

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно - інформаційних технологій та автоматизації
 (повне найменування інституту, назва факультету)

Автоматика та телекомунікації
 (повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри автоматики та
телекомунікацій

_____ **Валерій ПОЦЕПАСВ**
 (підпис) (ім'я та прізвище)

«__» _____ 2023 р.

Випускна кваліфікаційна робота

_____ **бакалавра**
 (освітньо-кваліфікаційний рівень)

на тему «Розробка автоматичної системи моніторингу та керування
параметрами клімату у теплиці»

Виконав : студент 4 курсу, групи АКТ-19
 (шифр групи)

спеціальності 151 Автоматизація та комп'ютерно-
інтегровані технології
 (шифр і назва напряму підготовки, спеціальності)

_____ **Какоєв Алан Азізович**

(прізвище та ініціали)

(підпис)

Керівник доц. каф. АТ, к.т.н.

(посада, науковий ступінь, вчене звання, прізвище та ім'я)

_____ **Ганна ТЕЛИЧКО**

(підпис)

Рецензент доц. каф. ЕТ, к.т.н., доц.

(посада, науковий ступінь, вчене звання, прізвище та ім'я)

_____ **Ганна ШЕЇНА**

(підпис)

Нормоконтроль:

_____ **ас. Дар'я ЖУКОВСЬКА**

(підпис)

*Засвідчую, що у цій дипломній
роботі немає запозичень з праць
інших авторів без відповідних
посилань.*

Студент _____
 (підпис)

Луцьк – 2023 р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету)

Кафедра автоматики та телекомунікацій
(повна назва кафедри)

Захист відбувся _____
(дата)

з оцінкою _____

Секретар ЕК _____
(підпис)

Випускна кваліфікаційна робота бакалавра

Тема: «Розробка автоматичної системи моніторингу та керування параметрами клімату у теплиці»

Виконавець, студент
 гр. АТК-19

Керівник

Консультанти:

		<u>Алан КАКОЄВ</u>
(підпис,	дата,	Ім'я, ПРІЗВИЩЕ)
		<u>Ганна ТЕЛИЧКО</u>
(підпис,	дата,	Ім'я, ПРІЗВИЩЕ)
		<u>Валерій ПОЦЕПАСВ</u>
(підпис,	дата,	Ім'я ПРІЗВИЩЕ)
		<u>Ганна ТЕЛИЧКО</u>
(підпис,	дата,	Ім'я, ПРІЗВИЩЕ)
		<u>Гліб СТУПАК</u>
(підпис,	дата,	Ім'я, ПРІЗВИЩЕ)

Нормоконтроль

		<u>Дар'я ЖУКОВСЬКА</u>
(підпис,	дата,	Ім'я, ПРІЗВИЩЕ)

Луцьк 2023р.

**Державний вищий навчальний заклад
"Донецький національний технічний університет"**

Інститут, факультет Комп'ютерно-інформаційних технологій та автоматизації
 Кафедра Автоматика та телекомунікації
 Освітньо-кваліфікаційний рівень бакалавр
 Галузі знань 15 Автоматизація та приладобудування
 (шифр і назва)
 Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
 (шифр і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри АТ
Валерій ПОЦЕПАСВ

« » 2023 року

З А В Д А Н Н Я

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Какоєву Алану Азізовичу
 (прізвище, ім'я, по батькові)

1. Тема роботи Розробка автоматичної системи моніторингу та керування параметрами клімату у теплиці

керівник роботи: к.т.н., доцент Теличко Г.О.
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від від 05.05.2023 року № 175

2. Строк подання студентом роботи

3. Вихідні дані до роботи: НДРС, практика

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1) Аналіз існуючих систем автоматичного моніторингу та керування клімату.

2) Проектування системи автоматичного моніторингу та керування.

3) Структурний та параметричний синтез системи автоматичного керування.

4) Заходи з охорони праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень).

1. Схема електрична структурна

2. Схема електрична функціональна

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
	Теличко Г.О., доц. каф. АТ		
	Поцепаєв В.В., в.о. зав. каф. АТ		
	Ступак Г.В., ст. викл. каф. АТ		
нормоконтроль	Жуковська Д.О.	-	22.06.2023р.

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз існуючих систем автоматичного моніторингу та керування клімату	03.05.2023	
2	Проектування системи автоматичного моніторингу та керування	10. 05.2023	
3	Розробка математичної моделі системи автоматичного моніторингу та керування	17. 05.2023	
4	Синтез САК	24.05.2023	
5	Моделювання перехідних процесів в системі автоматичного керування	30.05.2023	
6	Розробка заходів з охорони праці	06.06.2023	
7	Оформлення	15.06.2023	

Студент _____
(підпис)Алан КАКОЄВ
(прізвище та ініціали)Керівник роботи _____
(підпис)Ганна ТЕЛИЧКО
(прізвище та ініціали)

Лист зауважень

Посада П.І.Б.	Суть зауваження, оцінка та підпис

АНОТАЦІЯ

Какоєв Алан Азізович «Розробка автоматичної системи моніторингу та керування параметрами клімату у теплиці» / Випускна кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» зі спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології. – ДВНЗ «ДонНТУ», Луцьк, 2023.

Зміст пояснювальної записки до бакалаврської роботи: 125 сторінок, 56 рисунків, 20 посилань на використані джерела.

Об'єкт розробки – система автоматичного керування параметрами клімату теплиці, як об'єкта керування.

Мета роботи – модернізація системи керування параметрами клімату теплиці, що забезпечує кращі показники якості регулювання, підвищує рівень автоматизації та розширює функціонал системи на основі сучасної елементної бази.

Методи й засоби розробки: методи системного аналізу та декомпозиції, теорія автоматичного керування, комп'ютерне моделювання, програмування в середовищі Node-RED, методи аналізу та синтезу систем теорії автоматичного керування, математичне моделювання.

В даній роботі розроблено систему автоматизації на базі одноплатного комп'ютера Raspberry Pi та веб-додаток з інтерфейсом користувача. Також виконується розробка програмного забезпечення для автоматизації теплиць та віддаленого моніторингу показань датчиків в режимі реального часу з подальшим аналізом та вибором апаратного забезпечення, проведено порівняльний аналіз апаратних засобів, таких як мікроконтролери, Arduino та Raspberry Pi для автоматизації макетів теплиць.

Ключові слова: ТЕПЛИЦЯ, АВТОМАТИЗАЦІЯ, NODE-RED, ПОТОКИ, RASBERRY PI, MODBUS, MQTT, БРОКЕРИ, NODE JS, СИСТЕМА АВТОМАТИЧНОГО КЕРУВАННЯ, ІНДУСТРІЯ 4.0.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ОБ’ЄКТУ АВТОМАТИЗАЦІЇ	12
1.1. Опис об’єкту автоматизації.....	12
1.2 Опис технологічного процесу.....	18
1.3. Огляд автоматизації системи.....	19
1.4 Огляд існуючих програмних рішень для автоматизації	26
1.5 Мета та завдання роботи.....	28
Висновки до розділу 1.....	29
2 ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЧНОГО МОНІТОРИНГУ ТА КЕРУВАННЯ ТЕПЛИЦІ.....	30
2.1 Основні компоненти системи автоматизації системи моніторингу та керування параметрами клімату теплиці.....	30
2.2 Перелік підсистем автоматичного моніторингу та керування параметрами клімату теплиці.....	31
2.3 Додаткові компоненти системи.....	35
2.4 Вибір контролера системи автоматичного моніторингу та керування параметрами клімату теплиці	36
2.5 Вибір датчиків для автоматичного моніторингу та керування параметрами клімату теплиці	38
2.6 Вибір керованих пристроїв для автоматичного моніторингу та керування параметрами клімату теплиці	41
2.7 Вибір мережевих протоколів автоматичного моніторингу та керування параметрами клімату теплиці.....	44
Висновки до розділу 2	46
3 РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА КЕРУВАННЯ ТЕПЛИЦІ..	47
3.1 Структура схеми Node-RED	47
3.2 Функція керування вхідним сигналом автоматичного моніторингу	

та керування параметрами клімату теплиці.....	49
3.3 Інтерфейс користувача інформаційної панелі автоматичного моніторингу та керування параметрами клімату теплиці.....	50
3.4 Протокол MQTT у системі автоматичного моніторингу та керування параметрами клімату теплиці.....	54
3.5 Налаштування протоколу MODBUS у системі автоматичного моніторингу та керування параметрами клімату теплиці.....	59
3.6 Налаштування сервісу Google Sheets у системі автоматичного моніторингу та керування параметрами клімату теплиці.....	63
3.7 Налаштування телеграм-боту системи автоматичного моніторингу та керування параметрами клімату теплиці.....	70
Висновки до розділу 3.....	79
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	82
ДОДАТОК А – Охорона праці та безпека під час надзвичайних ситуацій на підприємстві.....	84
ДОДАТОК Б – Функція «Alarm».....	94
ДОДАТОК В – Технічні характеристики Raspberry Pi 3 model B.....	95
ДОДАТОК Г – Текст програми.....	96

ВСТУП

Темою даної роботи є підвищення рівня автоматизації об'єкта (тепличних господарств) до рівня Індустрії 4.0 промислової революції.

Актуальність роботи. Проект актуальний тим, що автоматизація програмно-апаратного комплексу дозволить скоротити витрати, зробити суттєву економію, а також підвищення якості продукції. Може використовуватися як на виробництві, так і в приватних господарствах.

Об'єкт дослідження – автоматизована теплиця.

Мета роботи – модернізація системи керування параметрами клімату теплиці, що забезпечує кращі показники якості регулювання, підвищує рівень автоматизації та розширює функціонал системи на основі сучасної елементної бази.

Методи й засоби розробки: методи системного аналізу та декомпозиції, теорія автоматичного керування, комп'ютерне моделювання, програмування в середовищі Node-RED, методи аналізу та синтезу систем теорії автоматичного керування, математичне моделювання.

Розробка автоматизованої системи передбачає розробку програмного забезпечення автоматизації теплиць, віддаленого моніторингу показань системи датчиків.

Концептуально підхід до розробки ефективних механізмів забезпечення інноваційності та цифровізації національної економіки в умовах глобалізованої конкуренції полягає в тому, щоб спочатку якісно оцінити поточний стан інновацій, потім сформулювати бачення майбутньої цифровізації національної економіки і, нарешті, розвивати українську промисловість. Формування Індустрії 4.0 є складним, багатогранним та довготривалим процесом, а створення та впровадження ключових технологій є фундаментальним викликом для України сьогодні. Це питання постійно набуває все більшої актуальності та гостроти в публічних виступах українських посадовців. Незважаючи на певні позитивні зрушення в цьому напрямку, вони все ще не є достатньо глибокими,

подекуди фрагментарними та розрізненими, а також не мають системного та комплексного характеру.

Модернізовані системи автоматичного керування можуть замінити існуючі системи при розробці або модернізації нового обладнання для технології автоматизації теплиць.

Наведено детальний огляд використання концепції технології Industry 4.0 у різних галузях промисловості, зокрема найбільше уваги приділяється саме сільському господарству, так як воно є одним із найпопулярніших галузей для впровадження нових рішень з автоматизації управління в рамках цієї концепції.

Крім того, проведено порівняльний аналіз апаратних засобів, таких як мікроконтролери, Arduino та Raspberry Pi для автоматизації макетів теплиць. Узагальнено основні датчики, що використовуються для моніторингу необхідних властивостей, таких як вологість ґрунту, освітленість, вентиляція та температура, розглянуті можливі алгоритми для створення мікроклімату.

Апробація роботи. Основний результат роботи обговорювався на семінарі кафедри автоматики та телекомунікацій. За допомогою апаратного та програмного забезпечення розроблено систему автоматизації на базі одноплатного комп'ютера Raspberry Pi та веб-додаток з інтерфейсом користувача. Використані компоненти, ідеї та додатки можуть бути застосовані у реальних сільськогосподарських чи виробничих проектах.

Структура та обсяг бакалаврської роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Обсяг дослідження викладено на 125 сторінках. В тому числі наявні 56 рисунків, 6 таблиць, 20 посилань на літературні джерела та 4 додатки.

1. АНАЛІЗ ОБ'ЄКТУ АВТОМАТИЗАЦІЇ

1.1 Опис об'єкту автоматизації

Теплиця - це рукотворна споруда, основним призначенням якої є захист рослин від небажаного шкідливого впливу компонентів зовнішньої погоди, наприклад, холодного повітря чи опадів, а також забезпечення бажаного мікроклімату для вирощування обраного виду рослин (декількох видів рослин). Завдяки теплиці створюються ідеальні або наближені до них умови вирощування овочів та фруктів.

Зазвичай використовується в холодні періоди, що небажані для вирощування більшості культур (пізня осінь, зима, та рання весна), для вирощування іноземних, теплолюбних рослин може використовуватися протягом усього року.

Як вже було згадано, теплиці - захисні споруди спеціального призначення для вирощування різних городніх культур та розсади переважно у холодну пору року. Принцип роботи заснований на дії інфрачервоного випромінювання, що випускається сонцем, а завдяки надійній конструкції каркасу та даху тепле повітря, що нагріває ґрунт та рослини, утримується всередині теплиці.

Технології виготовлення та використані матеріали (наприклад, скло, полікарбонат чи плівка) відрізняються, залежно від термінів а пори року використання. Основними критеріями вибору матеріалів є їх прозорість та міцність. Прозорість впливає на пропускання сонячного світла.

За типами теплиці бувають стелажні та земляні, а за конструкцією поділяються на підвісні та блокові. Існують два типи конструкції - каркасні та безкаркасні. Дахи теплиць можуть бути односхилими, двосхилими або багатоповерховими. Найчастішим використовуваним варіантом є двосхилий дах. Такі теплиці зручні та функціональні, особливо взимку, оскільки на поверхні даху не накопичується сніг. Односхилі дахи не використовуються

взимку, оскільки велика кількість снігу блокує сонячне світло, а також може призвести до обвалу конструкції. Для обігріву теплиці використовується природне сонячне світло, у випадку його нестачі можливо застосування теплогенераторів, обігрівачів, парового опалення.

У випадку вирощування деяких культур одночасно, або однієї з відмінним терміном посадки, використовуються ізольовані секції для розсади, щоб процес її розвитку не залежав від основної культури. Необхідна конструкція теплиць включає розсадну секцію з гідропонною установкою для зрошення розсади.

Розсадний відділ як окремий елемент, завданням якого є незалежне підтримання мікроклімату та незалежні системи зрошення забезпечує вирощування розсади у закритому ґрунті та, за необхідності, вирощування зеленої маси в міжсезоння.

На сьогоднішній день використовуваною є технологія дрібносерійного виробництва мінераловатних субстратів за методом крапельного зрошення. Мінеральна вата (скловата) є ідеальним субстратом, має такі властивості, як хімічна інертність та вологоутримуючі властивості її волокон, вона пресується в мати та упаковується в легку герметичну плівку.

Багато людей мають дачі або сади, за якими необхідно доглядати, щоб отримати хороший врожай. Це вимагає значних зусиль і часу, подібних до промислового сектору. Отже, автоматизація процесу вирощування з мінімальною участю людини завжди є актуальною. У таких випадках важливо автоматизувати багато різних процесів, що призводить до створення системи автоматизації (СА). СА складається з апаратних та програмних засобів, які централізовано керують об'єктом у виробничому або контрольованому середовищі. Прикладами автоматизованих систем є виробництво автомобілів, побутової техніки та великого машинобудування.

Розумні теплиці можна класифікувати наступним чином:

- Автономні - всі системи працюють виключно на тепловій або сонячній енергії.

- Енергозалежні - електроенергія подається з електромережі.

Кожен тип має свої переваги та вподобання, щодо яких тривають дискусії.

Автоматизовані теплиці без необхідності участі людини виконують наступні завдання: підтримують необхідну температуру в приміщенні на основі інформації від датчиків; здійснюють провітрювання приміщення, шляхом відкривання вікон; автоматично поливають рослини з урахуванням вологості ґрунту та ін.

Багато автоматизованих систем надають можливість локального керування або взагалі працюють без участі людини, опираючись лише на автоматичні параметри. Автоматизовані системи не лише виконують багато дій автоматично, але й мають доступ до Інтернету та можуть бути керовані дистанційно без присутності людини біля системи. Ці системи відомі як "Інтернет речей".

В даний час термін "Інтернет речей" (IoT) використовується для опису нового етапу, на якому кількість пристроїв, що підключені до Інтернету, перевищує кількість людей. Цей перехід відбувся приблизно між 2008 і 2009 роками, коли кількість підключених пристроїв перевищила кількість населення планети. IoT надає можливості компаніям скоротити витрати на працю та автоматизувати процеси. У сфері енергетики, дистанційний моніторинг за допомогою IoT покращує контроль над підстанціями та електропередавальними лініями. В медицині "розумні" пристрої відстежують показники здоров'я пацієнтів, що сприяє покращенню діагностики захворювань. У сільському господарстві "розумні" ферми та теплиці здатні автоматично контролювати кількість добрив та води. В галузі логістики, розвиток IoT допомагає знизити витрати на вантажоперевезення та мінімізувати вплив людського фактору. Крім того, IoT активно впроваджується в нафтогазовому та гірничодобувному секторах, де він дозволяє збільшити видобуток на вже виснажених родовищах завдяки детальному аналізу під час буріння свердловин. У транспорті, IoT

включає в себе кузови транспортних засобів, навігатори, електронні дисплеї, камери спостереження та системи безпеки.

Розвиток Інтернету речей став можливим завдяки поширенню Інтернету, смартфонів, бездротових мереж, дешевих електронних компонентів та обробки даних. У практичному застосуванні системи IoT зазвичай складаються з мережі розумних пристроїв, які підключені до хмарної платформи. Вони взаємодіють з системами зберігання, обробки та захисту даних, що були зібрані датчиками.

IoT включає в себе різноманітні технології, такі як розробка датчиків та протоколів для їх взаємодії. Об'єкти можуть комунікувати один з одним за допомогою різних методів, таких як Wi-Fi, Bluetooth, LPWAN, BLE, Ethernet, RFID та ZigBee. Технологія M2M (машина-машина) є конкретним прикладом використання Інтернету речей.

Архітектура Інтернету речей передбачає наявність функціональних рівнів, таких як сенсорні мережі, шлюзи та управління.

Поняття "спеціалізованих" протоколів стають необхідними у зв'язку з наявністю датчиків, сенсорів і реле на нижніх рівнях. Ці компоненти мають взаємодіяти один з одним і з більш високими рівнями, тому потрібні протоколи, які будуть відповідати умовам Інтернету речей. Стандартні прикладні протоколи не підходять для цієї мети. Датчики зазвичай мають невеликі розміри і обмежену пам'ять, але вони вимірюють фізичні параметри в реальному часі, споживаючи мало енергії. Результати їх вимірювань обробляються сенсорним вузлом і передаються на сервер. Хоча кількість інформації, яку генерує один сенсорний вузол, невелика, багато послуг Інтернету речей базуються на обробці інформації з кількох вузлів, таких як абонент-комунікаційні вузли для телефонії, клієнт-серверні взаємодії для передачі даних та інші. Це відрізняється від архітектури класичних мереж.

У Інтернеті речей джерелом даних служать різноманітні датчики, які можна підключити до мікроконтролера для збору та передачі даних для подальшого аналізу. Зазвичай створюються проекти систем автоматичного

управління на основі мікроконтролерів – спеціальних мікросхем, які використовуються для управління різними електронними пристроями.

Мікроконтролер об'єднує в собі процесор, пам'ять, периферійні схеми та програмне забезпечення на одній мікросхемі, яка виглядає як звичайна мікросхема. Кількість виробництва мікроконтролерів щорічно в кілька разів перевищує виробництво процесорів, оскільки вони широко використовуються вбудованих системах, побутовій техніці та домашній автоматизації. У таких випадках баланс між ціною і функціональністю є важливішим, ніж продуктивність.

Факт того, що майже будь-який розумний пристрій, від дитячої іграшки до бездротової гарнітури мобільного телефону, може містити мікроконтролер, є результатом широкого спектру застосувань у сучасному суспільстві.

Мікроконтролери отримують команди з пам'яті або вхідних портів і виконують їх. Кожна команда має певне значення, яке визначається системою команд. Архітектура мікроконтролера включає систему, в якій виконання коду команди представлене виконанням певних мікрооперацій внутрішніми елементами мікросхеми.

Зазвичай, мікроконтролери мають пам'ять розміром від 2 до 128 Кб, іноді більше. Для їх програмування використовують спеціальні мови, такі як Basic, Pascal або, найчастіше, C. Перед остаточним програмуванням мікроконтролер проходить тестування в програмному або апаратному емуляторі. Варто зауважити, що мікроконтролери не є просто процесорами, але також не є повноцінними комп'ютерами.

Мікроконтролери відрізняються наявністю вбудованого завантажувача. Завдяки цьому завантажувачу користувач може завантажити свою програму на мікроконтролер без використання окремого апаратного програматора. Для підключення до комп'ютера завантажувач зазвичай використовує USB-інтерфейс (якщо він доступний на платі) або адаптер UART-USB. Безкоштовне середовище розробки Arduino IDE має вбудовану підтримку завантажувача і дозволяє запустити процес завантаження лише одним клацанням миші.

Програмісту доступні різноманітні спеціальні функції портів вводу-виводу мікроконтролера.

Крім портів, можуть бути встановлені додаткові пристрої, такі як USB-інтерфейси або Ethernet. Також існують розширювальні модулі з додатковим набором зовнішніх периферійних пристроїв, таких як USB-пристрої (зазвичай у вигляді віртуальних COM-портів через FTDI FT232, деякі з емуляцією клавіатури та миші класу USB HID), комунікаційні модулі (дротові та бездротові) на материнських платах та платах розширення, GSM-модулі, модулі для підключення SD-карт, модулі керування двигунами, реле, електромагнітами та інші пристрої, такі як графічні РК-дисплеї та модулі з модельним полем.

Мікроконтролер може бути підключений до різноманітних датчиків та виконавчих механізмів від сторонніх виробників, таких як гіроскопи, компаси, термометри, релейні модулі та клавіатури.

Мікроконтролери зазвичай підключаються до дротової або бездротової мережі. У системі автоматичного управління програмне забезпечення відіграє важливу роль. Воно використовується для керування, зберігання, аналізу та відображення інформації. Для цих цілей часто використовуються веб-сервіси, які працюють на веб-серверах.

Веб-сервер - це сервер, який отримує запити від клієнтів і обробляє їх. Він може бути окремим комп'ютером або робочою станцією з встановленим сервісним програмним забезпеченням. Зазвичай клієнтом є веб-браузер, а веб-сервер надсилає HTTP-відповіді, які містять HTML-сторінки, зображення, медіапотоки та інші дані. HTTP є мережевим протоколом передачі даних, який забезпечує взаємодію між мережею та користувачем.

Веб-сервери виконують різні функції, такі як організація запитів до мережевих сервісів, баз даних, доступ до файлів, надсилання електронної пошти та інші додатки електронної комерції. Вони також підтримують виконання скриптів, написаних на мовах програмування, таких як PHP і Python.

Для правильної роботи необхідна база даних та файли з кодом сторінок. База даних є набором даних, пов'язаних з однорідними властивостями або характеристиками. За допомогою спеціалізованих алгоритмів зберігання та пошуку інформації, які використовуються в базах даних, можна швидко отримувати необхідну інформацію.

Для взаємодії з базою даних використовується система управління базами даних (СУБД). Це програмне забезпечення, яке використовується для створення та роботи з базою даних.

"Розумна" теплиця на базі мікроконтролера і централізованої системи управління - це проект з автоматизації процесу вирощування рослин. Він використовує мікроконтролер та централізовану систему управління для контролю та керування умовами в теплиці з метою покращення врожайності та ефективності.

1.2. Опис технологічного процесу

Розглянемо технологічний процес на прикладі вирощування томатів та огірків у захищеному ґрунті.

Ґрунт повинен відповідати певним вимогам: висока пористість (65-75%), вологоємність 45-50%, повітроємність 20-25% і щільність 0,4-0,6 г/см². До ґрунту додають компоненти з підвищеною пористістю і водопроникністю. Як правило, в тепличних ґрунтах використовують такі співвідношення компонентів:

- Легкі піщані або супіщані ґрунти - 20... 30 % за об'ємом
- Торф - 50...50 % за об'ємом. 60 % за об'ємом
- Компост - 20... 20 % за об'ємом. 30 % за об'ємом.

Перед посівом проводять захисні заходи з дезінфекції теплиці та насіння. Насіння обробляють біопрепаратами для знезараження від бактерій та грибків. Під час пікіровки коріння вкорочують на третину; через 18-20 днів розсаду

сортують; на м² розміщують 20-28 сіянців; потім розсаду розміщують у теплиці так, щоб на м² було не менше 20-28 сіянців. Висаджують розсаду на постійне місце, вертикально, не засипаючи стебла.

Теплиці є складними, нелінійними системами, і їхня поведінка змінюється з часом через різні збурення, багато з яких пов'язані з погодними умовами. Актуальною є потреба в більш складних та інтелектуальних способах об'єднання внутрішніх і зовнішніх зацікавлених сторін, таких як електроенергія, вода і CO₂, для забезпечення необхідних змін у тепличному середовищі для здорового розвитку культур і, в кінцевому рахунку, прибуткової роботи.

Ідеальна температура для росту рослин - 17-27°C. Для рослин, які потребують обігріву, рекомендується середня нічна температура 15-18,5°C і відносна вологість 50-80%; у листопаді, грудні та січні необхідно щонайменше 500-550 годин сонячного світла.

Фермери в усьому світі використовують зрошувальні системи, споживаючи велику кількість води з різних джерел, і це є основною проблемою. Тому необхідно приділяти більше уваги підвищенню ефективності використання води у зрошуваному землеробстві. Технології зрошення можна класифікувати як традиційні або сучасні, залежно від їхньої здатності зберігати воду. Традиційне поверхнєве зрошення наносить і розподіляє воду на поверхню ґрунту під дією сили тяжіння без використання датчиків або засобів контролю.

1.3. Огляд автоматизації системи

Система автоматичного управління мікрокліматом теплиці має наступні функції:

- Контроль температури та вологості в теплиці;
- Управління освітленням;
- Контроль опалення та вентиляції;

- Автоматична вентиляція;
- Контроль параметрів метеостанції;
- Управління положенням затінюючих екранів;
- Полив рослин.

Система вентиляції теплиці призначена для природного обміну повітря в закритому об'ємі з зовнішнім повітрям через вентиляційні отвори в даховій частині світлопрозорої конструкції. Отвори управляються автоматично і дистанційно. Отвори кріпляться до труб за допомогою алюмінієвих хомутів і направляються за допомогою алюмінієвих консольних рейок з двома пластиковими роликами на кожному даху. Вентиляція забезпечується через вентиляційні отвори, розташовані в шаховому порядку вздовж плит перекриття. Загальна площа вентиляційних отворів у даху теплиці становить 25%. Кут відкриття вентиляційних отворів становить 46°.

Основна конфігурація системи автоматичного управління теплицею:

- Метеостанція;
- Блоки датчиків;
- Додаткові блоки датчиків;
- Блок управління;
- Обладнання візуалізації та архівації.

Світлоповертаючі та світловідбиваючі системи завіс спеціально розроблені для затінення теплиць навесні і влітку, коли сонячна радіація висока, а також для зменшення тепловтрат в нічний час і в періоди низьких зовнішніх температур.

