

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

ОТЧЕТ
По лабораторной работе №9

Выполнил:
Студент 3 курса
группы АС-50
Пуцыкович А. В.
Проверил:
Давидюк Ю.И.

Брест 2020

Лабораторная работа №9

Цель работы:

приобрести практические навыки проектирования инфраструктуры $\frac{3}{4}$ умного дома, научиться основам программирования микроконтроллерных устройств

Необходимое ПО

Для выполнения лабораторной работы необходимо установить программу Cisco Packet Tracer версии не ниже 7.1. В процессе установки программы может потребоваться пройти регистрацию на официальном сайте компании Cisco.

Требования к оформлению отчета

Отчет по лабораторной работе должен содержать следующие разделы (примеры оформления отчетов можно найти в папке с заданиями):

- 1) Изложение цели работы.
- 2) Задание по лабораторной работе с описанием своего варианта.
- 3) Спецификации ввода-вывода программы.
- 4) Текст программы (кратко).
- 5) Выводы по проделанной работе.

Задание 1

Последовательно выполните задания, используя режим реального времени CPT. В качестве результата должна быть получена работающая сетевая инфраструктура, изображенная на рис. 1.

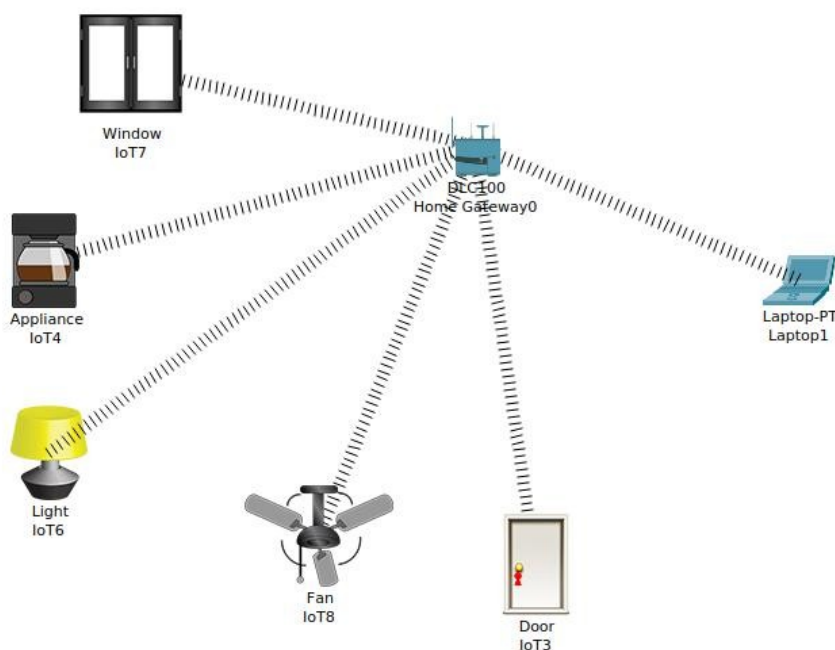
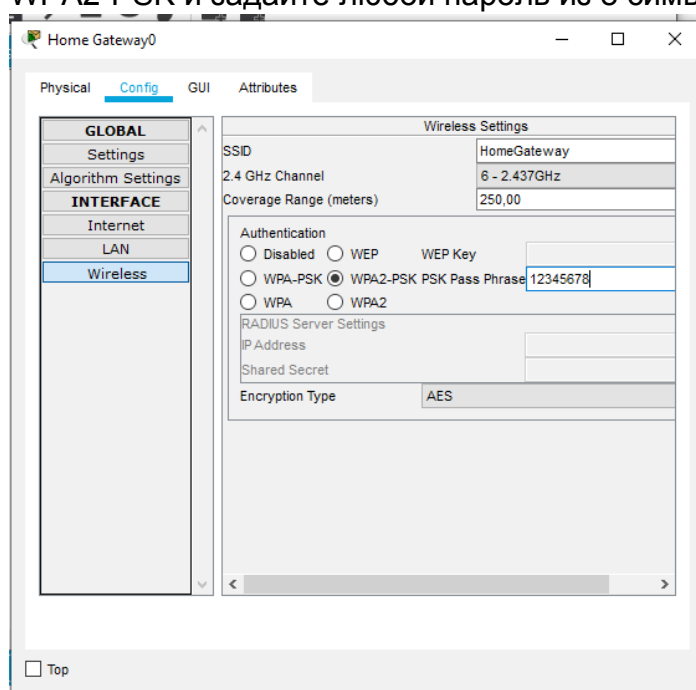


