

**Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»**

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня магістра
(бакалавра, спеціаліста, магістра)

студента Шерстюка Дениса Миколайовича
(ПІБ)

академічної групи 123М-19-1
(шифр)

спеціальності 123 «Комп'ютерна інженерія»
(код і назва спеціальності)

за освітньо-професійною програмою «Комп'ютерна інженерія»
(офіційна назва)

**на тему Обґрунтування структури та параметрів комп'ютерної системи
«розумного опалення» приміщень навчального закладу**
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Цвіркун Л.І.			
розділів:				
теоретичний розділ	проф. Цвіркун Л.І.			
синтез системи	доц. Ткаченко С.М.			
розроблення програмного забезпечення	ас. Бешта Л.В.			
експериментальний розділ	доц. Ткаченко С.М.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

**Дніпро
2020**

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії

(повна назва)

_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

«__» _____ 2020 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістр
(бакалавра, спеціаліста, магістра)

студенту Шерстюку Д.М. академічної групи 123М-19-1
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньою-професійною програмою 123 «Комп'ютерна інженерія»
(офіційна назва)

на тему Обґрунтування структури та параметрів комп'ютерної системи “розумного опалення” приміщень навчального закладу

затверджену наказом ректора НТУ «Дніпровська політехніка» від 22.10.2020 № 888-С

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел сформулювати наукове завдання, конкретизувати предмет та мету досліджень	21.09.2020
Теоретичний	Обґрунтувати теоретичну базу розв'язання наукового завдання, якому присвячено роботу	30.10.2020
Синтез системи	Розробка комп'ютерної системи	12.11.2020
Розроблення програмного забезпечення	Розробка програмного забезпечення	26.11.2020
Експериментальний розділ	Проведення і обробка результатів експериментів	06.12.2020

Завдання видано _____
(підпис керівника)

проф. Цвіркун Л. І.
(прізвище, ініціали)

Дата видачі 07 вересня 2020 р.Дата подання до екзаменаційної комісії 10.12.2020 р.

Прийнято до виконання _____
(підпис студента)

Шерстюк Д.М.
(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 85 с., 39 рис., 6 табл., 13 джерел, 1 додаток.

Об'єкт розробки: комп'ютерні аудиторії університету «Дніпровська політехніка» кафедри «інформаційних технологій та комп'ютерної інженерії».

Мета роботи: створення системи опалення в Національному технічному університеті «Дніпровська політехніка» на кафедрі «інформаційних технологій та комп'ютерної інженерії» на основі технології «розумний дім» та аналіз структури і параметрів системи.

У пояснювальній записці було проведено аналіз роботи попередника комп'ютерної системи «розумного опалення», огляд відомих існуючих подібних систем «розумного дому». На базі проаналізованих частин було сформульовано задачу на розробку системи «розумного опалення». Наступним кроком було вирішення наукового завдання результатом якого є створення математичної моделі системи контролю «розумного опалення». Після вирішення наукового завдання було наведено технічні вимоги, призначення розробки для подальшого створення системи контролю комп'ютерної системи. Далі виконувалася розробка програмного забезпечення з описом функціональних можливостей та технічних засобів для підтримки програми, створенням алгоритмів з обґрунтуванням технічних характеристик програми. Кінцевим кроком було проведення експерименту на базі створеної моделі системи масового обслуговування.

«РОЗУМНИЙ ДІМ», «РОЗУМНЕ ОПАЛЕННЯ», МОДЕЛЬ, НТУ «ДП»,
ZIGBEE, XIAOMI, HOME ASSISTANT, SIMULINK

ЗМІСТ

	Стор.
Перелік умовних позначень, скорочень та термінів	6
Вступ	7
1 Стан питання та постановка завдання	12
1.1 Стисла характеристика галузі та застосування систем «розумного дому» для опалення	12
1.2 Огляд існуючих систем	15
1.3 Постановка завдання дослідження	30
2 Теоретичний розділ	31
2.1 Загальна характеристика комп'ютерних класів НТУ «ДП»	31
2.2 Структура об'єкту дослідження	33
2.3 Принцип функціонування об'єкту	35
2.4 Обґрунтування і вибір методів дослідження	35
2.4.1 Метод аналізу і синтезу	35
2.4.2 Метод порівняння	37
2.4.3 Метод математичного моделювання	38
2.5 Висновки по розділу	46
3 Синтез системи контролю	47
3.1 Формулювання технічних вимог до системи контролю «розумного опалення»	47
3.1.1 Вимоги до реалізації системи	47
3.1.2 Вимоги до видів забезпечення	47
3.2 Розробка схеми функціональної структури	48
3.3 Розробка функціональної схеми	48
3.4 Вибір та обґрунтування застосування апаратних засобів	49
3.5 Висновки по розділу	57
4 Розробка програмного забезпечення «розумного опалення» комп'ютерних класів	58

4.1	Призначення й область застосування програмного забезпечення	58
4.2	Обґрунтування технічних характеристик програми	58
4.2.1	Постановка завдання на розробку програми	58
4.2.2	Опис алгоритму функціонування програми	59
4.3	Опис розробленої програми	60
4.3.1	Загальні відомості	60
4.3.2	Функціональне призначення	61
4.3.3	Опис логічної структури програми	62
4.3.4	Використовувані технічні засоби	74
4.3.5	Виклик і завантаження	74
4.4	Висновки по розділу	74
5	Експериментальний розділ	75
5.1	Мета і завдання експерименту	75
5.2	Методика експерименту	75
5.3	Оцінка результатів експерименту	75
5.4	Аналіз отриманих результатів	80
5.5	Висновки по розділу	80
	Висновки	81
	Перелік посилань	83
	Додаток А. Текст програми комп'ютерної системи контролю «розумного опалення»	85

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

НТУ «ДП» – Національний технічний університет «Дніпровська політехніка»

«ІТКІ» – Кафедра «Інформаційних технологій та комп'ютерної інженерії»

ОС – операційна система

YAML - Yet Another Markup Language

Хаб – мережеве пристрій, призначений для об'єднання кількох пристроїв Ethernet of Things в загальний сегмент мереж

ВСТУП

У 1777 році французький інженер М. Боннеман винайшов і застосував для обігріву інкубаторів першу водну систему опалення з природною циркуляцією, основні принципи та інженерні рішення якої знайшли застосування в опаленні помешкань тоді і застосовуються до цих пір.

У 1834 першої в Росії системою водяного опалення з природною циркуляцією стала система гірського інженера, професора П. Г. Соболевського. У 1875 році з'явилася перша не тільки в Росії, але і в Західній Європі квартира з окремою системою водяного опалення з використанням плоских опалювальних приладів, зроблених у вигляді пілястр. Підігрів води відбувався в невеликому нагрівачі, встановленому в кухонному вогнищі.

В період 1855-57 рр. російський промисловець Франц Карлович Сан-Галлі винайшов принципово нове для того часу обігрівальне пристрій - радіатор водяного опалення. Перші екземпляри радіаторів опалення представляли собою товсті труби з вертикальними дисками. Сан-Галлі назвав свій винахід «хайцкёрпер» (гаряча коробка), а пізніше придумав для нього російська назва «батарея». Батареї, вироблені на чавуноливарному заводі Сан-Галлі, швидко завоювали популярність в Петербурзі, а потім і по всьому світу.

До 1917 року в Росії багато прибуткові будинки, в основному елітні, оснащувалися системами водяного і парового опалення. Подача тепла в будинок здійснювалася від котельні, розташованої в підвалі або прибудові. На фабриках застосовувалося опалення відпрацьованою паром, який використовувався для роботи парових машин. У той же час, значна частина міських будівель і все індивідуальні будинки в містах, селах і селах опалювалися печами на дровах чи інших місцевих видах палива.

Широке впровадження систем центрального опалення почалося в епоху індустріалізації СРСР і супутньої їй урбанізації. У цей час формуються основні риси систем центрального опалення. При новозведених промислових

підприємствах будуються житлові райони («соцмістечка») з багатоквартирними будинками, оснащеними радіаторами водяного опалення.

На початок 1950-х років більшість сталінських будинків були оснащені системами центрального водяного опалення, які підключалися до котелень промислових підприємств, ТЕЦ або невеликим районним котельням. При неможливості підключення до центрального опалення окремі будинки мали власні котельні, а деякі малоповерхові будинки проектувалися з варіантом пічного опалення.

Остаточне впровадження центрального опалення багатоквартирних будинків сталося з початком масового житлового будівництва хрущовок. Поряд з підключенням будинків до ТЕЦ і котелень підприємств, в нових житлових масивах зводилися районні котельні. З середини 1960-х по початок 1990-х розвиток систем опалення в СРСР йшло в напрямку подальшої централізації. Невеликі котельні закривалися, а вдома підключалися до великих котелень та ТЕЦ. Проводилися закріплення систем опалення та впровадження закритої системи тепlopостачання з тепловими пунктами [1].

За роки служби в трубах і радіаторах накопичується бруд і продукти корозії металу, які значно знижують тепловіддачу. Старі батареї не здатні повноцінно віддавати тепло в приміщення, їх ефективність становить від сили 70%, а для пофарбованих товстим шаром фарби – всі 50% від можливого.

Розрахунковий термін експлуатації обладнання системи тепlopостачання становить 15 років для опалювальних приладів і 25 років для трубопроводів, тоді як обладнання експлуатується набагато довше, часто без дотримання рекомендацій заводів - виробників. Рівень втрат в теплових мережах країни досягає 40-45%, що майже в п'ять рази вище, ніж в Європі.

У зв'язку з фізичним зносом і естетично застарілим існуючим обладнанням системи тепlopостачання необхідно встановлювати обладнання, яке відповідає сучасним вимогам енергоефективності та надійності [2].

Система розумний будинок – набір компонентів, що забезпечують інтелектуальне управління всіма системами (опалення, водопостачання, відеоспостереження, освітлення, кондиціонування, відкривання воріт, дверей, штор, відео- і аудіотехніка, побутова техніка та інші) будинку або будівлі. Більшість робіт, які раніше виконувалися власником вручну, система бере на себе. Підтримка оптимальної температури в приміщенні, своєчасне включення вуличного освітлення, підтримання температури бойлера, фіксація несанкціонованого проникнення в житло, захист від пожежі, затопило, отруєння чадним газом. Ось тільки деякі з завдань, які автоматично вирішуються установкою системи розумний будинок.

Мало хто з власників будинків самостійно займаються своєю системою опалення. Зазвичай котли, насоси, автоматика розташовуються десь в підвалі, в котельні або навішені на стіну. Опалювальна техніка привертає увагу тільки коли потрібно провести техобслуговування і коли в будинку стає холодно. Це вимагає від власника певних зусиль і виконання налаштувань, що створює незручності. У розумному будинку початок опалювального сезону і, відповідно, температура в приміщенні залежить від налаштувань кривої опалення і від власника не вимагається додаткового участі. Завжди тепло і комфортно з мінімальним енергоспоживанням. "Інтелектуальна" техніка знижує вплив людського фактора. До того ж, вона порівняно недорога і добре вбудовується у вже існуючі системи [3].

Мета і завдання дослідження. *Метою роботи є створення системи опалення в Національному технічному університеті «Дніпровська політехніка» на кафедрі «інформаційних технологій та комп'ютерної інженерії» на основі технології «розумний дім» та аналіз структури і параметрів системи.*

Для досягнення поставленої мети необхідно вирішити такі завдання:

- скласти вимоги до функціонування системи;
- розробити функціональну структуру та схему;

- обрати апаратне забезпечення;
- розробити програмне забезпечення для автоматизації системи.

Об'єкт дослідження – є комп'ютерні класи 1-го корпусу учбового закладу НТУ «Дніпровська політехніка» кафедри «Інформаційних технологій та комп'ютерної інженерії» 1/130 та 1/131.

Предмет дослідження – створення системи контролю «розумного опалення» для поліпшення умов проведення класних занять.

Методи дослідження. Для досягнення поставленої мети використовувалися методи аналізу і синтезу, порівняння, Метод математичного моделювання.

Наукові положення:

1. Встановлено, що одне з найкращих рішень для опалення кабінетів навчальних закладів – це автоматична система з основним опаленням за допомогою інфрачервоних панелей на базі «розумного дому».

2. Створення системи «розумного опалення» за допомогою апаратного забезпечення різних представників відбувається при залученні до системи програмного додатку Home Assistant, як основний програмний концентратор, що дає змогу активувати й інші програмні додатки з відкритим кодом.

Наукові результати:

1. Одержані нові теоретичні відомості щодо створення математичних моделей систем з керованим опаленням.

2. Запропонований метод автоматичного контролю за допомогою додатку Home Assistant створює більш конфіденційність даних та системи «розумного дому».

3. Обґрунтовано застосування програмного додатку з відкритим кодом для невеликих або вагомих змін системи, що надає змогу у використанні не тільки компонентів екосистеми обраного представника розробленого апаратного забезпечення.

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій підтверджуються тим, що в роботі використані сучасні методи систем опалення, вибору новітнього обладнання, здійснення розробки програмного забезпечення, використання методу аналізу і синтезу, методу порівняння та методу математичного моделювання.

Практичне значення отриманих результатів полягає в розробці автоматичного контролю системи опалення в приміщеннях навчальних закладах.

1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ

1.1 Стисла характеристика галузі та застосування систем «розумного дому» для опалення

Система розумний будинок - набір компонентів, що забезпечують інтелектуальне управління всіма системами (опалення, водопостачання, відеоспостереження, освітлення, кондиціонування, відкривання воріт, дверей, штор, відео- і аудіотехніка, побутова техніка та інші) будинку або будівлі. Більшість робіт, які раніше виконувалися власником вручну, система бере на себе. Підтримка оптимальної температури в приміщенні, своєчасне включення вуличного освітлення, підтримання температури бойлера, фіксація несанкціонованого проникнення в житло, захист від пожежі, затопило, отруєння чадним газом. Ось тільки деякі з завдань, які автоматично вирішуються установкою системи розумний будинок.

Мало хто з власників будинків самостійно займаються своєю системою опалення. Зазвичай котли, насоси, автоматика розташовуються десь в підвалі, в котельні або навішені на стіну. Опалювальна техніка привертає увагу тільки коли потрібно провести техобслуговування і коли в будинку стає холодно. Це вимагає від власника певних зусиль і виконання налаштувань, що створює незручності. У розумному будинку початок опалювального сезону і, відповідно, температура в приміщенні залежить від налаштувань кривої опалення і від власника не вимагається додаткового участі. Завжди тепло і комфортно з мінімальним енергоспоживанням. "Інтелектуальна" техніка знижує вплив людського фактора. До того ж, вона порівняно недорога і добре вбудовується у вже існуючі системи.

Елементи системи розумний будинок, стосовно до опалення, це термостатичні головки і вентилі для радіаторів, програматори і термостати для регулювання температури в приміщенні. Такі пристрої можуть мати різну функціональність. Простіші пристрої підтримують задану температуру, економлячи при цьому близько 20% витрат на опалення. Більш складні

програмуються, підтримують певну температуру в залежності від часу доби і дня тижня. Найбільш «просунуті» пристроями можна керувати за допомогою смартфона через інтернет, що виключає перевитрата енергії і втрату комфорту, навіть якщо власник вибивається з графіка. Часто такі пристрої мають додаткову функціональність, наприклад автоматичне зниження температури, коли нікого немає вдома, відключення опалення на період провітрювання.

Завдяки впровадженню цифрових технологій дуже зручним, особливо для будинків великої площі, від 150-200 кв. метрів, є погодозалежне регулювання. Система включає / відключає котел в залежності від зовнішньої температури і завчасно прогріває приміщення або економить паливо, підтримуючи всередині стабільну температуру.

Техніка для розумного будинку - це щось більше, ніж просто гра. Вона полегшує обслуговування системи опалення, підвищує комфорт і допомагає уникнути непотрібних втрат енергії. У той же час, вона створює більш високий рівень безпеки і піклується про те, щоб розпізнати помилки ще до їх виникнення.

За високий рівень комфорту відповідає автоматичне керування системою опалення та поверхнями нагріву. Завдяки цьому в житлових приміщеннях завжди підтримується оптимальна температура навіть без ручної настройки користувачем. Якщо обладнанням можна управляти, не перебуваючи вдома, то користувач тільки виграє. Приклад - відпустка взимку. Якщо господарі відсутні протягом однієї або двох тижнів, навряд чи варто залишати систему опалення працювати на повну потужність. Якщо перед від'їздом вимкнути систему, після повернення в будинку буде холодно і незатишно. Але це не стосується системи опалення в розумному будинку. Адже її можна включити під час дороги додому через смартфон і забезпечити собі тим самим приємний теплий прийом.

Крім комфорту, програматори, термостати, термоголовки також економлять багато енергії - і це без дорогих заходів по реконструкції

будинку. Таке стало можливим, тому що розумні алгоритми вивчають особливості будинку і потреби користувача. Вони зіставляють обидва чинники, враховують температуру зовнішнього повітря і піклуються про те, щоб система опалення ніколи не виробляла більше тепла, ніж потрібно. Таким чином, система розумний будинок зменшує теплову потужність пристроїв, якщо господарі не вдома або на вулиці встановилася сонячна погода.

До речі: потенціал економії "інтелектуальних" термостатів і програматорів падає, якщо будинок має якісну теплоізоляцію. В таких умовах він втрачає мало тепла і зниження температури всередині приміщення менше впливає на витрати на опалення

Техніка для розумного будинку призначена не тільки для опалення, але і освітлення, управління побутовими приладами, відеоспостереження та іншим. Вся вона повинна бути зв'язана. Оскільки виробники розробляють все «під себе», сьогодні існують різноманітні стандарти передачі даних. Найвідоміші з них: Wi-Fi, ZigBee, Z-Wave, EnOcean і KNX (дротового або бездротового). Роблячи вибір на користь розумного будинку, слід порівняти плюси і мінуси цих стандартів і вибрати для себе той, який має широкі функціональні можливості. Зробимо невеликий огляд:

- Wi-Fi (бездротової, використовується в багатьох домогосподарствах, енергоємний з сильно перевантаженими діапазонами частот);
- ZigBee (бездротовий, низьке споживання енергії, але через перевантажені діапазони частот не дуже стійкий до перешкод)
- Z-Wave (бездротова передача даних в діапазоні частот з низьким рівнем навантаження, енергозберігаючий);
- EnOcean (бездротова передача даних в діапазоні частот з низьким рівнем навантаження. Робота без підключення до мережі без батарей із застосуванням технології "Energy Harvesting" надзвичайно економна);

- KNX (дротова передача даних без перешкод для забезпечення чіткого зв'язку і високого рівня безпеки; дорого коштує і складна для установки);

Діапазон функцій і робочі параметри - велика функціональність створює більший комфорт, дозволяє значно економити:

- надійність - краще купувати елементи системи розумний будинок від відомого і перевіреного виробника, наприклад, SALUS, WATTS. Більш надійними фахівці вважають кабельні системи, але вони більш складні в монтажі, особливо, в уже експлуатується будинку.
- масштабованість і вартість.

В Інтернеті легко порівняти можливості і ціни систем «Розумний дім» від різних виробників. Рекомендується все ж купувати таку від відомого виробника. Це регулярні оновлення, більший вибір додаткових пристроїв, простіше сервісне обслуговування [3].

1.2 Огляд існуючих систем

Існують системи автоматизації «Розумний дім» за основними ознаками: тут є три основних параметри:

- централізовані або децентралізовані;
- з відкритим протоколом або з закритим протоколом;
- Дротові або бездротові технології.

Централізовані системи автоматизації. Суть схеми роботи централізованої системи управління будинком в тому, що тут має місце програмування одного центрального елемента (логічного модуля). Найчастіше це контролер Розумного Дому з багатьма виходами. Для конкретного об'єкта пишеться програма, за допомогою якої здійснюється управління виконавчими пристроями і системами. Потім ця програма заливається в контролер Розумного Дому. Дана схема Розумного Дому дає можливість використовувати мультизадачні сценарії з використанням широкого вибору обладнання. Централізовані системи автоматизації можуть

бути як провідні, так і бездротові. На рисунку 1.1 схематично показані можливі елементи розумного будинку.

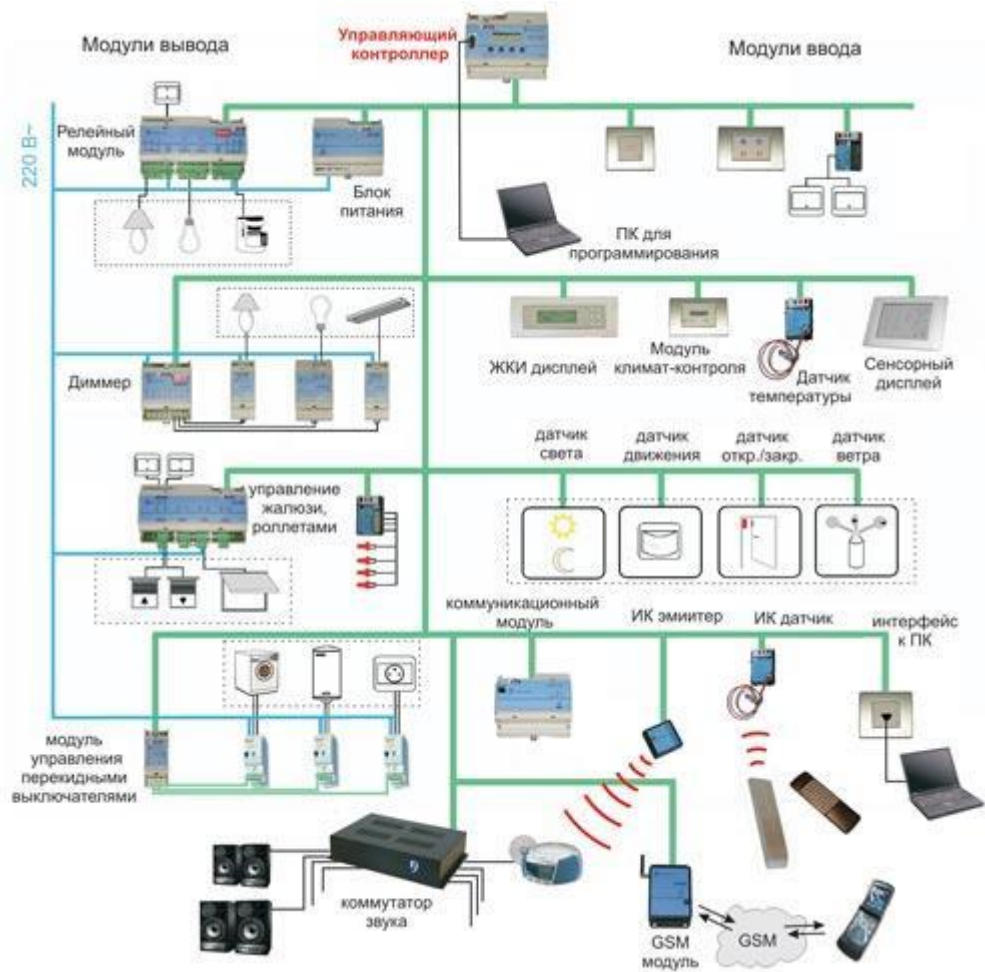


Рисунок 1.1 – Схема централізованої системи

Переваги централізованої системи автоматизації:

- єдиний інтерфейс для управління всіма системами;
- можливість створення мультизадачних сценаріїв;
- підключення до системи практично будь-якого обладнання.