Системи крапельного зрошення гарантують, що вода і поживні розчини потрапляють безпосередньо до коренів рослини, забезпечуючи постійну вологість кореневої системи. У цій системі вода подається до коренів кожної рослини індивідуально. Кількість води, що подається кожній рослині, можна індивідуально регулювати за допомогою крапельного дозатора. Основна перевага систем крапельного поливу полягає в тому, що рослини можна поливати цілодобово без втручання людини. При поливі за допомогою системи

крапельного зрошення не відбувається ущільнення верхнього шару ґрунту на поверхні ґрунту, чого неможливо уникнути при поливі рослин іншими способами.

Контролер теплиці вирішує завдання автоматичного регулювання температури двох опалювальних контурів, управління циркуляційним насосом і приводом фрамуги, а також включення і виключення освітлення. У теплиці застосована подвійна система управління, один з терморегуляторів встановлений на підлозі, а інший - під заслінкою даху.

На рис. 1.1 наведено систему автоматизації теплиці за технологією Industry 3.0

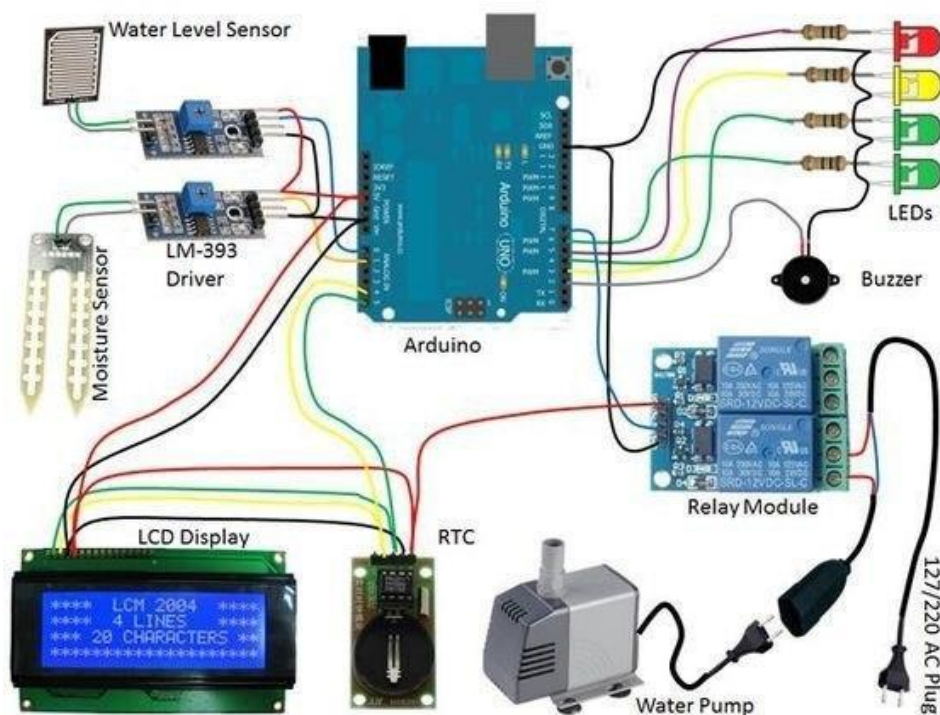


Рисунок 1.1 – Система автоматизації теплиці Industry 3.0

Для контролю проблемних зон (в кутах будинку або біля стін) будинок може бути обладнаний багатofункціональними пристроями, які вимірюють наступні параметри:

- Відносна вологість;
- Температура, T;

- Точка роси, T_d ;
- Абсолютна вологість, d_v ;
- Коефіцієнт перемішування, g ;
- Швидкість повітряного потоку, v ;
- Швидкість повітряного потоку, v ;
- Атмосферний тиск, p ;
- Вуглекислий газ.

Автоматичний режим роботи забезпечується контролером з датчиками та електронною схемою управління, яка забезпечує роботу приводів у необхідному режимі.

Стандарт визначає наступні фази життєвого циклу:

- Концепція;
- Розробка;
- Виробництво;
- Застосування;
- Підтримка впровадження;
- Кінець використання та амортизація.

Для систем, що розглядаються, фаза "впровадження" відноситься до системи, що використовується в даний час, тоді як фаза "розробка" відноситься до системи, описаної в цій роботі.

Автоматизація теплиць передбачає управління та моніторинг регульованих параметрів клімату.

Автоматизація підтримки мікроклімату сприяє кращому росту і підвищенню врожайності, а також зменшує витрати ручної праці. У технічних процесах необхідний високий ступінь автоматизації та механізації. Система управління може бути виражена як взаємозв'язок різних процесів і об'єктів управління. Загальною метою автоматизації управління є підвищення ефективності використання потенціалу об'єкта управління.

Існує три типи систем управління мікрокліматом.

- Ручне управління. До них відносяться візуальний контроль росту рослин, ручний полив, увімкнення та вимкнення терморегуляторів, ручне внесення добрив та пестицидів. Це менш точні та ненадійні системи, оскільки вони вимагають багато часу та мають високу ймовірність людської помилки.

- Частково автоматизовані. Це поєднання ручного управління і часткової автоматизації, багато в чому схоже на системи з ручним управлінням, але з меншими витратами, пов'язаними зі зрошенням і контролем параметрів.

- Повністю автоматизовані. Це складні установки, які можуть впоратися з більшістю кліматичних змін, що відбуваються в теплиці. Ці системи засновані на принципі зворотного зв'язку і можуть ефективно реагувати на зовнішні подразники.

Деякі з проблем, пов'язаних з вищезгаданими системами, такі, як:

- Важко відстежувати зміни кліматичних параметрів, таких як вологість, вологість ґрунту, освітленість, рН ґрунту і температура, які прямо чи опосередковано впливають на ріст рослин

- Високі витрати на обслуговування та потреба у кваліфікованому технічному персоналі. Сучасне обладнання використовує мобільні технології як систему зв'язку та бездротового збору даних, забезпечуючи глобальний доступ до інформації на фермі. Однак це має ряд обмежень, таких як складність конструкції, складність в ремонті та висока ціна.

- Більшість комерційних проектів автоматизації теплиць розроблені для тепличних комплексів площею в кілька гектарів, що залишає неосвоєним ринок автоматизації теплиць для фермерських та індивідуальних господарств.

Розумні теплиці - це повністю або майже повністю автоматизовані системи, що забезпечують контроль мікроклімату.

Сучасні тепличні комплекси будуються за стандартизованими проектами і оснащуються необхідними інженерними системами (опалення, зрошення, вентиляція і циркуляція повітря, дренаж і водопостачання, освітлення) для підтримки мікроклімату. Всі ці системи призначені для великих підприємств. Їх

монтаж та експлуатація складні та дорогі. Ці системи не застосовуються для індивідуальних фермерів або невеликих фермерських господарств.

Енергозберігаючі системи автоматичного управління - це системи з алгоритмами, спрямованими на управління технічними процесами з меншими витратами енергії, ніж раніше. Найефективніша АСУ ТП - це та, яка гарантує мінімальне споживання енергії:

$$Q = Q_t + Q_p \quad (1.1)$$

де Q_t - енергоспоживання технологічного процесу, а Q_p - втрати енергії через дефекти технології, обладнання або матеріалів.

У випадку теплиць, Q_t умовно приймається як витрати на компенсацію втрат тепла з поверхні теплиці в навколишнє середовище. Втрати енергії через недосконалість технології включають втрати, пов'язані з відкриттям дверей і воріт теплиць. Умова часто доповнюється обмеженнями, що характеризують діапазон зміни факторних величин, і обмеженнями на мінімально допустиму температуру, при якій погіршується ріст рослин. Для зменшення енергоемності процесу необхідно докласти зусиль для зменшення технічних витрат енергії Q_t . Ця величина залежить від розміру і теплових характеристик установки, швидкості повітряного потоку і різниці між температурою повітря і температурою навколишнього середовища. Чим нижча температура повітря в приміщенні, тим менше Q_t . Однак, оскільки нижча температура в приміщенні призводить до зниження продуктивності, зниження обмежується сільськогосподарськими вимогами. Для кожної культури експериментально визначено температуру, при якій продуктивність знижується. Тому завданням системи автоматичного керування є стабілізація температури на певному рівні. Але стабілізація температури і вологості в теплиці - не найкраще рішення проблеми. Насправді, в реальному житті розвиток рослин відбувається між змінами мікрокліматичних параметрів. Наприклад, вночі температура зазвичай

нижча, ніж вдень, а навесні та восени нижча, ніж влітку. Організми пристосувалися до цих змін протягом своєї довгої еволюційної історії. Тому, залежно від часу доби і стадії розвитку рослини, температура (і вологість), необхідна для її розвитку, повинна змінюватися. Відповідно має змінюватися і завдання. Системи, які реалізують це завдання, називаються програмними ESAU. Третій тип ESAU для контролю температури в теплицях, технічно спрямований на зниження корисного споживання енергії.

Короткочасні зниження і підвищення температури і вологості в приміщенні не змінюють продуктивність рослин. Тому кілька короткочасних відключень опалення протягом дня можна розглядати як своєрідне біологічне загартовування, що також може призвести до певної економії енергії. Допустимий час вимкнення обігріву для кожної культури потрібно визначати експериментально. Різновидом є так званий переривчастий режим, при якому потужність обігріву зменшується на прийнятний проміжок часу замість того, щоб повністю припиняти обігрів. Втрати енергії можна зменшити, покращивши динаміку керування, тобто зменшивши максимальне динамічне відхилення та час перехідних процесів. Цього можна досягти, перейшовши від системи регулювання за відхиленням до комбінованої системи. У цій системі керування відбувається за двома незалежними каналами: зміною об'єму гарячої води та температури води. При переході на таку систему управління знижується температура зворотної води і зменшується кількість циркуляційних насосів, що зменшує споживання енергії на роботу циркуляційних насосів. Для зменшення споживання енергії на одиницю продукції важливо підвищувати продуктивність та ефективність. Однак слід зазначити, що максимізація ефективності не завжди є економічно виправданою. Тому на додаток до систем керування, які гарантують оптимізацію за продуктивністю, доцільно використовувати системи керування, які гарантують оптимізацію за попередньо обраними технічними, економічними, екологічними та енергетичними критеріями. Технічний процес, який має найбільший вплив на будь-який з цих критеріїв, називається оптимальним. Цей процес характеризується певним

набором параметрів. Відхилення від раціонального режиму призводить до незворотних втрат продукції, енергії та грошей. Для забезпечення оптимальної технології необхідно організувати оптимальне управління. Завдання полягає в тому, щоб створити набір умов, які забезпечують оптимальне значення критеріїв ефективності. До таких критеріїв відносяться критерій приведених витрат, критерій споживання енергії та критерій чистого прибутку.

Після того, як необхідні системи встановлені і "розумна" теплиця запущена в експлуатацію, можна розглянути можливість її подальшого вдосконалення.

Можна розглянути наступні варіанти:

- Система подачі води в резервуари для води з насосами і поплавковими вимикачами
- Додаткове освітлення для збільшення світлового дня або затінення (наприклад, при вирощуванні редиски).
- Розширення системи вентиляції додаванням електричних вентиляторів.
- Система опалення з термостатом.

Модернізація системи керування параметрами клімату теплиці, що забезпечує кращі показники якості регулювання, підвищує рівень автоматизації та розширює функціонал системи на основі сучасної елементної бази.

1.4 Огляд існуючих програмних рішень для автоматизації

Існує кілька основних інструментів для розробки систем автоматизації з відкритим кодом. Залежно від платформи існують платні та безкоштовні рішення. Це можуть бути ПЛІС (програмовані логічні інтегральні схеми), контролери форм-фактора роутера і програмне забезпечення.

Eclipse IoT представляє собою набір відкритих технологій, створений промисловими компаніями у співпраці з науковими організаціями. За допомогою цього набору інструментів розробники IoT здійснюють створення

фреймворків, проводять розрахунки в межах обмежень та автоматизують процес виробництва.

IBM Watson - це хмарна платформа штучного інтелекту, яка використовується для ідентифікації, агрегації та трансформації даних з пристроїв IoT з метою прогнозування та зменшення ризиків. Платформа також надає ряд додаткових сервісів, таких як можливість підключення пристроїв через MQTT та HTTPS, розпізнавання неструктурованого тексту, обробка графіків та відео, а також архівування даних для подальшої аналітики на інших платформах або інтеграції з IoT-додатками.

Інші безкоштовні варіанти включають OpenHUB і Home Assistant; OpenHUB здається дуже завантаженим і важко розширюється. Наприклад, важко додати новий модуль Z-Wave, якщо його немає в OpenHUB; Home Assistant легко розширювати, але під час інтеграції складно підтримувати; інтеграція з Z-Wave в Home Assistant викликала питання. Тому був обраний інший безкоштовний аналог.

Інструмент програмування з відкритим вихідним кодом Node-RED написаний на JavaScript і графічно показує, як пристрої взаємодіють один з одним, так що виконуваний код можна генерувати через браузер. Схема містить блоки датчиків і приводів, а також функціональні блоки. Кожен з них має власну панель редагування властивостей, а деякі панелі також містять вікна для написання функцій JavaScript. Вузли сценарної моделі з'єднані лініями таким чином, що дані, надіслані з датчиків, проходять через функціональні блоки і потрапляють на входи актуаторів після необхідної модифікації. Передача інформації між вузлами сценарію відбувається за допомогою JavaScript-об'єкта msg.

Для розробки автоматичної системи моніторингу та керування параметрами клімату у теплиці був обраний Node-RED. Це готове рішення, яке пропонує багато інтеграцій з різними протоколами, є безкоштовним і дозволяє писати логіку автоматизації на JavaScript прямо в браузері. Також легко налагоджувати скрипти за допомогою Debug та Inject. Кожен потік в системі є

автономним, і вся комунікація між різними потоками здійснюється через шину MQTT. Також є можливість розподіляти завантажені потоки на різні сервери для обробки.

Для розширення функціональності інструменту та додавання блоків до стандартної палітри системи доступна низка бібліотек. Наприклад, він може працювати з виводами GPIO різних мікроконтролерів для зв'язку з фізичними пристроями. Наведені блоки датчиків та виконавчих механізмів вказують на номер порту, до якого підключено пристрій. Для бездротових датчиків блоки можуть використовувати відкритий протокол MQTT, який працює поверх стеку протоколів TCP/IP; за допомогою протоколу MQTT повідомлення передаються між і брокером повідомлень координатора, встановленим на сервері. Видавець надсилає повідомлення брокеру із зазначенням теми, а брокер надсилає повідомлення клієнтам, які підписані на цю тему. При використанні цього методу адреси брокера і теми вказуються у вхідних і вихідних блоках Node-RED при моделюванні сценарію.

1.5 Мета та завдання роботи

Мета роботи – модернізація системи керування параметрами клімату теплиці, що забезпечує кращі показники якості регулювання, підвищує рівень автоматизації та розширює функціонал системи на основі сучасної елементної бази.

Для досягнення поставленої мети необхідно було виконати наступні завдання:

- 1) Виконати аналіз системи моніторингу та керування параметрами клімату теплиці;
- 2) Розробити функціональну схему системи керування моніторингу та керування параметрами клімату теплиці та обрати обладнання для її реалізації;

3) Розробити алгоритм та програму керування параметрами клімату теплиці.

Висновки до розділу 1

1) Виконано аналітичний огляд тепличних господарств, як об'єктів автоматизації.

2) Проведено опис технологічного процесу, рівня автоматизації системи, огляд існуючих програмних рішень для автоматизації

3) Враховуючи дану оглядову інформацію, було встановлено, що для створення автоматизованого комплексу слід враховувати такі фактори:

- Використання (відповідність) технологіям Industry 4.0, за допомогою даного способу досягається максимальна автоматизація процесу.

- Для керування системою використовується система на основі мікроконтролера.

- Система автоматичного управління мікрокліматом теплиці має виконувати наступні функції:

- Контроль температури;
- Управління освітленням;
- Автоматична вентиляція;
- Полив рослин.

2. ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЧНОГО МОНІТОРИНГУ ТА КЕРУВАННЯ ТЕПЛИЦІ

2.1 Основні компоненти системи автоматизації системи моніторингу та керування параметрами клімату теплиці

Автоматизація системи моніторингу та керування параметрами клімату теплиці вимагає процесу управління для створення та підтримки необхідних умов середовища:

- Температурні режими (регулювання температури приміщення);
- Іригація та полив (контроль вологості);
- Освітлення (регулювання інтенсивності світла);
- Управління концентрацією CO₂;
- Датчики та виконавчі механізми використовуються для зчитування даних і зміни стану навколишнього середовища.

Основна роль належить контролеру, підключеному до системи дистанційного керування. Кожен пристрій в робототехнічному комплексі виконує певну функцію. З таких систем складається розумна теплиця

- Вентиляція. Ці пристрої регулюють повітрообмін між внутрішнім і зовнішнім середовищем. При надходженні сигналу ззовні крокові двигуни, пневматичні або гідравлічні пристрої переміщують вікна в потрібне положення. Відповідні сигнали генерують датчики температури або вітру. Якщо температура всередині теплиці занадто низька, включається функція обігріву. Оптимальна температура досягається за допомогою нагрівальних приладів, таких як термостати, нагрівальні елементи, інтенсивність яких залежить від команди датчика температури;

- Освітлення. До складу системи входять лампа і датчик освітлення, основною частиною якого є фоторезистор. Опір змінюється в залежності від

інтенсивності світлового потоку для формування керуючого сигналу. Крім ламп, рівень освітленості можна регулювати за допомогою автоматичних штор;

- Контроль рівня CO₂. Відповідні датчики підключаються до вентиляторів і використовуються для очищення приміщення від кисню, що виробляється рослинами. Забезпечення рослин вуглекислим газом підвищує продуктивність на 30 відсотків;

- Зрошення. Полив автоматизований за допомогою датчиків вологості (гігрометрів). Для економії в комплект входить датчик витрати води.

Автоматично підтримується оптимальна температура повітря.

При вирощуванні помідорів і огірків, типових домашніх культур, температура повинна бути від +18 до +25°C вдень і +16°C і нижче вночі. Крім того, бажана температура ґрунту +10°C і вище.

Коли температура підвищується, її можна знизити, використовуючи приводи для відкриття вентиляційних отворів у теплиці. Даний привід повинен бути оснащений датчиком температури, а також датчиком вітру, щоб уникнути пошкодження рослин. Для цього можна використовувати крокові двигуни, які відкривають вікна для провітрювання на потрібний кут.

2.2 Перелік підсистем автоматичного моніторингу та керування параметрами клімату теплиці

Розглянемо підсистему поливу рослин та вологості клімату. Автоматичний полив створюється за допомогою датчиків вологості, які обмежують полив, але прості і дешеві заземлені датчики швидко окислюються і виходять з ладу, тому в комплексі додатково використовуються датчики витрати води. Після того, як відома необхідна потреба у воді (яка залежить від площі теплиці, потреби рослин у воді у воді тощо), достатньо використовувати датчик витрати води для поливу за часом і датчик вологості для переливу, тривоги.

Важливою частиною автоматичного моніторингу та керування параметрами клімату теплиці є підсистема управління освітленістю. Автоматичне освітлення найпростіше реалізувати за допомогою фоторезистора. При зниженні рівня освітленості величина опору зростає і формується керуючий сигнал на включення світла в теплиці.

Особливості підсистеми підігріву: автоматична тепла підлога, як і повітряне опалення, використовує для регулювання температури замість приводів нагрівальні елементи та нагрівальні кабелі.

Автоматизація є основною основою для максимального контролю та моніторингу всіх важливих і критичних процесів у сучасних проектах теплого садівництва. Автоматизація дозволяє контролювати та контролювати процеси з будь-якої точки світу. Таким чином, автоматизація теплиць є відповіддю на зростаючий попит на повний контроль процесів у режимі реального часу.

Блоки керування в системах автоматизації використовують пристрої, які підтримують і аналізують інформацію від датчиків і видають керуючі сигнали на виконавчі механізми, нагрівальні елементи, водяні клапани і т.д.

Arduino — це мікропрограмний інструмент із попередньо встановленим завантажувачем, який дозволяє завантажувати програми на мікроконтролер без необхідності використання окремого програматора. Мікроконтролер на ньому запрограмований на мові Arduino, яка заснована на мові Wiring (C-like).

Результати всієї автоматизації теплиць можна візуально контролювати на комп'ютері. Веб-інтерфейс дозволяє користувачеві контролювати та керувати показаннями датчиків температури, вологості та освітленості. Також за теплицею можна спостерігати через веб-камеру.

Система управління теплицею керується центральною платою Arduino і працює наступним чином. Дані про навколишнє середовище, отримані датчиками температури, вологості та освітлення, надсилаються на центральний контролер, який порівнює поточні значення з заданими. Якщо якесь із значень

не збігається, активуються приводи для відновлення оптимальних умов. Потім Arduino надсилає дані на віддалений сервер для моніторингу через Інтернет.

Контроль наступних параметрів здійснюється спеціальним програмним блоком

- Опалення теплиці
- Частота і тривалість поливу
- Запуск та зупинка примусової вентиляції
- Освітлення.

Контроль температури вище двох порогів: верхнього і нижнього. При перевищенні верхньої межі відкриваються жалюзі і вмикаються вентилятори для охолодження повітря в теплиці (можна регулювати за допомогою штор); температура повітря нагрівається нижче нижньої межі, вентилятори зупиняються і вмикаються обігрівачі, щоб нагріти повітря до заданого рівня.

Контроль вологості за допомогою заданого користувачем порогу. Якщо вологість всередині теплиці падає нижче встановленого порогу, включається система автоматичного поливу, яка вимикається при відновленні оптимальних умов[18].

Умови роботи також контролюються двома параметрами. Верхня межа встановлюється при включеному світлі, а нижня - при вимкненому. Це в основному використовується для збільшення світлового дня або для компенсації недостатнього природного освітлення через погодні умови.

Розглянемо конструкцію системи попереднього рівня автоматизації Industry 3.0. На рис. 2.1 зображено структурну схему, на рис. 2.2 функціональну.

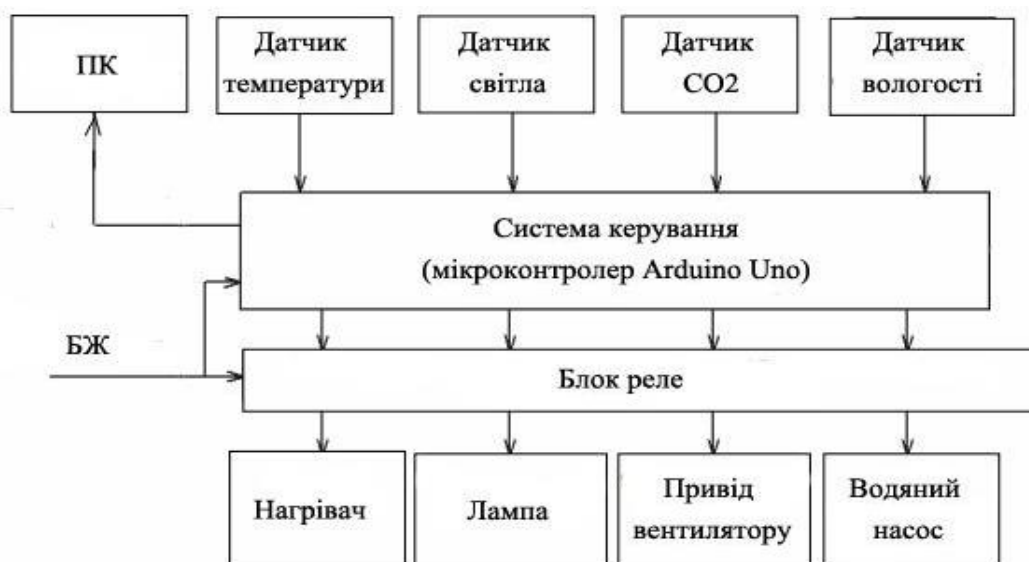


Рисунок 2.1 – Структурна схема автоматизації автоматичного моніторингу та керування параметрами клімату теплиці

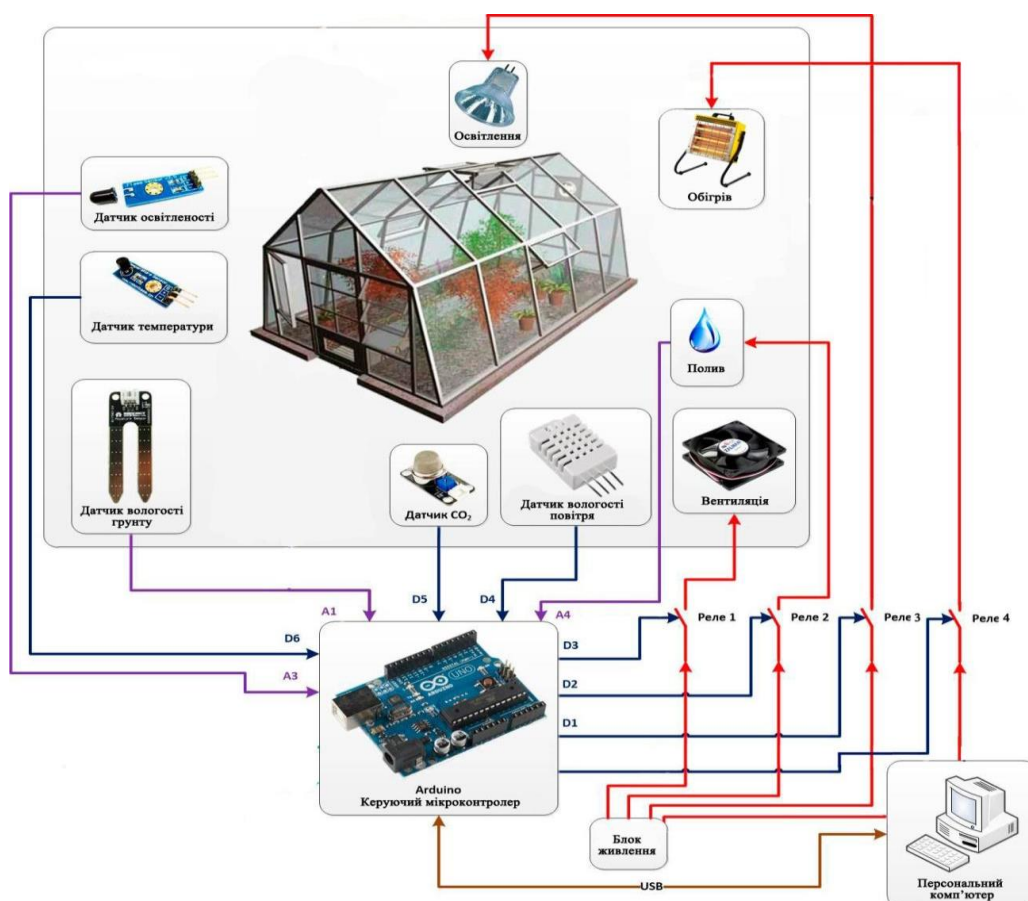


Рисунок 2.2 – Функціональна схема автоматичного моніторингу та керування параметрами клімату теплиці

2.3 Додаткові компоненти системи

Її можна використовувати на стороні Node-RED Edge або як хмарну програму; при використанні як концентратор, шлюз або маршрутизатор на стороні Node-RED використовується для збирання промислових продуктів на апаратній платформі Raspberry PI. Дані збираються з різних пристроїв за допомогою протоколів промислової мережі. В даний час найпоширенішим і найпростішим протоколом є Modbus. Таким чином, Raspberry PI взаємодіятиме з пристроями через Modbus TCP/IP, а з іншого боку — з хмарними програмами та сервісами.

