

# Politechnika Świętokrzyska w Kielcach

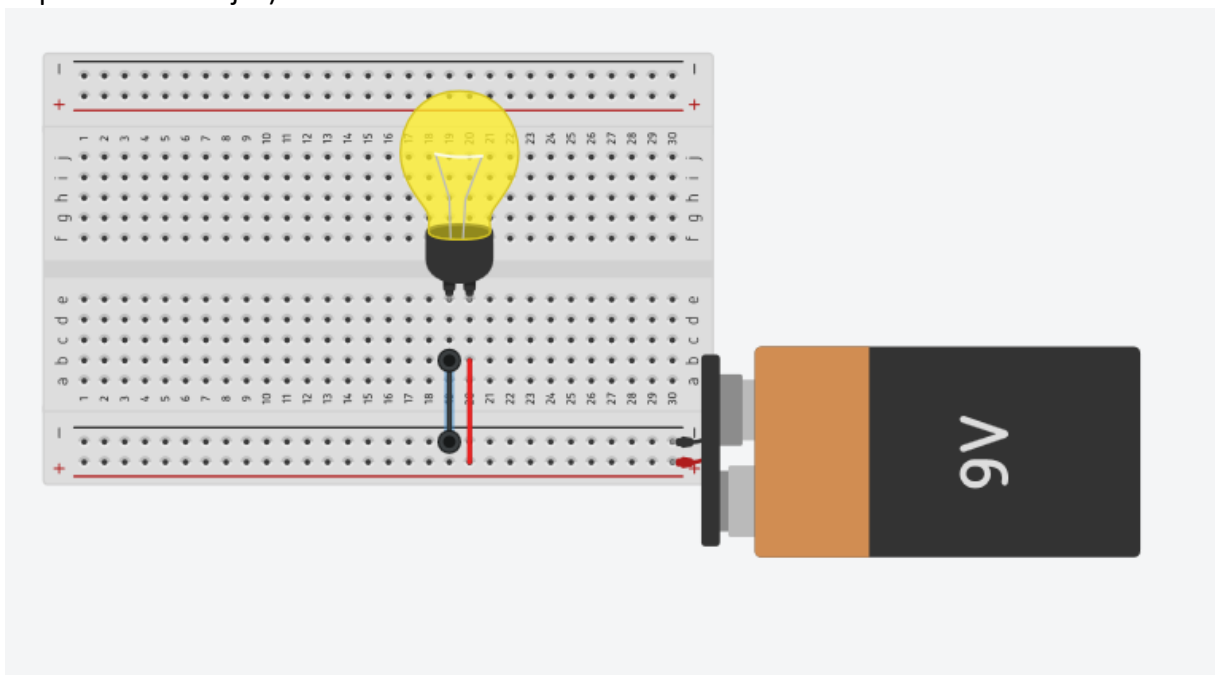
## Wydział Elektroniki, Automatyki i Informatyki