Рис. 1. Схема подключения устройств $\frac{3}{4}$ умного дома

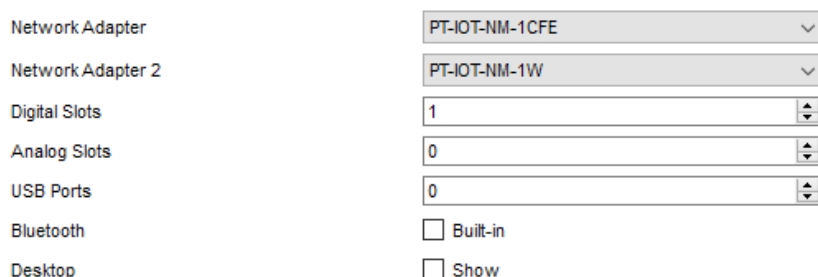
- 1) Все необходимые устройства могут быть найдены во вкладках End Devices ! End Devices, End Devices ! Home и Network Devices ! Wireless Devices. Ключевое устройство Home Gateway. Именно оно объединяет все устройства умного дома и клиентские терминалы (такие, как лэптоп) в общую беспроводную сеть. Это сервер IoT.



- 2) После размещения всех необходимых устройств в рабочей области откройте Home Gateway и во вкладке Config ! Interface ! Wireless определите тип аутентификации как WPA2-PSK и задайте любой пароль из 8 символом (например, cisco123).



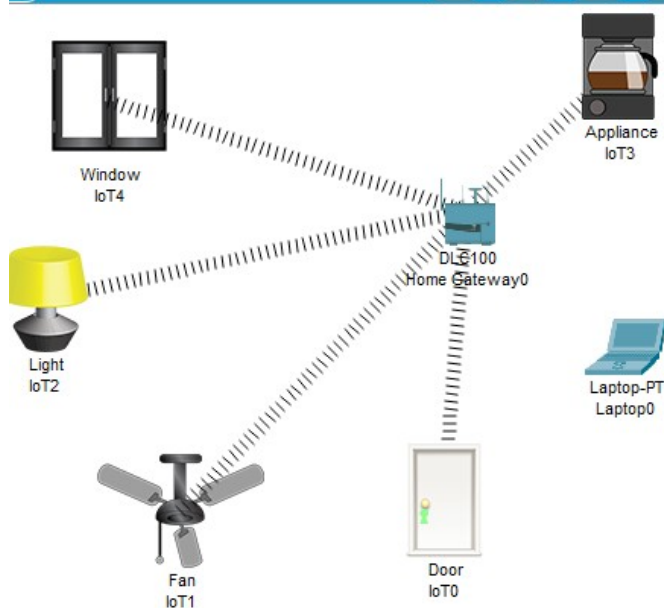
- 3) После настройки сервера, переходим на любое устройство IoT и открываем расширенные на-стройки (Advanced). Дело в том, что эти устройства по умолчанию не поддерживают беспроводную передачу данных. Откройте вкладку I/O Config. Далее в списке Network Adapter2 выберите беспроводной адаптер PT-IOT-NM-1W.



