

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці**

ЗВІТ

з лабораторної роботи №2

**з дисципліни «Розробка застосунків інтернету речей та
сенсорних мереж»**

**Тема: «Розробка смарт-системи будинку з
використанням запрограмованих плат»**

Варіант №17

Виконав:

Студент групи ТР-12

Ковальов Олександр Олексійович

Дата здачі: 12.02.2025

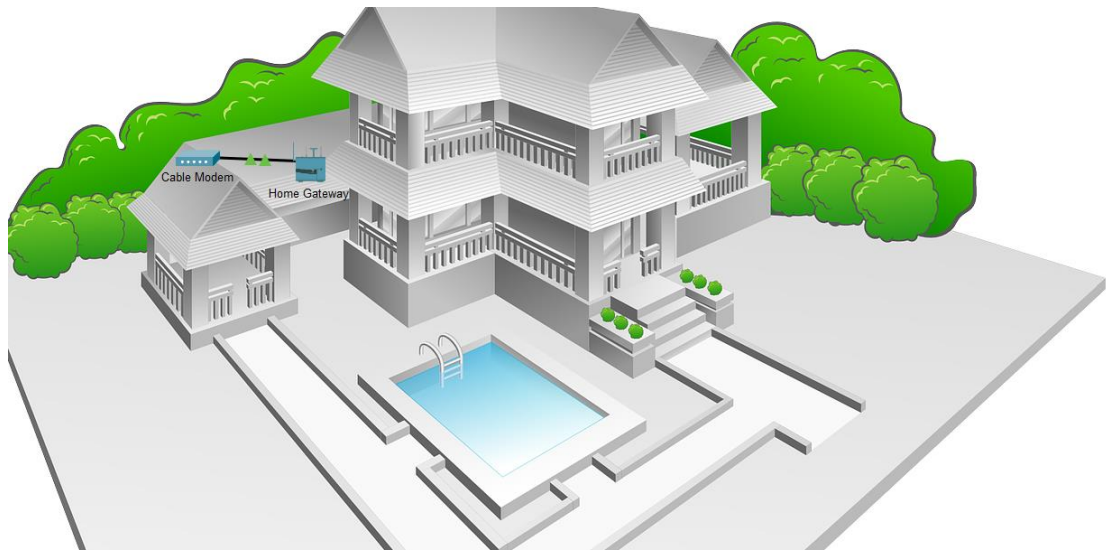
Мета роботи. Навчитись використовувати програмовані плати разом з IoT девайсами для впровадження автоматизованих систем, використовуючи набуті знання з попереднього комп'ютерного практикуму.

Індивідуальне завдання:

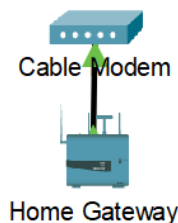
1. Додавання пристроїв та з'єднання їх із шлюзом:
 - а. Додавання планшета або смартфона для дистанційного контролю стану смарт-пристроїв і датчиків;
 - б. Додавання вікна, дверей, вентилятора, гаража, автомобіля та сенсора диму;
 - с. З'єднання пристроїв з домашнім шлюзом.
2. Підключення пристроїв до плати:
 - а. Вибір MCU з розділу «Компоненти» та з'єднання смарт-приладів з мікроконтролером.
3. Програмування мікроконтролера для контролю смарт-пристроїв.
 - а. Написання програмного коду для керування різними пристроями.
4. Тестування отриманого алгоритму:
 - а. Перевірка функціональності програмованої автоматизованої системи.
5. Розробка свого сценарію для розумних приладів:
 - а. Створення власного сценарію використання смарт-пристроїв у контексті розумного будинку.

Хід роботи.