Слід врахувати і деякі особливості при установці централізованих систем:

- уразливість. Вся система управління залежить від центрального модуля. Модуль виходить з ладу - завмирає вся система;
- людський фактор. Якщо втрачено зв'язок з програмістом, який спочатку писав код для вашої системи, то внести зміни в систему

іншій людині буде досить важко. Швидше за все доведеться писати програму заново.

Децентралізовані системи автоматизації. У децентралізованій системі кожне виконавчий пристрій в будинку містить в собі мікропроцесор, який в свою чергу містить в собі незалежну пам'ять. В цьому і полягає досить висока надійність даних систем.

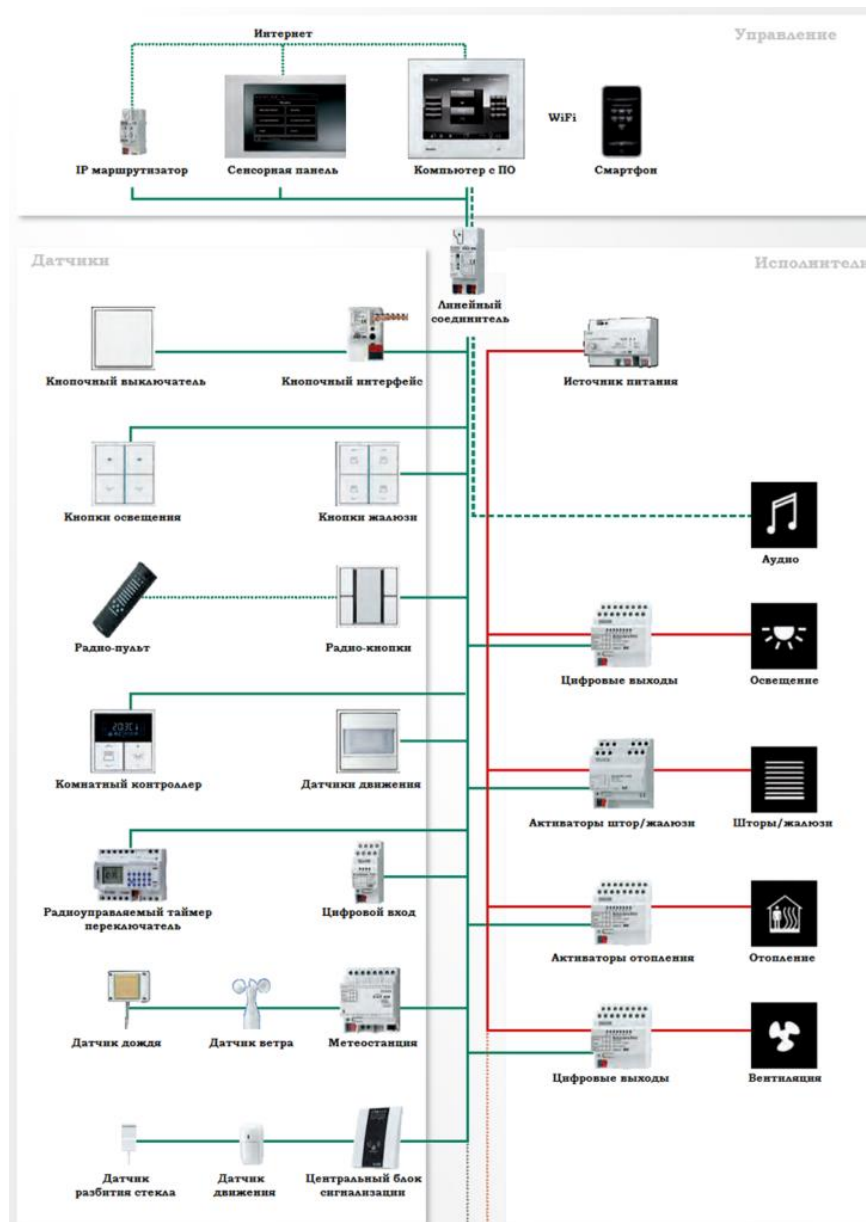


Рисунок 1.2 – Схема децентралізованої системи

Відповідно розуміємо: при поломці одного пристрою вся система працює коректно, природно, за винятком тих приладів, які були підключені

до пристрою, який вийшов з ладу. Прикладом децентралізованої системи є цифрові будинки, побудовані на технологіях KNX.

Переваги децентралізованої системи автоматизації "Розумний Дім»:

- надійність. Пристрої в системі не залежать один від одного. Наявність незалежної пам'яті;
- популярність. У більшій частині це стосується стандарту KNX. Тут не виникне питань з пуском / налагодженням, сервісним обслуговуванням, так як сертифікованих фахівців з даного устаткування в нашій країні (та й в нашій компанії) досить;
- дизайн. Вибір модельного ряду електрофурнітури, а також датчиків досить широкий як у плані дизайну, так і в плані його функціоналу.

Слід врахувати і деякі особливості при установці децентралізованих систем, точніше основна особливість, на яку ми хочемо звернути увагу, це великий обсяг щитового обладнання. Будьте готові до щитка від п'ятдесяти до ста посадочних місць.

Системи автоматизації з відкритим протоколом. Якщо взяти нами всіма улюблений і шанований протокол KNX, то він є відкритим, а це значить, що величезна кількість виробників обладнання для цифрових будинків роблять пристрої, які працюють саме на цій мові. Асоціація KNX тестує всі вироблені пристрою Розумного будинку, а знак KNX на пристрої дає гарантію високої якості товару. Хоча і вартість звідси впливає вище, ніж у систем на закритих протоколах. За гарантію якості доведеться доплатити. Та й розгулятися інженерам в їх креативні ідеї при створенні нових пристроїв не особливо вийде через дотримання стандартів асоціації.

Системи автоматизації з закритим протоколом. Дані системи, реалізовані на власних закритих протоколах, існують в основному для того, щоб спростити процес програмування. Та й фінансова витратна частина у виробників таких систем поменше буде – не потрібно комусь платити за схвалення свого продукту. Підкреслимо, що обладнання, яке працює на закритому протоколі, крім рідного виробника ніхто більше не робить. З

одного боку це дає ніким не обмежений політ фантазії виробника, з іншого боку – це повна залежність споживача перед цим же виробником.

Провідні системи автоматизації. Сенс провідної системи полягає в наступному: всі керуючі елементи розумного будинку - перемикачі, панелі управління, всілякі датчики і пристрої - пов'язані між собою єдиною провідний шиною. По цій шині вони віддають команди (сигнали, телеграми) виконавчих пристроїв (реле, актуаторами), які розташовані в щиті. Роль шини може виконувати спеціальний кабель, а іноді і звичайна вита пара.

Переваги провідної системи автоматизації:

- надійність. Передача сигналу по дроту завжди буде надійніше бездротової передачі;
- швидкість. Швидкість передачі телеграм по дроту досить висока, щоб не викликати у користувача дискомфорт, причиною якого часто буває значна затримка. А слідом за дискомфортом можуть з'явитись і технічні неполадки. Користувач може вирішити, що він погано натиснув клавішу вимикача, і почне натискати її кілька разів поспіль, іншими слова «клацати» вимикач. Це неминуче призведе до зависання шини через великої кількості відсилаються команд;
- широкий вибір елементів управління. Функціонал і різноманітність дизайнерських ідей для елементів управління в провідних мережах завжди був на голову вище в порівнянні зі своїми «бездротовими братами».
- різноманітність інтегрованих систем. Природно, спектр вирішуваних завдань у провідних систем набагато ширший, ніж у бездротових. Реалізація складних сценаріїв, в яких зав'язані мультизадачність управління кліматом, управління великою кількістю груп світла, система мультирум, можлива тільки на провідній системі. Бездротова система тут не впоратися;

- довговічність. В провідній системі немає елементів на батареях, які можуть привести до несвоєчасного відключення її елементів і до не завжди зручному процесу заміни елементів живлення.

Слід врахувати і деякі особливості при установці дротових систем:

- місця розташування керуючих елементів повинні бути попередньо і чітко обрані. Після прокладки проводу і установки вимикача зрушити його буде практично неможливо. Помилка у вимірах буде дорого коштувати;
- той факт, що перебитий інформаційний провід (шина) виводить з ладу або частина системи, або всю цілком, не піддається сумніву. Тому велику увагу варто приділити якісному монтажу та кваліфікації монтажників;
- провідна система не варіант для готового ремонту. Проектувати і встановлювати її потрібно тільки спочатку будівництва, поки не зроблена класична електропроводка.

Бездротові технології та системи автоматизації «розумний дім».

Знову ж з назви розуміємо, що команди від керуючого пристрою до виконавчого в такій системі надходять не по дроту, а по радіосигналу. Відсутність проводів дає економію простору, економію часу на установку і економію матеріальних засобів завдяки відсутності цих самих проводів. Головний козир бездротових систем - можливість монтажу при готовому ремонті і класичної електропроводці.

Переваги бездротової системи «розумний дім»:

- є варіант установки в приміщеннях з готовим ремонтом з класичною електропроводкою. Бездротовий вимикач, який працює на автономному елементі живлення, можна встановити хоч на стелі.
- менше проводів. Особливо це стосується дерев'яних будинків зі зрубу. Хоча тут потрібно враховувати не дуже високу пропускну здатність дерева для радіосигналу.

- економія на проекті, що не може не радувати. У більшості випадків дизайн-проект для бездротової системи автоматизації без потреби.
- ціна. На сьогоднішній день існує достатня кількість бюджетних бездротових систем, функціоналу яких достатньо для вирішення базових завдань.

Слід врахувати і деякі особливості при установці бездротових систем:

- якість зв'язку. Радіоканал є радіоканал. Він дуже крихкий і залежить і від якості радіозв'язку, і від різного роду перешкод від тих же радіотелефонів або мікрохвильовок, і від матеріалу стін і навіть від включеної новорічної гірлянди, яка також може вплинути на силу сигналу.
- елементи живлення. Ними потрібно заpastися. Звичайно, їх потрібно міняти не кожен день, але всі ми добре знаємо, що за законом підлості саме ця батарейка перегорить саме в цей невідповідний момент. А запасний немає.
- в системі можуть бути присутніми передавачі, які живляться від мережі змінного струму, а не від батарейки. Відповідно, для таких передавачів потрібно закласти нульовий провід в монтажну коробку.
- багатофункціональну, а головне - стабільно працюючу систему управління на радіоканалі створити досить важко. І швидше за все неможливо.
- кібернебезпечність. Так, звучить голосно, але якщо у випадку з провідний системою ми можемо використовувати її автономно, відключивши всі зовнішні зв'язки, то радіоканал в цьому плані схильний до небезпеки.

Найвідомішими виробниками систем «розумний дім» є наступні п'ять прикладів:

- 1.) Ајах, зображений на рисунку 1.3:



Рисунок 1.3 – Набір компонентів Ajax

Виробник: Україна, відповідно, за замовчуванням підтримується український і російський інтерфейси.

Дана система автоматизації будинку в повній мірі справляється відразу з двома важливими завданнями:

- забезпечує комфорт і зручність в управлінні життєзабезпеченням приміщення;
- гарантує безпеку житла в повній мірі, контролюючи кордону об'єкта на предмет злому, а також електричну, пожежну, газову та інші можливі загрози для будинку.

Устаткування «Розумний будинок» Ajax працює на надійно зашифрованою і захищеною двостороннього радіозв'язку Jeweller власної розробки, має повну автономність від електромережі завдяки резервного джерела живлення – хабу, характеризується стильним дизайном всіх своїх пристроїв.

Переваги:

- простий монтаж;
- бездротової канал зв'язку між системними елементами; велика зона дії сигналу (до 2000 м);
- наявність захисту від зняття будь-якого з датчиків (бампера);

- можливий доступ інших користувачів (повний або частковий);
- автономна робота хаба від акумулятора (до 16 годин); Wi-Fi і GSM-зв'язок;
- різноманітність способів інформування користувача (дзвінок, SMS, Push-повідомлення);
- розумна розетка показує витрата електроенергії (з урахуванням підключених приладів), автоматично відключається при перепадах напруги; установка по QR-коду і управління за допомогою смартфона (iOS, Android); підключення до 100 пристроїв;
- наявність тривожної кнопки на пульті (брелоку);
- невисока вартість комплекту (від 200 \$).

Недоліки:

- функціонування тільки з роботою центрального контролера (Hub), тобто відсутність автономності датчиків;
- немає власної камери відеоспостереження (але є можливість підключення стороннього обладнання);
- управління тільки через телефон, хоча це знімає необхідність встановлювати будь-які додаткові програми на ПК.

Ця система є багатофункціональна, надійна, зручна, компактна. Має якісний захист від зломів, чудовий дизайн і зрозумілий інтерфейс. Установка і настройка такого комплексу спрощена до мінімуму і цілком доступна навіть для технічно не підготовлених користувачів. Важливою перевагою є і досить демократична ціна на девайс, беручи до уваги його широкий функціонал.

2.) BroadLink, зображений на рисунку 1.4:

Виробник: Китай. За замовчуванням немає українського інтерфейсу, при необхідності його можна знайти.



Рисунок 1.4 – Набір компонентів BroadLink

Устаткування «розумний будинок» BroadLink є комплект сучасних цифрових пристроїв, створених для раціонального управління побутовою технікою, а також освітлювальної, енергетичної, охоронної та іншими системами в будинку. Кожен елемент такого комплексу може працювати як самостійно, так і взаємодіяти один з одним.

Переваги:

- швидко встановлюється, підключається і настраюється; має широкий асортимент датчиків (вологості, температури, освітлення, шуму, забруднення повітря);
- можна легко додавати і прибирати різні пристрої;
- функціонує без центрального хаба (автономна робота датчиків);
- бездротове взаємодія пристроїв між собою;
- є своя камера відеоспостереження;

- контролюється по Wi-Fi через Інтернет з будь-якої точки планети;
- демократична вартість обладнання (від 200 \$).

Недоліки:

- невелика дальність дії сигналу (до 50 м);
- відсутність резервного живлення хаба;
- пульт працює тільки на прийом сигналів.

Має широкий функціонал, якісне програмне забезпечення, простота в установці та користуванні, а також доступна вартість. Такий комплекс не вимагає наявності центрального контролера, адже все його пристрою хоч і взаємопов'язані, але можуть працювати повністю автономно. Робота побутової техніки в будинку налаштовується згідно сценаріями в додатку на смартфоні.

3.) Fibaro, зображений на рисунку 1.5:



Рисунок 1.5 – Набір компонентів Fibaro

Виробник: Польща (розробка та реєстрація бренду - США). Скрутно знайти русифікований інтерфейс.

«Розумний дім» Fibaro відноситься до професійного обладнання із забезпечення автоматизації та безпеки будинку з найширшим функціоналом. Однак, на відміну від багатьох подібних систем, потребує встановлення та налаштування своєї апаратури досвідченими фахівцями.

Переваги:

- чудове наповнення системи всілякими датчиками і пристроями;
- наявність камери відеоспостереження;
- величезний вибір сценаріїв для користувача;
- розсилка повідомлень відразу на кілька телефонів;
- робота на базі протоколу Z-Wave, що дозволяє успішно взаємодіяти з іншим подібним обладнанням;
- датчик протікання оснащений сиреною;
- розумна розетка відображає рівень енергоспоживання підключених пристроїв, а також вимикається при скачках напруги;
- невелика дальність сигналу системи збільшується за рахунок можливості кожного її елемента бути ретранслятором сигналу;
- голосове управління через сервіс Google, але тільки англійською мовою.

Недоліки:

- висока вартість обладнання (від 600 \$);
- тільки професійний монтаж і налагодження;
- обов'язкове підключення центрального контролера Fibaro Home Center до інтернету через LAN-кабель;
- неможливість функціонування без центрального хаба;
- відсутність резервного живлення хаба;
- обмежена дальність сигналу (до 50 м без перешкод, хоча ця проблема вирішувана);
- затримка Push-повідомлень;
- необхідність в обов'язкову установку програмного забезпечення на ПК, а також урізане мобільний додаток.

У порівнянні з системами «розумний будинок» інших виробників обладнання Fibaro має найкращу наповненість всілякими датчиками для контролю стану приміщень і забезпечення автоматизації в управлінні

побутовою технікою. Однак встановити і розібратися з таким комплексом допоможе тільки професіонал.

4.) Orvibo, зображений на рисунку 1.6:



Рисунок 1.6 – Набір компонентів Orvibo

Виробник: Китай. Відсутність українського і російського інтерфейсу за замовчуванням (але є англійська), що дещо ускладнює установку і настройку.

Orvibo – це недорогий комплект простого в експлуатації обладнання, головне завдання якого полягає забезпеченні безпеки будинку. І тільки в другу чергу така установка може служити базою для організації повноцінної системи Розумний будинок.

Переваги:

- простота в установці і підключенні, віддалений контроль через додаток на смартфоні;
- автоматичне знаходження та підключення сенсорів до центрального хабу;
- широкий вибір пристроїв і можливість масштабування системи (близько 100 датчиків), причому інших виробників;
- наявність власної відеокамери;

- бездротової протокол взаємодії між контролером і датчиками (ZigBee);
- автономність деяких пристроїв від центрального хаба;
- вибір сценаріїв роботи з технікою будинку;
- Wi-Fi зв'язок з телефоном, дзвінки до до 10 номерів;
- цілком доступна вартість (від 150 \$).

Недоліки:

- невелика зона дії сигналу (до 30 м);
- досить скромний набір пристроїв в базовій комплектації (тільки частковий охоплення багатокімнатної квартири або офісу);
- відсутність резервного живлення хаба на випадок відключення електроенергії;
- дротове підключення до Інтернету (для надійності роботи системи).

Можна сказати, що Orvibo Security Kit - це проміжне обладнання між класичною системою охорони приміщення і «розумним домом». Воно має просте і зрозуміле управління, прекрасні можливості для масштабування пристроями сторонніх виробників за рахунок універсального протоколу ZigBee. Однак з метою зробити цю систему більш доступною в плані покупки, тут використовуються досить примітивні датчики без захисту від злому і відключення, а камера розрахована тільки на роботу всередині приміщення.

5.) Хіаомі, зображений на рисунку 1.7:

Виробник: Китай. Відсутність українського і російського інтерфейсу за замовчуванням, тільки англійську і китайську, що накладає свої складності в процесі установки і настройки.

«Розумний будинок» від Хіаомі відноситься до бюджетного класу обладнання, яке дозволяє зробити управління різними пристроями і побутовою технікою в будинку максимально простим і зручним.



Рисунок 1.7 – Набір компонентів Xiaomi

Переваги:

- повна автономність пристроїв;
- можливість масштабування;
- наявність власної камери відеоспостереження;
- бездротової протокол ZigBee;
- зручне управління за допомогою смартфона через Wi-Fi;
- наявність настроюються сценаріїв; компактність і стильний дизайн;
- низька вартість базового комплекту (всього 90 \$).

Недоліки:

- дуже маленька зона дії сигналу (до 10 м);
- скромний набір сенсорів і виконавчих пристроїв в базовому наборі; на різні датчики потрібно своє становище;
- відсутність резервного живлення хаба.

Таким чином, комплект Xiaomi є відмінним стартовим платформою для підключення інших сенсорів і пристроїв, в тому числі сторонніх виробників. Її елементи працюють як самостійно, так і в якості єдиного цілого. З їх допомогою можна створити досить функціональну систему контролю

житлового простору, включаючи забезпечення безпеки. Дана система, враховуючи її вартість, підійде для ознайомлення з системою розумного будинку [4].

1.3 Постановка завдання дослідження

Так як на кафедрі функціонує централізоване опалення, яке не має контролю, найпершою задачею кваліфікаційної роботи – це створення комп'ютерної системи, яка зможе контролювати роботу опалення класів кафедри.

Далі необхідно замінити морально застаріле та фізично зношене обладнання системи опалення.

Система опалення повинна мати управління температурою в конкретному приміщенні, управління опаленням в залежності від пори року і доби.

2 ТЕОРЕТИЧНИЙ РОЗДІЛ

2.1 Загальна характеристика комп'ютерних класів НТУ «ДП»

Кафедра «Інформаційних технологій та комп'ютерної інженерії» (ІТКІ) була відкрита в березні 1996 року і носила на той час назву: кафедра «Геоінформатики». На той час у складі кафедри була науково-дослідна лабораторія геоінформаційних технологій (1/78). Основне приміщення кафедри – аудиторія 1/79. Кафедра була відкрита на геолого-розвідувальному факультеті (ГРФ).

З першого року існування, кафедра почала готувати фахівців з напрямку інформаційних технологій – бакалаврів, спеціалістів та магістрів зі спеціальності «Інформаційні управляючі системи та технології» (спеціалізація «Геоінформаційні системи та технології»).

