

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

Лабораторная работа №4

Выполнил:
Студент 3-го курса
Группы АС-50
Федорук Д.А.
Проверила:
Давидюк Ю.И.

Брест 2020

Цель работы:

приобрести практические навыки проектирования инфраструктуры «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.

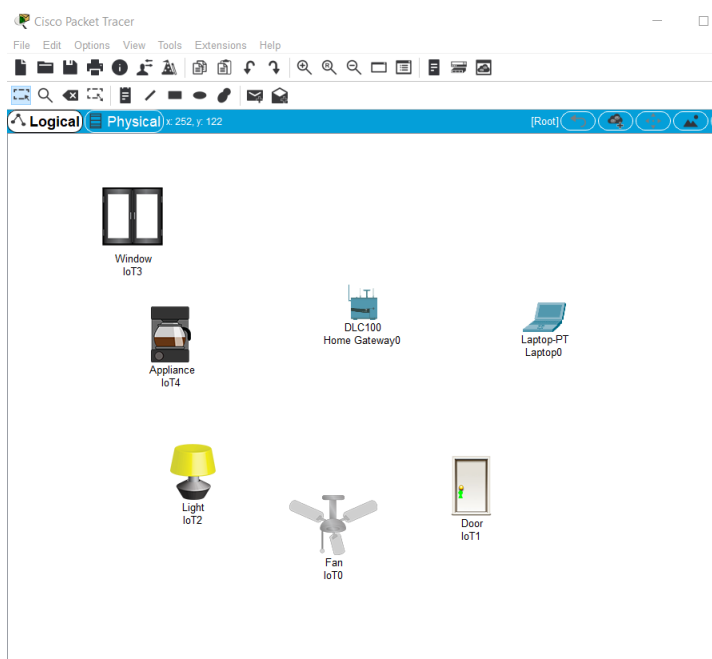
Требования к оформлению отчета

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

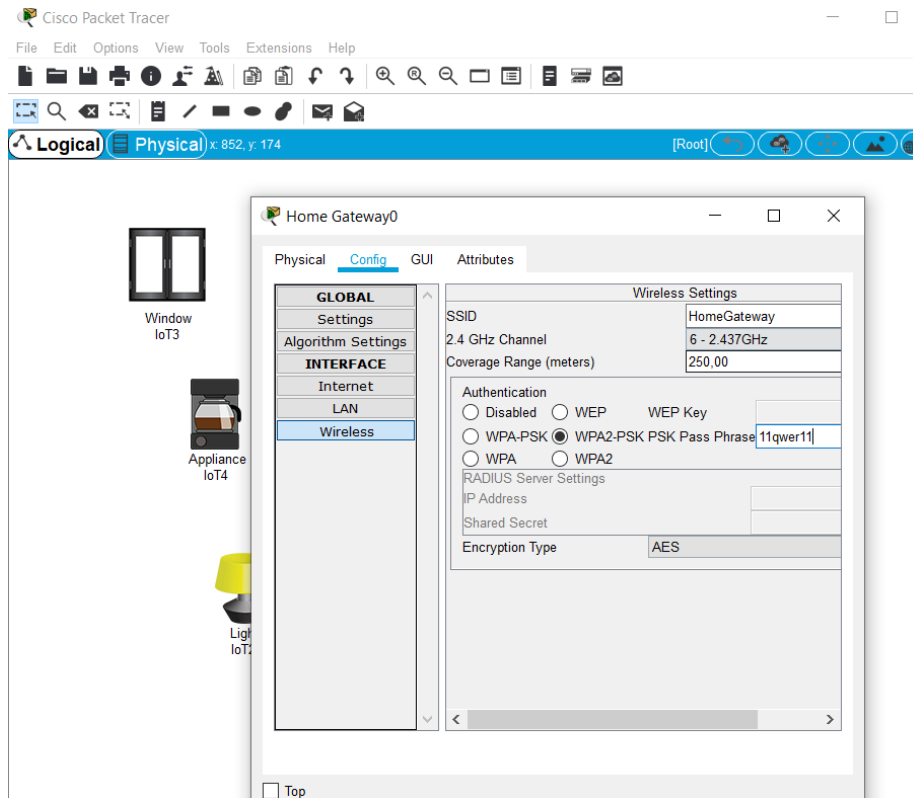
- 1) Изложение цели работы.
- 2) Задание по лабораторной работе с описанием своего варианта.
- 3) Спецификации ввода-вывода программы.
- 4) Текст программы (кратко).
- 5) Выводы по проделанной работе.