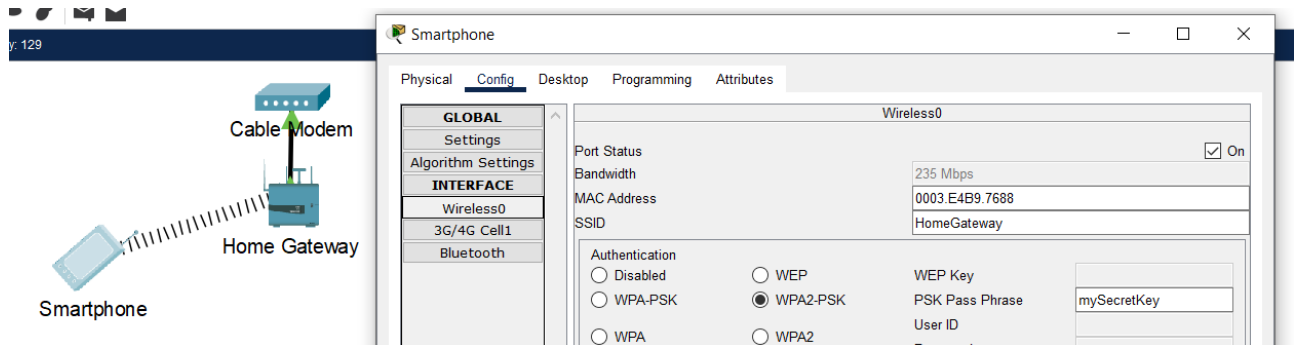
Завантажимо наданий проект-заготовку:



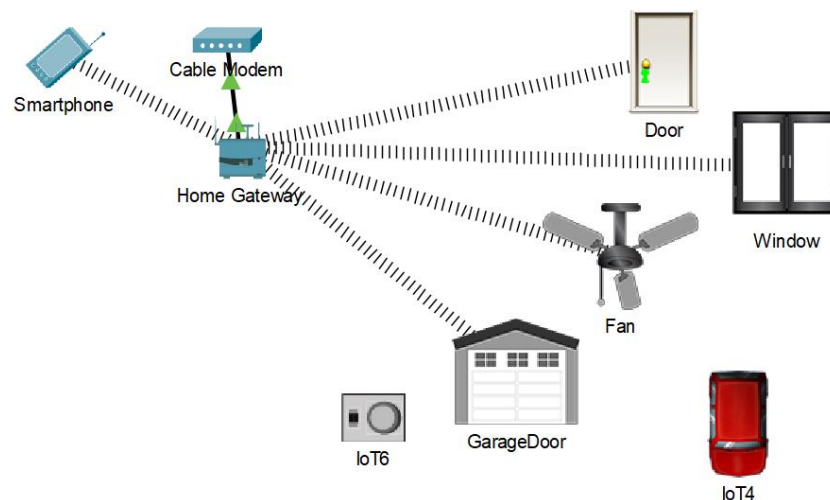
Щоб знизити візуальне навантаження, було видалено будинок на фоні.



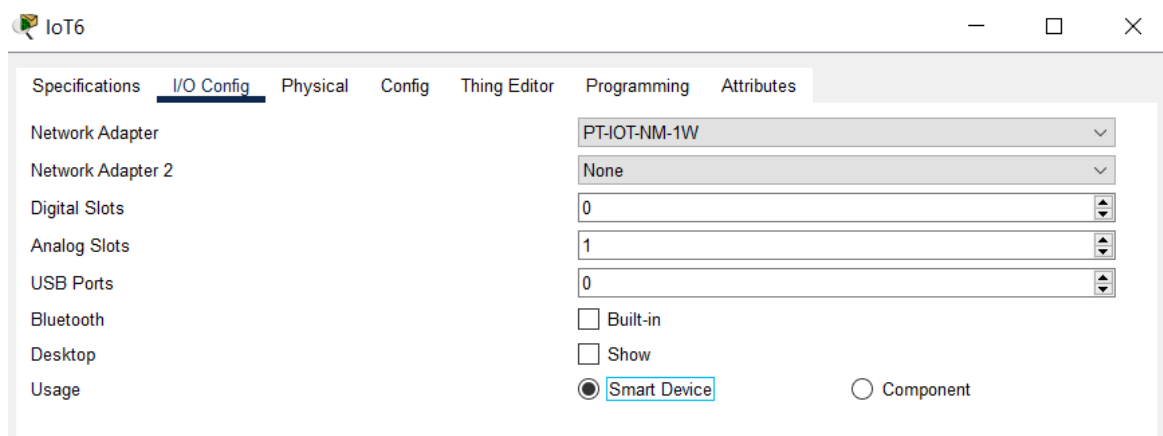
Був доданий смартфон для IoT моніторингу. Налаштований аналогічно минулій лабораторній. Різниця лише в SSID та паролі – нові значення можна знайти в налаштуваннях шлюзу, вони ж були використанні при встановленні зв'язку смартфона та шлюзу.



