

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра ИИТ

Лабораторная работа №4

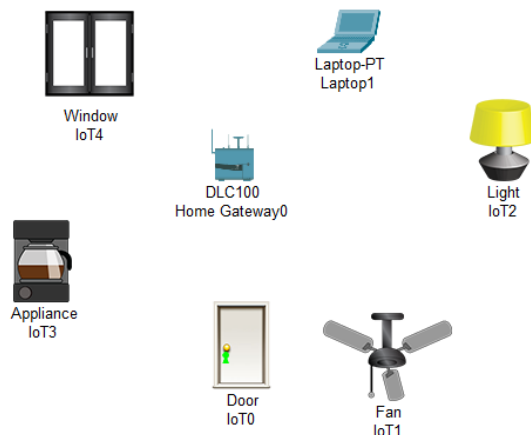
Выполнила:
студентка группы АС-51
Водянова Влада
Проверил:
Давидюк Ю.И.

Брест 2020

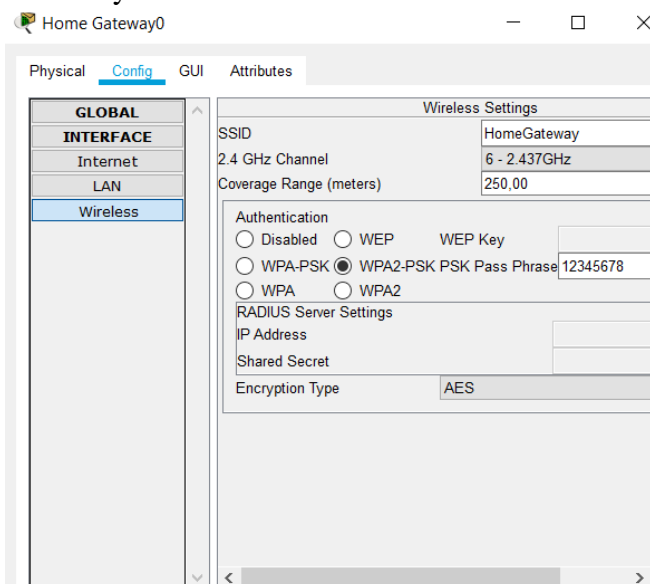
Цель работы: приобрести практические навыки проектирования инфраструктуры «умного дома», научиться основам программирования микроконтроллерных устройств.

Задание 1

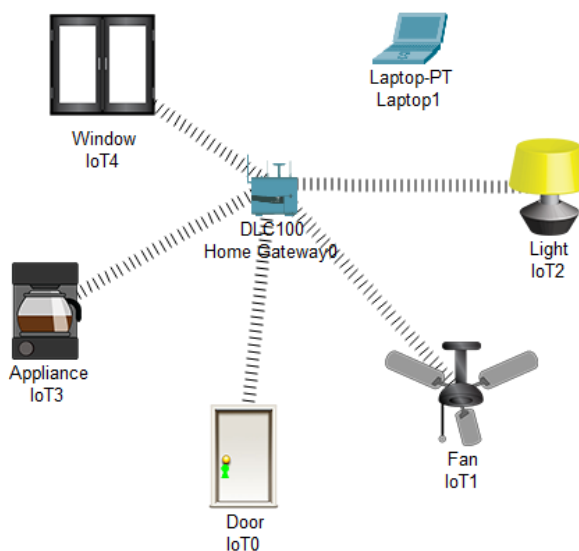
1) Разместить элементы, указанные в примере:



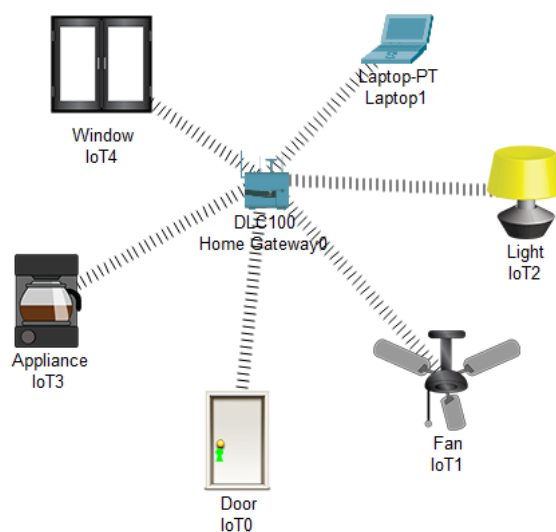
2) Настроить Home Gateway:



3) Настроить беспроводное подключение устройств IoT:



4) Настроить подключение ноутбука. Поэкспериментировать с кнопками включения/выключения устройств и изучите изменения, которые с ними происходят.



