

Politechnika Świętokrzyska

w Kielcach

Wydział Elektrotechniki, Automatyki i Informatyki

Sprawozdanie 2

Laboratorium Iot

Gałka Karolina

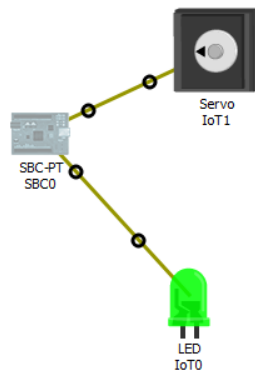
Sałata Mateusz

3ID15A

20.11.2018r.

Zadanie 1 Packet Tracer – Simulating IoT Devices

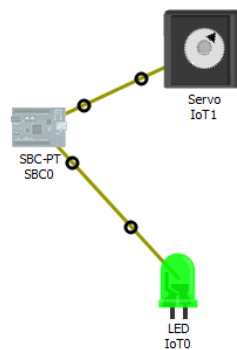
Naszym zadaniem było wykonanie topologii na rysunku poniżej.



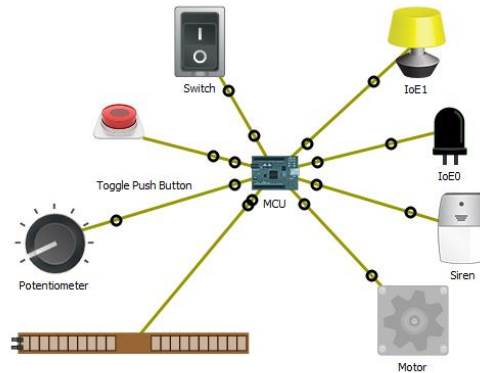
Następnie musieliśmy zmodyfikować plik main.py dopisując dwie linie:

```
customWrite(0, 127);  
customWrite(0, -127);
```

```
Blink (Python) - main.py  
Open New Delete Rename Import  
Reload Copy Paste Undo Redo Find Replace Zoom: + -  
Install to Desktop Stop Clear Outputs Help  
..main.py  
1 from gpio import *  
2 from time import *  
3  
4 def main():  
5     pinMode(1, OUT)  
6     print("Blinking")  
7     while True:  
8         digitalWrite(1, HIGH);  
9         customWrite(0, 127);  
10        delay(1000)  
11        digitalWrite(1, LOW);  
12        customWrite(0, -127);  
13        delay(500)  
14  
15 if __name__ == "__main__":  
16     main()  
17
```



Zadanie 2 Sensors and the PT Microcontroller in Packet Tracer 7.1



Naszym zadaniem było przeanalizowanie zasady działania tej topologii. Po lewej stronie znajdują się urządzenia wejściowe, natomiast po prawej wyjściowe. Każde z urządzeń wejściowych oddziałuje na inne urządzenie wyjściowe.

```
MCU (Python) - main.py
Open New Delete Rename Import
main.py
1 from gpio import * # imports all modules in the GPIO library
2 from time import * # imports all modules in the time library
3
4 switchValue = 0 # initialise Switch sensor value global variable to 0
5 togglePushButtonValue = 0 # initialise Toggle Push Button sensor value global variable to 0
6 potentiometerValue = 0 # initialise Potentiometer sensor value global variable to 0
7 flexSensorValue = 0 # initialise Flex Sensor value global variable to 0
8
9 def readFromSensors():
10     global switchValue # declare switchValue as global
11     global togglePushButtonValue # declare togglePushButtonValue as global
12     global potentiometerValue # declare potentiometerValue as global
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
42     pinMode(0, IN) # sets digital slot 0 (Switch) to input
43     pinMode(1, IN) # sets digital slot 1 (Toggle Push Button) to input
44     pinMode(2, OUT) # sets digital slot 2 (Light) to output
45     pinMode(3, OUT) # sets digital slot 3 (LED) to output
46     pinMode(4, OUT) # sets digital slot 4 (Siren) to output
47
48     while True: # loop indefinitely
49         readFromSensors() # call the readFromSensors function
50         writeToActuators() # call the writeToActuators function
51         delay(1000) # delay script execution for 1000 ms
52
53 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
54     main() # call the main function
```

Plik main.py zawiera instrukcje odpowiadające ze właściwe działanie tych urządzeń.