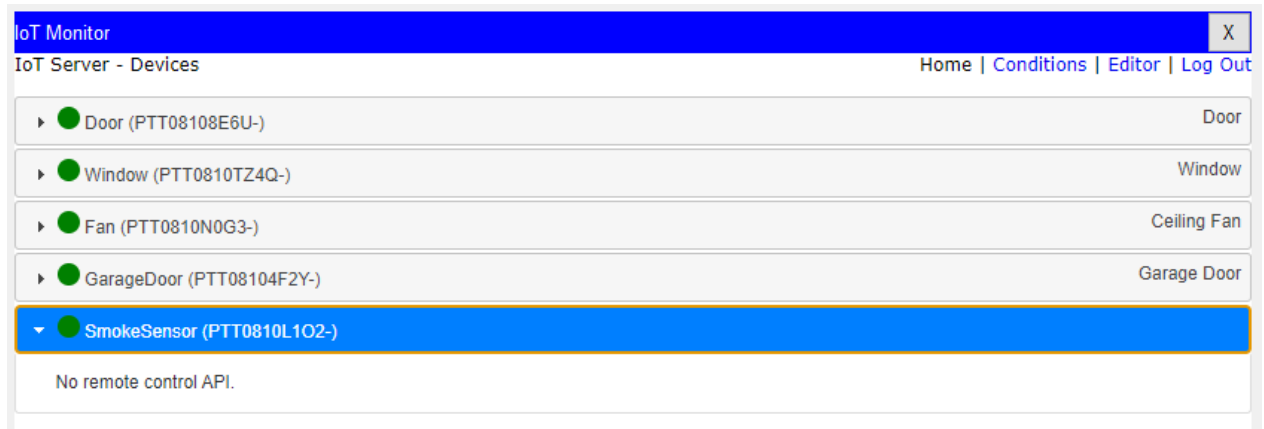
Були додані (сенсор диму, автомобіль) та налаштовані пристрої (двері, вікно, вентилятор, гараж):



Сенсор диму треба налаштовувати окремо. Спочатку треба ввімкнути «розширений» режим за допомогою кнопки «Advanced». Після цього на вкладці «I/O Config» встановити адаптер «PT-IOT-NM-1W», який дозволить під'єднатись до сенсору за допомогою Wi-Fi. Також, треба вказати що це смарт-девайс, а не компонент. Після цього налаштування повністю аналогічні іншим пристроям.

