

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

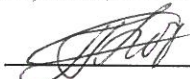
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект системи управління розумним
будинком на базі Raspberry Pi»

Виконав студент 4-го курсу, групи КІ-3
спеціальності 123 «Комп'ютерна інженерія»

 Р. В. Дядюшкін

Керівник к.т.н., доцент, доцент кафедри КІ та КБ

 І. В. Пулеко

Рецензент старший викладач кафедри КН

 Р. В. Петросян

Житомир – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ

СПЕЦІАЛЬНІСТЬ 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної
інженерії та кібербезпеки



Єфіменко А.А.

« 12 » 09 20 21 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу

Студента Дядюшкіна Романа Вікторовича

Тема роботи: Проект системи управління розумним будинком на базі Raspberry Pi

Затверджена Наказом університету від «12» квітня 2021 р. № 133\С

Термін здачі студентом закінченої роботи 20.06.21 р

Вихідні дані роботи (зазначається предмет і об'єкт дослідження) процес проектування,
розробки та моделювання системи управління розумним будинком; Raspberry Pi як основа
системи управління розумним будинком;

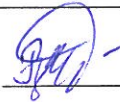
Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Пулеко І. В.	12.04.21	15.06.21
2	Пулеко І. В.	12.04.21	15.06.21
3	Пулеко І. В.	12.04.21	15.06.21

Керівник  Пулеко І. В.

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	12.04.21	виконано
2	Розробка плану роботи	15.04.21 – 17.04.21	виконано
3	Розробка розгорнутого плану	18.04.21 – 19.04.21	виконано
4	Подача чернетки керівнику для перевірки	25.04.21	виконано
5	Доопрацювання роботи та перевірка керівником	26.04.21 – 30.05.21	виконано
6	Затвердження графічних матеріалів у керівника	01.06.21 – 04.06.21	виконано
7	Проведення попереднього захисту	05.06.21	виконано
8	Отримання відгуку керівника	21.06.21	виконано
9	Допуск завідувача кафедри	22.06.21	виконано
10	Подання роботи на рецензування	22.06.21	виконано
11	Захист	23.06.21	

Студент  Дядюшкін Р. В.

Керівник  Пулеко І. В.

РЕЗЮМЕ

Кваліфікаційна робота бакалавра складається з вступу та трьох розділів, 74 сторінок тексту, з яких 62 сторінки основного тексту, 42 рисунки, 18 посилань на літературу.

Метою даної роботи є покращення рівня життя людини шляхом розробки проекту системи управління розумним будинком на базі одноплатного комп'ютера Raspberry Pi.

Методи дослідження: порівняльний аналіз, моделювання та оцінка літературних джерел за темою роботи; алгоритми роботи інформаційної системи управління розумним будинком.

Практична цінність полягає в розробці структурної схеми інформаційної системи управління будинком на Raspberry pi; розробці алгоритмів роботи підсистем інформаційної системи управління розумним будинком на Raspberry Pi; розробці програмного забезпечення інформаційної системи управління компонентами розумного будинку на Raspberry Pi; реалізація системи дозволить створити випуск сучасних аналогічних інформаційних систем управління розумним будинком, які будуть здійснювати управління розумним будинком.

КЛЮЧОВІ СЛОВА: РОЗУМНИЙ БУДИНОК, ІНТЕРНЕТ РЕЧЕЙ, RASPBERRY PI, ОДНОПЛАТНИЙ КОМП'ЮТЕР.

					KI.KP.B – 123 – 21 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект системи управління розумним будинком на базі Raspberry Pi	Літ.	Арк.	Аркушів
Розроб.		Дядюшкін Р. В.		22.06.21				
Керівник		Пулеко І.В.		22.06.21			4	74
Реценз		Петросян Р. В.		25.06.21		Житомирська політехніка, група KI-3		
Н. контр.								
Зав. каф.		Сфіменко А.А.		25.06.21				

RESUME

The bachelor's thesis consists of an introduction and three sections, 79 pages of text, of which 59 pages of the main text, 42 figures, 18 references.

The purpose of this work is to develop a smart home management information system on the Raspberry Pi, which allows you to control and manage devices in the house.

Research methods: comparative analysis, modeling and evaluation of literature sources on the topic of work; algorithms of operation of the smart home management information system.

The practical value lies in the development of a block diagram of the house management information system on the Raspberry pi; development of algorithms of work of subsystems of information system of management of the smart house on Raspberry Pi; development of software for information system management components of a smart home on the Raspberry Pi; implementation of the system will create a release of modern similar information systems for smart home management, which will manage the smart home.

Keywords: SMART HOME, INTERNET OF THINGS, RASPBERRY PI, SINGLE BOARD COMPUTER.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Актуальність теми.....	10
1.2. Аналіз існуючих систем	13
1.3. Аналіз типової архітектури IoT та обґрунтування вимог	17
1.4. Аналіз технічних характеристик Raspberry Pi.....	26
1.5. Постановка технічного завдання	30
РОЗДІЛ 2. РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ РОЗУМНИМ БУДИНКОМ НА БАЗІ RASPBERRY PI.....	33
2.1. Розробка структурної схеми системи управління розумним будинком на базі Raspberry Pi.....	33
2.2. Розробка функціональної схеми системи управління розумним будинком на базі Raspberry Pi.....	35
2.3. Розробка програмного забезпечення системи управління.....	37
РОЗДІЛ 3. ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ СИСТЕМИ УПРАВЛІННЯ РОЗУМНИМ БУДИНКОМ	43
3.1. Розробка моделі пристрою в Cisco Packet Tracer.....	43
3.2. Перевірка працездатності та ефективності по моделі пристрою управління.....	50
3.3. Рекомендації по застосуванню розробленого пристрою управління розумним будинком	58
ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	61
ДОДАТКИ.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

SBC (англ. Single Board Computer) – одноплатний комп'ютер

MCU (англ. Micro Controller Unit) - мікроконтролер

DHCP (англ. Dynamic Host Configuration Protocol) — протокол динамічної конфігурації вузла

DNS (англ. Domain Name System) – система доменних імен

IoT (англ. Internet of Things) – Інтернет речей

IP (англ. Internet Protocol) – протокол інтернету

ПК – персональний комп'ютер

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Сучасна людина потребує дуже високі вимоги до середовища існування:

- Естетичні (дизайн та стиль інтер'єру, ландшафу, краси та функціональності оточуючих предм;
- Кліматичні (комфортна температура та чисте повітря);
- Загальнопобутові (вода, газ, електроенергія, телебачення, інтернет, мобільний зв'язок, наявність кухонних машин та систем гігієни);
- Вимоги до безпеки та контролю (безпека житла та його мешканців);
- Вимоги до надійності складних систем (комп'ютери, пральні машини та інше)

В результаті інженерного оснащення квартир та домівок стрімко ускладнюються пристрої, які задіюються для формування середовища. Також стрімко збільшується їх кількість. Перекладати управління всіма системами на власника житла стає незручно, невигідно та іноді небезпечно. Комплексна система керування житлом «Розумний будинок» бере на себе всю рутинну роботу по вирішенню цієї задачі, залишаючи людині тільки приймання головних, базових рішень.

Ціллю курсової роботи є вивчення системи «Розумний будинок» та технологій її реалізації на основі Raspberry Pi.

Для вирішення поставленої цілі можна виділити певний список задач:

- Поняття системи «Розумний будинок» та принципи роботи системи;
- Технології системи розумний будинок
- Реалізація системи «Розумний будинок» на базі Raspberry Pi.

Метою роботи є розробка та моделювання системи управління розумним будинком на базі одноплатного комп'ютера Raspberry Pi.

Для досягнення цієї мети у роботі вирішені такі завдання:

- обґрунтовано актуальність вибраної теми кваліфікаційної роботи;

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

- проведено аналіз принципів побудови сучасних IoT систем;
- проведено аналіз існуючих рішень по реалізації систем управління розумним будинком від відомих фірм;
- розроблено структурну схему алгоритмів функціонування системи управління розумного будинку на Raspberry Pi;
- виконано розробку функціональної схеми системи управління розумного будинку на Raspberry Pi;
- проведено імітаційне моделювання системи управління розумного будинку на Raspberry Pi у програмному середовищі Cisco Packet Tracer;
- досліджено працездатність та розроблено рекомендації щодо особливостей проектування та застосування системи управління розумного будинку на Raspberry Pi;

Об’єктом роботи є система управління розумного будинку на Raspberry Pi;

Предметом роботи є процес розробки та моделювання системи управління розумного будинку на Raspberry Pi;

Практична значимість роботи полягає в розробці алгоритмів, які можна застосовувати для розробки системи управління розумного будинку на Raspberry Pi;

У роботі використані методи аналізу, математичного та імітаційного моделювання.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Актуальність теми

Одним з найбільш перспективних напрямків розвитку інформаційних технологій є IoT (Internet of Things), який набирає популярності у багатьох сферах людської діяльності. Від примітивних задач у сільському господарстві та промисловості до застосування у медицині.

Спектр застосування настільки широкий, що ідей вистачить не тільки на промислові процеси, але й на звичайних споживачів. Споживчі пристрої були одними з перших категорій предметів, які були підключені до інтернету. На сьогоднішній день мільйони домів оснащені лампочками Hue, термостатами Nest та віртуальним голосовим помічником Alexa. Крім того, люди користуються браслетами Fitbit та багатьма іншими портативними пристроями.

Ще у далекому 1991 році дослідник лабораторії Xerox PARC Марк Вейзер висунув концепцію майбутнього світу, «багато і непомітно насиченого сенсорами, дисплеями і обчислювальними елементами, з'єднаними в єдину мережу і являючими невід'ємні елементи предметів побуту». Фізична можливість здійснення цієї концепції з'явилася до кінця 2000-х років у міру тотального поширення дешевих і мініатюрних обчислювальних мобільних пристроїв, бездротових мереж і супутникової навігації. [1]

Повсюдний (всюдисущий, всепроникаючий, тотальний) комп'ютинг, що фігурує в спеціальній літературі під термінами «ubiquitous computing» і «Pervasive computing», попросту означає створення всюдисущих інтелектуальних інформаційних систем, що допомагають у щоденній людській рутині - вдома, в офісі, в лікарні, на роботі, в дорозі. Тотальний комп'ютинг ставить на чільне місце кінцевого користувача, який повинен отримувати обчислювальний обслуговування безперервно, 24 години на добу, 7 днів на тиждень, причому обслуговування самого різного роду - від наукових

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

обчислень до управління кухонними агрегатами. Так, наприклад, система персональної допомоги (Personal Assistance System, PAS) допомагає людям похилого віку в самообслуговуванні з використанням бездротової мережі, що об'єднує RFID-рідери, медтехніку з Bluetooth-інтерфейсом, програмне забезпечення та апаратуру, яка відслідковує переміщення людини в житлі, датчики падіння, системи безпеки і т.п.

На основі Інтернету речей можуть бути реалізовані всілякі «розумні» (smart) додатки в різних сферах діяльності і життя людини:

- «Розумна планета» - людина зможе буквально «тримати руку на пульсі» планети: своєчасно реагувати на упущення в плануванні господарств, забруднення та інші екологічні проблеми, а значить, ефективно розпоряджатися невідновлюваних ресурсами.
- «Розумне місто» - міська інфраструктура і супутні муніципальні послуги, такі як освіта, охорона здоров'я, громадська безпека, ЖКГ, стануть більш пов'язаними і ефективними.
- «Розумний будинок» - система буде розпізнавати конкретні ситуації, що відбуваються в будинку, і реагувати на них відповідним чином, що забезпечить мешканцям безпеку, комфорт і ресурсозбереження.
- «Розумна енергетика» - буде забезпечена надійна і якісна передача електричної енергії від джерела до приймача в потрібний час і в необхідній кількості.
- «Розумний транспорт» - переміщення пасажирів з однієї точки простору в іншу стане зручніше, швидше і безпечніше.
- «Розумна медицина» - лікарі і пацієнти зможуть отримати віддалений доступ до дорогого медичного обладнання або до електронної історії хвороби в будь-якому місці, буде реалізована система віддаленого моніторингу здоров'я, автоматизована видача лікарських препаратів хворим і багато іншого.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

Можливості Інтернету речей в області генерування, збору, передачі, аналізу та розподілу величезного обсягу даних в світовому масштабі дозволять людству, в кінцевому рахунку, отримати нові знання, які необхідні йому не тільки для виживання, але і для нинішнього добробуту протягом багатьох століть. Підтвердження цьому - включення Інтернету речей в перелік проривних технологій в США і в число семи формується національних стратегічних галузей промисловості в Китаї.

Єдині стандарти тільки зароджуються, але масштабні проекти в даному напрямку - свого роду «інтранет речей» - енергійно розвиваються вже зараз. Так, американське агентство NASA за підтримки компанії Cisco створює систему глобального збору даних про Землю «Шкіра планети» (Planetary skin). Про «розумні» будинки напевно багато чули, в Японії вже не рідкість «розумні» заводи, а в США в рамках національної ініціативи оцифровки мегаполісів Connected Urban Development «розумнішають» і міста. [1]

У різних країнах існують конкретні програми і плани практичного впровадження інтернету речей. Так, Євросоюз розвиває IoT за спеціальною програмою, що включає 14 напрямків. Згідно китайської державною програмою до 2015 року планується реалізувати 149 проектів. Не менш активно ведуться розробки в Англії, Австралії, Японії, Південної Кореї та інших країнах. [1]

До прикладів застосування IoT для звичайних споживачів можна віднести портативні пристрої (трекери здоров'я і руху, розумний одяг та аксесуари). Достатньо популярними є пристрої для відстеження місцезнаходження домашніх тварин або розумні двері для собаки. Проте найбільш популярним прикладом буде застосування розумних пристроїв для дому, таких як автоматизовані системи поливу, гаражні двері, замки, термостати та системи охорони. Все це можна віднести до сфери Розумний будинок.

Розумний будинок – це особливо пріоритетна сфера Інтернету речей.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

Насамперед це можливість комплексно вирішити питання автоматизації інженерних систем, вивільнення часу, яке раніше витрачалося на рутинні побутові задачі – все це серйозно підвищує якість життя, робить його більш комфортним.

Не диво, що з кожним роком по всьому світу інтерес до інтелектуальних систем тільки зростає – актуальність Розумного будинку давно не потребує підтверджень.

1.2. Аналіз існуючих систем

Z-Wave Fibaro Home Center 2 (FIB_HOMEC2) – це пристрій, призначений для автоматизації управління Розумним будинком. Fibaro Home Center 2 взаємодіє з електронними пристроями за допомогою провідних та безпроводних технологій. Одночасно може контролювати до 230 пристроїв. Включає в себе інтеграцію з голосовими асистентами Google Assistant та Amazon Alexa. [7]



Рисунок 1.1 – Модуль управління розумним будинком Z-Wave Fibaro Home Center 2. [7]

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Технічні характеристики:

- Процесор – Intel Atom, 1.6 ГГц;
- Диск – 4 Гб MLC Recovery Disk;
- ОЗУ – 1 Гб та 2 Гб SLC жорсткий диск;
- Протокол зв'язку – Z-Wave;
- Дальність передачі безпроводного сигналу – 50 м;
- Кількість USB портів – 4;
- Радіочастота передачі – 868.42 МГц
- Додаткові можливості – http команди, VoIP, управління голосом.

Wink Hub 2 – контролер управління розумним будинком. Особливістю даного пристрою є націленість розробників на сумісність з усіма відомими виробниками у даній сфері та масимально інтуїтивне використання.



Рисунок 1.2 – Модуль управління розумним будинком Wink Hub 2. [8]

Технічні характеристики:

- Підтримка протоколів – Bluetooth LE, Kiddle, Clear Connect, Z-Wave, ZigBee;

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Управління голосом – Google Assistant та Amazon Alexa;
- Кількість пристроїв – 400+;
- Підключення до інтернету – WiFi (2.4 та 5 ГГц), Ethernet;
- ОЗУ – 512 Мб
- Розміри – 19x19x3.2 см;

Xiaomi Mi Smart Home Gateway 2 – центральний контролер для розумного будинку.



Рисунок 1.3 – Модуль управління розумним будинком Xiaomi Mi Smart Home Gateway 2. [9]

Технічні характеристики:

- Стандарти зв'язку – WiFi (2.4 ГГц), ZigBee;
- Підтримка сценаріїв – є;
- Робоча напруга – 100-240В, 50/60 Гц;
- Клас захисту IP – IP20;
- Розміри – 80мм x 30мм;

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

Zipato ZipaTile – модуль управління розумним будинком.



Рисунок 1.4 – Модуль управління розумним будинком Zipato ZipaTile.

