Лисовская П.С. АС-51

Лабораторная работа №9

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

Задание 1

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 настройки (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).

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

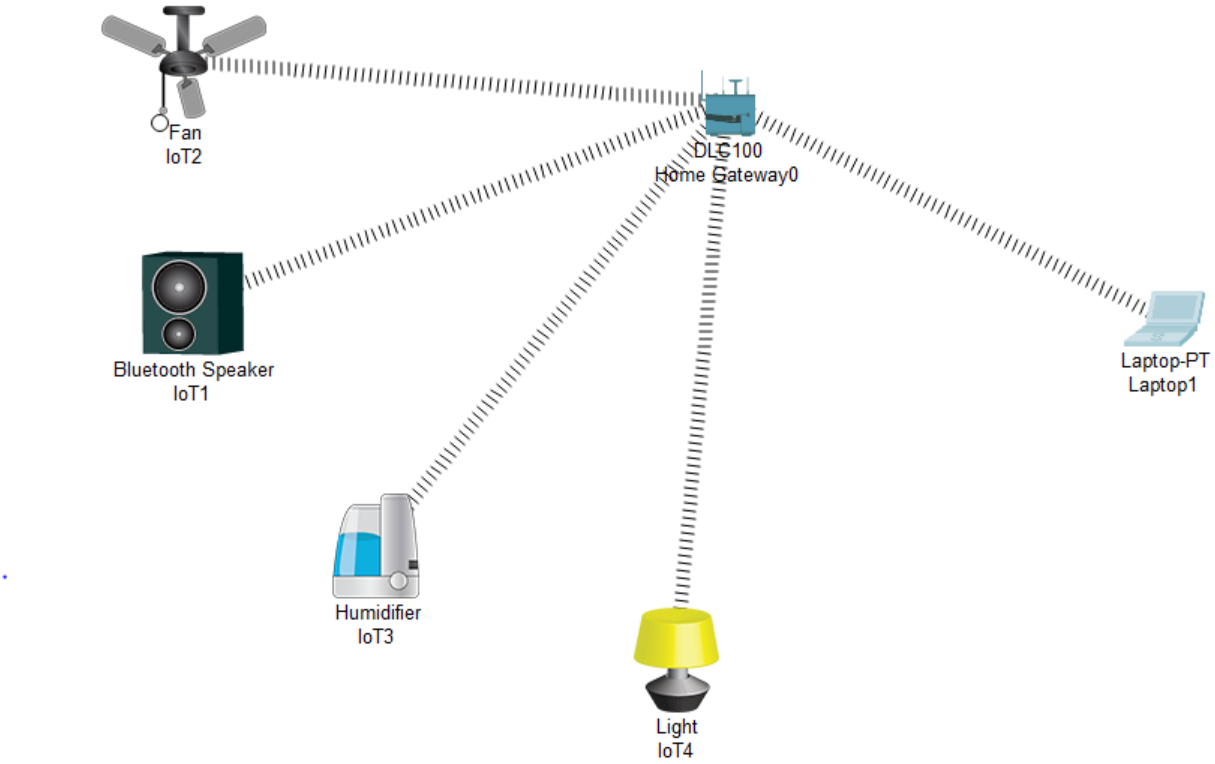
6) После выполнения всех этих действий, убедитесь, что между сервером и настраиваемым узлом появилось отображение беспроводной связи.

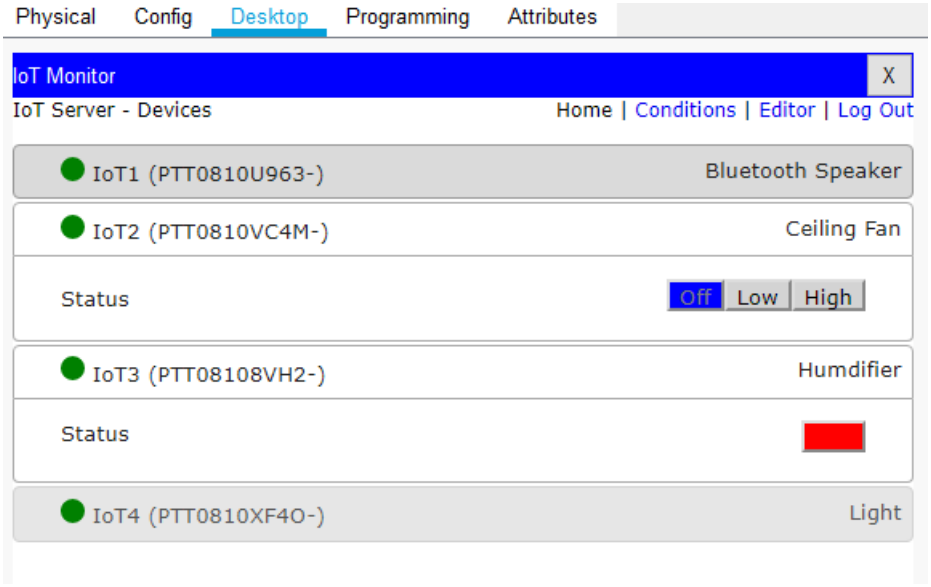
7) Проделайте действия 3-6 для других устройств, исключая лэптоп.