На кафедрі студенти здобувають знання в широкому спектрі сучасних дисциплін фундаментальної та загальноосвітньої спрямованості. Особлива увага в процесі навчання студентів звернена на професійне оволодіння різноманітними інформаційними технологіями й програмними продуктами. У комп'ютерних класах студенти працюють із різними версіями операційних систем MS Windows, а також засвоюють особливості програмних систем з відкритим кодом на базі ядер Linux. Проводиться вивчення широко відомих об'єктно-орієнтованих та спеціалізованих мов програмування: C/C++, Java, C#, Python, Ruby, SQL, VBA, HTML, JavaScript. Навчання студентів умінню взаємодіяти зі складними інформаційними системами проводиться в інтегрованих середовищах розробки програм, пакетах математичного моделювання, а також в ГІС та СУБД – Visual C/C++, IntelliJ IDEA, PyCharm, Netbeans, Code::Blocks, MathCAD, Matlab, ArcGIS, Oracle PL/SQL Server, MS SQL Server та ін.

У ході значної реорганізації та оптимізації кафедр НТУ «ДП», у середині серпня 2019 року наказом ректора університету академіка НАН України Г.Г. Півняка, на базі кафедри «Геоінформаційних систем» та частки

колективу кафедри «Автоматизації і комп'ютерних систем», при факультеті інформаційних технологій (ФІТ) створена кафедра «Інформаційних систем та технологій» з двома провідними спеціальностями:

- 126 «Інформаційні системи та технології»;
- 123 «Комп'ютерна інженерія».

На підставі рішень Ректорату НТУ «Дніпровська політехніка» та Вченої ради університету, з 1 вересня 2020 року назву кафедри змінено на «Кафедра інформаційних технологій та комп'ютерної інженерії» [5].

Об'єктом дослідження є комп'ютерні класи 1-го корпусу учбового закладу НТУ «Дніпровська політехніка» кафедри «Інформаційних технологій та комп'ютерної інженерії» 1/130 та 1/131.

Комп'ютерний клас — це спеціальне приміщення, обладнане об'єднаними в єдину мережу комп'ютерами і призначене для навчання роботі з персональним комп'ютером. Викладач за допомогою свого комп'ютера може повністю контролювати комп'ютери учнів, в тому числі і перевіряти відповіді учнів не відходячи від свого комп'ютера.

Комп'ютерний клас складається з ПК викладача, ПК учнів, комп'ютерної мережі (Ethernet або USB), спеціального програмного забезпечення. Також в комп'ютерний клас може бути укомплектований акустичною системою, принтером, сканером і т.п.

В випадку з класами кафедри інформаційних технологій та комп'ютерної інженерії, складовими є не тільки комп'ютери, а також і проектором.

Приміщення можуть використовуватись весь робочий день, з 8-ї ранку і до завершення останньої пари, також на вихідних можуть проводитись консультації для студентів весь навчальний рік.

Класи знаходяться на третьому поверсі відразу за переходом між першим та четвертим корпусами, площа 1/130 – 78,4 м², 1/131 – 69,6 м², що необхідна для подальшого вибору обладнання класів.

2.2 Структура об'єкту дослідження

Перший корпус НТУ «ДП», в якому знаходиться кафедра «ІТКІ», отоплюється за допомогою централізованого опалення. Воно організовано таким чином, що тепло виробляється за межами кабінетів, класів та аудиторій, а саме в підвальному приміщенні, в котельні, а потім надходить по довгим розгалуженим трубопроводах до цільового приміщення, опалення функціонує на воді. Основним приладом опалення системи є радіаторні труби. На рисунку 2.1 зображено розташування опалювальних пристроїв в першому корпусі.

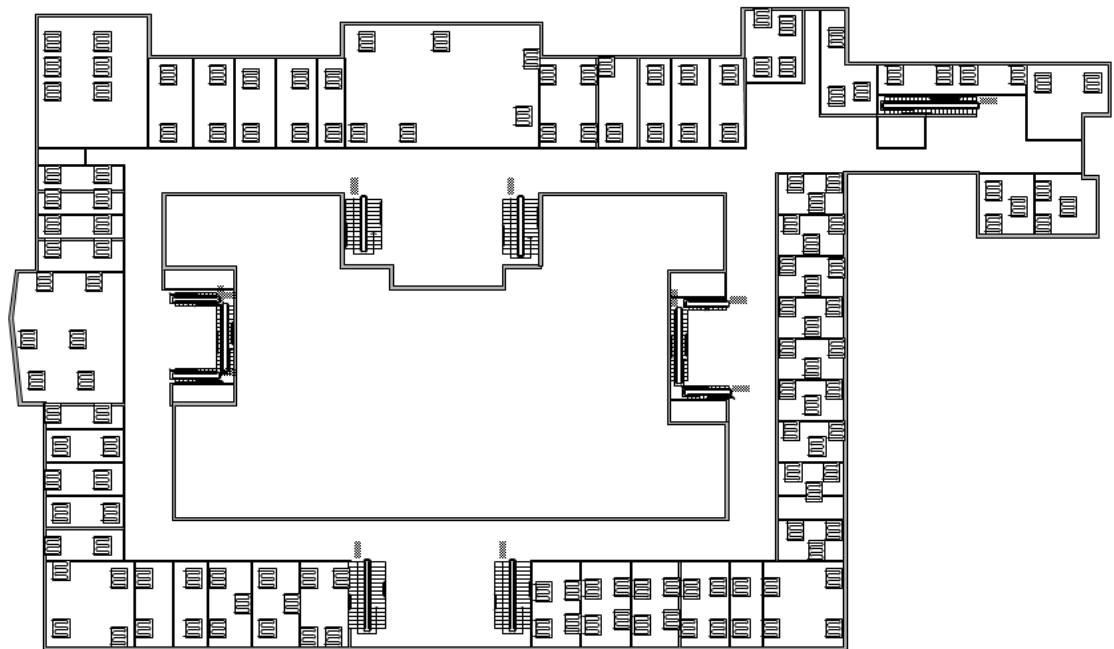



Рисунок 2.1 – Схема розташування опалювальних пристроїв в першому корпусі

Де:

 – сходи;

 – вінко;

 – радіаторна труба.

Комп'ютерні класи 1/130 та 1/131 знаходять на 3-му поверсі корпусу, біля переходу між першим та четвертим корпусами. На рисунках 2.2 та 2.3

зображено розташування опалювальних пристроїв у класах 1/130 1/131 відповідно.

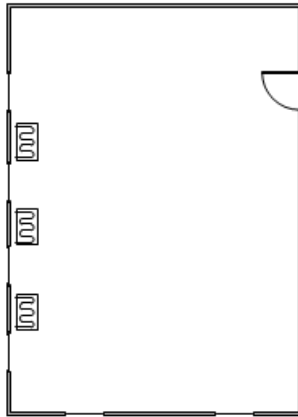




Рисунок 2.2 – Схема розташування опалювальних пристроїв у класі 1/130

Де:

 – двері;

 – вінко;

 – радіаторна труба.

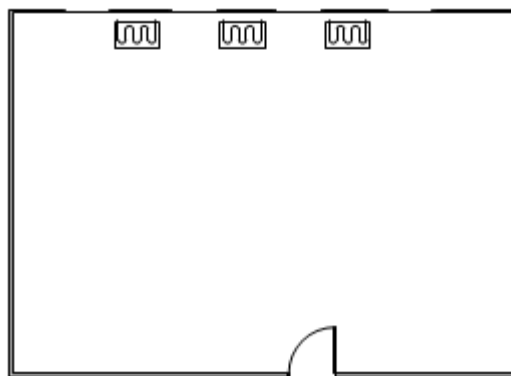




Рисунок 2.3 – Схема розташування опалювальних пристроїв у класі 1/131

Де:

 – двері;

 – вінко;

 – радіаторна труба.

2.3 Принцип функціонування об'єкту

Нагрітий в котлі теплоносієм, у нашому випадку – вода, рухається по системі, поступово віддаючи тепло трубам і опалювальним приладам, і далі – нагрівається приміщення.

Оскільки труби, котел і радіатори утворюють замкнену систему, то вода постійно рухається по колу. На рисунку 2.4 зображена схема роботи централізованого опалення.

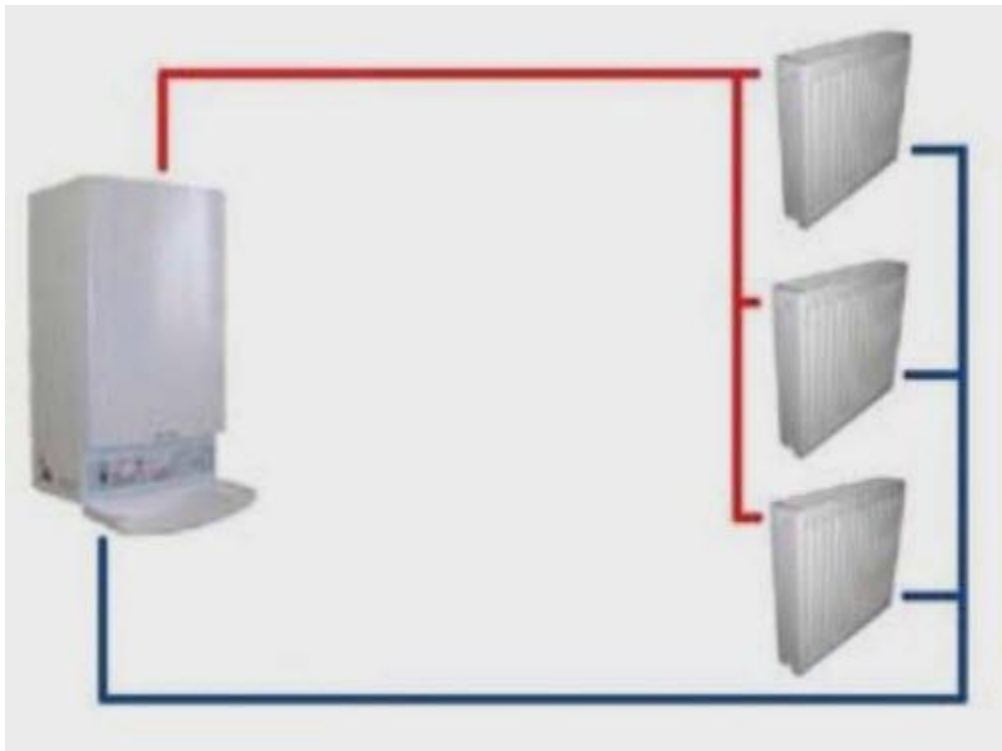


Рисунок 2.4 – Схема роботи централізованого опалення

2.4 Обґрунтування і вибір методів дослідження

Для вибору системи опалення кафедри «ІТКІ», найкраще підходять методи аналізу і синтезу, математичну модель, а також, метод порівняння, тому що відбувається аналіз між двома системами, що мають різну структуру роботи, функціональний набір та епоху використання.

2.4.1 Метод аналізу і синтезу

У випадку централізованого водяного опалення з часом виникає корозія труб, через неякісну діагностику виникають засмічення в трубах,

радіаторах. Не має можливості обрати часи опалення, через суворий графік сезону. Не має змоги змінити температуру приладів опалення. Виникає велика тепловтрата від того, що теплоносії проходить велику відстань у трубопроводі від місця нагріву. Централізоване опалення функціонує вже більше тридцяти років у першому корпусі НТУ «ДП», а повної заміни та ремонту приладів системи ще не було.

В підсумку, така система на даний момент застаріла, має велику вірогідність пробою, деякі радіатори в корпусі можуть не нагріватись достатньо або взагалі, через засмічення трубопроводу. Для її ремонту під час опалювального сезону необхідно повне припинення її функціонування в даній ділянці.

В роботі «розумного опалення» на основі електричного нагріву приборів першочерговою проблемою стає оплата електроенергії, а так як в плані енергозбереження в НТУ «ДП» є встановлення сонячних батарей, що матимуть змогу накопичувати електроенергію, то це дає змогу використовувати її для роботи електричних приладів опалення. Також в такій системі є можливість зміни температури обігріву кімнат, класів, аудиторій, холів. Діагностика може бути автономною. Знівельована тепловтрата. Можлива зміна вологості повітря. Можуть з'явитися додаткові електричні поля. Через вихід прибору з ладу є можливість великої температури поверхонь електричних приладів.

В підсумку, для встановлення цієї системи опалення не має великої кількості мінусів, вони зменшуються через фактори модернізації, а саме, оплата стане меншою після встановлення сонячних батарей, вихід приборів з ладу – зменшуються від запланованої діагностики системи, а також від можливості запрограмувати прилад для оповіщення збоїв.

2.4.2 Метод порівняння

Централізоване опалення плюси:

- виведення вибухонебезпечного технологічного обладнання з житлових будинків;
- точкова концентрація шкідливих викидів на джерелах, де з ними можна ефективно боротися;
- можливість роботи на різних видах палива, включаючи місцеве, смітті, а також відновлюваних енергоресурсах;
- можливість заміщати просте спалювання палива (при температурі 1500-2000°C для підігріву повітря до 20°C) тепловими відходами виробничих циклів, в першу чергу теплового циклу виробництва електроенергії на ТЕЦ;
- щодо набагато більш високий електричний ККД великих ТЕЦ і теплової ККД великих котелень працюють на твердому паливі.

Централізоване опалення мінуси:

- зайві тепловтрати в теплових мережах;
- вартість подачі тепла в разі центрального опалення досить висока (комунальні послуги);
- відкриття та закриття опалювального сезону в строго визначені терміни, що не залежать від бажань жителів. А це в свою чергу дискомфорт в «перехідні» періоди, коли холоду уже почалися, а опалювальний сезон - немає. Зворотне також вірно - оплата за опалення, коли воно вже не потрібно.

«Розумне опалення» на електричних приладах плюси:

- після нагрівання приміщення, для підтримки постійної температури включаються рідше, ніж конвектора. У підсумку, це впливає на економію електроенергії;
- зміна температури за бажанням;
- комфортна температура приміщення зберігається від 2-х і більше годин, після виключення системи;

- не сушить повітря, не спалює кисень, при роботі не виділяє шкідливих речовин;
- працює безшумно;
- довговічність (середня гарантія від виробника 5 років).

«Розумне опалення» на електричних приладах мінуси:

- поступається традиційному (водяному) і повітряному опаленням, так як теплоносієм даного типу є електрична енергія, яка має високу вартість, до того ж бувають збої в подачі її споживачеві.

В цілому централізована система опалення проявляла себе у гарному світлі, але мала функціональність і графік роботи цієї системи є незадовільними. Тим паче, з часом, строк служби приладів вже давно пройшов і вони не працюють, як раніше. Тому висновком порівняння є те, що «розумне опалення» кафедри мало б кращі показники.

2.4.3 Метод математичного моделювання

Математичне моделювання – метод дослідження процесів або явищ шляхом створення їхніх математичних моделей і дослідження цих моделей.

Модель буде створена у додатку Simulink – це середа динамічного міждисциплінарного моделювання складних технічних систем і основний інструмент для модельно-орієнтованого проектування.

Спершу потрібно задати систему опалення. Моделювання починається з визначення завдань, які виходять за межі програмного середовища Simulink. Вимоги моделі Define і виводять математичні рівняння. На рисунку 2.5 зображена схема алгоритму побудови моделі Define.

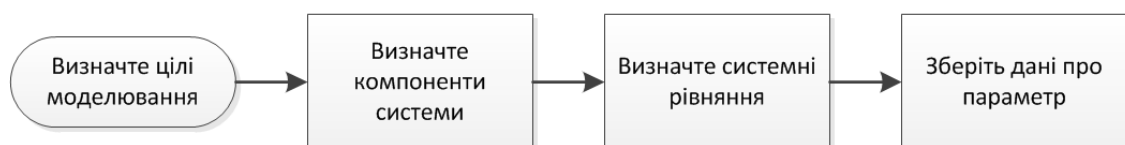


Рисунок 2.5 – Схема алгоритму моделі Define

Цілі: математична модель потрібна для системи контролю щоб спостерігати, як зміна зовнішньої температури впливає на внутрішню температуру та для дослідження ефекту параметрів, що змінюються, на внутрішній температурі.

Компоненти: для системи опалення з «розумними» пристроями – це термостат, або датчик температури, зовнішнє середовище та клімат у класі.

Системні рівняння: три залежні від часу змінні задають теплообмін в класі:

- температура кімнати T_{room} ;
- приплив теплоти Q_{gain} – теплова енергія передається від нагрівача в кімнату;
- втрата тепла – теплова енергія передається з кімнати до зовнішнього середовища.

Диференціальне рівняння визначає взаємозв'язок між цими змінними, але оскільки передача тепла визначається з точки зору зміни температури, лише кімнатна температура є змінною стану.

В формулі 2.1 описується: температура повітря в нагрівачі постійна при T_{heater} , а температура в приміщенні – T_{room} . Прибуток теплової енергії в приміщення відбувається за рахунок конвекції нагрітого повітря від нагрівача, теплоємність c_{air} . Приріст тепла для маси повітря в нагрівачі, $m_{heaterair}$, пропорційна різниці температур між нагрівачем та приміщенням:

$$Q_{gain} = m_{heaterair} \times c_{air} \times (T_{heater} - T_{room}) \quad (2.1)$$

Швидкість приросту теплової енергії (формула 2.2) від нагрівача становить:

$$\frac{dQ_{gain}}{dt} = \frac{dm_{heaterair}}{dt} \times c_{air} \times (T_{heater} - T_{room}) \quad (2.2)$$

Останнє спрощення у формулі 2.3: постійна кількість повітря, $M_{heaterair}$, протікає через нагрівач за одиницю часу і замінюється $\frac{dm_{heaterair}}{dt}$, з цією константою спрощує рівняння до:

$$\frac{dQ_{gain}}{dt} = M_{heaterair} \times c_{air} \times (T_{heater} - T_{room}) \quad (2.3)$$

В формулі 2.4 записані втрати теплової енергії з приміщення проводяться через стіни та вікна, і пропорційні різниці температур між кімнатною та зовнішньою температурою:

$$Q_{loss} = \frac{k \times A \times (T_{room} - T_{outside}) \times t}{D} \quad (2.4)$$

Швидкість втрат теплової енергії, формула 2.5, становить:

$$\frac{Q_{loss}}{dt} = \frac{k \times A \times (T_{room} - T_{outside})}{D} \quad (2.5)$$

У формулі 2.6 записана заміна $\frac{k \times A}{D}$ на $\frac{1}{R}$, де R тепловий опір, і спрощує рівняння до:

$$\frac{Q_{loss}}{dt} = \frac{(T_{room} - T_{outside})}{R} \quad (2.6)$$

Швидкість зміни температури в приміщенні визначається у формулі 2.7, віднявши швидкість втрат тепла від швидкості приросту тепла:

$$\frac{dT_{room}}{dt} = \frac{1}{m_{heaterair} \times c_{air}} \times \left(\frac{dQ_{gain}}{dt} - \frac{dQ_{loss}}{dt} \right) \quad (2.7)$$

Змоделюємо підсистеми, блок Subsystem, для компонентів системи опалення, на рисунку 2.6 зображений вигляд структури поверхневого рівня.

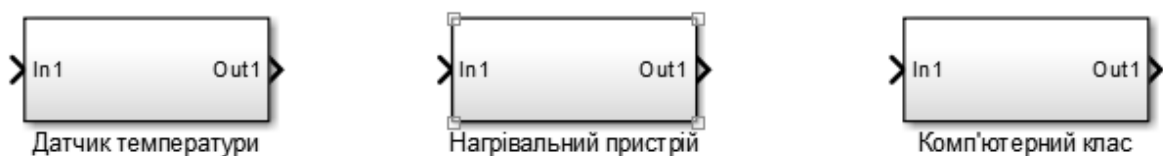


Рисунок 2.6 – Фото блоків Subsystem в Simulink

Більшість значень параметрів, необхідних для моделі опалення будинку, опубліковані в стандартних таблицях властивостей. Ці параметри вже занесені до робочого простору в Simulink, вони зображені на рисунку 2.7.

T_heater	50
M_heater_air	3600
c_air	1005.4
m_room_air	1470
T_roomIC	20
R_equivalent	4.329e-07

Рисунок 2.7 – Фото змінних для моделювання

Де,

T_{heater} – постійна температура повітря від нагрівача ($^{\circ}C$);

M_{heater_air} – постійна норма повітряної маси (кг/год);

C_{air} – питома теплоємність (Дж/кг* $^{\circ}C$);

M_{room_air} – маса повітря в приміщенні або обігрівачі (кг/год);

T_{roomIC} – початкова температура повітря в приміщенні ($^{\circ}C$);

$R_{equivalent}$ – термічний опір (год* $^{\circ}C$ /Дж).

В кожній підсистемі знаходять блоки вводу та виводу в одиничному екземплярі, хоча їх можна додавати більше, але вже самостійно. Кожен блок Inport створює вхідний порт в блоці Subsystem, і кожен блок Outport створює вихідний порт.

Розробимо модель підсистеми нагрівального пристрою. Вона складається з:

- в якості потоків приймає поточну температуру в приміщенні та керуючий сигнал від термостата;
- обчислює приплив теплоти від нагрівача;
- видає коефіцієнт нагрівання, коли ввімкнено контрольний сигнал.

Щоб змодельовати підсистему нагрівача, потрібно змодельовати рівняння швидкості припливу тепла, формула 2.3, за допомогою блоків Simulink. Ця схема зображена на рисунку 2.8.

Перелік дій необхідних для моделювання:

- 1) Відкрити підсистему «Нагрівальний пристрій».

- 2) Клацнути на модель і введіть Sum, щоб відобразити список блоків із Sum в назві. Клацнути блок Sum у списку. Коли з'явиться запит щодо списку знаків, ввести | - +, щоб розмістити - і + вхідні порти на блоці, і натисніть Enter. Вертикальна смуга (|) змінює положення вхідних портів, вставляючи проміжки між портами. Вертикальна смуга на початку списку знаків розміщує пробіл у верхній частині блоку та зміщує порти проти годинникової стрілки.
- 3) Додати блок Constant для моделювання постійної температури повітря від нагрівача. Встановити параметр блоку Constant value значення T_heater.
- 4) Додати другий блок Inport, щоб прийняти сигнал кімнатної температури від іншої частини моделі.
- 5) Додати блок підсилення до підсистеми Нагрівача. Встановити для параметра Gain значення $M_heater_air * c_air$.
- 6) Підключити вихід блоку Sum до входу блоку Gain.
- 7) Додати мітки до сигнальних ліній, щоб допомогти відстежувати компоненти моделі до рівнянь та вимог до моделі.
- 8) Видалити зв'язок між блоками In1 та Out1.
- 9) Додати блок Product. Підключити блок In1 до першого входу блоку, а вихід блоку до блоку Out1.
- 10) Підключити вихід з блоку посилення до другого входу.
- 11) Перейменувати блоки та з'єднати їх.

