

КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб керування автоматичним поливом кімнатних рослин
на основі Arduino Cloud

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ 240494.22.04.10 ПЗ

Виконав здобувач ІV курсу, група КІ2-22-4


Підпис

Назар СТРИЛЬЧУК

Ініціали, прізвище

Керівник д-р. техн.наук, проф.
Науковий ступінь, учене звання


Підпис

Олег САВЕНКО

Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц.
Науковий ступінь, учене звання


Підпис

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІС


Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«01» червня 2026 р.

дата

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІС


Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Стрільчуку Назару Михайловичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

Керівник проекту (роботи) Савенко Олег Станіславович, д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз предметної області та огляд відомих рішень

Проектування програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

Тестування та реалізація програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Архітектура програмно-технічного засобу

Схема електрична

Блок-схема алгоритму функціонування мікроконтролера ESP32

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання при

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Прим
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	викон
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	викон
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	викон
4	Робота над розділом 2 – проектування архітектури програмно-технічного засобу	01.04.2026	викон
5	Робота над розділом 3 – реалізація та тестування програмно-технічного засобу	29.04.2026	викон
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	викон
7	Попередній захист ВКР	25.05.2026	викон
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач

Підпис

Назар СТРІЛЬЧУК

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи

Підпис

Олег САВЕНКО

Імя, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud».

Автор роботи: Назар СТРІЛЬЧУК.

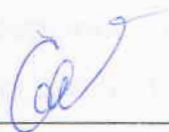
Керівник роботи: Олег САВЕНКО.

Пояснювальна записка: 61 с., 28 рис., 3 табл., 3 дод., 52 джерел.

Графічна частина: 3 креслення.

ВИМІРЮВАННЯ ВОЛОГОСТІ ҐРУНТУ, ESP32, ARDUINO CLOUD.

Використання сучасних мікроконтролерних платформ та хмарних сервісів дозволяє реалізувати системи автоматичного поливу, здатні самостійно контролювати стан ґрунту та виконувати необхідні дії без постійного втручання користувача. Особливий інтерес становлять рішення на основі платформи Arduino Cloud, яка забезпечує віддалений моніторинг параметрів системи, керування виконавчими пристроями через мережу Інтернет та зберігання даних у хмарному середовищі. Поєднання мікроконтролерів сімейства ESP32 із сервісами Arduino Cloud дозволяє створювати функціональні та доступні системи автоматизації для побутового використання. Це зумовлює актуальність розроблення програмно технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud.





Підпис здобувача

03.06.2026

Дата

ЗМІСТ

Вступ.....	4
1 Аналіз предметної області та огляд відомих рішень.....	6
1.1 Актуальність задачі автоматизації поливу кімнатних рослин	6
1.2 Огляд технологій та протоколів IoT для систем моніторингу і керування	7
1.4 Методи вимірювання вологості ґрунту	14
1.5 Постановка задачі.....	21
2 Проєктування програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud	23
2.1 Аналіз вимог до програмно-технічного засобу.....	23
2.2 Архітектура програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud	25
2.3 Схема електрична.....	29
2.4 Алгоритм функціонування мікроконтролера.....	33
2.5 Програмна реалізація взаємодії мікроконтролера з хмарною платформою	37
2.6 Висновки до другого розділу	41
3 Тестування та реалізація програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud.....	42
3.1 Розгортання середовища в Arduino Cloud	42
3.2 Створення та тестування віртуального прототипу програмно-технічного засобу в Wokwi	49
3.3 Налаштування дашборду та хмарних змінних.....	52
3.4 Практичне застосування та обмеження прототипу програмно-технічного засобу	58
3.5 Висновки до третього розділу.....	59
Висновки	60

КВРКІ. 240494.22.04.10 ПЗ						
Виконав	Назар Стрільчук	Підпис	Дата	Програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud		
Перевір.	Олег Савенко					
Н.контр.	Тетяна КИСІЛЬ		01.06	Літера	Аркуш	Аркушів
Затвер.	Ольга ПАВЛОВА			у	2	67
ХНУ КІ2-22-4						

Перелік джерел посилань	62
Додаток А Архітектура програмно-технічного засобу	68
Додаток Б Схема електрична	69
Додаток В Блок-схема алгоритму функціонування мікроконтролера ESP32	70

					КВРКІ. 240494.22.04.10 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Розвиток технологій IoT та засобів автоматизації побутових процесів сприяє широкому впровадженню інтелектуальних систем моніторингу і керування навколишнім середовищем. Однією з актуальних задач є автоматизація догляду за рослинами, оскільки підтримання оптимального рівня вологості ґрунту потребує регулярного контролю та своєчасного поливу. Відсутність користувача протягом тривалого часу або людський фактор можуть призводити до пересихання ґрунту та погіршення стану рослин.

Використання сучасних мікроконтролерних платформ та хмарних сервісів дозволяє реалізувати системи автоматичного поливу, здатні самостійно контролювати стан ґрунту та виконувати необхідні дії без постійного втручання користувача. Особливий інтерес становлять рішення на основі платформи Arduino Cloud, яка забезпечує віддалений моніторинг параметрів системи, керування виконавчими пристроями через мережу Інтернет та зберігання даних у хмарному середовищі. Поєднання мікроконтролерів сімейства ESP32 із сервісами Arduino Cloud дозволяє створювати функціональні та доступні системи автоматизації для побутового використання. Це зумовлює актуальність розроблення програмно технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud.

Метою дипломної роботи є розроблення та реалізація програмно технічного засобу керування автоматичним поливом кімнатних рослин на основі платформи Arduino Cloud.

Об'єктом дослідження є процес автоматизованого контролю вологості ґрунту та керування поливом кімнатних рослин із використанням технологій Інтернету речей.

Предметом дослідження є методи та засоби побудови програмно технічних систем автоматичного поливу на базі мікроконтролера ESP32, давачів вологості ґрунту, виконавчих механізмів та хмарної платформи Arduino Cloud.

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 4
Зм.	Арк.	№ докum.	Підпис	Дата		

У межах кваліфікаційної роботи планується розроблення програмно технічного засобу, який забезпечуватиме вимірювання рівня вологості ґрунту, автоматичне керування процесом поливу для декількох незалежних зон, передачу даних до хмарного середовища та дистанційний моніторинг параметрів системи.

Реалізація пристрою передбачається на базі мікроконтролера ESP32 із використанням бездротового зв'язку Wi Fi та сервісів Arduino Cloud для організації віддаленого доступу до інформації та керування виконавчими механізмами. Система забезпечуватиме безперервний моніторинг вологості ґрунту за допомогою датчиків вимірювання вологості ґрунту, автоматичний аналіз отриманих даних та прийняття рішень щодо необхідності поливу. Для кожної зони вирощування рослин передбачено незалежне вимірювання рівня вологості та окреме керування насосом подачі води. Використання платформи Arduino Cloud дозволить здійснювати візуалізацію поточних параметрів системи, зберігати історію вимірювань, а також забезпечувати дистанційне керування режимами роботи через вебінтерфейс або мобільний застосунок. Запропоноване рішення поєднує функції автоматичного керування, моніторингу та віддаленого доступу, що підвищує ефективність догляду за кімнатними рослинами та зменшує необхідність постійного втручання користувача в процес поливу.

Практична цінність роботи полягає у створенні програмно технічного засобу автоматичного поливу кімнатних рослин, який дозволяє зменшити участь користувача у процесі догляду за рослинами, забезпечити підтримання оптимального рівня вологості ґрунту, раціонально використовувати водні ресурси та здійснювати дистанційний контроль стану системи через мережу Інтернет.

					КВРКІ. 240494.22.04.10 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ВІДОМИХ РІШЕНЬ

1.1 Актуальність задачі автоматизації поливу кімнатних рослин

Кімнатне рослинництво стрімко набуває популярності як у приватних оселях, так і в офісних та комерційних приміщеннях. Рослини не лише покращують естетику інтер'єру, а й позитивно впливають на мікроклімат приміщення, знижуючи концентрацію вуглекислого газу та збільшуючи відносну вологість повітря. Водночас доглянути за рослинами в умовах напруженого ритму міського життя вдається далеко не завжди: нерегулярний або надмірний полив є однією з найпоширеніших причин загибелі кімнатних рослин. За даними маркетингових досліджень ринку розумного зрошення, глобальний ринок засобів автоматизованого поливу демонструє стаłe зростання і за прогнозами досягне мільярдних обсягів до 2030 року, що свідчить про виражений попит на відповідні рішення.

Проблема неналежного догляду за рослинами є особливо актуальною в контексті тривалих відряджень або відпусток, коли власник не має можливості своєчасно зволожити ґрунт. Традиційний підхід до розв'язання цієї проблеми – доручити полив сусідам або знайомим не є надійним і не дозволяє враховувати індивідуальні потреби різних видів рослин, адже тропічні рослини, суккуленти та трав'яні культури мають принципово різні вимоги до режиму зволоження. Застосування автоматизованих систем поливу на основі IoT-технологій дозволяє вирішити ці проблеми комплексно: система самостійно вимірює вологість ґрунту, приймає рішення про необхідність поливу і сповіщає власника про стан рослин через мобільний застосунок або веб-дашборд.

Концепція Інтернету речей передбачає підключення фізичних пристроїв до мережі Інтернет для обміну даними та дистанційного керування. У контексті автоматизації поливу IoT-підхід дозволяє реалізувати три ключові функціональні складові: безперервний збір даних про стан ґрунту за допомогою датчиків, хмарне зберігання та аналіз цих даних, а також дистанційне керування

					КВРКІ. 240494.22.04.10 ПЗ	Арк.
						6
Зм.	Арк.	№ докum.	Підпис	Дата		

виконавчими механізмами такими як насоси або клапани через веб-інтерфейс. Поєднання цих складових утворює повноцінну кіберфізичну систему, що замикає контур керування від вимірювання до дії.

Мікроконтролерна платформа ESP32 є одним з найпоширеніших рішень для побудови IoT-пристроїв початкового та середнього рівня складності. Завдяки інтегрованому Wi-Fi та Bluetooth-трансиверу, потужному двоядерному процесору та широкому набору периферії ESP32 дозволяє реалізувати повний функціональний стек IoT-пристрою на одному чипі без залучення додаткових мережевих модулів. Платформа Arduino IoT Cloud, у свою чергу, надає готову інфраструктуру для хмарного зберігання змінних, побудови дашбордів та управління розкладами без необхідності розгортати власний сервер. Поєднання цих двох компонентів є логічним і технічно виправданим вибором для розробки прототипу автоматизованої системи поливу кімнатних рослин у рамках бакалаврської дипломної роботи.

1.2 Огляд технологій та протоколів IoT для систем моніторингу і керування

Побудова IoT-системи автоматизованого поливу спирається на ряд базових технологій такі як бездротові протоколи зв'язку, хмарні платформи та стандарти обміну даними. Розуміння особливостей кожного з цих компонентів є необхідним для обґрунтованого вибору архітектурних рішень.

Серед бездротових протоколів зв'язку для IoT-пристроїв найбільшого поширення набули Wi-Fi (стандарт IEEE 802.11), Bluetooth Low Energy (BLE), Zigbee та LoRaWAN. Wi-Fi забезпечує широку смугу пропускання і пряме підключення до існуючої домашньої або офісної мережі без потреби у додатковому шлюзі, що робить його оптимальним вибором для пристроїв, розташованих у межах Wi-Fi покриття. Протокол BLE орієнтований на ультра-малоенергетичні пристрої з короткою дальністю зв'язку та доцільний для датчиків з батарейним живленням, але не підходить для пристроїв, що потребують

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 7
Зм.	Арк.	№ док.ум.	Підпис	Дата		

постійного підключення до хмари. Zigbee та Z-Wave орієнтовані на mesh-мережі «розумного будинку» і потребують координаторного пристрою (хаба). LoRaWAN забезпечує надширокий радіус дії (до десятків кілометрів) при мінімальному енергоспоживанні, але має обмежену пропускну здатність і не підходить для додатків із частою передачею даних.

Протокол MQTT (Message Queuing Telemetry Transport) є стандартом де-факто для IoT-комунікацій поверх TCP/IP. Він базується на моделі «видавець-підписник» (publish-subscribe), де пристрої публікують дані у тематичні канали (топіки) на MQTT-брокері, а клієнти (включаючи веб-дашборди) підписуються на необхідні топіки і отримують оновлення в реальному часі. MQTT характеризується мінімальним накладним навантаженням заголовків пакетів (лише 2 байти), що робить його ефективним для пристроїв з обмеженими ресурсами та нестабільним з'єднанням. Платформа Arduino IoT Cloud використовує MQTT поверх TLS (порт 8883) для захищеного обміну даними між мікроконтролером та хмарним брокером.

Для серіалізації даних в Arduino IoT Cloud застосовується формат CBOR (Concise Binary Object Representation), що є бінарним аналогом JSON, що забезпечує компактне кодування структурованих даних. На відміну від текстового JSON, CBOR дозволяє зменшити розмір пакетів на 20–50%, що суттєво при передачі даних через MQTT з обмеженою пропускну здатністю. HTTP/REST, що широко застосовується у веб-розробці, також використовується в IoT зокрема, платформа ThingSpeak підтримує передачу даних через HTTP GET/POST запити. Однак накладні витрати протоколу HTTP роблять його менш ефективним порівняно з MQTT для пристроїв з частими оновленнями.

Протокол NTP (Network Time Protocol) відіграє важливу роль у системах із розкладним керуванням. ESP32 не має апаратного годинника реального часу (RTC) зі збереженням часу при відключенні живлення, тому для отримання актуального системного часу після підключення до Wi-Fi пристрій звертається до NTP-серверів. Отриманий UNIX-timestamp використовується бібліотекою

ArduinoIoTCloud для коректного визначення активних вікон у розкладі CloudSchedule. Без синхронізації часу через NTP механізм розкладного поливу не може функціонувати коректно.

1. ПОРІВНЯННЯ БЕЗДРОВОТИХ ПРОТОКОЛІВ

Протокол	Wi-Fi (IEEE 802.11)	Bluetooth Low Energy (BLE)	LoRaWAN
Топологія	Інфраструктурна (через роутер)	Точка-точка (з можливістю broadcast)	Зіркова (через шлюз)
Діапазон дії	30–100 м (у приміщенні)	10–30 м	2–15 км (місто) / до 50 км (відкрита місцевість)
Пропускна здатність	Висока (до сотень Мбіт/с)	Низька–середня (до 2 Мбіт/с)	Низька (0.3–50 кбіт/с)
Енергоспоживання	Високе	Дуже низьке	Дуже низьке
Переваги	Широка смуга, просте підключення до Інтернету	Низьке енергоспоживання, для батарейних пристроїв	Великий радіус дії, низьке споживання
Недоліки	Високе споживання, залежність від Wi-Fi	Мала дальність, не для постійного підключення до хмари	Дуже низька швидкість, обмежений обсяг даних
Застосування	Пристрої з постійним живленням та доступом до Інтернету	Датчики, носимі пристрої, батарейні рішення	Моніторинг, IoT рішення на великих відстанях

Рисунок 1.1 – Порівняння бездротових протоколів

1.3 Огляд відомих рішень автоматизованого поливу рослин

Аналіз ринку систем автоматизованого поливу дозволяє виділити два основні класи рішень: комерційні готові пристрої і відкриті DIY-системи на базі мікроконтролерних платформ. Розглянемо найбільш показових представників кожного класу.

Gardena Smart Water Control (19031-20). Це комерційний клапан поливу від відомого виробника садового обладнання Gardena (Husqvarna Group). Пристрій підключається до водопровідного шлангу і керує подачею води через електромагнітний клапан (рис. 1.2).



Рисунок 1.2 – Gardena Smart Water Control (19031-20)

Живлення здійснюється від трьох батарей типу AA напругою 1,5 В, що забезпечує автономну роботу без підключення до електромережі. Зв'язок із хмарою відбувається через фірмовий шлюз GARDENA smart Gateway по протоколу 868 МГц, що є суттєвим обмеженням: без окремо придбаного шлюзу (GARDENA smart Gateway 19000) пристрій не може підключитися до хмарного сервісу. Програмування розкладу поливу здійснюється через мобільний застосунок GARDENA. Хмарний API платформи є відкритим (Husqvarna Developer Portal) і підтримує WebSocket та REST, що дозволяє інтегрувати систему з Home Assistant через стороннє розширення. Орієнтовна ринкова вартість клапана становить близько 12 000 грн без урахування вартості шлюзу, що робить це рішення відносно дорогим для побутового використання. Серед недоліків слід також відзначити відсутність вбудованого датчика вологості ґрунту. Таким чином система керує поливом виключно за розкладом, без урахування реального стану ґрунту.

Іншим рішенням є Xiaomi Mi Flora (Flower Care). Це компактний чотириканальний Bluetooth-датчик від Xiaomi, призначений для встановлення

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

безпосередньо у ґрунт кімнатного горщика (рис. 1.3). Пристрій вимірює вологість ґрунту, освітленість, температуру навколишнього середовища та електропровідність ґрунту (як непрямий показник вмісту поживних речовин). Живлення здійснюється від однієї батареї CR2032, що забезпечує роботу протягом 3–6 місяців. Пристрій передає дані через BLE у фірмовому форматі, для доступу до якого необхідно або використовувати офіційний застосунок Flower Care, або підключити прокси-пристрій на базі ESP32 з ESPHome для ретрансляції даних до Home Assistant. Mi Flora є виключно датчиком і не має жодних засобів керування поливом. Він лише повідомляє користувача про стан рослини, а рішення про полив приймає сам користувач. Вартість одного датчика становить близько 1000 грн., що є доступним рівнем, однак для трьох незалежних горщиків потрібні три окремі пристрої.



Рисунок 1.3 – Xiaomi Mi Flora

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

Іншим варіантом є комбінація ESP32 та Blynk платформи (DIY-система). Платформа Blynk є одним із найпопулярніших хмарних рішень для DIY IoT-проектів, що підтримує понад 400 апаратних платформ. Система автополиву на базі ESP32 та Blynk передбачає підключення ємнісного давача вологості до аналогового входу мікроконтролера, керування насосом через реле та публікацію даних на Blynk.Cloud через протокол MQTT.

Blynk надає готовий мобільний застосунок для iOS та Android з набором налаштовуваних віджетів, таких як Gauge, Button, Chart, Slider. Особливістю Blynk є концепція Virtual Pins, що є програмні канали, які несуть довільний тип даних між пристроєм і дашбордом. Безкоштовний план Blynk Free дозволяє підключити один пристрій і використовувати базовий набір віджетів без часових обмежень, однак накладає обмеження на кількість подій і частоту оновлення даних. Серед недоліків Blynk для завдань розкладного поливу слід відзначити відсутність нативного типу CloudSchedule. У Blynk розклади реалізуються через Timer-таймери або Eventor-тригери, що є менш гнучким порівняно з виділеним типом даних Arduino IoT Cloud.