Node-RED також може взаємодіяти з пристроями через протоколи промислової мережі, з одного боку, та з хмарними програмами, сервісами та іншими пристроями через MQTT, з іншого.

Node-RED також можна інтегрувати з хмарними сервісами та Telegram. Це дає можливість передавати дані з RaspberryPI у хмару для аналізу чи взаємодії з віддаленими користувачами.

Google Sheet – це хмарна програма Google для роботи з електронними таблицями. За функціоналом та принципом роботи він аналогічний Microsoft Excel. Усі створені таблиці зберігаються на Google Диску.

Боти Telegram забезпечують онлайн-зв'язок між користувачем та IRP.

2.4 Вибір контролера системи автоматичного моніторингу та керування параметрами клімату теплиці

Сфера міні-комп'ютерів пропонує багато моделей від різних виробників. Щоб визначити правильний міні-комп'ютер для проекту, було зроблено вибір із моделей різних виробників в одному сегменті. Порівняльна характеристика наведена у таблиці 2.1.

Таблиця 2.1. Характеристики мінікомп'ютерів

Ім'я мініПК	Raspberry Pi 3 model B	Orange Pi PC	Banana Pi M2
Процесор	64-бітний чотириядерний ARMv8 Cortex-A53, 1.2 ГГц	Чотирьохядерний 32-бітний процесор ARM Cortex-A7 1.2-1.6 ГГц	ARM Cortex-A7 dual core 1.0 ГГц
Графічний процесор	Двоядерний процесор (GPU) 400 МГц здатний кодувати, декодувати і виводити Full HD-відео	Mali400 MP2 600MHz	ARM Mali400 MP2
Сховище	слот для картки пам'яті MicroSDHC, USB Boot Mode	microSD до 64 ГБ (у якості постійною пам'яті і завантажувального диска)	слот для картки пам'яті MicroSDHC,
Ethernet	10/100 Мбіт Ethernet RJ45	10/100 Mb/s Ethernet RJ45	2× USB 2.0, 1x USB OTG
Wi-Fi/Bluetooth	Wi-Fi 802.11n та Bluetooth 4.1	ні	Wi-Fi 802.11n і Bluetooth 4.1
Відео Вхід	1 x CSI-2 для підключення камери	HDMI з підтримкою HDCP, HDMI CEC, HDMI 30 функцій	10/100/1000 Ethernet. Конектор RJ45
Відео вихід	1 x HDMI вихід, 1 x відеовихід 3.5 мм роз'єм 1 x DSI (Display Serial Interface) для підключення штатного дисплея;	1 x HDMI вихід, 1 x відеовихід 3.5 мм роз'єм	1 x HDMI вихід, 1 x відеовихід 3.5 мм роз'єм
USB-порти	4 порту USB 2.0	2 порту USB 2.0	4 порту USB 2.0
Периферія	40 портів введення-виведення спільного призначення (GPIO)	40 портів введення-висновку спільного призначення (GPIO)	40 портів введення-висновку спільного призначення (GPIO)
живлення	5 В, 2-2.5 А через порт micro-USB або GPIO	5 В 2 А	5 В 2 А
Підтримувані ОС	Ubuntu, Debian, Fedora	Android, Lubuntu, Debian, Raspbian	Android, Lubuntu

Вузли, що відповідають за збір і передачу даних і діють як веб-сервер, були обрані з ряду одноплатних мінікомп'ютерів, достатньо потужних для цього проекту.

Одним із найпопулярніших мікрокомп'ютерів, які використовуються сьогодні, є Raspberry Pi. Контролер Raspberry Pi можна використовувати як концентратор для розумних теплиць, а також в якості моста, який дозволяє різним електричним пристроям і обладнанню спілкуватися один з одним.

Було обрано та створено модель Raspberry Pi 3. На рисунку 2.1 показано Raspberry Pi 3 Model B з маркованими елементами на платі.

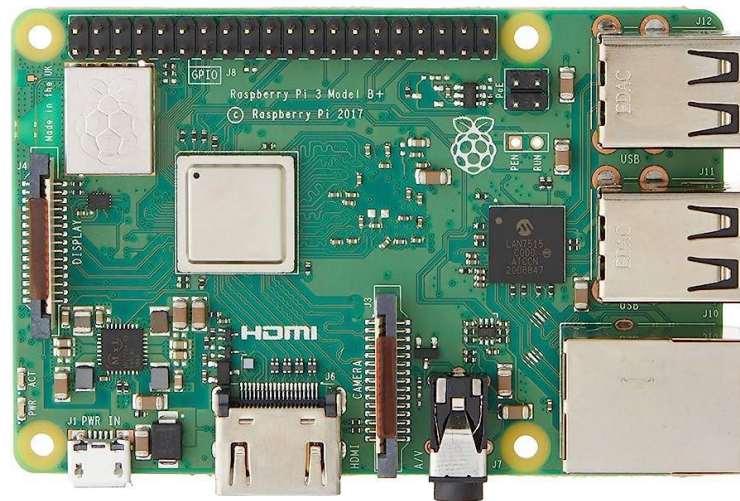


Рисунок 2.1 – Raspberry Pi 3

Ця модель має відмінні технічні характеристики, функціональність та достатню обчислювальну потужність для покращення майбутніх проектів, підключення додаткових модулів та одночасної участі в інших проектах.

Raspberry Pi 3 Model B живиться від 5-вольтового адаптера через роз'єм micro USB або контакти живлення GPIO.

У теплиці необхідно точно вимірювати як мінімальну температуру, і вологість. Для цього можливо використати електронний датчик, який може вимірювати декілька параметрів.

2.5 Вибір датчиків для автоматичного моніторингу та керування параметрами клімату теплиці

Для вимірювання температури та вологості повітря необхідно підібрати датчики, що відповідають характеристикам, що використовуються в даній роботі. Порівняння характеристик вибраних датчиків наведено в табл. 2.2.

Таблиця 2.2. Порівняння датчиків

Характеристики	DHT11	DHT22	DHT21
Напруга живлення, В	3-5	3-5	3-5
Температура, мін °C	0 ± 2	$-40 \pm 0,5$	$-40 \pm 0,5$
Температура, макс °C	50 ± 2	$125 \pm 0,5$	$80 \pm 0,5$
Вологість, мін %	20	0	0
Вологість, макс %	80	100	100

Для вимірювання температури та вологості було використано датчик DHT22, який працює в діапазоні напруги живлення від 3 до 5 вольт. Цей датчик має ширший діапазон вимірювання температури та вологості, а також меншу похибку порівняно з DHT11. Крім того, DHT22 має компактніші розміри в порівнянні з DHT21. Зображення обраного датчика наведено на рис. 2.4.



Рисунок 2.4 – Датчик DHT22

Одним з основних параметрів, який необхідно контролювати для визначення зрошення, є вологість ґрунту. Для цього потрібно підібрати відповідний датчик – гігрометр. Його властивості наведені в табл. 2.3.

Таблиця 2.3 Порівняльна характеристика датчиків

Найменування Характеристики	FC-28	YL-69	ICB510-01
Напруга живлення, В	3.3, 5	3.3 - 5	5 В, 12-24 В
Вихід	цифровий і аналоговий	цифровий і аналоговий	аналоговий
Споживаний струм, мА	35	35	50
Розмір, мм	30×60	20×60	50×208

Вибраний датчик YL-69 можна використовувати з аналоговим і цифровим входами контролера. Даний датчик зображено на рис. 2.4.



Рисунок 2.4 – Датчик вологості ґрунту YL-69

В одному з режимів роботи датчик з'єднаний з платою керування потенціометром (компаратором), що відповідає за налаштування порога спрацьовування при підключенні датчика до цифрового контакту.

Далі необхідно визначити час, що робиться за допомогою фоторезистора. Хоча ці фоторезистори виробляються різними фірмами, але характеристики у них плюс-мінус похибка однакові, тому порівнювати їх немає сенсу. Тому випускається фоторезистор з компаратором, який корисний для встановлення порогового значення, і підключається до цифрових висновків контролера, тому що потрібно лише визначати день і ніч. Фоторезистор – це тип датчика, електричний опір якого змінюється в залежності від інтенсивності світла, що падає на нього. Фоторезистор із компаратором показаний на рис. 2.5.

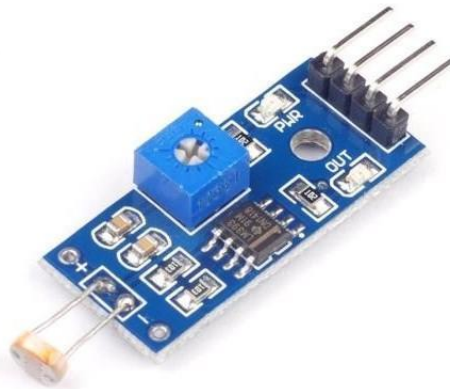


Рисунок 2.5 – Фоторезистор з компаратором

2.6 Вибір керованих пристроїв для автоматичного моніторингу та керування параметрами клімату теплиці

Щоб знизити температуру всередині теплиці, необхідно гаряче повітря випустити назовні, для цього використовуються сервоприводи. У табл. 2.4 порівнюються характеристики трьох моделей сервоприводів.

В якості сервоприводу для проекту було обрано SG-90 (рис. 2.6). Він легкий і має таку ж робочу напругу 5 вольт, як і більшість компонентів проекту. Має достатній крутний момент і невеликий розмір. Цього буде достатньо для реалізації необхідних функцій.

Таблиця 2.4 Порівняльні характеристики сервоприводів

Найменування Характеристики	SG-90	Turnigy TSS- 10MG	ASMC-04B
Напруга живлення	5В	5В	12 В
Кут повороту ротора, градуси	360	360	300
Крутний момент	1.6 кг/см	2.2 кг/см	90 кг/см
Внутрішній інтерфейс	аналоговий	цифровий	аналоговий
Матеріал шестірня	нейлон	метал	метал
Матеріал корпуси	пластик	пластик	метал
Габарити	23.2 × 12.5 × 22 мм	23.1 x12.0 x24.9 мм	60*95*102 мм
Вага	9 г	15 г	530 г



Рисунок 2.6 – Сервопривід SG90

Вирішено встановити в теплиці датчики руху для освітлення входу і внутрішнього простору теплиці в темний час доби. Це вимагає вибору датчика руху, обраного із трьох моделей модулів, перелічених у табл. 2.5.

Таблиця 2.5. Порівняльні характеристики датчиків руху

Найменування Характеристики	E18-D80NK	HC-SR501	HC-SR505
Напруга живлення, В	5	4.5 - 20	4.5 - 20
Кут зони виявлення, градусів	< 15	< 140	< 100
Робоча температура, С	-25 ... +55	-20...+80	-20...+80
Дальність виявлення, м	0.03 – 0.8	3-7м	3
Розміри плати, мм	45x17	32x24x18	10x23

Для поливу також потрібен насос, але так як ми будуємо модель теплиці, використовується невеликий за розміром і потужністю AD20P. Для контролю силового навантаження необхідний такий пристрій, як реле.

Реле - це комутаційні пристрої, які з'єднують і роз'єднують електричні та електронні кола при зміні значення вхідного струму. У табл. 2.6 наведені порівняльні характеристики реле.

Таблиця 2.6 Порівняльні характеристики реле

Найменування Характеристики	NCR HHG1- 1/032F- 38-100A	NCR HHG1- 1/032F- 38-30A	SRD-05VDC-SL- C
Тип	твердотільне	твердотільне	електромагнітне
Керівна напруга, В	3...32	3...32	5
Комутована змінна напруга, В	40...440	40...440	125...250
Максимальний струм навантаження, А	100	30	10

Електромагнітне реле SRD-05VDC-SL-C призначене для комутації силового навантаження. Поставляється у вигляді модуля для зручної та простої установки та легкої заміни навантаження. Крім самого реле, модуль містить оптоелектронний зв'язок із транзистором, що захищає висновки мікроконтролера від коливань напруги на котушці. На рис. 2.7 показаний приклад чотириканального релейного модуля на базі SRD-05VDC-SL-C.



Рисунок 2.7 – Релейний модуль

До цього модуля можна буде підключати необхідне навантаження: освітлення, обігрівач, насос для поливу і інше.

2.7 Вибір мережевих протоколів автоматичного моніторингу та керування параметрами клімату теплиці

Поява Інтернету речей відкрила широкі можливості для збору, аналізу та поширення даних, які перетворюються на інформацію та знання, що можуть бути застосовані для вирішення різних проблем людей. Вибір протоколу для забезпечення зв'язку між пристроями є важливим фактором у цьому процесі. Існує безліч протоколів для використання в різних завданнях, наприклад CoAP, MQTT, Modbus.

MQTT (Message Queuing) Telemetry Transport) – це асинхронний протокол передачі повідомлень за принципом видавець/передплатник з

використанням брокера в мережах із низькою пропускнуою здатністю. Особливістю протоколу є його компактність, так як величина заголовка пакету даних складає всього лише кілька байт. Працює поверх TCP/IP, який забезпечує простий та надійний потік даних. Основною метою MQTT є віддалений моніторинг даних, що збираються з великої кількості пристроїв, і їх телеметрія в IT- інфраструктурі. Протокол націлений на велику мережу невеликих пристроїв, які необхідно контролювати або керувати з хмари. Також він призначений для багатоадресної передачі даних для багатьох приймачів[19]. MQTT широко використовується в системах типу ІоТ (Промисловий Інтернет речей).

Протокол CoAP часто порівнюють з MQTT при розробці систем ІоТ. Хоча вони схожі, між ними є важлива відмінність: MQTT - це протокол "багато-до-багатьох", тоді як CoAP - це в першу чергу протокол "один-до-одного". Проте, CoAP надає метадані, можливості виявлення контенту та узгодження, яких, в той же час, немає в MQTT.

Modbus – це відкритий комунікаційний протокол для обміну даними між мережевими пристроями, заснований на архітектурі ведучий-відомий (master-slave) Перевагами протоколу Modbus є відкритість і масовість використання (безліч виробників випускають різні типи пристроїв, датчиків, підтримуючих даний протокол). Завданнями протоколу Modbus є контроль[20] та управління мережами, читання та запис даних у реєстри зберігання, доступ до файлів, діагностика стану пристроїв.

Для реалізації передачі між сервером і клієнтом вирішено вибрати протокол MQTT[19]. Протокол зроблений з урахуванням суворих умов експлуатації, та ідеальний для пристроїв з невеликою обчислювальною потужністю і обмеженим часом автономної роботи. До їх числа відносяться і смартфони, і постійно зростаюча кількість датчиків і підключених пристроїв і ін.

Однією з головних особливостей протоколу є те, що ініціаторами підключення виступають клієнти, це може виявитися корисним, коли

необхідно збирати дані із різних територіально розподілених джерел. При цьому брокер-сервер має білу ір-адресу, що дозволяє використовувати будь-який хостинг, а клієнти, які підключаються до брокеру, мають просто вихід в Інтернет.

MQTT має наступні особливості:

- Асинхронний протокол;
- Робота в умовах нестабільною зв'язку на лінії передачі даних;
- Ідеально підходить для розподілених комунікацій "один до багатьох» і роз'єднаних додатків;
- Нейтральний до вмісту повідомлення;
- Належить на TCP/IP для базових завдань зв'язку;
- Оснащений функцією LWT (Last Will and Testament, «остання воля і заповіт») для повідомлення сторін про аномальному відключенні клієнта;
- Підтримка кількох рівнів якості обслуговування (QoS): «максимум один раз», «мінімум один раз» і «рівно один раз».

Ці моменти роблять MQTT протоколом вибору для передачі даних між пристроями з обмеженою потужністю процесора і часом автономної роботи, а також в мережах з вузькою смугою пропускання, непередбачуваною стабільністю або високою затримкою. Тому MQTT відомий як ідеальний транспорт для IoT: MQTT побудований на протоколі TCP/IP, але існують MQTT SN для Bluetooth, UDP, ZigBee та інших мереж IoT, що не використовують протокол TCP/IP.

Висновки до розділу 2

1) Складено та зроблено опис модернізованої структурної та функціональної схем АСУ ТП автоматизації теплиці.

2) Відповідно до функціональної схеми обрано технічні засоби застосування.

3. РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА КЕРУВАННЯ ТЕПЛИЦІ

3.1 Структура схеми Node-RED

У попередньому розділі було описано огляд і налаштування апаратного забезпечення автоматичного моніторингу та керування параметрами клімату теплиці. В даній частині детально розглянуто потік node-red, включаючи конфігурацію інформаційної панелі користувача.

Node-RED - інструмент програмування для об'єднання апаратних пристроїв, API та онлайн-послуг.

Редактор функціонує на основі браузера, програма виглядає як об'єднані в потоки (flow) вузли (node), що вибираються з палітри[13].

Програма створена на Node-RED складається з потоків, які виконуються як умовно незалежні програми[12].

Система автоматичного моніторингу та керування параметрами клімату теплиці складається з чотирьох схем, схожих за конструкцією та виконуваними функціями. Вона виконує наступні функції:

- контроль температури в теплиці, включення вентиляції чи підігріву;
- контроль освітленості, автоматичне освітлення чи затінення;
- контроль вологості, керування поливом;
- контроль концентрації CO₂, автоматичне провітрювання.

На рис. 3.1 зображено загальний вигляд потоку спроектованої системи у конфігураторі Node-red.

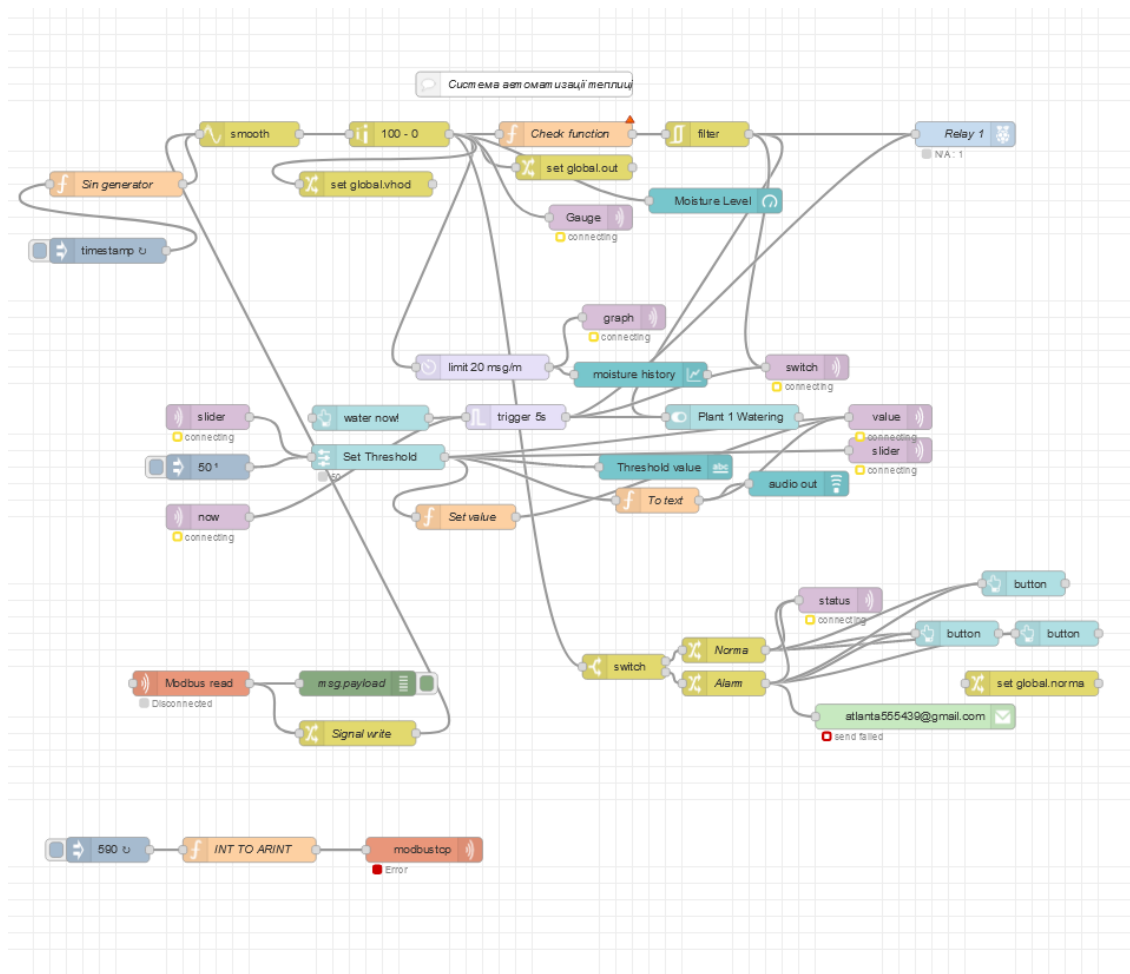


Рисунок 3.1 – Загальний вигляд потоку спроектованої системи для налаштування поливу теплиці в Node-RED

Для зручності опрацювання схема розділена на декілька блоків, а саме:

- Блок функції (БФ). Виконує перевірку відповідності вхідного значення заданому, при перебільшенні виконуються певні дії, в даному випадку відключається подача живлення на контакт, що відповідає за реле. Дана дія рівноцінна відключенню живлення водяного насосу.
- Блок панелі керування (БПК). Відповідає за відображення елементів на панелі керування Dashboard, яка також може виконуватися дистанційно по HTTP чи MQTT. Крім відображення значень з потоку, можливе також завдання деяких параметрів.

- Блок сигналізації (БС). Відбувається перевірка відповідності параметрів заданим, при перевищенні спрацьовує сигналізація. **Поріг спрацьовування може бути заданим вручну чи автоматично.**
- Блок входу (БВ). Відповідає за прийом та передачу даних по протоколу Modbus.
- Блок Telegam-боту (БТ). Виконує інформування користувача **про можливі виконувані команди, має основну функцію додаткової сигналізації, а також відображенні клавіатури, відображення і встановлення завдань.**
- Блок індикації (БІ). Генерує на виході контролера імпульси заданої частоти **для індикації роботи пристрою.**
- База даних (БД). Запис плинних значень у базу даних для тимчасового спостереження.
- Блок відправки сигналу (БВС). **Виконує тестову відправку сигналу для перевірки роботи протоколу Modbus.**
- Блок генерації сигналу (БГС). **Виконує генерацію сигналу для перевірки роботи системи за відсутності доступу до Modbus.**

3.2 Функція керування вхідним сигналом автоматичного моніторингу та керування параметрами клімату теплиці.

Ця частина стосується отримання даних, базової обробки та масштабування, а також керування релейним виходом для керування клапаном поливу (рис. 3.2).

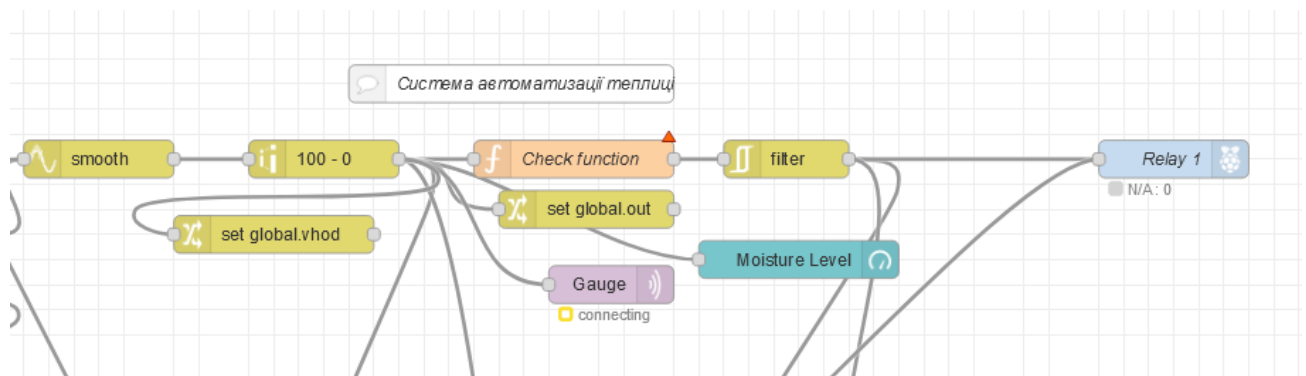


Рисунок 3.2 – Блок функції керування вхідним сигналом

Вихідні дані датчика проходять через вузол, який фільтрує шум, а потім через вузол діапазону, який перетворює значення вологості в зручніший діапазон. На підставі експериментів ми бачимо, що датчик видає приблизно 600, коли сухий, і 400, коли вставляється у вологий ґрунт. Нижче наведено вузол діапазону, який показує шкалу від 0 до 100 для збільшення рівня вологості.

Після цього функціональний вузол «перевірити порогове значення» порівнює поточне значення вологості з глобально збереженою пороговою змінною, і якщо вологість нижче цього значення, надсилає «1», щоб увімкнути реле, або «0», щоб вимкнути його.

Можна помітити два входи, що входять у вихідне реле. Другий сигнал надходить від інтерфейсу користувача (пояснено в наступному підрозділі), який дозволяє вручну запускати з інформаційної панелі.

3.3 Інтерфейс користувача інформаційної панелі автоматичного моніторингу та керування параметрами клімату теплиці.

Компоненти приладової панелі представляють інтерфейс користувача (оператора теплиці) для відображення даних і керування системою. Інформаційна панель наведена на рис. 3.3.

На початку рівень надсилається до вимірювального вузла («Рівень вологості»), а також швидкість обмежується (тут до 20 повідомлень на хвилину).

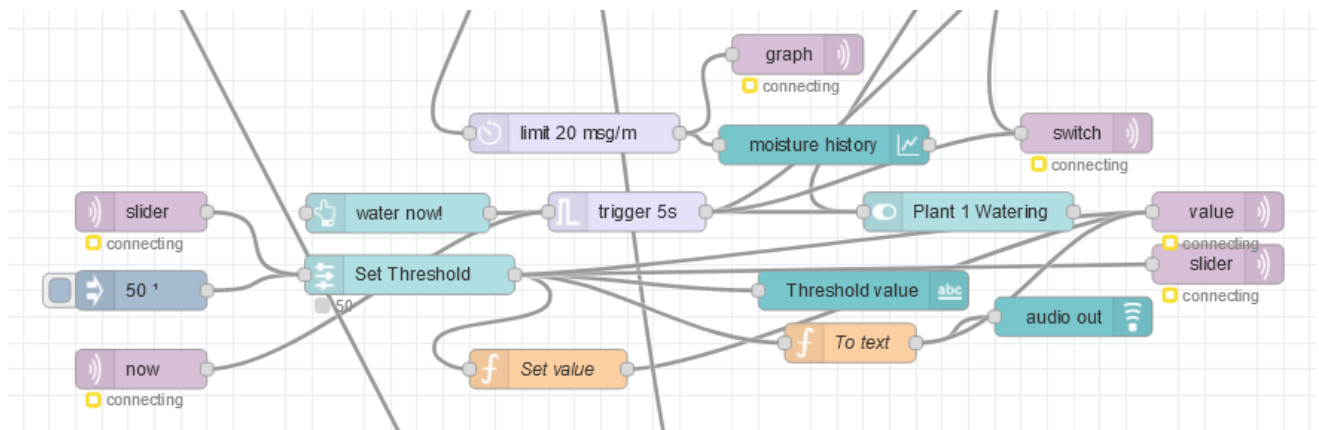


Рисунок 3.3 – Блоки інформаційної панелі (панель керування теплиці)

Вузол повзунка «Встановити поріг» дозволяє регулювати поріг поливу, що оновлює глобальну змінну у функціональному вузлі. Встановлене порогове значення також відображається в текстовому полі.