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 вами должен появиться список всех беспроводных устройств, подключенных к нашему серверу. Поэкспериментируйте с кнопками включения/выключения устройств и изучите изменения, которые с ними происходят.

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

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 включатьи 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 «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). 4 Крощенко А.А., Системное программное обеспечение, ЛР9, 2020

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: tcelsius = tsensor 1023 ∗ 200 − 100 (1) Функция float нужна для конвертации в вещественный тип

Specifications Physical Config Programming Attributes

Blink (Python) - main.py

Open New Delete Rename Import Run Clear Outputs Help

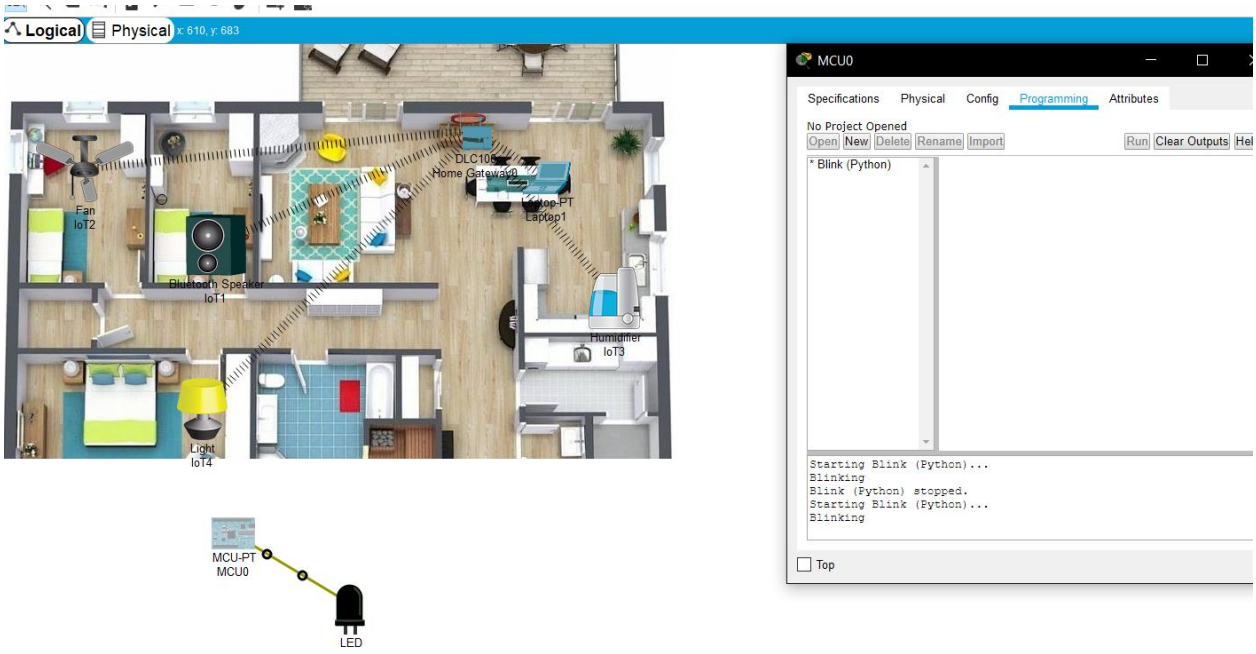
main.py

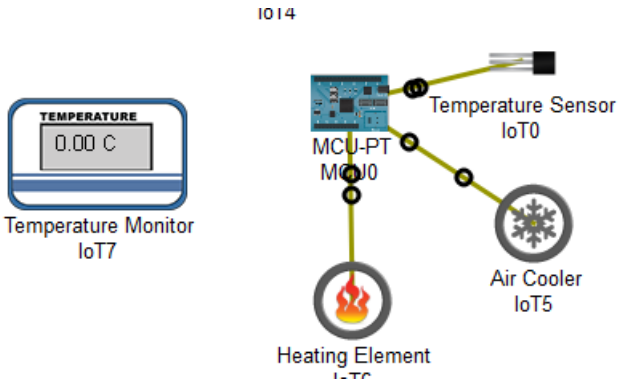
Reload Copy Paste Undo Redo Find Replace Zoom: +

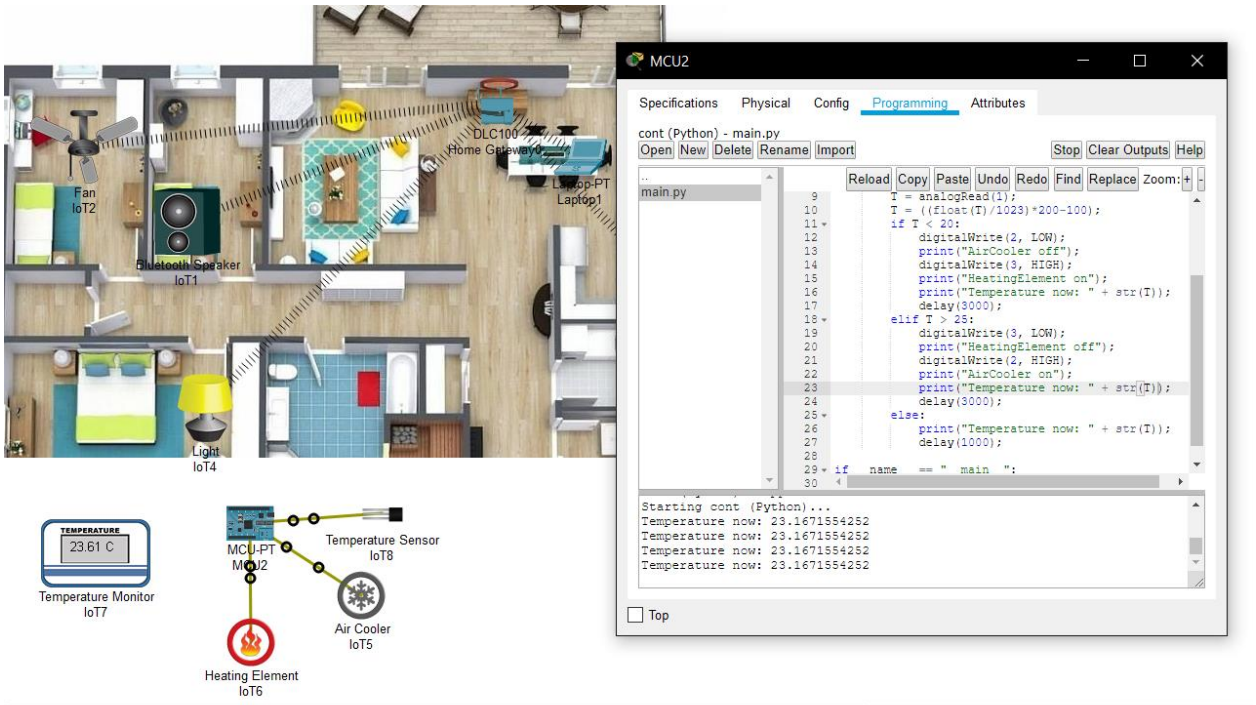
```
1  from gpio import *
2  from time import *
3
4  def main():
5      pinMode(1, OUT)
6      print("Blinking")
7      while True:
8          digitalWrite(1, HIGH);
9          sleep(1)
10         digitalWrite(1, LOW);
11         sleep(0.5)
12
13 if __name__ == "__main__":
14     main()
```