Ще одним DIY рішенням є комбінація мікроконтролера ESP32 та платформи ThingSpeak. ThingSpeak є IoT-аналітичною платформою компанії MathWorks (розробника MATLAB), що зберігає часові ряди даних від датчиків у окремих каналах. Кожен канал підтримує до восьми полів даних і дозволяє виконувати MATLAB-скрипти безпосередньо над хмарними даними для статистичного аналізу, побудови прогнозів та виявлення аномалій. Для реалізації системи поливу на основі ThingSpeak ESP32 надсилає показання датчиків через HTTP GET або MQTT, а ThingSpeak відображає їх у вигляді графіків.

Однак ThingSpeak є насамперед платформою для аналізу даних, а не для двостороннього керування: передача команд від хмари до пристрою (наприклад, включення насоса) реалізується через React/ThingHTTP і є значно складнішою, ніж у Blynk або Arduino IoT Cloud.

Альтернативним рішенням є використання Arduino IoT Cloud та ESP32. Платформа Arduino IoT Cloud є рідним хмарним рішенням для пристроїв Arduino та сумісних мікроконтролерів, включаючи ESP32. На відміну від Blynk і ThingSpeak, вона надає нативний тип CloudSchedule для реалізації розкладів поливу, вбудовану підтримку двостороннього зв'язку через MQTT/CBOR та автоматичну генерацію коду thingProperties.h на основі визначених у хмарі змінних. Дашборд доступний через веб-браузер та мобільний застосунок Arduino IoT Remote для iOS та Android. Безкоштовний план підтримує до 25 cloud-змінних на одному пристрої, що є достатньо для реалізації трьохканальної системи поливу даної роботи. Відсутність вбудованих обмежень на частоту оновлення (на відміну від ThingSpeak) і наявність типу CloudSchedule визначили вибір цієї платформи як основи для розробленого засобу.

Порівняльний аналіз існуючих рішень автоматизованого поливу наведено у таблиці 1.1.

Аналіз наведеної таблиці свідчить про те, що кожне розглянуте рішення має власні переваги та недоліки. Комерційні пристрої відрізняються якістю виконання і простотою розгортання, але мають закриту архітектуру, залежність від хмарних сервісів виробника та обмежену гнучкість у налаштуванні. DIY-рішення на базі мікроконтролерних платформ є відкритими і гнучкими, але потребують технічних компетенцій для реалізації.

Таблиця 1.1 – Порівняльний аналіз існуючих рішень автоматизованого поливу

Критерій	Gardena Smart Water Control	Mi Flora (Xiaomi)	ESP32 + Blynk	ESP32 + ThingSpeak	ESP32 + Arduino Cloud
Давач вологості	Відсутній	Так (BLE)	Так	Так	Так

Кінець таблиці 1.1

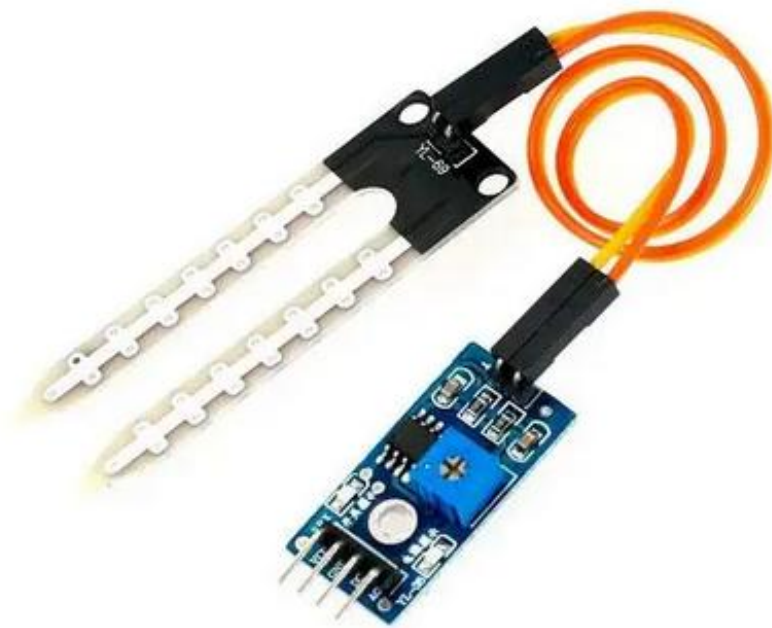
Виконавчий механізм	Клапан	Відсутній	Насос/реле	Насос/реле	Насос/реле
Протокол зв'язку	868 МГц	BLE	Wi-Fi/MQTT	Wi-Fi/HTTP/MQTT	Wi-Fi/MQTT/TLS
Хмарна платформа	Husqvarna Cloud	Xiaomi Cloud	Blynk.Cloud	ThingSpeak	Arduino IoT Cloud
Розклад поливу	Так	Ні	Через Timer	Через React	CloudSchedule
Веб-дашборд	Мобільний застосунок	Мобільний застосунок	Веб + мобільний	Веб	Веб + мобільний
Відкритість рішення	Частково	Частково	Так	Так	Так
Вартість рішення	Висока	Низька	Низька	Безкоштовно	Безкоштовно
Необхідний шлюз	Так (Gateway 19000)	Ні	Ні	Ні	Ні

1.4 Методи вимірювання вологості ґрунту

Вимірювання вологості ґрунту є ключовою функцією системи автоматичного поливу, оскільки саме цей параметр визначає необхідність активації насоса. Існує кілька фізичних методів вимірювання вологості ґрунту, що реалізуються у давачах різних типів.

Резистивні давачі вологості ґрунту є одним із найпоширеніших типів сенсорів, що використовуються у побутових та навчальних системах автоматичного поливу завдяки простоті конструкції, низькій вартості та легкості інтеграції з мікроконтролерними платформами. Конструктивно такий давач

складається з двох провідних електродів, які занурюються в ґрунт і утворюють електричний ланцюг через ґрунтовий розчин. Принцип роботи базується на вимірюванні електричного опору між електродами. При збільшенні вмісту води в ґрунті зростає кількість іонів, здатних переносити електричний заряд, унаслідок чого провідність середовища підвищується, а електричний опір зменшується. При висиханні ґрунту кількість води зменшується, що призводить до збільшення опору між електродами.



Рисуно 1.4 – Резистивний давач вимірювання вологості ґрунту

Більшість резистивних давачів формують аналоговий сигнал, величина якого пропорційна вологості ґрунту. Отримане значення може бути безпосередньо зчитане аналого цифровим перетворювачем мікроконтролера та використане для прийняття рішень щодо необхідності поливу. Завдяки простоті принципу дії такі давачі широко застосовуються у недорогих системах моніторингу стану рослин та навчальних проектах на базі Arduino і ESP32.

Основним недоліком резистивних давачів є низька довговічність, пов'язана з процесом електролітичної корозії електродів. Під час проходження

електричного струму через вологий ґрунт виникають електрохімічні реакції, унаслідок яких матеріал електродів поступово руйнується. Найбільш інтенсивно цей процес проявляється при безперервному живленні датчика постійним струмом. З часом на поверхні електродів утворюються продукти окиснення, що призводить до зміни електричних характеристик сенсора та зниження точності вимірювань. У більшості випадків термін експлуатації дешевих резистивних датчиків становить від кількох тижнів до кількох місяців залежно від складу ґрунту, рівня вологості та режиму роботи.

Ще одним суттєвим недоліком є залежність показань від фізико-хімічних властивостей ґрунту. На результати вимірювання впливають концентрація солей, кислотність, температура та щільність ґрунту. Один і той самий рівень вологості в різних типах ґрунтів може відповідати різним значенням електричного опору, тому для забезпечення прийнятної точності необхідне індивідуальне калібрування. Крім того, внесення добрив або зміна мінерального складу ґрунту може призводити до зміни показань навіть без фактичної зміни вологості.

Для часткового усунення проблеми корозії у сучасних системах автоматизації часто використовують режим періодичного живлення датчика. У такому випадку живлення подається лише на короткий проміжок часу безпосередньо перед виконанням вимірювання, після чого сенсор знову відключається. Це дозволяє суттєво зменшити швидкість руйнування електродів та збільшити термін служби пристрою. Проте навіть за таких умов резистивні датчики поступаються за надійністю та довговічністю емнісним сенсорам вологості ґрунту, які не мають прямого електричного контакту з вимірюваним середовищем.

Незважаючи на зазначені недоліки, резистивні датчики залишаються популярним рішенням для простих та недорогих систем автоматичного поливу, де основними вимогами є мінімальна вартість та простота реалізації. Їх

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 16
Зм.	Арк.	№ докum.	Підпис	Дата		

використання дозволяє реалізувати базовий контроль вологості ґрунту та забезпечити автоматизацію процесу поливу без значних фінансових витрат.

Ємнісні давачі вологості ґрунту є більш сучасним та надійним рішенням порівняно з резистивними сенсорами. Їх принцип роботи базується не на вимірюванні електричного опору ґрунту, а на визначенні його діелектричної проникності, значення якої безпосередньо залежить від кількості вологи. Оскільки діелектрична стала води становить приблизно 80, тоді як для сухого ґрунту цей показник перебуває в межах від 3 до 7, збільшення вмісту води призводить до суттєвої зміни електричних параметрів вимірювального елемента. Це дозволяє отримувати стабільніші та точніші результати вимірювання незалежно від електропровідності ґрунту.

Конструктивно ємнісний давач зазвичай виконується у вигляді друкованої плати, на якій сформовано два провідники спеціальної форми, що утворюють плоский гребінчастий конденсатор. Провідники ізольовані захисним шаром лаку або полімерного покриття, тому не мають безпосереднього контакту з вологою та агресивними компонентами ґрунту. При зануренні сенсора в ґрунт діелектричним середовищем між електродами виступає суміш ґрунту, повітря та води. Зміна співвідношення цих компонентів впливає на ємність конденсатора, яка надалі перетворюється електронною схемою давача у вихідний аналоговий сигнал.

На відміну від резистивних сенсорів, через ґрунт не протікає постійний електричний струм. Для вимірювання використовується високочастотний змінний сигнал, зазвичай із частотою близько 1 МГц. Завдяки цьому повністю усувається проблема електролітичної корозії електродів, яка є основною причиною виходу з ладу резистивних давачів. Відсутність електрохімічних процесів дозволяє значно збільшити термін експлуатації сенсора, який за нормальних умов використання може становити декілька років без помітного погіршення характеристик.

Додатковою перевагою ємнісних давачів є менша залежність результатів вимірювання від концентрації солей та мінерального складу ґрунту. Оскільки вимірюється діелектрична проникність середовища, а не його електропровідність, вплив добрив та інших домішок на показання є значно меншим. Це спрощує процедуру калібрування та підвищує достовірність отриманих даних у різних типах ґрунтів.

Більшість поширених ємнісних сенсорів формують аналоговий вихідний сигнал у межах від 1,2 до 2,8 В при напрузі живлення 3,3 В. Такий діапазон є зручним для використання з мікроконтролером ESP32, оскільки може безпосередньо підключатися до входів аналого цифрового перетворювача без застосування додаткових схем узгодження або масштабування сигналу. Отримані значення легко перетворюються у відсотковий показник вологості за допомогою програмного калібрування.



Рисунок 1.5 – Ємнісний давач вимірювання вологості ґрунту

Завдяки високій надійності, тривалому терміну служби, низькому впливу зовнішніх факторів та сумісності з сучасними мікроконтролерними платформами ємнісні давачі сьогодні вважаються оптимальним вибором для систем автоматичного поливу рослин та інших рішень Інтернету речей, де важливими є довготривала стабільність роботи та мінімальні витрати на технічне обслуговування.

Для розроблюваного програмно технічного засобу автоматичного поливу кімнатних рослин доцільним є використання саме ємнісного давача вологості, що забезпечить надійне вимірювання параметрів ґрунту та тривалу експлуатацію системи без необхідності заміни сенсорів.

Давачі вологості ґрунту на основі технології TDR (Time Domain Reflectometry) належать до класу високоточних професійних вимірювальних приладів, що широко використовуються у наукових дослідженнях, агрономії та системах точного землеробства. Принцип їх роботи базується на аналізі поширення електромагнітного імпульсу вздовж спеціального хвилеводу або металевого зонда, зануреного в ґрунт. Після подачі імпульсу вимірюється час його проходження та відбиття від кінця зонда. Оскільки швидкість поширення електромагнітної хвилі залежить від діелектричної проникності середовища, а діелектрична проникність ґрунту визначається насамперед вмістом води, отримані дані дозволяють з високою точністю обчислити об'ємну вологість ґрунту.

Головною перевагою TDR-давачів є висока точність вимірювання, яка може досягати похибки менше 2–3 %, а також мінімальна залежність результатів від мінерального складу ґрунту, температури та концентрації розчинених солей. Крім того, такі сенсори забезпечують стабільність характеристик протягом тривалого часу та практично не потребують повторного калібрування. Завдяки цьому вони широко використовуються для моніторингу вологості на дослідних ділянках, у тепличних комплексах та системах автоматизованого керування зрошенням великих сільськогосподарських угідь.

Разом із тим реалізація технології TDR потребує використання складних високочастотних електронних схем, генераторів коротких імпульсів та високошвидкісних вимірювальних модулів, що суттєво підвищує вартість обладнання. Вартість одного професійного TDR-сенсора може у десятки разів перевищувати вартість типових ємнісних давачів вологості. Через це їх використання у побутових системах автоматичного поливу кімнатних рослин є економічно недоцільним та технічно надлишковим.

Ще одним типом високоточних засобів контролю вологості є давачі матричного потенціалу, або тензіометри. На відміну від резистивних, ємнісних та TDR-сенсорів, вони не вимірюють безпосередньо вміст води в ґрунті, а визначають силу, з якою вода утримується його частинками. Конструктивно тензіометр складається з герметичної трубки, заповненої водою, пористого керамічного наконечника та вимірювального елемента для контролю тиску або вакууму. Після встановлення в ґрунт вода через пористу мембрану досягає рівноваги з навколишнім середовищем, а створений вакуум дозволяє оцінити доступність вологи для кореневої системи рослин.

Перевагою тензіометрів є висока точність визначення стану зволоження кореневої зони та можливість безпосередньої оцінки умов водозабезпечення рослин. Саме тому вони широко застосовуються у професійному сільському господарстві, системах краплинного зрошення та наукових дослідженнях. Однак такі пристрої потребують регулярного технічного обслуговування, контролю рівня води всередині вимірювальної системи, видалення повітряних бульбашок та періодичного калібрування. Крім того, вони мають значні габарити та вищу вартість порівняно з електронними сенсорами вологості.

Для даної роботи обрано ємнісні давачі вологості з аналоговим виходом, що підключаються до АЦП ESP32. Такий вибір обумовлений їхньою довговічністю, відсутністю корозії, достатньою точністю для завдань автополиву (помилка вимірювання $\pm 3-5\%$) та широкою доступністю й невисокою вартістю (150–250 грн. за одиницю). У симуляторі Wokwi ємнісні давачі замінюються

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

потенціометрами, що дозволяє перевірити логіку алгоритму без фізичних компонентів. Порівняльна характеристика методів вимірювання вологості ґрунту наведена у таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика методів вимірювання вологості ґрунту

Критерій	Резистивний давач	Ємнісний давач	TDR-давач
Принцип дії	Вимірювання опору ґрунту	Вимірювання діелектричної проникності	Відбиття ЕМ-імпульсу у TDR
Корозія електродів	Висока (DC через ґрунт)	Відсутня (AC на 1 МГц)	Відсутня
Точність	±5–10%	±3–5%	±1–2%
Термін служби	2–6 місяців	2–5 років	> 5 років
Вартість	Дуже низька	Низька	Висока
Складність інтеграції	Мінімальна	Мінімальна	Висока
Тип виходу	Аналоговий (напруга)	Аналоговий (напруга)	Цифровий
Застосування у проекті	Не обрано	Обрано	Не обрано

1.5 Постановка задачі

На підставі проведеного аналізу предметної області, огляду існуючих рішень, вивчення технологій IoT та методів вимірювання вологості ґрунту, викладених у попередніх підрозділах, сформульовано мету, задачі, об'єкт і предмет дослідження даної бакалаврської дипломної роботи.

Метою роботи є розроблення програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі мікроконтролера ESP32 та хмарної платформи Arduino IoT Cloud, що забезпечує трьохканальний незалежний моніторинг вологості ґрунту, автоматичний полив за розкладом при досягненні порогових значень вологості та дистанційне керування через веб-дашборд.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1) сформулювати функціональні та нефункціональні вимоги до програмно-технічного засобу, що охоплюють вимоги до моніторингу трьох незалежних каналів вологості, режимів автоматичного та ручного керування, розкладного поливу, хмарної взаємодії та живлення від акумулятора;

2) визначити архітектуру системи та описати структуру апаратних блоків.

3) розробити схему електричну принципову системи у середовищі EasyEDA;

4) розробити алгоритм функціонування мікроконтролера;

5) розробити програмне забезпечення прошивки ESP32 на основі бібліотеки ArduinoIoTCloud;

6) розгорнути хмарне середовище Arduino IoT Cloud;

7) побудувати віртуальний прототип системи у симуляторі Wokwi на базі ESP32;

8) провести тестування функціональності, перевірити коректність зчитування датчиків, двосторонню синхронізацію змінних між Wokwi та Arduino Cloud, логіку автоматичного режиму та ручного режиму, проаналізувати результати та визначити практичні обмеження прототипу.

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ КЕРУВАННЯ АВТОМАТИЧНИМ ПОЛИВОМ КІМНАТНИХ РОСЛИН НА ОСНОВІ ARDUINO CLOUD

2.1 Аналіз вимог до програмно-технічного засобу

Розроблення програмно-технічного засобу керування автоматичним поливом кімнатних рослин потребує формування вимог до його функціональних можливостей, апаратного забезпечення, програмного забезпечення та характеристик експлуатації. Визначення таких вимог є необхідним етапом проєктування, оскільки дозволяє сформувавши структуру системи, обрати відповідні технічні засоби та забезпечити відповідність розробленого рішення поставленим завданням.

Основним призначенням програмно технічного засобу є автоматизоване підтримання необхідного рівня вологості ґрунту для кімнатних рослин без постійної участі користувача. Для досягнення цієї мети система повинна здійснювати безперервний моніторинг вологості ґрунту, аналізувати отримані дані та автоматично активувати полив у разі зниження вологості нижче встановленого значення. Крім автоматичного режиму роботи повинна бути передбачена можливість дистанційного контролю та керування через хмарну платформу Arduino Cloud.

Важливою функціональною вимогою є підтримка декількох незалежних зон поливу. У межах даної роботи передбачено обслуговування трьох ділянок із кімнатними рослинами. Для кожної ділянки необхідно забезпечити окреме вимірювання вологості ґрунту та незалежне керування насосом. Такий підхід дозволяє враховувати різні потреби рослин у волозі та забезпечує гнучкість використання системи.