[18]

Технічні характеристики:

- Стандарти зв'язку – WiFi, Z-Wave
- Підтримка сценаріїв – є;
- Радіочастота передачі – 433 МГц;
- Сумісність з моделями – Z-Wave, Z-Wave Plus, ZigBee;
- Максимальна кількість пристроїв – 200;
- Розміри – 205мм x 205мм x 14мм;
- Діапазон робочих температур – від 0°C до +40°C
- Робоча вологість - $\leq 85\%$;

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

1.3. Аналіз типової архітектури IoT та обґрунтування вимог

Робота Інтернету речей починається з найпростіших датчиків, розміщених у найвіддаленіших куточках світу, і перетворює аналоговий фізичний вплив у цифрові сигнали. Потім перетворені дані подорожують у вигляді провідних та безпроводних сигналів, які формуються різними протоколами та у результаті потрапляють до інтернету. Звідти пакети даних передаються до хмари або в центри обробки даних.

Питаннями стандартизації та практичного впровадження окремих складових Інтернету речей (M2M, RFID та інші) займаються різні міжнародні організації, альянси виробників та операторів, партнерські проекти. В цілому для Інтернету речей, як відносно нового напрямку розвитку інфокомунікацій, в наш час визначені лише самі загальні концептуальні та архітектурні рішення.

В рамках діяльності сектору стандартизації телекомунікацій Міжнародного союзу електрозв'язку (МСЕ) існує три глобальних ініціативи GSI (Global Standards Initiative). Під глобальною ініціативою розуміється комплекс робіт, які виконуються паралельно різними дослідницькими комісіями МСЕ у відповідності до скоординованого плану робіт. Однією з таких ініціатив є стандартизація Інтернету речей – IoT-GSI (Global Standards Initiative on Internet of Things). В рамках серії МСЕ Y.2xxx уже затверджені перші рекомендації, присвячені спеціально Інтернету речей. [2]

В рекомендації Y.2060 приведена еталонна модель IoT, яка включає чотири базові горизонтальні рівні:

- Рівень додатків IoT;
- Рівень підтримки додатків та послуг;
- Мережевий рівень;
- Рівень пристроїв.

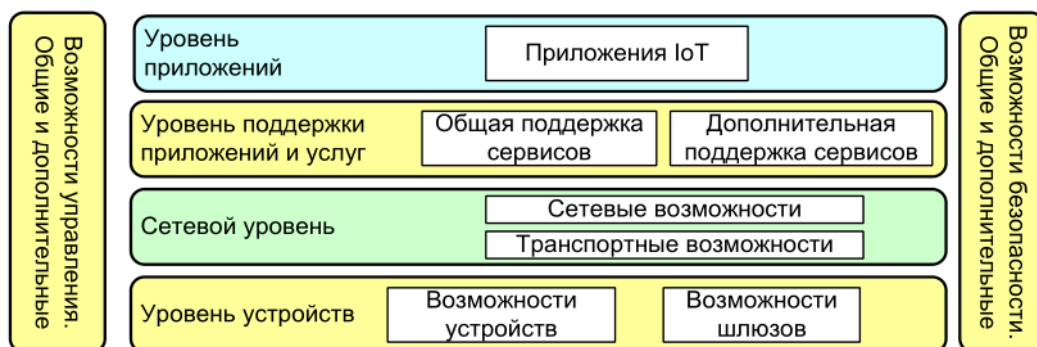


Рисунок 1.5 - Еталонна модель IoT згідно рекомендації МСЕ Y.2060.

[2]

Рівень додатків IoT в Рекомендації Y.2060 детально не розглядається.

Рівень підтримки додатків та послуг включає загальні можливості для різних об'єктів IoT по обробці та зберіганню даних. Мережевий рівень включає мережеві можливості (функції управління ресурсами мережі доступу та транспортної мережі, управління мобільністю, функції авторизації, аутентифікації та розрахунків) та транспортні можливості (забезпечення зв'язності мережі для передачі інформації додатків та послуг IoT. Рівень пристроїв включає можливості пристрою та можливості шлюза. Можливості пристрою допускають прямий обмін з мережею зв'язку, обмін через шлюз а також тимчасову зупинку та відновлення роботи пристроїв для енергозбереження. Можливості шлюза допускають підтримку багатьох інтерфейсів для пристрою (ZigBee, Bluetooth, WiFi та інші) та для мереж доступу/транспортних мереж(2G/3G, LTE, DSL та інші).

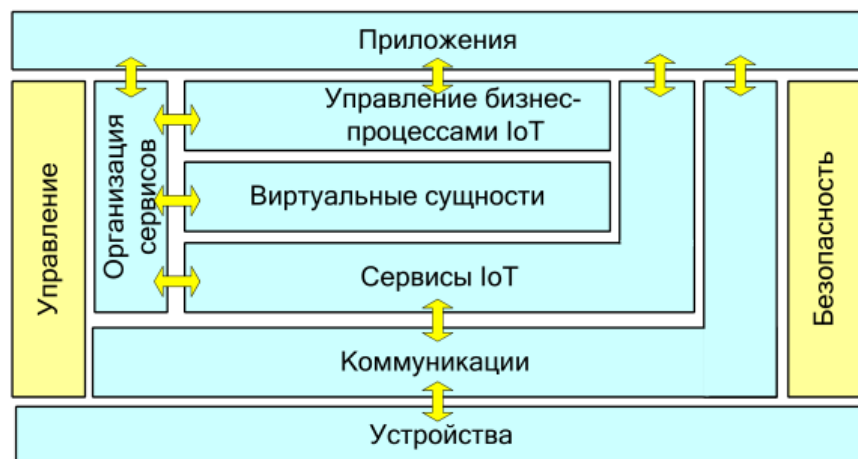


Рисунок 1.6 - Функціональна модель архітектури IoT-A. [2]

Основною ціллю проекту Європейського інтеграційного проекту IoT-A (Internet of Things – Architecture), учасниками якого є різні компанії, являється розробка еталонної архітектурної моделі Інтернету речей з описом основних складових компонентів, яка б дозволяла інтегрувати різноманітні технології IoT у єдину взаємопов'язану архітектуру.

Функціональна модель IoT-A (Рисунок 1.6) трохи відрізняється від моделі MCE (Рисунок 1.5), хоча вона теж є ієрархічною, вона складається з семи горизонтальних рівнів, які доповнюються двома вертикальними (управління та безпека), які беруть участь у всіх процесах.

IoT складається з набору різних інфокомунікаційних технологій, які забезпечують функціонування Інтернету речей, та його архітектура показує, як ці технології пов'язані між собою. Архітектура включає чотири функціональні рівні (Рисунок 1.7).

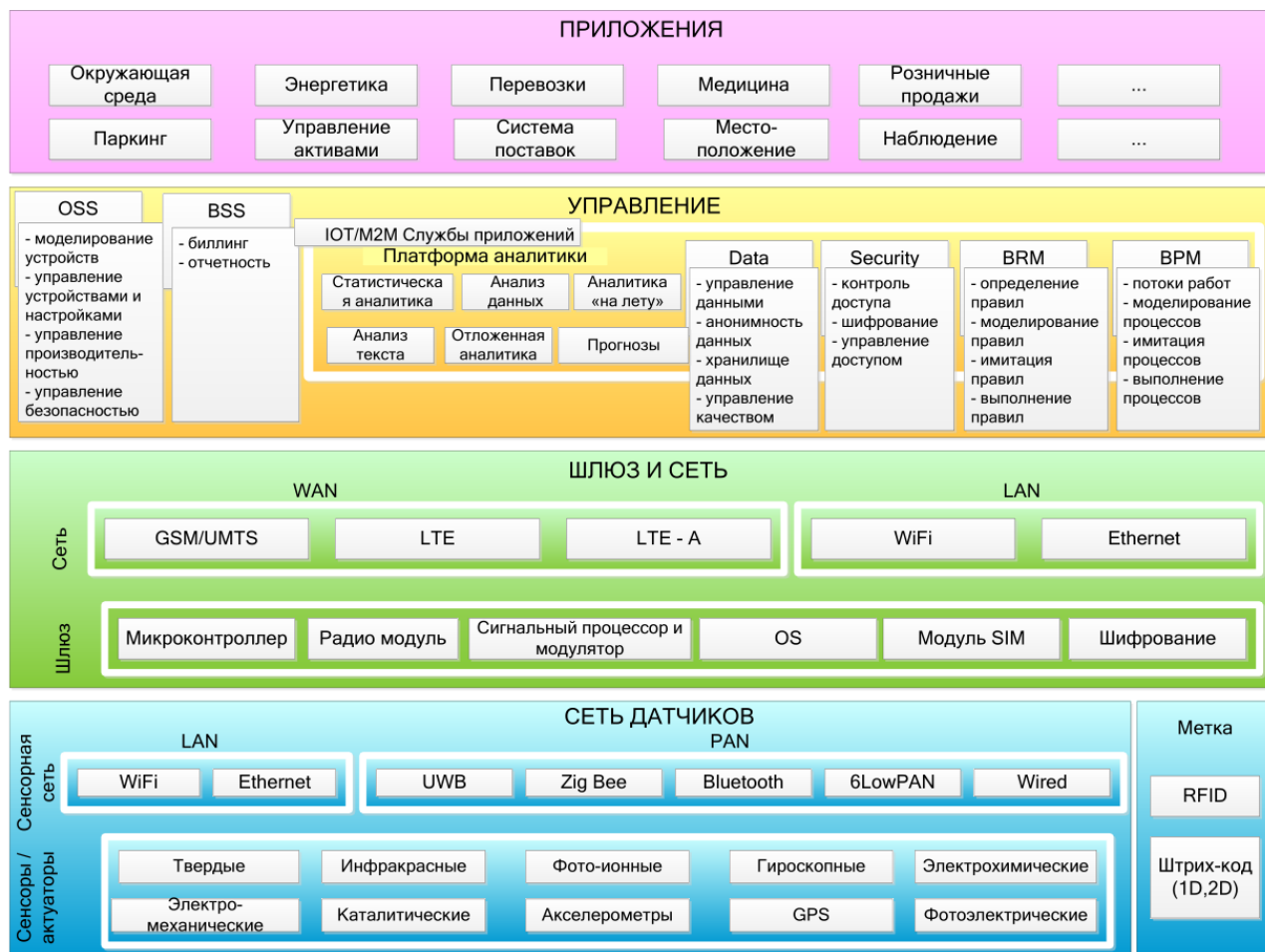


Рисунок 1.7 - Архитектура IoT. [2]

1. Рівень сенсорів та сенсорних мереж

Найважливіший рівень архітектури IoT складається з «розумних» об'єктів, інтегрованих з сенсорами(датчиками). Сенсори реалізують з'єднання фізичного та віртуального(цифрового) світів, забезпечуючи збір та обробку інформації у реальному часі. Існують різні типи сенсорів для відповідних цілей, наприклад, для вимірювання температури, тиску, швидкості руху та інші. Сенсор може замірювати фізичні параметри контрольованого об'єкта та перетворювати їх в сигнал, який може бути прийнятий відповідним пристроєм. Також сенсори класифікуються у відповідності до призначення, наприклад, сенсори навколишнього середовища, сенсори для тіла, сенсори побутової техніки, сенсори для транспортної техніки і т.д.

Більшість сенсорів потребують з'єднання з агрегатором сенсорів(шлюзом), які можуть бути реалізовані з використанням локальної обчислювальної мережі (LAN, Local Area Network), таких як Ethernet та Wi-Fi або персональної мережі (PAN, Personal Area Network), таких як ZigBee, Bluetooth.

1. Рівень шлюзів та мереж

Великий обсяг даних, що створюються на першому рівні IoT численними мініатюрними сенсорами, вимагає надійної та високопродуктивної провідний або бездротової мережевої інфраструктури в якості транспортного середовища. існуючі мережі зв'язку, що використовують різні протоколи, можуть бути використані для підтримки межмашинного комунікацій M2M і їх додатків. Для реалізації широкого спектру послуг і додатків в IoT необхідно забезпечити спільну роботу безлічі мереж різних технологій і протоколів доступу в гетерогенної конфігурації. Ці мережі повинні забезпечувати необхідні значення якості передачі інформації, і перш за все по затримці, пропускну здатності і безпеки. Даний рівень складається з конвергентної мережевої інфраструктури, яка створюється шляхом інтеграції різнорідних мереж в єдину мережеву платформу. Конвергентний абстрактний мережевий рівень в IoT дозволяє через відповідні шлюзи декільком користувачам використовувати ресурси в одній мережі незалежно і спільно без шкоди для конфіденційності, безпеки і продуктивності.

2. Сервісний рівень

Сервісний рівень містить набір інформаційних послуг, покликаних автоматизувати технологічні і бізнес операції в IoT: підтримки операційної і бізнес діяльності (OSS / BSS, Operation Support System / Business Support System), різної аналітичної обробки інформації (статистичної, інтелектуального аналізу даних і текстів, прогностична аналітика і ін.),

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

зберігання даних, забезпечення інформаційної безпеки, управління бізнес-правилами (BRM, Business Rule Management), управління бізнес-процесами (BPM, Business Process Management) і ін.

3. Рівень додатка

На четвертому рівні архітектури IoT існують різні типи додатків для відповідних промислових секторів і сфер діяльності (енергетика, транспорт, торгівля, медицина, освіта та ін.). Додатки можуть бути «вертикальними», коли вони є специфічними для конкретної галузі промисловості, а також

«Горизонтальними», (наприклад, управління автопарком, відстеження активів і ін.), Які можуть використовуватися в різних секторах економіки. Конкретні IoT додатки більш детально розглянуті в розділі 9.

У сфері Інтернету речей інженери працюють над вирішенням багатьох проблем, таких як: трансформація фізичних величин у цифрові сигнали, системи енергозбереження та накопичування енергії, управління великою кількістю пристроїв на великих відстанях та інші.

Сферу Інтернету речей можна розділити на окремі сегменти:

- Датчики: встроєні системи, операційні системи реального часу, джерела безперебійного живлення, мікроелектромеханічні системи;
- Системи зв'язку між датчиками: зона покриття безпроводних персональних мереж становить від 0 до 100 метрів. Для обміну даними між датчиками застосовуються низькочастотні малопотужні інформаційні канали, які часто побудовані не на протоколі IP;
- Локальні обчислювальні мережі: зазвичай це системи обміну даними на основі протоколу IP, наприклад, 802.11 Wi-Fi мережа для швидкого радіозв'язку;
- Маршрутизатори, шлюзи: постачальники вбудованих систем, виробники модулів та інші;
- Глобальна обчислювальна мережа: оператори мобільного зв'язку,

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

оператори супутникового зв'язку, оператори малопотужних глобальних мереж (Low-Power Wide-Area Network, LPWAN);

- Хмара: інфраструктура в якості постачальника послуг, платформа в якості постачальника послуг, постачальники послуг потокової та пакетної обробки даних тощо.

- Аналіз даних: великі масиви інформації передаються до хмари. Робота з великими об'ємами даних та отримання з них користі – це задача, яка потребує комплексної обробки подій, аналітики та засобів Machine Learning;

- Безпека: при зведенні всіх елементів архітектури в цілісну структуру виникають питання стосовно безпеки. Безпека торкається кожного компоненту: від датчиків фізичних величин до цифрового апаратного забезпечення, систем радіозв'язку та самих протоколів передачі даних. На кожному рівні необхідно забезпечити безпеку, достовірність та цілісність.

Архітектура IoT охоплює багато технологій. Складність та багатогранність інтернету речей пов'язана з тим, що ця технологія більше комплексна, ніж традиційні цифрові технології. Кількість можливих рішень вражає уяву. На даний момент існує більше ніж 1.5 мільйони різноманітних комбінацій архітектурних рішень.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

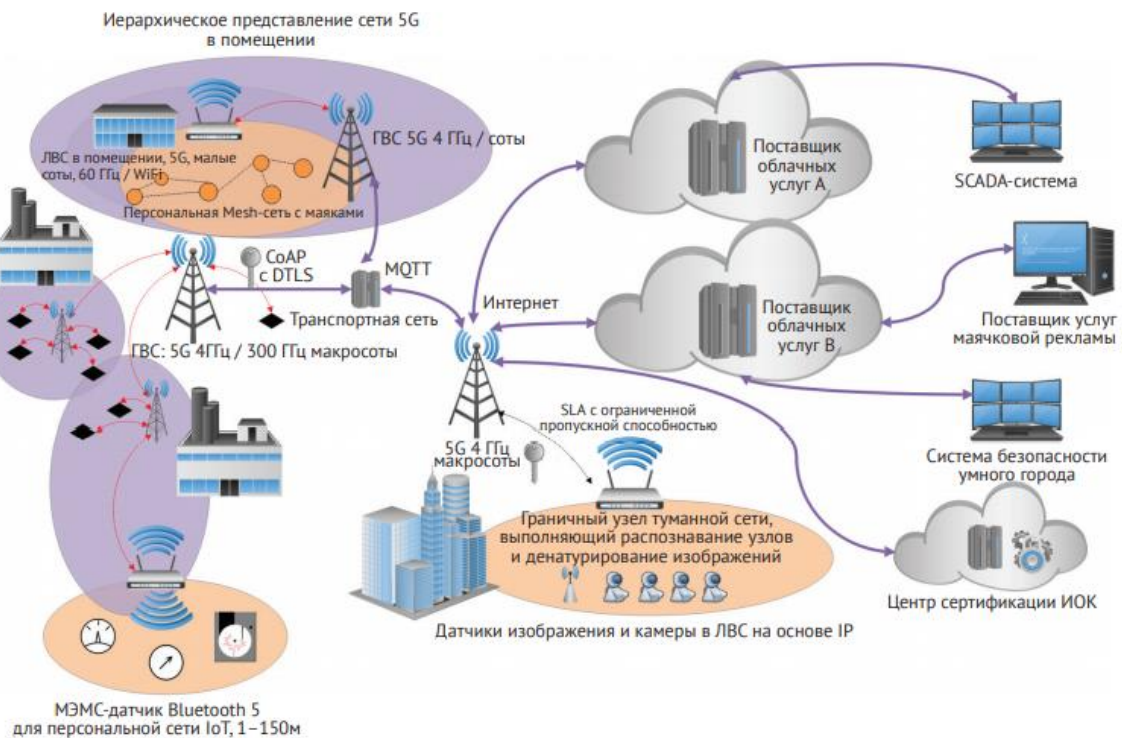


Рисунок 1.8 – Повний спектр різних варіантів на всій рівнях IoT-архітектури. [1]

Рівні реалізації архітектури IoT:

1. Датчики та живлення

Інтернет речей починається та закінчується одною подією: простий рух, зміна температури та багато інших. Навідміну від багатьох існуючих IT-пристроїв, інтернет речей у більшості випадків пов'язаний з фізичними подіями. Він реагує на якийсь фактор реального фізичного світу. Іноді при цьому один-єдиний датчик може згенерувати величезний об'єм даних. В інших випадках достатньо одного біта даних для передачі життєво важливих відомостей про стан здоров'я пацієнта.