Задание 1

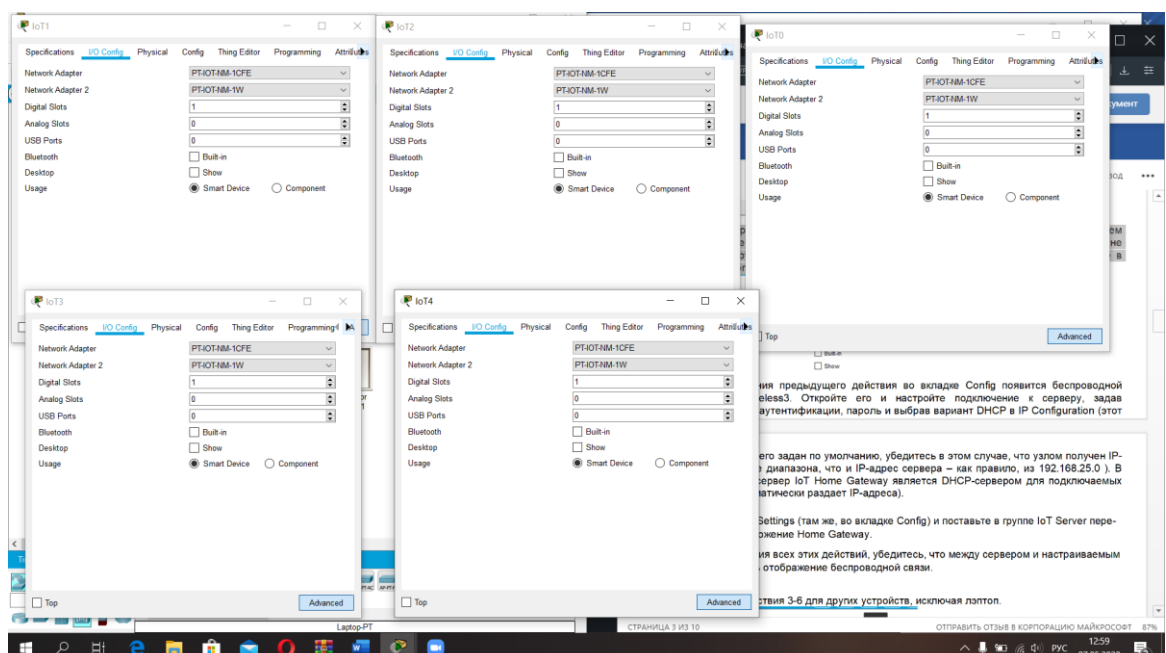
- 1) Все необходимые устройства могут быть найдены во вкладках End Devices → Home и Network Devices → Wireless Devices. Ключевое устройство Home Gateway. Именно оно объединяет все устройства умного дома и клиентские терминалы (такие, как лэптоп) в общую беспроводную сеть. Это сервер IoT.



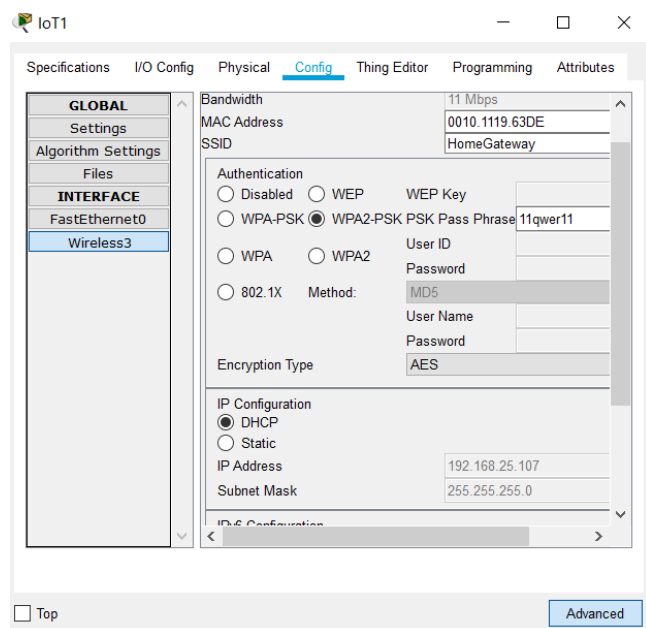
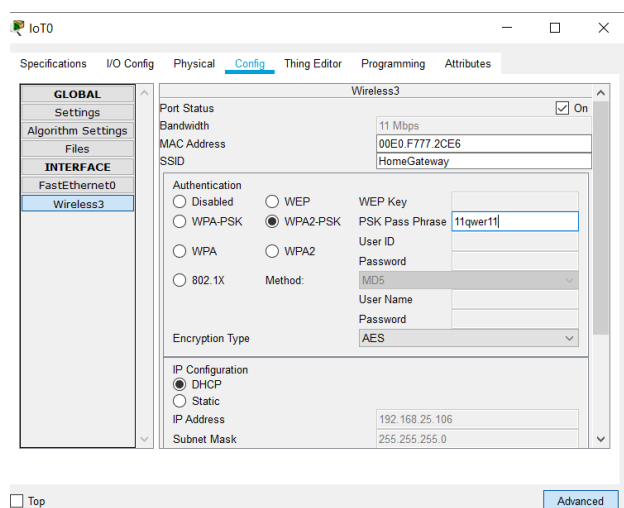
- 2) После размещения всех необходимых устройств в рабочей области откройте Home Gateway и во вкладке Config → Interface → Wireless определите тип аутентификации как WPA2-PSK и задайте любой пароль из 8 символом (например, cisco123)



- 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 интерфейс 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).