До вимог щодо вимірювання параметрів належить забезпечення отримання актуальних значень вологості ґрунту в режимі реального часу. Система повинна здійснювати періодичне опитування датчиків та виконувати обробку отриманих

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 23
Зм.	Арк.	№ докum.	Підпис	Дата		

сигналів. Результати вимірювання мають бути доступними як для локального алгоритму прийняття рішень, так і для передачі до хмарного середовища з метою подальшої візуалізації.

Алгоритм керування повинен забезпечувати автоматичне ввімкнення насоса при досягненні критичного рівня висихання ґрунту та його вимкнення після завершення циклу поливу. При цьому необхідно виключити часті перемикання виконавчих механізмів, які можуть негативно впливати на ресурс обладнання. З цією метою програмне забезпечення повинно використовувати порогові значення вологості та часові інтервали між послідовними циклами поливу.

Однією з ключових вимог є забезпечення дистанційного моніторингу. Користувач повинен мати можливість отримувати інформацію про поточний рівень вологості кожної ділянки, стан насосів та режим роботи системи за допомогою веб інтерфейсу або мобільного застосунку Arduino Cloud. Наявність такого функціоналу дозволяє здійснювати контроль за станом рослин незалежно від місця перебування користувача.

Крім автоматичного режиму, система повинна підтримувати режим ручного керування. У цьому випадку користувач через інформаційну панель може самостійно активувати або деактивувати насос певної ділянки. Така можливість є корисною під час тестування системи, проведення технічного обслуговування або в ситуаціях, коли необхідно виконати додатковий полив незалежно від показників датчиків.

До апаратних вимог належить використання мікроконтролера, який забезпечує достатню кількість входів і виходів для підключення всіх компонентів системи, підтримує бездротовий зв'язок та має достатні обчислювальні ресурси для виконання алгоритмів керування. Аналіз доступних платформ показав, що мікроконтролер ESP32 повністю задовольняє зазначені вимоги завдяки наявності інтегрованого модуля Wi Fi, великій кількості портів введення виведення та підтримці середовища Arduino.

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

Важливими є також вимоги до системи живлення. Пристрій повинен працювати автономно та забезпечувати стабільне функціонування в умовах відсутності постійного підключення до електричної мережі. Для цього доцільним є використання акумуляторної батареї з можливістю підзаряджання. Система живлення повинна містити вузол заряджання акумулятора, засоби стабілізації напруги та захисту від небажаних режимів роботи.

З точки зору програмного забезпечення необхідно забезпечити надійний обмін даними між мікроконтролером та хмарною платформою Arduino Cloud. Передача інформації повинна здійснюватися автоматично через мережу Wi Fi із мінімальними затримками.

До експлуатаційних вимог належить простота встановлення та налаштування системи. Користувач повинен мати можливість швидко підключити датчик, насоси та джерело живлення без використання складного спеціалізованого обладнання. Також важливо забезпечити можливість подальшого розширення функціональності системи шляхом підключення додаткових датчиків або виконавчих пристроїв. З огляду на особливості використання в житлових приміщеннях, програмно технічний засіб повинен характеризуватися низьким енергоспоживанням, компактними розмірами та високою надійністю роботи. Відмова окремого елемента не повинна призводити до втрати працездатності всієї системи, а програмне забезпечення має забезпечувати коректне відновлення роботи після перезавантаження або тимчасової втрати зв'язку з хмарним сервісом.

2.2 Архітектура програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

Архітектура програмно технічного засобу керування автоматичним поливом кімнатних рослин побудована за принципом взаємодії фізичного рівня вимірювання та виконавчих механізмів із хмарною платформою Arduino Cloud,

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

яка забезпечує централізований моніторинг та дистанційне керування системою. Запропонована структура орієнтована на автоматичне підтримання необхідного рівня вологості ґрунту для декількох незалежних ділянок вирощування рослин та дозволяє здійснювати віддалений контроль параметрів системи через веб інтерфейс або мобільний застосунок.

Архітектуру програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud подано схемою на рис. 2.1. Основу архітектури становить мікроконтролер ESP32, який виконує функції центрального вузла збору даних, обробки інформації та формування керуючих впливів. Використання ESP32 обумовлене наявністю вбудованого бездротового інтерфейсу Wi Fi, достатнім обсягом оперативної та програмної пам'яті, а також підтримкою інтеграції з платформою Arduino Cloud. Мікроконтролер забезпечує постійний обмін інформацією між фізичними компонентами системи та хмарною інфраструктурою, виконуючи роль шлюзу між реальними пристроями та їх цифровими представленнями.

На фізичному рівні система складається з трьох незалежних ділянок поливу. Кожна ділянка містить датчик вимірювання вологості ґрунту та насос для подачі води (для комутації насоса із мікроконтролером задіяно реле, на схемі не зображено). Датчики здійснюють безперервний контроль стану ґрунту та формують електричні сигнали, величина яких залежить від поточного рівня вологості. Отримані значення надходять до мікроконтролера ESP32 через аналогові або цифрові входи, де виконуються їх обробка, масштабування та перетворення у відсотковий показник вологості.

Інформація про рівень вологості кожної ділянки аналізується програмним забезпеченням мікроконтролера відповідно до заданих порогових значень. У випадку зниження вологості нижче встановленого рівня формується команда на активацію відповідного насоса. Таким чином забезпечується локальне автоматичне регулювання процесу поливу без необхідності постійного втручання користувача. Після подачі води насос працює протягом визначеного

проміжку часу або до досягнення необхідного рівня вологості, після чого вимикається. Такий підхід дозволяє уникнути надмірного зволоження ґрунту та сприяє раціональному використанню водних ресурсів.

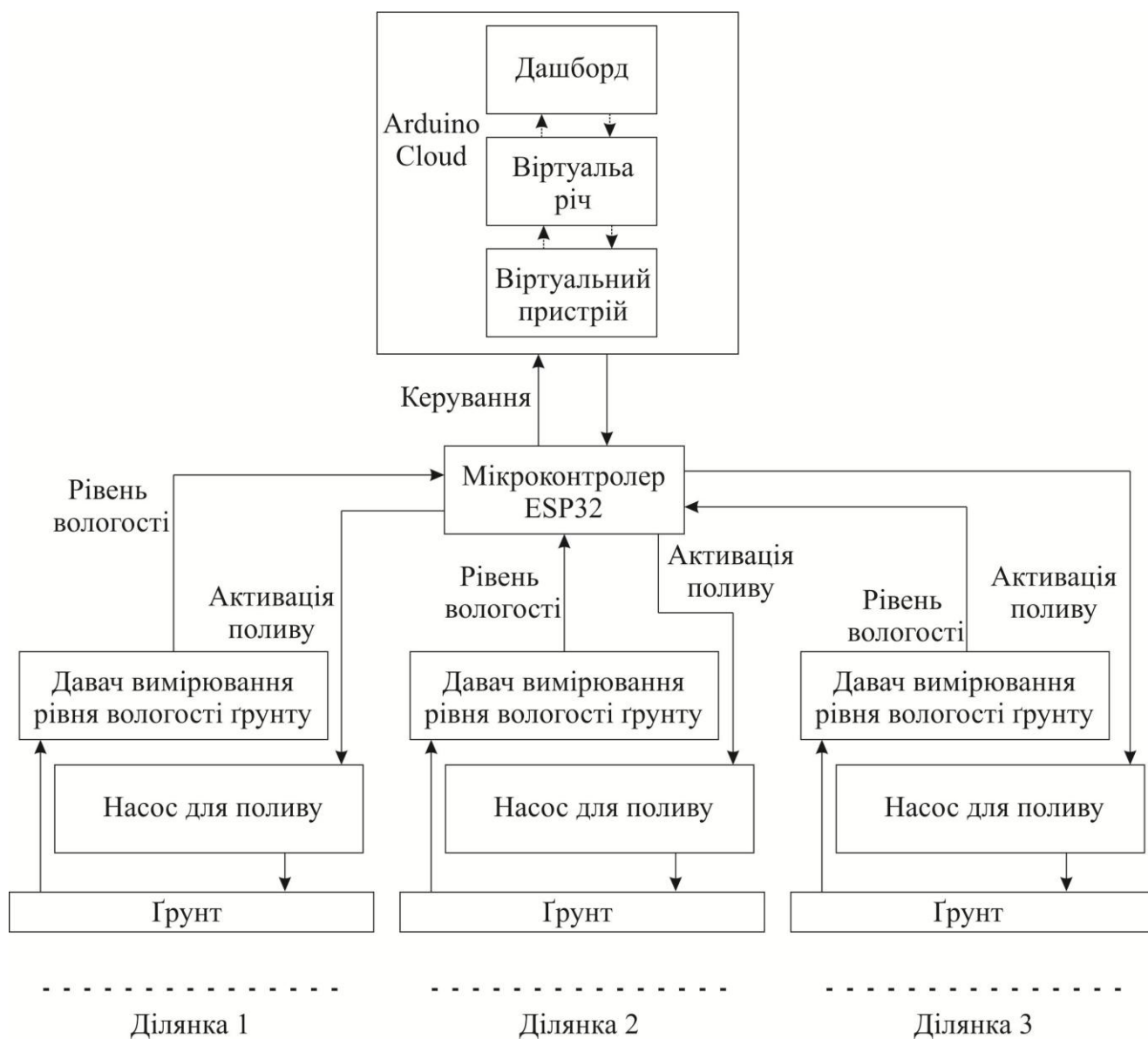


Рисунок 2.1 – Архітектура програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

Особливістю запропонованої архітектури є можливість незалежного обслуговування декількох зон поливу одним керуючим пристроєм. Кожна ділянка функціонує автономно з точки зору контролю вологості та керування насосом, проте всі дані централізовано обробляються мікроконтролером. Це

дозволяє масштабувати систему шляхом додавання нових зон без суттєвих змін логіки функціонування програмного забезпечення.

Для організації віддаленого моніторингу та керування використовується платформа Arduino Cloud. У межах даної платформи створюється віртуальний пристрій, який є цифровим відображенням фізичного мікроконтролера ESP32. Віртуальний пристрій забезпечує взаємозв'язок між апаратною частиною системи та хмарними сервісами, виконуючи функцію обміну телеметричними даними та керуючими командами.

На наступному рівні розташована віртуальна річ. Даний елемент представляє собою логічний контейнер, у якому визначаються змінні, що використовуються для обміну інформацією між пристроєм та хмарою. До таких змінних належать значення вологості для кожної ділянки, стан насосів, параметри автоматичного режиму роботи та команди ручного керування. Віртуальна річ забезпечує структуроване представлення параметрів системи та їх синхронізацію між фізичним обладнанням і користувацьким інтерфейсом.

Верхній рівень архітектури представлений інформаційною панеллю, або дашбордом. Дашборд є засобом візуалізації та керування, що дозволяє користувачу отримувати актуальні дані про стан системи в режимі реального часу. Через відповідні віджети відображаються показники вологості кожної ділянки, стан роботи насосів та інші службові параметри. Крім моніторингу, дашборд забезпечує можливість дистанційного впливу на роботу системи шляхом зміни режимів функціонування або ручного запуску поливу.

Процес обміну інформацією в системі відбувається у двох напрямках. У висхідному напрямку дані від датчиків вологості надходять до мікроконтролера, після чого передаються до Arduino Cloud для збереження та відображення. У низхідному напрямку команди користувача, сформовані через дашборд, передаються до віртуальної речі, далі до віртуального пристрою та надходять на ESP32, який виконує відповідні дії над виконавчими механізмами. Така організація забезпечує оперативне керування системою незалежно від місця

перебування користувача. Застосування хмарної архітектури дозволяє реалізувати додаткові функції, пов'язані зі зберіганням історичних даних, аналізом змін вологості ґрунту та контролем працездатності обладнання. Дані, що накопичуються в Arduino Cloud, можуть використовуватися для оцінювання ефективності поливу, визначення оптимальних параметрів зволоження та прогнозування потреб рослин у воді. Крім того, хмарна інфраструктура забезпечує доступ до інформації з різних пристроїв без необхідності створення власного серверного програмного забезпечення.

2.3 Схема електрична

Електрична схема програмно технічного засобу керування автоматичним поливом кімнатних рослин розроблена на основі мікроконтролера ESP32 та забезпечує виконання функцій моніторингу вологості ґрунту, керування виконавчими механізмами поливу, а також автономного живлення пристрою від акумуляторної батареї. Структура схеми передбачає підключення датчиків вологості, насосів для подачі води та вузлів живлення, що забезпечують стабільну роботу системи в автоматичному режимі. Схему електричну подано на рис. 2.2.

Центральним елементом схеми є модуль ESP32 (U1), який виконує функції збору даних від датчиків, обробки інформації та формування керуючих сигналів. Мікроконтролер містить достатню кількість цифрових та аналогових входів і виходів для підключення периферійних пристроїв, а також вбудований модуль бездротового зв'язку Wi Fi, необхідний для взаємодії з платформою Arduino Cloud.

Для підключення датчиків вологості ґрунту передбачено три клемні роз'єми X1, X2 та X3. Через ці роз'єми до мікроконтролера надходять сигнали, що характеризують поточний стан вологості ґрунту в кожній із контрольованих зон.

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

Кожний роз'єм містить лінії живлення та сигнальний провід, що забезпечує зручне підключення типових модулів вимірювання вологості ґрунту.

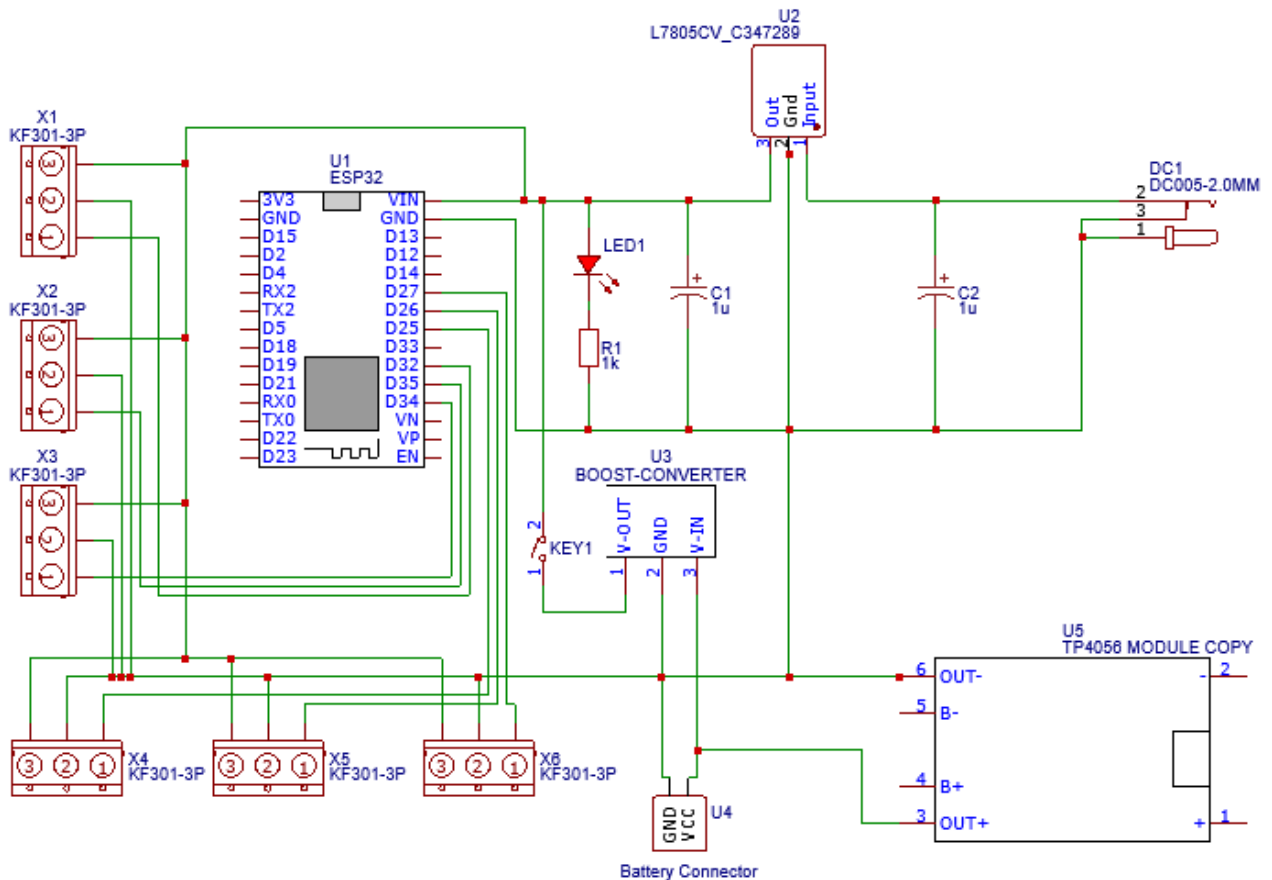


Рисунок 2.2 – Схема електрична

Керування насосами здійснюється через окремі вихідні лінії мікроконтролера. Для підключення виконавчих пристроїв використано клемні роз'єми X4, X5 та X6. Через ці роз'єми подаються сигнали керування насосами відповідних зон поливу. Після аналізу показників вологості ESP32 формує логічний сигнал на відповідному виході, який активує насос і забезпечує подачу води до рослини. Така організація дозволяє реалізувати незалежне керування кожною зоною поливу та підтримувати індивідуальні умови зволоження для різних рослин.

Система живлення побудована на використанні літій іонного акумулятора, що підключається через роз'єм U4. Для заряджання акумуляторної батареї

застосовується модуль TP4056 (U5), який забезпечує контроль процесу заряджання та захист акумулятора від перевищення допустимих режимів роботи. Використання даного модуля дозволяє здійснювати безпечно заряджання батареї від зовнішнього джерела живлення без необхідності використання додаткових схем керування.

Живлення всієї системи здійснюється через підвищувальний перетворювач напруги U3. Оскільки номінальна напруга літій іонного акумулятора є недостатньою для стабільного живлення ESP32 та периферійних компонентів, використовується модуль підвищення напруги, який формує необхідний рівень вихідної напруги. Вхід перетворювача підключений до акумуляторної батареї, а вихід використовується для живлення мікроконтролера та інших елементів схеми.

Для можливості ручного керування живленням системи передбачено вимикач S1. Вимикач встановлений у колі живлення підвищувального перетворювача та дозволяє повністю відключати навантаження від акумулятора під час транспортування, технічного обслуговування або тривалого зберігання пристрою. Це сприяє зменшенню енергоспоживання та запобігає небажаному розрядженню акумуляторної батареї.