На рис. 3.4 зображено загальний вигляд панелі керування параметрами теплиці.

Індикатор сигналізації дозволяє спостерігати перевищення рівня вологості в приміщенні.

Вузол “Audio out” виконує оповіщення про сигнал та задані параметри.

В нижній частині панелі розташований графік у вигляді діаграми для наочного спостереження значень параметрів з датчиків.

«Полив зараз» — це вузол кнопки, який активує 5-секундний тригер, який вмикає реле, а також виводить стан реле (вузол перемикача «Полив рослин 1»). Цей статус також керується автоматизованим керуванням.

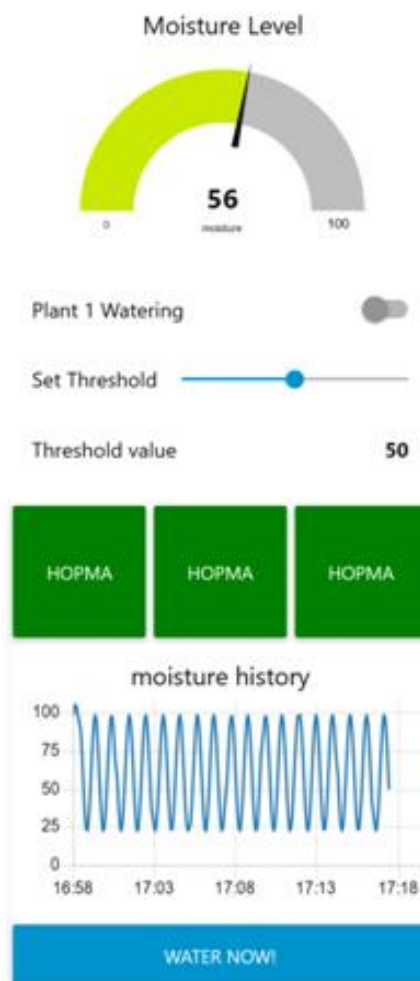


Рисунок 3.4 – Інформаційна панель у Dashboard (панель дистанційного керування теплицею)

3.4 Протокол MQTT у системі автоматичного моніторингу та керування параметрами клімату теплиці

Node-RED використовується на Edge-стороні в якості програмного забезпечення для концентратора, шлюзу або маршрутизатора, зокрема на апаратній платформі Raspberry PI. Воно здійснює збір даних з різних пристроїв за допомогою протоколів промислових мереж. Таким чином, Raspberry PI взаємодіє з пристроями по протоколам промислових мереж, а також взаємодіє з хмарними застосунками, сервісами та іншими пристроями

за допомогою MQTT. У даній частині роботи розглядається Node-RED з точки зору взаємодії з іншими пристроями по протоколу MQTT[12].

Налаштування протоколу MQTT виконано за допомогою вузлів MQTT In розділу палітри Input та їх налаштувань. Брокер MQTT має назву mosquitto. На рис. 3.5 зображено налаштування блоків протоколу MQTT.

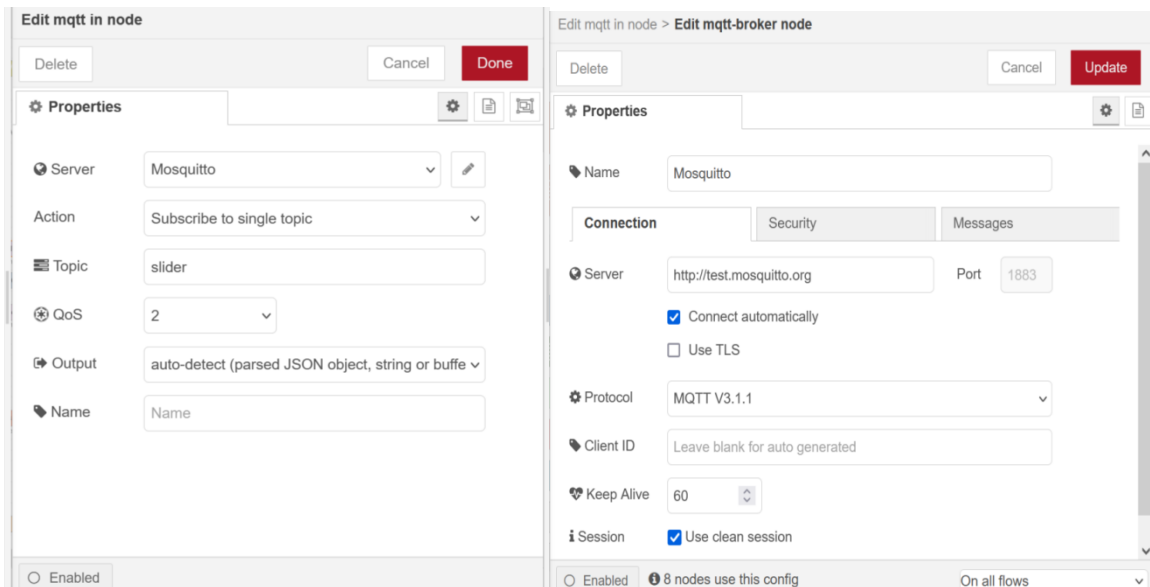


Рисунок 3.5 – Налаштування отримання даних та серверу MQTT.

У полі Topic вузла «MQTT in» доданий «slider». Це означає, що цей вузол підписується на цю тему з даної гілки. При розгортанні вузол «MQTT in» показує статус «Connected» (рис. 3.6).

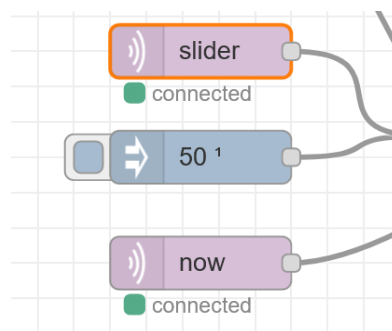


Рисунок 3.6 – Блоки для отримання даних (1,3) через протокол MQTT.

За допомогою вузла "MQTT out" було встановлено з'єднання між локальним графічним інтерфейсом та віртуальним пристроєм на тестовому сервері. Давайте розглянемо приклад з індикатором, розташованим у

верхній частині панелі. Для відображення підписів використовується блок "Gauge". Програма та зовнішній інтерфейс мають такий зовнішній вигляд. (рис.3.7, рис. 3.8):

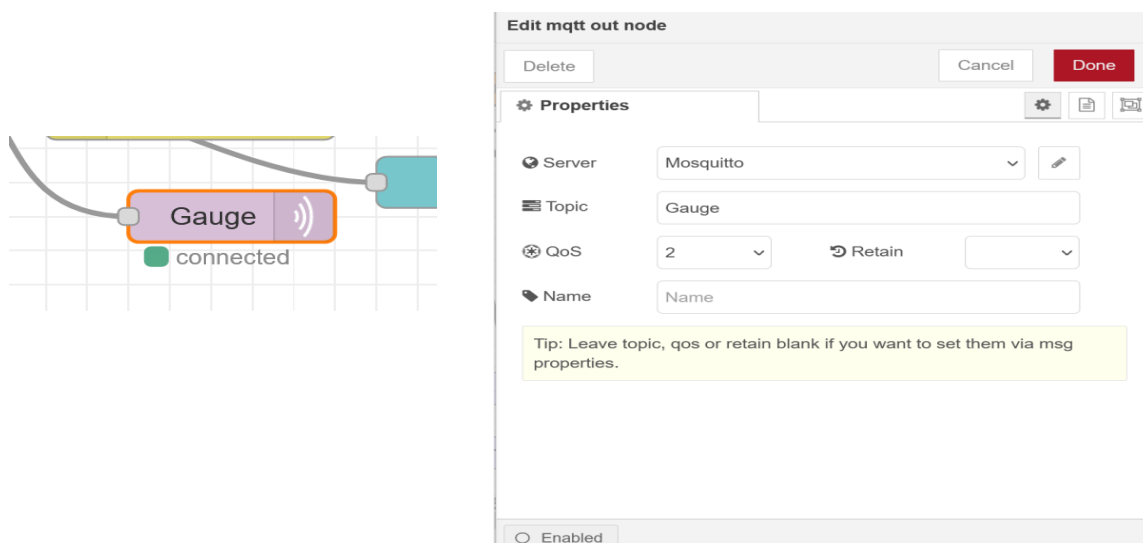


Рисунок 3.7 – Вигляд фрагменту програми Node-RED для візуального відображення даних та його налаштування

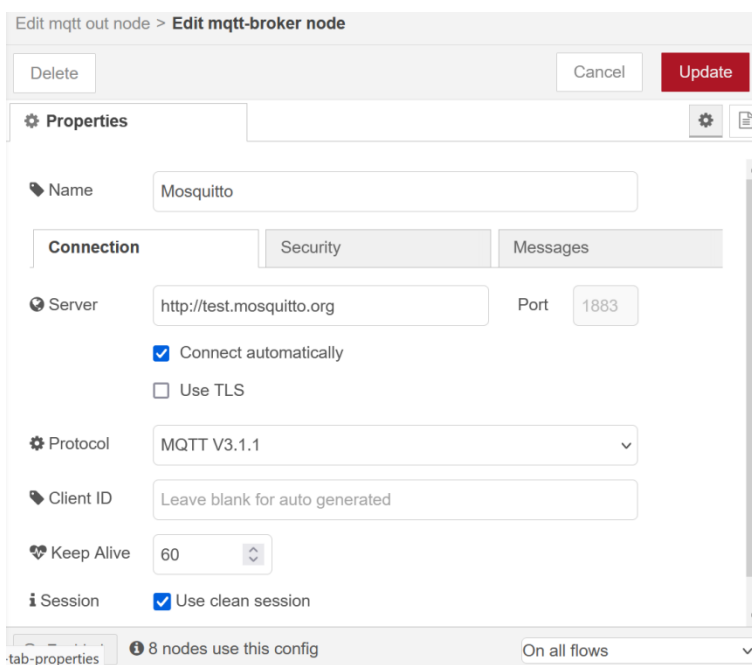


Рисунок 3.8 – Налаштування MQTT-брокеру

Для віддаленого моніторингу та дистанційного керування параметрами теплиці використовується пристрій з ОС Android. Цей пристрій

використовуватиметься як мобільний клієнт MQTT. Для цього було використано Android-додаток MQTT Client "IoT MQTT Panel", де було налаштовано підключення до MQTT брокера (MQTT Host). У розділі "Connection" застосунку "IoT MQTT Panel" використовуються ті ж самі налаштування, що й у попередніх кроках. Крім того, потрібно також додати пристрій (Device). Шляхом додавання графічних елементів і їх тем було створено інтерфейс користувача.

Налаштування зазначені на рис.3.9.

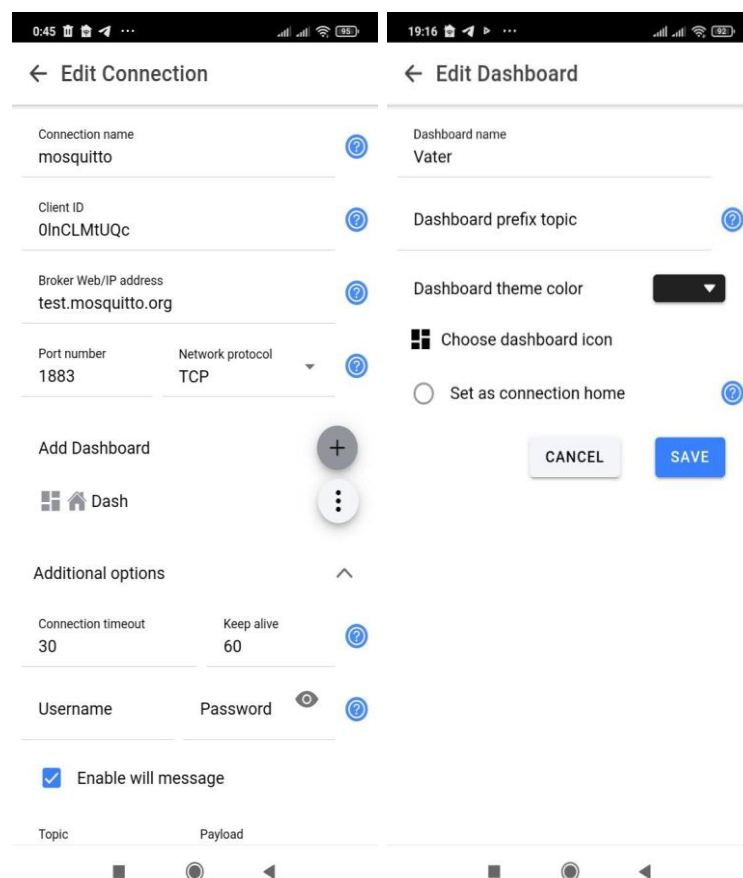


Рисунок 3.9 – Налаштування з'єднання з MQTT брокером та панелі керування (Dashboard)

Для відображення чи керування обраними параметрами додано відповідні елементи панелі керування.

На рис. 3.10 та 3.11 зображено налаштування елементів панелі керування.

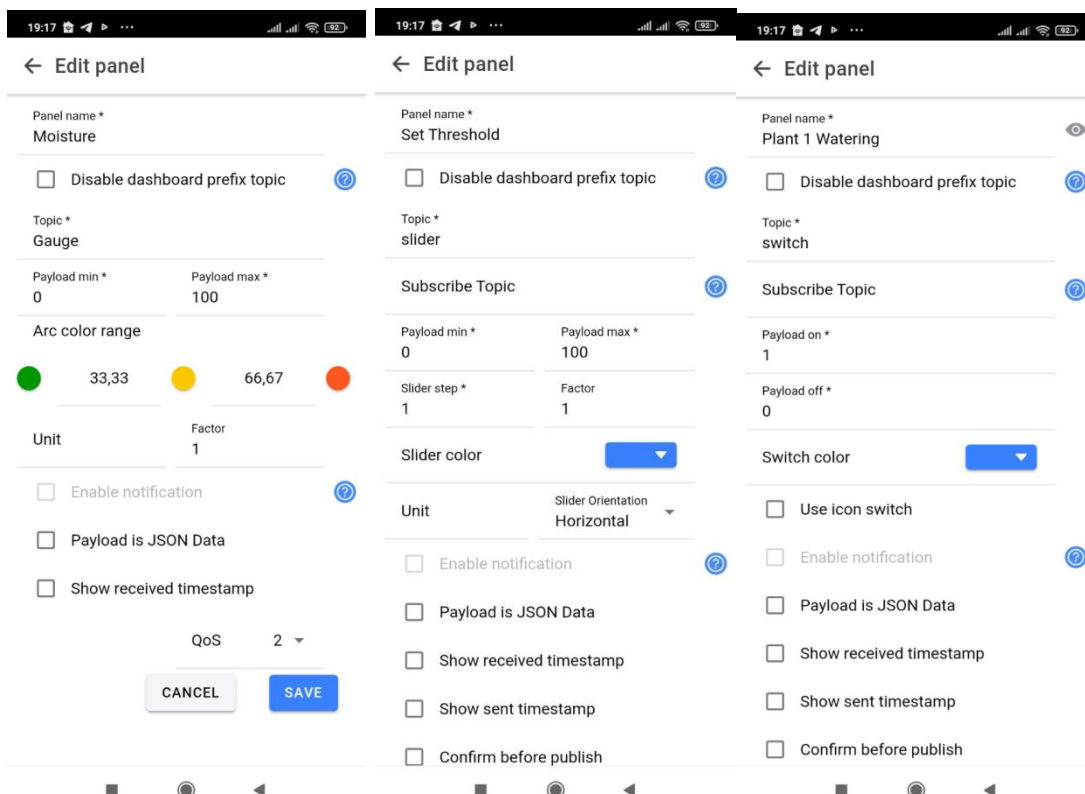


Рисунок 3.10 – Налаштування елементів gauge, slider, switch на панелі керування

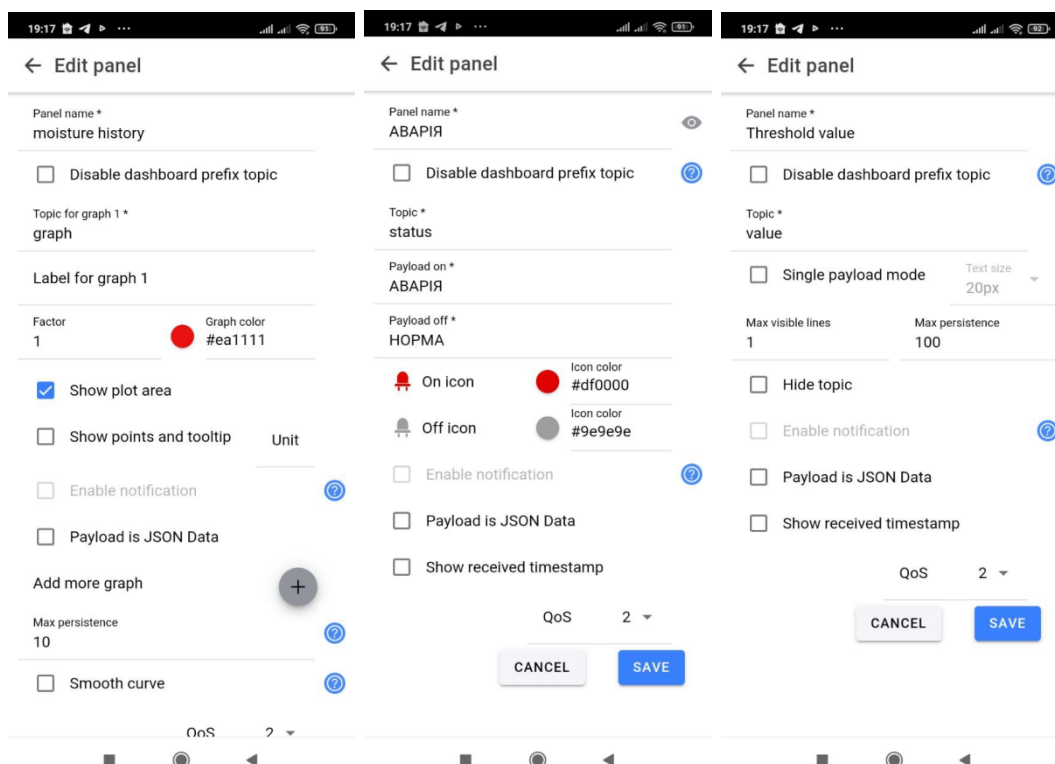


Рисунок 3.11 – Налаштування елементів graph, value та status

В результаті отримано наступний вигляд панелі керування (рис. 3.12), також доступна функція дублювання обраної панелі на інші пристрої.

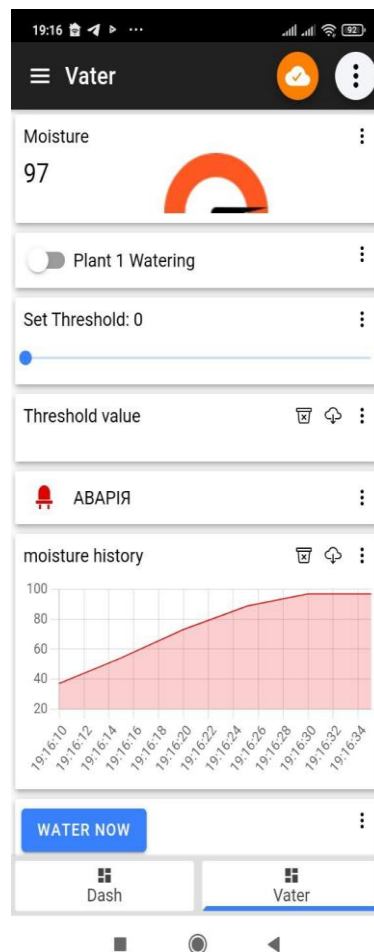


Рисунок 3.12 – Інтерфейс користувача модернізованої системи

3.5 Налаштування протоколу MODBUS у системі автоматичного моніторингу та керування параметрами клімату теплиці

Зазвичай Node-RED використовується або на стороні Edge, або в якості хмарного додатку. Коли Node-RED використовується на стороні Edge як програма для концентратора, шлюзу або маршрутизатора на апаратній платформі Raspberry PI, він може збирати дані з різних пристроїв за допомогою протоколів промислових мереж. На сьогоднішній день найпоширенішим і

простим протоколом на сьогоднішній день є Modbus. Raspberry PI буде взаємодіяти з пристроями по протоколу Modbus TCP/IP, а з іншого боку він буде взаємодіяти з хмарними додатками та сервісами[21].

З розділу палітри Inputs вставимо елемент modbustcp-read, зайдемо в налаштування та налаштуємо новий сервер.

Після створення серверу налаштуємо зчитування Holding регістрів починаючи з 0-го (рис. 3.13).

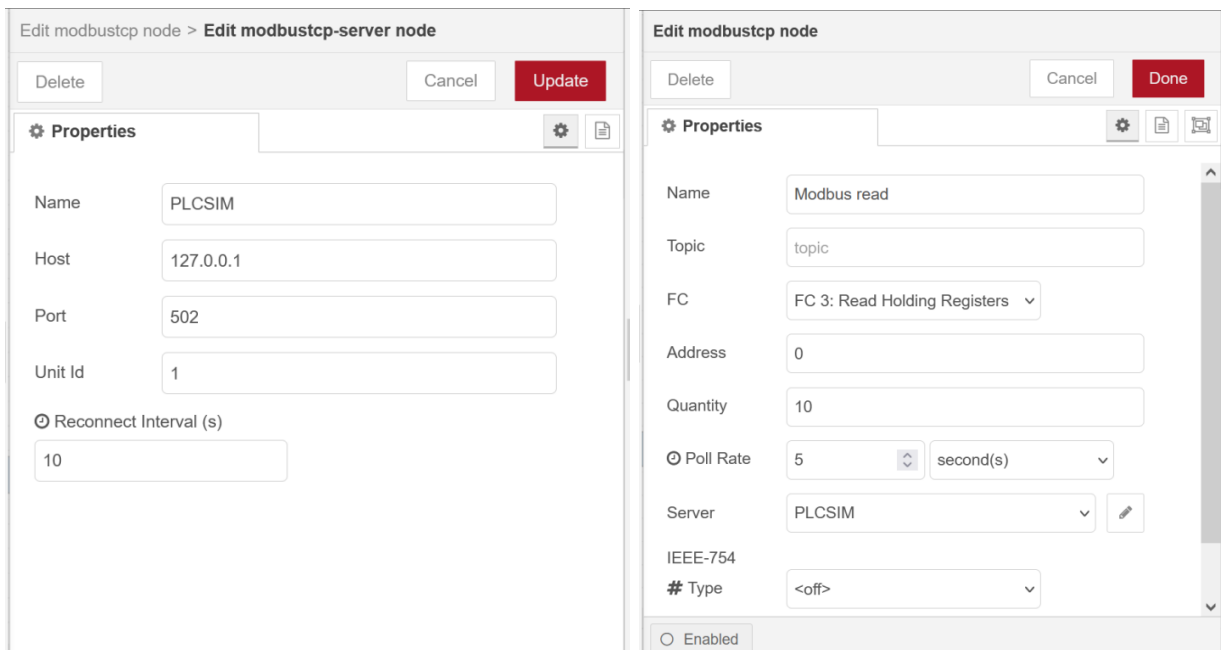


Рисунок 3.13 – Створення серверу Modbus TCP/IP та налаштування зчитування даних масиву

Фрагмент програми зображений на рис. 3.14.

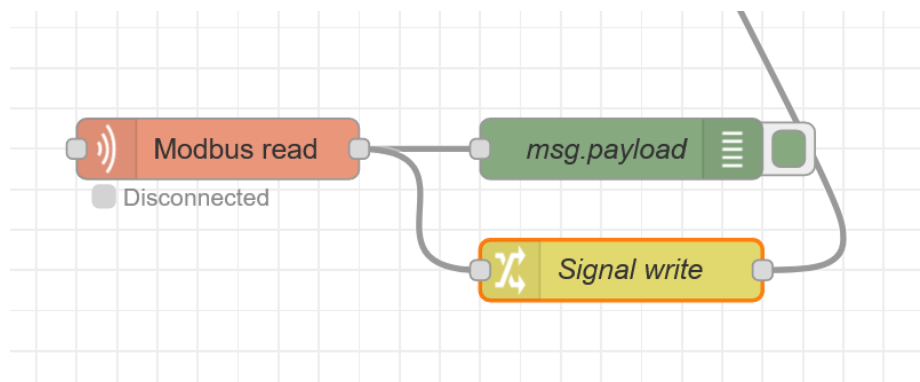


Рисунок 3.14 – Фрагмент програми зчитування даних протоколу modbus

Вузол Signal write виконує зчитування нульового елемента масиву msg.payload та запис значення в msg.payload(2).

Налаштування вузла зображено на рис. 3.15.

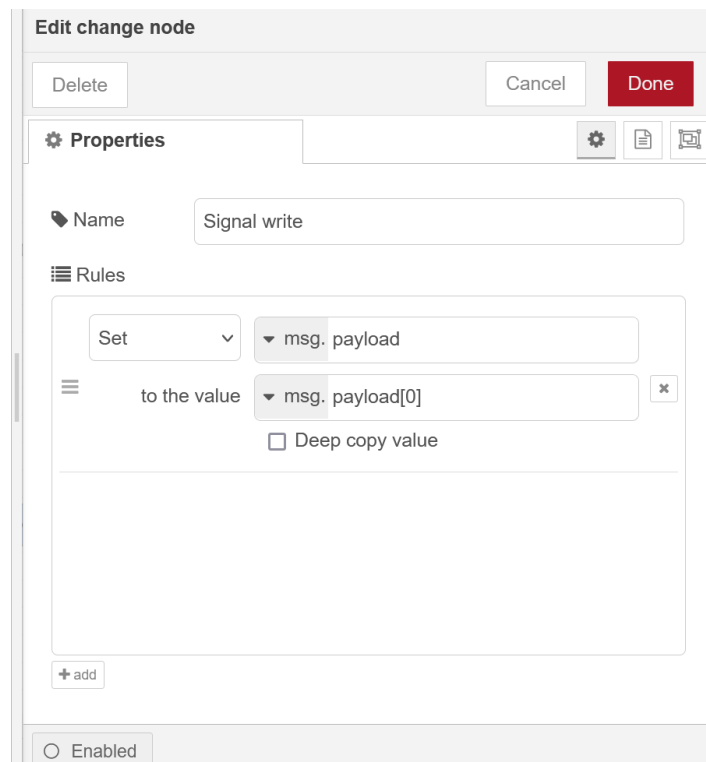


Рисунок 3.15 – Налаштування блоку Signal write

В даному випадку msg.payload є масивом з декількох елементів, тому для роботи з цими значеннями була виконана їх попередня обробка.

В даному випадку для перевірки налаштувань була використана програма-емулятор мережі Modbus Mod_RSsim (рис. 3.16), що відповідає за передачу та збереження даних. Дані представлені у вигляді масиву з 5 елементів, що зображують параметри, отримані відповідними датчиками.

Системою використовується значення п'яти датчиків – температури, вологості, CO2, освітлення, розходу води. Їх значення записані у відповідний масив.