IoT2

Specifications I/O Config Physical **Config** Thing Editor Programming Attributes

GLOBAL

Settings

Algorithm Settings

Files

INTERFACE

FastEthernet0

Wireless3

Wireless3

Port Status ☒ On

Bandwidth 11 Mbps

MAC Address 000D.BD26.2884

SSID HomeGateway

Authentication

☐ Disabled ☐ WEP WEP Key

☐ WPA-PSK ☒ WPA2-PSK PSK Pass Phrase 11qwer11

☐ WPA ☐ WPA2 User ID

☐ 802.1X Method: MD5 Password

User Name

Password

Encryption Type AES

IP Configuration

☒ DHCP

☐ Static

IP Address 192.168.25.104

Subnet Mask 255.255.255.0

IoT3

Specifications I/O Config Physical **Config** Thing Editor Programming Attributes

GLOBAL

Settings

Algorithm Settings

Files

INTERFACE

FastEthernet0

Wireless3

Wireless3

Port Status ☒ On

Bandwidth 11 Mbps

MAC Address 000A.F335.B488

SSID HomeGateway

Authentication

☐ Disabled ☐ WEP WEP Key

☐ WPA-PSK ☒ WPA2-PSK PSK Pass Phrase 11qwer11

☐ WPA ☐ WPA2 User ID

☐ 802.1X Method: MD5 Password

User Name

Password

Encryption Type AES

IP Configuration

☒ DHCP

☐ Static

IP Address 192.168.25.101

Subnet Mask 255.255.255.0

☐ Top [Advanced](#)

IoT4

Specifications I/O Config Physical **Config** Thing Editor Programming Attributes