Датчики постійно еволюціонують. Розвиток інформаційних технологій дозволяє зменшувати їх габарити до нанометрових розмірів та зменшити вартість. Нові датчики потребують менше енергії для функціонування, ніж раніше. Також еволюціонують технології забезпечення пристроїв живленням.

2. Передача даних

Інтернет речей не може існувати без надійних технологій передачі даних. Базова опора інтернету речей – це не датчики, і не програмні додатки, а можливість встановити з'єднання між усіма компонентами.

3. Інтернет-маршрутизація та протоколи

Для передачі даних від датчиків до інтернет-простору необхідно дві технології: маршрутизатор у якості шлюза та опорні інтернет протоколи, які забезпечують ефективність обміну даними. Маршрутизація відіграє важливу роль в плані безпеки передачі даних. На рівні маршрутизації часто застосовують віртуальні приватні мережі, управління та спостереження існуючих mesh-мереж.

4. Туманні та граничні обчислення, аналітика та машинне навчання.

На рівні туманних обчислень необхідно вирішити, що робити з потоком даних, які приходять з граничного вузла до хмарного сервісу. Дані, які були отримані методом перетворення аналогової фізичної дії в цифровий сигнал, можуть мати великі розміри. У таких випадках задіюють аналітику та процесори IoT-системи. В деяких ситуаціях граничний маршрутизатор реагує лише на аномальні скачки фізичних величин. В іншому випадку може бути велика кількість даних в режимі реального часу передається до хмари та потребує високої швидкості обробки, наприклад, для прогнозування.

5. Загроза та безпека в інтернеті речей.

IoT-системи не завжди обмежені безпечним простором дому або підприємства. Інтернет речей це велика єдина мішень для будь-яких видів хакерських атак. При проектуванні IoT-систем важливо розуміти які існують вразливості в IoT-системах та які існують рішення для їх попередження.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

1.4. Аналіз технічних характеристик Raspberry Pi

Raspberry Pi - третя за продажем комп'ютерна марка у світі. Raspberry Pi - це комп'ютер розміром із кредитну картку, який підключається до телевізора або дисплея, а також клавіатури та миші. Ви можете використовувати його для вивчення кодування та побудови проектів з електроніки, а також для багатьох речей, які робить ваш настільний ПК, таких як електронні таблиці, обробка текстів, перегляд Інтернету та гра в ігри. Він також відтворює відео високої чіткості. Raspberry Pi використовується дорослими та дітьми у всьому світі для вивчення програмування та цифрових технологій.

Raspberry Pi – це мікрокомп'ютер, який несподівано для його творців набув доволі великої популярності. Випускається в декількох версіях. Незважаючи на маленькі розміри (вміщується на долоні) – це справжній мікрокомп'ютер, на який встановлюється операційна система.

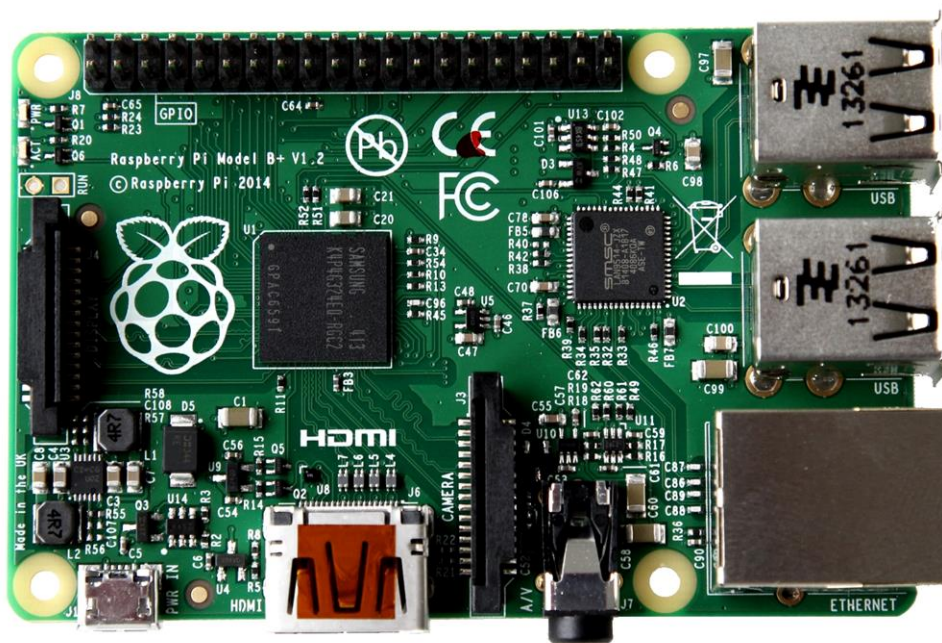


Рисунок 1.9 – Зображення Raspberry Pi 3 Model B+. [17]

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Пілюко І.В.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

Комп'ютери Raspberry Pi можна придбати окремо або у вигляді повних комплектів, що містять усе необхідне для початку, включаючи мишу, кабель HDMI, SD-карту, блок живлення.

Raspberry Pi може працювати під Windows CE, Debian, Gentoo, Arch, RISC OS, AROS або FreeBSD. Існує навіть Android для Raspberry. Також можна зустріти операційні системи, оптимізовані спеціально для даного мікрокомп'ютера, такі як Raspbian (на основі Debian) та FedoraRemix(на основі Fedora).

Крім стандартних для комп'ютера інтерфейсів, Raspberry Pi має піни для підключення зовнішніх пристроїв, що значно розширює область застосування одноплатного комп'ютера в системах автоматизації та при проектуванні інших цікавих систем: від розумного будинку та роботизованих систем до автопілотів.

Варто зазначити, що в першу чергу Raspberry Pi – це комп'ютер, а тільки потім плата, до якої можна підключити сенсори або зовнішні пристрої.

Технічні характеристики Raspberry Pi 3 Model B+:

- Процесор: SoC Broadcom серії BCM2837B0 (чотириядерний ARM Cortex-A53 1.4 ГГц з 64-бітною архітектурою);
- Оперативна пам'ять: 1 Гб;
- VideoCore IV 3D;
- Бездротові можливості: Wi-Fi(802.11ac/n), Bluetooth 4.2;
- Мережеві можливості: Ethernet 1000Base-T;
- 40-контактний роз'єм, що підключений до портів вводу та виводу;
- USB порти: 4 × USB 2.0;
- Відеовихід: HDMI;
- Аудіовихід: гніздо 3,5 мм з підтримкою композитного відеовиводу;
- Інтерфейс для CSI камери;

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

- DSI-конектор для LCD дисплея;
- Слот для карт microSD;
- Габарити: $85 \times 56 \times 17$ мм;

Потужною особливістю Raspberry Pi є ряд пінів GPIO (general-purpose input/output) вздовж верхнього краю плати (Рисунок 1.10). 40-контактний заголовок GPIO знаходиться на всіх поточних платах Raspberry Pi (відсутні на Pi Zero та Pi Zero W). До моделі Pi 1 Model B + (2014) дошки включали коротший 26-контактний роз'єм.

Будь-який із пінів GPIO може бути позначений (у програмному забезпеченні) як вхідний чи вихідний пін та використаний для широкого спектру задач.

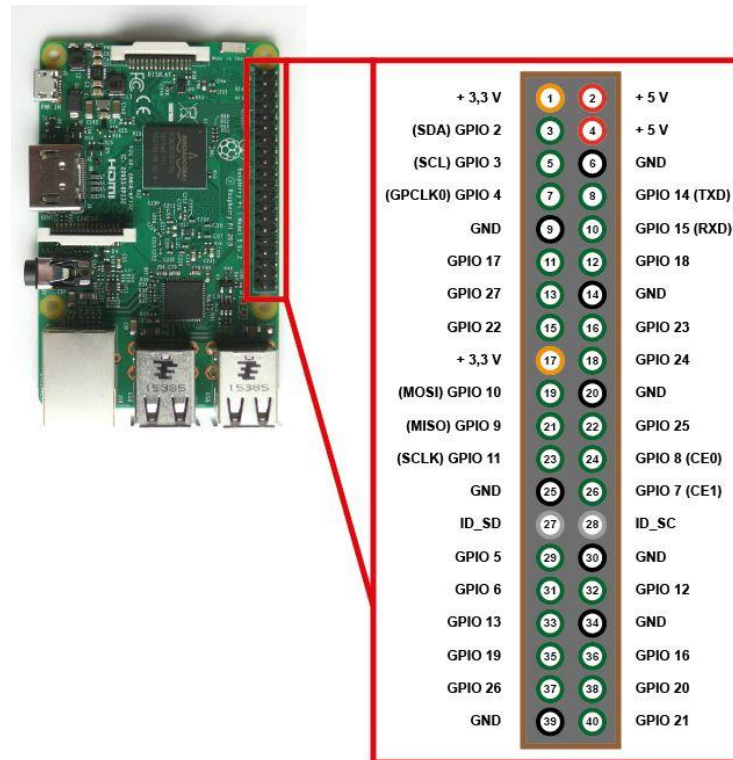


Рисунок 1.10 – Нумерація GPIO пінів. [10]

На платі присутні два піни живлення 5 В і два піни живлення 3.3 В, а також кілька заземлених виводів (0 В), які не можна налаштувати. Решта пінів - це піни загального призначення 3.3 В, тобто для виходів встановлено значення 3.3 В, а для входів - толерантні до 3.3 В.

Пін GPIO, призначений як вихідний, можна встановити на високий рівень напруги (3.3 В) або низький (0 В).

Вивід GPIO, призначений як вхідний, може бути прочитаний як високий (3.3 В) або низький (0 В). Це полегшується завдяки використанню внутрішніх підтягуючих або висувних резисторів. Виводи GPIO2 та GPIO3 мають фіксовані підтягуючі резистори, але для інших пінів це можна налаштувати в програмному забезпеченні.

Окрім простих пристроїв введення та виведення, піни GPIO можуть використовуватися з різними альтернативними функціями, деякі доступні на всіх пінах, інші лише на конкретних. Розглянемо основні з них:

PWM (pulse-width modulation, широтно-імпульсна модуляція) - це імпульсний сигнал постійної частоти і змінної шпаруватості, тобто відношення тривалості імпульсу до періоду його проходження. За допомогою завдання шпаруватості (тривалості імпульсів) можна міняти середнє значення напруги на виході ШІМ. Підтримується програмно на всіх пінах. Апаратно підтримують піни GPIO12, GPIO13, GPIO18, GPIO19.

SPI (Serial Peripheral Interface Bus) - виводи, які можуть бути використані для підключення і спілкування з пристроями SPI. Майже те ж, що і I2C, але використовується інший протокол.

SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)

SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

I2C (Inter Integrated Circuit) - створювався як простий інтерфейс з мінімальною кількістю ліній зв'язку. Він використовує всього дві лінії для зв'язку ведучого пристрою з веденим: двонаправлену лінію даних SDA (Serial Data) на піні GPIO2 та лінію тактування SCL (Serial CLock) на піні GPIO3 для синхронізації прийому і передачі даних.

Serial - двонаправлений послідовний інтерфейс, призначений для

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

обміну байтовою інформацією. У ролі передавача виступає TX (GPIO14) та у ролі приймача - RX (GPIO15).

1.5. Постановка технічного завдання

Завданням на кваліфікаційну роботу є проектування розумного будинку та розробка інформаційної системи керування для нього. Також необхідно змоделювати проект в програмному середовищі Cisco Packet Tracer.

Підставою для виконання роботи є завдання на кваліфікаційну роботу.

1.5.1. Умови експлуатації та параметри навколишнього середовища

Умови експлуатації системи в будинку:

Відносна вологість: 45-55%;

Температура повітря у приміщенні: 18 - 23 °С;

Точка роси: не більше 21 °С;

Швидкість зміни: не більше 5 °С; [1]

1.5.2. Вимоги до технічних характеристик

Головний функціонал системи управління розумним будинком:

- Контроль та підтримання певної температури у будинку;
- Контроль та підтримання рівня вологості у приміщенні;
- Система протипожежного захисту;
- Управління освітленням будинку;
- Керування дренажної системою.
- Автоматизація та централізоване управління освітленням з врахуванням пори доби та присутності людей у приміщенні. Дуже актуально для

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

будинків з дітьми, людьми похилого віку або для людей з певними фізичними вадами здоров'я;

- Доступна ціна;
- Система повинна бути масштабованою, дозволяти можливість розширення функціоналу шляхом додавання до системи нового устаткування;
- Інтуїтивно зрозумілий для користувачів інтерфейс;
- Можливість підлаштувати систему під конкретні потреби користувача(як інтерфейс так і функціонал);
- Контроль за протіканням води/газу;
- Окрім зручності система повинна орієнтуватися також на енергозбереження. Розумний дім — це доволі енергоефективна система (витрати електроенергії на 400 м² будуть сягати 55 W);
- Система повинна бути надійною, мати власне резервування, здатність відновлювати свою роботу самостійно.

Висновки до Розділу 1

У першому розділі було обгрунтовано актуальність теми дипломного проекту. Розглянуто можливості та технічні характеристики існуючих рішень для реалізації системи контролю розумним будинком. Ознайомлення з існуючими системами приводить до висновку, що більшість існуючих рішень мають закрите програмне забезпечення і значні обмеження у сумісності пристроїв.

До переваг можна віднести простоту в експлуатації, можливість підключення до 200 пристроїв одночасно, інколи наявні вбудовані до модуля управління датчики.

Серед недоліків можна відмітити вартість гоового рішення та обмежену сумісність пристроїв з деякими системами.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

Проведено аналіз архітектури IoT систем, функції та особливості кожного з рівнів реалізації. Застосування виробниками власних закритих розробок значно ускладнює вирішення проблеми сумісності через неузгодженість реалізації технологій комунікації між пристроями.