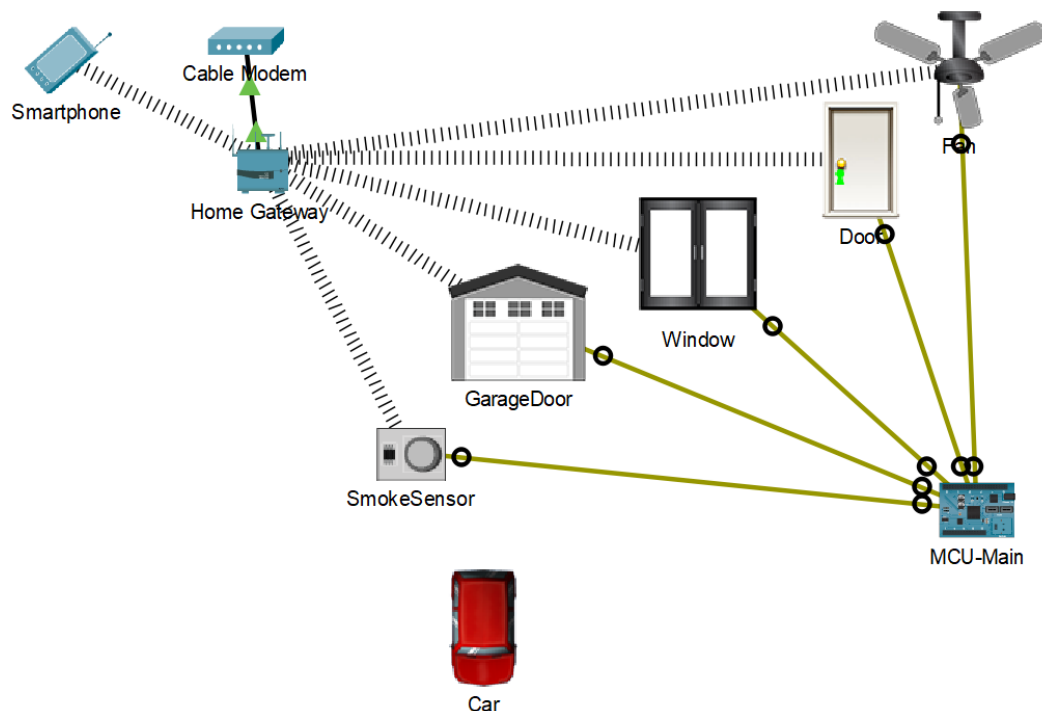


Сенсором диму не можна керувати через адмін-панель, так як до нього немає API дистанційного керування.



Була додана плата MCU з розділу «Components» – «Boards». Була під'єднана до інших девайсів за допомогою кабелю «IoT Custom Cable». Були використані такі порти:

- Плата – Сенсор: A0 – A0;
- Плата – Вентилятор: D2 – D0;
- Плата – Двері: D3 – D0;
- Плата – Вікно: D4 – D0;
- Плата – Гараж: D5 – D0;



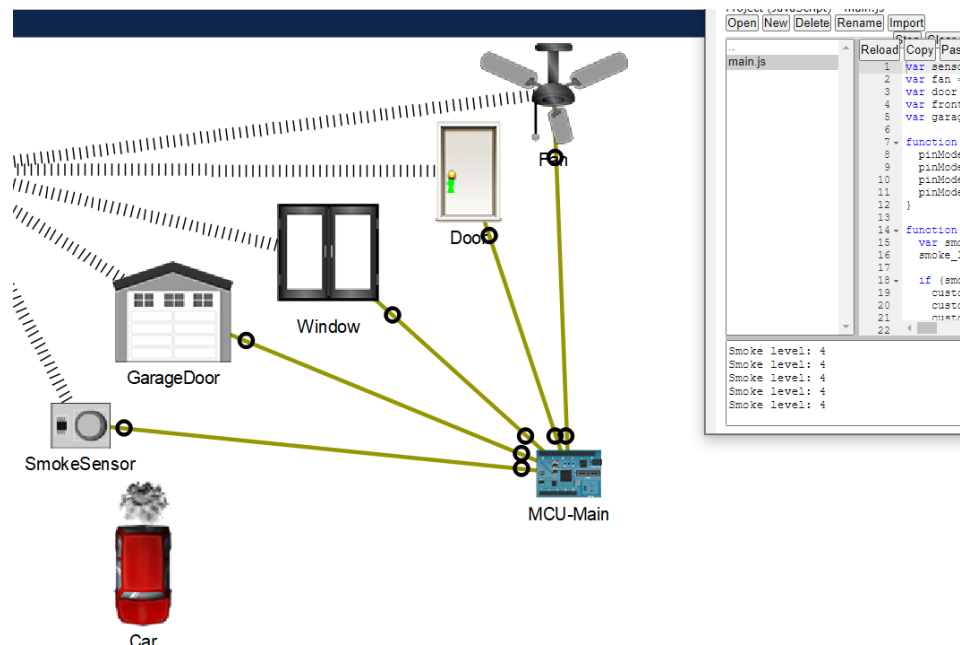
Для цих девайсів був написаний програмний код, який симулює роботу провітрювання приміщення. Тобто, машину можна завести на «Alt», при цьому буде підвищений рівень диму. В такому стані повинні відкритись вікно, двері, гараж, вентилятор повинен почати крутитись. І, відповідно, якщо диму немає – все повинно бути зачиненим, вентилятор не повинен працювати.

Був створений програмний проект. Для цього в Cisco Packet Tracer є власне середовище. На вкладці «Programming» можна створити проект певною мовою програмування, цього разу був обраний JavaScript. У файлі main.js наявні дві функції – setup та loop, перша встановлює режими пінів «на вихід». Тобто, можна записати певні дані на піни D2-D5 за допомогою функції customWrite. В функції loop читається значення на вхід з порту A0, тобто, з сенсору. Отримане значення ділиться на 10, для зручності обрахунків. Далі, перевіряється значення – якщо воно більше 10, то передається значення 1 (ON) на порти девайсів. З вентилятором ситуація дещо інша – згідно зі специфікацією, можна передати три значення: 0 (OFF), 1 (Low Speed), 2 (High Speed). Саме через це на порт-відповідник вентилятора передається значення 2. Якщо ж значення диму менше 1, то всі пристрої вимикаються. Значення диму логується.