GLOBAL

Settings

Algorithm Settings

Files

INTERFACE

FastEthernet0

Wireless3

Wireless3

Port Status ☒ On

Bandwidth 11 Mbps

MAC Address 0060.2F5B.6E67

SSID HomeGateway

Authentication

☐ Disabled ☐ WEP WEP Key

☐ WPA-PSK ☒ WPA2-PSK PSK Pass Phrase 11qwer11

☐ WPA ☐ WPA2 User ID

☐ 802.1X Method: MD5 Password

User Name

Password

Encryption Type AES

IP Configuration

☒ DHCP

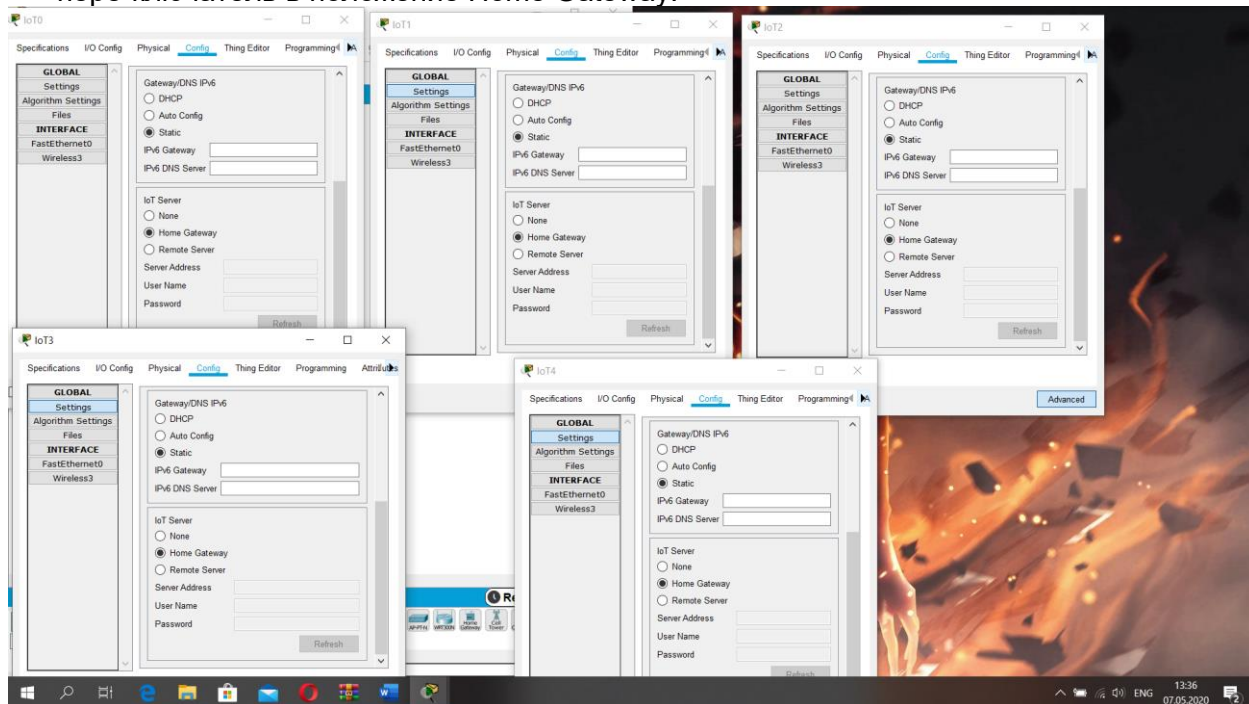
☐ Static

IP Address 192.168.25.103

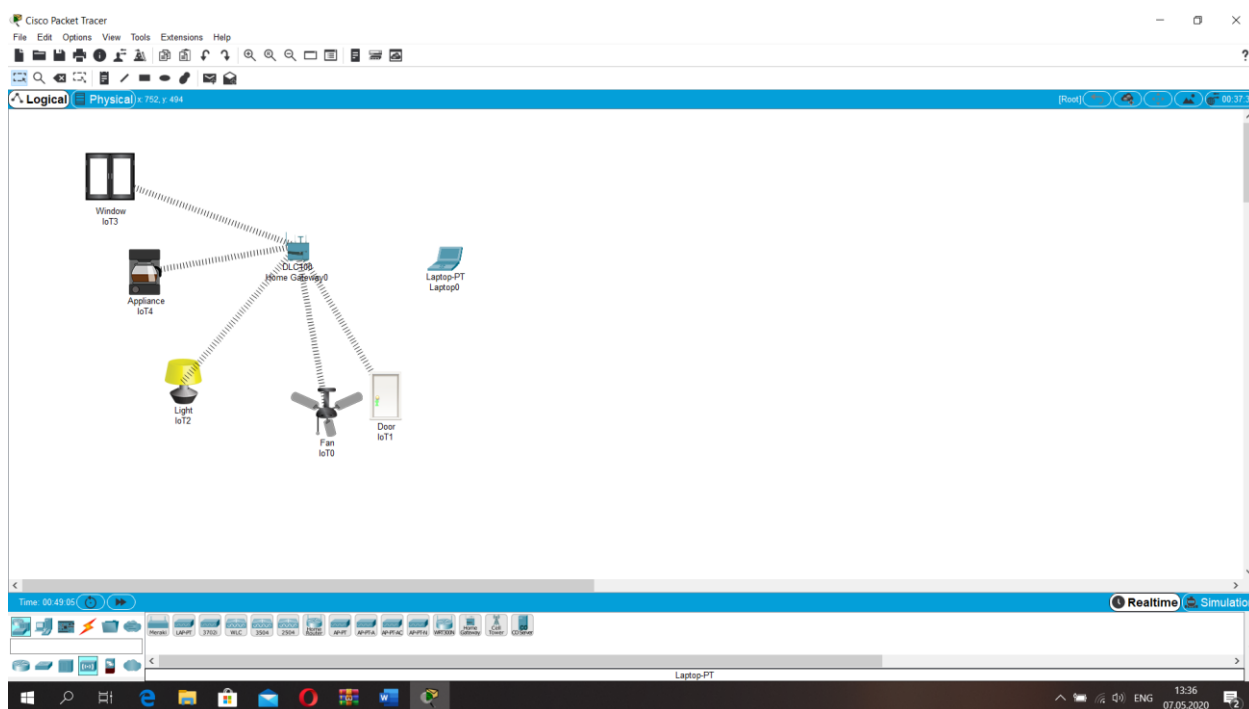
Subnet Mask 255.255.255.0

☐ Top [Advanced](#)

- 5) Далее откройте Settings (там же, во вкладке Config) и поставьте в группе IoT Server пере-ключатель в положение Home Gateway.