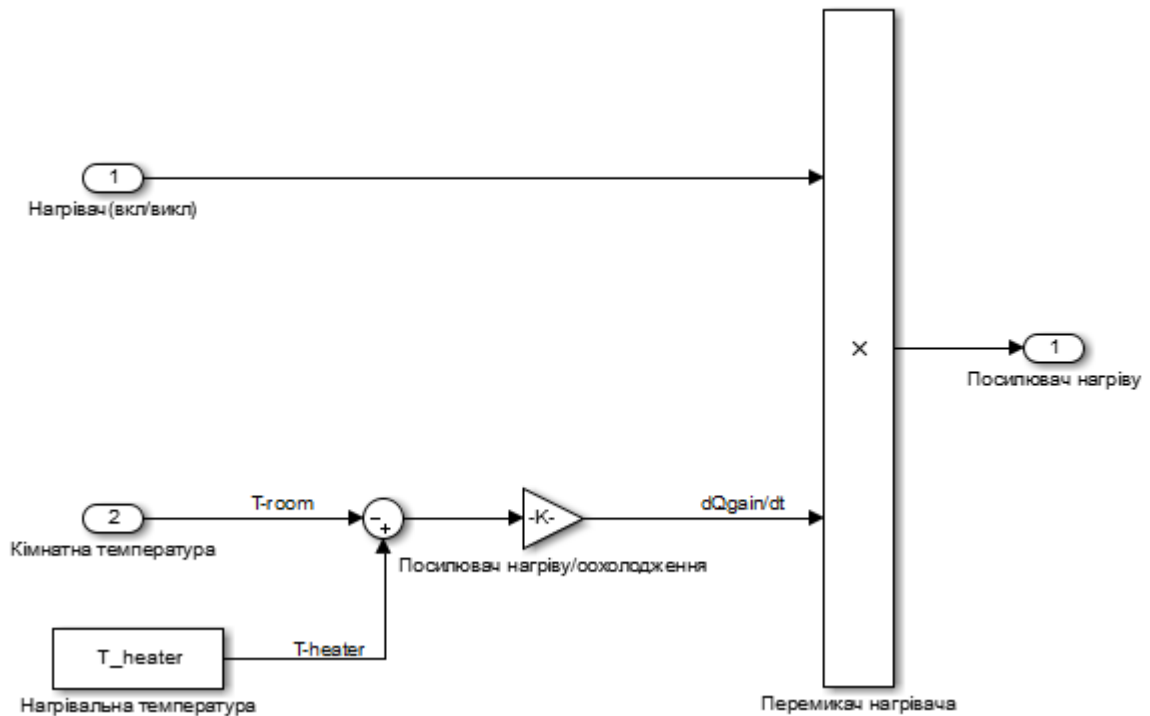


Рисунок 2.8 – Схема підсистеми «Нагрівальний пристрій»

Моделювання схеми датчика тепла не потребує рівняння. Лише вимоги:

- коли температура в приміщенні нижче встановленої температури, обігрівач увімкнено, а контрольний сигнал дорівнює 1. Коли температура в приміщенні перевищує встановлену температуру, контрольний сигнал дорівнює 0;
- щоб уникнути повторного перемикання навколо заданої температури, термостат дозволяє здійснювати гістерезис 0,5 градуса Цельсія навколо заданої температури. Якщо термостат увімкнений, перед вимкненням температура в приміщенні повинна підвищитися на 0,5 градуси вище заданої температури. Якщо термостат вимкнений, перед увімкненням температура в приміщенні повинна опуститися на 0,5 градуса нижче встановленої температури.

Цей компонент моделює роботу термостата, визначаючи, коли система опалення увімкнена або вимкнена. Він містить лише один блок реле, але

логічно представляє термостат у моделі. На рисунку 2.9 зображена ця модель. Пункти, що необхідно виконати для моделювання:

- 1) Відкрити підсистему «Датчик температури». Додати блок Sum.
- 2) Підключити блок Inport до входу «+» блоку Sum. Блок Inport встановлює кімнатну температуру.
- 3) Додати другий блок Inport і підключіть його до входу «-» блоку Sum. Цей другий блок Inport – це поточна температура приміщення від підсистеми «Комп'ютерний клас».
- 4) Додати блок «Relay». Встановіть параметр точки включення на 0.5, а параметр точки вимкнення – на -0.5.
- 5) З'єднати та перейменувати блоки.

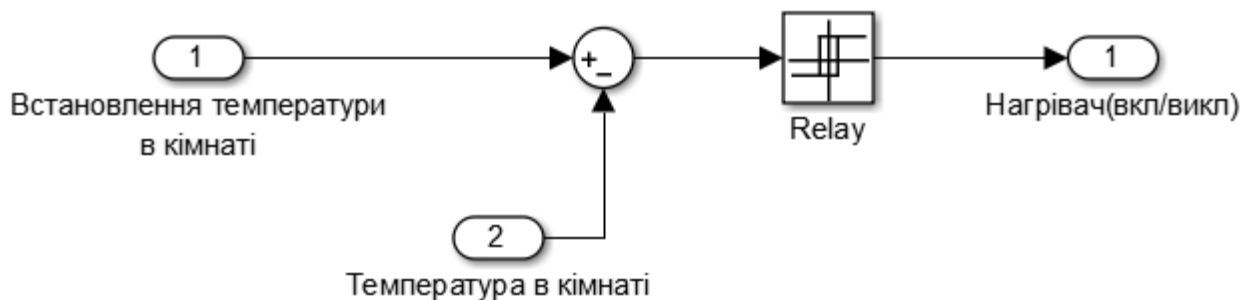


Рисунок 2.9 – Схема підсистеми «Датчик температури»

Далі змодельуємо компонент приміщення, схема зображена на рисунку 2.10. Вхідні компоненти приміщення – це тепловий потік від нагрівального компонента та зовнішня температура повітря. Компонент кімнати використовує ці входи для обчислення втрат тепла через стіни, втрат тепла через вікна та поточної кімнатної температури. Для проектування підсистеми приміщення використовуємо рівняння швидкості втрати тепла, формула 2.6, та рівняння зміни кімнатної температури, формула 2.7. Дії, які необхідно зробити для створення схеми:

- 1) У підсистемі «Комп'ютерний клас» додати блок Sum.
- 2) Підключити Troom до блоку Sum.

- 3) Додати ще один блок Inport і підключіть його до - входу блоку Sum.
- 4) Додати ще один блок посилення. Встановіть для параметра Gain значення $1/R_{equivalent}$.
- 5) З'єднати блоки.

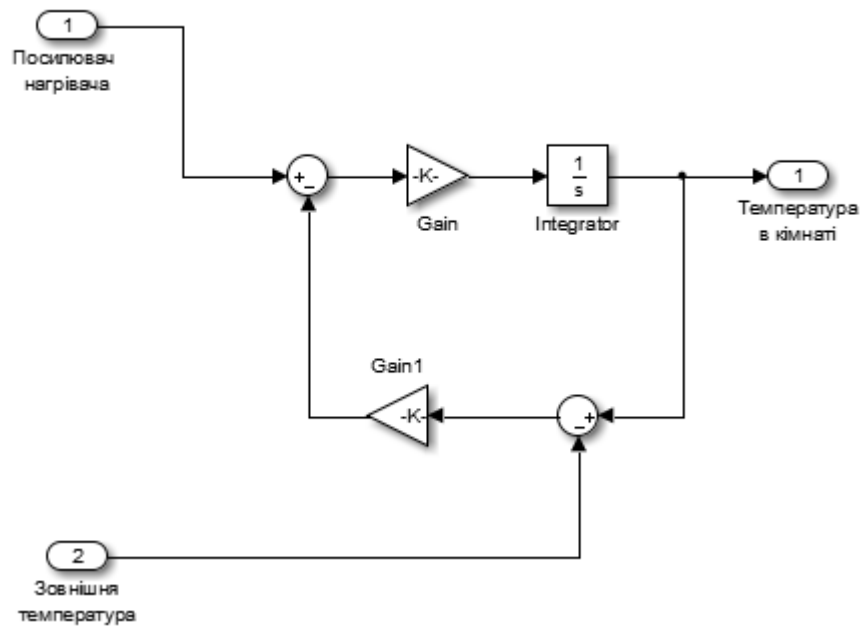


Рисунок 2.10 – Схема підсистеми «Комп'ютерний клас»

Далі виконується підготовка системи для симуляції. Повна схема зображена на рисунку 2.11. Для цього необхідно виконати наступні кроки:

- 1) Відкрити модель із готовими підсистемами.
- 2) Підключити підсистеми між собою.
- 3) Додати блок Constant для встановлення кімнатної температури. Встановити параметр Constant значення.
- 4) Додати блок Sine Wave, щоб відобразити зміну зовнішньої температури. Необхідно встановити параметри для синусоїди.
- 5) Додати блок Scope Viewer, щоб переглянути результати моделювання.
- 6) У програмі перегляду сигналів натиснути кнопку Signal Selector. У діалоговому вікні Signal Selector, на лівій панелі, вибрати

верхню ієрархію моделі. На правій панелі вибрати сигнали «Комп'ютерний клас» та Sine Wave.

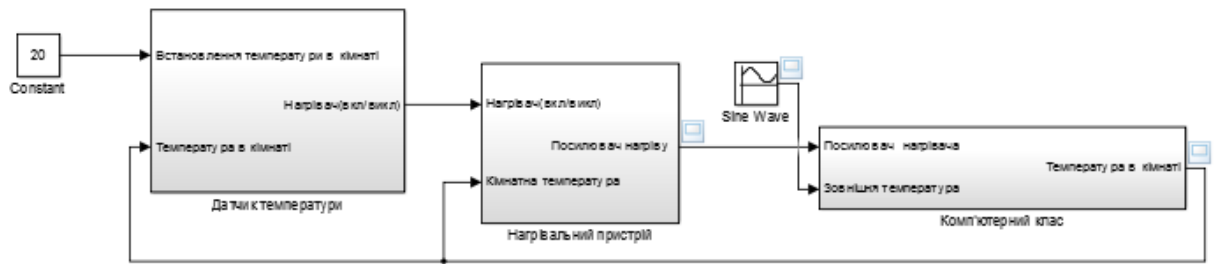


Рисунок 2.11 – Схема повної математичної моделі

2.5 Висновки по розділу

У теоретичному розділі було описано характеристики кафедри «ІТКІ», структуру самого корпусу, в якому знаходиться кафедра, за допомогою методу аналізу та синтезу, методу порівняння, обґрунтовано вибір системи опалення. Також була розроблена математична модель системи опалення, яка вже готова до тестів та експерименту.

3 СИНТЕЗ СИСТЕМИ КОНТРОЛЮ

3.1 Формулювання технічних вимог до системи контролю «розумного опалення»

3.1.1 Вимоги до реалізації системи

Для роботи системи в цілому, необхідно мати хаб – керуючий пристрій, що з'єднає всі елементи системи один з одним і зв'яже її з зовнішнім світом. Далі необхідно мати датчики температури, для того, щоб хаб розумів, яка зараз температура, чи потрібно її змінити, а також датчик руху, якщо будуть проводитись позачергові заняття в комп'ютерних класах. «Розумні розетки» також невід'ємна частина «розумного дому» для автоматизації роботи пристроїв системи. Нагрівальні пристрої.

Також необхідний персонал, який має високу кваліфікацію та не однократно розробляв подібні системи контролю.

3.1.2 Вимоги до видів забезпечення

Датчик повинен бути максимально чутливим і реагувати на зміни нагріву з мінімальною затримкою. Діапазон для датчика температури: 0 – +35°C.

Датчик руху має кут обзору вищий за 90°. Відстань спрацювання не перевищує 5м.

«Розумні» пристрої повинні підтримувати протоколи WI-FI, ZigBee або Z-Wave. Контролюючий пристрій має змогу працювати в різних екосистемах, наприклад Philips, Apple, Xiaomi, Samsung.

Пристрої обігріву повинні мати змогу не сушити повітря, легкої заміни, малу енергозатратність. Діапазон температур, для обігріву класів: +10 – +25°C.

3.2 Розробка схеми функціональної структури

Система контролю «розумного опалення» повинна виконувати наступні функції:

- вимір і зчитування температури;
- зчитування руху;
- включення вимикання приладів опалення;
- установка температури.

Всі функції обробляються на центральному пристрої керування.

Виходячи з переліку функцій схема функціональної структури системи матиме вигляд, як показано на рисунку 3.1.

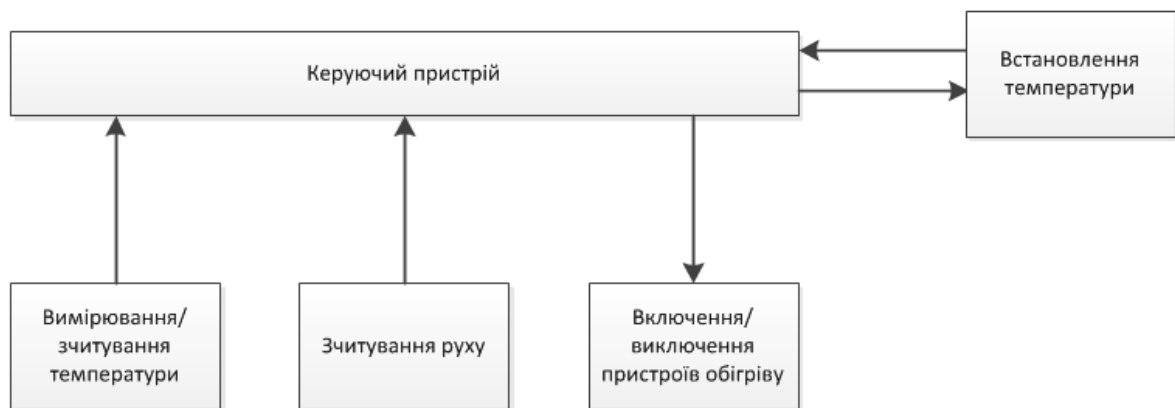


Рисунок 3.1 – схема функціональної структури системи

На основі побудованої функціональної структури можна створити функціональну схему.

3.3 Розробка функціональної схеми

Контроль та управління системою «розумного опалення» виконується на основі протоколу ZigBee. За вимогами до забезпечення системи, було висунута підтримка керуючим пристроєм протоколу, що має змогу підключити систему до локальної мережі кафедри. Це означає, що система буде функціонувати бездротовим способом, лише виконуючий пристрій підключається до джерела живлення. Функціональна схема системи «розумного опалення» зображена на рисунку 3.2.

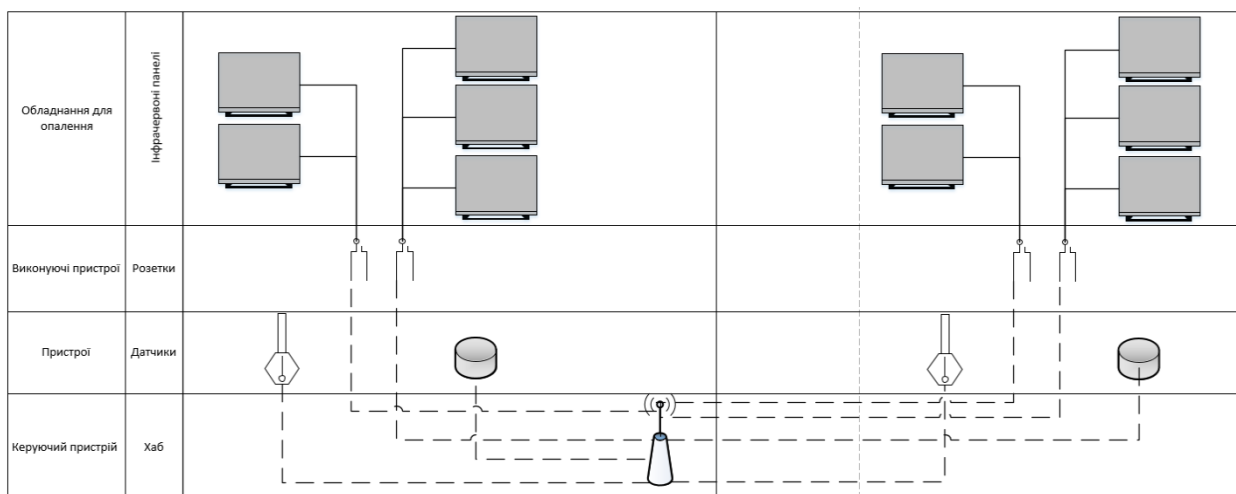


Рисунок 3.2 – Функціональна схема системи

3.4 Вибір та обґрунтування застосування апаратних засобів

Для вибору пристроїв системи необхідно визначити кількість виконуючих пристроїв та датчиків.

Після проведення аналізу вимог, щодо системи контролю «розумне опалення», були обрані наступні прилади.

В якості хабу системи контролю обрано – Xiaomi Mi Smart Home Multifunction Gateway 2 DGNWG02LM [6]. На рисунку 3.2 зображено фото хабу, а в таблиці 3.1 записані його характеристики. Мультифункціональний пристрій можна з'єднати з іншими смарт-гаджетами, при цьому навіть при нестабільній роботі WiFi весь «розумний будинок» буде працювати без розривів. Унікальна характеристика ZigBee дозволяє пристрою реагувати ще швидше і керувати великою кількістю потоків. Оснащений вбудованим датчиком освітленості, кольоровими світлодіодами, динаміком і on-line інтернет радіо. Цей пристрій забезпечує взаємодію між датчиками і обробку сигналів, що надходять з тим, щоб потім завантажити дані в Мережу. Сам хаб досить компактний. Його зручно влаштувати в будь-якій частині квартири. При цьому установка навіть не вимагає додаткових інструментів.



Рисунок 3.3 – Фото Xiaomi Mi Smart Home Multifunction Gateway 2

Таблиця 3.1 – Характеристики Xiaomi Mi Smart Home Multifunction Gateway 2

Призначення	Управління розумним будинком
Тип пристрою	Комутатор (блок керування)
Автономна робота	Так
Інтерфейс	Wi-Fi, Zigbee
Кількість пристроїв, що підключаються	37
Особливості	Робоча температура 0 °С -40 °С; Вологість 5% -95% відносної вологості, без конденсації; Вхідна напруга 100-240В, 50Гц / 60Гц; Вбудоване радіо, датчик освітленості
Джерело живлення	Від мережі

Датчик температури Mi Home MiJia [7] – ізотермічний датчик температури також підтримує протокол ZigBee, що дає йому змогу вільно інтегруватися з іншими датчиками системи Smart Home через його шлюз системи. Це дасть власнику змогу нормалізувати температурний режим для підтримання комфортного та здорового середовища у своїй квартирі.

Датчик й температури від Xiaomi виготовлений із використанням передової технології Sensihon, яка здатна визначати температуру та вологість із неймовірною точністю завдяки надчуттєвим датчикам. Похибка в обчисленні температурних даних становить усього 0.3°C, а в разі обчислення вологості — тільки 3%. На рисунку 3.3 зображено фото датчика, а в таблиці 3.2 записані його характеристики.



Рисунок 3.4 – Фото датчика температури Mi Home MiJia

Таблиця 3.2 – Характеристики датчика температури та вологості Mi Home MiJia

Призначення	Моніторинг навколишнього середовища
Тип пристрою	Датчик навколишнього середовища
Інтерфейс	Zigbee
Сумісність	Пристрої розумного будинку Xiaomi
Особливості	Матеріал: високоякісний пластик; Вимірювання температури навколишнього середовища з точністю 0.3 градусів Цельсія; Вимірювання вологості навколишнього середовища з точністю до 3%
Джерело живлення	Батарейки CR2032 x 1

Xiaomi Mi Smart Socket 2 ZigBee GMR4004CN [8] – розетка має компакту форму і виконана з вогнестійкого матеріалу, який витримує температуру до 750° С. У виготовленні розетки використовувалася

технологія хвильової пайки. Вбудований датчик температури здійснює моніторинг роботи надпотужних приладів і контролює внутрішню температуру розетки. Якщо відбувається перегрів, розетка зупиняє подачу струму з метою уникнення аварійної ситуації або поломки. Для безпеки розетка оснащена захисними шторками. На рисунку 3.4 зображено фото розетки, а в таблиці 3.3 записані її характеристики.



Рисунок 3.5 – Фото розетки Xiaomi Mi Smart Socket 2 ZigBee

Таблиця 3.3 – Характеристики розетки Xiaomi Mi Smart Socket 2 ZigBee

Управління	ZigBee
Монтаж	Навісний
Стандарт виделки	Європейський Американський Китайський
Максимальна потужність навантаження	2200 Вт
Максимальний струм	10 А
Функціональність розетки	Відключення приладів при перегріванні, Вкл/відкл за розкладом, Датчик температури, Таймер
Особливості	Міжнародний штепсель, Шторки
Матеріал	Пластик
Робоча температура	0°C - 40°C

Продовження таблиці 3.3

Додатково	Захист від перевантаження по струму Заземлення Індикація (підсвічування) Шторки Таймер Термостат (терморегулятор) Параметри входу 100-250V ~ 50 Hz
------------------	--

В якості датчику руху був обраний датчик руху Mijia Move Detector (UTC4016CN) [9]. Інфрачервоний датчик руху Mijia Move Detector – один з пристроїв набору Smart Home. Датчик невибагливий і досить практичний. Для встановлення Mijia Move Detector не потрібно порушувати дизайну інтер'єру, просвердлюючи в стінах дірки і псуючи ремонт. Досить просто поставити датчик в зручному місці і він готовий до роботи. Mijia Move Detector оснащений індукційними лінзами (індикаторами стану). Дальність роботи датчика складає до семи метрів при встановленні на висоті близько двох метрів. Горизонтальний кут огляду – 170 градусів. На рисунку 3.5 зображено фото датчика, а в таблиці 3.4 записані його характеристики.