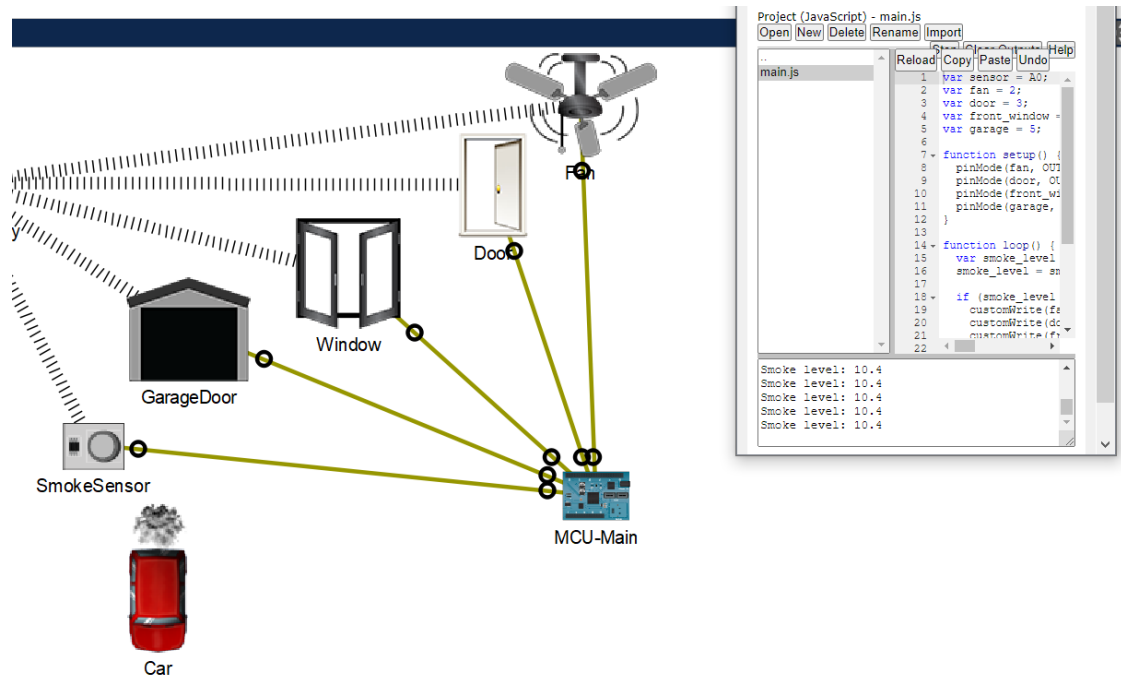
```

1 var sensor = A0;
2 var fan = 2;
3 var door = 3;
4 var front_window = 4;
5 var garage = 5;
6
7 function setup() {
8   pinMode(fan, OUTPUT);
9   pinMode(door, OUTPUT);
10  pinMode(front_window, OUTPUT);
11  pinMode(garage, OUTPUT);
12 }
13
14 function loop() {
15   var smoke_level = analogRead(sensor);
16   smoke_level = smoke_level / 10;
17
18   if (smoke_level > 10) {
19     customWrite(fan, 2);
20     customWrite(door, 1);
21     customWrite(front_window, 1);
22     customWrite(garage, 1);
23   } else if (smoke_level < 1) {
24     customWrite(fan, 0);
25     customWrite(door, 0);
26     customWrite(front_window, 0);
27     customWrite(garage, 0);
28   }
29   Serial.println("Smoke level: " + smoke_level);
30 }
31
  
