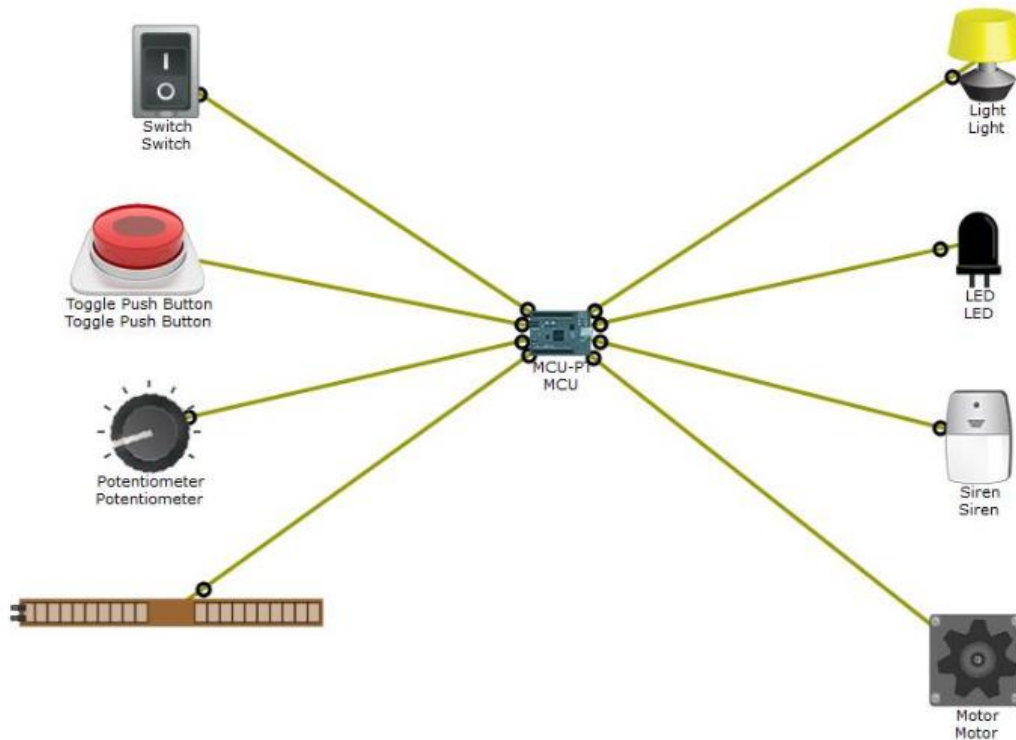
Challenge question: Alternating Current (AC) creates square waves or sine waves? Explain.

AC – prąd przemienny tworzy fale sinusoidalne ponieważ w tym przypadku wartości chwilowe podlegają zmianom w powtarzalny, okresowy sposób, z określoną częstotliwością.

Największe znaczenie praktyczne mają prąd i napięcie o **przebiegu sinusoidalnym**. W żargonie technicznym nazwa prąd przemienny często oznacza po prostu prąd sinusoidalny. Jeśli zakłócenia lub nieliniowość powodują zdeformowanie sinusoidalnego kształtu, wówczas taki niesinusoidalny przebieg nosi nazwę przebiegu odkształconego.

2. Packet Tracer - Sensors and the PT Microcontroller

Topologia:



W celu zmiany działania układu tak aby to przełącznik kontrolował diodę LED a przycisk światło, musieliśmy zmodyfikować domyślny kod. Wystarczyło zmienić dwie linijki kodu:

Przed zmianami:

```
13 global flexSensorValue # declare flexSensorValue as global
14
15 switchValue = digitalRead(0) # read Switch sensor value
16 togglePushButtonValue = digitalRead(1) # read Toggle Push Button sensor value
17 potentiometerValue = analogRead(A0) # read Potentiometer sensor value
18 flexSensorValue = analogRead(A1) # read Flex Sensor value
19
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
41 def main(): # defines the main function
```

Po zmianach:

Reload Copy Paste Undo Redo Find Replace Zoom: +

```
13 global flexSensorValue # declare flexSensorValue as global
14
15 switchValue = digitalRead(0) # read Switch sensor value
16 togglePushButtonValue = digitalRead(1) # read Toggle Push Button sensor value
17 potentiometerValue = analogRead(A0) # read Potentiometer sensor value
18 flexSensorValue = analogRead(A1) # read Flex Sensor value
19
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 LED
23     else:
24         digitalWrite(3, LOW) # turn off the LED
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 Light
28     else:
29         customWrite(2, "0") # turn off the Light
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
41 def main(): # defines the main function
```

3. Wnioski

To zadanie zapoznało nas z podstawowymi elementami i pojęciami dotyczącymi obwodów elektrycznych. Dzięki stosowaniu narzędzi symulacyjnych takich jak Falstad's Circuit Simulator Tool możemy badać przepływ prądu w obwodzie, oraz np. jak zmiany napięcia wpływają na poszczególne elementy obwodu.

Zadanie w Packet Tracer pokazało nam jak MCU może kontrolować działanie elementów układu IoT i jak ważne jest sprawne orientowanie się w kodzie w celu wprowadzenia pożądaných zmian.