

POLITECHNIKA ŚWIĘTOKRZYSKA		
IoT laboratorium 3		
Kierunek: Informatyka	Rok: IV	Semestr: VII
Wykonał: Krzysztof Siwoń		

I

1. Wstęp

Głównym celem ćwiczenia było stworzenie symulacji schematu sterowanego przez moduł MCU.

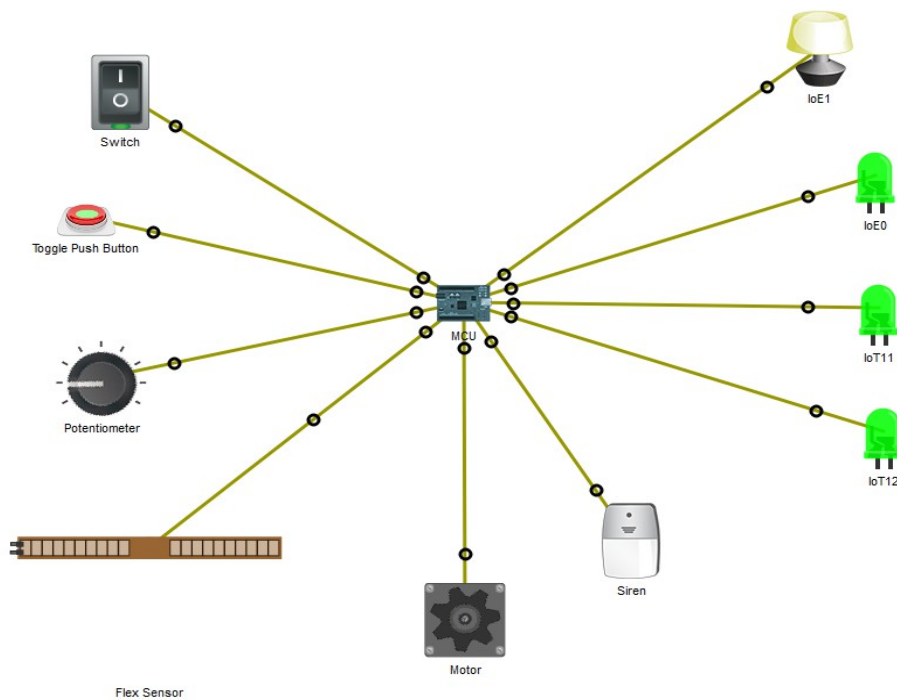
MCU - MicroController Unit - scalony system mikroprocesorowy, zrealizowany w postaci pojedynczego układu scalonego zawierającego jednostkę centralną (CPU), pamięć RAM oraz rozbudowane układy wejścia-wyjścia i na ogół pamięć programu jako FRAM, MRAM, ROM lub Flash.

Schemat tworzony był w programie PacketTracer

2. Przebieg ćwiczenia:

- Do kontrolera MCU podłączyliśmy:
- Przełącznik, sterujący światłem
- Włącznik, kontrolujący LEDy
- Potencjometr, ustawiający głośność Syreny
- Flex Sensor, sterujący motor,
- Silnik (motor)
- Syrena
- Diody LED
- Lampa (light)

W celu interakcji z danym elementem, należy nacisnąć na klawiaturze Alt+click.



Do połączenia elementów użyliśmy IoT Custom Cable.

Następnie należało zaprogramować kontroler MCU aby cały układ spełniał założenia.

```
from gpio import * # imports all modules in the GPIO library
from time import * # imports all modules in the time library

switchValue = 0 # initialize Switch sensor value global variable to 0
togglePushButtonValue = 0 # initialize Toggle Push Button sensor value global variable to 0
potentiometerValue = 0 # initialize Potentiometer sensor value global variable to 0
flexSensorValue = 0 # initialize Flex Sensor value global variable to 0

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

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

def writeToActuators():
```

```

    if (switchValue == HIGH): # evaluates to True if the Switch sensor value is digital HIGH,
otherwise false
        customWrite(2, "2") # turn on the Light
    else:
        customWrite(2, "0") # turn off the Light

    if (togglePushButtonValue == HIGH): # evaluates to True if the Toggle Push Button sensor
value is digital HIGH, otherwise false
        digitalWrite(3, HIGH)

    else:
        digitalWrite(3, LOW) # turn off the LED

    if (potentiometerValue > 512): # evaluates to True if the Potentiometer is turned at least half
way
        customWrite(4, HIGH) # turn on the Siren
    else:
        customWrite(4, LOW) # turn off the Siren

    if (flexSensorValue > 0): # evaluates to True if the Flex Sensor is bent, otherwise false
        analogWrite(5, flexSensorValue) # turn on the motor with speed equal to the Flex
Sensor value
    else:
        analogWrite(5, 0) # turn off the motor

    if(togglePushButtonValue ==HIGH):
        digitalWrite(9, HIGH)
    else:
        digitalWrite(9,LOW)

    if(togglePushButtonValue ==HIGH):
        digitalWrite(8, HIGH)
    else:
        digitalWrite(8,LOW)

def main(): # defines the main function
    pinMode(0, IN) # sets digital slot 0 (Switch) to input
    pinMode(1, IN) # sets digital slot 1 (Toggle Push Button) 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 (LED) to output

    while True: # loop indefinitely
        readFromSensors() # call the readFromSensors function
        writeToActuators() # call the writeToActuators function
        delay(1000) # delay script execution for 1000 ms

if __name__ == "__main__": # Evaluates to True if this module is the script being executed,
otherwise False if this module is being imported into another module
    main() # call the main function

```

Założenie zadania przewidywało sterowanie 8diodami LED.
Udało mi się wykonać w/w zadanie tylko dla 3 diód.

3. Wnioski

Dzięki programowi PacketTracer poznaliśmy zasady działania poszczególnych kontrolerów, napisaliśmy skrypt, który sterował poszczególnymi modułami. Pokazało to, jaki potencjał istnieje w IoT, jak sami możemy zbudować i kontrolować elementy obsługujące np. SmartDom.