Рисунок 3.6 – Фото датчика руху Mijia Move Detector

Таблиця 3.4 – Характеристики датчика руху Mijia Move Detector

Призначення	Безпека
Тип пристрою	Датчик руху
Інтерфейс	Zigbee
Сумісність	Пристрої розумного будинку Xiaomi

Продовження таблиці 3.4

Особливості	Матеріал: високоякісний пластик; Дальність виявлення до 7 метрів; Кут виявлення 170°; Індукційні лінзи (індикатори стану); Скидання (прив'язка пристрої / видалення)
Джерело живлення	Акумулятор

Для обігріву класів обрані ІЧ панелі компанії «Елтрон ІП», а саме Елтрон ІП-600МП.

Особливості:

- у якості нагрівача використовується стрічка з аморфного металу оригінального сплаву, спеціально розробленого для даної серії інфрачервоних панелей, товщиною 20 мкм;
- у якості ізоляції застосовані поліефірні смоли, які запобігають виникненню запаху та передчасному старінню матеріалів;
- оригінальна рецептура декоративного покриття панелей дозволяє отримати максимальну зносостійкість і тепловіддачу при експлуатації.

Це система прямого електричного опалення, що регулюється у кожному окремому приміщенні.

Це означає що:

- початок опалювального сезону – коли вирішить сам Споживач;
- температура повітря в приміщенні – яка комфортна Споживачу;
- узгодження, проект, дозвіл на підключення – нікуди йти не треба;
- додаткова потужність для підключення – достатньо існуючих;
- можливості монтажу та демонтажу – необмежені.

Змонтувати систему можна за декілька годин згідно з інструкцією. При необхідності переїзду – все легко демонтується.

Нагрівальна панель «Елтрон ІП» виробляє тепло з низьким енергоспоживанням.

Панель абсолютно безпечна, не загрожує життю і здоров'ю людей.

Вона – не взривається, не горить, не створює електричне поле, не висушує повітря, не підіймає пил, вологостійка. Вирішує проблему холодних сирих стін та кутів приміщення. Не потребує сервісного обслуговування.

Втілює у собі унікальні якості, які дарують не тільки комфортне тепло, а й надають оздоровчий вплив на організм людини, так як довгохвильове низькотемпературне тепло ще з давніх давен називали – vital rays – проміні життя. Тепловий комфорт в приміщенні, що опалюється, настає вже при температурі 16-18°C, що на 2-3 градуси нижче, ніж при будь-якому конвективному опаленні. Прогрів приміщення рівномірний, без створення різниці температур між полом та стелею [10]. На рисунку 3.6 зображено фото панелі, а в таблиці 3.5 записані її характеристики.



Рисунок 3.7 – Фото ІЧ панелі Елtron ІІ-600МП

Таблиця 3.5 – Характеристики ІЧ панелі Елtron ІІ-600МП

Габаритний розмір, мм	600x1250x25
Потужність	600 Вт
Площа обігріву	15,5 кв.м.
Робоча температура поверхні	до 110 °С
Напруга живлення	~ 220 В ± 10%
Вага	8,7 кг
Площа тепловипромінюваної поверхні	0,75 кв.м.

Також необхідно виділити сервер для програмного додатку Home Assistant [11]. Це ПЗ з відкритим кодом для автоматизації розумного будинку, що орієнтується на локальне управління і конфіденційність. Home Assistant буде відстежувати стан усіх пристроїв. Керуються усі пристрої за допомогою єдиного, зручного для мобільних пристроїв інтерфейсу. Home Assistant дозволяє керувати всіма своїми пристроями, не зберігаючи жодних даних у хмарі. Можна налаштувати розширені правила управління пристроями.

По суті, Home Assistant система інтеграції та автоматизації пристроїв розумного будинку. Написана на мові програмування Python і має відкритий вихідний код. Має понад тисячу інтеграцій пристроїв різних брендів. Для полегшення деяких автоматизацій, а так само для часто викликаються дій в системі представлений такий інструмент, як сцени. Не дивлячись на самодостатність Home Assistant, його можна використовувати для взаємодії з іншими популярними розумними будинками. На даний момент підтримуються наступні інтеграції: Google Home, Apple Homekit. Може бути встановлений на Raspberry Pi або сервер чи комп'ютер на ОС Linux.

Для серверу був обраний комп'ютер Vinga Advanced A0248 [12]. На рисунку 3.7 зображено фото комп'ютера, а в таблиці 3.6 записані його характеристики.



Рисунок 3.8 – Фото Vinga Advanced A0248

Таблиця 3.6 – Характеристики Vinga Advanced A0248

Тип ПК	Для роботи та навчання
Серія процесора	AMD Ryzen 3
Кількість ядер	4 ядра
Частота процесора, ГГц	3.6
Сокет	AM4
Чіпсет	AMD A320
Тип відеокарти	вбудована
Об'єм встановленої пам'яті	8 ГБ
Кількість слотів	2
Тип пам'яті	DDR4
Частота встановленої пам'яті	2666 MHz
Максимальний об'єм пам'яті	32 ГБ
Об'єм SSD	120 GB
Зовнішні порти і роз'єми	1 x RJ45, 4 x USB 2.0, 4 x USB 3.0, 1 x HDMI, 1 x VGA, 1 x Headphone, 1 x Microphone, 3 x Audio, 2 x PS/2
Потужність блоку живлення	400 Вт

3.5 Висновки по розділу

Під час синтезу системи контролю «розумного опалення» першим кроком було встановлення вимог до системи, створення функціональної структури. Після чого була розроблена функціональна схема. Далі обиралися пристрої до системи, а саме пристрої «Xiaomi Mi Smart Home».

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ «РОЗУМНОГО ОПАЛЕННЯ» КОМП'ЮТЕРНИХ КЛАСІВ

4.1 Призначення й область застосування програмного забезпечення

Програмне забезпечення виступає в якості центрального концентратора контролерів «розумного дому», який має всі стандартні функції створення умовних операторів, які можна очікувати від сучасної платформи домашньої автоматизації для управління простою технологією розумного будинку, включаючи шлюз і міст (перетворювач протоколів) для пристроїв і служб, які використовують різні IoT технології, передові автоматизації будівель, управління сигналізацією про охоронної сигналізації та домашньої безпеки систем і моніторинг енергетичних вимірювальних приладів (будинки енергії моніторів, а також центральних інтелектуальних лічильників) та інших датчиків. Він надає системи дій і сценаріїв на основі правил для створення автоматизації, з обробкою часу і умов подій, плануванням завдань, повідомленнями і голосовим управлінням, а також функціями для прямих дій і дій за запитом.

Область застосування – кафедра «ІТКІ», вищого навчального закладу НТУ «ДП».

4.2 Обґрунтування технічних характеристик програми

4.2.1 Постановка завдання на розробку програми

Програма повинна управляти системою контролю «розумного опалення», а саме активувати сценарії та скрипти автоматизації, бути зв'язком між «розумними» пристроями. Проводити моніторинг цих пристроїв, виводити данні на сервер, на якому знаходиться програма. Вимикати прилади що вийшли з ладу або були пошкодженні. Повідомляти про це на сервер, в додаток або на переносний пристрій через локальну мережу.

4.2.2 Опис алгоритму функціонування програми

Доцільно буде виконати опис функціонування програми за допомогою алгоритму. Алгоритм – набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій. Система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час. Для візуалізації алгоритмів часто використовують блок-схеми, що якнайкраще підходять для візуалізації функціонування програми. Він зображений на рисунку 4.1.



Рисунок 4.1 – Схема алгоритму функціонування програми

В алгоритмі можна наглядно побачити роботу сценаріїв автоматизації. Першою справою програми є старт, після чого вона відразу починає моніторинг приладів системи «розумне опалення», якщо помічає зміну датчика руху чи датчика температури, що прийняв певне значення, включає один з сценаріїв автоматизації, який відповідає поставленим умовам. Після виконання автоматизації – повідомляє до додатку Home Assistant, що програма зробила. Якщо ж оброблені дані з датчиків не змінилися чи змінилися, але не відходять умовам запуску автоматизації, то програма залишається в стані спокою і повідомляє про це додатку.

4.3 Опис розробленої програми

4.3.1 Загальні відомості

Для розробки програми контролю «розумних» приладів перш за все необхідно обрати мову програмування.

Програма буде написана за допомогою мови YAML, так як конфігурації додатка Home Assistant написані цією ж мовою.

YAML (рекурсивне скорочення від «YAML Ain't Markup Language») [13] - це зручна для читання мова серіалізації даних. Він зазвичай використовується для файлів конфігурації та в додатках, де дані зберігаються або передаються. YAML націлений на багато тих самих програм зв'язку, що і Extensible Markup Language (XML), але має мінімальний синтаксис, який навмисно відрізняється від SGML. Він використовує відступи у стилі Python для позначення вкладеності, так і більш компактний формат, який використовує [...] для списків і {...} для карт, що робить YAML надмножиною JSON.

Вперше YAML був запропонований Кларком Евансом у 2001 році, який спроектував його разом з Ingy döt Net та Oren Ben-Kiki. Спочатку говорилося, що YAML означає ще одну мову розмітки (Yet Another Markup Language), посилаючись на своє призначення як мову розмітки з ще однією конструкцією (Yet Another – ідіоматично означає у назві комп'ютерної

програми, організації чи події, що вона є неоригінальною), але потім він був перероблений як YAML Ain't Markup Language, рекурсивне скорочення, щоб розрізнити його призначена як орієнтована на дані, а не як розмітка документа.

Код програми можна написати декількома способами, а саме безпосередньо через систему Linux, на якій встановлений додаток Home Assistant, корегуючи файли за допомогою утиліти *nano*, або за допомогою компонента додатку – File Editor, він зображений на рисунку 4.2. Через основну властивість компоненту – перевірка написаного коду на помилки, а також через відображення всіх знаків, краще використовувати File Editor.

Текст програми буде записуватись та зберігатися на сервері.

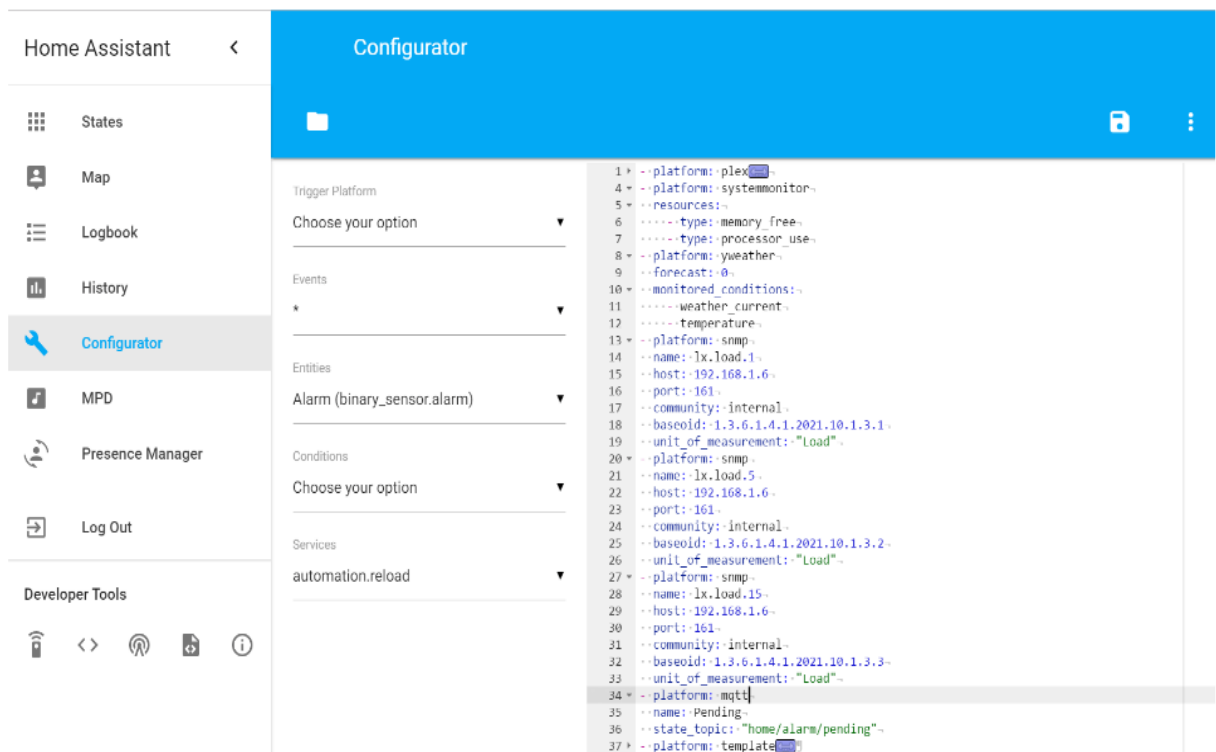


Рисунок 4.2 – Фото компоненту File Editor

4.3.2 Функціональне призначення

Програмне забезпечення системи контролю «розумне опалення» для кафедри «ІТКІ» виконуватиме такі функції:

- моніторинг статусу роботи датчиків, приладів та їх компонентів;
- замір температури в комп'ютерних класах кафедри;

- перевірка наявності руху в приміщеннях в вихідні та у вільний час, поза часами аудиторних занять;
- самостійне ввімкнення/вимкнення приладів обігріву згідно до налаштувань системи
- ручне або віддалене редагування налаштувань системи обігріву
- оповіщення щодо перевищення температури приладів, їх пошкодження, або вихід з ладу
- самостійне вимкнення приладів, якщо вони були пошкоджені, або вийшли з ладу.

Також основною функцією програми є надання потрібних можливостей управління звичайному користувачеві та адміністратору системи.

4.3.3 Опис логічної структури програми

Розробимо та розглянемо алгоритми певних елементів системи:

На рисунку 4.3 зображений алгоритм активування опалення. Програма перевіряє стан датчика температури, якщо він показує температуру 17°C або вище, то ніяких дій проводити не треба, а в додаток Home Assistant надходить повідомлення про те, що панелі не працюють. Коли датчик показує температуру меншу заданої виконується підключення розетки до мережі живлення від чого відбувається і включення ІЧ панелей. Такий спосіб увімкнення панелей обумовлений тим, що вони не мають модифікацій для підключення до системи керування напряму. В додатку відображається оповіщення, що опалення включено.

Текст частини програми, що відповідає умовам цього алгоритму:

```
- alias: heating_on_normal_130 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # Умова або декілька умов
    - platform: state # Вибір платформи, зміна статусу
      entity_id: sensor.temperature_sensor_class_130 # Датчик температури №1
(1/130)
```

below: 17 # Температура менша 17°C

- platform: time_pattern # Регулярний запуск
- minutes: "/5" # Кожні 5 хвилин

condition: # Додаткова умова або декілька

- condition: state # Вибір платформи, зміна статусу
- entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
- state: 'off' # Статус вимкнена
- condition: state # Вибір платформи, зміна статусу
- entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
- state: 'off' # Статус вимкнена

action: # Виконання дії

- service: switch.turn_on # Ввімкнення
- entity_id: # Розетки комп'ютерного класа 1/130
- switch.plug_class_130_1
- switch.plug_class_130_2

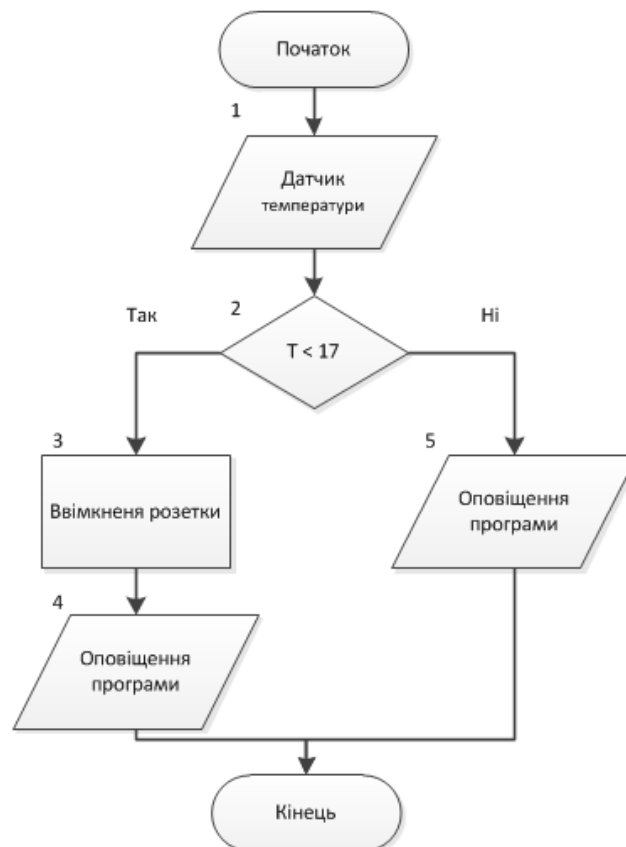


Рисунок 4.3 – Схема алгоритму увімкнення опалення

На рисунку 4.4 зображений алгоритм припинення роботи ІЧ панелей. Якщо під час роботи розетки температура, що показує датчик є меншою за задане значення, то проводити додаткові дії не потрібно, а в додатку Home Assistant досі буде оповіщення про роботу опалення. Після нагріву кімнати до потрібної температури, розетки автоматично вимикаються та у додатку зв'являється повідомлення про припинення роботи панелей.

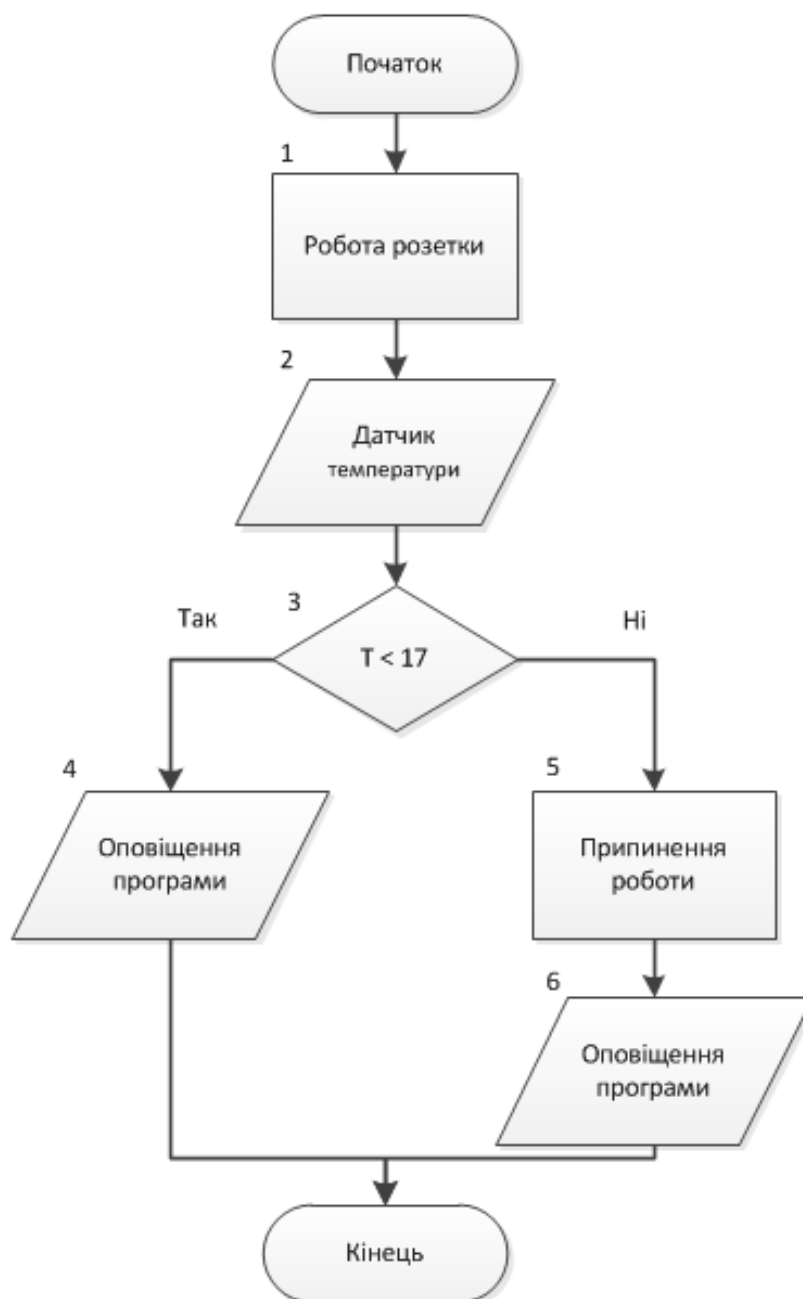


Рисунок 4.4 – Схема алгоритму вимкнення опалення

Текст програми припинення роботи ІЧ панелей:

```
- alias: heating_off_normal_131 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # Умова або декілька умов
- platform: state # Вибір платформи, зміна статусу
  entity_id: sensor.temperature_sensor_class_131 # Датчик температури №1
(1/131)
  above: 16 # Температура більша 16°C
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: state # Вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
  state: 'on'# Статус ввімкнена
- condition: state # Вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
  state: 'on'# Статус ввімкнена
action: # Виконання дії
- service: switch.turn_off # Вимкнення
  entity_id: # Розетки комп'ютерного класа 1/131
  - switch.plug_class_131_1
  - switch.plug_class_131_2
```

У роботі програми повинні бути додаткові режими роботи, це «консультація» та робота «поза часом занять». Режим «консультації» може спрацювати з 7-ї години ранку і до половини 9-го вечора, в суботу та неділю, а режим «поза часом занять» – з 6-ї ранку і до 8-и годин та з 20:25 і до 22:00, в будні.

На рисунку 4.5 зображена схема вибору режиму «консультація». Для цього необхідні дані з датчику руху. У додатку Home Assistant присутній моніторинг датчика руху. Якщо рух відсутній, то і додаткових дій не

потрібно, лише оповіщення програмою в додаток про відсутність рухів. Коли на датчачі з'являється рух, то програма перевіряє який зараз день тижня, тобто умову включення режиму. Після отриманих результатів програмою, у додатку з'явиться оповіщення про обраний режим роботи. Такий самий алгоритм дійсний до роботи режиму «поза часом занять».