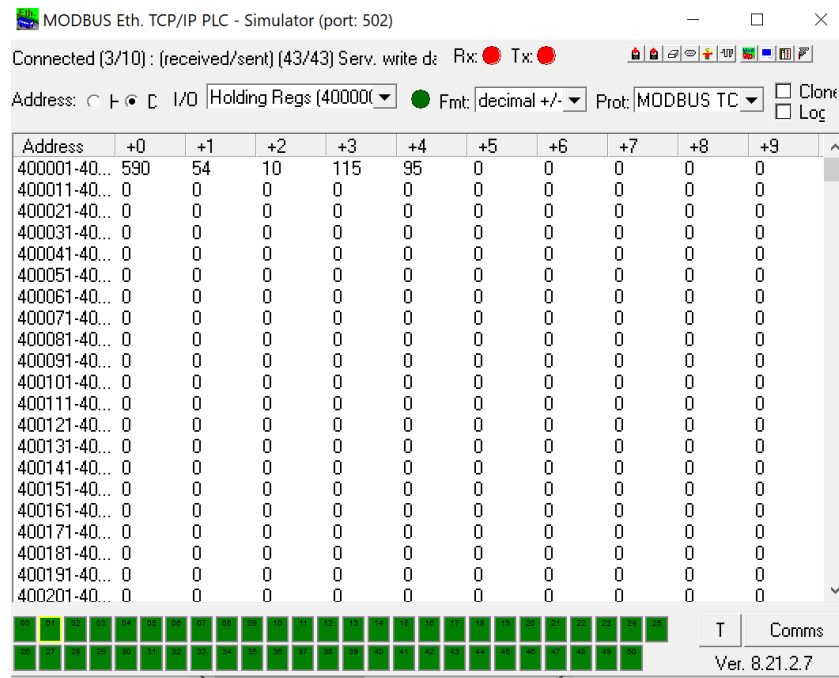


Рисунок 3.16 – Зовнішній вигляд програми Mod_RSsim з масивом елементів

Для запису по Modbus використовується вузол modbustcp-write з розділу outputs (рис. 3.17).

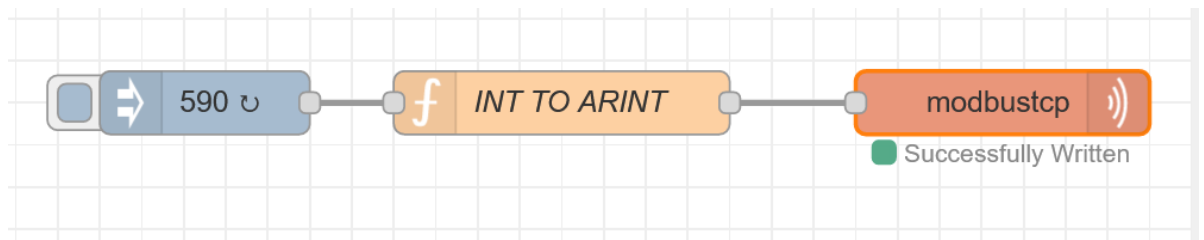


Рисунок 3.17 – Фрагмент програми та налаштування запису по Modbus.

Налаштування вузла Write виконуються аналогічно до попереднього, включаючи параметри серверу (рис. 3.18).

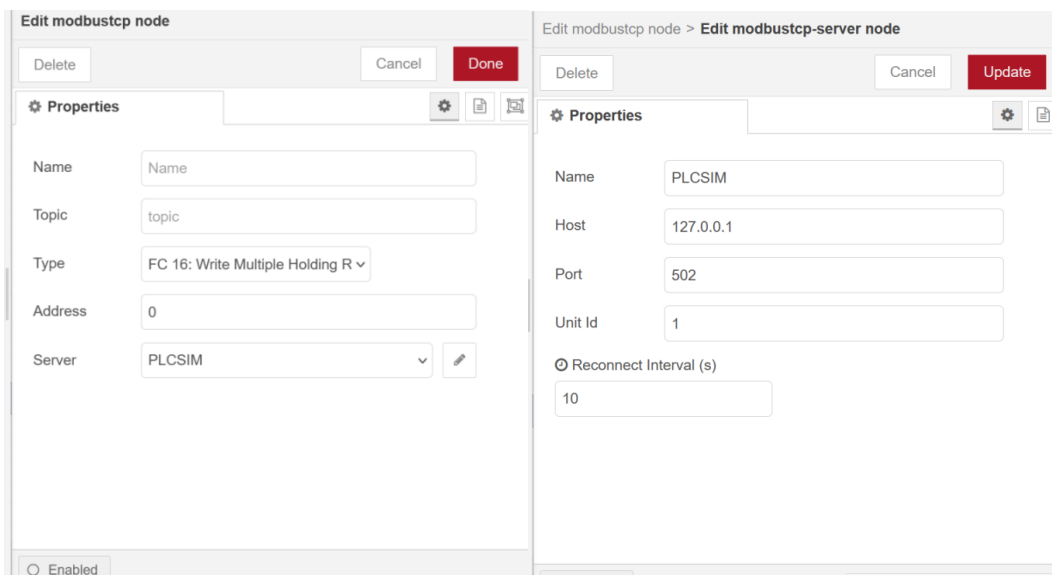


Рисунок 3.18 – Налаштування вузла Write та веб-серверу

3.6 Налаштування сервісу Google Sheets у системі автоматичного моніторингу та керування параметрами клімату теплиці.

Node-RED інтегрований з хмарним додатком Google Sheet.

Це дозволяє передавати дані з RaspberryPI в хмару для аналізу та взаємодії з віддаленими користувачами.

Для цього створений проект налаштування API для сервісів GoogleCloud <https://console.cloud.google.com/apis>

Створення проекту зображено на рис. 3.19.

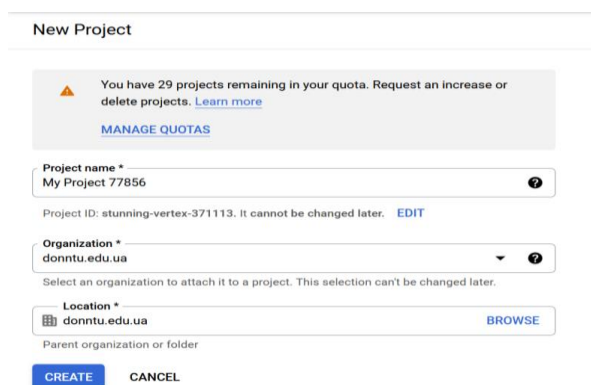


Рисунок 3.19 – Створення проекту

Файл ключа зберігається файл у форматі JSON (рис. 3.20).

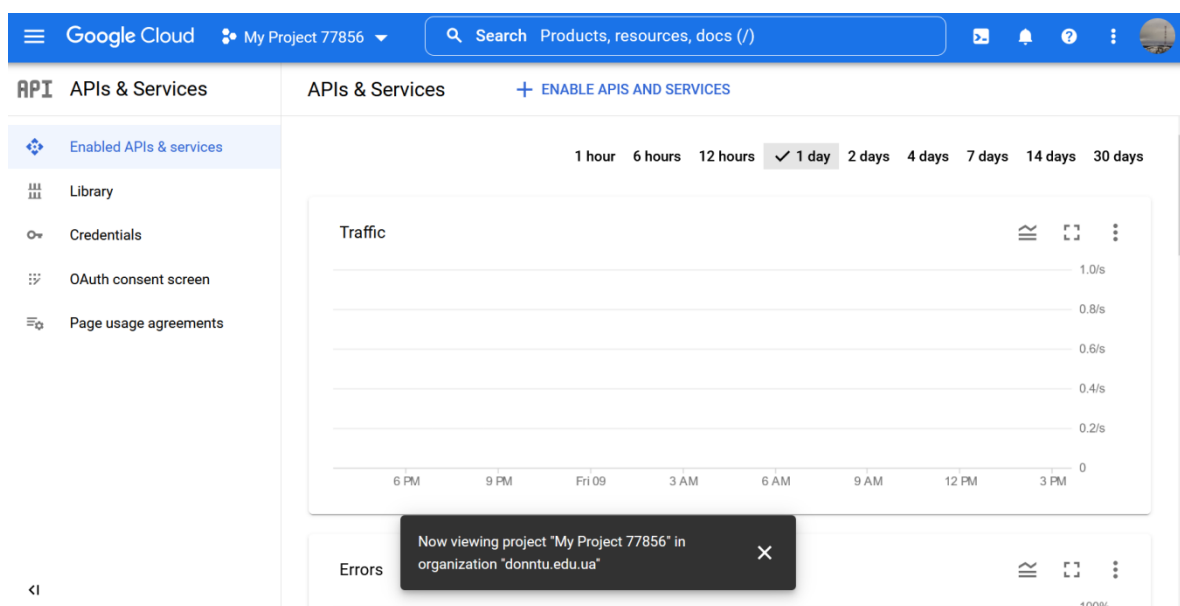


Рисунок 3.20 – Вигляд панелі налаштувань

Виконаємо створення сервісного акаунту у відповідному розділі головного меню (рис. 3.21, 3.22).

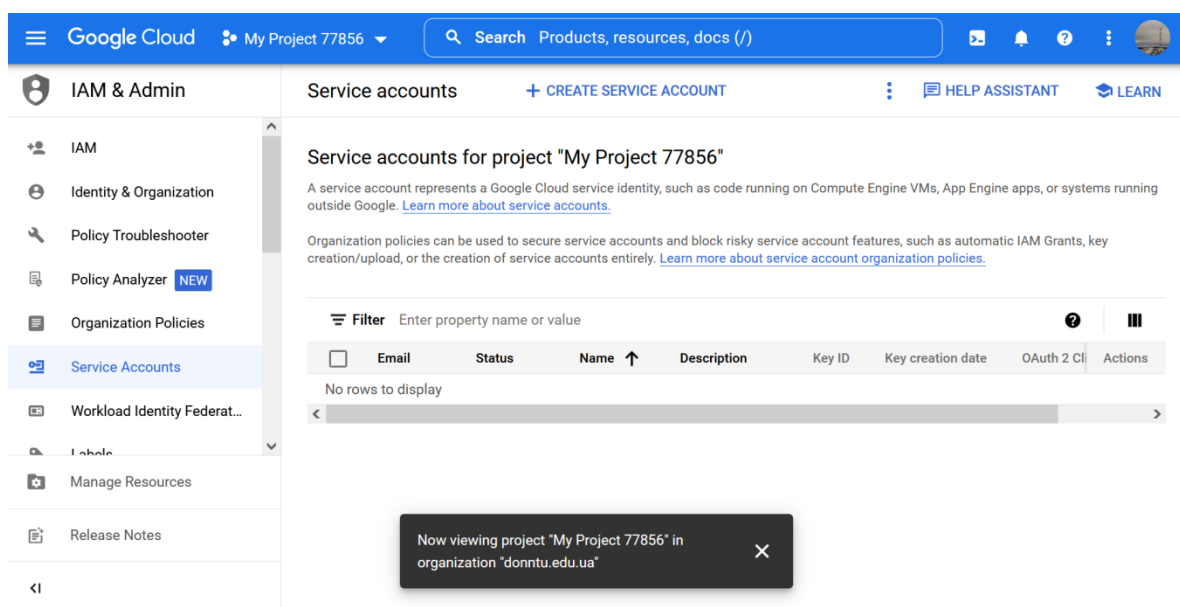


Рисунок 3.21 – Створення сервісного акаунту

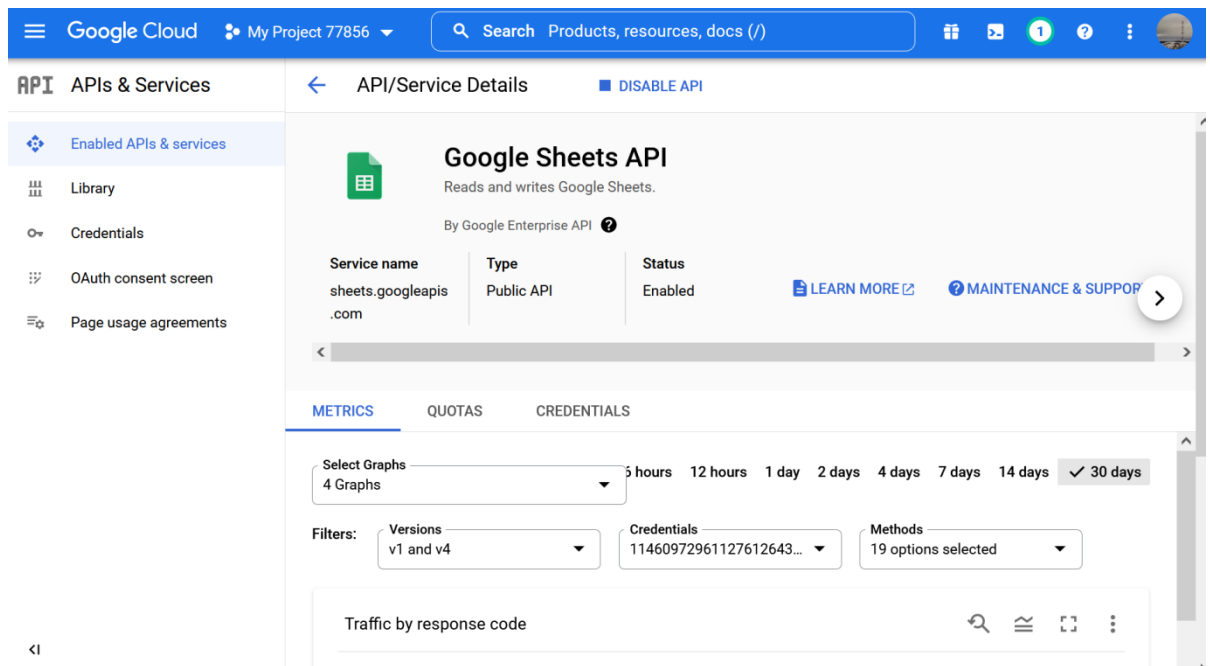


Рисунок 3.22 – Включення активації сервісу

Потрібно забезпечити можливість запису значень з Node-RED в електронну таблицю, яку обробляє хмарний сервіс електронних таблиць Google Sheets.

Створюється таблиця Google Sheet (рис. 3.23).

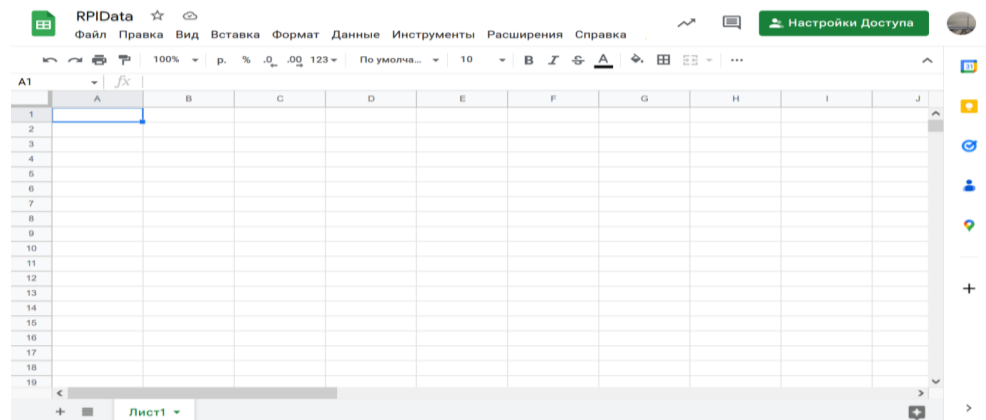


Рисунок 3.23 – Створення таблиці Google Sheet

У новому вікні виконаємо створення нової електронної таблиці, змінимо назву документу.

Необхідно створити фрагмент коду, який буде містити буфер останніх 60 значень датчиків разом з датою та часом їх зміни (рис. 3.24). Кожне нове

значення буде додаватися до верху масиву. Спочатку масив буде заповнюватися до 60 елементів. Коли кількість елементів досягне 61, найстаріші елементи будуть видалені, а масив зсунеться вниз. Для збереження даних між викликами, контекст потоку буде використовуватися.

Також, в коді передбачено перевірку на переповнення масиву: якщо кількість елементів перевищує 60, масив буде обрізаний до 60 елементів, навіть якщо вихідна кількість елементів більша. Така ситуація, в принципі, не повинна відбуватися, але рекомендується передбачати такі випадки для запобігання проблем[12].

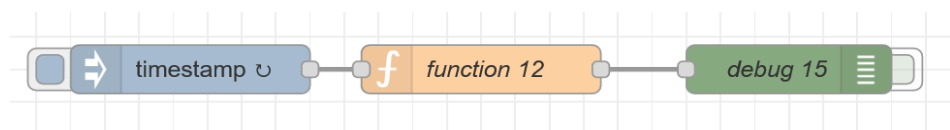


Рисунок 3.24 – Блок перевірки на переповнення масиву

Код функції (рис. 3.25):

```
let now = new Date(); //сьогодні
//trend - 2-мірний масив 3x60, на 60 останніх значень
//читання з контексту потоку або ініціалізація масиву-буферу
let trend = flow.get("trend") || [[now.getTime(), [0], [0]]];
let l = trend[0].length; //довжина масиву
//з кожним викликом збільшуємо градуси в радіанах
let out = global.get("out") || {};
let rad = out; //Math.PI/trend[l][l-1]
//if (rad >= Math.PI * 2) rad = 0; //обнуляємо після повного кола
let val = Math.sin (rad); //вираховуємо синус
//на "верх" масиву додаємо елементи
trend[0].push (now.toLocaleString()); //дата час
trend[1].push (rad); //радіани
trend[2].push (val); //значення синусу
//якщо масив заповнився до 60
if (trend[0].length > 59)
{trend[0].shift(); //вилучаємо перший (найстаріший) елемент
trend[1].shift();
trend[2].shift();
//після цього масив повинні зменшитися на 1 елемент (60)
//і зсунутися вниз
//у випадку, якщо раптом елементів більше 60
//наприклад були добавлені випадково стороннім кодом
//зробити кількість елементів =60
trend[0].length = 59;
trend[1].length = 59;
trend[2].length = 59;
}
flow.set("trend", trend); //записати в контекст потоку
let rtdb = { //глобальна змінна
trend: trend, rad: rad, val: val
}
global.set("rtdb", rtdb);
return msg;
```

Рисунок 3.25 – Фрагмент коду перевірки на переповнення масиву

Часом оновлення 1 блоку Inject задається період оновлення даних, другого – оновлення таблиці (рис. 3.26, 3.27).

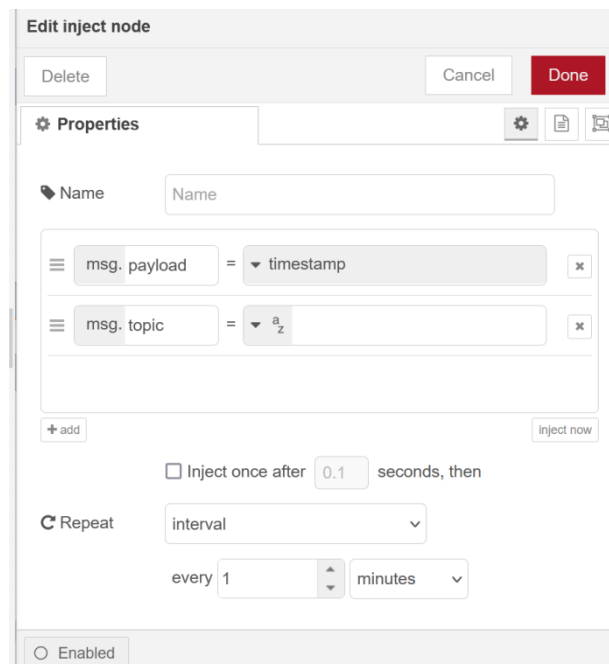


Рисунок 3.26– Налаштування першого блоку Inject

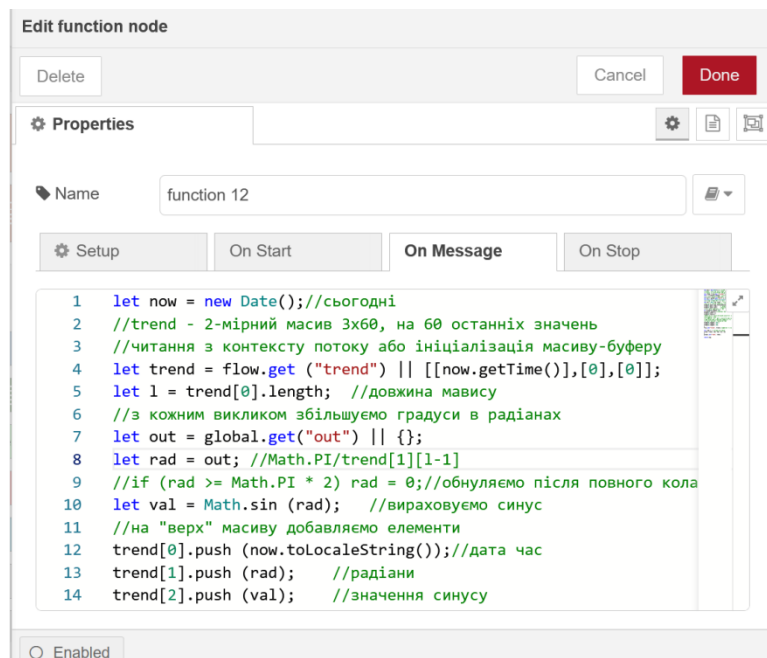


Рисунок 3.27 – Фрагмент програми для запису даних в масив

Надається доступ до Гугл-таблиці створеному раніше сервісному аканту (рис. 3.28):

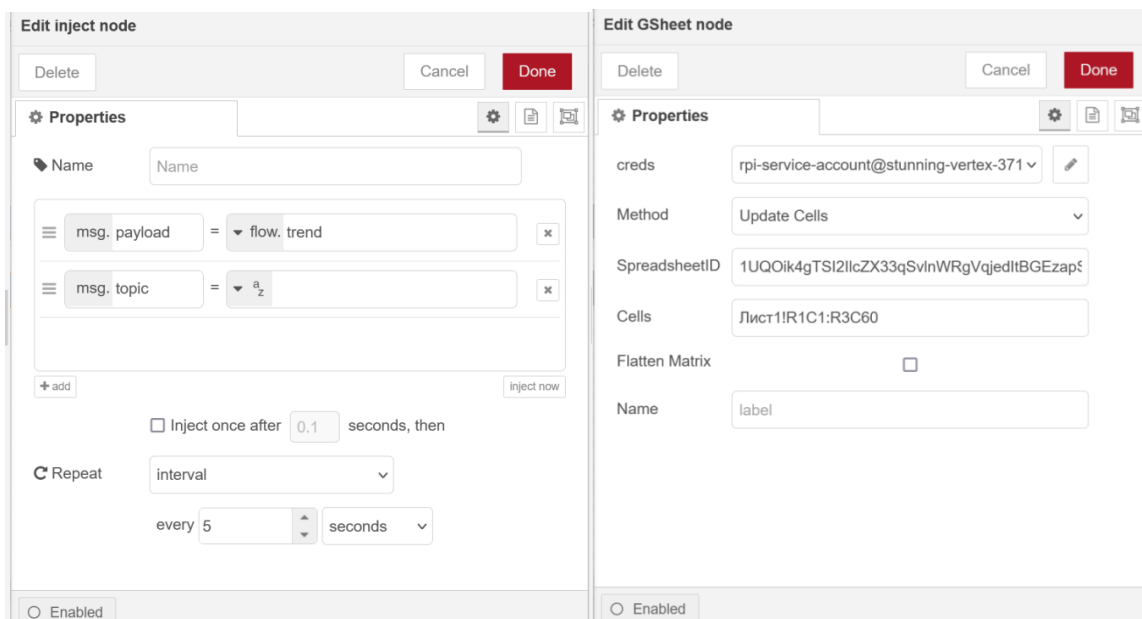


Рисунок 3.28 – Налаштування блоків Inject та GSheet

Також оновимо потік, додавши вузол Google Sheet (рис. 3.29).

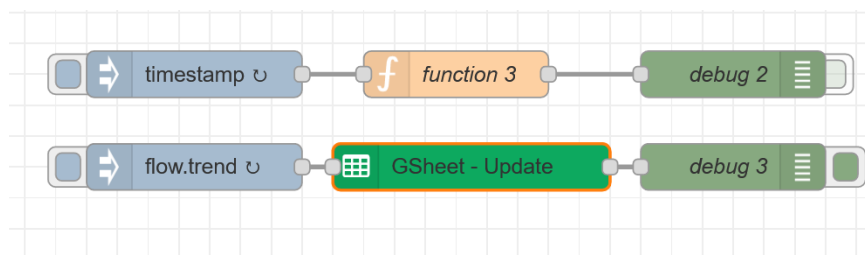


Рисунок 3.29 – Зібрана схема Google Sheets

Після розгортання потоку у Гугл-таблиці відображаються необхідні значення (в даному випадку для однієї змінної) (рис. 3.30).

The screenshot shows a Google Sheet titled 'RPIData' with a table of data. The table has 8 columns (A-H) and 8 rows (1-8). The data is as follows:

	A	B	C	D	E	F	G	H
1	Дата	14.01.2023, 18:28:02	14.01.2023, 18:29:02	14.01.2023, 18:30:02	14.01.2023, 18:31:02	14.01.2023, 18:32:02	14.01.2023, 18:33:02	14.01.2023, 18:34:02
2	Вологість у %	28	24	22	22	26	32	40
3	0.6832617147	0.2709057883	-0.905578362	-0.00885130929	-0.00885130929	0.7625584505	0.5514266812	0.74511
4								
5								
6								
7								
8								

Рисунок 3.30 – Таблиця з записаними даними

Також створено діаграму залежності значень змінних від часу. Діаграма зображена на рис. 3.31.

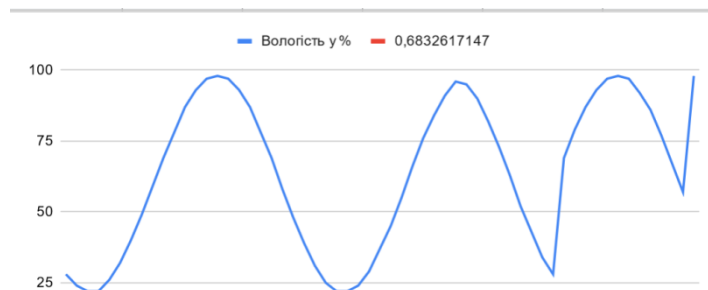


Рисунок 3.31 – Відображення діаграми за записаними даними

3.7 Налаштування телеграм-боту системи автоматичного моніторингу та керування параметрами клімату теплиці.

Дана частина роботи вимагає налаштування Telegram-бота, який забезпечує онлайн-комунікацію між користувачами та RPI.

Знайдемо @BotFather, бота, який створює інших ботів.

Щоб створити нового бота, виберемо команду newbot. Буде визначено ім'я Telegram-бота. Введемо ім'я користувача бота, яке має закінчуватися на bot. (рис. 3.32):

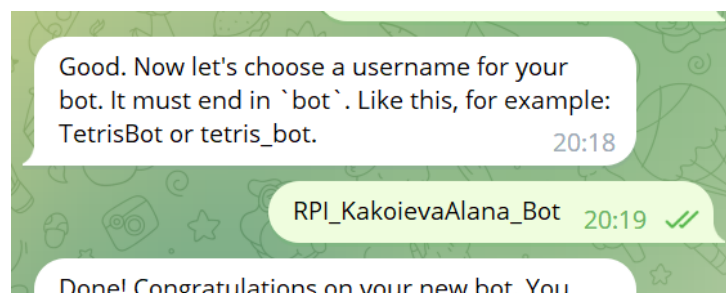


Рисунок 3.32 – Надання імені користувача телеграм-боту

Телеграм-бот буде використовуватися в діалозі з використанням команд. Перша команда, яку можна використовувати, - /start. Викликання цієї команди виведе список доступних команд.