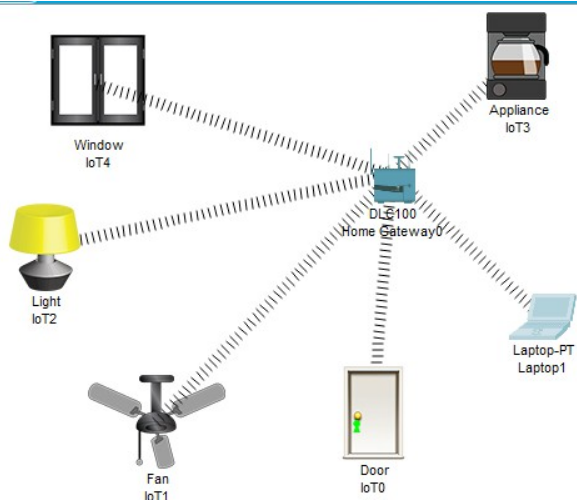
- 4) После выполнения предыдущего действия во вкладке Config появится беспроводной интер-фейс Wireless3. Откройте его и настройте подключение к серверу, задав правильный тип аутентификации, пароль и выбрав вариант DHCP в IP Configuration (этот

вариант чаще всего задан по умолчанию, убедитесь в этом случае, что узлом получен IP-адрес из того же диапазона, что и IP-адрес сервера – как правило, из 192.168.25.0). В данном случае сервер IoT Home Gateway является DHCP-сервером для подключаемых устройств (автоматически раздает IP-адреса).

- 5) Далее откройте Settings (там же, во вкладке Config) и поставьте в группе IoT Server переключатель в положение Home Gateway.
- 6) После выполнения всех этих действий, убедитесь, что между сервером и настраиваемым узлом появилось отображение беспроводной связи.
- 7) Проделайте действия 3-6 для других устройств, исключая лэптоп.



- 8) Откройте лэптоп и изучите его физическую конфигурацию. Вы можете заметить, что на нем также, как и на IoT-устройствах не установлен модуль беспроводной связи. Это можно исправить следующим образом: извлеките установленный Fast Ethernet-модуль (предварительно выключив лэптоп) и поместите в свободный слот модуль PT-LAPTOP-NM-1W. После этого включите устройство и произведите похожие настройки беспроводного интерфейса (укажите SSID, тип аутентификации и пароль). Между сервером и лэптопом должна появиться визуализация беспроводной связи.



- 9) Откройте вкладку Desktop лэптопа и далее IoT Monitor. Нажмите Ok в окне авторизации на сервере, убедившись в правильности написанного IP-адреса сервера. После этого

перед ва-ми должен появиться список всех беспроводных устройств, подключенных к нашему серверу. Поэкспериментируйте с кнопками включения/выключения устройств и изучите изменения, ко-торые с ними происходят.