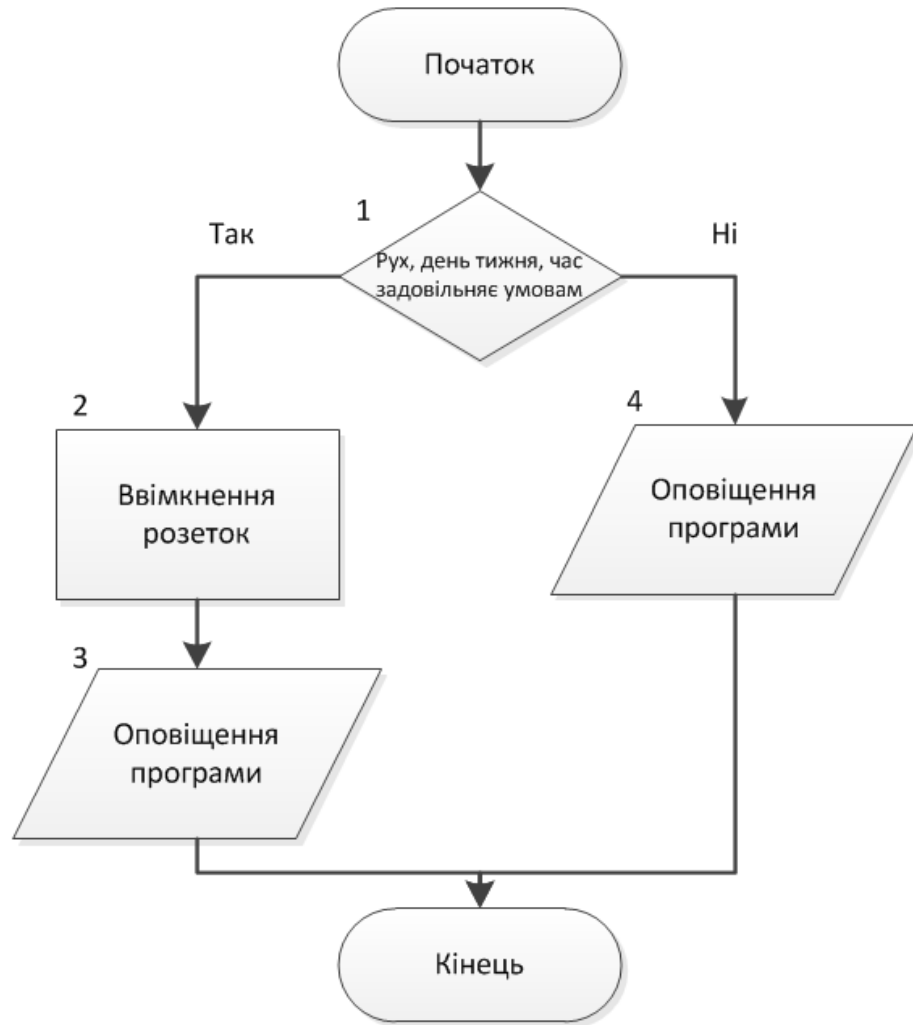


Рисунок 4.5 – Схема алгоритму ввімкнення режиму «консультант»

Текст програми про вибір додаткового режиму роботи:

- alias: heating_on_konsultaciya_130 # Назва автоматизації
- initial_state: true # Статус що показує робить автоматизація чи ні
- trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
- entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)

```

to: 'on' # Э рух
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
  state: 'off'# Статус вимкнена
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
  state: 'off'# Статус вимкнена
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
  - sat
  - sun
- condition: time # вибір платформи, зміна часу або дня
  after: '07:00:00' # після
  before: '20:30:00' # перед
action: # Виконання дії
- service: switch.turn_on # Ввімкнення
  entity_id: # Розетки комп'терного класа 1/130
  - switch.plug_class_130_1
  - switch.plug_class_130_2

```

Якщо ж датчик руху сповістив про рух в інший час – то буде ввімкнено режим «тривога», під час якого спрацює динамік на хабі Xiaomi та буде надіслано повідомлення в «Телеграмі» до чату охорони або викладача. Також у додатку відобразиться ввімкнений режим. Схема алгоритму дій однакова для обох режимів «консультація» та «поза часом занять», якщо умови їх роботи не повністю відповідають.

Відключення режиму «тривога» можлива двома способами.

Перший – система сама вимкне цей режим, після того як у класі зникне рух, а другий – надіслання повідомлення до чату телеграм-бота, що оповістив про включення режиму «тривоги». На рисунку 4.7 зображений алгоритм автоматичного вимкнення режиму «тривоги».

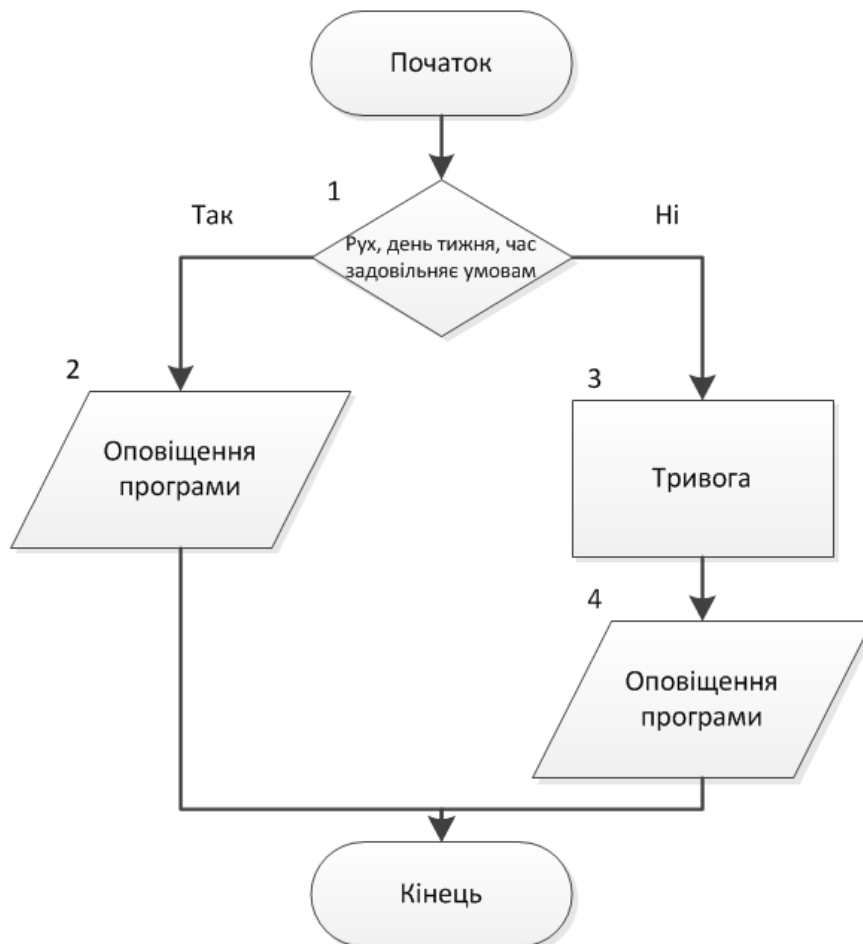


Рисунок 4.6 – Схема алгоритму роботи додаткових режимів

Текст програми спрацювання режиму «тривога»:

- alias: allarm_1_130 # Назва автоматизації
- initial_state: true # Статус що показує робить автоматизація чи ні
- trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
 - entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
 - to: 'on' # Є рух
- platform: time_pattern # Регулярний запуск
 - minutes: "/5" # Кожні 5 хвилин

condition: # Додаткова умова або декілька

- condition: time # вибір платформи, зміна часу або дня

weekday: # дні

- mon

- tue

- wed

- thu

- fri

- condition: time # вибір платформи, зміна часу або дня

after: '22:05:00' # після

before: '05:55:00' # перед

action: # Виконання дії

- service: xiaomi_aqara.play_ringtone # Включення рингтону

data:

gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу

ringtone_id: 7 # порядковий номер рингтону, атака з повітря

ringtone_vol: 88 # гучність, у відсотках

- service: script.turn_on # Активування скрипта

entity_id: script.send_message_1 # "Увага! Помічений рух в класі

1/130!"

Текст програми вимкнення режиму «тривога»:

- alias: allarm_off_1_131 # Назва автоматизації

initial_state: true # Статус що показує робить автоматизація чи ні

trigger: # умова або декілька умов

- platform: state # вибір платформи, зміна статусу

entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)

to: 'off' # Немає руху

- platform: time_pattern # Регулярний запуск

minutes: "/5" # Кожні 5 хвилин

condition: # Додаткова умова або декілька

```

- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
    - mon
    - tue
    - wed
    - thu
    - fri
- condition: time # вибір платформи, зміна часу або дня
  after: '22:05:00' # після
  before: '05:55:00' # перед
action: # Виконання дії
- service: xiaomi_aqara.stop_ringtone # Виключення рингтону
  data:
    gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу
- service: script.turn_on # Активування скрипта
  entity_id: script.send_message_4 # "Увага! Руху в класі 1/131 більше немає, сирена вимкнена!"

```

Текст програми вимкнення режиму «тривога» за допомогою надіслання команди в чат телеграм-бота:

```

- id: Allert_off
  alias: Повідомлення до телеграм-бота
  initial_state: true
  trigger:
    platform: event
    event_type: telegram_command
    event_data:
      command: '/allrtoff'
  action: # Виконання дії
    - service: script.turn_on # Активування скрипта
      entity_id:

```

```

- script.send_message_5 # "Увага! Сирена вимкнена!"
- service: xiaomi_aqara.stop_ringtones # Виключення рингтону
data:
gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу

```

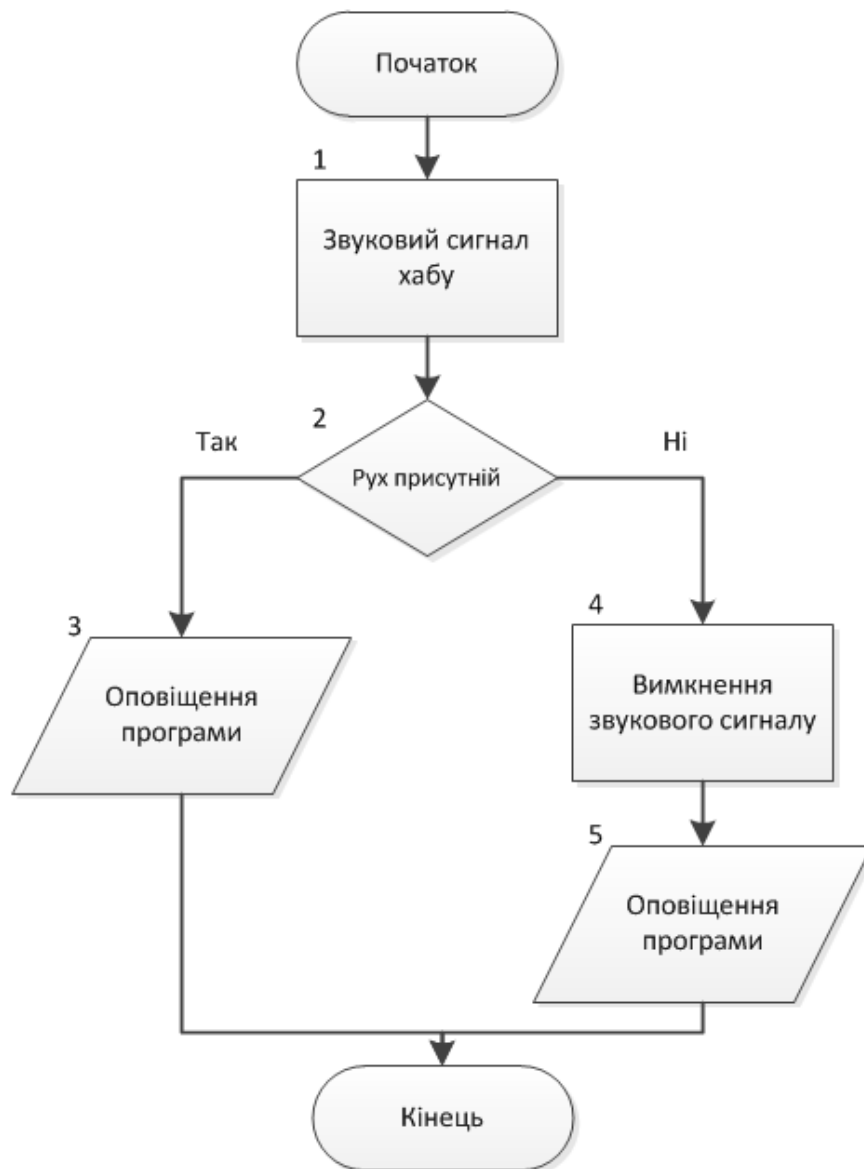


Рисунок 4.7 – Схема алгоритму вимкнення режиму «тривога»

Перевірка роботи додаткових режимів відбувається після включення одного з них кожні 5 хвилин. На рисунку 4.8 зображено схему роботи програми під час активного додаткового режиму. Під час увімкненого опалення датчик перевіряє чи наявний рух в класі. Якщо рух є – додаткових дій не потрібно, а в додатку Home Assistant відображається стан роботи

одного з режимів. Коли рух відсутній – програма вимикає розетки з панелями, в додатку з’являється повідомлення про припинення опалення.

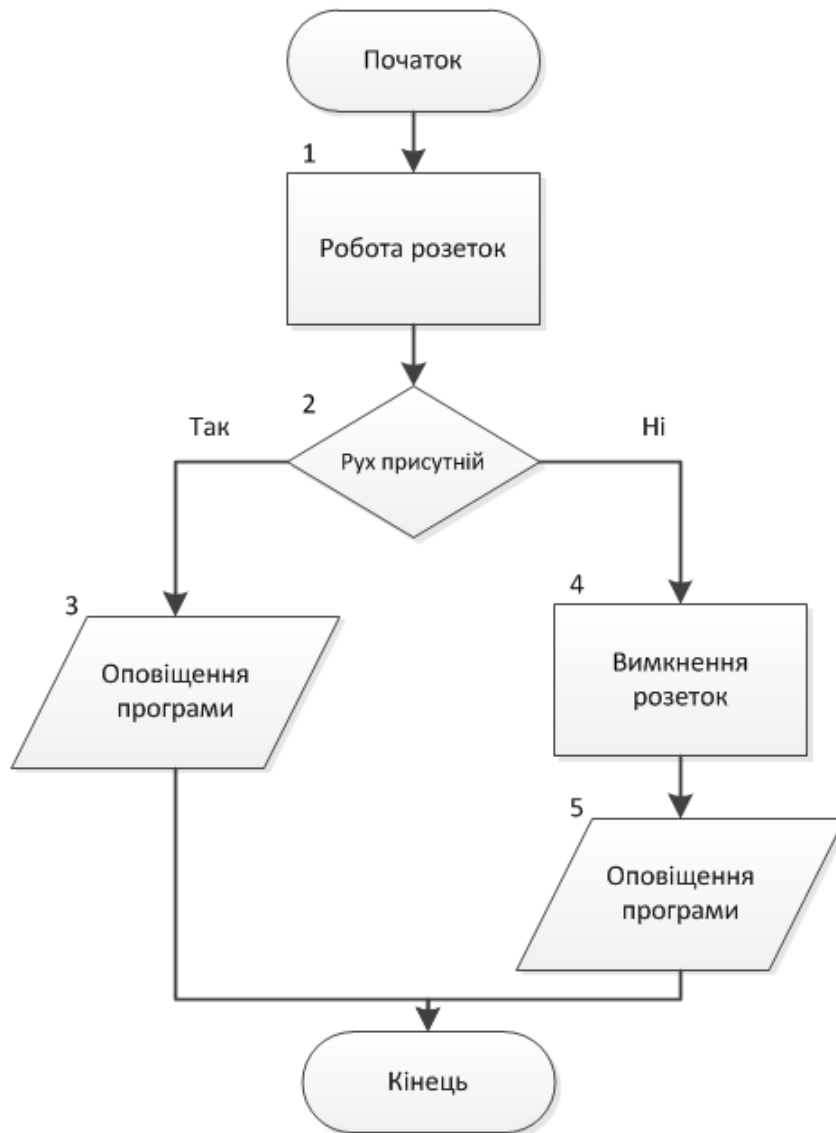


Рисунок 4.8 – Схема алгоритму вимкнення додаткових режимів

Приклад тексту програми про припинення роботи режиму «поза часу занять»:

```

- alias: heating_off_after_study_131 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
  
```



```

to: 'off' # Э рух
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
  state: 'on'# Статус вимкнена
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
  state: 'on'# Статус вимкнена
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
  - mon
  - tue
  - wed
  - thu
  - fri
- condition: or # Одна з декількох умов, АБО
  conditions:
  - condition: time # вибір платформи, зміна часу або дня
    after: '06:00:00' # після
    before: '08:30:00' # перед
  - condition: time # вибір платформи, зміна часу або дня
    after: '20:25:00' # після
    before: '22:00:00' # перед
action: # Виконання дії
- service: switch.turn_off # Ввимкнення
  entity_id: # Розетки комп'терного класа 1/131
  - switch.plug_class_131_1
  - switch.plug_class_131_2

```

4.3.4 Використовувані технічні засоби

В роботі програми будуть задіяні усі «розумні» прилади, що входять до складу системи контролю (датчики руху, температури, хаб, розетки), додаток Home Assistant, як програмний центр контролю, а також сервер на якому встановлений додаток, процесор повинен мати частоту не меншу за 1.2GHz, ОЗП – від 512 МВ , накопичувач з об'ємом не менше 30GB та доступ в мережу на швидкості від 100Мбіт/с.

4.3.5 Виклик і завантаження

Програма може бути викликана прямо з серверу на якому знаходиться додаток Home Assistant, безпосередньо з веб-інтерфейсу додатку з будь-якого пристрою підключеного до локальної мережі серверу, за адресою <http://192.168.59.132:8123>.

Вхід в програму відбувається в тому випадку, коли один з датчиків отримує встановлене значення (температуру або наявність руху) або один з приладів стає пошкодженим чи виходить з ладу.

Додаток Home Assistant в сумі з програмою контролю та автоматизації використовує 240МВ оперативної пам'яті та займає 400МВ без обліку бази даних попередніх повідомлень про зміни роботи програми, стану системи.

4.4 Висновки по розділу

Для системи контролю «розумного опалення» кафедри «ІТКІ» була розроблена програма моніторингу та управління, за допомогою зіставлених алгоритмів функціонування системи та окремих автоматизацій пристроїв. Був обраний склад використовуваних технічних засобів, встановлено як викликається програма та на скільки вона завантажує сервер.

5 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

5.1 Мета і завдання експерименту

Мета експерименту: встановлення достатньої температури у системі «розумного опалення» навчальних приміщень першого корпусу НТУ «ДП», кафедри «ІТКІ», в різних умовах навколишнього температурного режиму.

Завдання експерименту: дослідним шляхом перевірити адекватність моделі системи контролю та її складових, встановити найбільш коректну температуру, якої необхідно дотримуватися системі для опалення, в певних кліматичних умовах.

5.2 Методика експерименту

Експеримент буде проводитися за допомогою створеної моделі контролю температурного режиму в 2-му розділі. Запуск та проведення експерименту відбувається у середовищі Simulink.

Для кожного місяця опалювального сезону, з листопаду до березня, будуть проводитись повтори експерименту з різною встановленою температурою в класах при зміні температури, що відповідає певному місяцю.

5.3 Оцінка результатів експерименту

Зміна температури відбуваються за законом синусоїди, тому в моделі перемінною буде Sine Wave. Для кожного місяця, за даними метеоцентрів, були обрані середні показники температури та приблизну амплітуду змін температури.

Спочатку було перевірено роботу окремих складових моделі, а саме спрацювання «Датчику температури», нагрівання «Нагрівального пристрою» та робота підсистеми «Комп'ютерний клас». Всі успішно пройшли тести і були знову підключені один до одного.

Для опалення класів, будуть порівнюватися температури 15°C, 17°C та 19°C.

Листопад: середньомісячна температура дорівнює 6°C, максимальна – 19°C, мінімальна – -2°C, тому приймемо амплітуду зміни – 11,5°C. На рисунку 5.1 зображені результати поведінки системи за листопад, з них можна побачити, що при встановленій температурі на датчику в 19°C система постійно буде включена. Це може призвести до швидкого погіршення стану нагрівальних приладів та розеток, які постійно перемикають свої положення. При встановлені на датчику 17°C, система вже має простір для паузи в роботі нагрівачів, приблизно в 4 години. При встановлені датчику на температуру в 15°C, система знаходиться на паузі роботи нагрівачів приблизно третину дня.

З цього слідує, що оптимальною температурою для обігріву в листопаді буде не менше 15°C, так як ця цифра є мінімально допустимою температурою під час проведення занять. В не робочий час рекомендується змінювати температуру обігріву до мінімального значення.