```

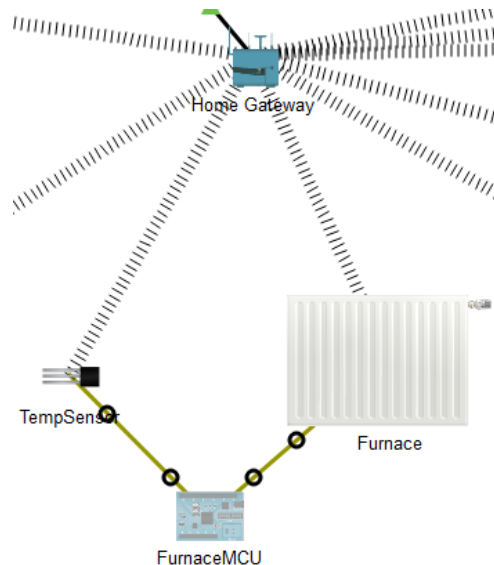
«Заводимо» машину. Низький рівень диму:



Високий рівень диму:



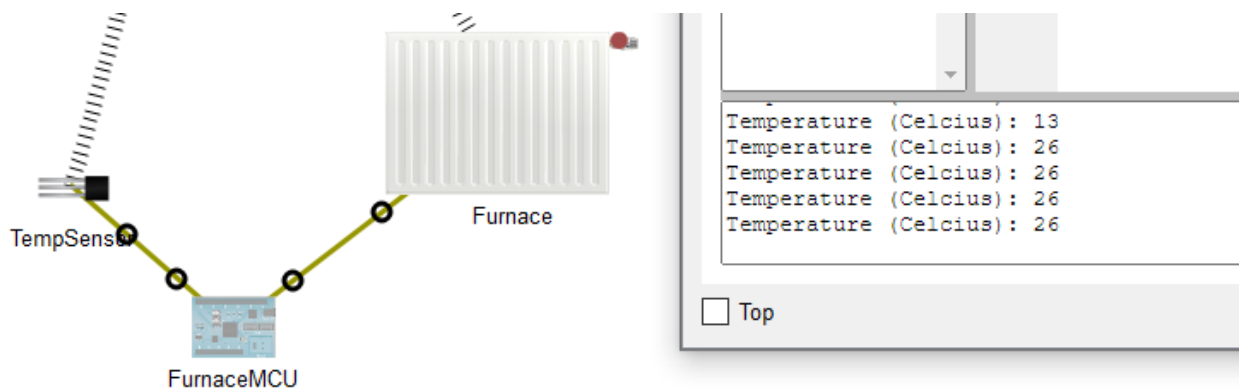
Основне завдання виконане. Було створено ще 2 сценарії.
Сценарій 1. Обігрівач, температурний сенсор:



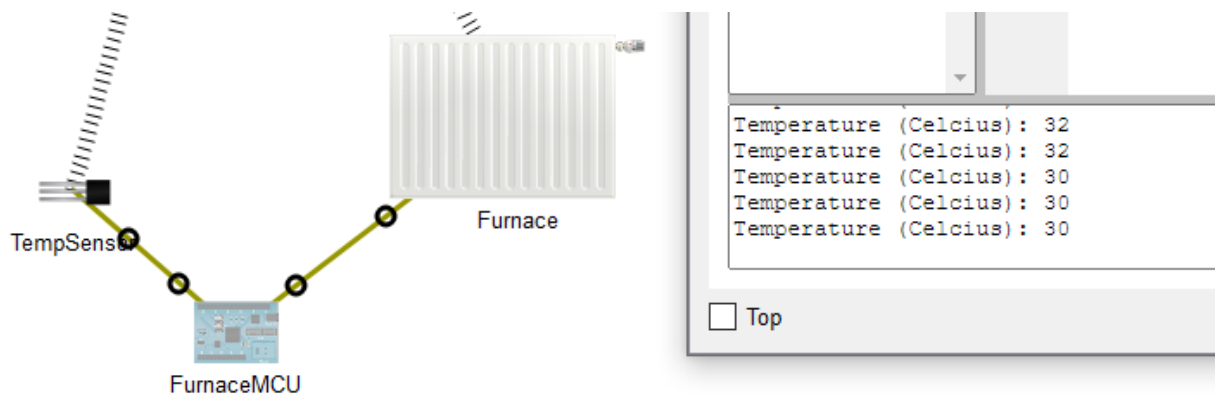
Плата підключена до сенсора за допомогою портів A0-A0, плата до обігрівача – D2-D0. Мінімальна прийнятна температура – 20°, максимальна – 27°. Спочатку, в функції setup встановлюється режим піну A2 «на вихід». В функції loop спочатку в аналоговому режимі читається значення сенсора, яке лежить в діапазоні 0-1023. Цей діапазон має відповідник -100 – 100 градусів. Таким чином, треба «замапити» отримане значення на інший діапазон. Це можна зробити за допомогою функції map. Після цього перевіряється значення температури, і якщо треба передати значення обігрівачу, це робиться за допомогою функції digitalWrite, бо, згідно документації, саме так потрібно

передавати значення цьому пристрою. Також, передається не 0 або 1, а HIGH або LOW. Після цього значення температури логується.