Serial Outputs

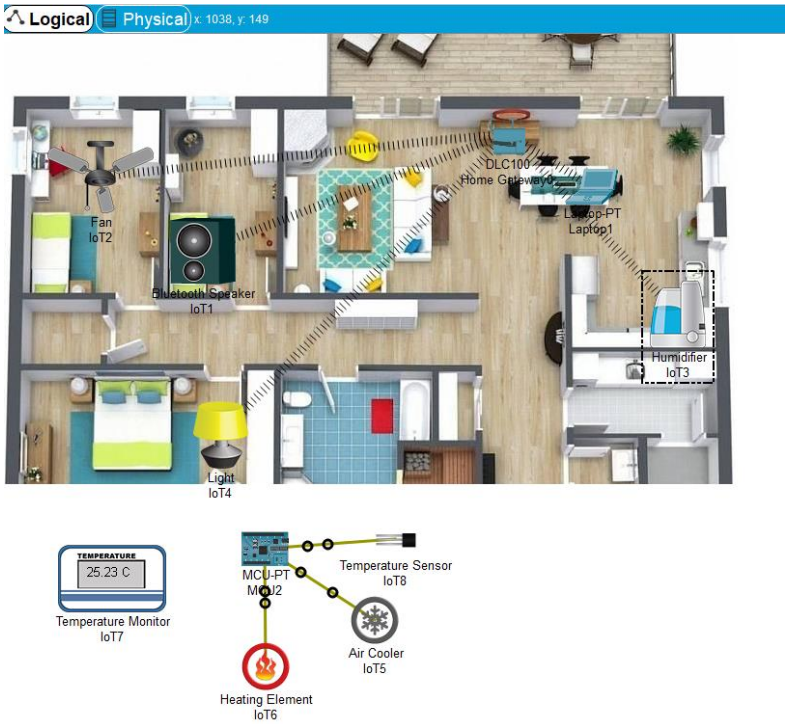
Top

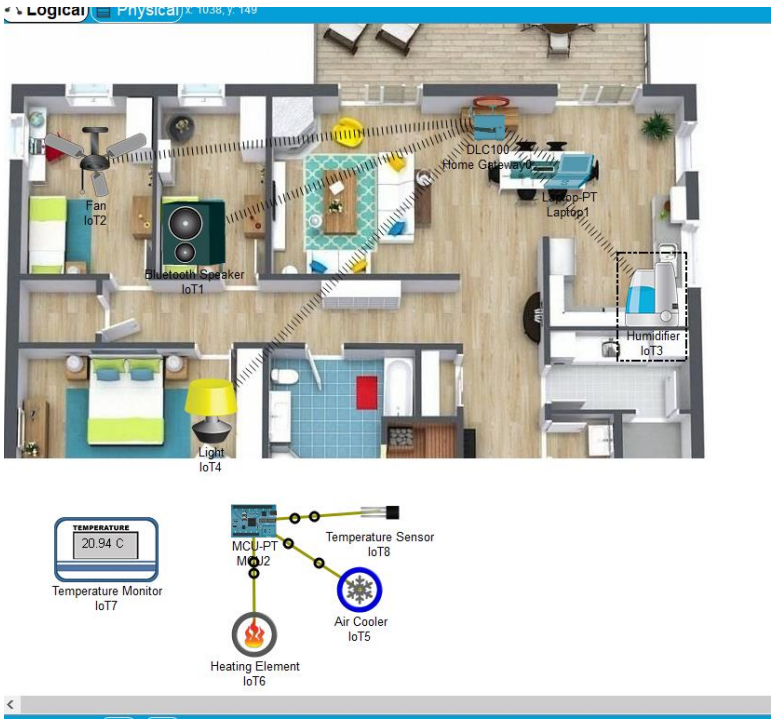


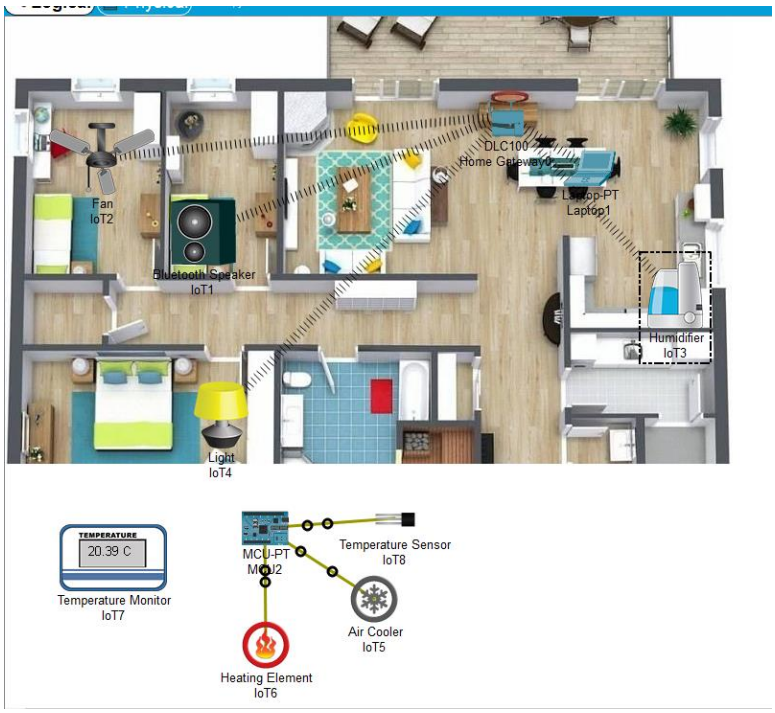

IoT4
TEMPERATURE
0.00 C
Temperature Monitor
IoT7
MCU-PT
MCU0
Temperature Sensor
IoT0
Air Cooler
IoT5
Heating Element


DLC100
Home Gateway0
Laptop-PT
Laptop1
Fan
IoT2
Bluetooth Speaker
IoT1
Light
IoT4
MCU2
Specifications
Physical
Config
Programming
Attributes
cont (Python) - main.py
Open
New
Delete
Rename
Import
Stop
Clear Outputs
Help
Reload
Copy
Paste
Undo
Redo
Find
Replace
Zoom:
main.py
T = analogRead(1);
T = ((float(T)/1023)*200-100);
if T < 20:
digitalWrite(2, LOW);
print("AirCooler off");
digitalWrite(3, HIGH);
print("HeatingElement on");
print("Temperature now: " + str(T));
delay(3000);
elif T > 25:
digitalWrite(3, LOW);
print("HeatingElement off");
digitalWrite(2, HIGH);
print("AirCooler on");
print("Temperature now: " + str(T));
delay(3000);
else:
print("Temperature now: " + str(T));
delay(1000);
if __name__ == "__main__":
Starting cont (Python)...
Temperature now: 23.1671554252
Temperature now: 23.1671554252
Temperature now: 23.1671554252
Temperature now: 23.1671554252
Top
TEMPERATURE
23.61 C
Temperature Monitor
IoT7
MCU-PT
MCU2
Temperature Sensor
IoT8
Air Cooler
IoT5
Heating Element
IoT6


DLC100
Home Gateway0
Laptop-PT
Laptop1
Fan
IoT2
Bluetooth Speaker
IoT1
Humidifier
IoT3
Light
IoT4
TEMPERATURE
24.45 C
Temperature Monitor
IoT7