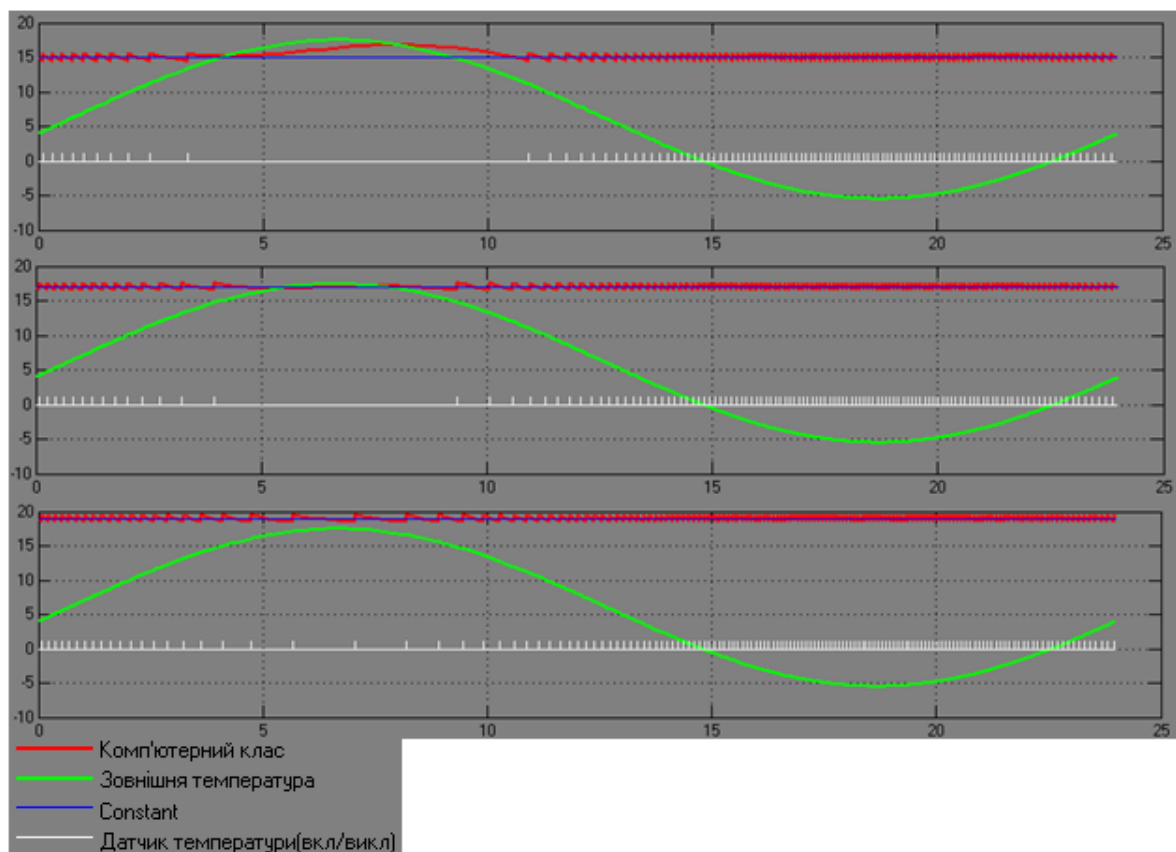


Рисунок 5.1 – Результати експерименту за листопад

Грудень: середньомісячна температура дорівнює 3°C , максимальна – 15°C , мінімальна – -8°C , тому приймемо амплітуду зміни – 12°C . На рисунку 5.2 зображені результати поведінки системи за грудень, з них можна побачити, що при встановленій температурі на датчику в 19°C система постійно буде включена. Наслідки такої роботи вже відомі. При встановлені на датчику 17°C , система, також, буде працювати постійно. При встановлені датчику на температуру в 15°C , система матиме паузу в роботі нагрівачів.

З цього слідує, що система буде постійно функціонувати з виставлення мінімально допустимого значення опалення. Тому буде наполегливо рекомендовано зменшувати температуру обігріву в не робочий час.

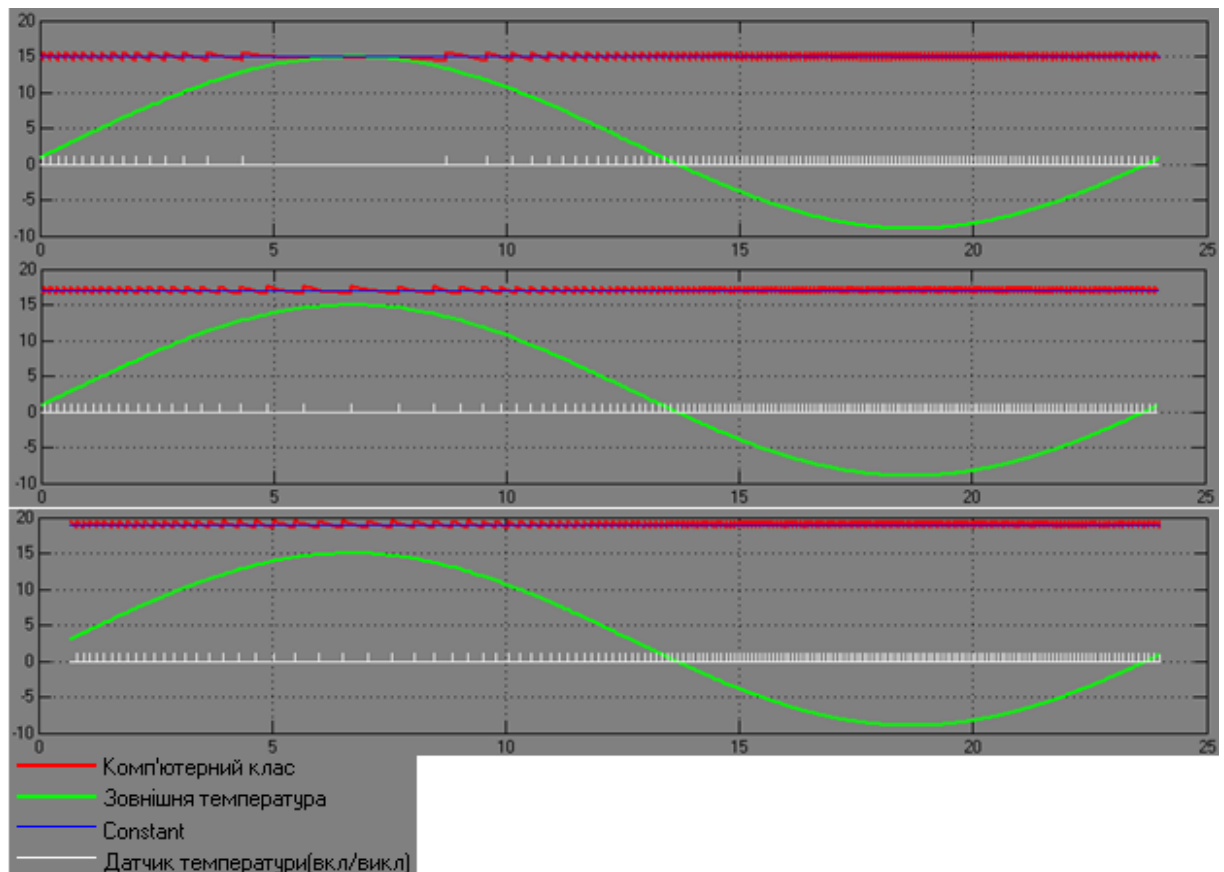


Рисунок 5.2 – Результати експерименту за грудень

Січень: середньомісячна температура дорівнює -4°C , максимальна – 5°C , мінімальна – -18°C , тому приймемо амплітуду зміни – $11,5^{\circ}\text{C}$. На

рисунку 5.3 зображені результати поведінки системи за січень. Результати цього експерименту такі ж як і в грудні.

З цього слідує, система буде так само навантажена. Тому і рекомендації будуть ідентичні.

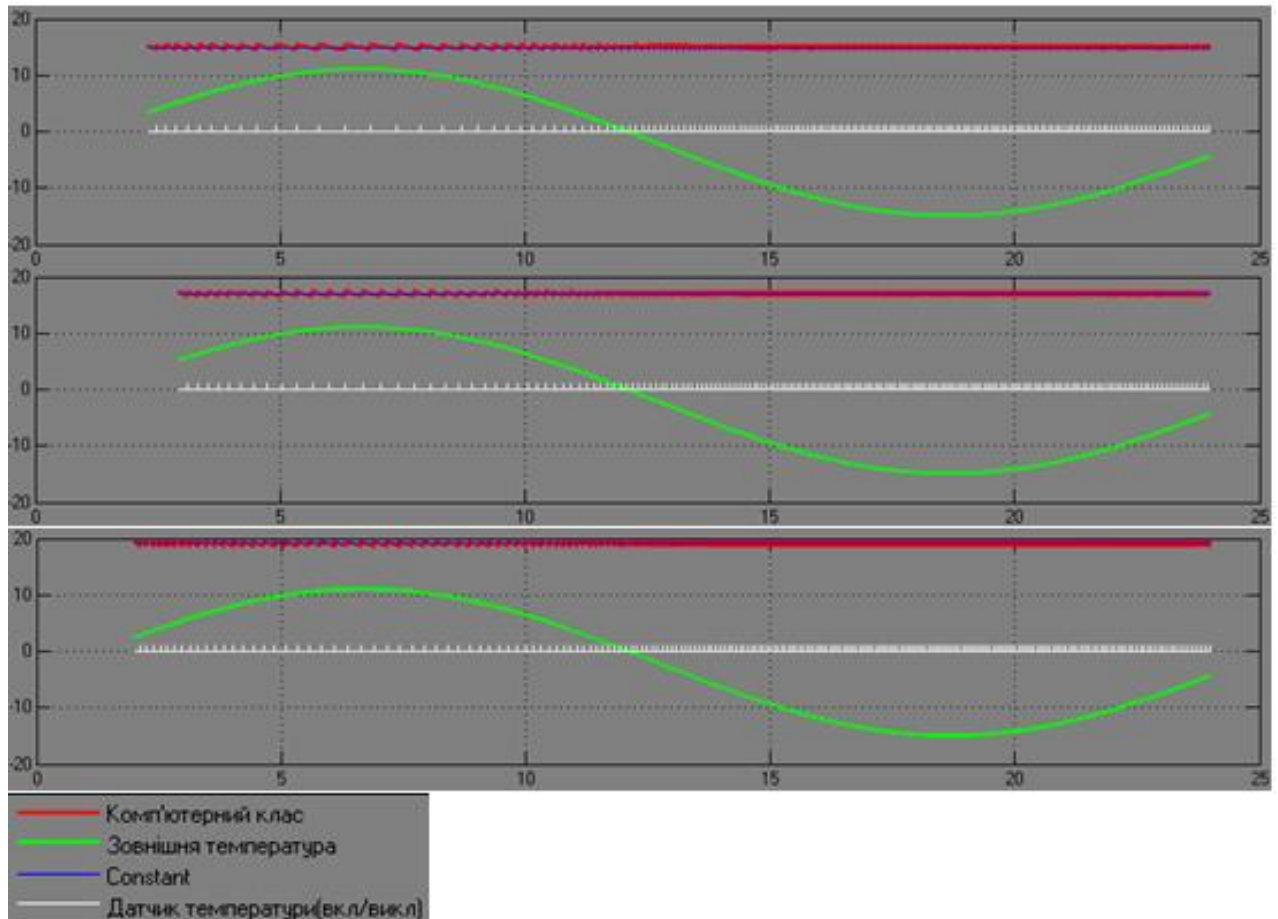


Рисунок 5.3 – Результати експерименту за січень

Лютий: середньомісячна температура дорівнює -1°C , максимальна – 5°C , мінімальна – -14°C , тому прийmemo амплітуду зміни – $9,5^{\circ}\text{C}$.

Березень: середньомісячна температура дорівнює 3°C , максимальна – 13°C , мінімальна – 1°C , тому прийmemo амплітуду зміни – 6°C .

На рисунку 5.4 і 5.5 зображені результати експерименту за лютий і за березень відповідно. Експеримент за березень, як і за всю зиму, має ідентичні результати, тому наполегливо рекомендується в часи, що не є робочими, знижувати температуру обігріву до мінімально допустимої.

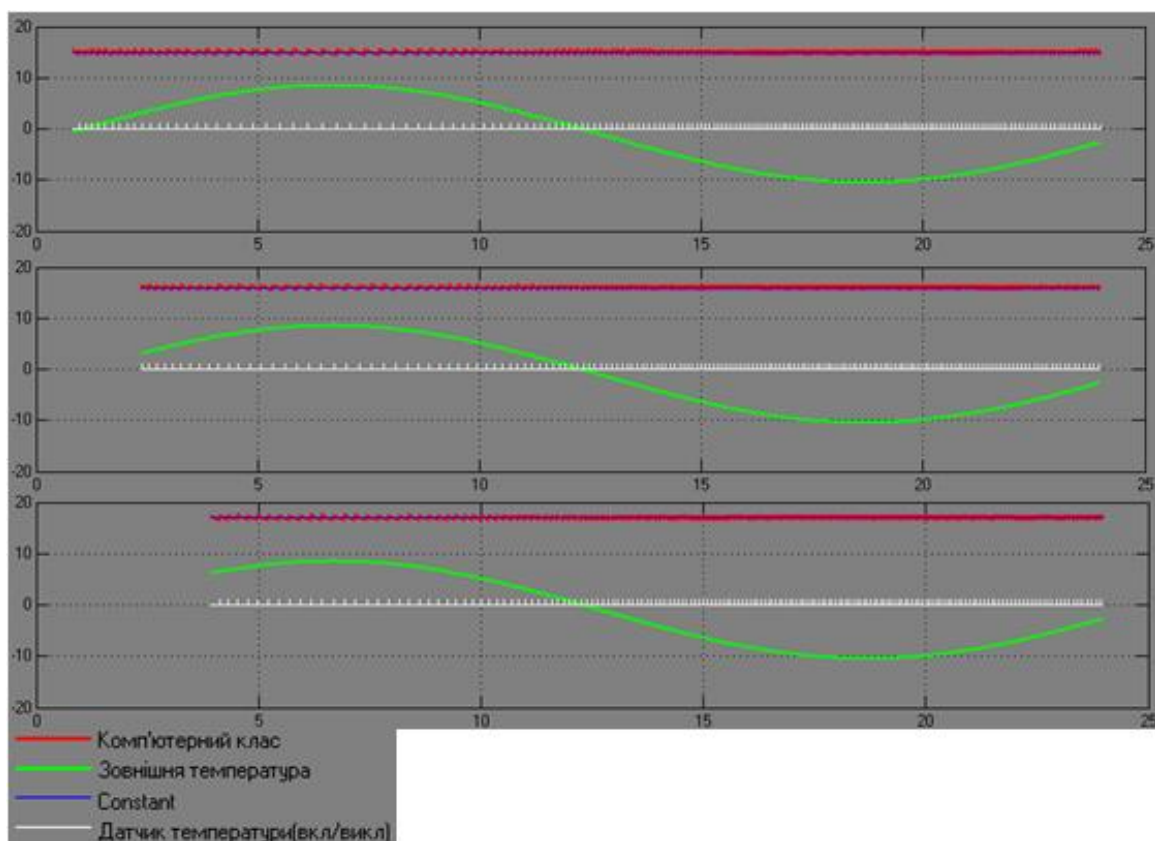


Рисунок 5.4 – Результати експерименту за лютий

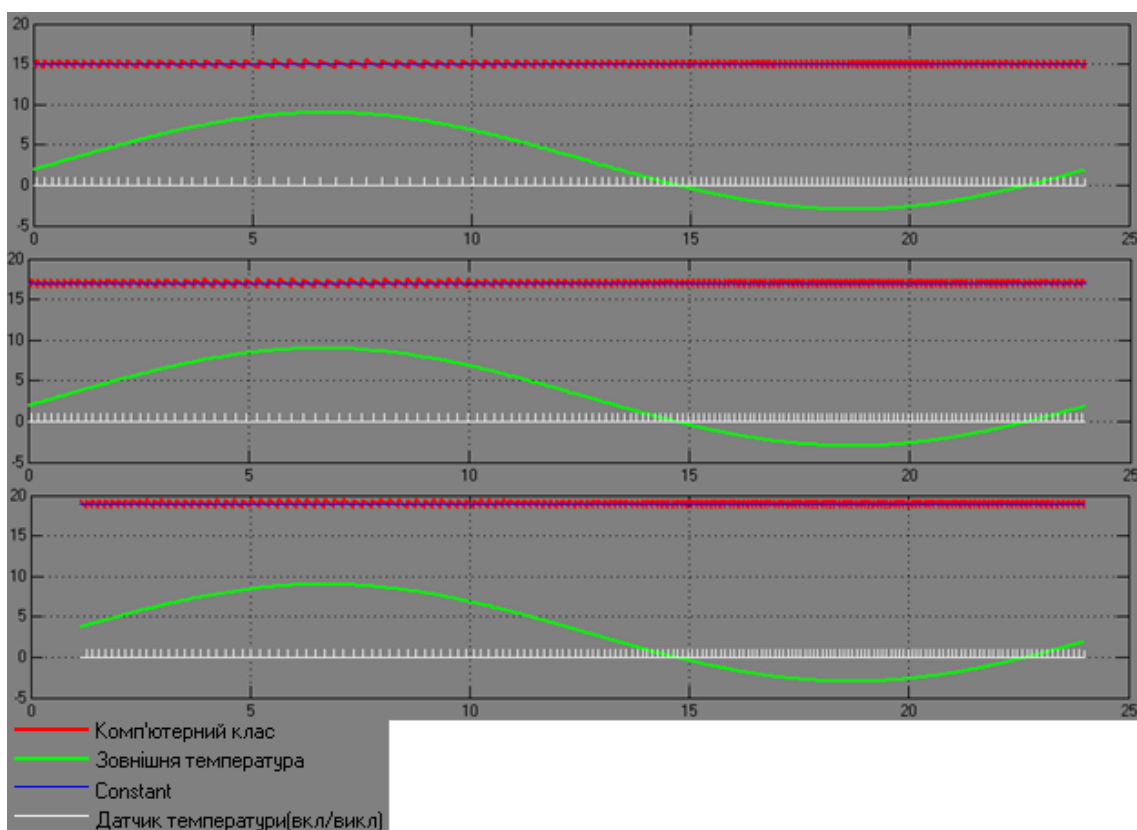


Рисунок 5.5 – Результати експерименту за березень

5.4 Аналіз отриманих результатів

В підсумку отриманих результатів експерименту можна зазначити, що оптимальною температурою для опалення комп'ютерних класів на кафедрі «ІТКІ» є 17°C , при якій тільки в листопаді буде самостійно припинена робота нагрівальних пристроїв. Також було рекомендовано зменшувати температуру обігріву в не робочі години. Це забезпечує більшу ефективність роботи приладів, а також має економічний ефект.

5.5 Висновки по розділу

В ході написання експериментального розділу, були обрані завдання та мета самого експерименту, було описано його методику та обсяг. Експеримент виконувався на математичній моделі, що була розроблена у 2-му пункті. Були отримані результати роботи, за допомогою яких був проведений аналіз всього експерименту.

ВИСНОВКИ

Кваліфікаційна робота є завершеною науковою роботою, в якій вирішена науково-практична задача створення системи «розумного опалення» в Національному технічному університеті «Дніпровська політехніка» на кафедрі «інформаційних технологій та комп'ютерної інженерії» на основі технології «розумний дім» та аналіз структури і параметрів системи.

Основні висновки і результати роботи полягають у наступному:

1. Визначено основні плюси впровадження системи «розумного опалення» до об'єкту дослідження завдяки методам аналізу і синтезу, а також методу порівняння.

2. Розроблено математична модель за принципом моделювання Define. В ній висунуті: цілі – математична модель потрібна для системи контролю щоб спостерігати, як зміна зовнішньої температури впливає на внутрішню температуру та для дослідження ефекту параметрів, що змінюються, на внутрішній температурі; компоненти – для системи опалення з «розумними» пристроями – це термостат, або датчик температури, зовнішнє середовище та клімат у класі.

3. Синтезована комп'ютерна система контролю «розумного опалення», хоча й має повний перелік необхідних апаратних засобів, але без додатку Home Assistant не має можливості до масштабованості у систему «розумного» з пристроями інших представників.

4. Розробка програмного забезпечення автоматизації системи «розумного опалення» проводилась в компоненті додатку Home Assistant, File Editor, що має схожий функціонал з відомими платформами розробки програмного забезпечення.

5. Мова програмування YAML якнайкраще підходить до створення скриптів автоматизації розумного дому, спираючись на конструкцію: trigger, condition, action.

6. Під час проведення експерименту встановлено які додаткові дії необхідні для додаткового економічного ефекту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Вільна енциклопедія «Вікіпедія». Стаття про опалення [Електронний ресурс]. URL: <https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BE%D0%BF%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5> (дата звернення: 25.09.2020);
2. Семенцова А.М. «Проблемы в системах отопления»; Ахметов И.Г. «Молодой ученый» Международный научный журнал № 5 (243) / 2019 с. -26;
3. Интернет-магазин отопления и водоснабжения UniDim. Умный дом: отопление, unidim.com.ua [Електронний ресурс]. URL: <https://unidim.com.ua/blog/umnyy-dom-otoplenie> (дата звернення: 03.10.2020);
4. Интернет-магазин VENCON. ТОП 5: Рейтинг лучших систем «Умный дом» по производителям 2021 года [Електронний ресурс]. URL: <https://vencon.ua/articles/rejting-sistem-umnyy-dom-po-proizvoditelyam> (дата звернення: 07.10.2020);
5. Сайт Національний ТУ «Дніпровська політехніка». Історія створення кафедри [Електронний ресурс]. URL: https://it.nmu.org.ua/ua/about_department_ist/history_department.php (дата звернення: 12.10.2020);
6. Интернет-магазин ЦИТРУС. Модуль управления умным домом Mi Smart Home Gateway 2 [Електронний ресурс]. URL: https://www.citrus.ua/ustrojstva-umnogo-doma/mi-smart-home-gateway-2-603540.html?gclid=CjwKCAiAt9z-BRBCEiwA_bWv-LudNw8HuIrrqd5DldtbYvroI6cgc6EKqd-lKeKT7mLtbze6_KncBoCfpwQAvD_BwE&gclsrc=aw.ds (дата звернення: 13.10.2020);
7. Интернет-магазин SMARTS.UA Датчик Температуры и Влажности Xiaomi MijiaTemperature And Humidity Sensor WSDCGQ01LM [Електронний ресурс]. URL: https://smarts.ua/product/datchik-temperature-i-vlazhnosti-xiaomi-mijiateperature-and-humidity-sensor-wsdcgq01lm/?gclid=Cj0KCQiA5bz-BRD-ARIsABjT4nh6sgPPc1Czlqu-uVe-bVN-RN_obHNba526R7ugf8f3BNuUWuIp4I4aAmRUEALw_wcB (дата звернення: 15.10.2020);
8. Интернет-магазин ALLO. Характеристики Розумна розетка Xiaomi Mi Smart Socket 2 ZigBee GMR4004CN (ZNCZ02LM) [Електронний ресурс]. URL: <https://allo.ua/product/rozетка-xiaomi-mi-smart-socket-2-zigbee-gmr4004cn-zncz02lm> (дата звернення: 15.10.2020);

- ресурс]. URL: <https://allo.ua/ua/umnye-rozetki/xiaomi-mi-smart-socket-2-zigbee-gmr4004cn-zncz02lm.html?tab=specs> (дата звернення: 16.10.2020);
9. Интернет-магазин ЦИТРУС. Датчик движения Mijia Move Detector (YTC4016CN) [Електронний ресурс]. URL: <https://www.citrus.ua/ustrojstva-umnogo-doma/datchik-dvizhenie-xiaomi-562196.html?tab=characteristic> (дата звернення: 17.10.2020);
10. Сайт ТОВ "І.К.Елtron". Характеристика Елtron ІІ-600МІІ [Електронний ресурс]. URL: <https://www.eltron-ip.com/company> (дата звернення: 22.10.2020);
11. Портал по тематике умного дома и интернета верей Sprut.ai. Умный дом на Home Assistant [Електронний ресурс]. URL: <https://sprut.ai/client/article/2140> (дата звернення: 24.10.2020);
12. Интернет-магазин IT box Комп'ютер Vinga Advanced A0248 (R3M8INT.A0248) [Електронний ресурс]. URL: https://www.itbox.ua/ua/product/Компьютер_Vinga_Advanced_A0248_R3M8INTA0248-p619556/ (дата звернення: 05.11.2020);
13. Вільна енциклопедія «Вікіпедія». Стаття про мову програмування YAML [Електронний ресурс]. URL: [https://en.wikipedia.org/wiki/YAML#:~:text=YAML%20\(a%20recursive%20acronym%20for,is%20being%20stored%20or%20transmitted](https://en.wikipedia.org/wiki/YAML#:~:text=YAML%20(a%20recursive%20acronym%20for,is%20being%20stored%20or%20transmitted) (дата звернення: 03.12.2020);

ДОДАТОК А

Текст програми комп'ютерної системи контролю «розумного опалення»

**Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
КОМП'ЮТЕРНОЇ СИСТЕМИ КОНТРОЛЮ «РОЗУМНОГО
ОПАЛЕННЯ»**

Текст програмы

804.02070743.200022-01 12 01

Листів 17

АНОТАЦІЯ

Даний документ містить вихідний код програми комп'ютерної системи контролю «розумного опалення» кабінетів.