5) Добавить фон



Задание 2

1) Добавить микроконтроллерную плату

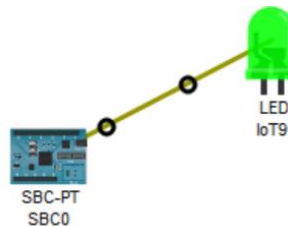


2) Изучить скрипт main.py

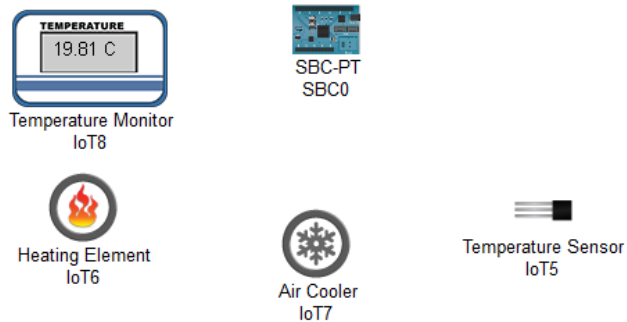
```
main.py
1 from gpio import *
2 from time import *
3
4 def main():
5     pinMode(1, OUT)
6     print("Blinking")
7     while True:
8         digitalWrite(1, HIGH);
9         delay(1000)
10        digitalWrite(1, LOW);
11        delay(500)
12
13 if __name__ == "__main__":
14     main()
15
```

Serial Outputs

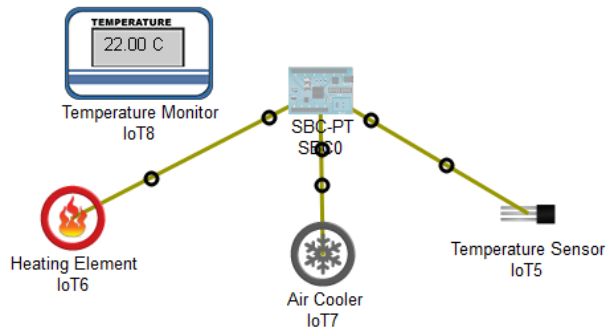
3) Добавить светодиод (LED). Открыть программу, попытайтесь изучить и понять ее содержимое:



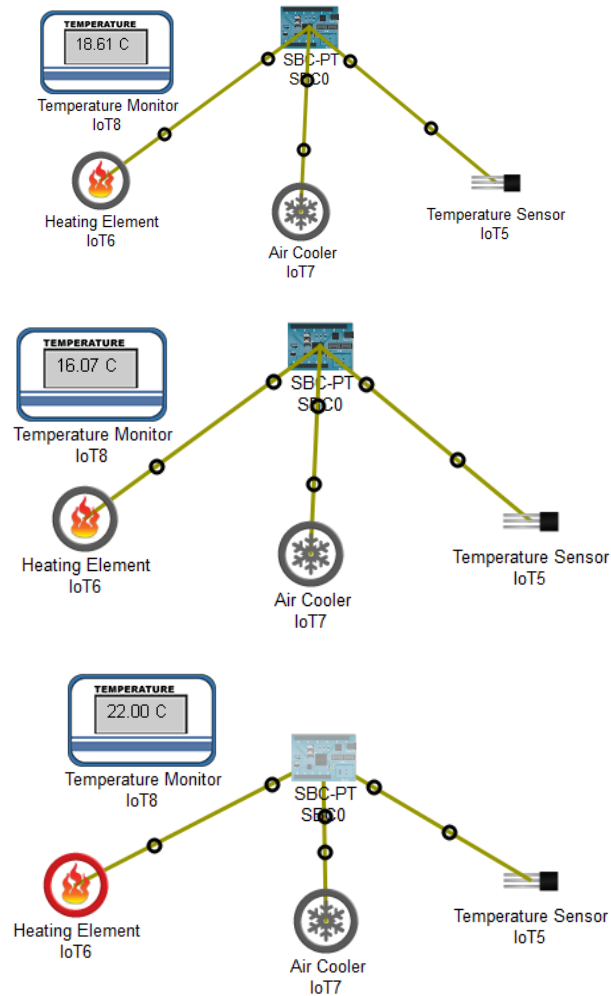
4) Удалить LED из рабочей области. Добавить другие компоненты, необходимые для реализации проекта.