Також було розглянуто особливості роботи мікрокомп'ютера Raspberry Pi та його технічні характеристики. Одноплатний комп'ютер крім своїх розмірів має дуже привабливу ціну у порівнянні з достатньою для повноцінного виконання мінімальних задач потужністю.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

РОЗДІЛ 2. РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ РОЗУМНИМ БУДИНКОМ НА БАЗІ RASPBERRY PI

2.1. Розробка структурної схеми системи управління розумним будинком на базі Raspberry Pi

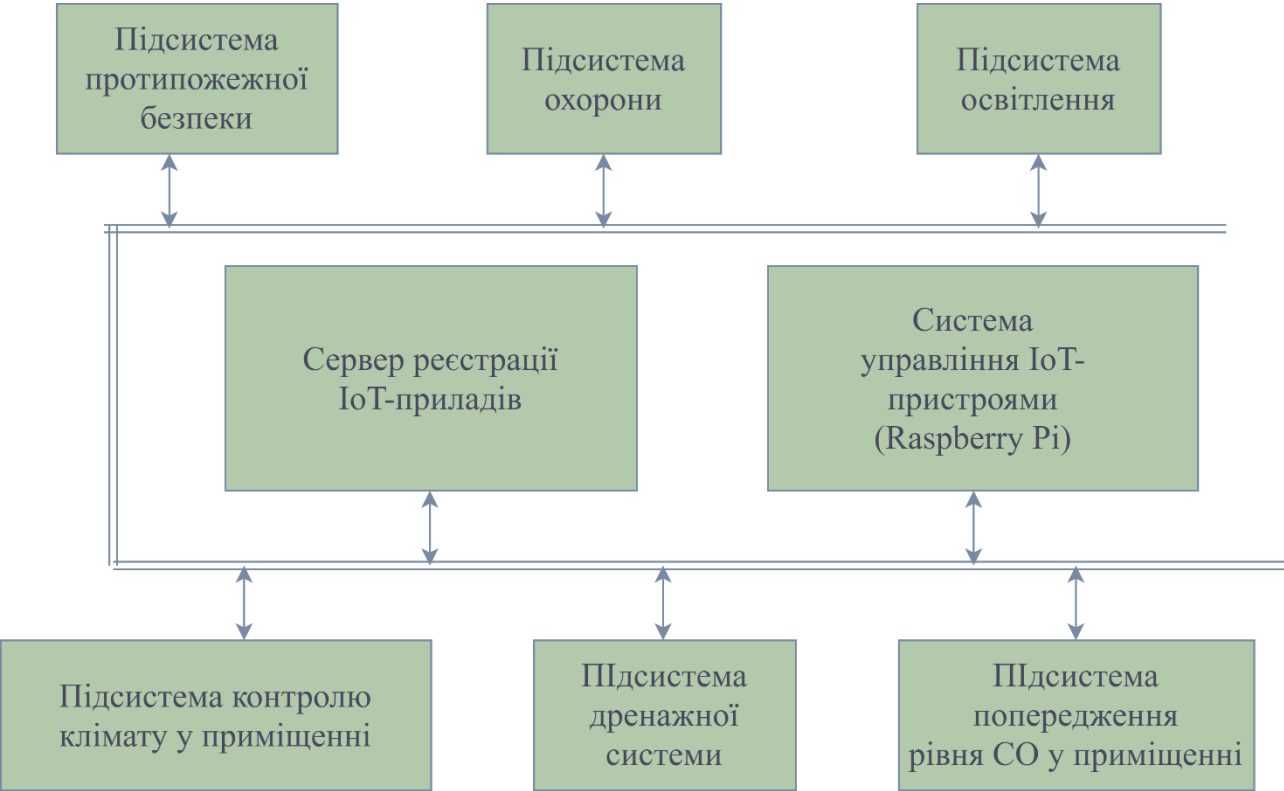


Рисунок 2.1 - Структурна схема системи управління розумним будинком на Raspberry Pi.

Сервер реєстрації IoT-приладів – це частина системи, яка призначена для реєстрації даних, які датчики надсилають для подальшої обробки. Також сервер відповідає за автентифікацію доступу до системи управління розумним будинком. Саме на сервері створюються облікові записи користувачів, фіксуються усі зміни та зберігається статистика показників кожного пристрою.

Система управління IoT-пристроями – це частина системи, яка

виконує формування команд керування підсистемами та відповідає за взаємодію підсистем між собою.

Підсистема протипожежної безпеки – це частина системи, яка відповідає за реагування системи на виявлення вогню. Датчик вогню постійно моніторить навколишнє середовище, та посиляє сигнал до системи управління, у разі виявлення небезпеки. На такий випадок система управління має підготовлений сценарій дій, який повинен допомогти попередити знищення майна господарів використовуючи лише власні ресурси. Протипожежна безпека може відкрити вікна, ввімкнути сигналізацію, надіслати сповіщення до власника будинку, викликати пожежників. Всі дії виконуються автоматично без участі людини, ніяких затримок у прийнятті рішень.

Підсистема охорони – частина системи, в обов’язках якої є реагування на відкриття дверей. Сповіщення власника у разі несанкціонованого проникнення до будинку. Також можна фіксувати на камеру кожне відкриття дверей для фіксації осіб.

Підсистема контролю клімату – це частина системи, яка контролює підтримання певних кліматичних норм в приміщенні. Ціль може досягатися абсолютно різними способами. Найпопулярнішими з них є системи кондиціонування, вентиляції, керування опаленням будинку, управління підігрівом підлоги та інші. Керування може відбуватися практично за будь-якими ознаками, правилами. Від існуючих обмежень на вхідну потужність до балансування навантаження на електромережу будинку.

Інтелектуальний будинок завжди забезпечить комфортний мікроклімат, враховуючи температуру повітря, рівень відносної вологості, кондиціонування, фільтрацію повітря та інше.

Підсистема дренажної системи – це частина системи яка виконує функцію попередження перевищення допустимого рівня води. Вона дозволить впоратися з проблемами сезонного підтоплення, захистити фундамент

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

будинку від руйнування. Також можна використати для попередження затоплення самого будинку.

2.2. Розробка функціональної схеми системи управління розумним будинком на базі Raspberry Pi

Проведемо розробку функціональної схеми системи управління розумним будинком на основі раніше створеної структурної схеми.

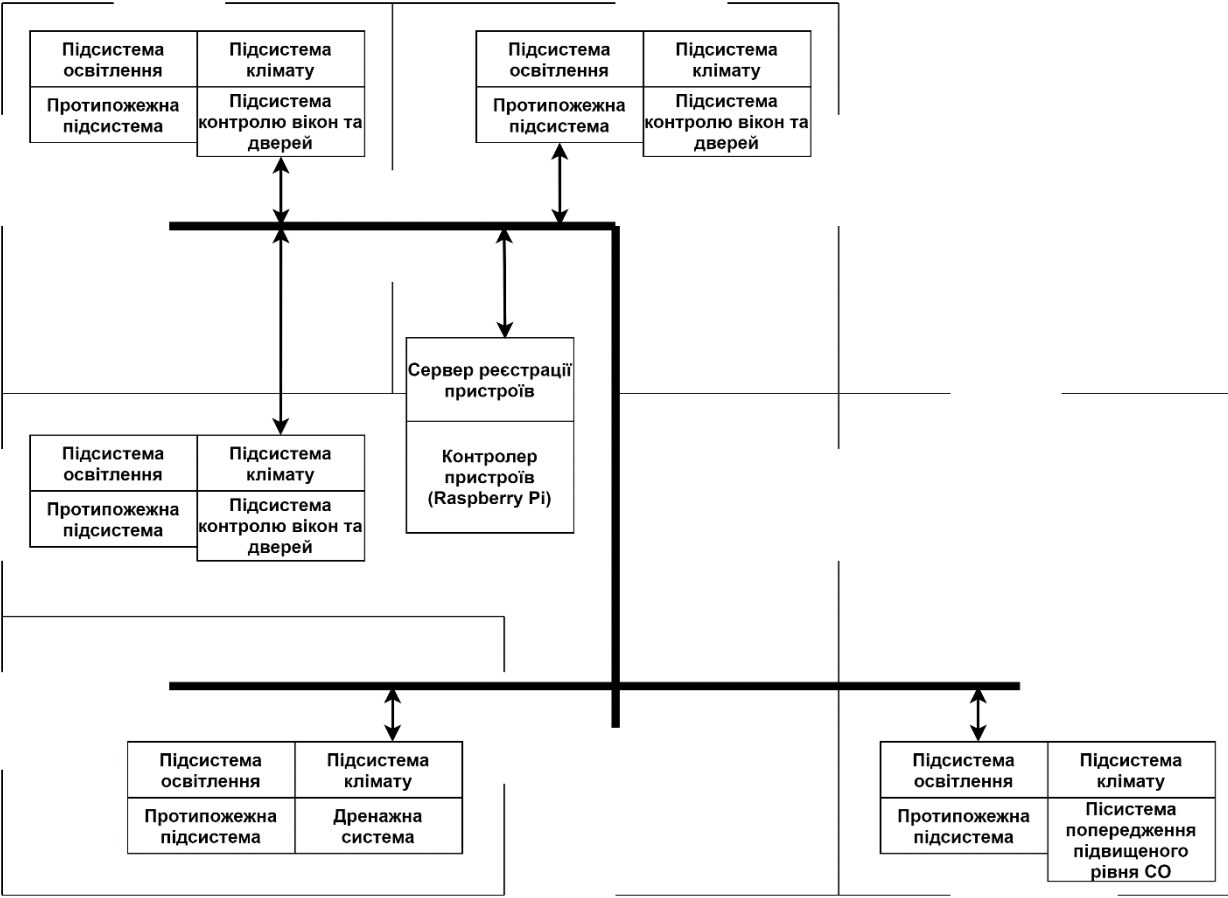


Рисунок 2.2 – Функціональна схема системи управління розумним будинком на базі Raspberry Pi

Типові набори пристроїв підсистем освітлення, охорони та безпеки розміщуються у кожній кімнаті. Виключенням є ванна кімната та гараж. До цих кімнат буде використовуватися особливий підхід з певними особливостями. В

гаражі присутня система контролю рівня СО(датчики чадного газу). Ванна кімната буде оснащена дренажною підсистемою, для попередження перевищення допустимого рівня води у приміщенні.

Роль реєстрації пристроїв та збереження статистики датчиків буде виконувати сервер реєстрації. На цьому ж сервері буде налаштовано роботу таких сервісів як DHCP та DNS. Сервіси не входять до складу системи, але є необхідними для коректного її функціонування.

Сервер реєстрації IoT. В якості елемента зберігання даних та управління буде використовуватись сервер реєстрації IoT. На даному сервері також реалізуються сервіси DHCP та DNS.

DHCP (англ. Dynamic Host Configuration Protocol — протокол динамічної конфігурації вузла) — це мережний протокол, що дозволяє комп'ютерам автоматично одержувати IP-адресу й інші параметри, необхідні для роботи в комп'ютерній мережі TCP/IP. Для цього комп'ютер звертається до спеціального серверу, під назвою сервер DHCP. Мережевий адміністратор може задати діапазон адрес, що розподіляють серед комп'ютерів. Це дозволяє уникнути ручного налаштування комп'ютерів або пристроїв мережі й зменшує кількість помилок. Протокол DHCP використовується в більшості великих мереж TCP/IP.

У нашому випадку сервер DHCP буде видавати IP-адреси мережі 192.168.0.0 з маскою 255.255.255.0 контрольованим пристроям нашої IoT мережі.

DNS (англ. Domain Name System – доменна система імен) — це сервер, який містить базу даних публічних IP-адрес і пов'язаних з ними імен хостів. Як правило, DNS-сервер виконує роль перекладача, дозволяючи або переводячи імена хостів в IP-адреси.

На сервері DNS в таблиці записів буде прописано відповідність між IP-адресами та іменами пристроїв.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

Реєстраційний сервер буде також зберігати контролювати облікові записи, створені для доступу до керування системою розумного будинку. Відповідно користувачі даної системи мають доступ до облікових записів відповідно до прав, які може встановити адміністратор або перший зареєстрований користувач.

Симулятор Cisco Packet Tracer має обмеження стосовно облікових засобів. Там можна використовувати лише один обліковий запис, тому керувати пристроями через Local Control або Remote Control можна тільки з одного облікового запису.

2.3. Розробка програмного забезпечення системи управління

На основі спроектованої структурної схеми необхідно створити алгоритми роботи кожної з підсистем. Це допоможе спростити задачу реалізації всієї системи на етапі створення модельного прикладу в програмному середовищі.

Робота підсистеми безпеки передбачає наявність датчиків вогню, датчика диму та датчиків CO та CO₂. Датчики надсилають відомості про навколишнє середовище до сервера реєстрації, який моніторить надходження даних та у випадку перевищення допустимого рівня певних параметрів переходить у режим тривоги. Режим тривоги передбачає виконання певного алгоритму дій. У разі реакції датчиків на вогонь або дим контролер буде вмикати протипожежні спринклери та звукову сигналізацію. Також необхідно в такій ситуації розблокувати вікна та двері аварійного приміщення для евакуації. Коли стан навколишнього середовища нормалізується, контролер пристроїв продовжує моніторити покази датчиків.

За таким же принципом працює система попередження рівня чадного газу у приміщенні. Загальну схему роботи системи безпеки приведено на рисунку 2.3.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37



Рисунок 2.3 – Алгоритм функціонування протипожежної системи

		Дядюшкін Р. В.		
		Гулько І. В.		
Змн.	Арк.	№ докум.	Підпис	Дата

КІ.КР.Б – 123 – 21 – ПЗ

Арк.

38

Головну роль у контролі температури виконує термостат. Він моніторить температуру навколишнього середовища та виконує один із варіантів, залежно від налаштувань, або активувати нагрівальний елемент, або ввімкнути кондиціонування. Також користувач зможе налаштовувати підсистему на свій розсуд, використовуючи певні умови, наприклад, час доби, пора року або температура навколишнього середовища. Загальна схема алгоритму роботи підсистеми контролю клімату зображено на рисунку 2.4.

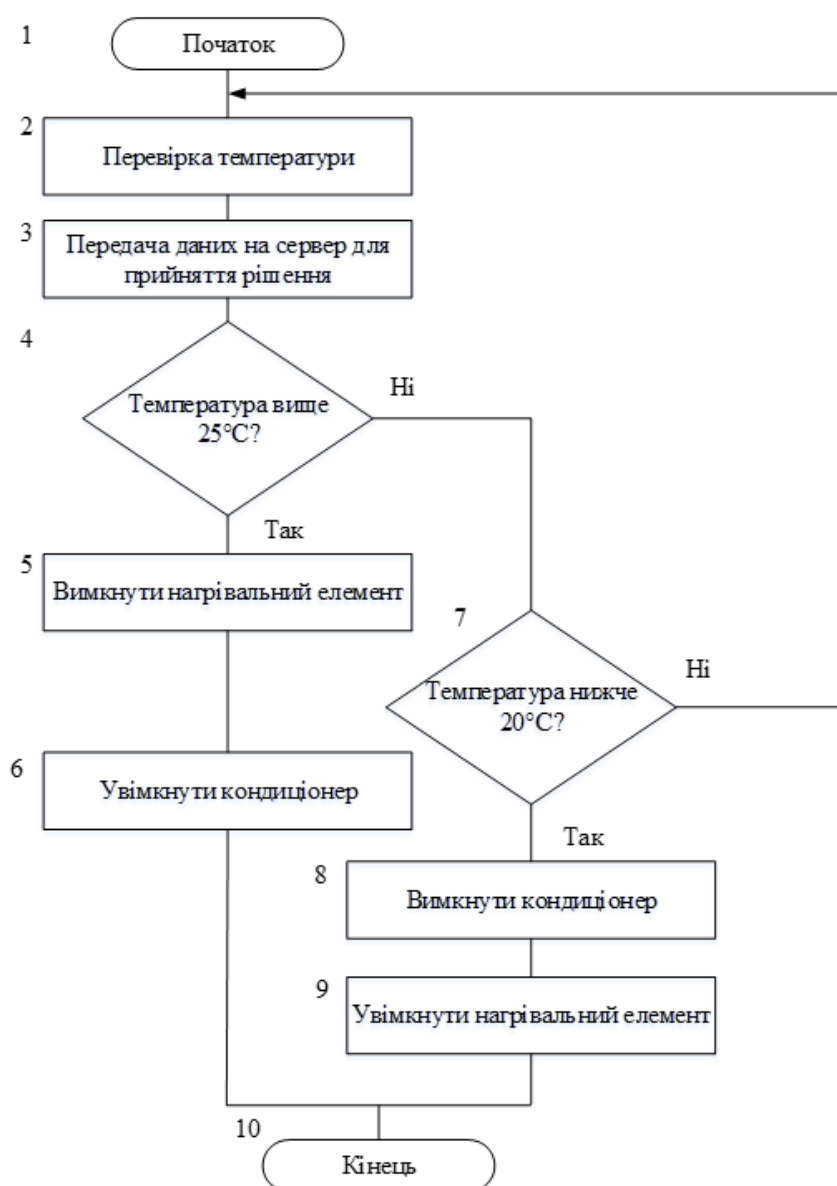


Рисунок 2.4 – Алгоритм роботи підсистеми контролю клімату.

Функціонування підсистеми контролю освітлення передбачає наявність датчиків освітлення в кожній кімнаті. Датчики будуть надсилати дані до сервера, який на основі створених правил та вхідних даних буде регулювати роботу пристроїв освітлення.

Логіку роботи підсистеми освітлення можна встановити досить різними алгоритмами. Основний перелік умов та факторів буде прописано на сервері. Користувач зможе налаштувати алгоритм роботи на свій смак оперуючи існуючими умовами контролю. Загальна схема алгоритму роботи підсистеми освітлення зображена на рисунку 2.5.



Рисунок 2.5 – Загальний алгоритм роботи системи освітлення

Суть роботи дренажної системи у перевірці рівня води в кухні та ванній кімнаті. Датчик буде передавати дані до сервера реєстрації та на основі вхідних даних реагувати відповідним чином. За умов перевищення допустимого рівня води буде посилатися сигнал до дренажів з командою на відкриття. Після повернення рівня води до допустимого значення дренажі перейдуть до закритого стану та продовжиться моніторинг рівня води. Алгоритм роботи дренажної підсистеми зображено на рисунку 2.6.