### Technologie IoT rozproszone sieci sensoryczne

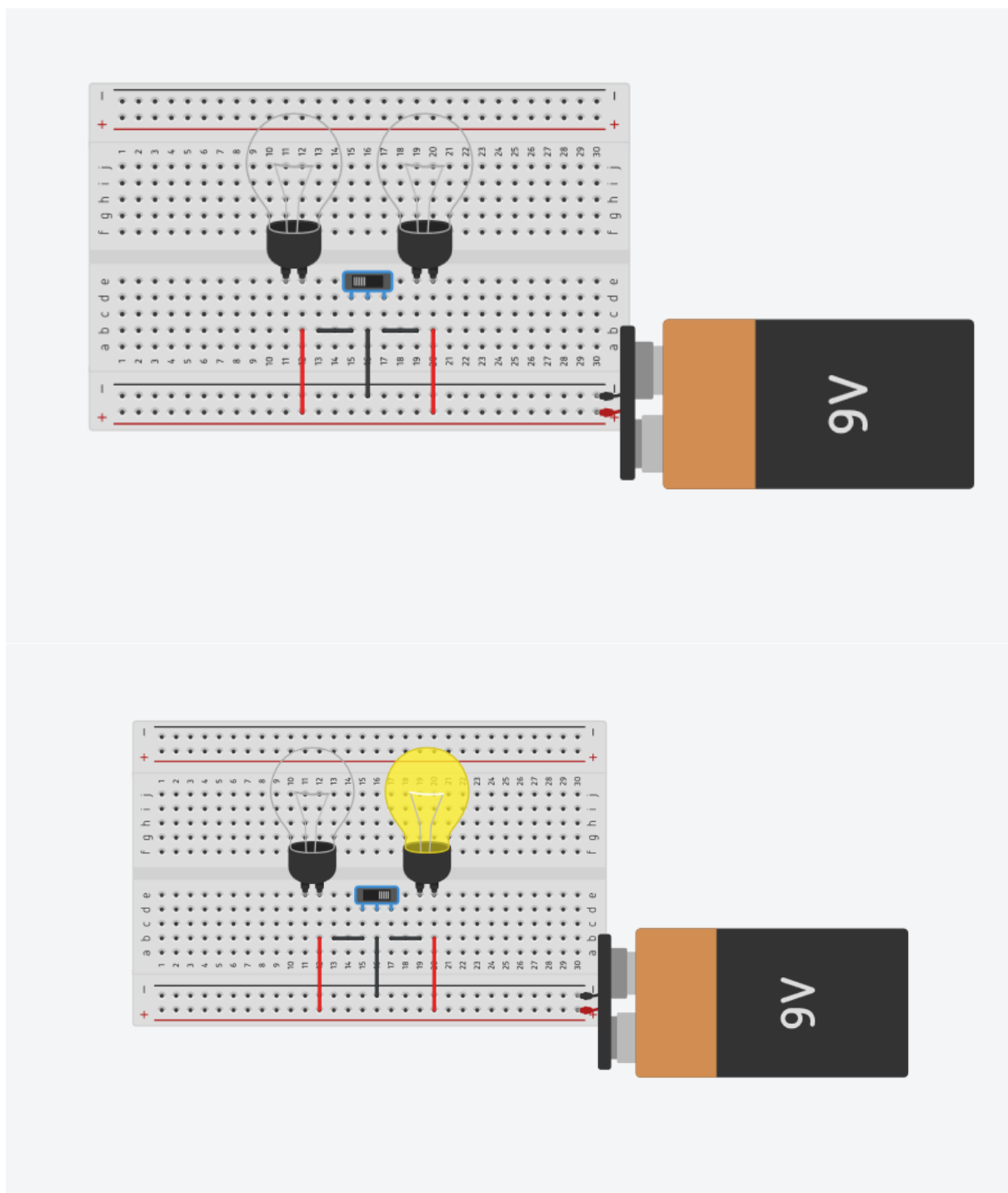
Temat: Designing a Circuit from Start to Finish and Packet Tracer – Simulating IoT Devices

Grupa: **3ID15A**  
Konrad Nowakowski  
Michał Ortyl

1. Jako pierwsze zadanie mieliśmy utworzyć układ składających się z płytki arduino, baterii 9v, żarówki oraz kabli. Układ wyglądał tak (już pod podłączeniu kabli w odpowiednie miejsca) :



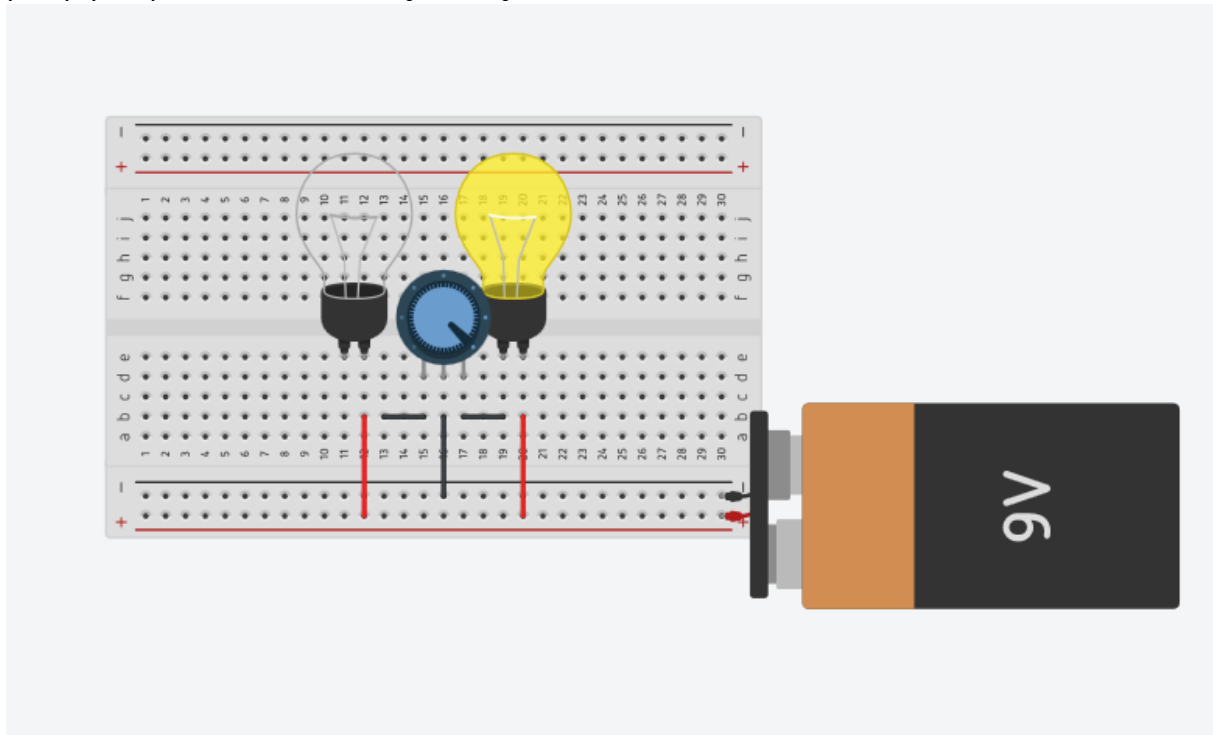
- 1.2 Następnym celem było zmodyfikowanie układu tak aby układ zawierał 2 żarówki oraz przełącznik usuwając przy tym jeden kabel „minusowy” tak wyglądał układ z wyłączonym (żarówka się nie świeci) przełącznikiem oraz włączonym (żarówka się świeci) :