Створюємо новий блок (рис. 3.33):

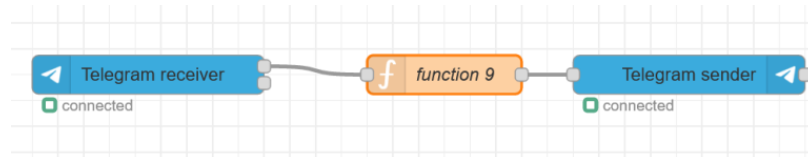


Рисунок 3.33 – Формування стартового повідомлення допомоги

Функція має наступний вигляд (рис. 3.34):

```

let txtmsg = "/start - показати базові команди\r\n";
txtmsg += "/sp - виставлення уставок\r\n";
msg.payload.content = txtmsg;
return msg;

```

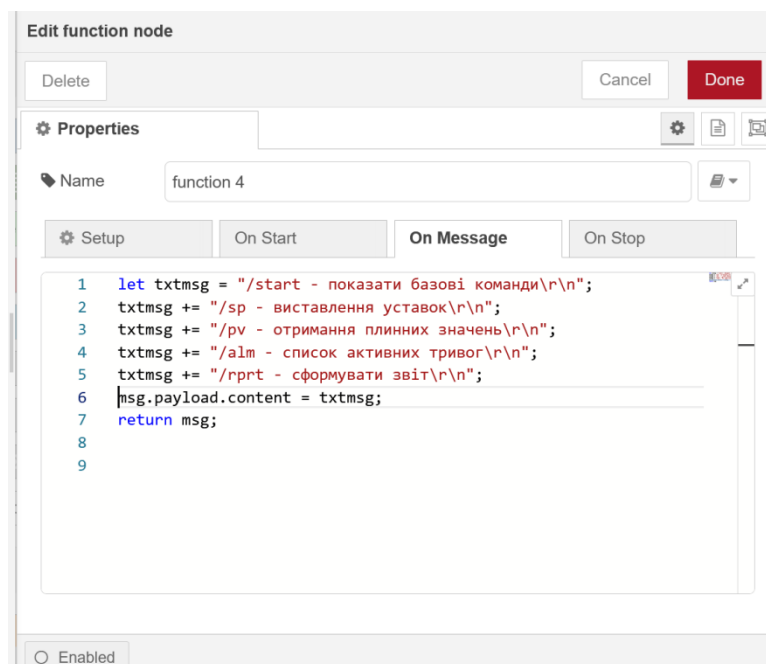


Рисунок 3.34 – Блок Function

Зробимо розгортання потоку, можна спостерігати відповідь на команду /start у чаті (рис. 3.35).

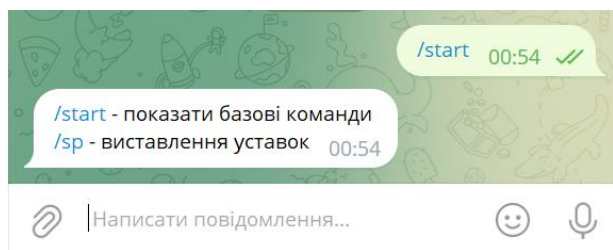


Рисунок 3.35 – Повідомлення з командами

Зробимо формування повідомлення в Телеграм, коли значення змінних будуть перевищувати задані.

Задамо значення змінних спрацьовування сигналізації.

Цей фрагмент записує буфер та значення змінних в глобальний контекст (рис. 3.36).

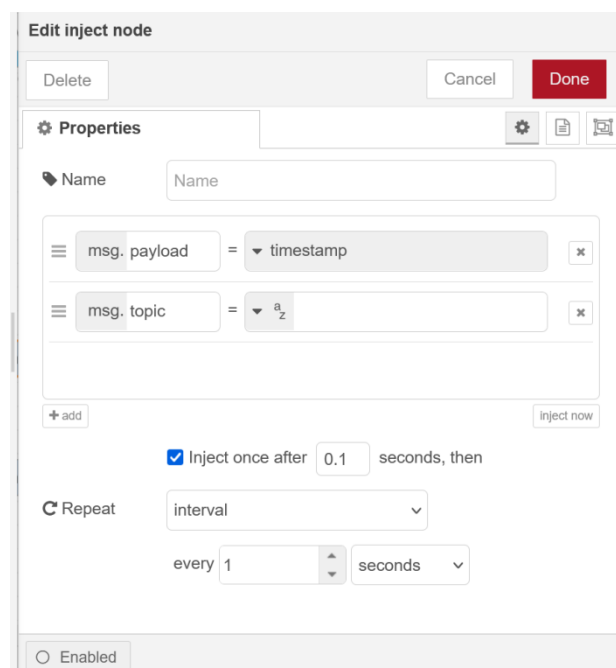


Рисунок 3.36 – Налаштування блоку Inject

Фрагмент сигналізації наведено на рис. 3.37.

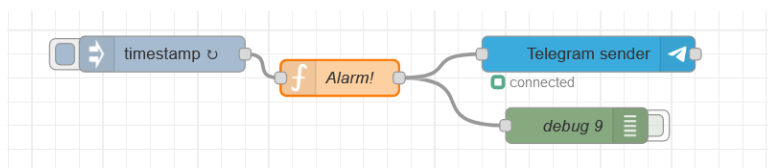


Рисунок 3.37 – Фрагмент потоку

Функція Alarm наведена у додатку Б (рис. 3.38).

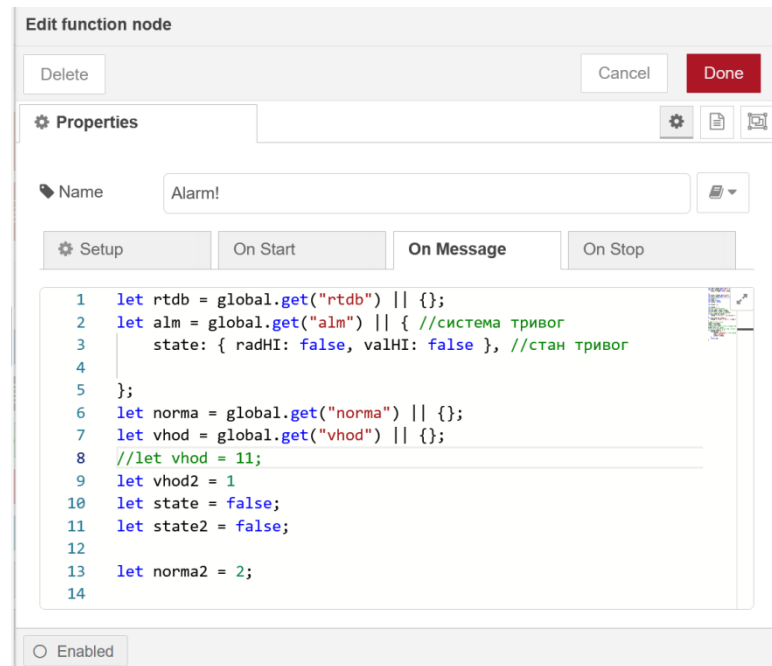


Рисунок 3.38 – Налаштування блоку “Function”

Вузол change використовується для зміни заданого значення змінної. Задання значення зберігається як `global.norma` у глобальному контексті. (рис. 3.39).

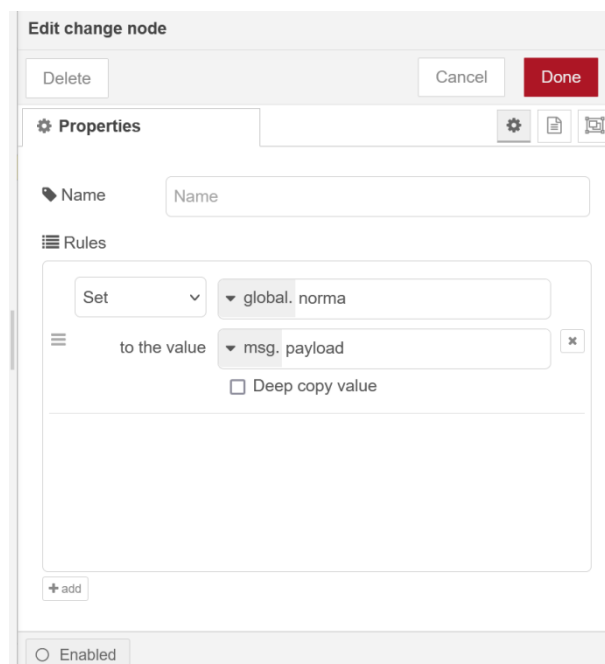


Рисунок 3.39 – Налаштування блоку «Change»

Використовуючи фрагмент з вузлом timestamp та change сформуємо значення `global.norma` рівним 3. У результаті з певним періодом в чаті генеруються тривоги про перевищення заданих значень (рис. 3.40).

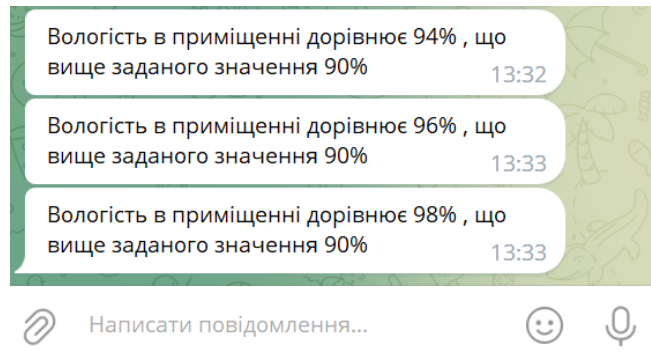


Рисунок 3.40 – Перевищення значення тривоги, повідомлення телеграм-боту

Змінимо налаштування сигналізації. Для цього скористаємося командою `/sp`. Команда `/sp` створює клавіатуру з трьома кнопками:

- `rad` - вибирає значення верхнього рівня налаштування змінної `rad` і надсилає відповідну команду;
- `val` - вибирає значення верхнього рівня налаштування змінної `val` і надсилає відповідну команду;
- `Cancel` - скасовує вибір.

Крім того, вибір змінної генерує відповідну команду, яка обробляється іншим обробником із запитом на введення значень налаштувань цих змінних. (рис. 3.41). Налаштування функції зображено на рис. 3.42.

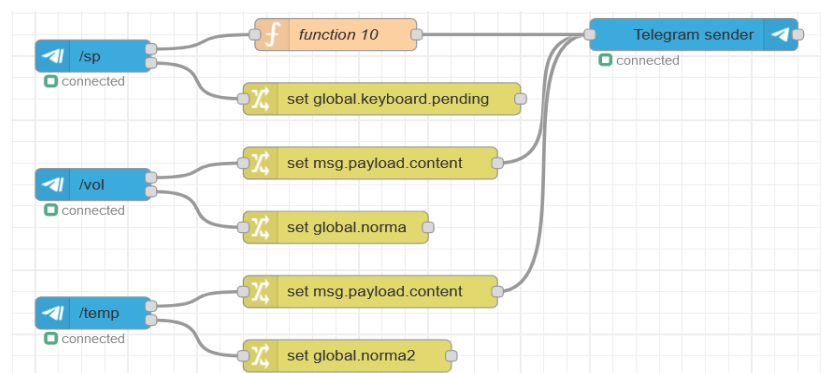


Рисунок 3.41 – Блоки уставок сигналізації

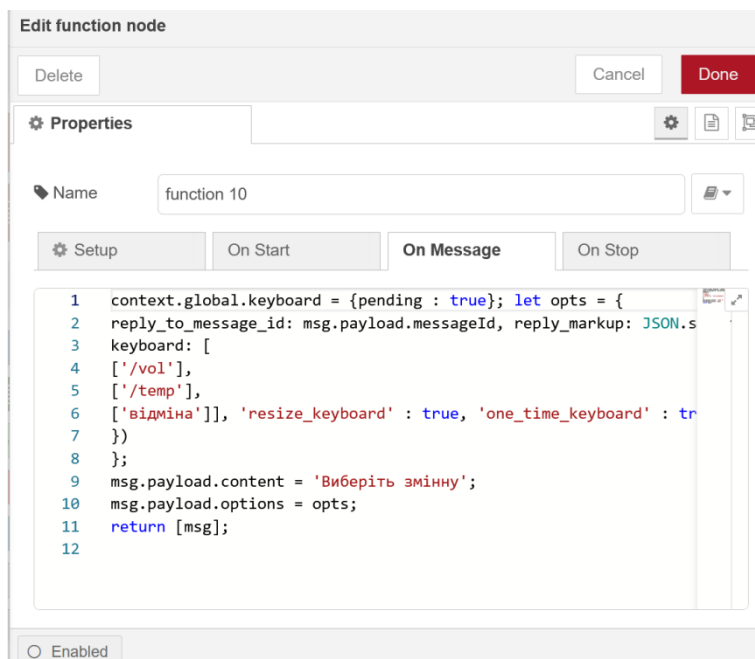


Рисунок 3.42 – Налаштування блоку «Function»

Текст функції спрацювання сигналізації наведено на рисунку 3.43.

```
context.global.keyboard = {pending : true}; let opts = {
reply_to_message_id: msg.payload.messageId, reply_markup: JSON.stringify({
klavye: [
['/vol'],
['/temp'],
['відміна']], 'resize_keyboard' : true, 'one_time_keyboard' : true
})
};
msg.payload.content = 'Виберіть змінну';
msg.payload.options = opts; ; msg.payload.content = 'Виберіть змінну'
return [msg];
```

Рисунок 3.43 – Код функції спрацювання сигналізації

Налаштування блоку «Change» – рис. 3.44.

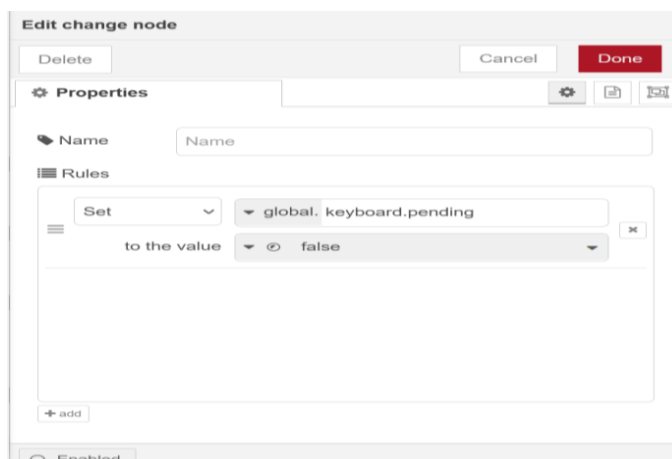


Рисунок 3.44 – Налаштування одного з блоків «Change»

Спостерігаємо реакцію на команду /sp (рис. 3.45).

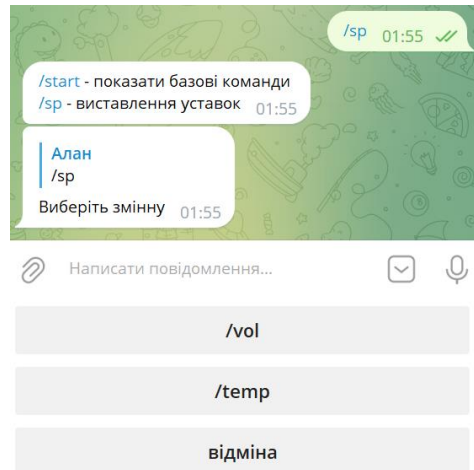


Рисунок 3.45 – Реакція на команду /sp

При збільшенні значення норми тривоги не спрацьовують.

На рис. 3.46 наведено структурну схему системи керування теплицею. На рис. 3.47 зображена функціональна схема.

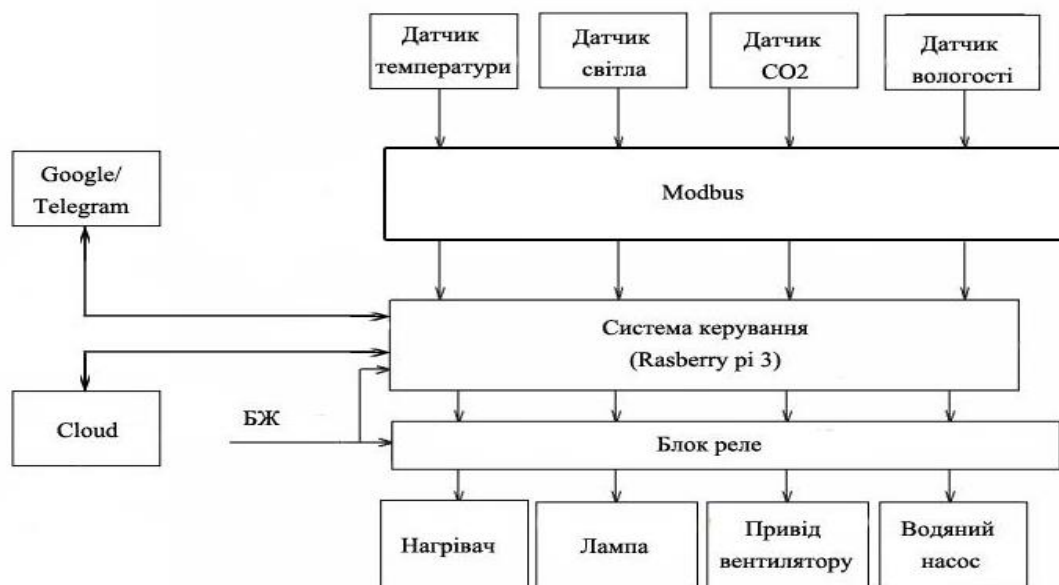


Рисунок 3.46 – Структурна схема системи автоматичного моніторингу та керування параметрами клімату теплиці

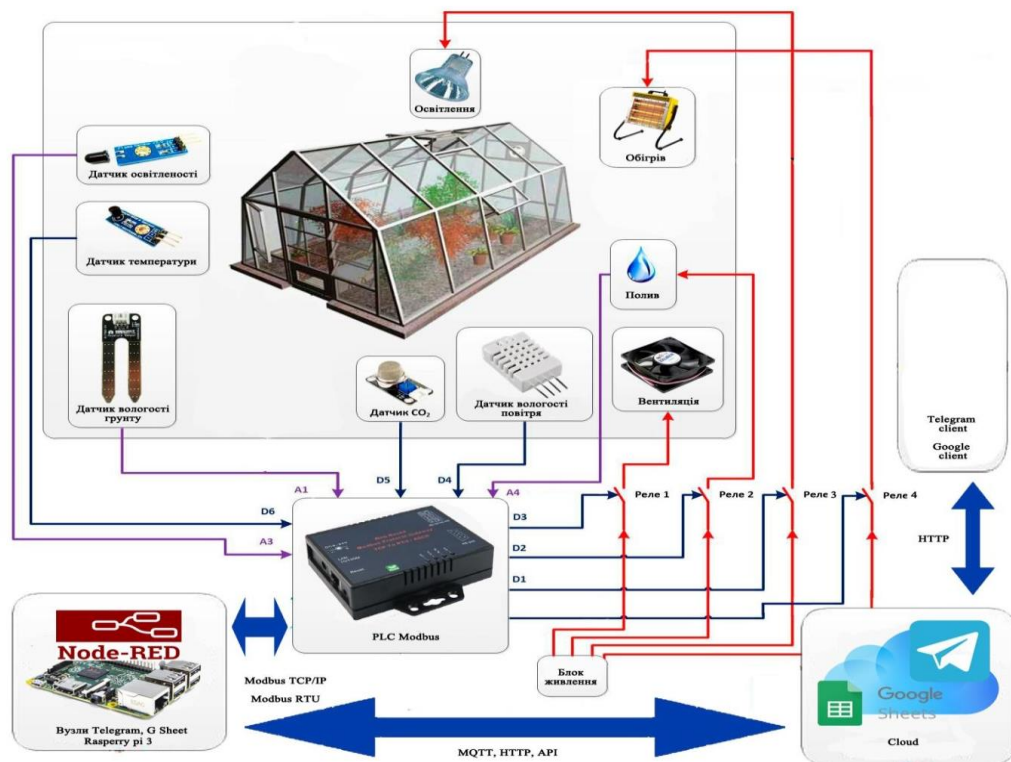


Рисунок 3.47 – Функціональна схема системи автоматичного моніторингу та керування параметрами клімату теплиці

Висновки до розділу 3

1) Були розроблені алгоритми, які базуються на модернізованих функціональних схемах, для управління системою, що контролюється контролером Raspberry Pi 3.

2) На основі модернізованих алгоритмів були складені схеми отриманої системи та програма для керування теплицею.

ВИСНОВКИ

Дана робота заснована на роботі з обладнанням, яке призначене для вирішення експлуатаційних завдань в теплицях, зокрема автоматизації процесів, поліпшення працездатності та зручності використання. Також розглядається тема підвищення рівня автоматизації теплиць до рівня Індустрії 4.0 промислової революції.

Об'єктом розробки є автоматизована система управління обладнанням в теплицях. Предметом розробки є системи автоматичного управління, які забезпечують певні характеристики об'єкта. Для розробки використовуються методи системного аналізу та декомпозиції, теорія автоматичного керування, комп'ютерне моделювання та програмування в середовищі Node-RED.

Метою даної роботи є розробка програмного забезпечення для автоматизації теплиць, включаючи віддалений моніторинг показників датчиків в реальному часі, аналіз та вибір апаратного забезпечення. Також представлений огляд застосування концепції Інтернету речей (IoT) в промисловості, зокрема в сільському господарстві.

Крім того, проведено порівняльний аналіз апаратних засобів, таких як мікроконтролери Arduino та Raspberry Pi, для автоматизації планування роботи теплиць. Розглянуті основні датчики для моніторингу характеристик, таких як вологість ґрунту та освітленість, а також можливі алгоритми для створення мікроклімату.

На основі вибраного апаратного та програмного забезпечення розроблено автоматизовану систему на базі одноплатного комп'ютера Raspberry Pi 3 та створено веб-додаток з користувацьким інтерфейсом.

В даній роботі також було:

- 1) Виконано аналітичний огляд тепличних господарств, як об'єктів автоматизації.
- 2) Проведено опис технологічного процесу, рівня автоматизації системи, огляд існуючих програмних рішень для автоматизації

3) Враховуючи дану оглядову інформацію, було встановлено, що для створення автоматизованого комплексу слід враховувати такі фактори:

- Для керування системою використовується система на основі мікроконтролера.

- Використання (відповідність) технологіям Industry 4.0, за допомогою даного способу досягається максимальна автоматизація процесу.

Отже, мета бакалаврської роботи досягнута, її завдання виконано.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Климентовський, Ю. А.. Технічні засоби автоматики: навчальний посібник / Ю.А. Климентовський, А.М. Гладкий. - К., 2003. - 238 с.
- 2 Технічні засоби автоматизації / В.В. Ткачов, В.П. Чернишев, М.М. Одиновол ; Нац. гірн. ун-т. - Д.: НГУ, 2007. - 174 с.
- 3 E.S. Bondar, A.S. Hordienko, V.A. Mihailov, G.V. Nimich, Ventilation and air conditioning systems automation. Kiev:LLC Tipografiya "Avanpost-Prim", 2005, pp. 95-101.
- 4 Проць Я.І., Савків В.Б., Шкодзінський О.К., Ляшук О.Л. Автоматизація виробничих процесів. Навчальний посібник. — Тернопіль: ТНТУ ім. І.Пулюя, 2011.— 344 с.
- 5 Автоматика для теплиці на мікроконтролері [Електронний ресурс]. – Режим доступу: <https://infracom74.com/avtomatika-dlya-teplitsy-na-mikrokontrollere/>
- 6 Вибір концепції побудови мережі [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/10056924/page:3>
- 7 Node-RED: Введення/Запуск на Raspberry Pi [Електронний ресурс]. – Режим доступу: https://wikihandbk.com/wiki/Node-RED:%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5/%D0%97%D0%B0%D0%BF%D1%83%D1%81%D0%BA_%D0%BD%D0%B0_Raspberry_Pi
- 8 Використання сучасних засобів автоматизації у теплиці [Електронний ресурс]. – Режим доступу: <http://elektrik.info/main/automation/1478-ispolzovaniye-sovremennyh-sredstv-avtomatizacii-v-teplice.html>
- 9 Датчик температури і вологості DHT22 [Електронний ресурс]. – Режим доступу: <https://arduinka.biz.ua/uk/datchik-temperaturi-i-vologosti-dht22-p564c74.html>

10 Індустрія 4.0 [Електронний ресурс]. – Режим доступу: <https://koeebox.com/ua/terminy-i/industrija-4-0/>

11 Жураковський Б.Ю., Федорова Н.В., Гаврилко Є.В., Зенів О.І. Технології створення Інтернету речей. — Київ,: КПІ ім. І. Сікорського, 2021.— 127 с.

12 Журнал «Наукові праці Національного університету харчових технологій». Том 27, № 1 [Електронний ресурс]. – Режим доступу: <https://dspace.nuft.edu.ua/jspui/bitstream/123456789/32860/3/%d0%9d%d0%b0%d1%83%d0%ba%d0%be%d0%b2%d1%96%20%d0%bf%d1%80%d0%b0%d1%86%d1%96%20%d0%9d%d0%a3%d0%a5%d0%a2%20%d0%a2%d0%be%d0%bc%2027%20%e2%84%961.pdf>

13 МАТЕРІАЛИ XXIV науково-технічної конференції [Електронний ресурс]. – Режим доступу: <https://web.znu.edu.ua/NIS/2019/tom-3.pdf>

14 Surface related multiple imaging based on one-way wave equation [Електронний ресурс]. – Режим доступу: <https://library.seg.org/doi/10.1190/segam2013-0041.1>

15 Плата міні-комп'ютера Raspberry Pi 3 Model B [Електронний ресурс]. – Режим доступу: <https://arduino.ua/ru/prod1449-raspberry-pi-3-b>

16 9 найкращих інструментів розробки для Інтернету речей у 2020 році [Електронний ресурс]. – Режим доступу: <https://proglib.io/p/9-luchshih-instrumentov-razrabotki-dlya-interneta-veshchey-v-2020-godu-2020-10-19>

17 Технічне забезпечення інформаційних технологій в агропромисловому комплексі [Електронний ресурс]. – Режим доступу: www.tsatu.edu.ua/tsst/wp-content/uploads/sites/6/materialy-3-mnpk-tehnichne-zabezpechennja-innovacijnyh-tehnolohij-v-ahropromyslovomu-kompleksi-m.-melitopol-01-26.11.2021.pdf

18 Курс. Освітній портал [Електронний ресурс]. – Режим доступу: https://learn.ztu.edu.ua/pluginfile.php/69335/mod_folder/content/0/Л-8.pdf?forcedownload=1

19 Технології сучасних кібер-фізичних систем: Навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2020. – 327 с. [Електронний ресурс]. – Режим доступу:

https://ela.kpi.ua/bitstream/123456789/43394/1/TKFS_NP_2020_Lektsii.pdf

20 Технології Індустрії 4.0 [Електронний ресурс]. – Режим доступу: asu.in.ua/mod/book/view.php?id=123&chapterid=309

ДОДАТОК А

Охорона праці та безпека під час надзвичайних ситуацій на підприємстві

А.1 Основні шкідливі виробничі фактори, що впливають на умови праці робітників відділу АСУ ТП

Виробниче середовище працівників сфери автоматизованих систем управління (АСУ) піддається впливу різних шкідливих та небезпечних факторів. Серед них можна виділити: навантаження на очі та стомлення зору, навантаження на руки та пальці, статичні умови роботи, які викликають застої в організмі, різні види випромінювання (рентгенівське, електромагнітне, інфрачервоне, статичні поля), шум від роботи кулерів, дисководів та оргтехніки, іонізація повітря, присутність хімічних речовин у повітрі (наприклад, озон, триметилфосфат, біфеніли) та статична електрика. Психологічні фактори також впливають на працівників і включають розумове перенапруження та нервово-емоційне перевантаження, які можуть сприяти розвитку професійних захворювань, таких як синдром комп'ютерного зору, радіоперешкоди, синдром сухості рогівки, синдром зап'ястного каналу, шкірні захворювання, кишкові та серцево-судинні захворювання та комп'ютерна алергія.