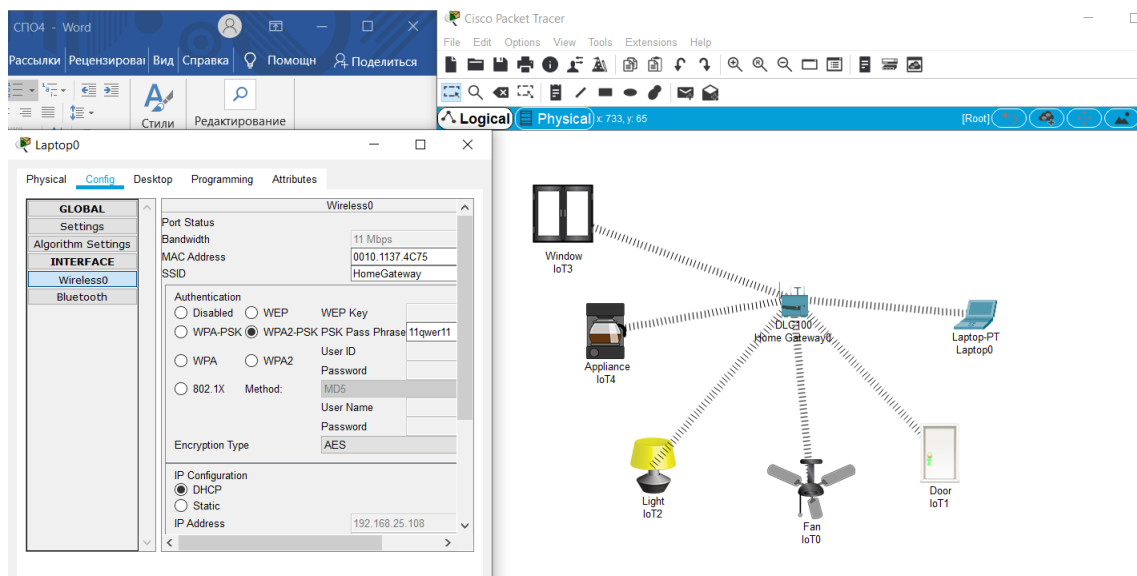
- 6) После выполнения всех этих действий, убедитесь, что между сервером и настраиваемым узлом появилось отображение беспроводной связи.



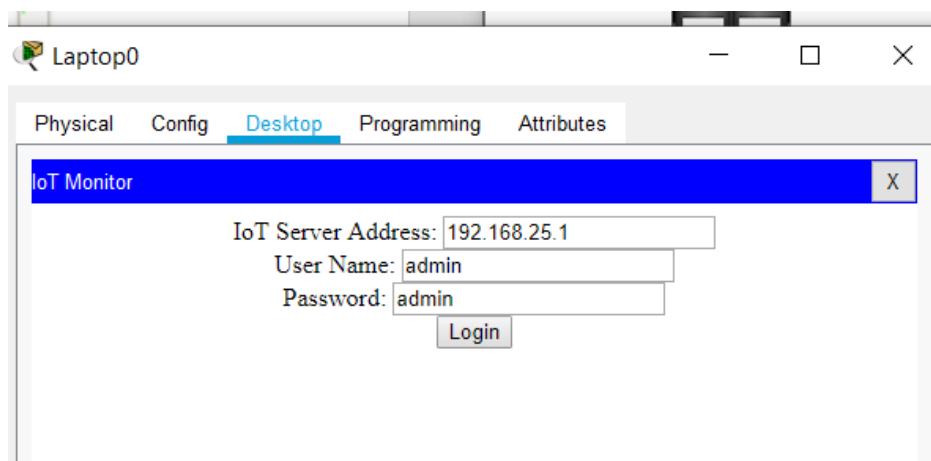
- 7) Проделайте действия 3-6 для других устройств, исключая лэптоп. Проделано сразу для всех