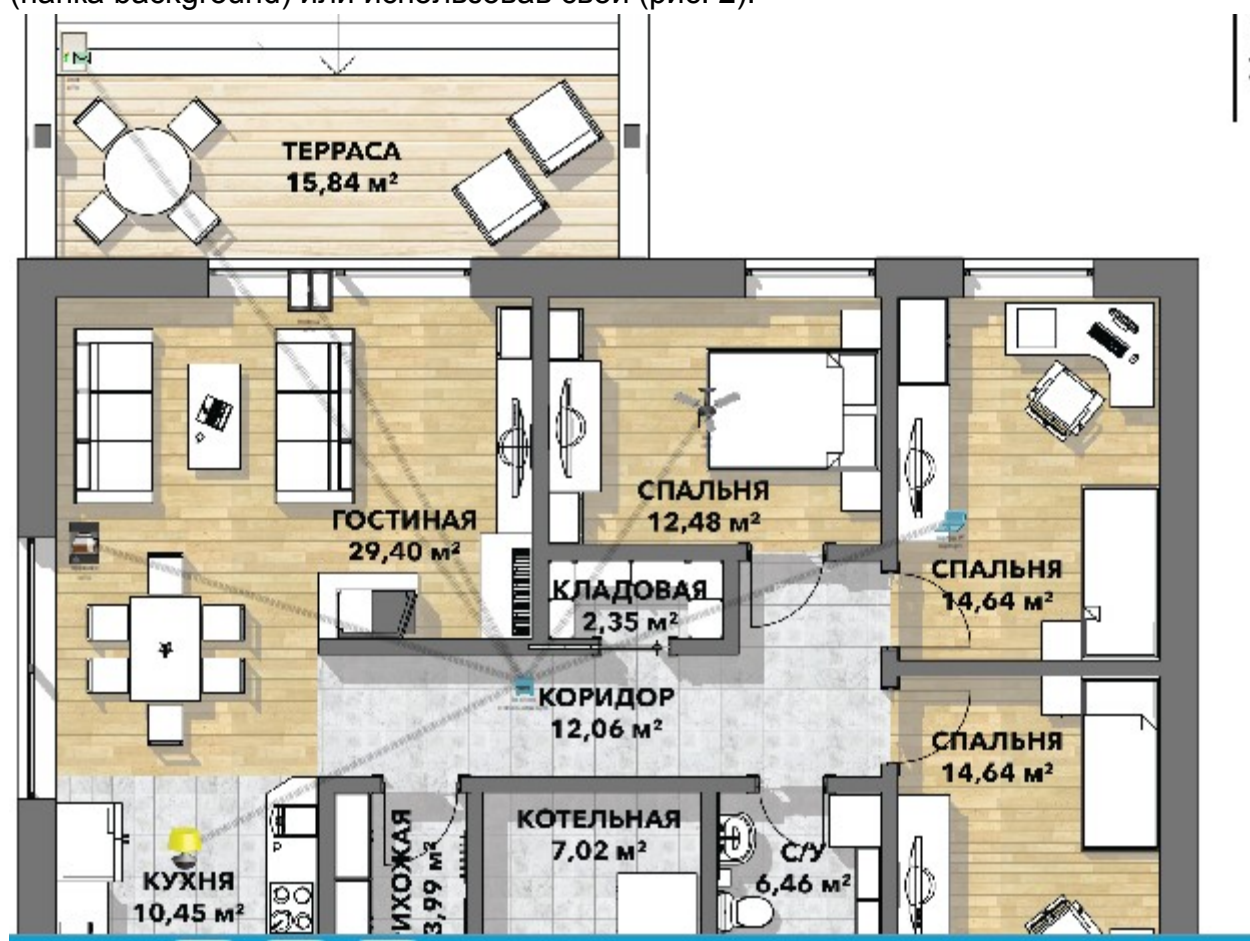
```
Pinging 169.254.60.48 with 32 bytes of data:

Reply from 169.254.60.48: bytes=32 time=8ms TTL=128
Reply from 169.254.60.48: bytes=32 time=3ms TTL=128
Reply from 169.254.60.48: bytes=32 time=1ms TTL=128
Reply from 169.254.60.48: bytes=32 time=3ms TTL=128

Ping statistics for 169.254.60.48:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 8ms, Average = 3ms

C:\>
```

- 10) Добавьте фон для построенной инфраструктуры, воспользовавшись предложенными (папка background) или используя свой (рис. 2).



Задание 2

В первом задании, несмотря на наличие IoT-устройств, сформирована лишь сетевая инфраструктура, но не полноценное IoT-решение. Это так, поскольку все устройства контролируются (пусть и удаленно), но человеком. Т.е. человек принимает решения о включении/выключении устройств, а не сама система. Попробуем создать решение, которое будет обладать определенной автономностью.



Рис. 2. Фон для сетевой инфраструктуры

Для этого воспользуемся микроконтроллерными устройствами, которые будут принимать решение о активации тех или иных узлов системы. Спроектируем систему для поддержания комфортной температуры внутри помещения, изображенную на рис. 3