А.2 Заходи щодо поліпшення умов праці робітників відділу АСУ ТП

Щоб покращити умови праці працівників у секторі автоматизованих систем управління технологічними процесами, необхідно вжити певні заходи. При організації робочих місць з персональними комп'ютерами необхідно уникати встановлення ПК у підвальних приміщеннях. Приміщення повинні бути опалюваними, вентильованими та кондиціонованими. Рекомендується використовувати антистатичні кондиціонери для підтримання відносної вологості повітря на рівні 50-60%. Підлогу також слід виконати з антистатичного матеріалу і дотримуватися параметрів роботи ПК. Для зменшення негативного впливу роботи з ПК необхідно дотримуватися таких гігієнічних вимог: екран і клавіатура повинні бути розташовані на оптимальній відстані від очей

користувача (600-700 мм); висота робочої поверхні столу повинна регулюватися в межах 680-800 мм, а площа робочого столу повинна бути не менше 600 мм у висоту, 500 мм у ширину, 450 мм у глибину (нижче колін) і 650 мм у висоту витягнутої ноги. Також рекомендується скоротити тривалість дзвінків і вибрати мобільний телефон з низьким питомим коефіцієнтом поглинання (SAR), який вимірює максимальну випромінювану потужність мобільного телефону. У Європі максимальне значення SAR становить 2 Вт/кг. Крім того, після встановлення з'єднання з абонентом рекомендується тримати телефон ближче до вуха, оскільки випромінювана потужність збільшується під час з'єднання. Всі ці вимоги щодо поліпшення умов праці представлені на схемі, показаній на рисунку 4.1.

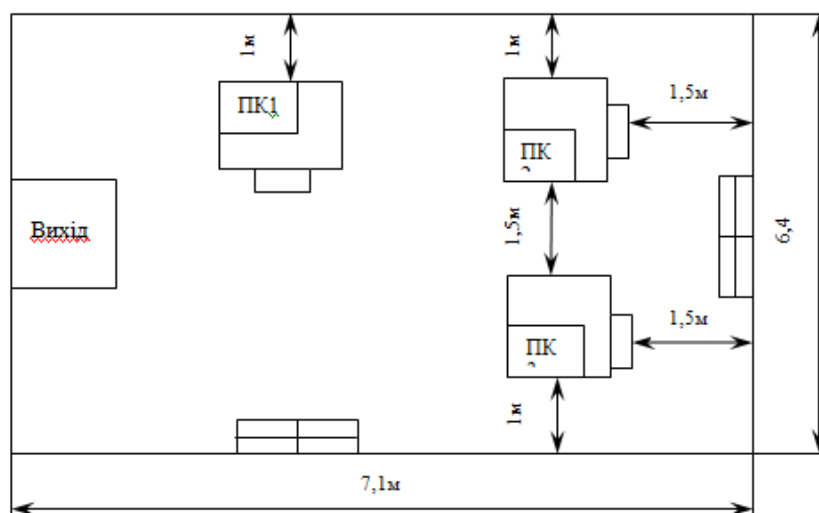


Рисунок 4.1 – Облаштуваність робочих місць із ПК

Таким чином, дотримано всіх вимог. При цьому працюючі не звернені обличчям до вікна при роботі за комп'ютером.

А.3 Розрахунок кондиціонування та вентиляції для відділу АСУ ТП

У відділі АСУ ТП є джерела екстремальних температур, тому необхідно визначити необхідні умови їх вентилявання. Витрату повітря у відділі з додатковим тепловиділенням визначаємо по формулі:

$$L = \frac{Q_{над}}{c \cdot p (t_b - t_n)}, \quad (A.1)$$

де $Q_{над}$ - надлишкове виділення тепла в робочому приміщенні, ккал/год.;

c - теплоємність повітря (0,237 ккал/кг);

p - обсягова вага повітря (1,226 кг/ м³);

t_b - температура витяжного повітря (26°С);

t_n - температура повітря, що притікає (18°С).

Розрахуємо надлишкове надходження тепла по формулі:

$$Q_{над} = Q_{уст} + Q_{пер} + Q_{осв} + Q_{ср}, \quad (A.2)$$

де $Q_{уст}$ - виділення тепла від устаткування;

$Q_{пер}$ - виділення тепла робітниками;

$Q_{осв}$ - надходження тепла від електричного освітлення;

$Q_{ср}$ - надходження тепла від сонячної радіації через вікна.

Визначимо виділення тепла від устаткування по формулі:

$$Q_{уст} = P \cdot K_a \cdot K_b \cdot 860 \text{ ккал/год}, \quad (A.3)$$

де P - сумарна потужність устаткування, кВт/год;

K_a - коефіцієнт установленної потужності (0,95);

K_b - коефіцієнт одночасної роботи (1,0).

$$Q_{уст} = [(x_1 \cdot k_1) + (x_2 \cdot k_2) + (x_3 \cdot k_3) + (x_4 \cdot k_4)] \cdot K_a \cdot K_b \cdot 860 \text{ ккал/год}, \quad (A.4)$$

де x - кількість системних блоків, моніторів, принтерів, кондиціонерів відповідно; k - потужність системних блоків, моніторів, принтерів, кондиціонерів відповідно.

Розрахуємо:

$$Q_{уст} = [(3 \cdot 0,5) + (3 \cdot 0,1) + (1 \cdot 0,4) + (1 \cdot 6,3)] \cdot 0,95 \cdot 1 \cdot 860 = 490,2 \text{ ккал/год}. \quad (A.5)$$

Визначимо виділення тепла від обслуговуючого персоналу за допомогою формули:

$$Q_{\text{пер}} = n \times g, \quad (\text{A.6})$$

де n - кількість працюючих;

g - кількість тепла, що виділяє один працівник за годину (100 ккал/год.)

Розрахуємо:

$$Q_{\text{пер}} = 6 \times 100 = 600 \text{ ккал/год.} \quad (\text{A.7})$$

Визначимо надходження тепла від електричного освітлення по формулі:

$$Q_{\text{осв}} = E_{\text{м}} \cdot g_1 \cdot S, \quad (\text{A.8})$$

де $E_{\text{м}}$ – нормована освітленість для цієї зорової роботи, приймаємо рівним 400 лк;

g_1 – питома тепловиділення на 1 м² підлоги при 1 лк освітленості (для люмінесцентних ламп – 0,05 ккал/год.);

S – площа приміщення, м².

Розрахуємо:

$$Q_{\text{осв}} = 400 \cdot 0,05 \cdot 45,44 = 908,8 \text{ ккал/год.} \quad (\text{A.9})$$

Визначимо надходження тепла від сонячної радіації через вікна по формулі:

$$Q_{\text{ср}} = F \cdot g_2 \cdot K_{\text{осл}}, \quad (\text{A.10})$$

де F - площа віконних прорізів (1,55 м²);

g_2 - кількість тепла, що надходить через 1 м² віконного прорізу (65 ккал/год.);

$K_{\text{осл}}$ - коефіцієнт ослаблення, приймаємо - 0,4.

Розрахуємо:

$$Q_{cp}=1,55 \cdot 65 \cdot 0,4 = 40,3 \text{ ккал/год.} \quad (\text{A.11})$$

Визначимо кількість надлишкового тепла:

$$Q_{над} = 490,2 + 600 + 908,8 + 40,3 = 2039,3 \text{ кал/год.} \quad (\text{A.12})$$

Визначимо витрати повітря в приміщенні:

$$L = \frac{2039,3}{0,237 \cdot 1,226 \cdot (26 - 18)} = 877,31 \text{ м}^3 / \text{год} \quad (\text{A.13})$$

Існуюча в наявності система вентилявання має продуктивність 600 куб. м./годину, але це не задовольняє необхідним нормативам. Потрібно замінити функціонуючу систему вентилявання на більш потужнішу. Параметри мікроклімату на робочих місцях регламентуються ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень». Відповідно доданих санітарних норм температура повітря, швидкість руху повітря і відносна вологість у холодні періоди року повинна складати 22-24 градуса по Цельсію, 0,1 метра в секунду і 40-60 % відповідно. У теплі періоди року температура повітря повинна складати 23-25 градусів Цельсія, рухливість повітря 0,1-0,2 метрів секунду, вологість 40-60 %. Температура може коливатися від 22 до 26 градусів Цельсія при збереженні всіх інших параметрів мікроклімату. Вище зазначені норми цілком відповідають фактичним відділу АСУ ТП.

А.4 Розрахунок системи загального рівномірного освітлення з лампами розжарювання

Розміри відділу: довжина (а=7,1 м), ширина (в=6,4 м), висота (h=3 м). Визначимо норми освітлення для та розрахункову висоту для виділених зон. Мінімальна освітленість за нормами E=400 лк. Приміщення має світлу побілку:

коефіцієнт відбиття - $P_{\text{стелі}} = 70\%$, $P_{\text{стін}} = 50\%$. Висота робочих поверхонь (столів) $h_p = 0,7$ м. Для освітлення прийнято світильники типу УПМ-15, які підвищуються до стелі, відстань від світильника до стелі $h_c = 0,5$ м. Мінімальна освітленість за нормами $E=400$ лк.

1) Визначимо висоту підвісу світильників над підлогою:

$$h_0 = H - h_c = 3 - 0,5 = 2,5 \text{ м.} \quad (\text{A.14})$$

Для світильників загального освітлення з лампами розжарювання потужністю до 200 Вт мінімальна висота підвісу над підлогою відповідно до СНіП П-4-79 повинна бути у межах 2,5 - 4,0 м, залежно від характеристики світильника. В нашому випадку по відповідає цій вимозі.

2) Визначимо висоту підвісу світильника над робочою поверхнею:

$$h = h_0 - h_p = 2,5 - 0,7 = 1,8 \text{ м.} \quad (\text{A.15})$$

Рівномірність освітлення досягається при відповідному співвідношенні відстані між світильниками (L) і висоти їх підвісу (h).

3) Визначимо рекомендовану відстань між світильниками:

$$L = 0,7h = 0,7 \cdot 1,8 = 1,26 \text{ м.} \quad (\text{A.16})$$

4) Розрахуємо необхідну кількість світильників:

$$N = \frac{ab}{L^2} = \frac{7,1 \cdot 6,4}{1,16^2} = 33,77 \approx 35 \quad (\text{A.17})$$

Приймаємо 35 світильників, враховуючи розміри приміщення розміщуємо їх у 3 ряди по 7 штук.

5) Світловий потік лампи світильника визначається за формулою:

$$\Phi_{\text{л}} = \frac{E \cdot K_3 \cdot S \cdot Z}{N \cdot n \cdot \eta} \text{ ккал / год} , \quad (\text{A.18})$$

де E - нормативна освітленість, лк;

K_3 - коефіцієнт запасу, що враховує зниження освітленості в результаті забруднення та старіння ламп (1,5);

S - площа приміщення, що освітлюється, м^2 ;

Z - коефіцієнт нерівномірності освітлення для ламп розжарювання (=1,15);

N - кількість світильників;

n - кількість ламп у світильнику;

η - коефіцієнт використання світового потоку, який визначається за світлотехнічними таблицями залежно від показника приміщення (i) та коефіцієнтів відбиття стін та стелі.

6) Визначимо показник приміщення для кожної зони:

$$i = \frac{ab}{h(a+b)} = \frac{7,1 \cdot 6,4}{1,8 \cdot (7,1 + 6,4)} = 1,8 . \quad (\text{A.19})$$

7) З таблиці 3.25 [2] знаходимо коефіцієнт використання ($\eta = 0,52$) для світильників УПМ-15 (при $P_{\text{стелі}} = 70\%$, $P_{\text{стін}} = 50\%$).

8) Світловий потік одного світильника, а значить і лампи, оскільки за конструктивним виконанням у світильнику цього типу встановлена лише одна лампа, дорівнює:

$$\Phi = \frac{E \cdot S \cdot K_3 \cdot Z}{N \cdot \eta} = \frac{400 \cdot 45,44 \cdot 1,5 \cdot 1,15}{35 \cdot 0,52} = 1722,73 \text{ лм} . \quad (\text{A.20})$$

9) З таблиці 3.27 [2] обираємо лампу БК (біспіральна криптонова) потужністю 100 Вт, світловий потік якої становить 1450 лм. Хоча це значення менше розрахованого на 14%, однак не перевищує встановлену норму (-

$0\% < \Delta\Phi_{\text{л}} < +20\%$). Сумарна електрична потужність усіх світильників, встановлених у приміщенні становить:

$$\Sigma P_{\text{св}} = P \cdot N = 100 \cdot 35 = 3,5 \text{ кВт.} \quad (\text{A.21})$$

А.5 Пожежна безпека

Основні заходи пожежної безпеки в приміщеннях Відділу систем управління викладені в Інструкції з пожежної безпеки в офісних приміщеннях. Вони є обов'язковими для виконання всіма працівниками. Інструкція про заходи пожежної безпеки в офісних приміщеннях забороняє влаштовувати тимчасові джерела електроживлення; використовувати саморобні запобіжники; прокладати дроти безпосередньо до паливного столу; експлуатувати лампи без ковпаків (розсіювачів); використовувати саморобні подовжувачі, які не відповідають вимогам Правил улаштування електроустановок; переробляти вимикачі, розетки для вішання одягу тощо; обгортати світильники та лампи; закривати частини електромережі легкозаймистими матеріалами. Блокування легкозаймистою тканиною або папером; використання побутових електрочайників, кип'ятильників тощо без негорючих підставок; залишення без нагляду підключених до електромережі кондиціонерів, комп'ютерів, калькуляторів, друкарських машинок тощо. Перешкоджати доступу до засобів пожежогасіння; використовувати гідранти, рукави та протипожежний інвентар не за призначенням; зберігати документи, різні матеріали, предмети та обладнання в підсобних шафах (нішах); палити (крім місць, спеціально відведених адміністрацією та позначених як "місця для паління" і обладнаних попільничками або попільничками з негорючих матеріалів); проводити зварювальні та інші вогневі роботи без встановленого дозволу, використання легкозаймистих рідин тощо.

А.6 Безпека при надзвичайних ситуаціях на підприємстві

Меблі та обладнання на робочих місцях компанії повинні бути розташовані таким чином, щоб забезпечити вільний шлях евакуації (шириною не менше 1 м) до вихідних дверей. Евакуаційні шляхи та виходи повинні бути завжди вільними від перешкод. Засоби протипожежного захисту в приміщеннях повинні утримуватися в належному робочому стані.

У разі виникнення пожежі необхідно: негайно повідомити про пожежу в Національну пожежну службу за номером "101", вказавши адресу, номер поверху, місце виникнення, наявність або відсутність людей та своє прізвище; повідомити про пожежу керівництву, а в нічний час - черговому охоронцю;

Якщо є можливість приступити до гасіння пожежі наявними засобами, організувати пожежну команду.

У разі виникнення пожежі на ранніх стадіях можливе виділення тепла, токсичних продуктів горіння та руйнування будівельних конструкцій. Тому людей слід евакуювати з палаючих будівель якнайшвидше. Показником ефективності евакуації є час, за який працівники можуть залишити окремі приміщення або всю будівлю, якщо це необхідно. Безпека евакуації досягається тоді, коли час евакуації не перевищує часу критичного періоду пожежі, тобто часу між початком пожежі і досягненням шкідливих для людини факторів пожежі (таких як критичні температури, рівні задимлення і знижений рівень кисню). Кількість евакуаційних виходів має бути не менше двох. Вони повинні бути розосереджені та розташовані.

Двері на шляхах евакуації повинні відчинятися в напрямку виходу з будівлі. План евакуації на випадок пожежі повинен бути вивішений на видному місці в кожному приміщенні. У разі виникнення пожежі необхідно враховувати небезпеку та механізми дії вогню на людей.

Якщо приміщення не горить, але заблоковане від основних шляхів евакуації вогнем, димом або високою температурою, в першу чергу необхідно запобігти потраплянню диму і продуктів горіння в це приміщення. Для цього негайно закрийте всі щілини в дверях і під ними змоченими водою ганчірками, рушниками або робочим одягом.

Якщо в приміщенні все ще багато диму, підповзійте до вікна і закрийте рот і ніс вологою тканиною, яка буде діяти як фільтр і забезпечить певний захист від продуктів горіння.

У задимлених приміщеннях, де більша частина нагрітих газоподібних токсичних речовин і диму накопичується у верхній частині приміщення, а також у приміщеннях, де температура на рівні очей у шість разів вища за температуру на рівні підлоги, а рівень кисню завжди вищий внизу.

ДОДАТОК Б

Функція «Alarm»

```

let rtdb = global.get("rtdb")||{}
let alm = global.get("alm")||{/alm system
  state: { radHI: false, valHI: false }, //стан тривоги.

};
norma = global.get("norma")||{}
let vrod = global.get("vrod")||{}
// let vrod = 11;
let vrod2 = 1
let state = false; // let vrod = 11; let vrod2 = 1
let state2 = false;

let norma2 = 2;

let almmsg = "";
// Чи дійсний стан тривоги?
let alm_radHI = vrod > norma;
let alm_valHI = vrod2 > norma2;
// якщо тривога щойно спрацювала
if (alm_radHI && !state) { (if alm_radHI)
  almmsg += "Вологість у приміщенні на " + vrod + "% перевищує значення, встановлене у " +
norm + "%\r\n."
}
if (alm_valHI && !state2){.
  almmsg += "Синус Долвіна " + vrod2 + " більше заданого значення " + norma2 + "%\r\n".
}
// Записати стан тривоги.
state = alm_radHI;
state2 = alm_valHI;
// Зберегти стан тривоги в global.
global.set("alm", alm);
// якщо виникла хоча б одна тривога - відправити повідомлення
if (almmsg.length > 1) { (almmsg.length>1){.
  msg.payload = {
    chatId: "627794772", //Ідентифікатор має бути тут.
    type: "message",
    content: almmsg.
  }
} return msg;
}

```

ДОДАТОК В

Технічні характеристики Raspberry Pi 3 model B

№	Найменування	Значення
1	SoC	Broadcom BCM2837 ARM Cortex-A53 Quad Core 1.2 GHz
2	Кількість ядер процесора	4
3	GPU	VideoCore IV 3D
4	RAM	1 GB
5	Сховище	microSD
6	Мережеві можливості	Ethernet 10/100
7	Відео вихід	HDMI
7	USB порти	4 x USB 2.0
8	Бездротові можливості	Bluetooth WiFi 2.4GHz 150 mbps
9	Аудіо вихід	3,5 Jack
10	GPIO	40 pins
11	Живлення	5V 2.5A
12	Додаткові можливості	Діод живлення та активності, композитний відео-вивихід через 3.5мм Jack

ДОДАТОК Г

Текст програми

```
[
  {
    "id": "1b6d4b38.cc6c95",
    "type": "tab",
    "label": "Greenhouse",
    "disabled": false,
    "info": ""
  },
  {
    "id": "7e18a8a7.dea448",
    "type": "rpi-gpio out",
    "z": "1b6d4b38.cc6c95",
    "name": "Relay 1",
    "pin": "27",
    "set": true,
    "level": "0",
    "freq": "",
    "out": "out",
    "bcm": true,
    "x": 1160,
    "y": 180,
    "wires": []
  },
  {
    "id": "d500a2a9.53b83",
    "type": "rbe",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "func": "rbe",
    "gap": "",
    "start": "",
    "inout": "out",
    "septopics": true,
    "property": "payload",
    "topi": "topic",
    "x": 850,
    "y": 180,
    "wires": [
      [
        "442146e2.5be678",
        "7e18a8a7.dea448",
        "61e9a0160b01beab"
      ]
    ]
  },
  {
    "id": "640c0a49.a92a94",
```

```

    "type": "function",
    "z": "1b6d4b38.cc6c95",
    "name": "Check function",
    "func": "num = flow.get(\"threshold\") || 50;\n//node.log(\"retr thres = \" + num);\n\nif\n(msg.payload > num)\n  msg.payload = 0;\nelse\n  msg.payload = 1;\n\n\nreturn msg;",
    "outputs": "1",
    "noerr": 2,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 680,
    "y": 180,
    "wires": [
      [
        "d500a2a9.53b83"
      ]
    ]
  },
  {
    "id": "d4bf05ce.fe0c08",
    "type": "function",
    "z": "1b6d4b38.cc6c95",
    "name": "Set value",
    "func": "\n\nflow.set(\"threshold\",msg.payload); \nnode.log(\"threshold set to \" +\nmsg.payload);\n\n",
    "outputs": "1",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 560,
    "y": 640,
    "wires": [
      [
        "25616d4c2dc0bb0a"
      ]
    ]
  },
  {
    "id": "eb8a28de.544e28",
    "type": "comment",
    "z": "1b6d4b38.cc6c95",
    "name": "Система контролю вологості теплиці",
    "info": "",
    "x": 650,
    "y": 120,
    "wires": []
  },
  {
    "id": "f137eb44.ee9a48",
    "type": "trigger",
    "z": "1b6d4b38.cc6c95",

```

```

    "name": "",
    "op1": "1",
    "op2": "0",
    "op1type": "num",
    "op2type": "num",
    "duration": "5",
    "extend": false,
    "overrideDelay": false,
    "units": "s",
    "reset": "",
    "bytopic": "all",
    "topic": "topic",
    "outputs": 1,
    "x": 620,
    "y": 520,
    "wires": [
      [
        "7e18a8a7.dea448",
        "442146e2.5be678",
        "61e9a0160b01beab"
      ]
    ]
  },
  {
    "id": "af69006.4bbb",
    "type": "delay",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "pauseType": "rate",
    "timeout": "5",
    "timeoutUnits": "seconds",
    "rate": "20",
    "nbRateUnits": "1",
    "rateUnits": "minute",
    "randomFirst": "1",
    "randomLast": "5",
    "randomUnits": "seconds",
    "drop": true,
    "allowrate": false,
    "outputs": 1,
    "x": 580,
    "y": 460,
    "wires": [
      [
        "bce4eed3.35b51",
        "856cc7760c7e4e67"
      ]
    ]
  },
  {
    "id": "409336d1.5be148",
    "type": "range",

```

```

    "z": "1b6d4b38.cc6c95",
    "minin": "400",
    "maxin": "600",
    "minout": "100",
    "maxout": "0",
    "action": "scale",
    "round": true,
    "property": "payload",
    "name": "",
    "x": 480,
    "y": 180,
    "wires": [
      [
        "640c0a49.a92a94",
        "d306f048.27de7",
        "af69006.4bbb",
        "61b58ef8f287838e",
        "1cd8ab3502830d52",
        "53a37a51613a5bb6",
        "6e97ed6998b21f63"
      ]
    ]
  },
  {
    "id": "6e9041ab.6698a",
    "type": "ui_button",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "group": "a4a65d36.51d68",
    "order": 6,
    "width": 0,
    "height": 0,
    "passthru": false,
    "label": "water now!",
    "tooltip": "",
    "color": "",
    "bgcolor": "",
    "className": "",
    "icon": "",
    "payload": "1",
    "payloadType": "num",
    "topic": "",
    "topicType": "str",
    "x": 445.5,
    "y": 521,
    "wires": [
      [
        "f137eb44.ee9a48"
      ]
    ]
  },
  {

```

```

    "id": "442146e2.5be678",
    "type": "ui_switch",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "label": "Plant 1 Watering",
    "tooltip": "",
    "group": "a4a65d36.51d68",
    "order": 2,
    "width": "6",
    "height": "1",
    "passthru": true,
    "decouple": "false",
    "topic": "",
    "topicType": "str",
    "style": "",
    "onvalue": "1",
    "onvalueType": "num",
    "onicon": "",
    "oncolor": "",
    "offvalue": "0",
    "offvalueType": "num",
    "officon": "",
    "offcolor": "",
    "animate": true,
    "className": "",
    "x": 880,
    "y": 520,
    "wires": [
      []
    ]
  },
  {
    "id": "2c4908e9.865a98",
    "type": "ui_text",
    "z": "1b6d4b38.cc6c95",
    "group": "a4a65d36.51d68",
    "order": 4,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "Threshold value",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "className": "",
    "x": 800,
    "y": 580,
    "wires": []
  },
  {
    "id": "d306f048.27de7",
    "type": "ui_gauge",
    "z": "1b6d4b38.cc6c95",

```

```

"name": "",
"group": "a4a65d36.51d68",
"order": 1,
"width": "0",
"height": "0",
"gtype": "gage",
"title": "Moisture Level",
"label": "moisture",
"format": "{{value}}",
"min": 0,
"max": "100",
"colors": [
  "#9f0003",
  "#e6e600",
  "#0aff00"
],
"seg1": "",
"seg2": "",
"diff": false,
"className": "",
"x": 860,
"y": 260,
"wires": []
},
{
  "id": "bce4eed3.35b51",
  "type": "ui_chart",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "group": "a4a65d36.51d68",
  "order": 5,
  "width": 0,
  "height": 0,
  "label": "moisture history",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm",
  "interpolate": "linear",
  "nodata": "",
  "dot": false,
  "ymin": "0",
  "ymax": "100",
  "removeOlder": "1",
  "removeOlderPoints": "1440",
  "removeOlderUnit": "86400",
  "cutout": 0,
  "useOneColor": false,
  "useUTC": false,
  "colors": [
    "#1f77b4",
    "#aec7e8",
    "#ff7f0e",

```

```

        "#2ca02c",
        "#98df8a",
        "#d62728",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "outputs": 1,
    "useDifferentColor": false,
    "className": "",
    "x": 770,
    "y": 469,
    "wires": [
        []
    ]
},
{
    "id": "81dd3b03.edb5f8",
    "type": "ui_slider",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "label": "Set Threshold",
    "tooltip": "",
    "group": "a4a65d36.51d68",
    "order": 3,
    "width": "0",
    "height": "0",
    "passthru": true,
    "outs": "all",
    "topic": "threshold",
    "topicType": "str",
    "min": 0,
    "max": "100",
    "step": "",
    "className": "",
    "x": 454.5,
    "y": 568,
    "wires": [
        [
            "d4bf05ce.fe0c08",
            "2c4908e9.865a98",
            "d69a0a41a9c75dc2",
            "d52874b3c4ee17a9",
            "25616d4c2dc0bb0a"
        ]
    ]
},
{
    "id": "4a22d5b5.db323c",
    "type": "smooth",
    "z": "1b6d4b38.cc6c95",
    "name": "",

```

```

    "property": "payload",
    "action": "mean",
    "count": "5",
    "round": "",
    "reduce": false,
    "x": 300,
    "y": 180,
    "wires": [
      [
        "409336d1.5be148"
      ]
    ]
  },
  {
    "id": "61b58ef8f287838e",
    "type": "switch",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "property": "payload",
    "propertyType": "msg",
    "rules": [
      {
        "t": "btwn",
        "v": "0",
        "vt": "num",
        "v2": "90",
        "v2t": "num"
      },
      {
        "t": "gte",
        "v": "90",
        "vt": "num"
      }
    ],
    "checkall": "true",
    "repair": false,
    "outputs": 2,
    "x": 750,
    "y": 820,
    "wires": [
      [
        "b7c023aba908cb29"
      ],
      [
        "2c92fed87924727e"
      ]
    ]
  },
  {
    "id": "b7c023aba908cb29",
    "type": "change",
    "z": "1b6d4b38.cc6c95",