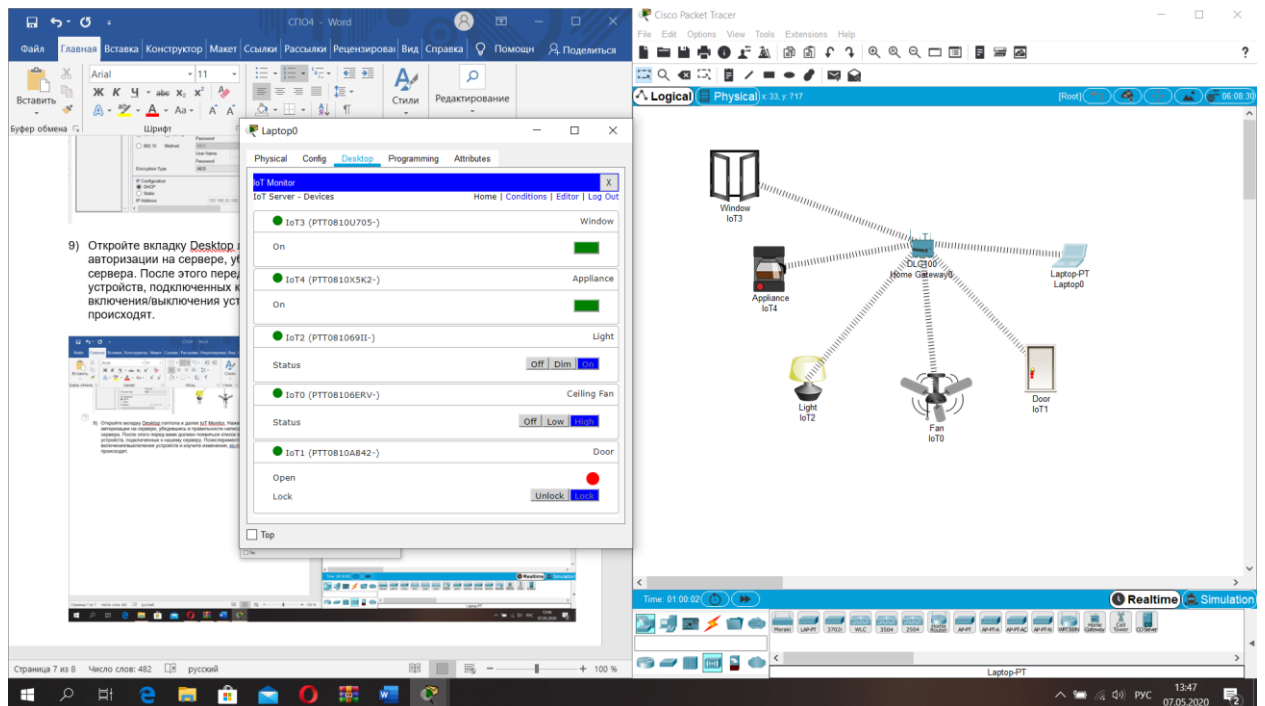
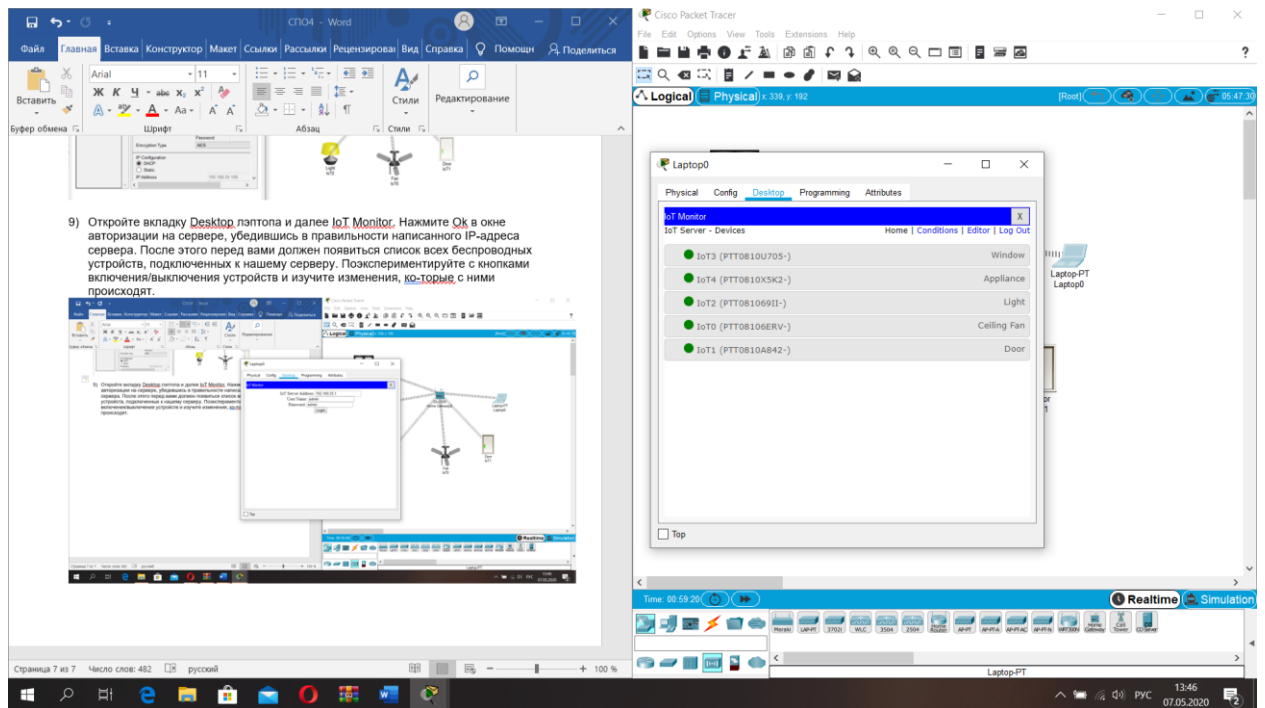
- 8) Откройте лэптоп и изучите его физическую конфигурацию. Вы можете заметить, что на нем также, как и на IoT-устройствах не установлен модуль беспроводной связи. Это можно исправить следующим образом: извлеките установленный Fast Ethernet-модуль (предварительно выключив лэптоп) и поместите в свободный 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 ноутбуком должна появиться визуализация беспроводной связи.