- 1) Для начала добавьте микроконтроллерную плату в рабочую область (вкладка Components ! Boards). Выберите из предложенных плату SBC Board.
- 2) Откройте добавленную плату на вкладке Programming. Далее в списке слева выберите пункт Blink (Python) и далее скрипт main.py. Программирование для такой платы производится на языке Python. Он является достаточно простым скриптовым языком с большим количеством разработанных библиотек (подробнее о языке можно почитать в предложенной презентации). Скрипт, который откроется, нужен для решения простой задачи – он включает и выключает пин (разъем) на нашей плате, активируя подключенную к нему нагрузку. В качестве такой нагрузки может выступать светодиоды, разные датчики, LCD-экраны и т.д.
- 3) Попробуйте добавить светодиод (LED) с вкладки Components ! Actuators к рабочей области. Затем во вкладке Connections выберите тип соединения IoT Custom Cable и соедините пин D1 вашей платы с пином D0 светодиода. Запустите программу, нажав на кнопку Run. Вы должны увидеть мигающий светодиод. Откройте программу, попытайтесь изучить и понять ее содержимое. Команда pinMode нужна для определения режима, в котором будет работать наш пин платы (это может быть IN или OUT – для выходных и входных сигналов соответственно). Как следует из программы, мы делаем пин D1 (или просто пин с номером 1) выходным, для того, чтобы регулировать уровень напряжения и включать и выключать его. Пины бывают цифровыми (D) и аналоговыми (A). Цифровые пины оперируют 0 и 1 (или LOW и HIGH) и лучше всего описывают взаимодействие с устройствами, которые нужно включать

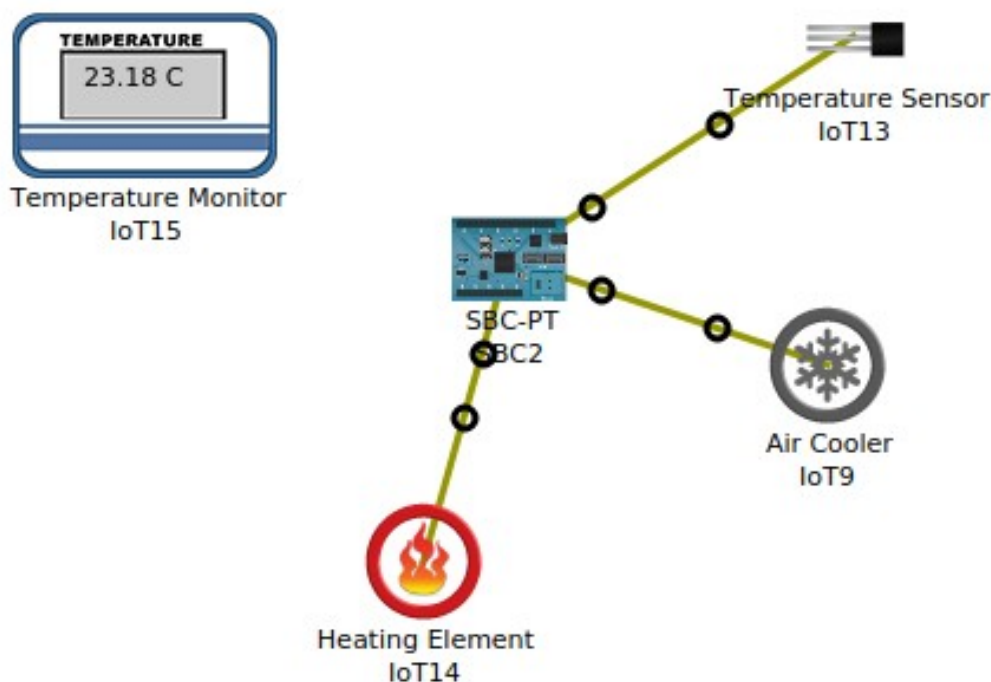
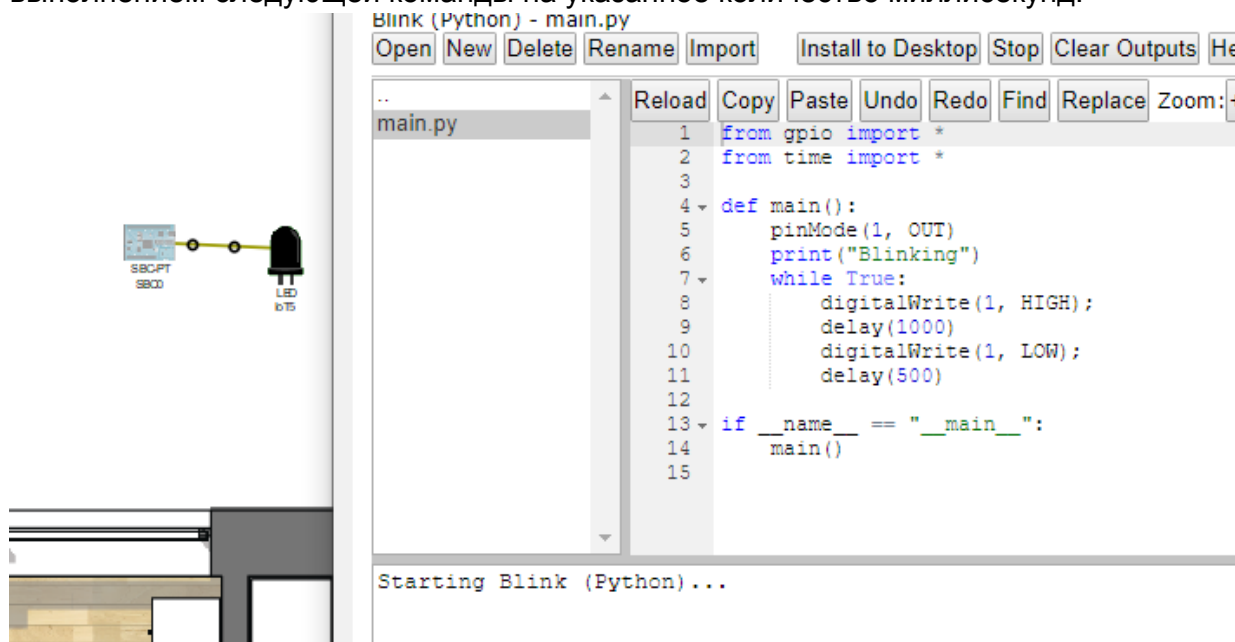