Низька температура:



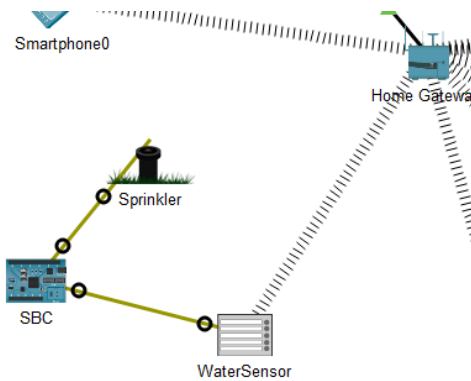
Висока температура:



Код:

```
1 var sensor = A0;
2 var furnace = 2;
3
4 var MIN_ACCEPTABLE_TEMP = 20;
5 var MAX_ACCEPTABLE_TEMP = 27;
6
7 function setup() {
8   pinMode(furnace, OUTPUT);
9 }
10
11 function loop() {
12   var temperature = analogRead(sensor);
13   temperature = Math.floor(map(temperature, 0, 1023, -100, 100));
14
15   if (temperature < MIN_ACCEPTABLE_TEMP) {
16     digitalWrite(furnace, HIGH);
17   }
18   if (temperature > MAX_ACCEPTABLE_TEMP) {
19     digitalWrite(furnace, LOW);
20   }
21
22   Serial.println("Temperature (Celcius): " + temperature);
23 }
```

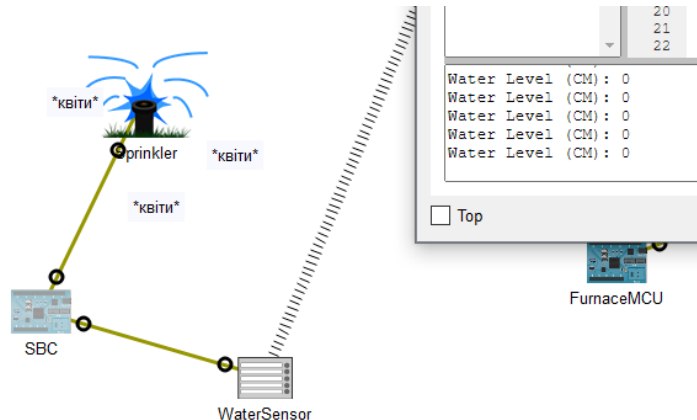
Сценарій 2 стосується поливу. Був використаний сенсор рівня води, розбризкувач, та плата SBC (хоча в ній тут потреби не було, це зроблено заради експерименту).



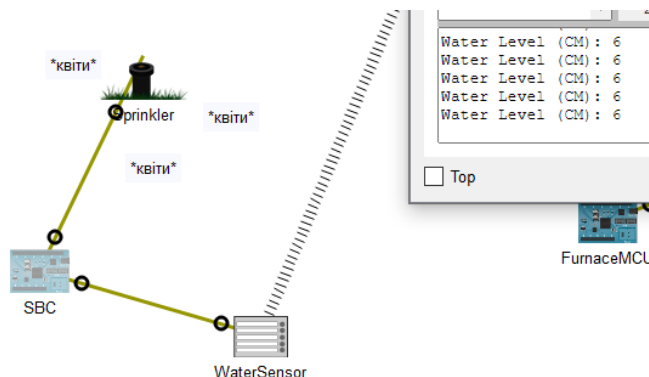
Плата під'єднана до сенсору за допомогою портів D0-A0, і до розбризкувача – D1-D0.

При використанні плати SBC також треба встановлювати піни не тільки в режим виходу, але й входу. Це відбувається в функції `setup`. Далі, в функції `loop`, читається рівень води. Він мапиться з 0-1023 на діапазон 0-20 (згідно специфікації), і далі, аналогічно минулому прикладу, порівнюються рівні. За допомогою `customWrite` передаються значення розбризкувачу. Значення рівня води логуються.