- 9) Откройте вкладку Desktop ноутбука и далее IoT Monitor. Нажмите Ok в окне авторизации на сервере, убедившись в правильности написанного IP-адреса сервера. После этого перед вами должен появиться список всех беспроводных устройств, подключенных к нашему серверу. Поэкспериментируйте с кнопками включения/выключения устройств и изучите изменения, которые с ними происходят.





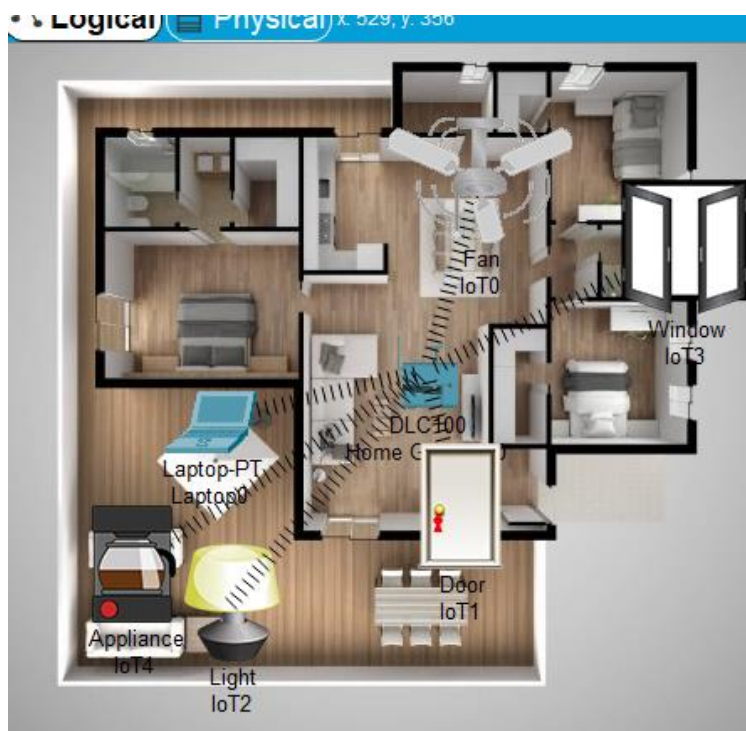
Все устройства открыты (произошли визуальные изменения: двери раскрылись, на кофеварке загорелась лампочка, лампа включилась, вентилятор заработал (эти волны вокруг похожи на порывы воздуха), на ручке двери появилось что-то красное (что видимо символизирует, что она открыта))

10) Добавьте фон для построенной инфраструктуры, воспользовавшись предложенными (папка background) или использовав свой

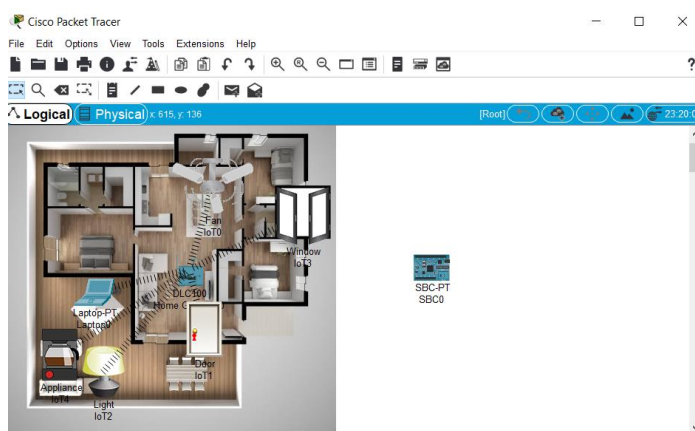


Задание 2

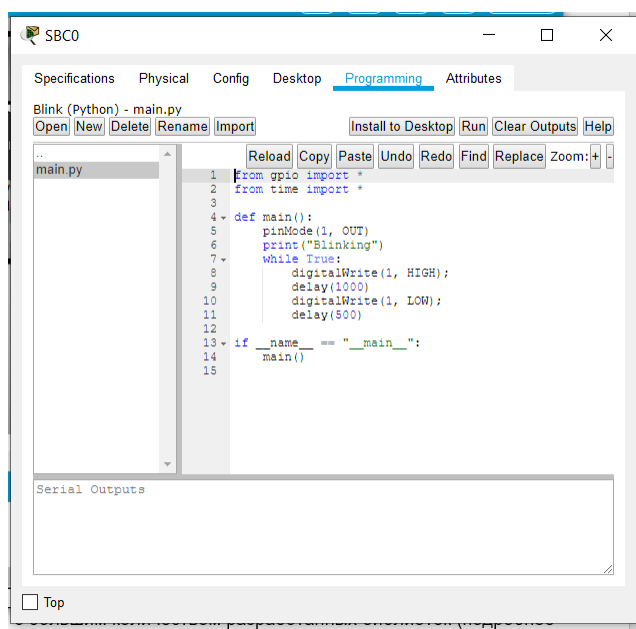
В первом задании, несмотря на наличие IoT-устройств, сформирована лишь сетевая инфраструктура, но не полноценное IoT-решение. Это так, поскольку все устройства контролируются (пусть и удаленно), но человеком. Т.е. человек принимает решения о включении/выключении устройств, а не сама система. Попробуем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



- 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.