5) Подсоединить все указанные датчики к плате, выбрав произвольные пины:



6) Изучить изменение температуры в течение суток:



7) Написать программу, которая будет поддерживать текущую температуру в заданном интервале.

The screenshot shows the SBCU programming interface with a Python script for temperature control. The script is titled "Blink (Python) - main.py" and is being edited in the "Programming" tab. The script uses the GPIO module to control the heating element and air cooler, and the time module to implement a delay. The temperature sensor is connected to pin 1, the heating element to pin 2, and the air cooler to pin 3. The script reads the temperature from the sensor and compares it to a setpoint range (20 to 25 degrees Celsius). If the temperature is below 20 degrees, the heating element is turned on. If the temperature is above 25 degrees, the air cooler is turned on. If the temperature is within the range, both are turned off. The script also prints the current temperature and the status of the heating element and air cooler.

```

1 from gpio import *
2 from time import *
3
4 def main():
5     pinMode(1, IN)
6     pinMode(2, OUT)
7     pinMode(3, OUT)
8
9     print("Blinking")
10    while True:
11        TEMP = analogRead(1);
12        TEMP_CEL = ((float(TEMP)/1023)*200-100);
13        if TEMP_CEL < 20:
14            digitalWrite(2, LOW);
15            print("Low temperature, turn on the heating element");
16            digitalWrite(3, HIGH);
17            print("Heating element finished");
18            print("Temp now = ");
19            print(TEMP_CEL);
20            delay(500);
21        elif TEMP_CEL > 25:
22            digitalWrite(2, HIGH);
23            print("High temperature, turn on the air cooler");
24            digitalWrite(3, LOW);
25            print("Air cooler finished");
26            print("Temp now = ");
27            print(TEMP_CEL);
28            delay(500);
29        else:
30            digitalWrite(2, LOW);
31            digitalWrite(3, LOW);
32            print("Temperature is normal");
33            delay(500);
34
35    main()

```

The output window shows the following results:

```

Temp now = 22.38514174
The temp is normal
Temp now = 23.9491691105

```

On the right side of the interface, a diagram shows the temperature control system at 22.90 C. The Temperature Monitor IoT8 displays 22.90 C. The Heating Element IoT6 is inactive (flame icon with a red circle), and the Air Cooler IoT7 is inactive (snowflake icon with a red circle). The Temperature Sensor IoT5 is connected to the SBC-PT SOC0 module.

Код программы:

```
from gpio import *
from time import *

def main():
    pinMode(1, IN)
    pinMode(2, OUT)
    pinMode(3, OUT)

    print("Blinking")
    while True:
        TEMP = analogRead(1);
        TEMP_CEL = ((float(TEMP)/1023)*200-100);
        if TEMP_CEL < 20:
            digitalWrite(2, LOW);
            print("Low temperature, turn on the heating element");
            digitalWrite(3, HIGH);
            print("Heating element finished");
            print("Temp now = ");
            print(TEMP_CEL);
            delay(500);
        elif TEMP_CEL > 25:
            digitalWrite(3, LOW);
            print("High temperature, turn on the air cooler");
            digitalWrite(2, HIGH);
            print("Air cooler finished");
            print("Temp now = ");
            print(TEMP_CEL);
            delay(500);
        else:
            print("The temp is normal");
            print("Temp now = ");
            print(TEMP_CEL);
            delay(500);

if __name__ == "__main__":
    main()
```

Вывод: в ходе работы мной были приобретены практические навыки проектирования инфраструктуры «умного дома», изучены основам программирования микроконтроллерных устройств.