Низький рівень води:



Високий рівень води:



Код:

```
1 var sensor = 0;
2 var sprinkler = 1;
3
4 var MIN_ACCEPTABLE_LEVEL_CM = 3;
5 var MAX_ACCEPTABLE_LEVEL_CM = 12;
6
7 function setup() {
8   pinMode(sensor, INPUT);
9   pinMode(sprinkler, OUTPUT);
10 }
11
12 function loop() {
13   var water_level_cm = analogRead(sensor);
14   water_level_cm = Math.floor(map(water_level_cm, 0, 1023, 0, 20));
15
16   if (water_level_cm < MIN_ACCEPTABLE_LEVEL_CM) {
17     customWrite(sprinkler, 1);
18   }
19   if (water_level_cm > MAX_ACCEPTABLE_LEVEL_CM) {
20     customWrite(sprinkler, 0);
21   }
22
23   Serial.println("Water Level (CM): " + water_level_cm);
24 }
```

Висновок: У ході виконання лабораторної роботи було розроблено смарт-систему будинку з використанням запрограмованих плат. Було додано та налаштовано різні пристрої, з'єднано їх із домашнім шлюзом і запрограмовано мікроконтролер для керування системою. Проведене тестування підтвердило правильність роботи алгоритмів, що дозволило реалізувати автоматизовані сценарії, такі як контроль диму, температури та рівня води. Отримані навички можуть бути застосовані для подальшої розробки IoT-рішень.

Контрольні питання:

1. *Що таке розумний будинок і які основні переваги використання автоматизованих систем у такому будинку?*
Розумний будинок – це житлове приміщення, оснащене автоматизованими системами керування освітленням, кліматом, безпекою, мультимедіа та іншими функціями. Основні переваги включають зручність, енергоефективність, підвищену безпеку, дистанційне керування та можливість адаптації до потреб користувача.
2. *Які основні елементи становлять систему розумного будинку?*
Система розумного будинку складається з контролера (центру керування, адмін-панелі), датчиків (руху, температури, диму тощо), виконавчих пристроїв (реле, клапани, розетки), комунікаційних модулів (Wi-Fi, Zigbee, Z-Wave), програмного забезпечення та інтерфейсу для користувача.
3. *Як з'єднуються смарт-пристрої з домашнім шлюзом? Чому це важливо?*
Смарт-пристрої з'єднуються з домашнім шлюзом через бездротові протоколи (Wi-Fi, Zigbee, Bluetooth, Z-Wave) або дротові з'єднання (Ethernet). Це важливо, оскільки шлюз координує обмін даними між пристроями та забезпечує їх підключення до Інтернету для віддаленого керування.

4. *Яка роль мікроконтролерів у керуванні смарт-пристроями? Дайте приклади їхнього використання.*

Мікроконтролери відіграють ключову роль у керуванні смарт-пристроями, виконуючи команди та обробляючи дані з датчиків. Наприклад, ESP8266 та ESP32 використовуються для підключення пристроїв до Wi-Fi, а Arduino керує розумним освітленням або системами автоматичного поливу.

5. *Які основні кроки програмування мікроконтролера для керування смарт-пристроями?*

Програмування мікроконтролера для смарт-пристроїв включає вибір апаратної платформи, написання коду (часто на C/C++ або MicroPython), завантаження прошивки, тестування роботи та інтеграцію з системою керування.

6. *Які методи і критерії використовуються для тестування автоматизованої системи розумного будинку?*

Тестування автоматизованої системи розумного будинку включає перевірку функціональності (реакція пристроїв на команди), продуктивності (швидкість обробки запитів), сумісності (робота з іншими пристроями), безпеки (захист від несанкціонованого доступу) та енергоефективності.

7. *Які заходи безпеки і конфіденційності необхідно врахувати при проектуванні смарт-систем будинку?*

При проектуванні смарт-систем необхідно враховувати безпеку мережі (шифрування, VPN, оновлення ПЗ), захист даних користувача (анонімізація, локальне зберігання), обмеження доступу (авторизація, контроль прав) та стійкість до атак (захист від DDoS, ізоляція IoT-пристроїв від основної мережі).