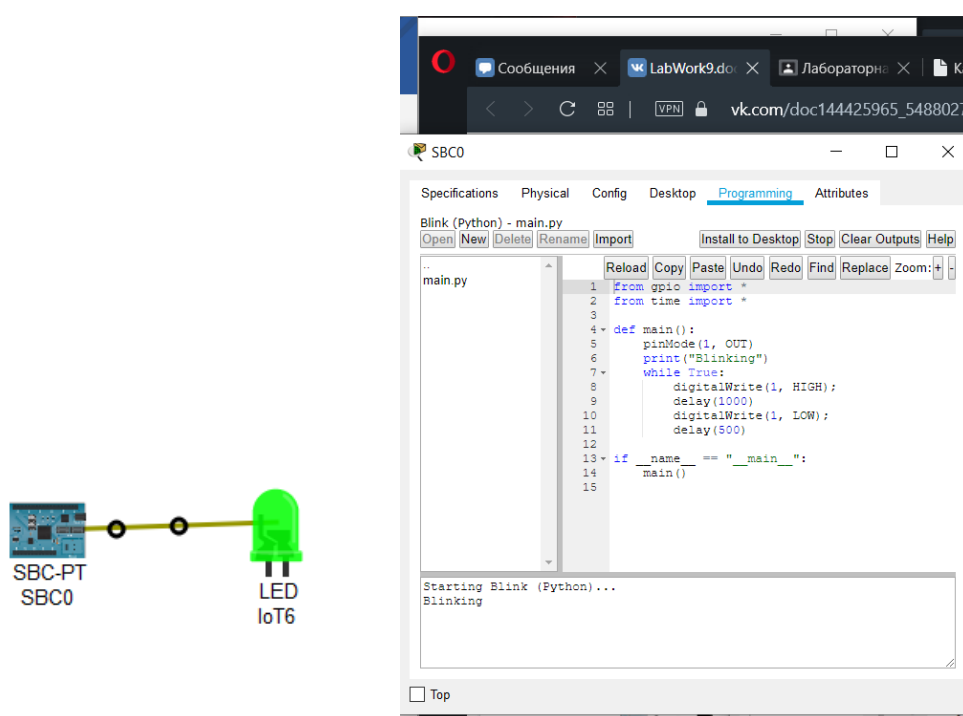


- 3) Попробуйте добавить светодиод (LED) с вкладки Components → Actuators к рабочей области. Затем во вкладке Connections выберите тип соединения IoT Custom Cable и соедините пин D1 вашей платы с пином D0 светодиода. Запустите программу, нажав на кнопку Run. Вы должны увидеть мигающий светодиод. Откройте 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 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.

При нажатии Run лампочка загорается.



- 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 Sensor 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 $\frac{3}{4}$ текущим 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:

$$t_{celsius} = \frac{t_{sensor}}{1023} * 200 - 100$$

Функция float нужна для конвертации в вещественный тип

Код программы

```
from gpio import*
from time import*
def main():
    pinMode(2, OUT)
    pinMode(3, OUT)
    pinMode(0, IN)
    print("Blinking")
    while True:
        a = ((analogRead(0)*200/1023)-100)
        if a<20:{
            digitalWrite(2, HIGH);
            digitalWrite(3, LOW);
        }
        if a>25:{
            digitalWrite(3, HIGH);
            digitalWrite(2, LOW);
        }
        print(a);
        delay(1000)
if __name__ == "__main__":
```

main()

LabWork9.docx

Продук

SBC0

Specifications Physical Config Desktop **Programming** Attributes

Blink (Python) - main.py

Open New Delete Rename Import Install to Desktop Run Clear Outputs Help

main.py

```
1 from gpio import*
2 from time import*
3 def main():
4     pinMode(2, OUT)
5     pinMode(3, OUT)
6     pinMode(0, IN)
7     print("Blinking")
8     while True:
9         a = ((analogRead(0)*200/1023)-100)
10        if a<20:{
11            digitalWrite(2, HIGH);
12            digitalWrite(3, LOW);
13        }
14        if a>25:{
15            digitalWrite(2, HIGH);
16            digitalWrite(3, LOW);
17        }
18        print(a);
19        delay(1000)
20    if __name__ == "__main__":
21        main()
22
```

Blink (Python) finish
Starting Blink (Pytho
Blinking
-0.82
Blink (Python) finish

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

☐ Top