Стабілізація напруги живлення забезпечується конденсаторами C1 (470 мкФ, 25В) та C2 (100 мкФ, 16В), які встановлені на вході та виході стабілізатора U2 типу L7805CV. Конденсатори виконують функцію згладжування пульсацій напруги та підвищують стійкість роботи електронних компонентів при зміні навантаження. Стабілізатор формує стабільну напругу живлення для вузлів системи та забезпечує захист від коливань вхідної напруги.

Для індикації наявності живлення використовується світлодіод LED1 з обмежувальним резистором R1. Після подачі живлення на схему світлодіод переходить у активний стан, що дозволяє візуально контролювати працездатність пристрою та наявність напруги на основних вузлах системи.

Підключення зовнішнього джерела живлення або зарядного пристрою здійснюється через роз'єм DC1. Даний роз'єм забезпечує можливість заряджання акумулятора та живлення системи від зовнішнього адаптера. Поєднання роз'єму DC1 та модуля TP4056 дозволяє реалізувати автономну систему живлення з можливістю багаторазового заряджання акумулятора.

На основі розробленої електричної схеми було виконано прототипування пристрою та розроблено друковану плату. Проектування здійснювалося у середовищі EasyEDA із формуванням двовимірної моделі плати для трасування провідників і розміщення компонентів. Після завершення етапу компоновки було створено тривимірну модель друкованої плати, яка дозволила оцінити взаємне розташування елементів, габаритні розміри пристрою та зручність подальшого монтажу. Розроблені 3D (рис. 2.3) та 2D (рис. 2.4) моделі підтвердили коректність обраної структури та забезпечили можливість підготовки проєкту до виготовлення друкованої плати.

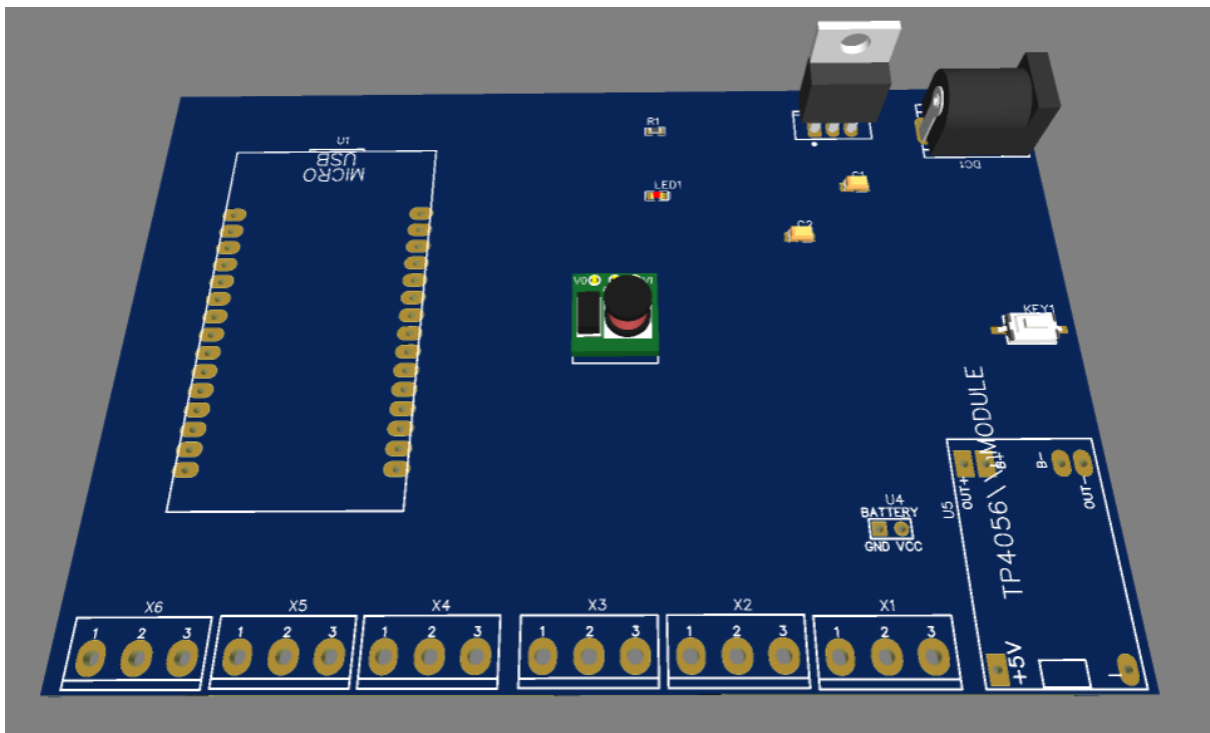


Рисунок 2.3 – 3 D вигляд макетної плати апаратної частини проєктованого засобу

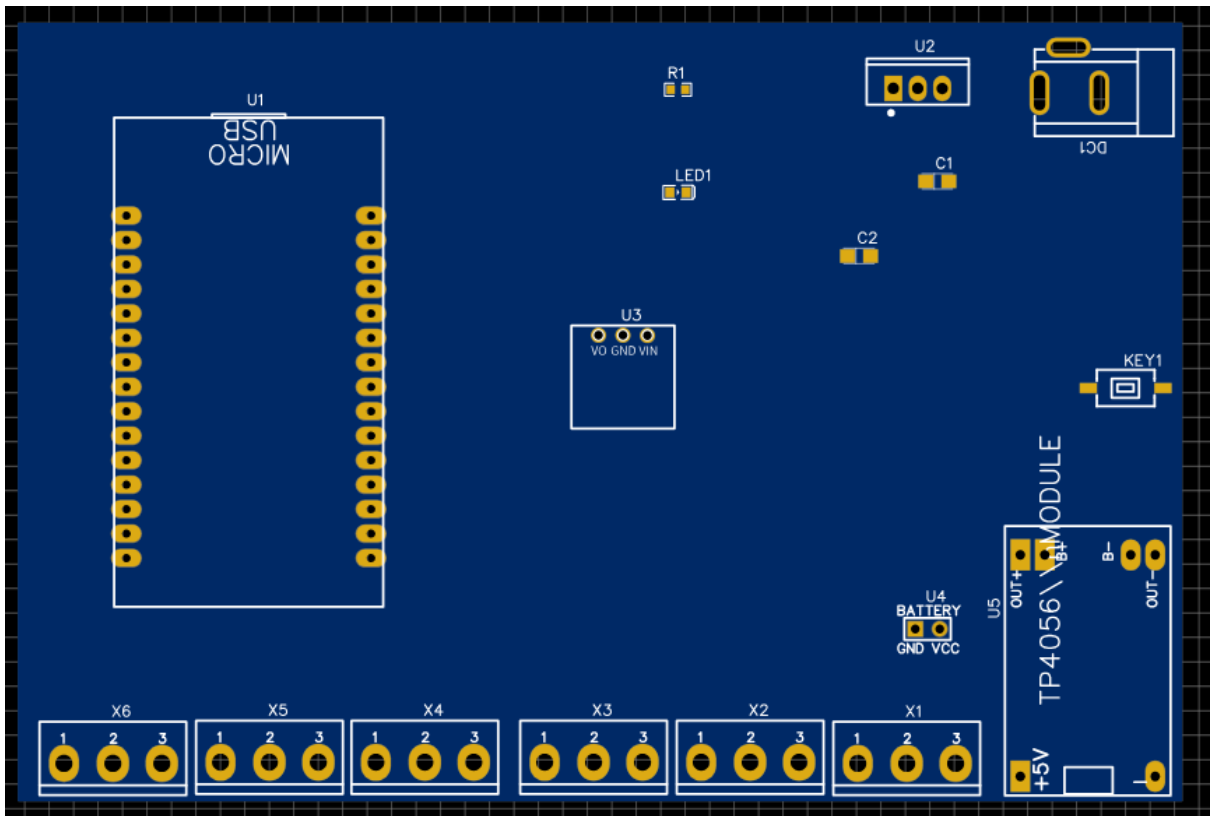


Рисунок 2.4 – 2 D вигляд макетної плати апаратної частини проєктованого засобу

2.4 Алгоритм функціонування мікроконтролера

Алгоритм функціонування мікроконтролера ESP32 визначає послідовність дій, що виконуються від моменту подачі живлення до завершення кожного ітераційного циклу головного виконавчого контуру. Алгоритм розроблений за принципом реактивного керування, де всі рішення щодо активації насосів приймаються безпосередньо на пристрої на основі актуальних показань датчиків та параметрів, завантажених з хмари, без необхідності звернення до зовнішнього сервера в момент прийняття рішення. Хмарне з'єднання використовується виключно для публікації вимірювань та синхронізації налаштувань. Блок-схему алгоритму наведено на рисунку 2.5.

Алгоритм починається з блоку ініціалізації, що виконується один раз після подачі живлення або апаратного скидання.

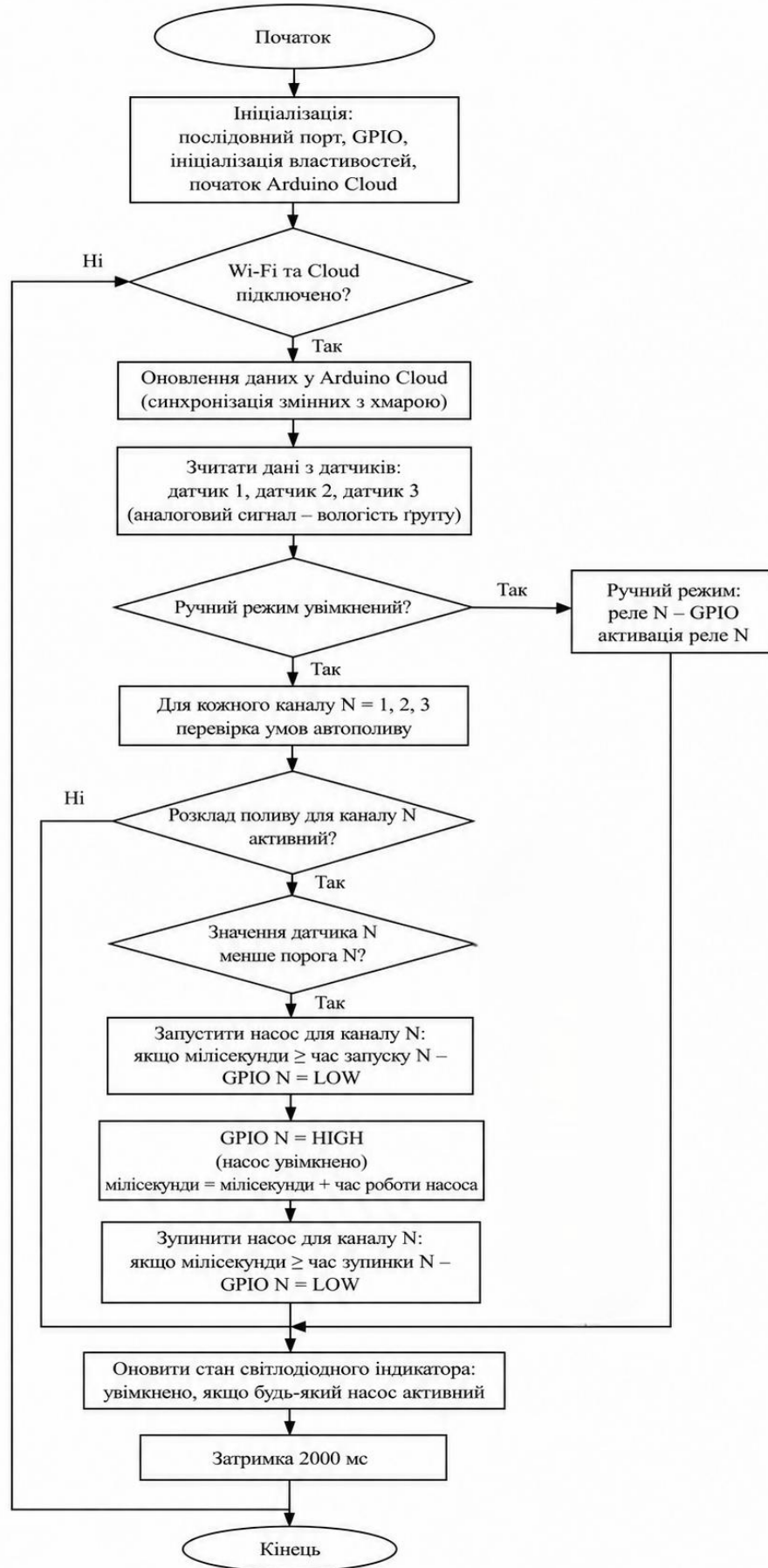


Рисунок 2.5 – Блок-схема алгоритму функціонування мікроконтролера ESP32

Зм.	Арк.	№ докум.	Підпис	Дата

На початку роботи алгоритму виконуються такі дії: відкривається Serial-з'єднання з швидкістю 115 200 бод для діагностичного виведення, усі три пін-виводи насосів (GPIO26, GPIO27, GPIO14) конфігуруються як цифрові виходи з початковим рівнем LOW, пін статусного LED (GPIO2) конфігурується як вихід, викликається функція `initProperties()`, що реєструє всі cloud-змінні у системі `ArduinoIoTCloud`, і нарешті `ArduinoCloud.begin()` ініціює підключення до Wi-Fi. Після ініціалізації алгоритм переходить до блоку очікування підключення.

Якщо Wi-Fi та Cloud з'єднання ще не встановлене, алгоритм залишається у циклі очікування без виконання будь-якої логіки керування. Це запобігає некоректній роботі системи до завершення синхронізації cloud-змінних. Після успішного підключення всі раніше збережені на сервері значення (`threshold_N`, `automatic_mode`, `plant_schedule_N`, `relay_N`) автоматично завантажуються до локальних змінних ESP32, що забезпечує відновлення поточного стану налаштувань після будь-якого перезапуску пристрою.

Основний виконавчий контур починається з виклику `ArduinoCloud.update()`. Ця функція є ядром хмарної взаємодії і такі операції як обробка вхідно черги повідомлень від Arduino Cloud (оновлення змінних з дашборду, зміни розкладу, команди ручного керування), надсилає до хмари накопичені зміни локальних змінних (оновлені показання датчиків). Якщо з'єднання було втрачено, бібліотека автоматично ініціює процедуру повторного підключення.

Наступним кроком є зчитування показань трьох датчиків вологості через функцію `readSoilMoisture()`. Для кожного каналу виконується аналогово-цифрове перетворення відповідного GPIO-виводу (34, 35, 32) з отриманням 12-бітного значення в діапазоні 0–4095. Це значення масштабується за допомогою функції `map()` до відсоткового діапазону 0–100% з використанням калібрувальних констант `ADC_DRY` та `ADC_WET`, що відповідають показанням АЦП при повністю сухому та насиченому водою ґрунті (слід відзначити, що при іншому датчику визначення рівня вологості ґрунту дані значення слід також емпірично

визначити). Отримане значення обмежується функцією `constrain()` для виключення виходу за межі діапазону при нестабільних показаннях. Оновлені значення зберігаються у cloud-змінних `sensor_1`, `sensor_2`, `sensor_3` і при наступному виклику `ArduinoCloud.update()` передаються до хмари.

Після зчитування давачів алгоритм перевіряє стан прапорця `automatic_mode`. Якщо автоматичний режим вимкнений, алгоритм переходить до гілки ручного керування: стан кожного насоса визначається виключно значенням відповідної булевої cloud-змінної `relay_N`. Зміна цієї змінної через дашборд або мобільний застосунок `Arduino Cloud` викликає callback-функцію `onRelayNChange()`, яка безпосередньо встановлює відповідний рівень HIGH або LOW на GPIO насоса. При ручному режимі автоматичні таймери поливу не активуються, що виключає конфлікт між ручним та автоматичним управлінням.

Якщо автоматичний режим увімкнений, алгоритм виконує перевірку умов автополиву для кожного з трьох каналів послідовно. Для каналу N перевіряється перша умова, що визначає активність часового вікна розкладу через метод `plant_schedule_N.isActive()`. Метод порівнює поточний системний час мікроконтролера ESP32 синхронізований через протокол NTP під час підключення до Wi-Fi з часовими мітками початку та завершення активного вікна, збереженими у об'єкті `CloudSchedule`. Якщо поточний момент не потрапляє у жодне активне вікно, подальша перевірка для цього каналу пропускається і алгоритм переходить до наступного каналу.

Якщо розклад активний, перевіряється друга умова, яка визначає чи є показання вологості ґрунту нижчими за встановлений поріг (`sensor_N < threshold_N`). При виконанні цієї умови перевіряється третя умова — чи не перебуває насос N вже у активному циклі поливу (`stopAt_N != 0`). Це запобігає повторному запуску насоса до завершення поточного циклу, що виключає надмірне зволоження. Якщо всі три умови виконані, насос N вмикається шляхом подачі HIGH на відповідний GPIO, а в змінну `stopAt_N` записується значення

millis() + PUMP_ON_MS, де PUMP_ON_MS константа тривалості одного циклу поливу у мілісекундах.

На кожній ітерації циклу після перевірки всіх каналів виконується функція autoStopPump() для кожного насоса. Вона порівнює поточне значення millis() зі збереженою міткою зупинки: якщо час минув, GPIO насоса переводиться у LOW, мітка скидається до нуля. Такий неблокуючий підхід до управління часом дозволяє всім трьом насосам працювати з незалежними таймерами одночасно, не блокуючи основний цикл і не перешкоджаючи виконанню ArduinoCloud.update(). Після оновлення стану насосів оновлюється статусний LED: він горить, якщо хоча б один насос є активним. Цикл завершується затримкою 2000 мс, після чого починається наступна ітерація.

2.5 Програмна реалізація взаємодії мікроконтролера з хмарною платформою

Програмна реалізація взаємодії ESP32 з платформою Arduino IoT Cloud є одним з головних аспектів проєктованого засобу, оскільки саме вона забезпечує дистанційний моніторинг показань датчиків, передачу команд керування та синхронізацію параметрів між пристроєм і веб-дашбордом. Взаємодія реалізована на основі бібліотеки ArduinoIoTCloud версії 2.9.0 у зв'язці з Arduino_ConnectionHandler.

З архітектурної точки зору бібліотека ArduinoIoTCloud реалізує паттерн спостерігач. Cloud-змінні є спостережуваними об'єктами, зміни яких автоматично публікуються до хмари або ініціюють виклик зареєстрованих callback-функцій при отриманні оновлень зі сторони сервера. Розробник прикладного рівня не взаємодіє безпосередньо з MQTT-пакетами чи TLS-сокетами, а оперує виключно змінними стандартних C++-типів та callback-функціями, що суттєво спрощує розробку і знижує ймовірність помилок на мережевому рівні.