Текст програми реалізований на мові YAML.

Середовище розробки та налагодження – File Editor.

ЗМІСТ

Configuration.yaml	4
Autimations.yaml	5
Скрипт включення опалення в кабінеті	5
Скрипт виключення опалення в кабінеті	6
Скрипт включення режиму «консультація»	6
Скрипт віклучення режиму «консультація»	7
Скрипт включення режиму «поза часу занять»	8
Скрипт віклучення режиму «поза часу занять»	10
Скрипт віклучення режиму «тривога»	11
Скрипт віклучення режиму «тривога»	13
Ui-lovelace.yaml	16
Вкладка інтерфейсу «Головна»	16
Вкладка інтерфейсу «Погода»	17


```
homeassistant:
  name: Dnipro
  latitude: 48.455439556210976
  longitude: 35.06218735343465
  elevation: 120
  unit_system: metric
  time_zone: Europe/Kiev
# Configure a default setup of Home Assistant (frontend, api, etc)
packages: !include_dir_merge_named includes/packages/

default_config:
  lovelace:
    mode: yaml

# Text to speech
tts:
  - platform: google_translate

group: !include includes/groups.yaml
automation: !include includes/automations.yaml

scene: !include includes/scenes.yaml
recorder: !include includes/recorder.yaml

sensor: !include_dir_merge_list includes/sensor
script: !include_dir_merge_named includes/scripts/

telegram_bot:
  platform: polling
  api_key: '1042440402:AAFUKpbzACvaTCBAMCYCO4wCckrmtBgDrEM'
  allowed_chat_ids:
    - 491250655

notify:

  - name: telegram_bot
    platform: telegram
    chat_id: 491250655
```

- alias: heating_on_normal_130 # Назва автоматизації
- initial_state: true # Статус що показує робить автоматизація чи ні
- trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
 - entity_id: sensor.temperature_sensor_class_130 # Датчик температури №1 (1/130)
 - below: 17 # Температура менша 17°C
- platform: time_pattern # Регулярний запуск
 - minutes: "/5" # Кожні 5 хвилин
- condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
 - entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
 - state: 'off' # Статус вимкнена
- condition: state # вибір платформи, зміна статусу
 - entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
 - state: 'off' # Статус вимкнена
- action: # Виконання дії
- service: switch.turn_on # Ввімкнення
 - entity_id: # Розетки комп'ютерного класа 1/130
 - switch.plug_class_130_1
 - switch.plug_class_130_2

- alias: heating_on_normal_131 # Назва автоматизації
- initial_state: true # Статус що показує робить автоматизація чи ні
- trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
 - entity_id: sensor.temperature_sensor_class_131 # Датчик температури №1 (1/131)
 - below: 17 # Температура менша 17°C
- platform: time_pattern # Регулярний запуск
 - minutes: "/5" # Кожні 5 хвилин
- condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
 - entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
 - state: 'off' # Статус вимкнена
- condition: state # вибір платформи, зміна статусу
 - entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
 - state: 'off' # Статус вимкнена
- action: # Виконання дії
- service: switch.turn_on # Ввімкнення
 - entity_id: # Розетки комп'ютерного класа 1/131
 - switch.plug_class_131_1
 - switch.plug_class_131_2

- alias: heating_off_normal_130 # Назва автоматизації
- initial_state: true # Статус що показує робить автоматизація чи ні
- trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
 - entity_id: sensor.temperature_sensor_class_130 # Датчик температури №1 (1/130)
 - above: 16 # Температура більша 16°C
- platform: time_pattern # Регулярний запуск
 - minutes: "/5" # Кожні 5 хвилин
- condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу

- entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
- state: 'on'# Статус ввімкнена
- condition: state # вибір платформи, зміна статусу
- entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
- state: 'on'# Статус ввімкнена
- action: # Виконання дії
- service: switch.turn_off # Вимкнення
- entity_id: # Розетки комп'ютерного класа 1/130
- switch.plug_class_130_1
- switch.plug_class_130_2

- alias: heating_off_normal_131 # Назва автоматизації
- initial_state: true # Статус що показує робить автоматизація чи ні
- trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
- entity_id: sensor.temperature_sensor_class_131 # Датчик температури №1 (1/131)
- above: 16 # Температура більша 16°C
- platform: time_pattern # Регулярний запуск
- minutes: "/5" # Кожні 5 хвилин
- condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
- entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
- state: 'on'# Статус ввімкнена
- condition: state # вибір платформи, зміна статусу
- entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
- state: 'on'# Статус ввімкнена
- action: # Виконання дії
- service: switch.turn_off # Вимкнення
- entity_id: # Розетки комп'ютерного класа 1/131
- switch.plug_class_131_1
- switch.plug_class_131_2

- alias: heating_on_konsultaciya_130 # Назва автоматизації
- initial_state: true # Статус що показує робить автоматизація чи ні
- trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
- entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
- to: 'on' # Э рух
- platform: time_pattern # Регулярний запуск
- minutes: "/5" # Кожні 5 хвилин
- condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
- entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
- state: 'off'# Статус вимкнена
- condition: state # вибір платформи, зміна статусу
- entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
- state: 'off'# Статус вимкнена
- condition: time # вибір платформи, зміна часу або дня
- weekday: # дні
- sat
- sun
- condition: time # вибір платформи, зміна часу або дня

```

after: '07:00:00' # після
before: '20:30:00' # перед
action: # Виконання дії
- service: switch.turn_on # Ввімкнення
  entity_id: # Розетки комп'ютерного класа 1/130
  - switch.plug_class_130_1
  - switch.plug_class_130_2

- alias: heating_on_konsultaciya_131 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
  - platform: state # вибір платформи, зміна статусу
    entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
    to: 'on' # Є рух
  - platform: time_pattern # Регулярний запуск
    minutes: "/5" # Кожні 5 хвилин
  condition: # Додаткова умова або декілька
  - condition: state # вибір платформи, зміна статусу
    entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
    state: 'off' # Статус вимкнена
  - condition: state # вибір платформи, зміна статусу
    entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
    state: 'off' # Статус вимкнена
  - condition: time # вибір платформи, зміна часу або дня
    weekday: # дні
    - sat
    - sun
  - condition: time # вибір платформи, зміна часу або дня
    after: '07:00:00' # після
    before: '20:30:00' # перед
  action: # Виконання дії
  - service: switch.turn_on # Ввімкнення
    entity_id: # Розетки комп'ютерного класа 1/131
    - switch.plug_class_131_1
    - switch.plug_class_131_2

- alias: heating_off_konsultaciya_130 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
  - platform: state # вибір платформи, зміна статусу
    entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
    to: 'off' # Немає рух
  - platform: time_pattern # Регулярний запуск
    minutes: "/5" # Кожні 5 хвилин
  condition: # Додаткова умова або декілька
  - condition: state # вибір платформи, зміна статусу
    entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
    state: 'on' # Статус ввімкнена
  - condition: state # вибір платформи, зміна статусу
    entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
    state: 'on' # Статус ввімкнена
  - condition: time # вибір платформи, зміна часу або дня

```

```

weekday: # дні
- sat
- sun
- condition: time # вибір платформи, зміна часу або дня
  after: '07:00:00' # після
  before: '20:30:00' # перед
action: # Виконання дії
- service: switch.turn_off # Вимкнення
  entity_id: # Розетки комп'ютерного класа 1/130
  - switch.plug_class_130_1
  - switch.plug_class_130_2

- alias: heating_off_konsultaciya_131 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
  entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
  to: 'off' # Э рух
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
  state: 'on'# Статус ввімкнена
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
  state: 'on'# Статус ввімкнена
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
  - sat
  - sun
- condition: time # вибір платформи, зміна часу або дня
  after: '07:00:00' # після
  before: '20:30:00' # перед
action: # Виконання дії
- service: switch.turn_off # Вимкнення
  entity_id: # Розетки комп'ютерного класа 1/131
  - switch.plug_class_131_1
  - switch.plug_class_131_2

- alias: heating_on_after_study_130 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
  entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
  to: 'on' # Э рух
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
  state: 'off'# Статус вимкнена

```

- condition: state # вибір платформи, зміна статусу
entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
state: 'off'# Статус вимкнена
- condition: time # вибір платформи, зміна часу або дня
weekday: # дні
 - mon
 - tue
 - wed
 - thu
 - fri
- condition: or # Одна з декількох умов, АБО
conditions:
 - condition: time # вибір платформи, зміна часу або дня
after: '06:00:00' # після
before: '08:30:00' # перед
 - condition: time # вибір платформи, зміна часу або дня
after: '20:25:00' # після
before: '22:00:00' # перед
- action: # Виконання дії
 - service: switch.turn_on # Ввімкнення
entity_id: # Розетки комп'ютерного класа 1/130
 - switch.plug_class_130_1
 - switch.plug_class_130_2
- alias: heating_on_after_study_131 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # умова або декілька умов
 - platform: state # вибір платформи, зміна статусу
entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
to: 'on' # Э рух
 - platform: time_pattern # Регулярний запуск
minutes: "/5" # Кожні 5 хвилин
- condition: # Додаткова умова або декілька
 - condition: state # вибір платформи, зміна статусу
entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
state: 'off'# Статус вимкнена
 - condition: state # вибір платформи, зміна статусу
entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
state: 'off'# Статус вимкнена
 - condition: time # вибір платформи, зміна часу або дня
weekday: # дні
 - mon
 - tue
 - wed
 - thu
 - fri
- condition: or # Одна з декількох умов, АБО
conditions:
 - condition: time # вибір платформи, зміна часу або дня
after: '06:00:00' # після
before: '08:30:00' # перед
 - condition: time # вибір платформи, зміна часу або дня

```

    after: '20:25:00' # після
    before: '22:00:00' # перед
action: # Виконання дії
- service: switch.turn_on # Ввімкнення
  entity_id: # Розетки комп'ютерного класа 1/131
  - switch.plug_class_131_1
  - switch.plug_class_131_2

- alias: heating_off_after_study_130 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
  - platform: state # вибір платформи, зміна статусу
    entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
    to: 'off' # Э рух
  - platform: time_pattern # Регулярний запуск
    minutes: "/5" # Кожні 5 хвилин
  condition: # Додаткова умова або декілька
  - condition: state # вибір платформи, зміна статусу
    entity_id: switch.plug_class_130_1 # Розетка №1 (1/130)
    state: 'on'# Статус ввімкнена
  - condition: state # вибір платформи, зміна статусу
    entity_id: switch.plug_class_130_2 # Розетка №2 (1/130)
    state: 'on'# Статус ввімкнена
  - condition: time # вибір платформи, зміна часу або дня
    weekday: # дні
    - mon
    - tue
    - wed
    - thu
    - fri
  - condition: or # Одна з декількох умов, АБО
    conditions:
    - condition: time # вибір платформи, зміна часу або дня
      after: '06:00:00' # після
      before: '08:30:00' # перед
    - condition: time # вибір платформи, зміна часу або дня
      after: '20:25:00' # після
      before: '22:00:00' # перед
action: # Виконання дії
- service: switch.turn_off # Ввімкнення
  entity_id: # Розетки комп'ютерного класа 1/130
  - switch.plug_class_130_1
  - switch.plug_class_130_2

- alias: heating_off_after_study_131 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
  - platform: state # вибір платформи, зміна статусу
    entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
    to: 'off' # Э рух
  - platform: time_pattern # Регулярний запуск
    minutes: "/5" # Кожні 5 хвилин

```

```

condition: # Додаткова умова або декілька
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_1 # Розетка №1 (1/131)
  state: 'on'# Статус вимкнена
- condition: state # вибір платформи, зміна статусу
  entity_id: switch.plug_class_131_2 # Розетка №2 (1/131)
  state: 'on'# Статус вимкнена
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
  - mon
  - tue
  - wed
  - thu
  - fri
- condition: or # Одна з декількох умов, АБО
  conditions:
  - condition: time # вибір платформи, зміна часу або дня
    after: '06:00:00' # після
    before: '08:30:00' # перед
  - condition: time # вибір платформи, зміна часу або дня
    after: '20:25:00' # після
    before: '22:00:00' # перед
action: # Виконання дії
- service: switch.turn_off # Ввимкнення
  entity_id: # Розетки комп'ютерного класа 1/131
  - switch.plug_class_131_1
  - switch.plug_class_131_2

- alias: allarm_1_130 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
  entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
  to: 'on' # Є рух
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
  - mon
  - tue
  - wed
  - thu
  - fri
- condition: time # вибір платформи, зміна часу або дня
  after: '22:05:00' # після
  before: '05:55:00' # перед
action: # Виконання дії
- service: xiaomi_aqara.play_ringtone # Включення рингтону
  data:
  gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу
  ringtone_id: 7 # порядковий номер рингтону, атака з повітря

```



```

    ringtone_vol: 88 # гучність, у відсотках
- service: script.turn_on # Активування скрипта
  entity_id: script.send_message_1 # "Увага! Помічений рух в класі 1/130!"

- alias: allarm_1_131 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
    - platform: state # вибір платформи, зміна статусу
      entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
      to: 'on' # Є рух
    - platform: time_pattern # Регулярний запуск
      minutes: "/5" # Кожні 5 хвилин
  condition: # Додаткова умова або декілька
    - condition: time # вибір платформи, зміна часу або дня
      weekday: # дні
        - mon
        - tue
        - wed
        - thu
        - fri
    - condition: time # вибір платформи, зміна часу або дня
      after: '22:05:00' # після
      before: '05:55:00' # перед
  action: # Виконання дії
    - service: xiaomi_aqara.play_ringtone # Включення рингтону
      data:
        gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу
        ringtone_id: 7 # порядковий номер рингтону, атака з повітря
        ringtone_vol: 88 # гучність, у відсотках
    - service: script.turn_on # Активування скрипта
      entity_id: script.send_message_2 # "Увага! Помічений рух в класі 1/131!"

- alias: allarm_2_130 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
    - platform: state # вибір платформи, зміна статусу
      entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
      to: 'on' # Є рух
    - platform: time_pattern # Регулярний запуск
      minutes: "/5" # Кожні 5 хвилин
  condition: # Додаткова умова або декілька
    - condition: time # вибір платформи, зміна часу або дня
      weekday: # дні
        - sat
        - sun
    - condition: time # вибір платформи, зміна часу або дня
      after: '20:35:00' # після
      before: '06:55:00' # перед
  action: # Виконання дії
    - service: xiaomi_aqara.play_ringtone # Включення рингтону
      data:
        gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу

```

```

ringtone_id: 7 # порядковий номер рингтону, атака з повітря
ringtone_vol: 88 # гучність, у відсотках
- service: script.turn_on # Активування скрипта
entity_id: script.send_message_1 # "Увага! Помічений рух в класі 1/130!"

- alias: allarm_2_131 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
  entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
  to: 'on' # Є рух
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
  - sat
  - sun
- condition: time # вибір платформи, зміна часу або дня
  after: '20:35:00' # після
  before: '06:55:00' # перед
action: # Виконання дії
- service: xiaomi_aqara.play_ringtone # Включення рингтону
  data:
    gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу
    ringtone_id: 7 # порядковий номер рингтону, атака з повітря
    ringtone_vol: 88 # гучність, у відсотках
- service: script.turn_on # Активування скрипта
  entity_id: script.send_message_1 # "Увага! Помічений рух в класі 1/131!"

- alias: allarm_off_1_130 # Назва автоматизації
initial_state: true # Статус що показує робить автоматизація чи ні
trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
  entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
  to: 'off' # Немає руху
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
  - mon
  - tue
  - wed
  - thu
  - fri
- condition: time # вибір платформи, зміна часу або дня
  after: '22:05:00' # після
  before: '05:55:00' # перед
action: # Виконання дії
- service: xiaomi_aqara.stop_ringtone # Виключення рингтону
  data:

```

```

    gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу
- service: script.turn_onn # Активування скрипта
  entity_id: script.send_message_3 # "Увага! Руху в класі 1/130 більше немає, сирена
вимкнена!"

- alias: allarm_off_1_131 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
  entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
  to: 'off' # Немає руху
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
- mon
- tue
- wed
- thu
- fri
- condition: time # вибір платформи, зміна часу або дня
  after: '22:05:00' # після
  before: '05:55:00' # перед
action: # Виконання дії
- service: xiaomi_aqara.stop_ringtone # Виключення рингтону
  data:
    gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу
- service: script.turn_onn # Активування скрипта
  entity_id: script.send_message_4 # "Увага! Руху в класі 1/131 більше немає, сирена
вимкнена!"

- alias: allarm_2_off_130 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
- platform: state # вибір платформи, зміна статусу
  entity_id: binary_sensor.move_sensor_class_130 # Датчик руху (1/130)
  to: 'off' # Є рух
- platform: time_pattern # Регулярний запуск
  minutes: "/5" # Кожні 5 хвилин
condition: # Додаткова умова або декілька
- condition: time # вибір платформи, зміна часу або дня
  weekday: # дні
- sat
- sun
- condition: time # вибір платформи, зміна часу або дня
  after: '20:35:00' # після
  before: '06:55:00' # перед
action: # Виконання дії
- service: xiaomi_aqara.stop_ringtone # Включення рингтону
  data:
    gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу

```

```

- service: script.turn_on # Активування скрипта
  entity_id: script.send_message_3 # "Увага! Руху в класі 1/130 більше немає, сирена
вимкнена!"

- alias: allarm_2_off_131 # Назва автоматизації
  initial_state: true # Статус що показує робить автоматизація чи ні
  trigger: # умова або декілька умов
  - platform: state # вибір платформи, зміна статусу
    entity_id: binary_sensor.move_sensor_class_131 # Датчик руху (1/131)
    to: 'off' # Є рух
  - platform: time_pattern # Регулярний запуск
    minutes: "/5" # Кожні 5 хвилин
  condition: # Додаткова умова або декілька
  - condition: time # вибір платформи, зміна часу або дня
    weekday: # дні
    - sat
    - sun
  - condition: time # вибір платформи, зміна часу або дня
    after: '20:35:00' # після
    before: '06:55:00' # перед
  action: # Виконання дії
  - service: xiaomi_aqara.stop_ringtone # Включення рингтону
    data:
      gw_mac: 01:2a:51:10:1a:02 # MAC адреса хабу
  - service: script.turn_on # Активування скрипта
    entity_id: script.send_message_4 # "Увага! Руху в класі 1/131 більше немає, сирена
вимкнена!"

```

title: Дніпровська Політехніка - сервер
views:

- title: Головна
icon: mdi:home-assistant

cards:

- type: vertical-stack
cards:

- type: markdown
content: >
 ****Текущее состояние****

- type: entities
show_header_toggle: false
entities:
 - entity: sensor.time
 name: Текущее время
 icon: mdi:clock-outline
 - entity: sensor.date
 name: Дата
 icon: mdi:calendar-check-outline

- type: entities
show_header_toggle: false
entities:
 - entity: sensor.maria_db_size
 name: Объем БД
 icon: mdi:database

- type: vertical-stack
cards:

- type: markdown
content: >
 ****Графики****

- type: horizontal-stack
cards:

- type: gauge
name: Загрузка ОЗУ
unit: '%'
entity: sensor.memory_use_percent
severity:
 green: 0
 yellow: 60
 red: 85

- type: horizontal-stack

cards:

- type: gauge
name: Диск заполнен
unit: '%'
entity: sensor.disk_use_percent
severity:
 green: 0
 yellow: 60
 red: 85

- type: gauge
name: Свободное место
unit: 'Гб'
entity: sensor.disk_free
severity:
 red: 2
 yellow: 6
 green: 8

- type: history-graph
entities:
 - entity: sensor.maria_db_size
 name: Maria DB

- title: Погода
icon: mdi:weather-partly-snowy-rainy

cards:

- type: weather-forecast
entity: weather.home_assistant