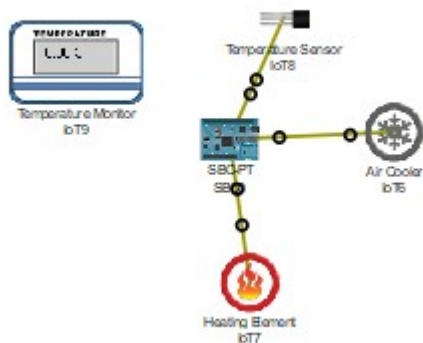
Рис. 3. Микроконтроллерная схема

и выключать. Аналоговые пины нужны для передачи какой-то многоуровневой информации (например, уровня температуры и влажности).

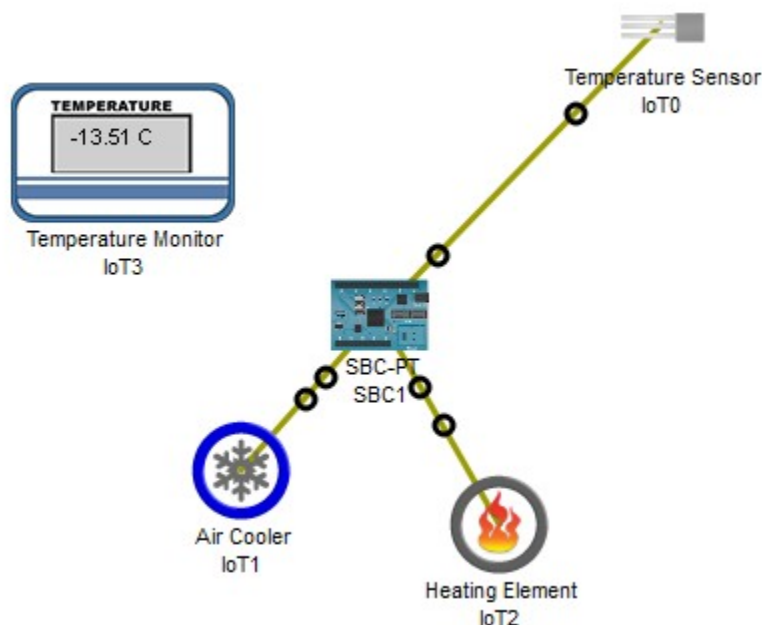
Как вы видите, в программе мы записываем попеременно высокий и низкий сигнал в пин номер 1, что приводит к миганию светодиода (это делается с помощью функции `digitalWrite` с указанием номера пина и уровня сигнала). Функция `delay` вызывает задержку перед выполнением следующей команды на указанное количество миллисекунд.