Центральним файлом хмарної інтеграції є `thingProperties.h`. У ньому оголошуються глобальні змінні відповідних типів: `int` для `sensor_N` та `threshold_N`, `bool` для `relay_N` та `automatic_mode`, `CloudSchedule` для `plant_schedule_N`. Функція `initProperties()` реєструє кожну змінну у системі `ArduinoCloud` через виклики `ArduinoCloud.addProperty()` з трьома параметрами: посиланням на змінну, напрямком синхронізації (`READ` або `READWRITE`) та вказівником на `callback`-функцію (`NULL` якщо `callback` не потрібен). Для змінних типу `READ` (наприклад для змінних, які відповідають за отримання даних із датчиків вологості ґрунту) `callback` не реєструється, оскільки ці змінні лише публікуються до хмари, а не отримуються з неї. Для `READWRITE`-змінних реєструються `callback`-функції, що викликаються автоматично при отриманні оновлення з сервера.

Механізм передачі даних між пристроєм і хмарою функціонує таким чином. При зміні локальної `cloud`-змінної (наприклад, `sensor_1` оновлюється після зчитування АЦП) бібліотека позначає цю змінну як «брудну» (`dirty`). При наступному виклику `ArduinoCloud.update()` всі брудні змінні серіалізуються у бінарний формат `CBOR` (`Concise Binary Object Representation`) та публікуються до відповідного `MQTT`-топіка пристрою на брокері `Arduino Cloud`. Брокер зберігає останнє значення кожної змінної та розсилає його всім підписаним клієнтам, зокрема, веб-дашборду користувача.

У зворотному напрямку, коли оператор змінює значення змінної на дашборді (наприклад, переміщує слайдер `threshold_1`), веб-клієнт публікує нове значення у відповідний `MQTT`-топік. Брокер пересилає це повідомлення на `ESP32`. Бібліотека `ArduinoIoTCloud` при обробці повідомлення в `ArduinoCloud.update()` десеріалізує `CBOR`-значення, оновлює відповідну локальну змінну та викликає зареєстрований `callback` у даному випадку `onThreshold1Change()`, яка логує нове значення та оновлює поріг для першого каналу.

Особливої уваги заслуговує обробка типу CloudSchedule. Цей спеціалізований тип даних бібліотеки ArduinoIoTCloud зберігає параметри розкладу у форматі структури з полями: початковий UNIX-timestamp, тривалість активного вікна у секундах, тип повторення (одноразово, щодня, щотижня тощо) та кінцева дата повторення. Коли оператор налаштовує розклад через Scheduler-віджет на дашборді, ці параметри серіалізуються та передаються на пристрій. Метод isActive() об'єкта CloudSchedule використовує поточний системний час ESP32, синхронізований через NTP-сервер під час підключення до Wi-Fi, для визначення того, чи потрапляє поточний момент у будь-яке активне вікно з урахуванням заданого типу повторення. Це дозволяє мікроконтролеру самостійно вирішувати, коли виконувати полив, без постійного звернення до сервера.

Надійність хмарного з'єднання забезпечується вбудованим механізмом автовідновлення бібліотеки Arduino_ConnectionHandler. При втраті Wi-Fi або MQTT-з'єднання обробник автоматично починає спроби перепідключення з експоненційним відкладенням між спробами. Під час відсутності з'єднання пристрій продовжує виконувати основний цикл поливу, використовуючи останні отримані значення cloud-змінних, що забезпечує стійку автономну роботу навіть за умов нестабільного Wi-Fi. Після відновлення з'єднання відбувається повна синхронізація стану. Пристрій публікує накопичені зміни, а сервер надсилає актуальні значення всіх READWRITE-змінних.

Повний перелік хмарних змінних для проєктованого засобу наведено у таблиці 2.1.

Таблиця 2.1 – Cloud-змінні системи та їх характеристики

Змінна	Тип	Доступ	Callback	Призначення
sensor_1	int	READ	NULL	Вологість ґрунту каналу 1, %

Продовження таблиці 2.1

sensor_2	int	READ WRITE	NULL	Вологість грунту каналу 2, %
sensor_3	int	READ WRITE	NULL	Вологість грунту каналу 3, %
relay_1	bool	READ WRITE	onRelay1Change	Ручне керування насосом 1
relay_2	bool	READ WRITE	onRelay2Change	Ручне керування насосом 2
relay_3	bool	READ WRITE	onRelay3Change	Ручне керування насосом 3
automatic_mode	bool	READ WRITE	onAutomaticModeChange	Режим роботи: авто / ручний
threshold_1	int	READ WRITE	onThreshold1Change	Поріг вологості каналу 1, %
threshold_2	int	READ WRITE	onThreshold2Change	Поріг вологості каналу 2, %
threshold_3	int	READ WRITE	onThreshold3Change	Поріг вологості каналу 3, %

Зм.	Арк.	№ докум.	Підпис	Дата

КВРКІ. 240494.22.04.10 ПЗ

Арк.
40

Кінець таблиці 2.1

plant_schedule_1	CloudSchedule	READ WRITE	onPlantSchedule1Change	Розклад поливу каналу 1
plant_schedule_2	CloudSchedule	READ WRITE	onPlantSchedule2Change	Розклад поливу каналу 2
plant_schedule_3	CloudSchedule	READ WRITE	onPlantSchedule3Change	Розклад поливу каналу 3

2.6 Висновки до другого розділу

У результаті виконання даного розділу було проведено аналіз предметної області та сформовано вимоги до програмно технічного засобу керування автоматичним поливом кімнатних рослин. Визначено основні функціональні можливості системи, які включають автоматичний контроль вологості ґрунту, керування процесом поливу, дистанційний моніторинг параметрів та взаємодію з користувачем через хмарну платформу Arduino Cloud. На основі сформованих вимог розроблено архітектуру програмно технічного засобу, що забезпечує взаємодію між давачами вологості ґрунту, виконавчими механізмами поливу, мікроконтролером ESP32 та хмарним середовищем Arduino Cloud. Запропонована архітектура дозволяє реалізувати централізоване керування декількома незалежними зонами поливу та забезпечує можливість віддаленого доступу до даних системи. Також було розроблено електричну схему пристрою, яка включає вузли вимірювання вологості, керування насосами, автономного живлення та бездротового обміну даними. На основі електричної схеми виконано проектування друкованої плати та створено її двовимірну і тривимірну моделі, що підтвердило можливість практичної реалізації розробленого пристрою.

3 ТЕСТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ КЕРУВАННЯ АВТОМАТИЧНИМ ПОЛИВОМ КІМНАТНИХ РОСЛИН НА ОСНОВІ ARDUINO CLOUD

3.1 Розгортання середовища в Arduino Cloud

Arduino IoT Cloud є хмарною платформою компанії Arduino, яка забезпечує зберігання даних, двосторонній обмін інформацією між мікроконтролером і веб-інтерфейсом, а також надає інструменти для побудови інтерактивних дашбордів. Для реалізації програмно-технічного засобу керування автоматичним поливом кімнатних рослин саме ця платформа була обрана як основна хмарна інфраструктура, оскільки вона безпосередньо підтримує модуль ESP32, має нативну бібліотеку ArduinoIoTCloud та надає безкоштовний рівень доступу з достатніми для прототипу можливостями.

Архітектура Arduino IoT Cloud побудована на взаємодії декількох логічних компонентів, кожен з яких виконує окрему функцію у процесі обміну даними між користувачем та фізичним пристроєм. Основними елементами платформи є Пристрої, речі, змінні та інформаційні панелі (Devices, Things, Variables та Dashboards), які утворюють єдину систему керування IoT-проектом.

Найнижчим рівнем архітектури є пристрій. Під пристроєм розуміється фізичний мікроконтролер або плата розробки, зареєстрована в Arduino Cloud. У даному проєкті таким пристроєм виступає модуль ESP32. Під час реєстрації пристрою платформа генерує унікальні параметри автентифікації, за допомогою яких забезпечується захищене підключення до хмарного сервісу. Один фізичний пристрій може бути пов'язаний із певним програмним проєктом та використовуватися для обміну даними з Arduino Cloud.

Наступним логічним рівнем є річ. Об'єк річ являє собою цифрове представлення IoT-пристрою у хмарному середовищі. Якщо пристрій описує фізичне обладнання, то річ описує його функціональність та дані, з якими працює система. Саме всередині речі створюються хмарні змінні,

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

налаштовуються права доступу до них та визначаються механізми взаємодії між обладнанням і користувачем. Кожна річ пов'язується із конкретним пристроєм, що дозволяє платформі знати, які саме дані необхідно передавати між обладнанням та хмарою.

Центральним елементом речі є змінні. Змінна являє собою канал обміну даними між мікроконтролером та сервером Arduino Cloud. Кожна змінна має визначений тип даних, наприклад логічний, цілочисельний, дійсний або текстовий. Крім того, для змінної задається напрямок обміну інформацією. Дані можуть передаватися лише від пристрою до хмари, лише від хмари до пристрою або в обох напрямках. Усі значення змінних автоматично синхронізуються між фізичним пристроєм, сервером та користувацьким інтерфейсом.

Для взаємодії користувача з системою використовуються дашборди або інформаційні панелі. Дашборд є графічною панеллю керування, що містить набір віджетів для відображення та зміни значень хмарних змінних. Кожен віджет прив'язується до конкретної змінної речі. Наприклад, індикатор може відображати значення вологості ґрунту, кнопка може керувати насосом, а графік може показувати історію зміни параметрів. Завдяки такій прив'язці будь-яка зміна значення змінної автоматично відображається на дашборді, а дії користувача на панелі керування миттєво передаються до пристрою.

Взаємодія між компонентами Arduino IoT Cloud відбувається за наступною схемою. Фізичний пристрій підключається до мережі Інтернет та обмінюється даними із сервером Arduino Cloud. Сервер зберігає та синхронізує значення змінних, які належать відповідній речі. Дашборд отримує доступ до цих змінних через хмарну платформу та відображає їх користувачеві у зручному графічному вигляді. Таким чином формується ланцюг: Device – Thing – Variables – Dashboard. Таким чином усі компоненти взаємодіють через хмарну інфраструктуру, що забезпечує віддалений доступ до системи незалежно від місця розташування користувача та пристрою. Архітектуру взаємодії в Arduino IoT Cloud подано на рис. 3.1.

Також варто додати, що окрім основних компонентів, Arduino IoT Cloud містить додаткові засоби автоматизації, зокрема тригери та шаблони. Тригери дозволяють виконувати визначені дії при досягненні заданих умов, наприклад надсилати повідомлення при зниженні вологості ґрунту нижче встановленого порогу. У свою чергу шаблони використовуються для збереження типових конфігурацій проєктів та їх подальшого повторного використання. Завдяки цьому платформа забезпечує не лише передачу даних, а й реалізацію елементів автоматизованого керування IoT-системами.

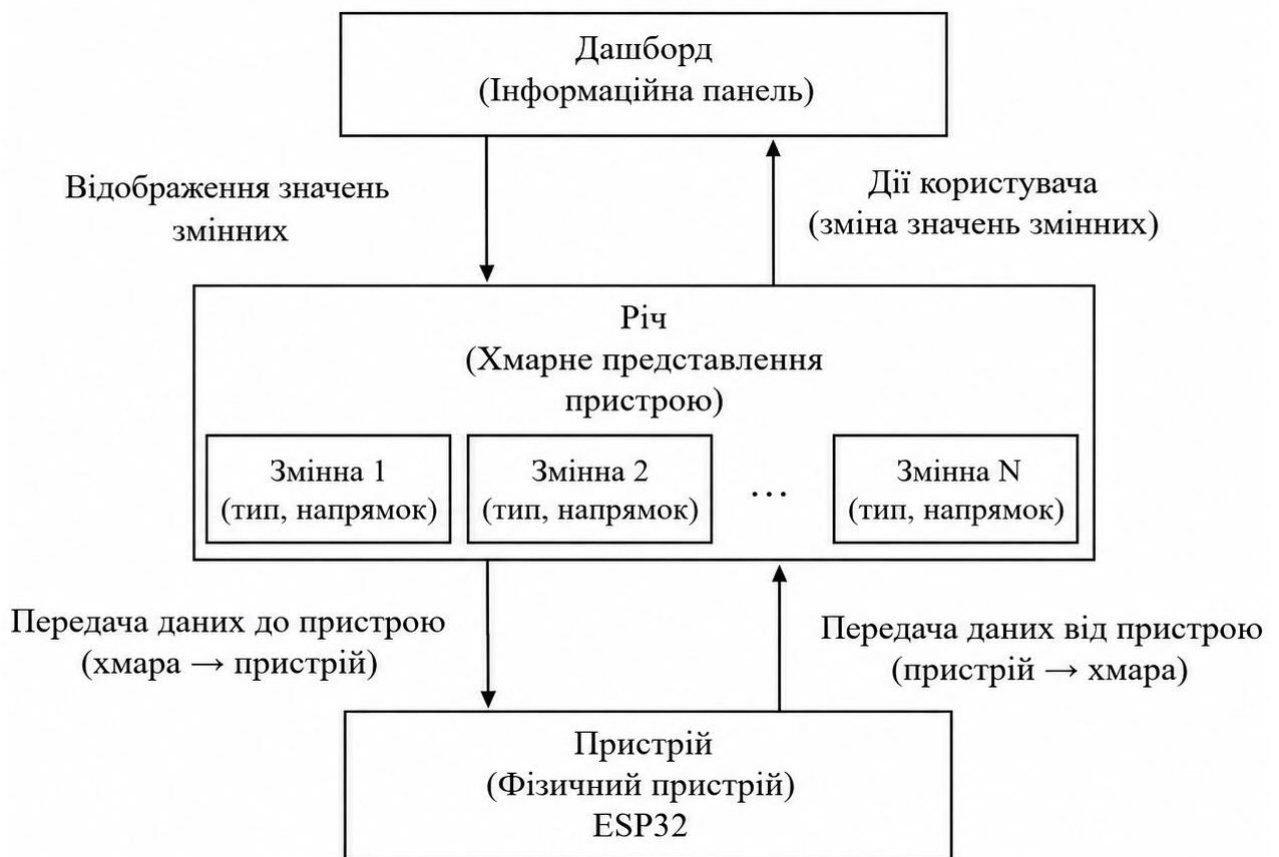


Рисунок 3.1 – Архітектура взаємодії в Arduino IoT Cloud

В рамках проєктування та реалізації програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі Arduino Cloud процес розгортання середовища розпочинався зі створення облікового запису на порталі create.arduino.cc.

Головна навігаційна панель платформи містила такі розділи: Things (сутності, що представляють пристрої), Devices (реєстрація фізичних або симульованих пристроїв), Dashboards (інтерфейси для взаємодії з даними), Triggers, Templates та Cloud Editor.

Після реєстрації на платформі першим кроком була створення пристрою в системі. Для цього у розділі Devices обрала опція додавання нового пристрою, після чого відкрився майстер налаштування, що пропонував кілька варіантів підключення: автоматичне налаштування для плат Arduino, підключення сумісних сторонніх пристроїв (до яких відноситься ESP32), ручне налаштування для розширених сценаріїв, а також підключення будь-якого пристрою через Python, MicroPython або Node.js (рис. 3.2).

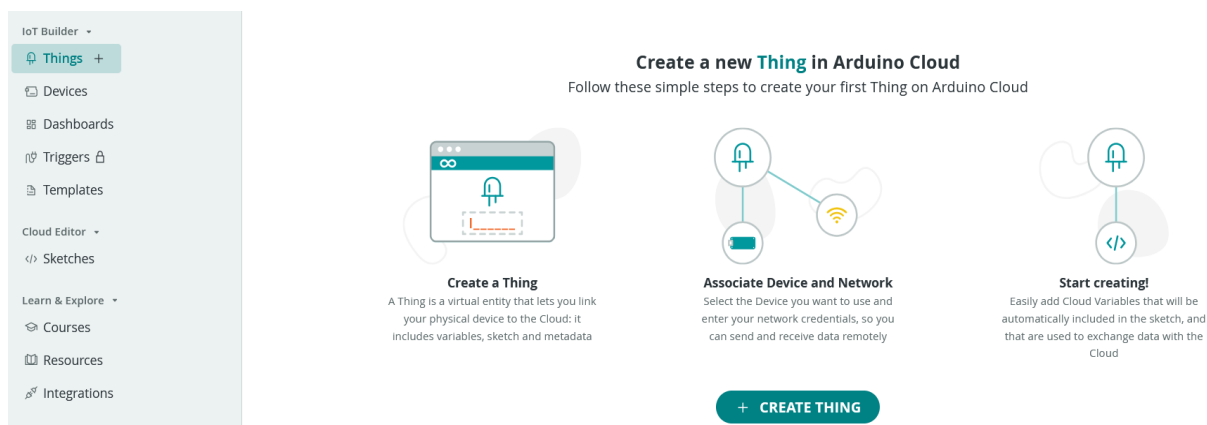


Рисунок 3.2 – Інтерфейс додавання нового пристрою в Arduino Cloud

Для даного проекту обрано варіант Compatible device, що відповідає ESP32 Dev Module. У діалозі пристрої provisioning виконувся вибір типу мікроконтролера – ESP32, після чого задається його модель ESP32 Dev Module (рис. 3.3). На другому кроці було призначено ім'я пристрою (у даному випадку ESP32_waterPlant). Після підтвердження вибору система згенерувала унікальні ідентифікаційні дані: Device ID та Secret Key. Device ID є унікальним ідентифікатором пристрою в хмарній системі, а Secret Key слугує паролем для автентифікації при встановленні з'єднання з ArduinoIoTCloud.

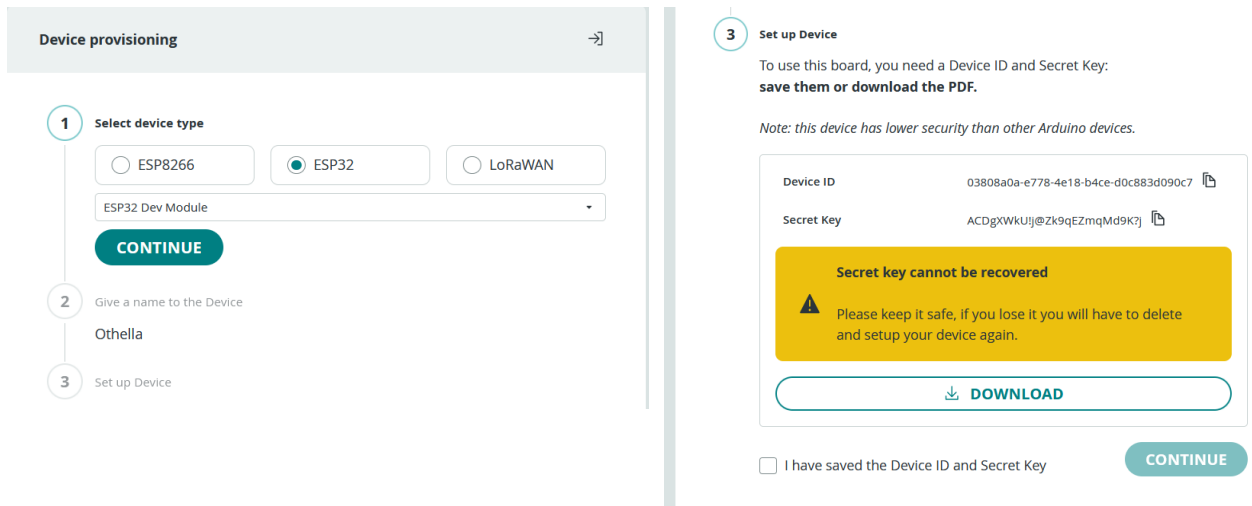


Рисунок 3.3 – Процес провізювання пристрою

Після успішної реєстрації створений пристрій ESP32_waterPlant відображався у списку Devices зі статусом OFFLINE, тобто пристрій створений, але його ще не під'єднано фізично. У панелі пристрою відображаються деталі: тип пристрою ESP32 Dev Module, ідентифікатор у форматі UUID, FQBN (Fully Qualified Board Name) – esp32:esp32:esp32, а також версія IoT-бібліотеки 2.9.0 та часові мітки активності.