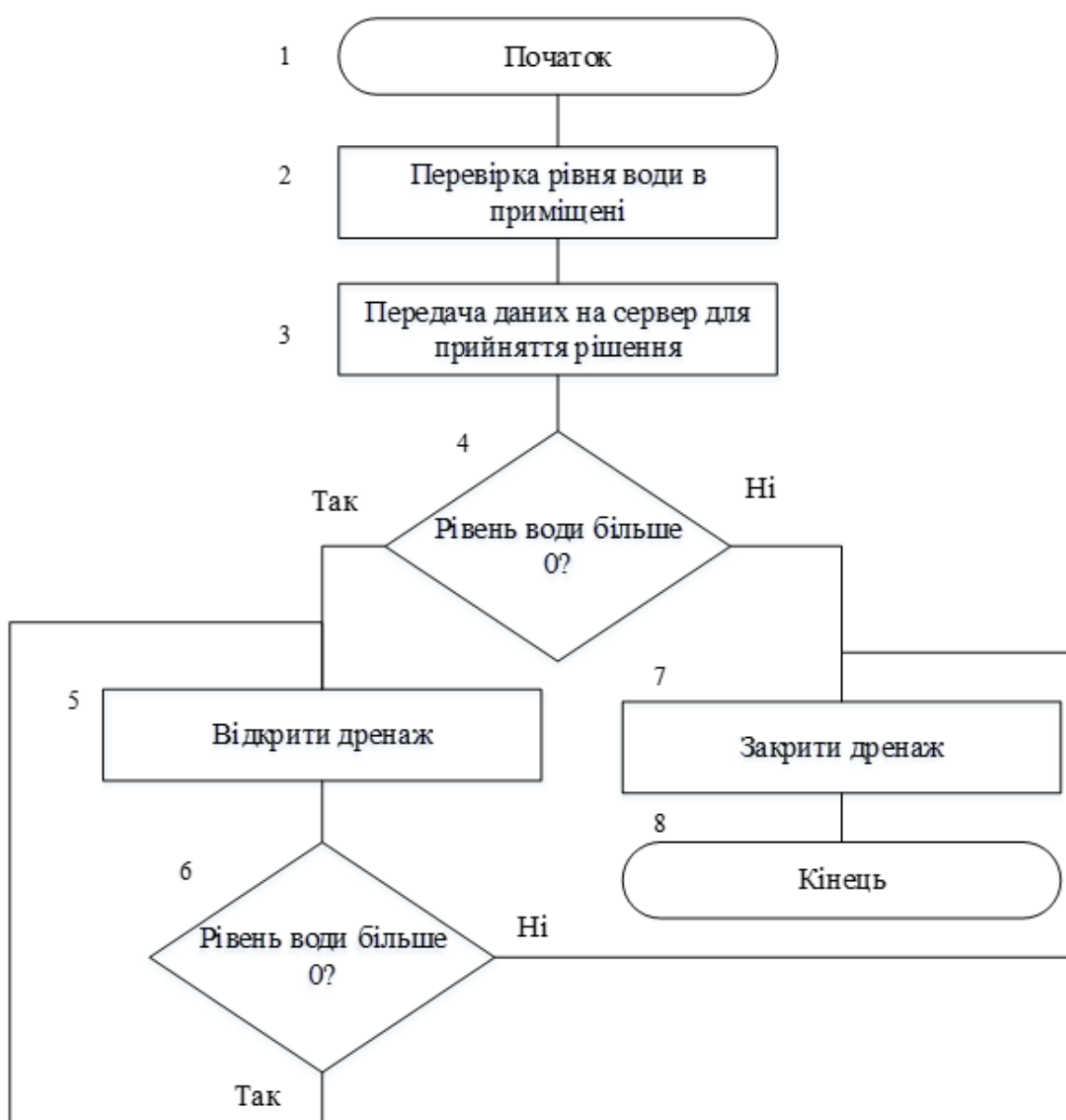


Рисунок 2.6 – Алгоритм роботи дренажної системи

Висновки по 2 розділу

У ході виконання другого розділу було виконано розробку структурної та функціональної схеми для системи управління розумним будинком на базі Raspberry Pi, яка містить у собі підсистеми контролю освітлення та мікроклімату, системи протипожежної безпеки та системи управління підсистемами розумного будинку.

Розроблено алгоритми функціонування для кожної підсистеми на прикладі одноповерхового житлового будинку.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

РОЗДІЛ 3. ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ СИСТЕМИ УПРАВЛІННЯ РОЗУМНИМ БУДИНКОМ

3.1. Розробка моделі пристрою в Cisco Packet Tracer

Cisco Packet Tracer – це потужний інструмент для моделювання та візуалізації комп'ютерних мереж, створений компанією Cisco. Програма-симулятор дозволяє налаштовувати різні мережеві пристрої фірми Cisco (маршрутизатори, комутатори, IP-телефони, сервери, міжмережні екрани Cisco ASA та багато іншого).

Для розробки моделі пристрою в програмному середовищі Cisco Packet Tracer спочатку необхідно створити комп'ютерну мережу для нашого розумного будинку. Головна частина буде складатися з комутатора, маршрутизатора та шлюзу розумного будинку.

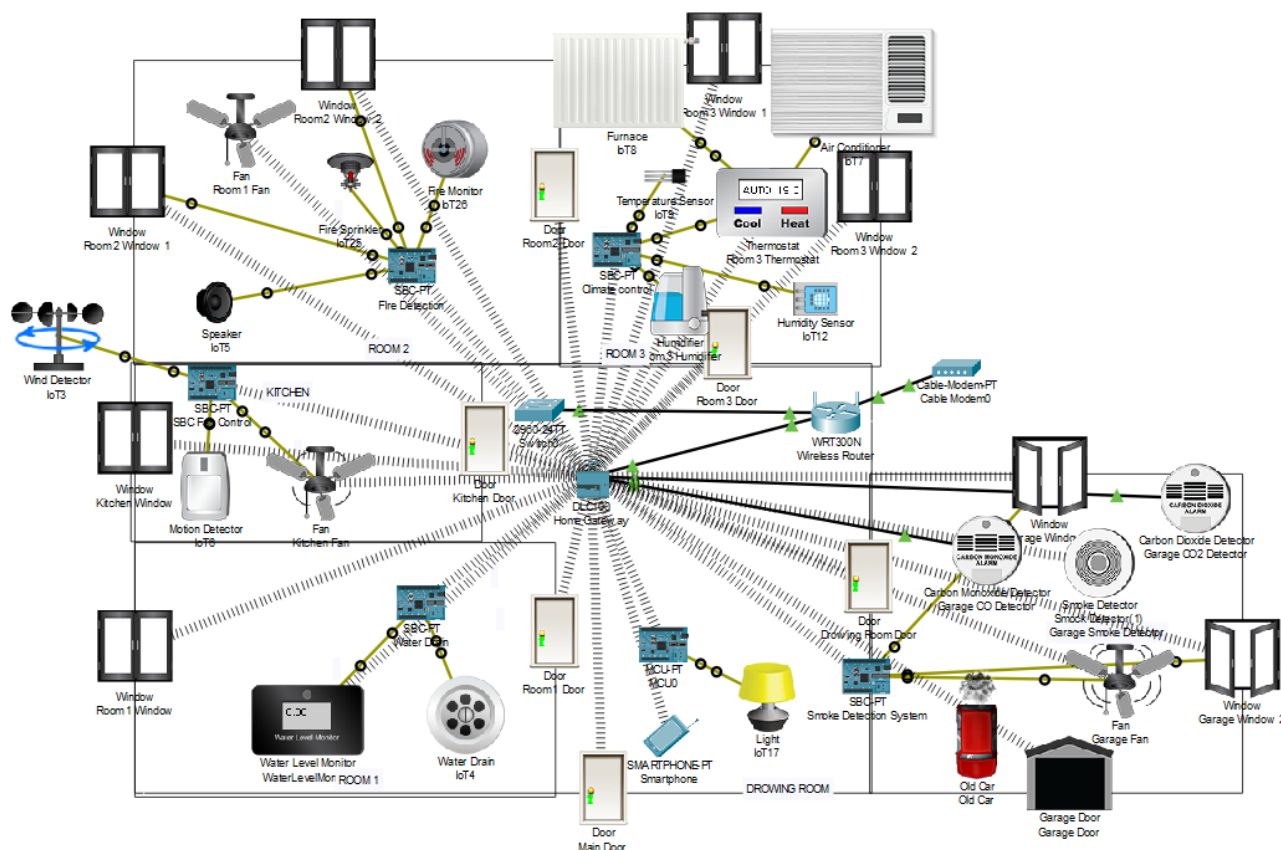


Рисунок 3.1 – Вигляд спроектованої моделі системи управління розумним будинком.

Для того, щоб налаштувати Home Gateway необхідно перейти в налаштування пристрою(вкладка Config). У розділі Internet виберемо DHCP(динамічне налаштування параметрів IP-адресації) (Рисунок 3.1).

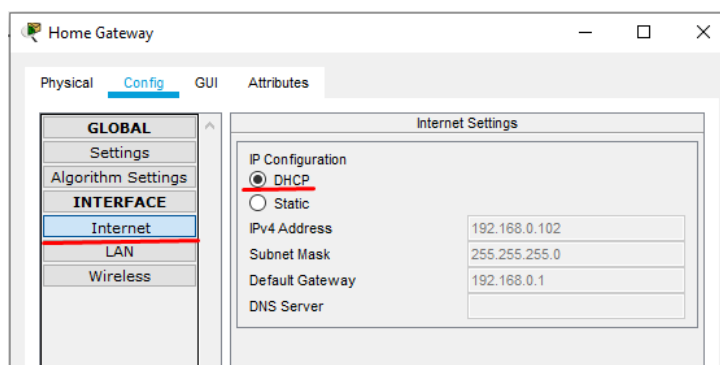


Рисунок 3.2 – Базові налаштування Home Gateway для подальшої роботи.

Наступним кроком буде налаштування точки доступу Wi-Fi, до якої будуть під'єднуватися пристрої для взаємодії та керування. Змінимо SSID по замовчуванню на SmartHomeGateway та налаштуємо WPA2-PSK автентифікацію Wi-Fi мережі. Пароль – PKTCisco.

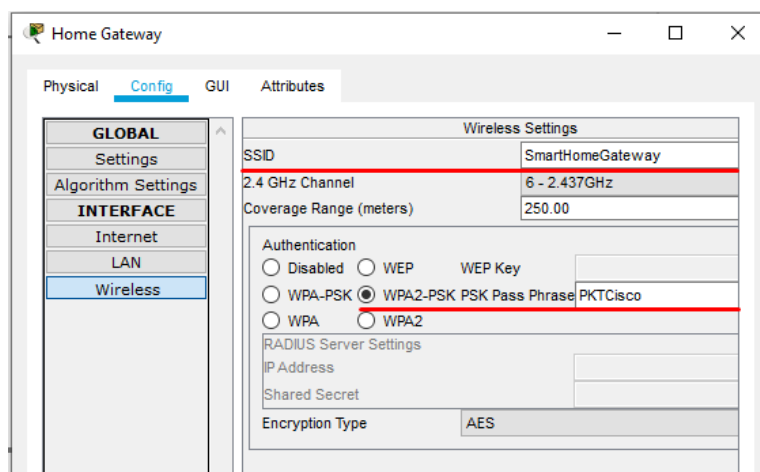


Рисунок 3.3 – Налаштування Wi-Fi мережі на Home Gateway.

Для контролю та моніторингу пристроїв нам знадобиться Smart Device. Під'єднаємо його до раніше створеної Wi-Fi мережі. Отримаємо динамічно IP адресу, обравши пункт DHCP у секції конфігурації IP-адресації.

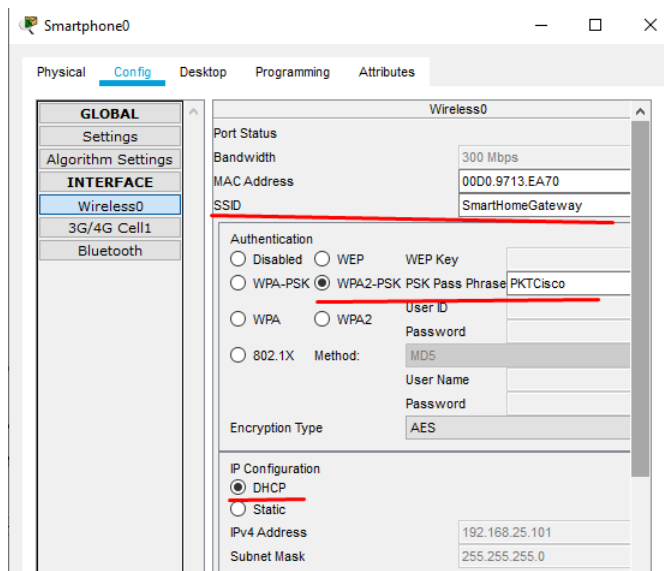


Рисунок 3.4 – Налаштування IP-адресації розумного девайса та підключення його до Home Gateway.

Після налаштування Smart Device необхідно перейти до вкладки Desktop та запустити утиліту IoT Monitor. Утиліта запитає логін та пароль для автентифікації(по замовчуванню admin/admin). Також необхідно вказати IP-адресу Home Gateway пристрою або реєстраційного сервера IoT-пристроїв.

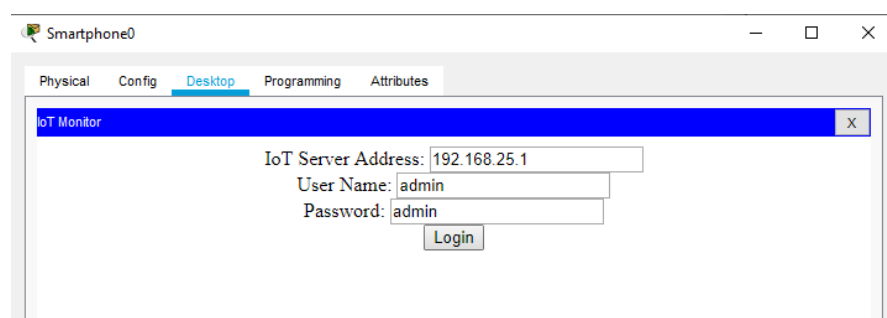


Рисунок 3.5 – Логін до системи управління Home Gateway.

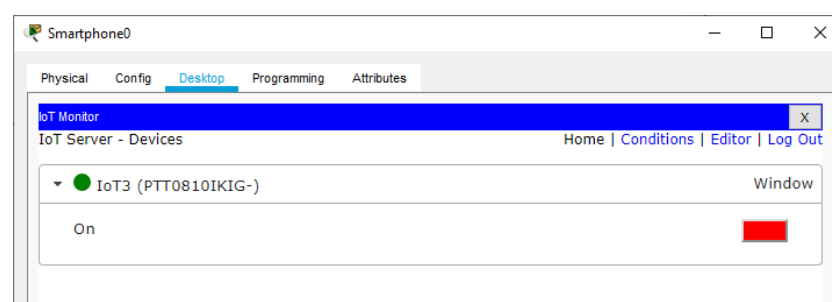


Рисунок 3.6 – Відображення першого доданого девайсу.

Програмний симулятор може симулювати наступні типи взаємодії між девайсами:

Local Control – виконання локального контролю девайсів шляхом управління з використанням SBC(Single Board Computer), MCU(Micro Controller Unit) або Thing. Також може бути задіяно реєстраційний сервер, який повинен бути підключений до однієї підмережі з пристроєм. Саме цей вид взаємодії було обрано для подальшої розробки; [11]

Direct Control – прямий контроль девайсів шляхом безпосередньої взаємодії з ним. Для цього в симуляторі існує комбінація клавіш Alt+ЛКМ;

Remote Control – підключення, яке необхідно для віддаленого керування пристроями(які знаходяться в іншій підмережі). Програмний симулятор Packet Tracer підтримує цей тип контролю з певними обмеженнями, які будуть заважати в процесі подальшої розробки.

Одними з найпростіших IoT-девайсів в програмному симуляторі є вікна та двері. Їх буде під'єднано до Wi-Fi мережі девайса Home Gateway. Керування більшістю дверей та вікон буде реалізовано через локальний контроль без використання SBC/MCU. Керування буде відбуватися через Home Gateway. Двері та вікна можна віднести до групи пристроїв, які відносяться до протипожежної безпеки.

SBC – це комп'ютер, всі основні компоненти якого розміщуються на одній платі. Найчастіше, одноплатні комп'ютери зустрічаються як промислові комп'ютери, та останнім часом як вбудовані або комплексні офісні комп'ютери.