```

```

"name": "Norma",
"rules": [
  {
    "t": "set",
    "p": "payload",
    "pt": "msg",
    "to": "HOPMA",
    "tot": "str"
  },
  {
    "t": "set",
    "p": "color",
    "pt": "msg",
    "to": "GREEN",
    "tot": "str"
  }
],
"action": "",
"property": "",
"from": "",
"to": "",
"reg": false,
"x": 870,
"y": 800,
"wires": [
  [
    "4535c911cf2d17c1",
    "412286cdd41e38f7",
    "18560abd511367dd",
    "01f64467a9623549"
  ]
]
},
{
  "id": "2c92fed87924727e",
  "type": "change",
  "z": "1b6d4b38.cc6c95",
  "name": "Alarm",
  "rules": [
    {
      "t": "set",
      "p": "payload",
      "pt": "msg",
      "to": "АВАРИЯ",
      "tot": "str"
    },
    {
      "t": "set",
      "p": "color",
      "pt": "msg",
      "to": "RED",
      "tot": "str"
    }
  ]
}

```

```

    }
  ],
  "action": "",
  "property": "",
  "from": "",
  "to": "",
  "reg": false,
  "x": 870,
  "y": 840,
  "wires": [
    [
      "4535c911cf2d17c1",
      "f3b0ace3ba91c281",
      "412286cdd41e38f7",
      "18560abd511367dd",
      "01f64467a9623549"
    ]
  ]
},
{
  "id": "4535c911cf2d17c1",
  "type": "ui_button",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "group": "a4a65d36.51d68",
  "order": 4,
  "width": "2",
  "height": "2",
  "passthru": false,
  "label": "{{msg.payload}}",
  "tooltip": "",
  "color": "",
  "bgcolor": "{{msg.color}}",
  "className": "",
  "icon": "",
  "payload": "",
  "payloadType": "str",
  "topic": "topic",
  "topicType": "msg",
  "x": 1150,
  "y": 780,
  "wires": [
    []
  ]
},
{
  "id": "f3b0ace3ba91c281",
  "type": "e-mail",
  "z": "1b6d4b38.cc6c95",
  "server": "smtp.gmail.com",
  "port": "465",
  "secure": true,

```

```

    "tls": true,
    "name": "atlanta555439@gmail.com",
    "dname": "",
    "x": 1100,
    "y": 880,
    "wires": []
  },
  {
    "id": "9a9bb794e5ab27ff",
    "type": "inject",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "",
    "crontab": "",
    "once": true,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "50",
    "payloadType": "num",
    "x": 250,
    "y": 580,
    "wires": [
      [
        "81dd3b03.edb5f8"
      ]
    ]
  },
  {
    "id": "d69a0a41a9c75dc2",
    "type": "function",
    "z": "1b6d4b38.cc6c95",
    "name": "To text",
    "func": "msg.payload = msg.payload.toString();\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 790,
    "y": 620,
    "wires": [
      [
        "7fc5fc979b5bb429",

```

```

        "25616d4c2dc0bb0a"
    ]
}
{
    "id": "7fc5fc979b5bb429",
    "type": "ui_audio",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "group": "a4a65d36.51d68",
    "voice": "urn:moz-tts:sapi:Microsoft Irina Desktop - Russian?ru-RU",
    "always": "",
    "x": 960,
    "y": 600,
    "wires": []
},
{
    "id": "d1a2f10b09c2a622",
    "type": "debug",
    "z": "1b6d4b38.cc6c95",
    "name": "msg.payload",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 430,
    "y": 840,
    "wires": []
},
{
    "id": "e1f53e20a57fca3f",
    "type": "change",
    "z": "1b6d4b38.cc6c95",
    "name": "Signal write",
    "rules": [
        {
            "t": "set",
            "p": "payload",
            "pt": "msg",
            "to": "payload[0]",
            "tot": "msg"
        }
    ],
    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,

```

```

"x": 430,
"y": 900,
"wires": [
  [
    "4a22d5b5.db323c"
  ]
]
},
{
  "id": "e1b5b2a82e9120a4",
  "type": "function",
  "z": "1b6d4b38.cc6c95",
  "name": "INT TO ARINT",
  "func": "msg.payload = [msg.payload];\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 300,
  "y": 1040,
  "wires": [
    [
      "43cd740f54dfc2ac"
    ]
  ]
},
{
  "id": "43cd740f54dfc2ac",
  "type": "modbustcp-write",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "topic": "",
  "dataType": "HoldingRegisters",
  "adr": "0",
  "server": "9b5be17afec189c0",
  "x": 510,
  "y": 1040,
  "wires": []
},
{
  "id": "25d213c42403b74f",
  "type": "modbustcp-read",
  "z": "1b6d4b38.cc6c95",
  "name": "Modbus read",
  "topic": "",
  "dataType": "HoldingRegister",
  "adr": "0",
  "quantity": "10",
  "rate": "5",
  "rateUnit": "s",
  "server": "9b5be17afec189c0",

```

```

    "ieeeType": "off",
    "ieeeBE": true,
    "x": 230,
    "y": 840,
    "wires": [
      [
        "d1a2f10b09c2a622",
        "e1f53e20a57fca3f"
      ]
    ]
  },
  {
    "id": "7d0687c13844195f",
    "type": "inject",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "1",
    "crontab": "",
    "once": true,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "590",
    "payloadType": "num",
    "x": 130,
    "y": 1040,
    "wires": [
      [
        "e1b5b2a82e9120a4"
      ]
    ]
  },
  {
    "id": "9ffaac3cbaf2aa7a",
    "type": "telegram receiver",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "bot": "3d92383986361339",
    "saveDataDir": "",
    "filterCommands": false,
    "x": 150,
    "y": 1200,
    "wires": [
      [

```

```

        "7639d1ac703f8553"
    ],
    []
]
},
{
    "id": "7639d1ac703f8553",
    "type": "function",
    "z": "1b6d4b38.cc6c95",
    "name": "function 9",
    "func": "let txtmsg = \"/start - показати базові команди\r\n\"; \ntxtmsg += \"/sp -  
виставлення уставок\r\n\";\n\nmsg.payload.content = txtmsg;\nreturn msg;\n\n",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 380,
    "y": 1200,
    "wires": [
        [
            "90485aa33f13044e"
        ]
    ]
},
{
    "id": "ab777eeca194af4",
    "type": "telegram sender",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "bot": "3d92383986361339",
    "haserroroutput": false,
    "outputs": 1,
    "x": 610,
    "y": 1380,
    "wires": [
        []
    ]
},
{
    "id": "094187bf5a67d8c1",
    "type": "debug",
    "z": "1b6d4b38.cc6c95",
    "name": "debug 9",
    "active": false,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",

```

```

    "x": 600,
    "y": 1440,
    "wires": []
  },
  {
    "id": "63550b49f298acc5",
    "type": "inject",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "1",
    "crontab": "",
    "once": true,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 270,
    "y": 1480,
    "wires": [
      []
    ]
  },
  {
    "id": "b7fbc692865743fc",
    "type": "change",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "rules": [
      {
        "t": "set",
        "p": "norma",
        "pt": "global",
        "to": "90",
        "tot": "num"
      }
    ],
    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,
    "x": 480,
    "y": 1480,

```

```

    "wires": [
      []
    ]
  },
  {
    "id": "ef2c6318b19e288b",
    "type": "inject",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "5",
    "crontab": "",
    "once": true,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 250,
    "y": 1380,
    "wires": [
      [
        "7dd24966cb178f6b"
      ]
    ]
  },
  {
    "id": "61db4dc8e544b878",
    "type": "telegram sender",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "bot": "3d92383986361339",
    "haserroroutput": false,
    "outputs": 1,
    "x": 770,
    "y": 1620,
    "wires": [
      []
    ]
  },
  {
    "id": "e16edd226e407b07",
    "type": "function",
    "z": "1b6d4b38.cc6c95",
    "name": "function 10",

```

```

"func": "context.global.keyboard = {pending : true}; let opts = {\nreply_to_message_id:
msg.payload.messageId, reply_markup: JSON.stringify({\nkeyboard:
[\n['/vol'],\n['/temp'],\n['відміна']], 'resize_keyboard' : true, 'one_time_keyboard' :
true\n})\n});\nmsg.payload.content = 'Виберіть змінну'; \nmsg.payload.options = opts;\nreturn
[msg];\n",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 460,
  "y": 1620,
  "wires": [
    [
      "61db4dc8e544b878"
    ]
  ]
},
{
  "id": "5aa065e3c22cfc4e",
  "type": "change",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "rules": [
    {
      "t": "set",
      "p": "keyboard.pending",
      "pt": "global",
      "to": "false",
      "tot": "bool"
    }
  ],
  "action": "",
  "property": "",
  "from": "",
  "to": "",
  "reg": false,
  "x": 500,
  "y": 1680,
  "wires": [
    []
  ]
},
{
  "id": "446a0fe38e8128ea",
  "type": "change",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "rules": [
    {
      "t": "set",
      "p": "payload.content",

```

```

        "pt": "msg",
        "to": "Введіть значення уставки змінної vol",
        "tot": "str"
    }
],
"action": "",
"property": "",
"from": "",
"to": "",
"reg": false,
"x": 490,
"y": 1740,
"wires": [
    [
        "61db4dc8e544b878"
    ]
]
},
{
    "id": "9016ce6780c61932",
    "type": "change",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "rules": [
        {
            "t": "set",
            "p": "norma",
            "pt": "global",
            "to": "payload.content",
            "tot": "msg"
        }
    ],
    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,
    "x": 460,
    "y": 1800,
    "wires": [
        []
    ]
},
{
    "id": "52d9a7368c4b0c62",
    "type": "change",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "rules": [
        {
            "t": "set",
            "p": "payload.content",

```

```

    "pt": "msg",
    "to": "Введіть значення уставки змінної temp",
    "tot": "str"
  }
],
"action": "",
"property": "",
"from": "",
"to": "",
"reg": false,
"x": 490,
"y": 1860,
"wires": [
  [
    "61db4dc8e544b878"
  ]
]
},
{
  "id": "496498c86196b635",
  "type": "change",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "rules": [
    {
      "t": "set",
      "p": "norma2",
      "pt": "global",
      "to": "payload.content",
      "tot": "msg"
    }
  ],
  "action": "",
  "property": "",
  "from": "",
  "to": "",
  "reg": false,
  "x": 470,
  "y": 1920,
  "wires": [
    []
  ]
},
{
  "id": "7dd24966cb178f6b",
  "type": "function",
  "z": "1b6d4b38.cc6c95",
  "name": "function 11",
  "func": "let rtdb = global.get(\"rtdb\") || {};\nlet alm = global.get(\"alm\") || { //система\n    тривог\n    state: { radHI: false, valHI: false }, //стан тривог\n\n    norma =\n    global.get(\"norma\") || {};\nlet vhod = global.get(\"vhod\") || {};\n//let vhod = 11;\nlet vhod2 =\n1\nlet state = false;\nlet state2 = false;\n\nlet norma2 = 2;\nlet almmmsg = \"\";\n//умова

```

```

тривоги активна?\nlet alm_radHI = vnod > norma;\nlet alm_valHI = vnod2 > norma2;\n//якщо
тривога тільки активувалася \nif (alm_radHI && !state) {\n  almmmsg += \"Вологість в
приміщенні дорівнює \" + vnod + \"% , що вище заданого значення \" + norma +
\"%\r\n\";\n}\nif (alm_valHI && !state2) {\n  almmmsg += \"Синус дорівнює \" + vnod2 + \" ,
що вище заданого значення \" + norma2 + \"%\r\n\";\n}\n//запис у стан тривоги \nstate =
alm_radHI;\nstate2 = alm_valHI;\n//збереження станів тривоги в глобальному контексті
\nglobal.set(\"alm\", alm);\n//якщо хоча виникла хоча б одна тривога - відправка
повідомлень \nif (almmmsg.length > 1) {\n  msg.payload = {\n    chatId: \"627794772\", //тут
має бути ваш ідентифікатор \n    type: \"message\", \n    content: almmmsg\n  }\n  return
msg;\n}\n\n\",

```

```

  \"outputs\": 1,
  \"noerr\": 0,
  \"initialize\": \"\",
  \"finalize\": \"\",
  \"libs\": [],
  \"x\": 420,
  \"y\": 1400,
  \"wires\": [
    [
      \"094187bf5a67d8c1\",
      \"ab777eeca194af4\"
    ]
  ]
},
{
  \"id\": \"90485aa33f13044e\",
  \"type\": \"telegram sender\",
  \"z\": \"1b6d4b38.cc6c95\",
  \"name\": \"\",
  \"bot\": \"3d92383986361339\",
  \"haserroroutput\": false,
  \"outputs\": 1,
  \"x\": 570,
  \"y\": 1200,
  \"wires\": [
    []
  ]
},
{
  \"id\": \"6e97ed6998b21f63\",
  \"type\": \"change\",
  \"z\": \"1b6d4b38.cc6c95\",
  \"name\": \"\",
  \"rules\": [
    {
      \"t\": \"set\",
      \"p\": \"vhod\",
      \"pt\": \"global\",
      \"to\": \"payload\",
      \"tot\": \"msg\"
    }
  ]
},

```

```

    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,
    "x": 440,
    "y": 240,
    "wires": [
      []
    ]
  },
  {
    "id": "ae64b9df5e2cddb0",
    "type": "inject",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "5",
    "crontab": "",
    "once": true,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 130,
    "y": 320,
    "wires": [
      [
        "13c433f439aca316"
      ]
    ]
  },
  {
    "id": "bd85859f6f12f344",
    "type": "change",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "rules": [
      {
        "t": "set",
        "p": "norma",
        "pt": "global",
        "to": "payload",
        "tot": "msg"
      }
    ]
  }

```

```

    }
  ],
  "action": "",
  "property": "",
  "from": "",
  "to": "",
  "reg": false,
  "x": 1240,
  "y": 840,
  "wires": [
    []
  ]
},
{
  "id": "d2af35530c3b10c1",
  "type": "telegram command",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "command": "/sp",
  "description": "",
  "registercommand": false,
  "language": "",
  "scope": "default",
  "bot": "3d92383986361339",
  "strict": false,
  "hasresponse": true,
  "userregex": false,
  "removeregexcommand": false,
  "outputs": 2,
  "x": 250,
  "y": 1640,
  "wires": [
    [
      "e16edd226e407b07"
    ],
    [
      "5aa065e3c22cfc4e"
    ]
  ]
},
{
  "id": "6a673f479898461e",
  "type": "telegram command",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "command": "/vol",
  "description": "",
  "registercommand": false,
  "language": "",
  "scope": "default",
  "bot": "3d92383986361339",
  "strict": false,

```

```

    "hasresponse": true,
    "useregex": false,
    "removeregexcommand": false,
    "outputs": 2,
    "x": 250,
    "y": 1760,
    "wires": [
      [
        "446a0fe38e8128ea"
      ],
      [
        "9016ce6780c61932"
      ]
    ]
  },
  {
    "id": "fba8cd2fec27326a",
    "type": "telegram command",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "command": "/temp",
    "description": "",
    "registercommand": false,
    "language": "",
    "scope": "default",
    "bot": "3d92383986361339",
    "strict": false,
    "hasresponse": true,
    "useregex": false,
    "removeregexcommand": false,
    "outputs": 2,
    "x": 250,
    "y": 1880,
    "wires": [
      [
        "52d9a7368c4b0c62"
      ],
      [
        "496498c86196b635"
      ]
    ]
  },
  {
    "id": "872e9894792c5817",
    "type": "function",
    "z": "1b6d4b38.cc6c95",
    "name": "Toggle 0/1 on input",
    "func": "\ncontext.state = context.state || 0;\n\n(context.state == 0) ? context.state = 1 :
context.state = 0;\nmsg.payload = context.state;\n\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",

```

```

    "finalize": "",
    "libs": [],
    "x": 510,
    "y": 2040,
    "wires": [
      [
        "097f3d56c9e4e731",
        "847c6ce9f727e2f0"
      ]
    ]
  },
  {
    "id": "097f3d56c9e4e731",
    "type": "debug",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "active": false,
    "x": 710,
    "y": 2020,
    "wires": []
  },
  {
    "id": "d0948f521a19113b",
    "type": "inject",
    "z": "1b6d4b38.cc6c95",
    "name": "tick every 1 sec",
    "repeat": "1",
    "crontab": "",
    "once": false,
    "topic": "",
    "payload": "",
    "x": 290,
    "y": 2040,
    "wires": [
      [
        "872e9894792c5817"
      ]
    ]
  },
  {
    "id": "847c6ce9f727e2f0",
    "type": "rpi-gpio out",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "pin": "4",
    "set": false,
    "freq": "",
    "out": "out",
    "bcm": true,
    "x": 690,
    "y": 2080,
    "wires": []
  }

```

```

},
{
  "id": "e32d1568b4c9c74f",
  "type": "inject",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "props": [
    {
      "p": "payload"
    },
    {
      "p": "topic",
      "vt": "str"
    }
  ],
  "repeat": "60",
  "crontab": "",
  "once": false,
  "onceDelay": 0.1,
  "topic": "",
  "payload": "",
  "payloadType": "date",
  "x": 310,
  "y": 2220,
  "wires": [
    [
      "a6cf0bac567788f5"
    ]
  ]
},
{
  "id": "1ac0062f32c5cc35",
  "type": "debug",
  "z": "1b6d4b38.cc6c95",
  "name": "debug 15",
  "active": false,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "payload",
  "targetType": "msg",
  "statusVal": "",
  "statusType": "auto",
  "x": 660,
  "y": 2220,
  "wires": []
},
{
  "id": "a6cf0bac567788f5",
  "type": "function",
  "z": "1b6d4b38.cc6c95",
  "name": "function 12",

```

```

"func": "let now = new Date();//сьогодні\n//trend - 2-мірний масив 3x60, на 60 останніх
значень\n//читання з контексту потоку або ініціалізація масиву-буферу\nlet trend =
flow.get(\"trend\") || [[now.getTime(),[0],[0]]; \nlet l = trend[0].length; //довжина масиву\n//з
кожним викликом збільшуємо градуси в радіанах\nlet out = global.get(\"out\") || {}; \nlet rad
= out; //Math.PI/trend[1][l-1]\n//if (rad >= Math.PI * 2) rad = 0; //обнуляємо після повного
кола\nlet val = Math.sin (rad);\n//вираховуємо синус\n//на \"верх\" масиву додаємо
елементи \ntrend[0].push (now.toLocaleString());//дата час \ntrend[1].push (rad);\n//радіани
\ntrend[2].push (val);\n//значення синусу\n//якщо масив заповнився до 60\nif (trend[0].length
>59)\n{trend[0].shift();//вилучаємо перший (найстаріший) елемент \ntrend[1].shift
();\ntrend[2].shift();\n//після цього масиви повинні зменшитися на 1 елемент (60)\n//і
зсунутися вниз\n//у випадку, якщо раптом елементів більше 60\n//наприклад були
додані випадково стороннім кодом\n//зробити кількість елементів =60 \ntrend[0].length
= 59;\n// \ntrend[1].length = 59;\ntrend[2].length = 59;\n}\nflow.set(\"trend\", trend);//записати в
контекст потоку\n\nlet rtdb = { //глобальна змінна \ntrend: trend, rad: rad, val:
val\n}\n\nglobal.set(\"rtdb\", rtdb);\n\nreturn msg;\",
  \"outputs\": 1,
  \"noerr\": 0,
  \"initialize\": \"\",
  \"finalize\": \"\",
  \"libs\": [],
  \"x\": 480,
  \"y\": 2220,
  \"wires\": [
    [
      \"1ac0062f32c5cc35\"
    ]
  ]
},
{
  \"id\": \"12b1e6a172616b9a\",
  \"type\": \"inject\",
  \"z\": \"1b6d4b38.cc6c95\",
  \"name\": \"\",
  \"props\": [
    {
      \"p\": \"payload\"
    },
    {
      \"p\": \"topic\",
      \"vt\": \"str\"
    }
  ],
  \"repeat\": \"3540\",
  \"crontab\": \"\",
  \"once\": false,
  \"onceDelay\": 0.1,
  \"topic\": \"\",
  \"payload\": \"trend\",
  \"payloadType\": \"flow\",
  \"x\": 310,
  \"y\": 2280,
  \"wires\": [

```

```

    [
      "acec2dfe02233552"
    ]
  ],
  {
    "id": "049611894fd004d4",
    "type": "debug",
    "z": "1b6d4b38.cc6c95",
    "name": "debug 16",
    "active": false,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "false",
    "statusVal": "",
    "statusType": "auto",
    "x": 660,
    "y": 2280,
    "wires": []
  },
  {
    "id": "acec2dfe02233552",
    "type": "GSheet",
    "z": "1b6d4b38.cc6c95",
    "creds": "95c93f8506a16e72",
    "method": "update",
    "action": "",
    "sheet": "1UQOik4gTSI2IlcZX33qSvlnWRgVqjedItBGEzapSkhE",
    "cells": "Лист1!R1C2:R3C60",
    "flatten": false,
    "name": "",
    "x": 490,
    "y": 2280,
    "wires": [
      [
        "049611894fd004d4"
      ]
    ]
  },
  {
    "id": "13c433f439aca316",
    "type": "function",
    "z": "1b6d4b38.cc6c95",
    "name": "Sin generator",
    "func": "var a;\nna = msg.payload/10000;\nmsg.payload = (Math.sin (a) + 1.0)*100.0 +
380\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],

```

```

"x": 140,
"y": 240,
"wires": [
  [
    "4a22d5b5.db323c"
  ]
]
},
{
  "id": "412286cdd41e38f7",
  "type": "ui_button",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "group": "a4a65d36.51d68",
  "order": 4,
  "width": "2",
  "height": "2",
  "passthru": false,
  "label": "{{msg.payload}}",
  "tooltip": "",
  "color": "",
  "bgcolor": "{{msg.color}}",
  "className": "",
  "icon": "",
  "payload": "",
  "payloadType": "str",
  "topic": "topic",
  "topicType": "msg",
  "x": 1330,
  "y": 780,
  "wires": [
    []
  ]
},
{
  "id": "18560abd511367dd",
  "type": "ui_button",
  "z": "1b6d4b38.cc6c95",
  "name": "",
  "group": "a4a65d36.51d68",
  "order": 4,
  "width": "2",
  "height": "2",
  "passthru": false,
  "label": "{{msg.payload}}",
  "tooltip": "",
  "color": "",
  "bgcolor": "{{msg.color}}",
  "className": "",
  "icon": "",
  "payload": "",
  "payloadType": "str",

```

```

    "topic": "topic",
    "topicType": "msg",
    "x": 1470,
    "y": 780,
    "wires": [
      []
    ]
  },
  {
    "id": "1cd8ab3502830d52",
    "type": "change",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "rules": [
      {
        "t": "set",
        "p": "out",
        "pt": "global",
        "to": "payload",
        "tot": "msg"
      }
    ],
    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,
    "x": 690,
    "y": 220,
    "wires": [
      []
    ]
  },
  {
    "id": "53a37a51613a5bb6",
    "type": "mqtt out",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "Gauge",
    "qos": "2",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "c331a3972af37052",
    "x": 710,
    "y": 280,
    "wires": []
  },
  {

```

```

    "id": "61e9a0160b01beab",
    "type": "mqtt out",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "switch",
    "qos": "2",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "c331a3972af37052",
    "x": 970,
    "y": 460,
    "wires": []
  },
  {
    "id": "d52874b3c4ee17a9",
    "type": "mqtt out",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "slider",
    "qos": "2",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "c331a3972af37052",
    "x": 1070,
    "y": 560,
    "wires": []
  },
  {
    "id": "b18936f6be8c6433",
    "type": "mqtt in",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "slider",
    "qos": "2",
    "datatype": "auto-detect",
    "broker": "c331a3972af37052",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 230,
    "y": 480,
    "wires": [
      [

```

```

        "81dd3b03.edb5f8"
    ]
]
},
{
    "id": "25616d4c2dc0bb0a",
    "type": "mqtt out",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "value",
    "qos": "2",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "c331a3972af37052",
    "x": 1070,
    "y": 520,
    "wires": []
},
{
    "id": "01f64467a9623549",
    "type": "mqtt out",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "status",
    "qos": "2",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "c331a3972af37052",
    "x": 1010,
    "y": 740,
    "wires": []
},
{
    "id": "856cc7760c7e4e67",
    "type": "mqtt out",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "graph",
    "qos": "2",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",

```

```

    "expiry": "",
    "broker": "c331a3972af37052",
    "x": 750,
    "y": 400,
    "wires": []
  },
  {
    "id": "945cbc88df557147",
    "type": "mqtt in",
    "z": "1b6d4b38.cc6c95",
    "name": "",
    "topic": "now",
    "qos": "2",
    "datatype": "auto-detect",
    "broker": "c331a3972af37052",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 230,
    "y": 660,
    "wires": [
      [
        "f137eb44.ee9a48"
      ]
    ]
  },
  {
    "id": "a462fba5e09b9a4e",
    "type": "ui_spacer",
    "z": "1b6d4b38.cc6c95",
    "name": "spacer",
    "group": "a4a65d36.51d68",
    "order": 1,
    "width": 1,
    "height": 1
  },
  {
    "id": "e889b9681bc732f1",
    "type": "ui_spacer",
    "z": "1b6d4b38.cc6c95",
    "name": "spacer",
    "group": "a4a65d36.51d68",
    "order": 1,
    "width": 1,
    "height": 1
  },
  {
    "id": "a4a65d36.51d68",
    "type": "ui_group",
    "name": "PiPlant Control",
    "tab": "9b452f14.609d3",

```

```

    "order": 1,
    "disp": true,
    "width": "6",
    "collapse": false,
    "className": ""
  },
  {
    "id": "9b5be17afec189c0",
    "type": "modbustcp-server",
    "name": "PLCSIM",
    "host": "127.0.0.1",
    "port": "502",
    "unit_id": "1",
    "reconnecttimeout": "10"
  },
  {
    "id": "3d92383986361339",
    "type": "telegram bot",
    "botname": "RBI Kakoieva Alana",
    "usernames": "",
    "chatids": "",
    "baseapiurl": "",
    "updatemode": "polling",
    "pollinterval": "300",
    "usesocks": false,
    "sockshost": "",
    "socksprotocol": "socks5",
    "socksport": "6667",
    "socksusername": "anonymous",
    "sockspassword": "",
    "bothost": "",
    "botpath": "",
    "localbotport": "8443",
    "publicbotport": "8443",
    "privatekey": "",
    "certificate": "",
    "useselfsignedcertificate": false,
    "sslterminated": false,
    "verboselogging": false
  },
  {
    "id": "95c93f8506a16e72",
    "type": "gauth",
    "name": "rpi-service-account@stunning-vertex-371113.iam.gserviceaccount.com"
  },
  {
    "id": "c331a3972af37052",
    "type": "mqtt-broker",
    "name": "Mosquitto",
    "broker": "http://test.mosquitto.org",
    "port": "1883",
    "clientid": ""
  }

```

```

    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "4",
    "keepalive": "60",
    "cleansession": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closeQos": "0",
    "closePayload": "",
    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  },
  {
    "id": "9b452f14.609d3",
    "type": "ui_tab",
    "name": "PiPlant",
    "icon": "dashboard",
    "order": 1
  }
]

```