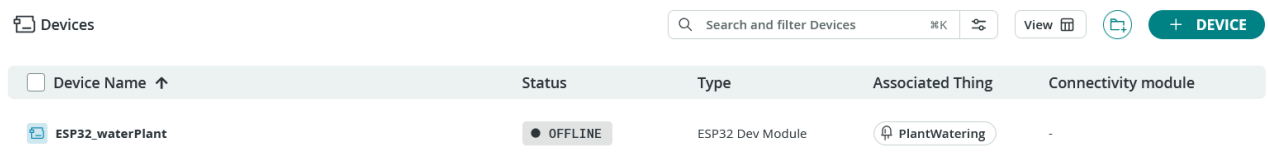


Рисунок 3.4 – Створені пристрої в Arduino Cloud

Наступним етапом було створення речі – абстрактної хмарної сутності, яка об'єднує пристрій, набір змінних та скетч. У розділі Things було обрано CREATE THING, після чого відкрився інтерфейс налаштування (рис. 3.5). Концептуально річчю є віртуальний двійник фізичного (або симульованого) пристрою: вона зберігає набір cloud-змінних, що синхронізуються між хмарою та мікроконтролером у режимі реального часу.

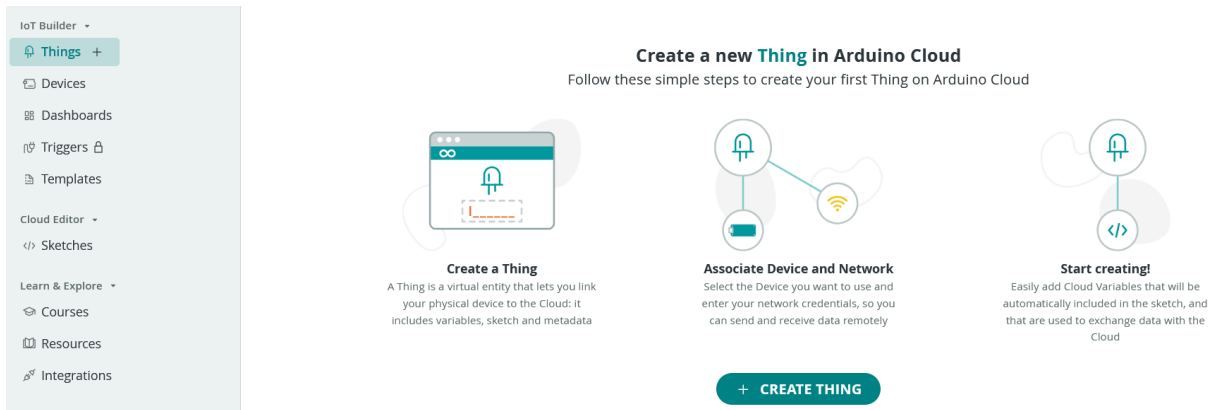


Рисунок 3.5 – Вітальний екран створення нового Thing в Arduino Cloud

Після створення речі (з іменем PlantWatering) відкривається вкладка Data, де можна додавати cloud-змінні. Платформа підтримує різноманітні типи даних, такі як Boolean, Integer Number, Floating Point Number, Character String, а також спеціалізовані типи, об'єднані у категорії Energy, Light and Color, Size and Motion, Smart Home та Time. Для даного проекту були визначені та додані такі змінні.

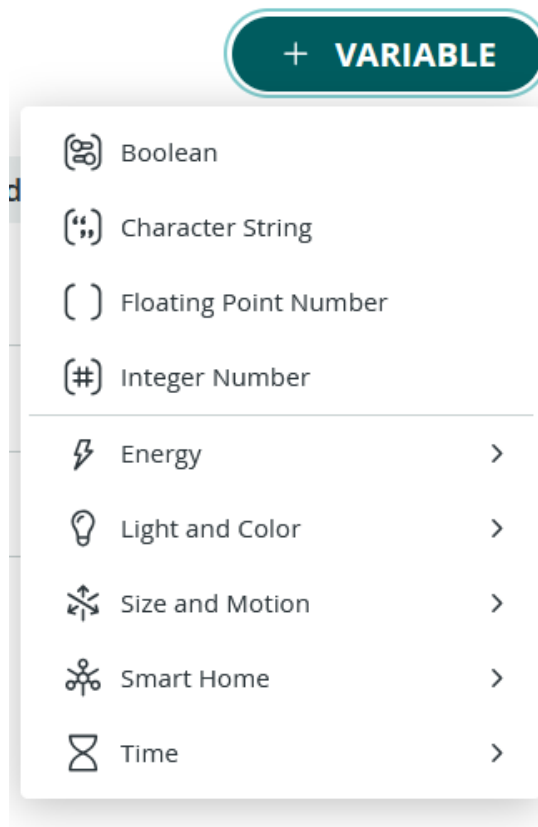


Рисунок 3.6 – Типи даних, доступні для cloud-змінних у Arduino Cloud

Змінна `sensor_1` (і аналогічні `sensor_2`, `sensor_3`) має тип `Integer Number` і налаштована як `Read Only` – тобто пристрій публікує значення вологості ґрунту у відсотках, а хмара лише зберігає та відображає їх. Змінна `automatic_mode` типу `Boolean` є `Read/Write`, що дозволяє як пристрою повідомляти хмару про поточний режим роботи, так і оператору змінювати режим з дашборду. Змінні `relay_1`, `relay_2`, `relay_3` також типу `Boolean` і `Read/Write` відповідають за ручне керування насосами. Змінні `threshold_1`, `threshold_2`, `threshold_3` типу `Integer Number` з доступом `Read/Write` дозволяють встановлювати пороги вологості для кожного каналу окремо. Нарешті, `plant_schedule_1`, `plant_schedule_2`, `plant_schedule_3` відносяться до спеціального типу `CloudSchedule` з категорії `Time` і забезпечують розкладне керування циклами поливу.

Під час виконання даної роботи було використано безкоштовний план `Arduino Cloud`, що обмежує загальну кількість змінних у п'ять одиниць. Тому було створено тільки по одній змінній для одного модуля, тобто по одній змінній на реле, давач вологості ґрунту, планувальник, порогове значення поливу та автоматичного поливу. Цю інформацію можна побачити на світлинці поданій на рис. 3.7.

Name	Declaration	Last Value	Last Update
automatic_mode	automatic_mode	N/A	-
plant_schedule_1	plant_schedule_1	N/A	-
relay_1	relay_1	N/A	-
sensor_1	sensor_1	N/A	-
threshold_1	threshold_1	N/A	-

Рисунок 3.7 – Список cloud-змінних `Thing PlantWatering` після їх додавання

Після додавання всіх змінних було виконано прив'язку пристрою до речі. У секції `Associated Device` обирається раніше зареєстрований пристрій `ESP32_waterPlant`. З цього моменту хмарна бібліотека `ArduinoIoTCloud`

автоматично генерує файл `thingProperties.h`, що містить оголошення всіх змінних та функцію `initProperties()` для їх реєстрації. Цей файл включається у головний скетч і забезпечує автоматичну двосторонню синхронізацію між локальними змінними на ESP32 та їх хмарними відповідниками.

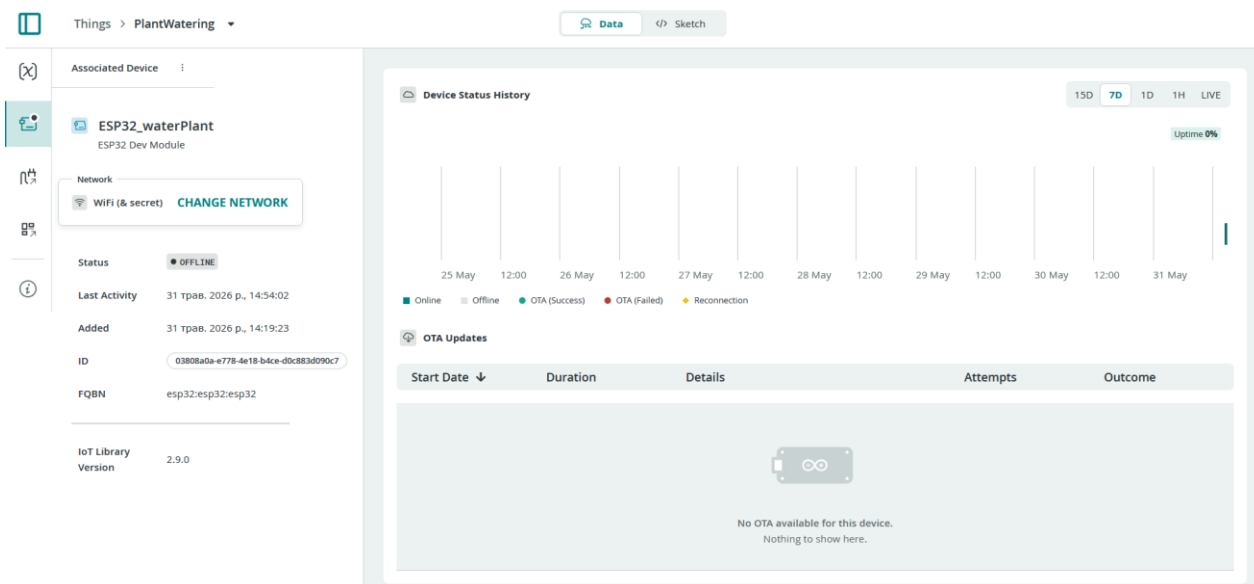


Рисунок 3.8 – Річ PlantWatering з прив’язаним пристроєм ESP32_waterPlant та графіком активності

3.2 Створення та тестування віртуального прототипу програмно-технічного засобу в Wokwi

Wokwi є сучасним онлайн-симулятором електронних схем, що підтримує широкий спектр мікроконтролерних платформ, зокрема ESP32, Arduino Uno, STM32 та інші. Принциповою перевагою цього інструменту у контексті даної роботи є вбудована підтримка мережевої взаємодії. Симульований ESP32 отримує доступ до мережі через внутрішню Wi-Fi точку доступу Wokwi-GUEST і може встановлювати реальне TCP/IP з’єднання з зовнішніми сервісами, зокрема з Arduino IoT Cloud. Це дозволяє повноцінно тестувати хмарну взаємодію без необхідності фізичного прошивання реального пристрою.

Схема віртуального прототипу, побудована в середовищі Wokwi, відтворює функціональну топологію реальної системи поливу і складається з таких компонентів як мікроконтролер, виконавчі пристрої та датчі (рис .3.9).

Центральним елементом є мікроконтролерна плата ESP32 Dev Module, яка вона виконує весь прикладний код, зчитує аналогові сигнали з датчів, управляє виходами насосів та підтримує з'єднання з Arduino Cloud через протокол MQTT по верх TLS.

Три потенціометри підключені до аналогових входів GPIO34, GPIO35 та GPIO32 відповідно і імітують ємнісні датчі вологості ґрунту. Обертання ручки потенціометра змінює напругу на виводі від 0 до 3,3 В, що після аналогово-цифрового перетворення та масштабування відображається як відсоток вологості від 0 до 100%.

Три світлодіоди, підключені до цифрових виходів GPIO26, GPIO27 та GPIO14 через обмежувальні резистори номіналом 220 Ом, замінюють реальні водяні насоси, таким чином що увімкнений LED сигналізує про активний цикл поливу відповідного каналу.

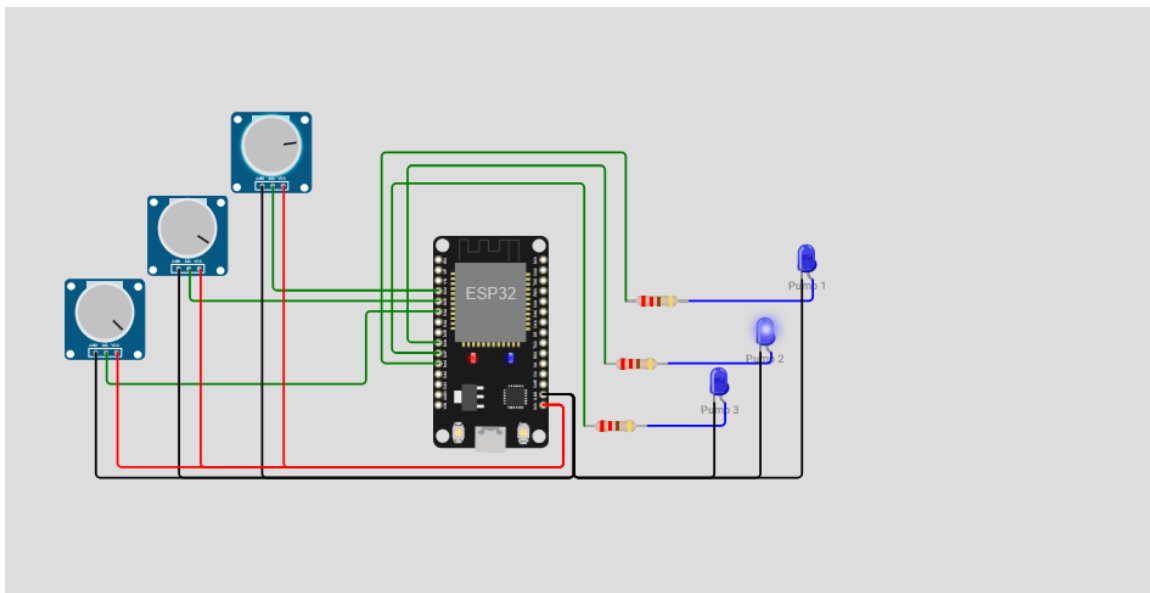


Рисунок 3.9 – Схема симуляції в Wokwi: ESP32, три потенціометри (датчі вологості) та три LED (насоси)

Кожен потенціометр підключений трьома виводами: середній (сигнальний) вивід через захисний резистор 1 кОм з'єднаний з аналоговим входом ESP32, лівий вивід до шини живлення 3,3 В, правий до загальної землі GND. Використання захисних резисторів обмежує струм при можливих коротких замиканнях та зменшує ємнісні завади на лінії ADC. Для виходів насосів кожен цифровий вивід ESP32 з'єднаний з анодом LED через резистор 220 Ом, катод LED підключено до GND. У реальній схемі замість LED встановлено транзисторні ключі або модулі реле, однак в рамках симуляції Wokwi LED є достатнім функціональним еквівалентом для перевірки логіки керування.

Програмне забезпечення прототипу складається з двох основних файлів: головного скетчу `sketch.ino` та заголовного файлу `thingProperties.h`. Файл `thingProperties.h` містить оголошення всіх cloud-змінних, реєстрацію їх у системі `ArduinoIoTCloud` через функцію `initProperties()`, а також об'єкт `WiFiConnectionHandler`, що відповідає за встановлення Wi-Fi з'єднання. Секретні дані `Device ID`, `Secret Key`, а також облікові дані мережі передаються через механізм `Wokwi Secrets`, щоб не зберігатися у відкритому коді.

Головна функція `setup()` виконує послідовну ініціалізацію системи. Спочатку встановлюється `Serial`-з'єднання зі швидкістю 115200 бод, конфігуруються пини насосів як виходи з початковим станом `LOW`, викликається `initProperties()` для реєстрації cloud-змінних, після чого `ArduinoCloud.begin()` ініціює процедуру підключення до хмарного сервісу. Основний цикл `loop()` виконується безперервно. Спочатку викликається `ArduinoCloud.update()` для обробки вхідних повідомлень від хмари та надсилання накопичених оновлень, потім зчитуються значення всіх трьох аналогових датчиків і публікуються у відповідних cloud-змінних. Якщо активний автоматичний режим, для кожного каналу перевіряється комбінація двох умов: активність розкладу (`plant_schedule_N.isActive()`) та недостатня вологість (`sensor_N < threshold_N`). При виконанні обох умов відповідний насос вмикається на фіксований час 5000 мс, після чого автоматично вимикається завдяки механізму таймерних міток.

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 51
Зм.	Арк.	№ док.ум.	Підпис	Дата		

Якщо автоматичний режим вимкнений, насоси керуються виключно через булеві змінні `relay_N`, значення яких надходять безпосередньо з дашборду.

Для верифікації функціональності системи в Wokwi виконано ряд тестових сценаріїв. У першому сценарії перевірялась коректність зчитування давачів: потенціометри поступово повертались від мінімального до максимального положення, і відповідні значення `sensor_1`, `sensor_2`, `sensor_3` оновлювались у реальному часі на дашборді Arduino Cloud, підтверджуючи правильність масштабування через функцію `map()`. У другому сценарії тестувалось ручне керування: при вимкненому `automatic_mode` перемикання `relay_1` з дашборду миттєво відображалось у стані LED Pump 1, що підтверджує двосторонню синхронізацію cloud-змінних. Третій сценарій перевіряв автоматичний режим: при встановленому розкладі та зниженні показників давача нижче порогу `threshold_1` відповідний LED автоматично вмикався на 5 секунд, після чого вимикався без будь-якого ручного втручання.

3.3 Налаштування дашборду та хмарних змінних

Дашборд у Arduino Cloud є центральним елементом взаємодії оператора із системою поливу. Він являє собою веб-сторінку з набором налаштовуваних віджетів, кожен з яких прив'язаний до однієї або декількох cloud-змінних і відображає їх поточне значення або дозволяє змінювати його. Для переходу до створення дашборду використовується розділ Dashboards у бічній навігаційній панелі.