На відміну від настільного персонального комп'ютера, одноплатні комп'ютери часто не покладаються на слоти розширення для периферійних функцій. Одноплатні комп'ютери були побудовані з використанням широкого спектру мікропроцесорів. Прості конструкції, побудовані комп'ютерними любителями, часто використовують статичну оперативну пам'ять та недорогі 8- або 16-бітні процесори.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

MCU – це невеликий комп'ютер, побудований на системі мікросхеми (SoC). Він схожий на SBC, але містить меншу обчислювальну потужність та функціональність. Він містить ядро процесора, пам'ять та програмовані периферійні пристрої вводу/виводу. Мікроконтролери призначені для вбудованих програм або програм, які потребують небагато комп'ютерних ресурсів.

Для реалізації протипожежної безпеки буде застосовано датчик вогню(Fire Detector), гучномовець(Speaker), протипожежна система(Fire Sprinkler) та два вікна.

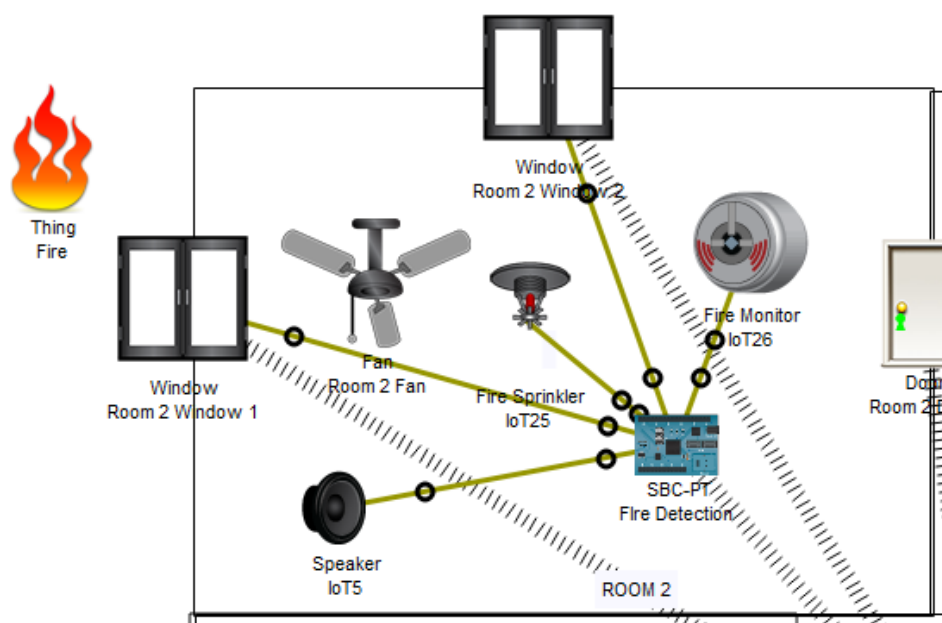


Рисунок 3.7 – Побудова протипожежної сисистеми.

Для системи контролю клімату буде задіяно наступні пристрої: термостат(Thermostat), кондиціонер(Air Conditioner), обігрівач(Furnace), зволожувач повітря(Humidifier), датчик вологості повітря(Humidity Monitor).

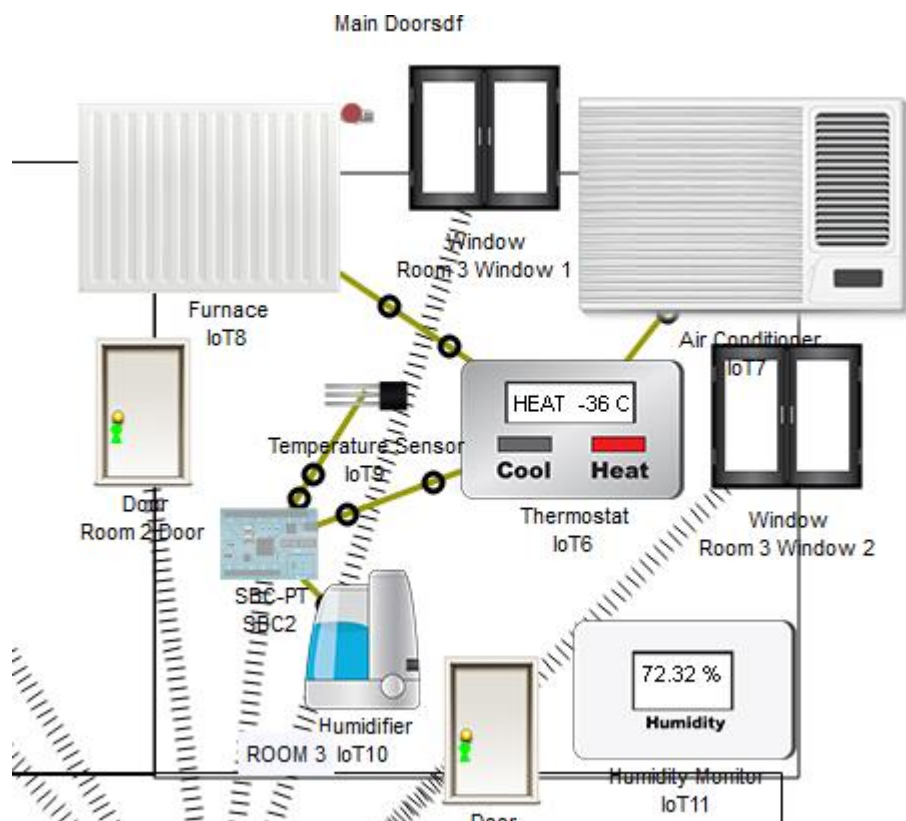


Рисунок 3.8 – Побудова контролю клімату та вологості повітря.

Дренажна система буде складатися з датчика рівня води(Water Level Monitor) та стоку(Water Drain).

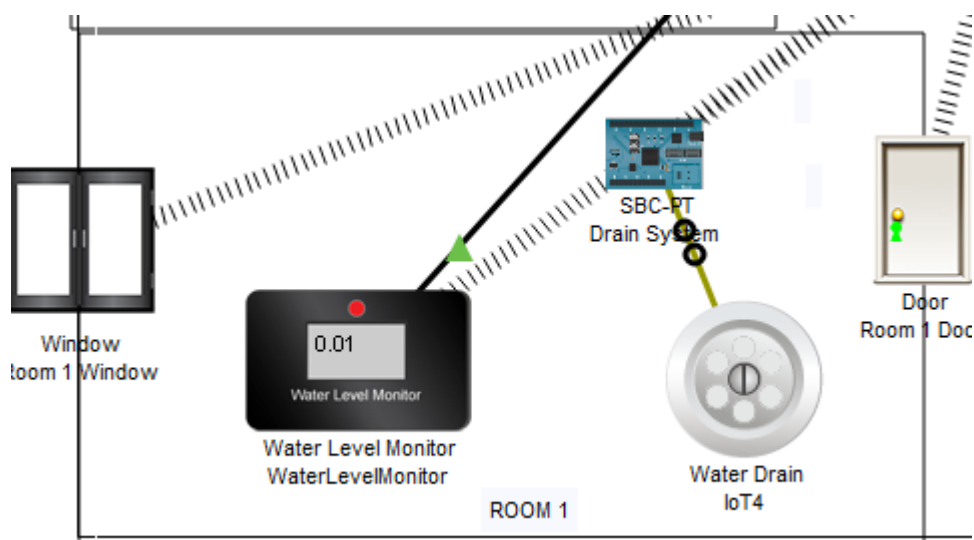


Рисунок 3.9 – Побудова дренажної системи.

		Дядюшкін Р. В.		
		Гулько І. В.		
Змн.	Арк.	№ докум.	Підпис	Дата

КІ.КР.Б – 123 – 21 – ПЗ

Арк.

48

Розумний гараж буде функціонувати на основі датчика диму (Smoke detector), датчика CO (Carbon Monoxide Detector), датчика CO2 (Carbon Dioxide Detector), вентилятора(Fan) та двох вікон.

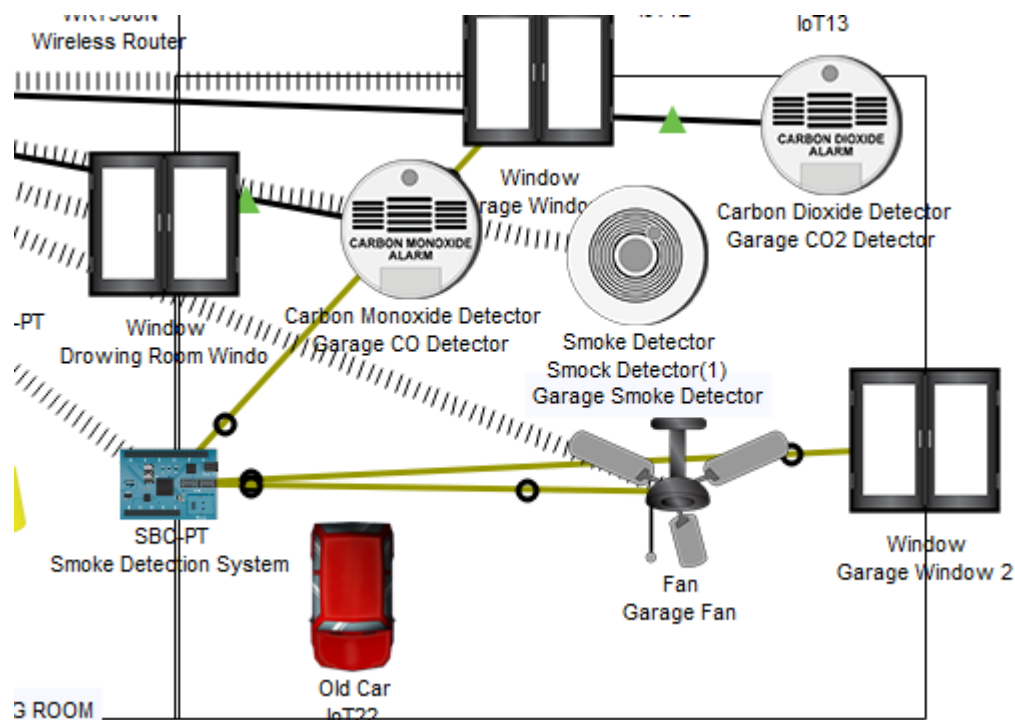


Рисунок 3.10 – Побудова системи виявлення диму для гаража.

Розумне освітлення буде містити у собі Photo Sensor (фотодатчик) та Light (пристрій для освітлення). Роботою системи розумного освітлення можна буде керувати як через Home Gateway, так і через SBC.

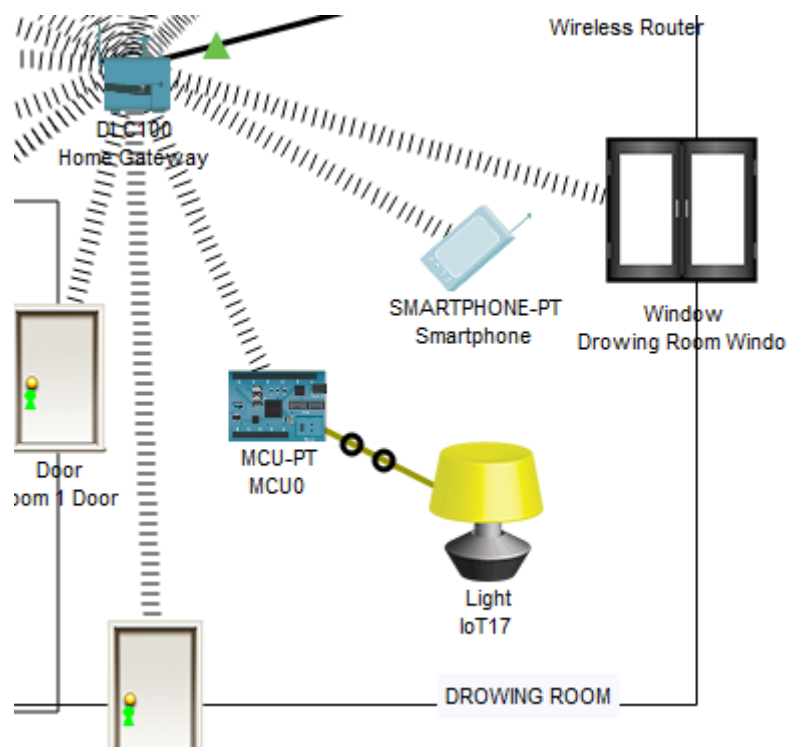


Рисунок 3.11 – Побудова системи контролю розумного освітлення.

3.2. Перевірка працездатності та ефективності по моделі пристрою управління

Настав час дослідити працездатність готової моделі системи управління розумним будинком.

3.2.1. Перевірка роботи протипожежної безпеки.

Для перевірки протипожежної безпеки необхідно створити імітацію пожежі. Візьмемо для цього пристрій Thing у вкладці Programming пропишемо наступні рядки:

```
function setup() {
  setDeviceProperty(getName(), "IR", 900);
}
```

Відповідно створеному раніше алгоритму, при виявленні вогню, одноплатний комп'ютер повинен виконати наступні кроки:

1. Ввімкнути водопостачання протипожежної системи

2. Подати сигнал на чучномовець
3. Відкрити вікна
4. Відправити відомість про подію на Home Gateway

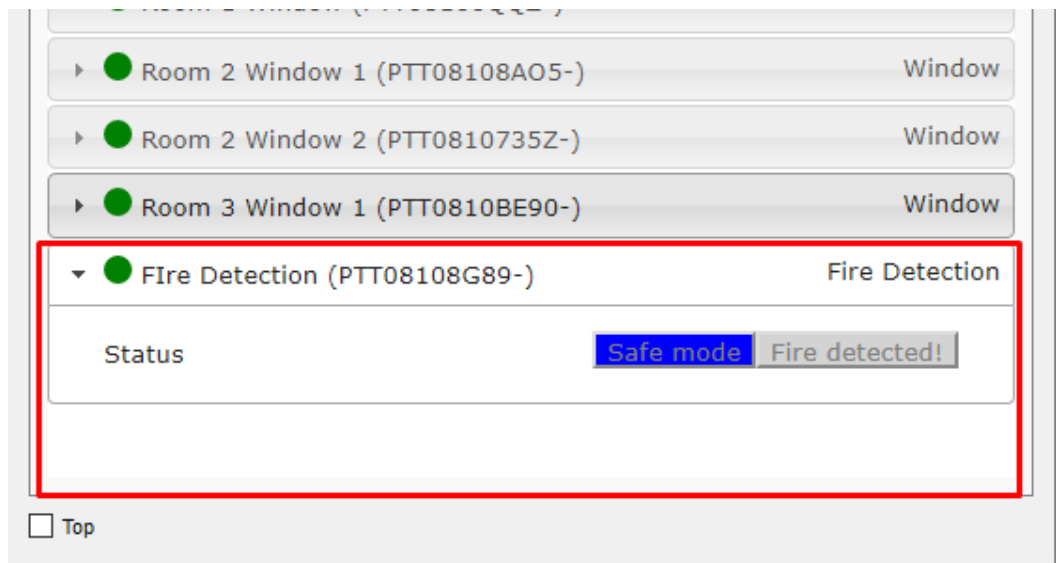


Рисунок 3.12 – Початковий стан протипожежної системи на Home Gateway.

Судячи з зображення, протипожежна система знаходиться у початковому стані(Safe mode).

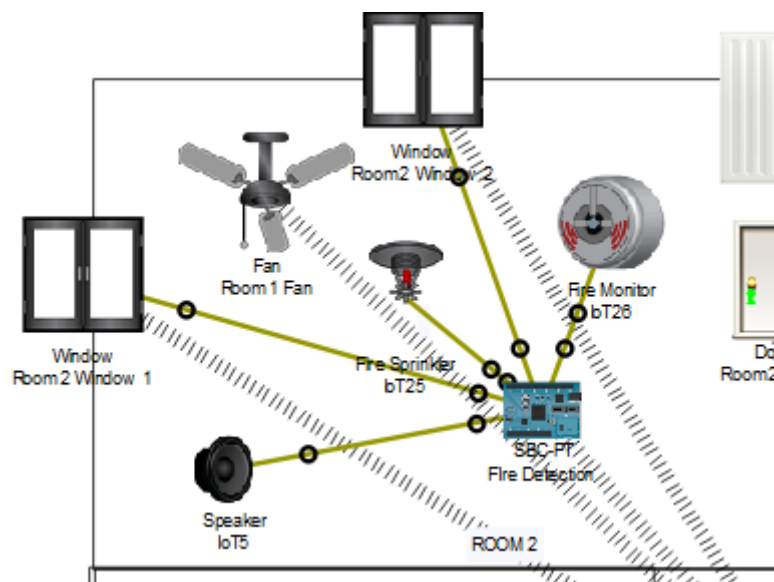


Рисунок 3.13 – Режим роботи протипожежної системи у режимі Safe mode.

Піднесемо до датчика вогню раніше створений вогонь та подивимося на результат.