### **Wnioski do części pierwszej :**

Na laboratorium nauczyliśmy się symulować proste układy arduino, łączyć różne elementy i sprawdzać jak zachowuje się układ. Na sam koniec zostaliśmy zapytani: „What would happen if a potentiometer replaced the slide switch in the drawing?” odpowiedź znajduje się na poniższym obrazku przy czym potencjometrem możemy kontrolować

przepływ np. tak ze żarówka będzie się lekko mienić.

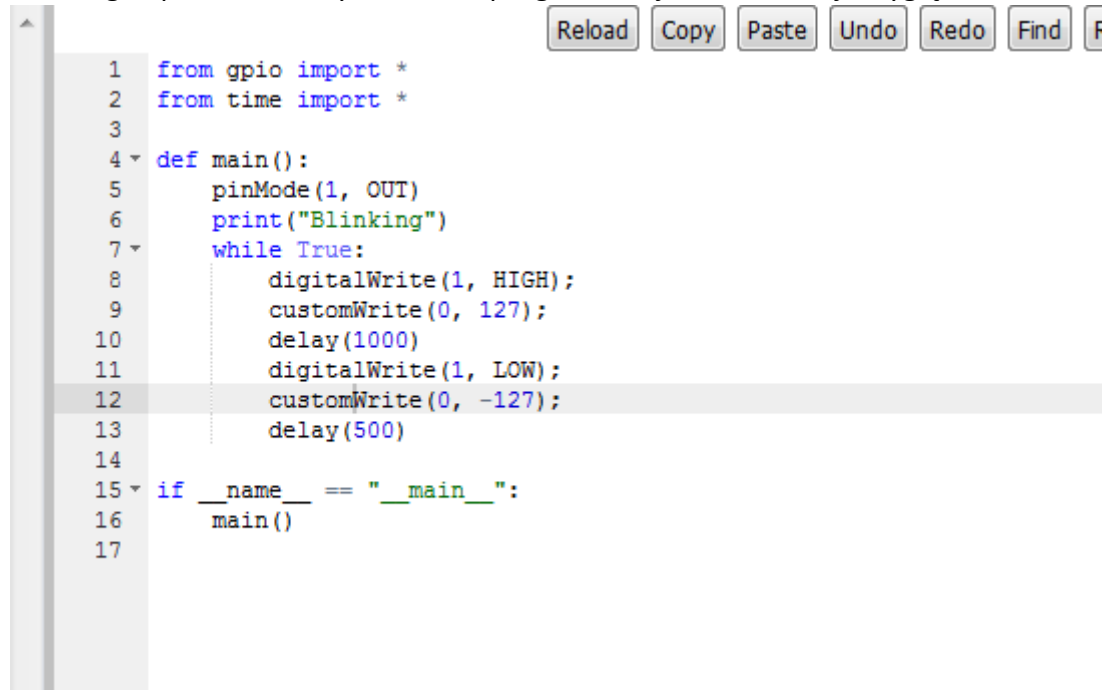


2 W drugiej części mieliśmy zbudować układ w PT oraz pierwsze jakie polecenie było to zmodyfikowanie kodu w jaki sposób chcieliśmy. U nas to wyglądało tak:

```
an New Delete Rename Import Install to Desktop
Reload Copy Paste Undo Redo
n.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         delay(1000)
10        digitalWrite(1, LOW);
11        delay(5000)
12
13 if __name__ == "__main__":
14     main()
15
```

Program został zmieniony tak ze żarówka rzadziej świeci.

2.1 Drugim poleceniem było zmienić program tak jak w instrukcji, wyglądało to tak:



```
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
```

Program ten działa teraz tak że „servo” będzie się teraz włączał i w tym samym momencie dioda led będzie się zapalać.

### Wnioski (do części drugiej):

Na laboratorium nauczyliśmy się modyfikować kod programu tak żeby urządzenie się zachowywało tak jak tego potrzebujemy. Zostaliśmy również zapytani " What could be changed to make the servo turn in the opposite direction while the LED is blinking ?" odpowiedź znajduje się na poniższym obrazku:



```
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
```