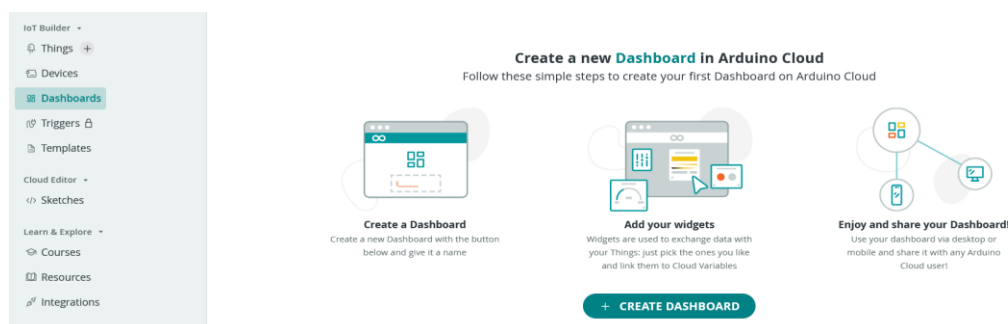


Рисунок 3.10 – створення нового дашборду WaterPlantDashboard

У рамках проекту було сформовано дашборд WaterPlantDashboard, що охоплює декілька типів віджетів. Перший тип це Gauge (стрілочний індикатор) використовується для відображення поточного рівня вологості ґрунту кожного з трьох каналів. Після розміщення віджета на полотні дашборду відкривається панель Widget Settings, де задається діапазон відображуваних значень (Min: 0, Max: 100) та виконується прив'язка до cloud-змінної через кнопку Link Variable. У вікні прив'язки відображається структурована ієрархія: Thing (PlantWatering) – Variables (sensor_1, sensor_2, sensor_3 тощо) – властивості вибраної змінної (тип Int, Permission: Read-Only, Update policy: On change). Такий підхід унеможливляє випадкову прив'язку змінної з некоректним типом або рівнем доступу.

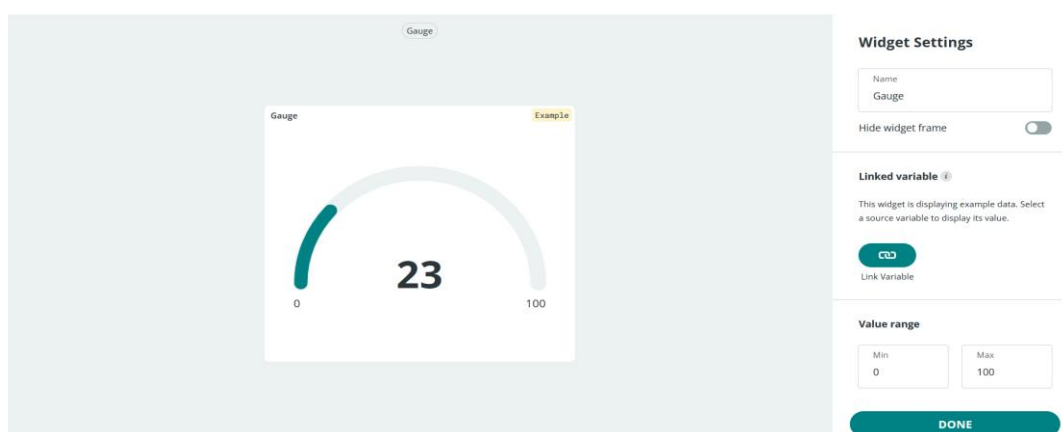


Рисунок 3.11 – Налаштування віджету Gauge для відображення вологості ґрунту

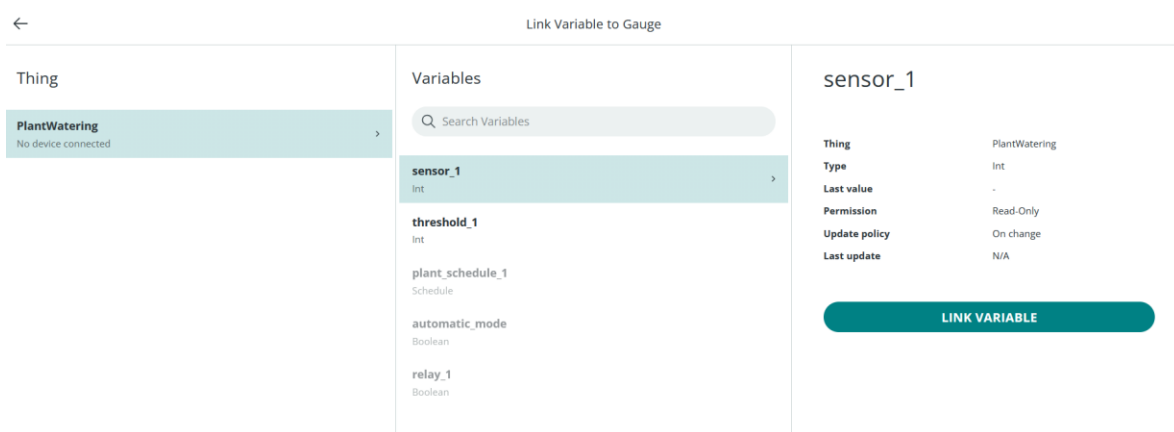


Рисунок 3.12 – Прив'язка змінної sensor_1 до віджету Gauge

Другий тип – Switch (перемикач) — застосовується для відображення та зміни булевих змінних таких як relay_1, relay_2, relay_3 та automatic_mode. Перемикачі relay забезпечують ручне вмикання насосів поза розкладом, тоді як перемикач automatic_mode визначає режим роботи всієї системи. Прив’язка relay_1 до Switch виконується аналогічно до Gauge, проте у правій панелі деталей змінної відображається Permission: Read/Write, що підтверджує можливість запису з дашборду.

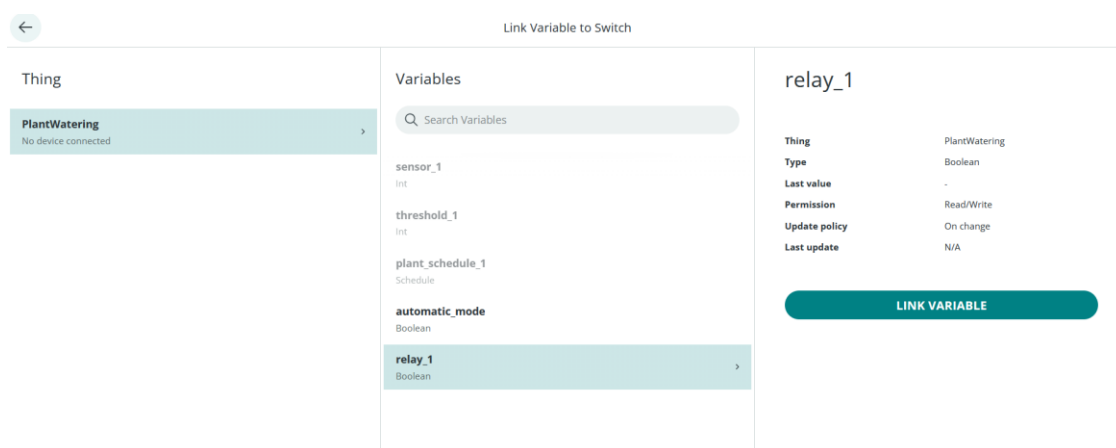


Рисунок 3.13 – Прив’язка змінної relay_1 до Switch-віджету з доступом Read/Write

Третій тип Slider або повзунок призначений для динамічного встановлення порогових значень вологості threshold_1, threshold_2, threshold_3. Повзунок налаштовано на горизонтальну орієнтацію з діапазоном від 0 до 100, що відповідає відсоткам вологості. Зміна положення повзунка на дашборді миттєво відображається у хмарній змінній і через механізм callback-функції onThresholdNChange() передається мікроконтролеру, де оновлює порогове значення для відповідного каналу поливу.

Четвертий тип Scheduler (планувальник) є унікальним для Arduino Cloud і відповідає за керування об’єктами CloudSchedule. Цей віджет відображає розклад поливу у вигляді чотирьох полів: Starting on (дата та час початку), Duration (тривалість активного вікна), Repeat every (частота повторення: Day,

Week тощо) та End recurrence (дата закінчення дії розкладу або Never). Часова зона відображається відповідно до налаштувань браузера у даному випадку Europe/Kyiv.

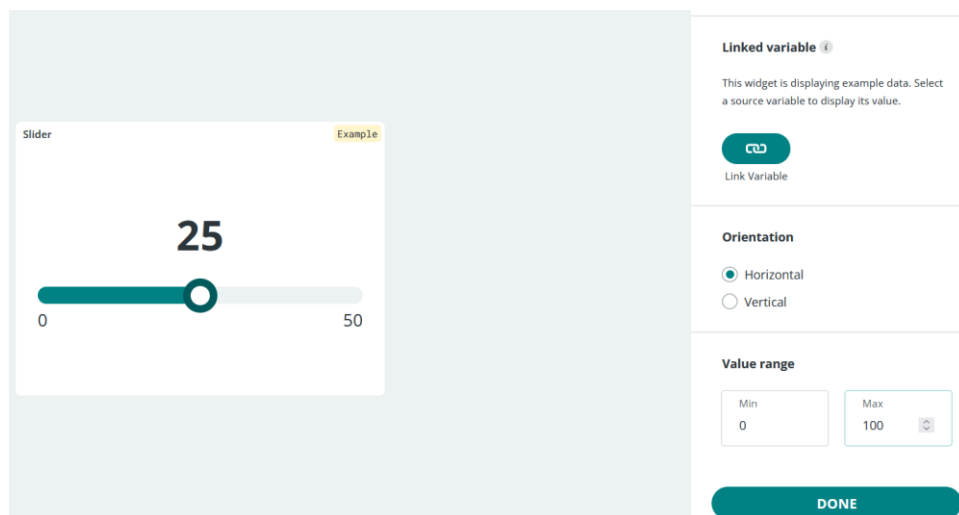


Рисунок 3.14 – Налаштування Slider-віджету для порогу вологості (діапазон 0–100)

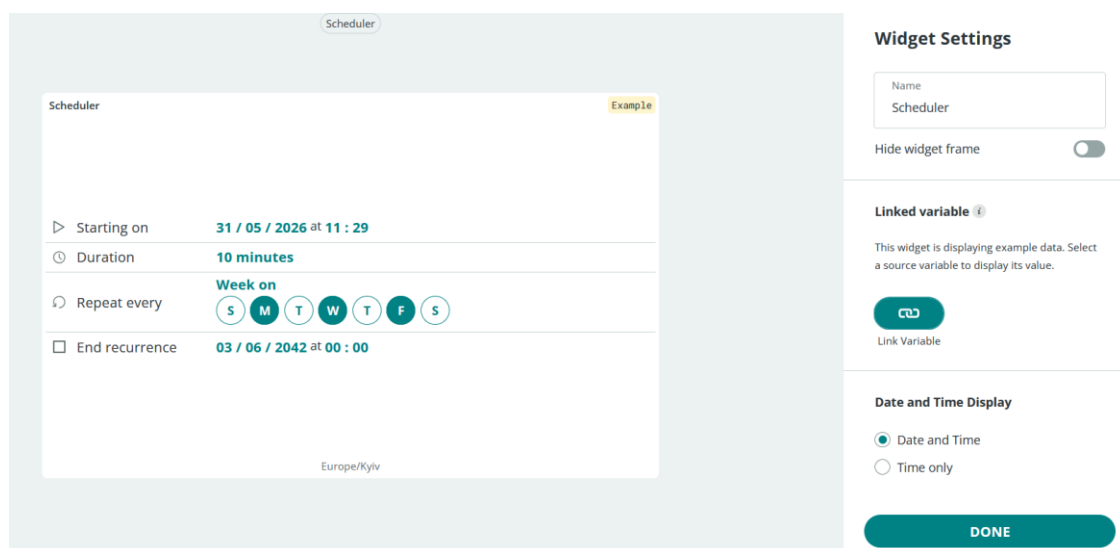


Рисунок 3.15 – Налаштування Scheduler-віджету для plant_schedule_1

При натисканні на елемент розкладу відкривається діалогове вікно Job Schedule, де у деталізованому вигляді налаштовуються параметри: час початку (Starting at), тривалість у годинах, хвилинах та секундах (Duration), частота

повторення (Repeat every: Day, Week, Month), а також умова завершення повторення (End recurrence: Never або конкретна дата). Наприклад, для щоденного поливу встановлено такі значення:

- Starting at: 14:45;
- Duration: 1 minute;
- Repeat every: Day;
- End recurrence: Never.

Цей розклад завантажується у cloud-змінну `plant_schedule_1` і через механізм `ArduinoCloud` синхронізується з мікроконтролером, де метод `isActive()` повертає `true` саме в момент активного вікна.

Job Schedule

Set Job

Starting at: 14 : 45 Europe/Kyiv

Duration: 00 HOURS 01 MINUTES 00 SECONDS

Repeat every: Day

End recurrence: Never On

31 / 05 / 2026

14 : 35 Europe/Kyiv

DONE

Рисунок 3.16 – Налаштування Job Schedule: час, тривалість та частота повторення

Фінальний вигляд дашборду `WaterPlantDashboard` у режимі перегляду відображає всі налаштовані віджети на єдиному полотні. Gauge відображає поточне значення вологості ґрунту (у тестовому сеансі показник сягнув 100%, що відповідає крайньому правому положенню потенціометра у симуляторі

Зм.	Арк.	№ докум.	Підпис	Дата

Wokwi). Перший Switch (`automatic_mode` або `relay_1`) знаходиться у стані ON, другий – у стані OFF. Slider показує значення 25, що відповідає встановленому порогу вологості 25%. Scheduler відображає налаштований розклад: Starting at 14:45, Duration 1 minute, Repeat every Day.

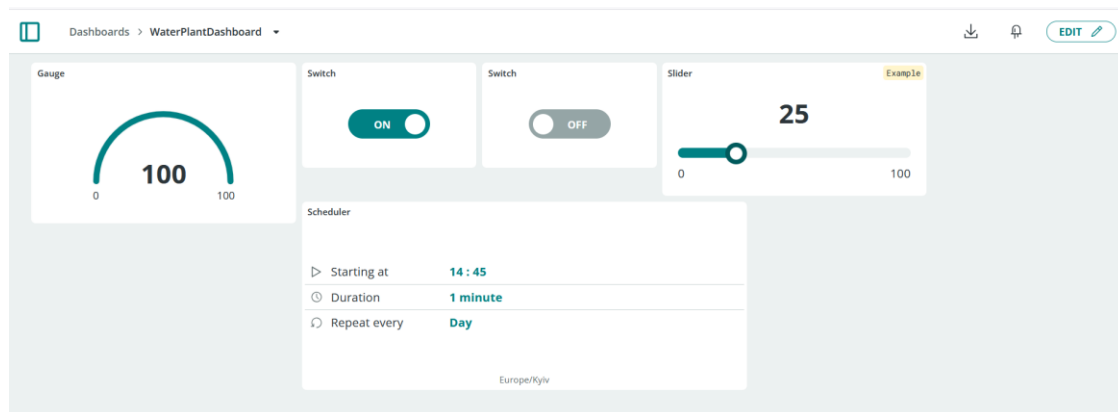


Рисунок 3.17 – Фінальний вигляд дашборду WaterPlantDashboard з активними даними

Під час тестового запуску симуляції в Wokwi при вмикнутому `automatic_mode` та активному розкладі було зафіксовано коректне вмикання відповідного LED при зниженні показань потенціометра нижче порогу (рис. 3.18). Після закінчення 5-секундного циклу LED автоматично вимкнувся. Паралельно на дашборді оновлювалось значення Gauge відповідно до повертання потенціометра, що підтверджує синхронізацію даних між Wokwi та Arduino Cloud у реальному часі.

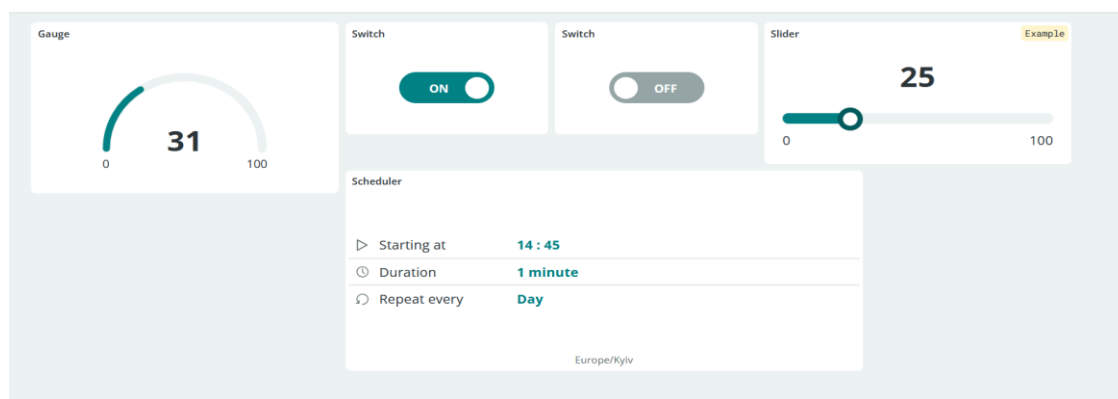


Рисунок 3.18 – Дашборд під час тестового сеансу

3.4 Практичне застосування та обмеження прототипу програмно-технічного засобу

Розроблений програмно-технічний засіб демонструє виразний практичний потенціал у сценаріях автоматизації поливу кімнатних рослин. Система надає можливість одночасно обслуговувати три незалежні канали поливу, кожен з яких має власний давач вологості, насос, налаштовуваний поріг вологості та розклад. Це дозволяє розмістити в одному приміщенні рослини з принципово різними вимогами до режиму зволоження, наприклад, тропічні рослини, кактуси та трав'яні культури і налаштувати параметри кожного каналу незалежно.

Ключовою практичною перевагою є можливість дистанційного моніторингу та керування через будь-який пристрій з браузером або мобільним додатком Arduino Cloud. Оператор може у будь-який момент переглянути актуальні показники вологості ґрунту, змінити порогові значення, скоригувати розклад поливу або примусово запустити насос у ручному режимі все це без фізичного доступу до пристрою. Архітектура системи передбачає збереження всіх cloud-змінних на боці хмари, тому навіть після тимчасової втрати з'єднання пристрій відновлює актуальний стан усіх параметрів після перепідключення.

Поєднання двох режимів роботи автоматичного та ручного забезпечує гнучкість у нестандартних ситуаціях. Якщо давач вологості надає некоректні показання або рослина потребує позачергового поливу, оператор може тимчасово перейти в ручний режим, виконати полив, а потім повернутись до автоматичного режиму, не вносячи змін у код прошивки.

Разом з тим прототип має ряд обмежень. По-перше, безкоштовний рівень Arduino Cloud (Free Plan) обмежує кількість cloud-змінних до 25, частоту оновлення даних до 1 разу на 15 секунд, а також накладає обмеження на кількість одночасно підключених пристроїв. Для масштабування системи на більшу кількість каналів або рослин може знадобитись перехід на платний тарифний план. По-друге, система повністю залежить від наявності стабільного Wi-Fi

з'єднання та доступності сервісів Arduino Cloud. Відсутність з'єднання унеможлиблює отримання оновлень розкладу та зміни порогових значень з дашборду, хоча останні збережені в пам'яті мікроконтролера значення продовжують виконуватись локально. По-третє, в поточній реалізації давачі вологості симулюються потенціометрами, а насоси – світлодіодами. При переході до реального розгортання необхідно враховувати особливості роботи з ємнісними давачами вологості, зокрема стабілізацію показань після занурення зонда, а також правильний вибір транзисторів або модулів реле для комутації водяних насосів постійного струму.