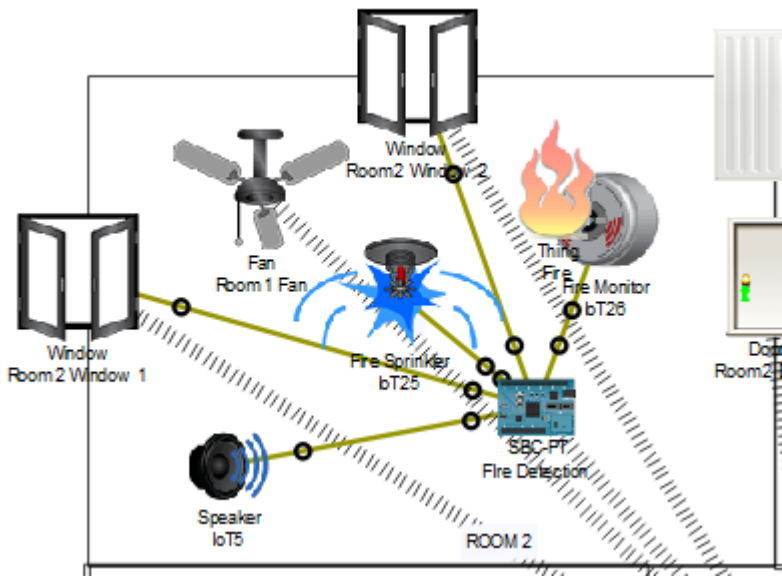


Рисунок 3.14 – Реакція пристроїв протипожежної системи на вогонь.

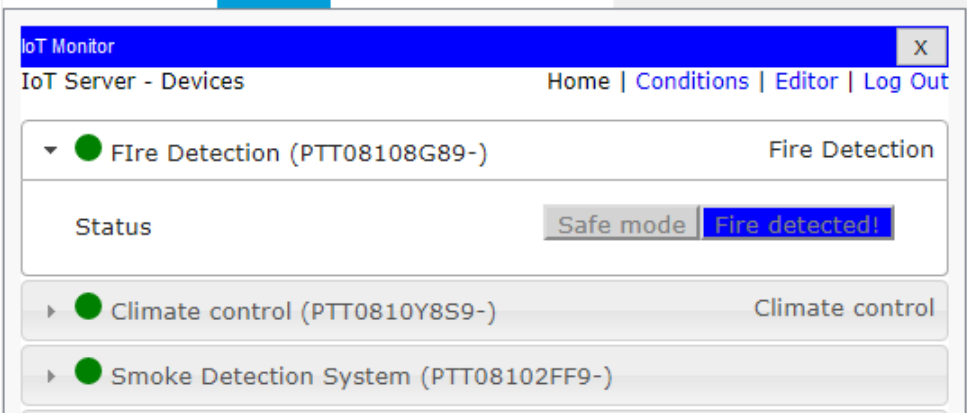


Рисунок 3.15 – Реакція пристроїв протипожежної системи на вогонь.

Усі необхідні пристрої перейшли до потрібного стану. Протипожежна система спрацювала коректно.

3.2.2. Перевірка роботи дверей та вікон

Для перевірки коректності роботи вікон та дверей перейдемо до інтерфейсу управління нашого Home Gateway та розгорнемо відомості про

стан усіх вікон, дверей та датчиків CO та CO2 гаража. Активуємо автомобіль та перевіримо реакцію пристроїв на підвищення рівня CO та CO2 у приміщенні.

У разі невдачі буде необхідно перевірити підключення до Home Gateway та налаштування IP-адресації пристроїв.

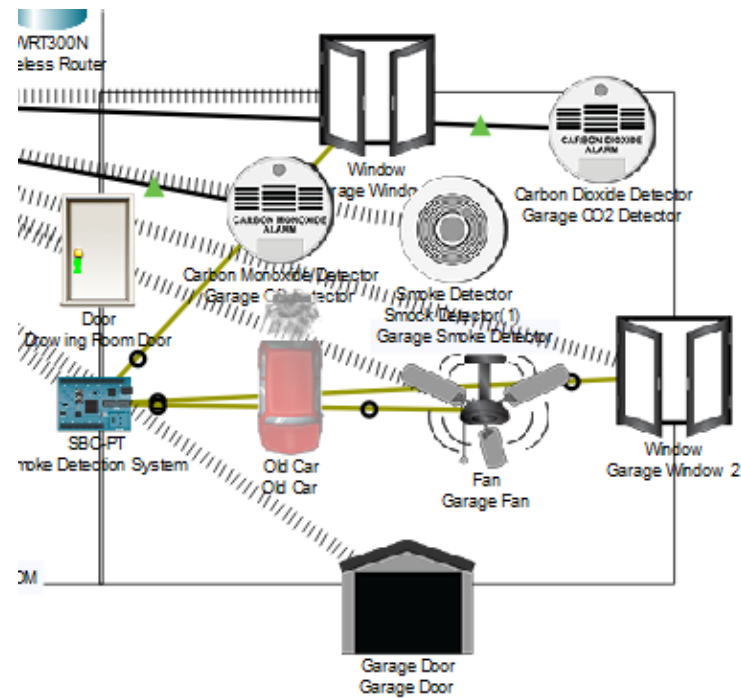


Рисунок 3.16 – Коректна реакція пристроїв на датчик CO.

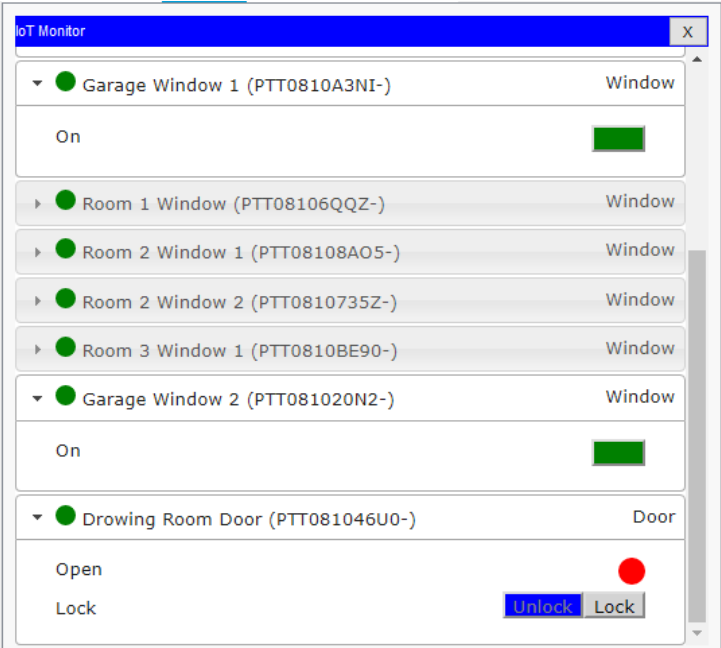


Рисунок 3.17 – Коректна реакція пристроїв на команди Home Gateway.

3.2.3. Перевірка роботи вентилятора

Для налаштування роботи вентилятора було використано сам вентилятор, датчик руху, датчик вітру та одноплатний комп'ютер. Вентилятор має працювати лише за умов наявності виявлення руху датчиком руху та відсутності вітру. Запрограмуємо SBC Fan Control наступним чином:

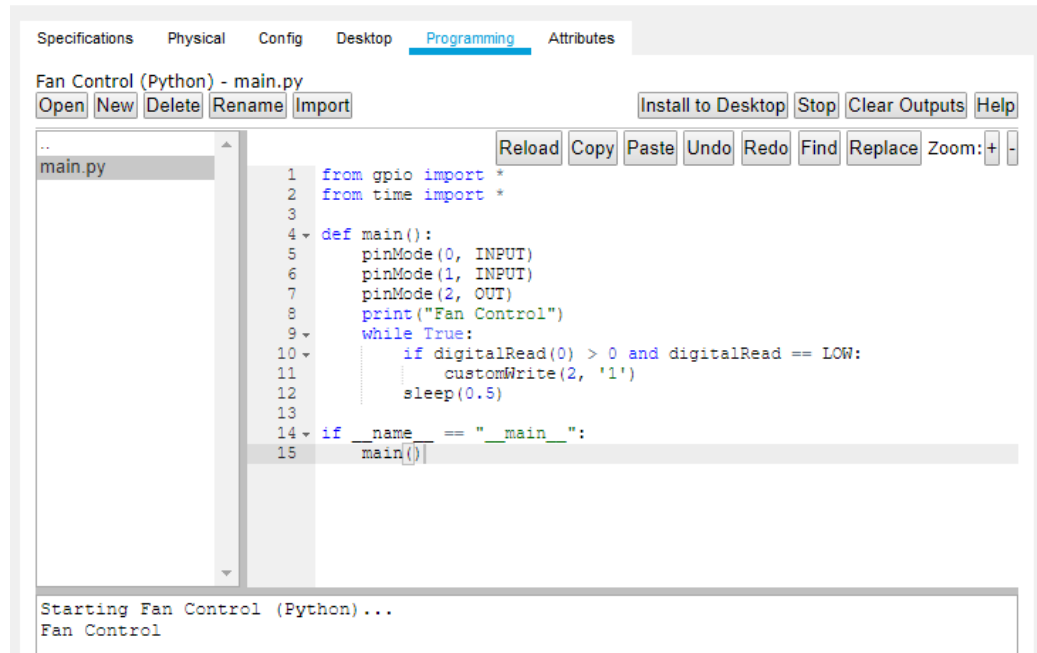


Рисунок 3.18 – Програма для контролю вентилятора з одноплатного комп'ютера.

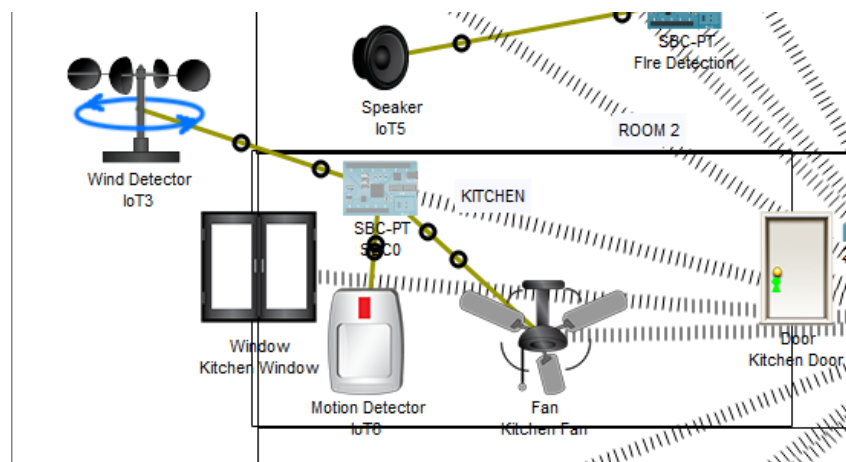


Рисунок 3.19 – Приклад реакції вентилятора на датчик руху та на відсутність вітру.

3.2.4. Перевірка системи контролю клімату

За управління температурою повітря відповідає термостат, який управляється одноплатним комп'ютером. По замовчуванню він перебуває у вимкненому стані



Room 3 Thermostat (PTT0810ZY77-)

Status: **Off** | Cooling | Heating | Auto

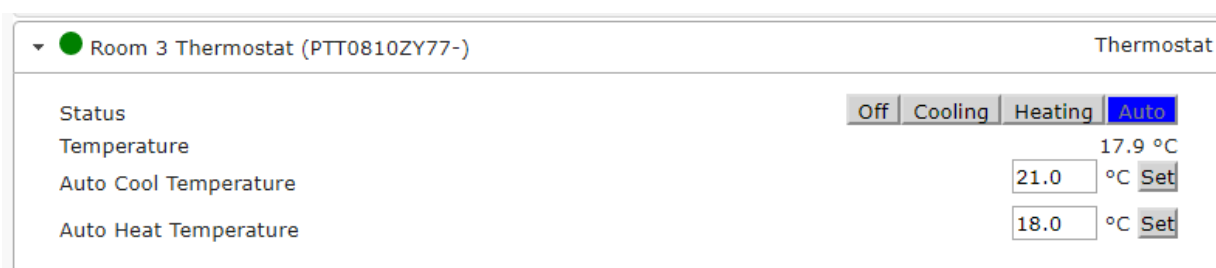
Temperature: 17.1 °C

Auto Cool Temperature: N/A °C **Set**

Auto Heat Temperature: N/A °C **Set**

Рисунок 3.20 – Початковий стан роботи термостата.

Переведемо його в автоматичний режим на будемо спостерігати за поведінкою девайсів нагрівання та охолодження повітря.



Room 3 Thermostat (PTT0810ZY77-)

Status: Off | Cooling | Heating | **Auto**

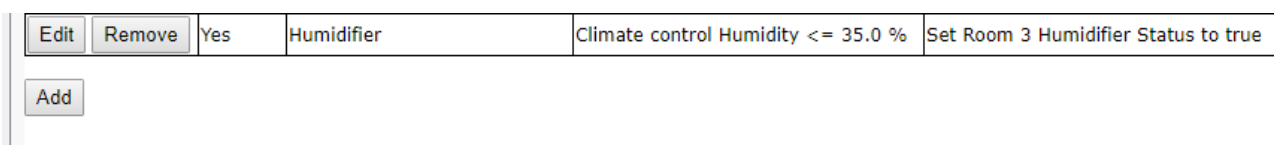
Temperature: 17.9 °C

Auto Cool Temperature: 21.0 °C **Set**

Auto Heat Temperature: 18.0 °C **Set**

Рисунок 3.21 – Приклад налаштування термостата.

Також налаштуємо умови роботи зволожувача повітря.



Edit Remove Yes Humidifier Climate control Humidity <= 35.0 % Set Room 3 Humidifier Status to true

Add

Рисунок 3.22 – Правило роботи зволожувача повітря.

тригер паралельно спостерігаючи за станом пристроїв дренажної системи.

Add Rule

Name

Water Drain Rule

Enabled

☒

If:

Match

All

WaterLevelMonitor

Water Level

>=

5

cm

+ Condition

+ Group

Then set:

Water Drain

On

to

true

+ Action

Рисунок 3.25 – Правило роботи для дренажної системи.

Рисунок 3.26 – Результат спрацювання правила.

Навіть при прямому водопостачанні рівень води не збільшується за рахунок відкритого стоку. Система працює коректно та вірно реагує на визначені події.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3. Рекомендації по застосуванню розробленого пристрою управління розумним будинком

Для моделювання системи розумного будинку пропонуються наступний алгоритм:

1. Першим етапом буде підготовка технічного завдання для майбутнього проекту. Цей етап передбачає адаптацію вимог роботи та функцій системи до планування будинку. Також враховуються приблизна вартість майбутнього рішення.
2. На основі даних попереднього етапу відбувається вибір комплектуючих пристроїв системи та налаштування програмного забезпечення під потреби конкретного користувача.
3. Наступним кроком буде моделювання проекту у програмному середовищі Cisco Packet Tracer для перевірки працездатності системи та задоволення вимог замовника.
4. Четвертим кроком буде адаптація готової моделі до реального будинку. На цьому етапі буду плануватися особливості розміщення пристроїв на основі особливостей конструкції будинку. Відбувається планування прокладання кабельних зв'язків та розміщення мережевого обладнання.
5. Налаштування окремих компонентів системи управління розумним будинком, таких як датчики, контролери та одноплатний комп'ютер Raspberry Pi.
6. Налаштування зовнішніх підключень локальної внутрішньої мережі будинку та перевірка наявності підключення всіх пристроїв до сервера реєстрації.
7. Тестування механізмів управління пристроями для попередження непередбачених сценаріїв у ході експлуатації системи та безпечного користування механізмами управління підсистемами.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

8. Попередня оцінка вартості всього проекту для подальшого вибору необхідних комплектуючих з урахуванням бюджету замовника на реалізацію проекту.
9. Погодження готової моделі та попередньої оцінки вартості з замовником. За умов погодження готової моделі проекту відбувається розробка технічної документації та відбувається реалізації проєкту системи управління розумним будинком. У разі виникнення зауважень стосовно реалізації та додаткових побажань повторюється алгоритм моделювання починаючи з пункту 3 та відбувається корекція проєкту на основі зворотного зв'язку від замовника.

Висновки по 3 розділу

У третьому розділі було виконано моделювання проєкту системи управління розумним будинком на базі одноплатного комп'ютера Raspberry Pi у програмному середовищі Cisco Packet Tracer.

Також було досліджено та перевірено працездатність моделі.

Проведено розробку рекомендацій по моделюванню та адаптації проєкту розумного будинку до реальних умов використання.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено аналіз та обґрунтування актуальності розробки систем управління розумним будинком на базі одноплатного комп'ютера Raspberry Pi.

Виконано аналіз типових архітектурних рішень систем Інтернету речей та перспектив подальшого розвитку та покращень.

Було розроблено структурну та функціональну схеми для системи управління розумним будинком на базі Raspberry Pi, яка містить у собі підсистеми контролю освітлення та мікроклімату, системи протипожежної безпеки та системи управління підсистемами розумного будинку.

Розроблено алгоритми функціонування для кожної підсистеми на прикладі одноповерхового житлового будинку.