- 4) Удалите LED из рабочей области. Добавьте другие компоненты, необходимые для реализации проекта (вкладка Actuators), а также цифровой термометр для отслеживания температуры (End Devices ! Home! Temperature Monitor)). Температурный сенсор находится на вкладке (Components ! Sensors! Temperature Sensor).



- 5) Heating Element нужен для повышения температуры, Air Cooler для понижения. О характеристиках этих устройств можно почитать, кликнув по ним. Для нас важно то, что они включаются и выключаются как цифровые устройства (т.е. вызовом команды `digitalWrite`). Temperature Monitor нужен для считывания данных о температуре. Это аналоговый датчик, поэтому для считывания данных применяется функция `analogRead` с указанием единственного параметра – номера пина. Подсоедините все указанные датчики к плате, выбрав произвольные пины (запомните свой выбор). Для Temperature Monitor выберите пин A0 на нем.



- 6) Далее изучите изменение температуры в течение суток с помощью показателей температурного монитора. В CPT можно изменять текущее время суток (это делается нажатием на кнопку с `¼текущим` временем или `Shift + E`). Как вы заметите, температура изменяется. Хотелось бы, чтобы она оставалась в определенном заданном интервале (например, от 20 до 25 градусов).
- 7) Итак, мы подошли к самому главному. Теперь вам нужно написать программу, которая будет поддерживать текущую температуру в заданном интервале. Используйте пины, активируя устройства для обогрева и охлаждения на основании данных, считанных с температурного датчика. Имейте в виду, что датчик возвращает данные в интервале от 0 до 1023, соответствующие температуре -100 до 100 градусов. Используйте следующую формулу для получения значения температуры:

$$t_{\text{celsius}} = \frac{t_{\text{sensor}} - 100}{1023 - 100} \quad (1)$$

Функция `float` нужна для конвертации в вещественный тип.