3.6 Висновки до третього розділу

У третьому розділі виконано практичну реалізацію та всебічне тестування програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі ESP32 та Arduino IoT Cloud. Описано процес розгортання хмарного середовища, зокрема реєстрацію пристрою у системі Arduino Cloud, отримання та збереження автентифікаційних даних, створення віртуальної речі та визначення повного набору cloud-змінних. Також розглянуто побудову віртуального прототипу в симуляторі Wokwi. Описано структуру програмного коду прошивки, логіку автоматичного та ручного режимів керування, а також механізм автоматичного вимкнення насосів через таймерні мітки. Проведено налаштування дашборду для відображення даних про поливі.

					КВРКІ. 240494.22.04.10 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У роботі вирішено актуальну задачу проектування, реалізації та симуляції програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі мікроконтролера ESP32 та хмарної платформи Arduino IoT Cloud, що дозволяє забезпечити безперервний трьохканальний моніторинг вологості ґрунту, автономний розкладний полив та дистанційне керування через веб-дашборд. У ході виконання роботи отримано такі результати.

Проведено аналіз предметної області та огляд існуючих рішень автоматизованого поливу кімнатних рослин. Розглянуто комерційні системи та відкриті DIY-рішення, виявлено їх переваги й обмеження. Встановлено, що платформа Arduino IoT Cloud є найбільш доцільним вибором для реалізації трьохканальної системи завдяки нативній підтримці типу CloudSchedule, вбудованому механізму двонаправленої синхронізації cloud-змінних через MQTT/SBOR та безкоштовному тарифному плану з достатньою кількістю змінних для даного проекту.

Сформульовано перелік функціональних та нефункціональних вимог до системи, зокрема трьохканальний моніторинг вологості в реальному часі, автоматичний полив за розкладом при досягненні порогу, ручне керування насосами з дашборду, динамічне налаштування порогів без перепрошивання, живлення від літій-іонного акумулятора з можливістю заряджання, захист від повторного запуску насоса в поточному циклі та інші.

Розроблено структуру апаратної частини засобу, що включає п'ять функціональних блоків. Розроблено схему електричну принципову у середовищі EasyEDA, що відображає повну топологію з'єднань між усіма компонентами системи.

Розроблено алгоритм функціонування мікроконтролера та наведено його блок-схему. Також розроблено програмне забезпечення прошивки ESP32 на основі бібліотеки ArduinoIoTCloud 2.9.0 та описано програмну реалізацію

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

хмарної взаємодії. Визначено і зареєстровано тринадцять cloud-змінних у файлі thingProperties.h: sensor_1–3, relay_1–3 та automatic_mode, threshold_1–3, plant_schedule_1–3. Реалізовано десять callback-функцій для реактивного реагування на команди з дашборду. Описано механізм CBOR-серіалізації та паттерн Observer бібліотеки ArduinoIoTCloud як основу двонаправленої синхронізації.

Виконано розгортання хмарного середовища Arduino IoT Cloud, зокрема зареєстровано пристрій ESP32_waterPlant типу ESP32 Dev Module, отримано унікальні облікові дані, створено Thing PlantWatering з повним набором cloud-змінних та прив'язано пристрій до Thing. Зафіксовано статус ONLINE та активну синхронізацію між симулятором і хмарою.

Побудовано віртуальний прототип системи у симуляторі Wokwi. Визначено практичні обмеження прототипу: залежність від стабільного Wi-Fi-з'єднання для оновлення параметрів з дашборду; обмеження безкоштовного тарифного плану Arduino IoT Cloud (25 змінних, один пристрій); необхідність емпіричного калібрування констант ADC_DRY та ADC_WET при переході від симуляції до реального заліза; відсутність у Wokwi точного відтворення часових характеристик реального АЦП та затримок Wi-Fi.

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Грачов О. Plant monitoring system: III-система для розумного моніторингу рослин. *Інформаційні технології та суспільство*. 2025. № 1 (16). С. 59–64. DOI: <https://doi.org/10.32689/maup.it.2025.1.7>

2. Internet of Things для фермерів і дачників, кейс "Розумна садиба". *Evergreens*. URL: <https://evergreens.com.ua/ua/articles/case-smart-farmstead.html> (дата звернення: 01.06.2026).

3. Підвищення ефективності сільського господарства за допомогою IoT-технологій Milesight. *Megatrade*. URL: <https://megatrade.ua/news/reviews/pidvishchennya-efektivnosti-silskogo-gospodarstva-za-dopomogoyu-iot-tekhnologiy-milesight/> (дата звернення: 01.06.2026).

4. Про екосистему IoT. *Aeroponics.one*. URL: https://aeroponics.one/?page_id=1052 (дата звернення: 01.06.2026).

5. Використання технологій Інтернету речей в агробізнесі. *Agronomy.com.ua*. URL: <https://agronomy.com.ua/statti/1625-vykorystannia-tekhnologii-internetu-rechei-v-ahrobiznesi.html> (дата звернення: 01.06.2026).

6. Сучасні цифрові технології в агробізнесі: воєнні ризики та адаптація до нових реалій. *Liga Zakon*. URL: https://biz.ligazakon.net/analytics/238394_suchasntsifrov-tekhnolog-v-agrobznes-vonn-riziki-ta-adaptatsya-do-novikh-realy (дата звернення: 01.06.2026).

7. Розумні теплиці. *iotji.io*. URL: <https://iotji.io/solutions/rozumni-teplytsi/> (дата звернення: 01.06.2026).

8. IoT-Enabled Environment Monitoring: Real-Time Data for Greenhouses. *Miilkiiia Grow*. URL: <https://www.miilkiiagrow.com/uk/news/iot-enabled-environment-monitoring-real-time-data-for-greenhouses/> (дата звернення: 01.06.2026).

9. Чому LoRaWAN краща за інші технології для розумних сільськогосподарських рішень. *Jooby*. URL: <https://jooby.eu/uk/blog/chomu->

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

lorawan-krasha-za-inshi-tehnologiyi-dlya-rozumnih-silskogospodarskih-rishen/ (дата звернення: 01.06.2026).

10. Розбираємо роль хмарних технологій в IoT пристроях: в чому їх переваги? *GigaCloud*. URL: <https://gigacloud.ua/articles/rozbyrayemo-rol-hmarnyh-tehnologij-v-iot-prystroyah-v-chomu-yih-perevagu/> (дата звернення: 01.06.2026).

11. Розробка Інтернет речей (AIoT) у розумному сільському господарстві. *Panda Team*. URL: <https://pandateam.net.ua/razrobotka-aiot/> (дата звернення: 01.06.2026).

12. IoT протоколи (COAP). *Greenhouse*. URL: <https://greenhouse.cv.ua/?p=598> (дата звернення: 01.06.2026).

13. MQTT для IoT: чому це головний протокол інтернету речей. *Aziot*. URL: <https://aziot.com.ua/expert/mqtt-dlya-iot-chomu-tse-holovnyy-protokol-internetu-rechey/> (дата звернення: 01.06.2026).

14. Arduino Cloud | Build, Control, Monitor Your IoT Projects. *Arduino Cloud*. URL: <https://cloud.arduino.cc> (дата звернення: 01.06.2026).

15. Abayomi-Alli O., Odusami M., Ojinaka D., Shobayo O., Misra S., Damasevicius R. Smart-Solar Irrigation System (SMIS) for Sustainable Agriculture. In: Florez H., Diaz C., Chavarriaga J. (eds.). *Applied Informatics*. Cham: Springer International Publishing, 2018. Vol. 942. P. 198–212.

16. Mogili U. R., Deepak B. B. V. L. An intelligent drone for agriculture applications with the aid of the MAVlink protocol. In: *Innovative Product Design and Intelligent Manufacturing Systems: Select Proceedings of ICIPDIMS 2019*. Singapore: Springer, 2020. P. 195–205.

17. Raja M., Nithish N. M., Shankar B. S., Sadhurwanth D. Automatic Irrigation and Crop Protection System Based on IoT. In: Peter J. D., Fernandes S. L., Alavi A. H. (eds.). *Disruptive Technologies for Big Data and Cloud Applications*. Singapore: Springer, 2022. Vol. 905. P. 355–363.

18. Mogili U. R., Deepak B. B. V. L. Study of takeoff constraints for lifting an agriculture pesticide sprinkling multi-rotor system. In: *Advances in Materials and*

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

Manufacturing Engineering: Proceedings of ICAMME 2019. Singapore: Springer, 2020. P. 203–210.

19. Boursianis A. D., Papadopoulou M. S., Gotsis A., Wan S., Sarigiannidis P., Nikolaidis S., Goudos S. K. Smart Irrigation System for Precision Agriculture—The AREThOU5A IoT Platform. *IEEE Sens. J.* 2021. Vol. 21. P. 17539–17547.

20. Stolojescu-Crisan C., Butunoi B. P., Crisan C. An IoT Based Smart Irrigation System. *IEEE Consum. Electron. Mag.* 2022. Vol. 11. P. 50–58.

21. Garg D., Alam M. Smart Agriculture: A Literature Review. *J. Manag. Anal.* 2023. Vol. 10. P. 359–415.

22. Gamal Y., Soltan A., Said L. A., Madian A. H., Radwan A. G. Smart Irrigation Systems: Overview. *IEEE Access.* 2024. P. 1–1.

23. Tephila M. B., Sri R. A., Abinaya R., Lakshmi J. A., Divya V. Automated Smart Irrigation System Using IoT with Sensor Parameter. *Proceedings of the 2022 International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin, India, 16–18 March 2022. Piscataway: IEEE, 2022. P. 543–549.

24. Wokwi - World's most advanced ESP32 Simulator. *Wokwi*. URL: <https://wokwi.com> (дата звернення: 02.06.2026).

25. Що таке модуль реле 5В і як він працює. *QJChina Relay*. URL: <https://ua.qjchina-relay.com/info/what-is-a-5v-relay-module-and-how-does-it-work-17381390877484032.html> (дата звернення: 02.06.2026).

26. Що таке резистор Pull-Up і як він працює. *Ariat Tech*. URL: <https://ua.ariat-tech.com/blog/What-Is-a-Pull-Up-Resistor-and-How-Does-It-Works.html> (дата звернення: 02.06.2026).

27. Вибір та застосування резисторів підтяжки та резистори, що знижують. *Altera Price*. URL: <https://www.altera-price.com/ua/news/selection-and-application-of-pull-up-resistors-and-pull-down-resistors.htm> (дата звернення: 02.06.2026).

28. Розумний дім #2: Wi-Fi керування лампочкою через реле. *Geekmatic*. URL: https://geekmatic.in.ua/ua/umnyj_dom_esp32_wifi_lampochka_rele (дата звернення: 02.06.2026).

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 64
Зм.	Арк.	№ доквм.	Підпис	Дата		

29. Мікроконтролер ESP32. *IT Master*. URL: <https://itmaster.biz.ua/directory/microcontrollers/esp32.html> (дата звернення: 02.06.2026).

30. Stojescu-Crisan C., Crisan C., Butunoi B. P. An IoT-Based Smart Home Automation System. *Sensors*. 2021. Vol. 21. P. 3784.

31. Yassine A., Singh S., Hossain M. S., Muhammad G. IoT big data analytics for smart homes with fog and cloud computing. *Future Gener. Comput. Syst.* 2019. Vol. 91. P. 563–573.

32. Froiz-Míguez I., Fernández-Caramés T. M., Fraga-Lamas P., Castedo L. Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes. *Sensors*. 2018. Vol. 18. P. 2660.

33. Patil N., Nimbalkar A., Pawar G., Shardul S. Home Automation System Using ESP32 and Alexa. *Int. J. Fundam. Multidiscip. Res. (IJFMR)*. 2023. Vol. 5. P. 101–110.

34. Sonawane S. S., Patil S. S. IoT Based Smart Home Automation Using ESP32. *Int. J. Eng. Res. Technol. (IJERT)*. 2023. Vol. 12.

35. Thomas L., MV M. K., SL S. D., BS P. Towards Comprehensive Home Automation: Leveraging the IoT, Node-RED, and Wireless Sensor Networks for Enhanced Control and Connectivity. *Eng. Proc.* 2023. Vol. 59. P. 173.

36. Patil N., Nimbalkar A., Pawar G., Shardul S. Voice Controlled Smart Home Automation Using Alexa and ESP32. *Int. J. Fundam. Multidiscip. Res. (IJFMR)*. 2023. Vol. 5. P. 101–110.

37. Dawande N. A., Morye R., Sarode D., Siddiqui N. IOT Based Home Automation System over Cloud. *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)*. 2023. Vol. 11. P. 7034–7039.

38. Lee J., Park H. Hybrid Architectures for Industrial and Domestic IoT. *Sensors*. 2022. Vol. 22. P. 3120.

39. Demrozi F., Turetta C., Pravadelli G. SHPIA 2.0: An Easily Scalable, Low-Cost, Multi-purpose Smart Home Platform for Intelligent Applications. *SN Comput. Sci.* 2024. Vol. 5. P. 42.

40. Huang K. C., Huang W. Y., Liu T. L., Zhuang Z. X. Integrating Virtual and Real System Platforms Based on IIoT Technology: A Case Study of PLC and Automation Studio. *Genet. Evol. Comput.* 2024. Vol. 1145. P. 298–308.

41. Espressif Systems. ESP32-Wi-Fi and Bluetooth SoC. 2025. URL: <https://www.espressif.com.cn/en/products/socs/esp32> (дата звернення: 10.03.2025).

42. Raspberry Pi Foundation. Raspberry Pi-Official Website. 2025. URL: <https://www.raspberrypi.com/> (дата звернення: 10.03.2025).

43. MQTT Organization. MQTT-The Standard for IoT Messaging. 2025. URL: <https://www.elkoep.com/mqtt> (дата звернення: 10.03.2025).

44. Espressif Systems. ESP32 SoCs—Espressif Systems. 2025. URL: <https://www.espressif.com/en/products/socs> (дата звернення: 10.02.2025).

45. Espressif Systems. ESP32 Series Datasheet Version 5.0. *Espressif Systems, 2.4 GHz Wi-Fi + Bluetooth + Bluetooth LE SoC*. 2025. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (дата звернення: 10.06.2025).

46. FreeRTOS Project. FreeRTOS Official Website, 2025 Real-Time Operating System for Microcontrollers and Small Microprocessors, MIT License, 40+ Architectures Supported. URL: <https://www.freertos.org/> (дата звернення: 02.09.2025).

47. Zephyr Project Contributors. Zephyr Project Documentation (Latest), 2025 Documentation for Zephyr Version 4.2.99, Apache 2.0 Licensed, Last Generated on 2 September 2025. URL: <https://docs.zephyrproject.org/latest/index.html> (дата звернення: 02.09.2025).

48. Serepas F., Papias I., Bellos N., Marinakis V. A Comprehensive Approach to Real-Time and Batch Processing for Energy-Efficient IoT Homes: Leveraging Lambda Architecture and Data Lakes. *Proceedings of the 2024 15th International*

Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 17–19 July 2024. P. 1–6.

49. Guoqiang S., Yanming C., Chao Z., Yanxu Z. Design and Implementation of a Smart IoT Gateway. *Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, Beijing, China, 20–23 August 2013. P. 720–723.

50. Li S., Xu L., Zhao S. The internet of things: A survey. *Inf. Syst. Front.* 2014. Vol. 17. P. 243–259.

51. Trappey A. J. C., Trappey C. V., Govindarajan U. H., Chuang A. C., Sun J. J. H. A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. *Adv. Eng. Inform.* 2017. Vol. 33. P. 208–229.

52. Morabito R., Petrolo R., Loscrì V., Mitton N. LEGIoT: A Lightweight Edge Gateway for the Internet of Things. *Future Gener. Comput. Syst.* 2018. Vol. 81. P. 1–15.

					КВРКІ. 240494.22.04.10 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Назар Стрільчук

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-4

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

3 червня 2026 року



Thu Jun 04 11:55:43 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилоч в документах: 13%**

ID: 273523 Назва: БКР Програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud Додано в БД: 2026-06-04 Автора: Назар СТРИЛЬЧУК Керівники: Олег САВЕНКО Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	88225	631	1504 (2%)	19 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Назар СТРИЛЬЧУК

Співавтор:

Назва: Програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

Експерт: Олег САВЕНКО

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 4.53%

Коефіцієнт подібності 2: 1.78%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-06-04 17:51:37.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2026-06-04

Дата



Доцент Андрій Нічепорук

експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

Автор Назар СТРІЛЬЧУК

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: д.т.н, професор, Олег САВЕНКО

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 4,53% і адресується до 21 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

05.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Підпис

Підпис

Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК
Ім'я, ПРІЗВИЩЕ

Олег САВЕНКО
Ім'я, ПРІЗВИЩЕ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Назар Стрільчук

Тема: Програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 61

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано програмно-технічний засіб керування автоматичним поливом кімнатних рослин на основі Arduino Cloud

2. Висновок про відповідність роботи дипломному завданню _____
Дипломний проект відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз актуальності задачі автоматизації поливу кімнатних рослин, розглянуто сучасні підходи до моніторингу вологості ґрунту та технології IoT, що застосовуються для побудови систем керування. Також проаналізовано методи вимірювання вологості ґрунту та сформульовано постановку задачі. У другому розділі здійснено проектування програмно-технічного засобу керування автоматичним поливом кімнатних рослин на основі платформи Arduino Cloud. Розглянуто вимоги до системи, розроблено її архітектуру, електричну схему, алгоритм роботи мікроконтролера та описано програмну реалізацію взаємодії з хмарною платформою. У третьому розділі представлено процес тестування та практичної реалізації розробленого програмно-технічного засобу. Описано розгортання середовища в Arduino Cloud, створення та перевірку віртуального прототипу в Wokwi, налаштування дашборду та хмарних змінних, а також наведено результати практичного застосування та обмеження розробленого рішення.

4. Позитивні сторони роботи: Розроблений програмно-технічний засіб забезпечує трьохканальний автономний полив кімнатних рослин у режимі реального часу з можливістю дистанційного моніторингу вологості ґрунту

5. Негативні сторони роботи: У роботі не розглянуто питання енергоефективності системи в режимі очікування, зокрема можливість переведення ESP32 у режим глибокого сну для мінімізації споживання акумулятора у тривалих паузах між циклами поливу.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

8. Інші зауваження: —

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 75 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)_

Бєєвський Леонід Петрович, зав. каф. ІІІЗ, ХНУ