Виконано моделювання проекту у програмному середовищі Cisco Packet Tracer, перевірено функціонування кожної підсистеми та приведено список рекомендованих етапів для проектування та моделювання системи під конкретного замовника.

В перспективі буде можливим покращення системи за рахунок розширення функціоналу, підвищення рівня зручності використання, розширення сумісності системи управління з більшою кількістю пристроїв. Також буде доречно автоматизувати процес налаштування та розгортання системи.

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ли П. Архитектура интернета вещей / пер. с англ. М. А. Райтмана. – М.: ДМК Пресс, 2019. – 454 с.: ил.
2. Росляков, А.В. Интернет вещей: учебное пособие [текст] / А.В. Росляков, С.В. Ваяшин, А.Ю. Гребешков. – Самара: ПГУТИ, 2015. – 200 с.
3. Муромцев Д.И., Шматков В.Н. «Интернет Вещей: Введение в программирование на arduino» – СПб: Университет ИТМО, 2018. – 36 с.
4. Петин В.А. «Arduino и Raspberry Pi в проектах Internet of Things» – БХВ-Петербург, 2016. – 320 с.
5. Петин В.А. «Микрокомпьютеры Raspberry Pi. Практическое руководство» – БХВ-Петербург, 2015. – 240 с.
6. Архітектура та технології IoT. [Електронний ресурс]. Режим доступу: <https://learn.ztu.edu.ua/enrol/index.php?id=1985>
7. FIBARO smart home. Home Center 2. [Електронний ресурс]. Режим доступу: <https://www.fibaro.com/en/products/home-center-2>
8. Модуль управління розумним будинком Wink Hub 2. [Електронний ресурс]. Режим доступу: <https://www.wink.com/products/wink-hub-2/>
9. Xiaomi Smart Home Gateway 2 — шлюз управління розумним будинком. [Електронний ресурс]. Режим доступу: <https://xiaomi-smarthome.ru/gateway-2/>
10. Raspberry Pi. Работа с портами ввода-вывода GPIO. [Електронний ресурс]. Режим доступу: <https://microtechnics.ru/raspberry-pi-rabota-s-portami-vvoda-vyvoda-gpio/>
11. Packet Tracer – Connect and Monitor IoT Devices. [Електронний ресурс]. Режим доступу: <https://honim.typepad.com/files/6.1.1.2-packet-tracer---connect-and-monitor-iot-devices.pdf>
12. Implementation Of Smart Home By Using Packet Tracer. [Електронний ресурс]. Режим доступу: <http://www.ijstr.org/final-print/feb2020/Implementation->

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

Of-Smart-Home-By-Using-Packet-Tracer.pdf

13. Packet Tracer 8.0 - IoT devices configuration. [Електронний ресурс].
Режим доступу: <https://www.packettracernetwork.com/internet-of-things/pt7-iot-devices-configuration.html>

14. IoT Simulations with Cisco Packet Tracer. [Електронний ресурс].
Режим доступу: <https://www.theseus.fi/bitstream/handle/10024/150158/Andrea%20Finardi%20%20Master%20of%20Engineering%20%20Information%20technology.pdf>

15. Packet Tracer 7.0 - IoT advanced programming & automation. [Електронний ресурс]. Режим доступу: <https://www.packettracernetwork.com/internet-of-things/iot-advanced-programming.html>

16. Архітектура та технології IoT. [Електронний ресурс]. Режим доступу: <https://learn.ztu.edu.ua/enrol/index.php?id=1985>

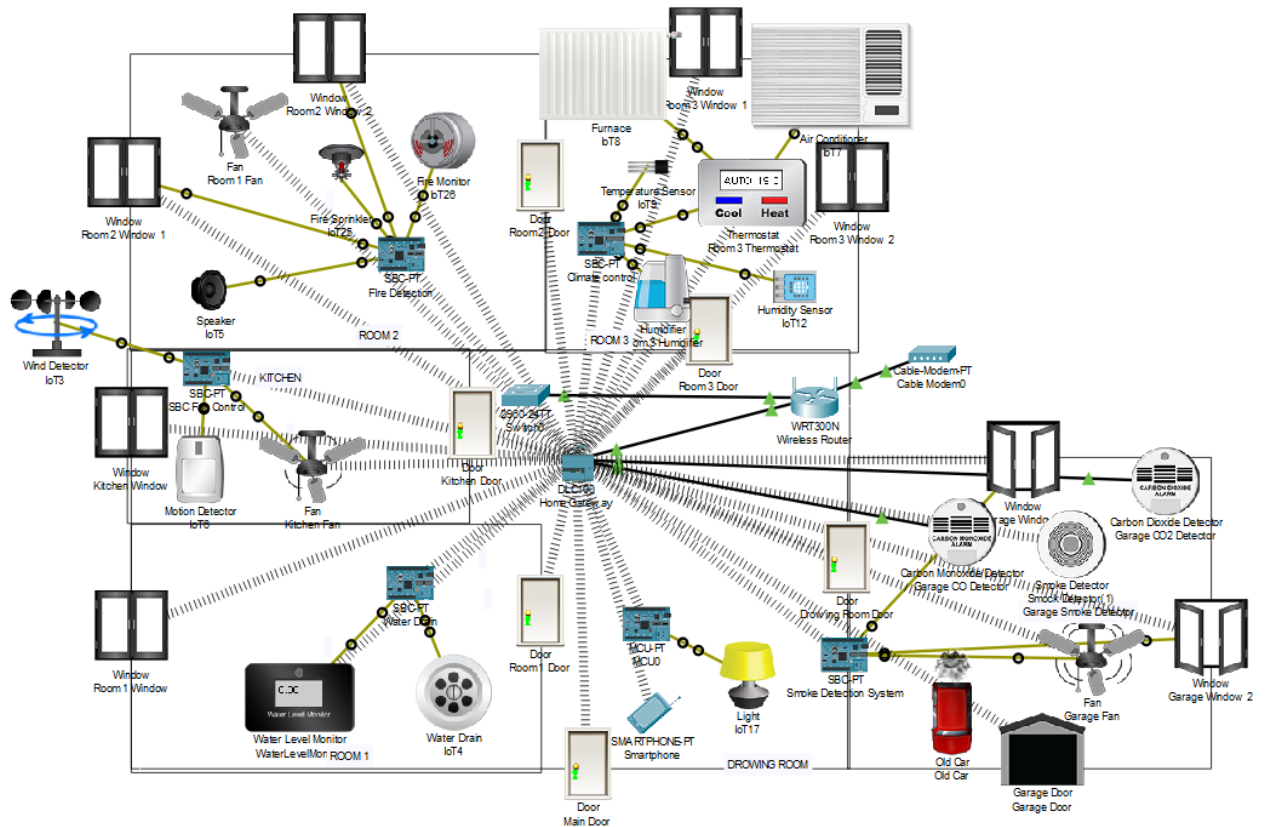
17. Raspberry Pi Documentation. [Електронний ресурс]. Режим доступу: <https://www.raspberrypi.org/documentation>

18. Модуль управління Zipato ZiraTile. [Електронний ресурс]. Режим доступу: <https://www.zipato.com/product/zipatile-zbee>

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

ДОДАТКИ

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		63



		Дядюшкін Р. В.		
		Пілюк І. В.		
Змн.	Арк.	№ докум.	Підпис	Дата

КІ.КР.Б – 123 – 21 – ПЗ

Арк.

64

```

from gpio import *
from time import *
from ioeclient import *
from environment import Environment
from physical import *

```

```
state = '0'
```

```
def main():
```

```
    setup()
```

```
    pinMode(0, INPUT)
```

```
    pinMode(1, OUT)
```

```
    print("Fire Alarm System")
```

```
    while True:
```

```
        fire = digitalRead(0)
```

```
        print("Fire level is " + str(fire))
```

```
        #digitalWrite(1, HIGH)
```

```
        if fire == 1023:
```

```
            customWrite(1, '1')
```

```
            customWrite(2, '1')
```

```
            customWrite(3, '1')
```

```
            digitalWrite(4, HIGH)
```

```
            setState('1')
```

```
        else:
```

```
            customWrite(1, '0')
```

```
            customWrite(2, '0')
```

```
            customWrite(3, '0')
```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

```

        digitalWrite(4, LOW)
        setState('0')
        sleep(1)

def setup():
    IoEClient.setup({
        'type': 'Fire Detection',
        'states': [{
            'name': "Status",
            'type': 'options',
            'options': {
                '0': 'Safe mode',
                '1': 'Fire detected!'
            }
        }]
    })

def setState(fireDetection):
    global state
    state = fireDetection

    customWrite(1, state)
    IoEClient.reportStates(fireDetection)
    setDeviceProperty(getName(), "state", state)

if __name__ == "__main__":
    main()

```

Додаток В

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

```

from gpio import *
from time import *
from ioecclient import *
from physical import *
from environment import *
import math

state = 0

def main():
    setup()
    while True:
        loop()

def setup():
    IoEClient.setup({
        "type": "Window",
        "states": [{
            "name": "On",
            "type": "bool",
            "controllable": True
        }]
    })

    IoEClient.onInputReceive(onInputReceiveDone)
    add_event_detect(0, detect)

    setState(state)

```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

```

def onInputReceiveDone(data):
    processData(data, True)

def detect():
    processData(customRead(0), False)

def loop():
    delay(1000)

def processData(data, bIsRemote):
    if len(data) <= 0 :
        return
    data = data.split(",")
    setState(int(data[0]))

def setState(newState):
    global state
    if newState is 0 :
        digitalWrite(0, LOW)
    else:
        digitalWrite(0, HIGH)

    state = newState
    customWrite(0, state)
    IoEClient.reportStates(state)
    setDeviceProperty(getName(), "state", state)

```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

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

Додаток Г

```
from gpio import *
from time import *
from ioeclient import *
from environment import Environment
from physical import *
```

```
state = '0'
temp = 20
```

```
def main():
```

```
    setup()
    pinMode(0, OUT)
    pinMode(1, INPUT)
    pinMode(3, INPUT)
```

```
    print("Climate control")
```

```
    while True:
        temp = digitalRead(1)
        humidity = analogRead(3)/10
        if temp >= 600:
            customWrite(0, '1')
```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

```

        setState('1', temp/17, humidity)
    elif temp <= 550:
        customWrite(0, '2')
        setState('2', temp/17, humidity)
    delay(500)
    print(temp)

```

```

def setup():
    IoEClient.setup({
        'type': 'Climate control',
        'states': [{
            'name': "Mode",
            'type': 'options',
            'options': {
                '0': 'Off',
                '1': 'Cooling',
                '2': 'Heating',
                '3': 'Auto'
            },
            'controllable': True
        },
        {
            'name': "Temperature",
            'type': "number",
            'unit': "C",
            'imperialUnit': "F",
            'toImperialConversion': "x*1.8+32",
            'toMetricConversion': "(x-32)/1.8",
            'decimalDigits': 1
        }
    ])

```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		70

```

    },
    {
        'name': 'Humidity',
        'type': 'number',
        'unit': '%',
        'decimalDigits': 1
    }
])
})

```

```

def setState(thermMode, temp, humidity):
    #global state
    #state = newState

    IoEClient.reportStates([thermMode, temp, humidity])
    setDeviceProperty(getName(), "state", [thermMode, temp, humidity])

```

```

if __name__ == "__main__":
    main()

```

Додаток Д

```

from gpio import *
from time import *

```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

```

def main():
    pinMode(1, OUT)
    print("Blinking")
    while True:
        digitalWrite(1, HIGH);
        delay(1000)
        digitalWrite(1, LOW);
        delay(500)

if __name__ == "__main__":
    main()

```

Додаток Ж

```

from gpio import *
from options import Options
from time import *
from physical import *
from ioeclient import IoEClient
from environment import Environment

autoMaxTemp = 22
autoMinTemp = 21

METRIC = ""

state = 0  # 0 off, 1 cooling, 2 heating, 3 auto
STATE_NAMES = ["OFF", "COOL", "HEAT", "AUTO"]
temp = 0
unit = ""

```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		72

def setup ():

global autoMaxTemp , autoMinTemp , temp , state

IoEClient.setup({

"type": "Thermostat",

"states": [{

"name": "Status",

"type": "options",

"options": {

0: "Off",

1: "Cooling",

2: "Heating",

3: "Auto"

},

"controllable": True

}, {

"name": "Temperature",

"type": "number",

"unit": "°C",

"imperialUnit": "°F",

"toImperialConversion": "x*1.8+32",

"toMetricConversion": "(x-32)/1.8",

"decimalDigits": 1

}, {

"name": "auto max Temperature",

"type": "number",

"unit": "°C",

"imperialUnit": "°F",

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гулько І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

```

        "toImperialConversion": "x*1.8+32",
        "toMetricConversion": "(x-32)/1.8",
        "decimalDigits": 1,
        "controllable": True,
        "minValue": 10,
        "maxValue": 100
    }, {
        "name": "auto min Temperature",
        "type": "number",
        "unit": "&deg;C",
        "imperialUnit": "&deg;F",
        "toImperialConversion": "x*1.8+32",
        "toMetricConversion": "(x-32)/1.8",
        "decimalDigits": 1,
        "controllable": True,
        "minValue": -100,
        "maxValue": 20
    }]
})

```

```
IoEClient.onInputReceive(onInputReceive)
```

```
add_event_detect(0, onEventDetect_0)
```

```
temp = Environment.get("Ambient Temperature")
```

```
measurementSystemChangeEvent()
```

```
state = restoreProperty("state", 0)
```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

setState(state)

autoMaxTemp = restoreProperty("auto max temp", 20)

autoMinTemp = restoreProperty("auto min temp", 19)

Environment.removeCumulativeContribution("Ambient Temperature")

updateEnvironment()

sendReport()

onEventDetect_0()

def onEventDetect_0 () :

 prepareData(customRead(0), False)

def onInputReceive(input) :

 prepareData(input, True)

def restoreProperty (propertyName, defaultValue):

 value = getDeviceProperty(getName(), propertyName)

 if not (value == "" or value == None) :

 if isinstance(defaultValue, (int, float)) :

 value = int(value)

 setDeviceProperty(getName(), propertyName, value)

 return value

return defaultValue

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

```
def loop ():
    global temp
    temp = Environment.get("Ambient Temperature")
    updateEnvironment()
    measurementSystemChangeEvent()
    sendReport()
    delay(1000)
```

```
def measurementSystemChangeEvent ():
    global METRIC, unit
    METRIC = Options.isUsingMetric()
    unit = "C" if METRIC else "F"
    setCustomText(27, 15, 200, 100,
        STATE_NAMES[state] + " " +
        convertTemperature(temp) + " " + unit)
```

```
def mouseEvent (pressed, x, y, firstPress):
    global state
    if firstPress:
        state+=1
        if state >= 4:
            state = 0

    setState(state)
```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

```

def prepareData (data, bIsRemote):
    global autoMaxTemp, autoMinTemp
    if data:
        data = data.split(",")
        setState(int(data[0]))
        autoMaxTemp = float(data[2])
        autoMinTemp = float(data[3])
        if autoMaxTemp < autoMinTemp:
            autoMaxTemp,autoMinTemp = autoMinTemp,autoMaxTemp

```

```

def setState (newState):

```

```

    global state
    state = newState

```

```

    analogWrite(A1, state)
    setCustomText(27, 15, 200, 100,
        STATE_NAMES[state] + " " +
        convertTemperature(temp) + " " + unit)

```

```

    sendReport()

```

```

def sendReport ():
    report = str(state) + "," + str(temp) + "," + str(autoMaxTemp) + "," +
str(autoMinTemp)
    customWrite(0, report)
    IoEClient.reportStates(report)

```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

```

setDeviceProperty(getName(), "state", state)
setDeviceProperty(getName(), "temperature", temp)
setDeviceProperty(getName(), "auto max temp", autoMaxTemp)
setDeviceProperty(getName(), "auto min temp", autoMinTemp)

```

```
def updateEnvironment ():
```

```

    global autoMode
    autoMaxTemp = float(getDeviceProperty(getName(), "auto max temp"))
    autoMinTemp = float(getDeviceProperty(getName(), "auto min temp"))

```

```

    digitalWrite(1, LOW)
    digitalWrite(2, LOW)

```

```

    if state == 0 :
        digitalWrite(1, LOW)
        digitalWrite(2, LOW)

```

```

    elif state == 1 :
        digitalWrite(2, HIGH)

```

```

    elif state == 2 :
        digitalWrite(1, HIGH)

```

```

    elif state == 3 :
        if temp >= autoMaxTemp:
            digitalWrite(2, HIGH)

```

```

        elif temp < autoMinTemp:

```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

```
digitalWrite(1, HIGH)
```

```
def convertTemperature (value):  
    if METRIC:  
        return "%.0f"%(value,)   
    else:  
        return "%.0f"%(value * 1.8 + 32,)
```

```
if __name__ == "__main__":  
    setup()  
    while True:  
        loop()
```

		Дядюшкін Р. В.			КІ.КР.Б – 123 – 21 – ПЗ	Арк.
		Гуляко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		79