MCU-PT
MCU2
Temperature Sensor
IoT8
Air Cooler
IoT5
Heating Element
IoT6


Logical
Physical
x: 1038, y: 149
DLC100
Home Gateway0
Laptop-PT
Laptop1
Fan
IoT2
Bluetooth Speaker
IoT1
Humidifier
IoT3
Light
IoT4
TEMPERATURE
25.23 C
Temperature Monitor
IoT7
MCU-PT
MCU2
Temperature Sensor
IoT8
Air Cooler
IoT5
Heating Element
IoT6


Logical
Physical
DLC100
Home Gateway0
Laptop-PT
Laptop1
Fan
IoT2
Bluetooth Speaker
IoT1
Humidifier
IoT3
Light
IoT4
TEMPERATURE
20.94 C
Temperature Monitor
IoT7
MCU-PT
MCU2
Temperature Sensor
IoT8
Air Cooler
IoT5
Heating Element
IoT6



Код программы:

```
from gpio import *

from time import *


def main():
    pinMode(1, IN)
    pinMode(2, OUT)
    pinMode(3, OUT)
    while True:
        T = analogRead(1);
        T = ((float(T)/1023)*200-100);
        if T < 20:
            digitalWrite(2, LOW);
            print("AirCooler off");
            digitalWrite(3, HIGH);
            print("HeatingElement on");
            print("Temperature now: " + str(T));
            delay(3000);
```

```
        elif T > 25:
            digitalWrite(3, LOW);
            print("HeatingElement off");
            digitalWrite(2, HIGH);
            print("AirCooler on");
            print("Temperature now: " + str(T));
            delay(3000);
        else:
            print("Temperature now: " + str(T));
            delay(1000);

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