Код программы

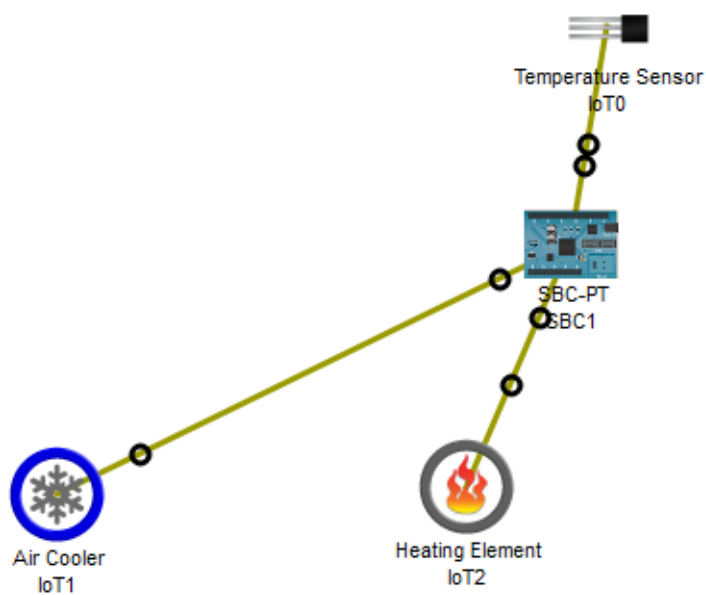
```
from gpio import *
from time import *

def main():
    pinMode(0, IN)
    pinMode(2, OUT)
    pinMode(3, OUT)
    print("Blinking")

    analogRead(1);
    a = analogRead(1)*0.19
    c = a-100
    if c > 2:
        digitalWrite(3, HIGH);
        digitalWrite(2, LOW);

    else:
        digitalWrite(2, HIGH);
        digitalWrite(3, LOW);
    print(c);
    delay(1000)

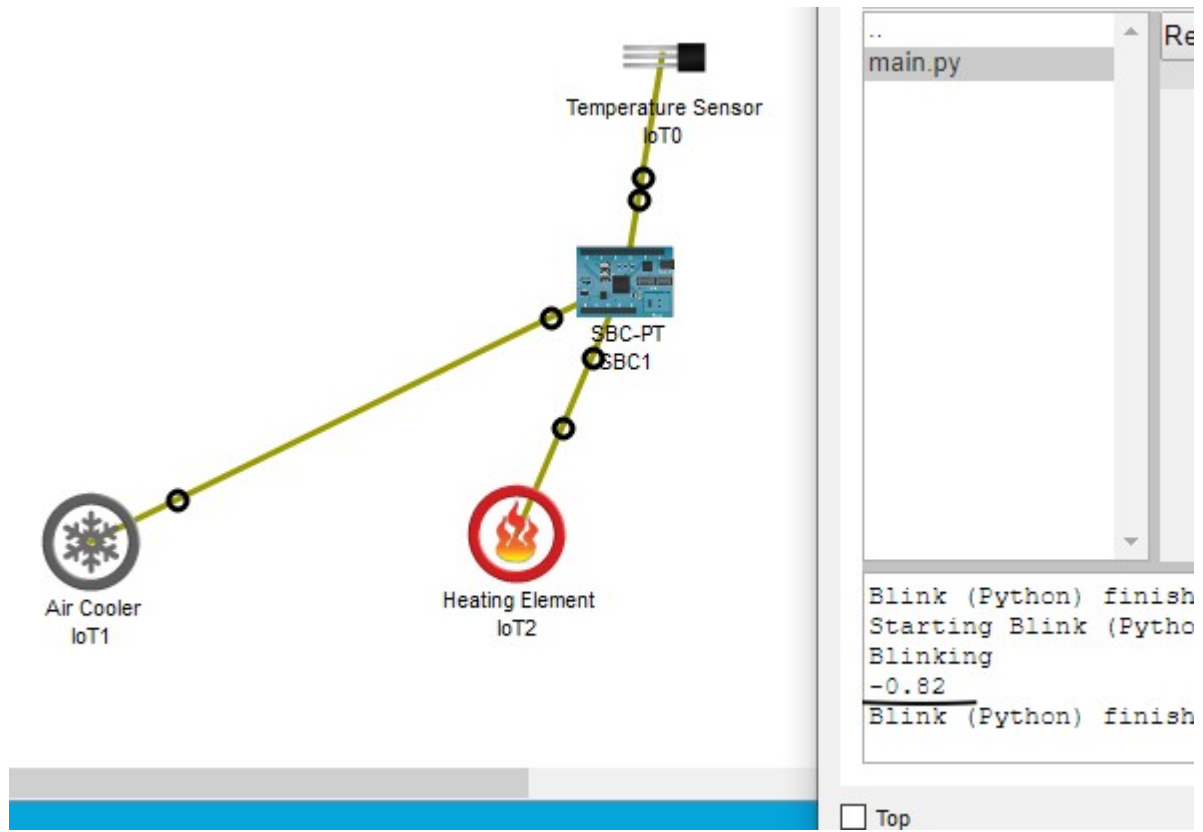
if __name__ == "__main__":
    main()
```



```
..
main.py
1 from gpio
2 from time
3
4 def main()
5     pinMode
6     pinMode
7     pinMode
8     print(
9
10    analog
11    a = an
12    c = a-
13    if c >
14        di
15        di
16
17    else:
18        di
```

```
Blink (Python) finished running.
Starting Blink (Python)...
Blinking
2.22
Blink (Python) finished running.
```

☐ Top



Вывод: Получил знания в построении системы умный дом в программе CISCO Packet Tracer. Разработал скрипт на языке Python.