

POLITECHNIKA ŚWIĘTOKRZYSKA		
IoT – Laboratorium 2		
Kierunek: Informatyka	Rok: 4	Semestr: VII
Student: Krzysztof Siwoń		Data wykonania: 23.11.2018
Temat: Symulacja urządzeń IoT		

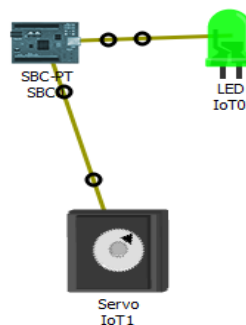
1. Cel laboratorium

W pierwszej części ćwiczenia mieliśmy za zadanie wykorzystać Packet Tracer do zbudowania prostego układu składającego się z SBC, diody LED oraz silniczka Servo. Następnie należało przeprogramować płytkę SBC w języku Python, aby wymusić konkretne działanie servo i diody.

W drugiej części zadania korzystaliśmy ze strony www.tinkercad.com, umożliwiającej symulowanie układów elektronicznych. Do wykonania mieliśmy od podstaw układ składający się z płytki prototypowej, baterii 9V i żarówki, przesymulować jego działanie, a następnie modyfikować go w różny sposób dodając elementy takie jak dodatkowe okablowanie, kolejną żarówkę, przekaźnik, czy też potencjometr.

2. Wykonanie zadania

a) Zadanie w Packet Tracer wyglądało następująco:



Oprogramowanie płytki:

```

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(500)
12
13  if __name__ == "__main__":
14      main()

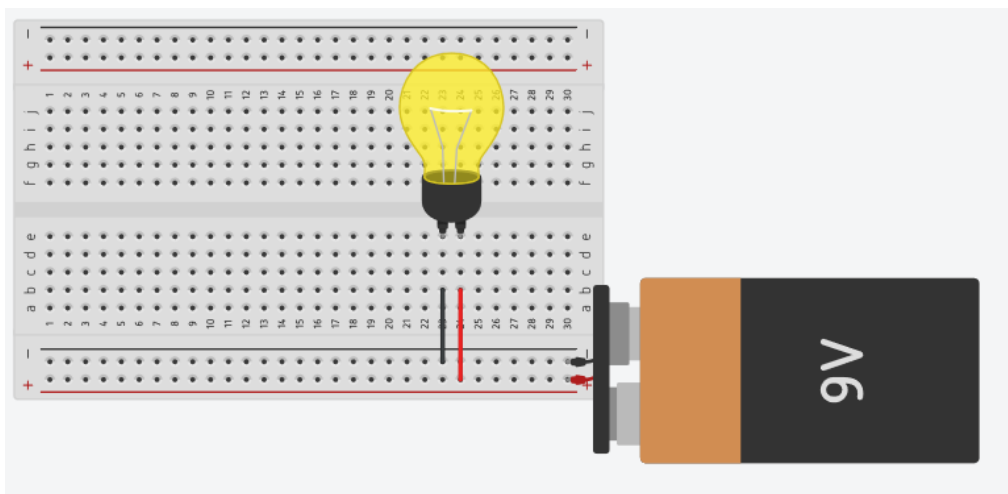
```

Uruchomienie skryptu powodowało to, że dioda włączała się na 1000 ms i następnie przebywała w stanie wyłączenia przez 500 ms. Pętla while wymuszała takie działanie naprzemiennie, aż do momentu wyłączenia skryptu. Naszym zadaniem było zmodyfikować w taki oto sposób:

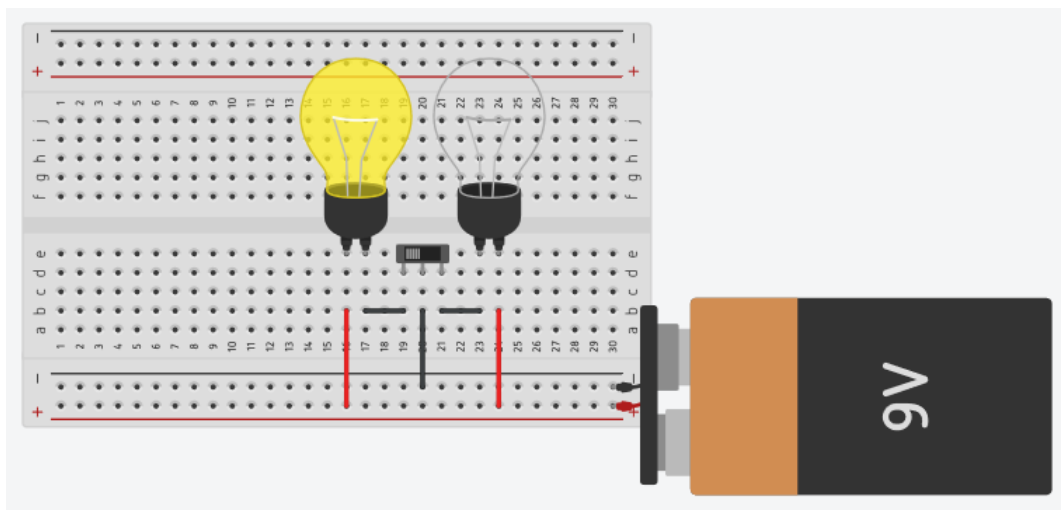
```
1  from gpio import *
2  from time import *
3
4  def main():
5      pinMode(1, OUT)
6      print("Blinking")
7      while True:
8          digitalWrite(1, HIGH);
9          customWrite(0, 127);
10         delay(1000)
11         digitalWrite(1, LOW);
12         customWrite(0, -127);
13         delay(500)
14
15  if __name__ == "__main__":
16      main()
17
```

Dodanie dwóch linijek do kodu aktywowało servo, który działał synchronicznie z diodą LED. Zakres kąta o jaki obracał się servo wynosił domyślnie 1-160 stopni. W celu zmiany tego zakresu, a jednocześnie spowodowania żeby np. obracał się o 180 stopni, prawdopodobnie należałoby zmienić wartość maksymalnego kąta(max) w oprogramowaniu tego urządzenia. Sterując nim z SBC można było maksymalnie osiągnąć wartość 160 niezależnie jak duży był drugi argument funkcji customWrite.

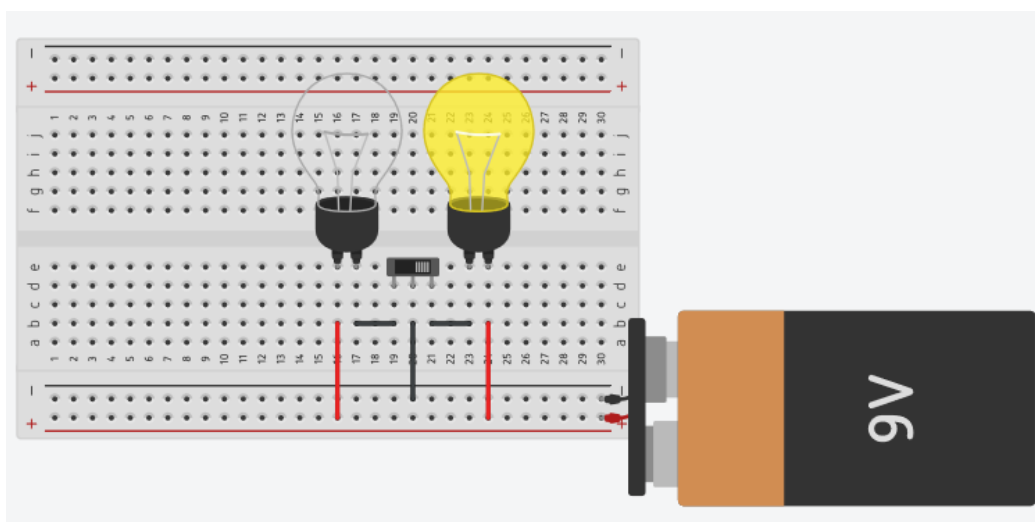
b) W zadaniu na stronie tinkercad w pierwszej fazie należało podłączyć do płytki żarówkę i baterię 9V:



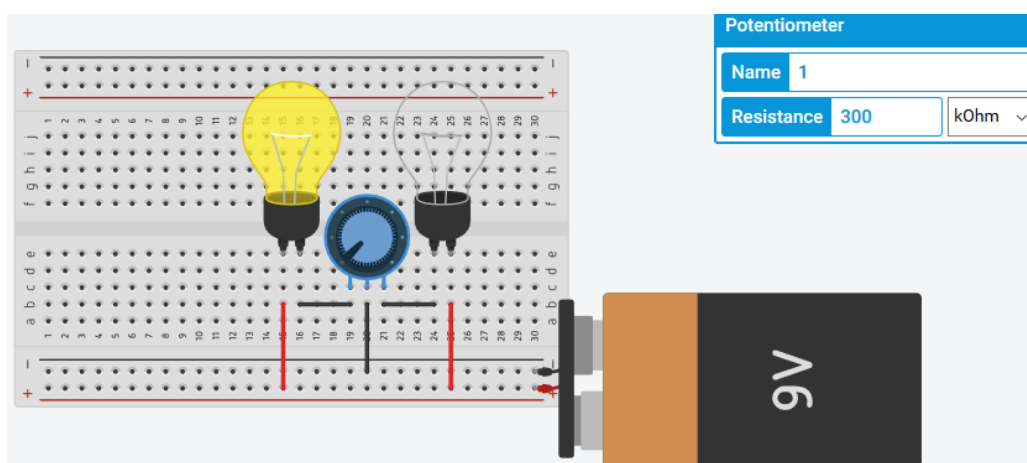
Jak widać po podłączeniu kabli(czerwonego oznaczającego +, czarnego -) i uruchomieniu symulacji, żarówka się zapaliła. Kolejnym etapem była modyfikacja układu poprzez dodanie przełącznika i jeszcze jednej żarówki oraz podłączenie ich okablowaniem:



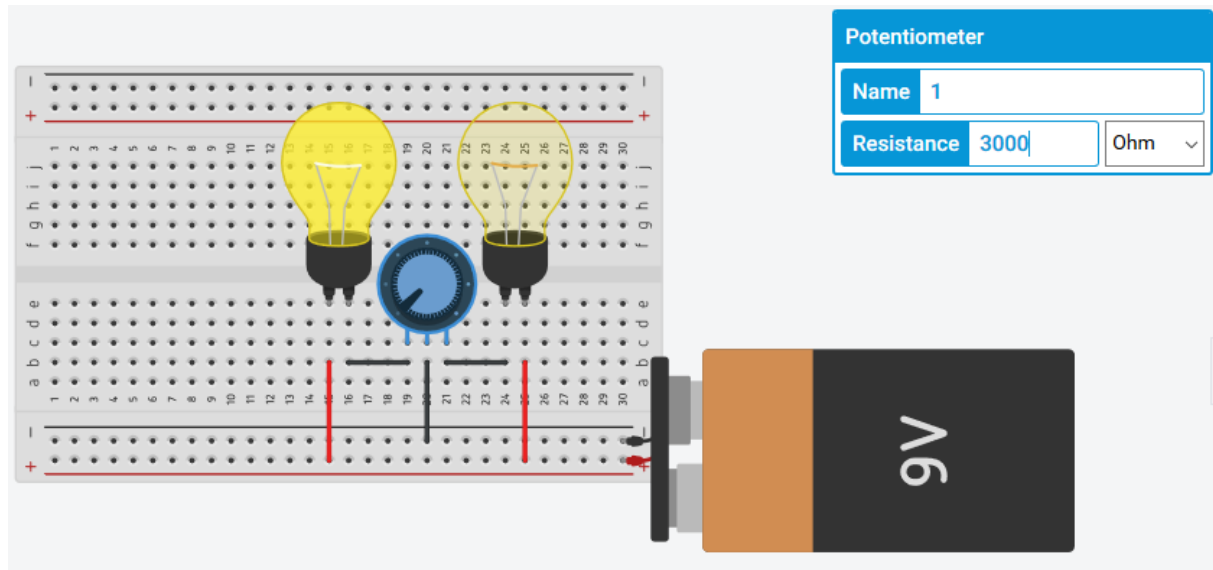
Pozycja przełącznika decydowała o przepływie prądu, jak wyżej widać pozycja w lewo zapalała lewą żarówkę, przełączenie go w prawo zapalało tę po prawej:



Zastąpienie slideswitch'a przez potencjometr skutkowało podobnym efektem, lecz należy zwrócić uwagę, że cały prąd był przekazywany do żarówki wówczas gdy regulator potencjometru był maksymalnie skręcony w prawo lub lewo i dodatkowo odbywało się to przy jego dużej rezystancji tj, w tym przypadku 300 kOhm:



Wraz ze spadkiem rezystancji potencjometru można było zauważyć, że natężenie prądu jest rozdzielane pomiędzy obie żarówki w stopniu zależnym od ustawienia pokrętła potencjometru:



Jak widać przy oporze 100 razy mniejszym potencjometr nie jest już w stanie pełnić roli przełącznika.