

УДК 007.5

КОМП'ЮТЕРИЗОВАНІ ОПТИЧНІ ЕКСПЕРИМЕНТИ З ВИКОРИСТАННЯМ ПОРТАТИВНИХ СПЕКТРОМЕТРІВ ТА ІНТЕГРАЦІЮ З AMAZON WEB SERVICES

Андрій Крупич^{ORCID}, Іван Карбовник^{ORCID}

Львівський національний університет імені Івана Франка,
вул. Ген. Тарнавського, 107, 79017 Львів, Україна

Крупич, А., Карбовник, І. (2025). Комп'ютеризовані оптичні експерименти з використанням портативних спектрометрів та інтеграцією з Amazon Web Services. *Електроніка та Інформаційні технології*, 29, 123-134. <https://doi.org/10.30970/eli.29.11>

АНОТАЦІЯ

Передумови. У статті представлено систему автоматизації оптичних експериментів із використанням портативних спектрометрів компанії StellarNet та хмарних сервісів Amazon Web Services (AWS). Актуальність дослідження зумовлена необхідністю організації віддаленого доступу до лабораторного обладнання, що стало особливо критичним під час пандемії COVID-19. Запропоноване рішення дозволяє проводити експерименти в режимі реального часу, автоматизувати збір та обробку спектральних даних, а також зберігати їх у хмарі для подальшого аналізу.

Методи та інструменти. Методологічна частина містить огляд сучасних підходів до автоматизації експериментальних лабораторій. Розглянуто різні апаратні та програмні рішення, включаючи використання вбудованих засобів, таких як Raspberry Pi, та середовищ розробки на основі платформ LabVIEW та Інтернету речей (IP, Internet of Things, - IoT). Особлива увага приділена реалізації віддаленого доступу та можливості інтеграції з іншими пристроями Інтернету речей. Описано використане апаратне забезпечення, а також програмні інструменти.

Результати та обговорення. У результаті дослідження продемонстровано розроблений Python-скрипт для інтеграції спектрометра з AWS IoT Core. Описано процес налаштування підключення, калібрування спектрометра та безперервного збору спектральних даних у реальному часі. Дані передаються в хмарне середовище через протокол MQTT (Message Queue Telemetry Transport, Телеметричний транспорт черги повідомлень), що дозволяє проводити віддалений моніторинг та аналіз результатів. Розглянуто технічні аспекти реалізації, включаючи механізми безпечного з'єднання, підписки та публікації повідомлень у хмарному середовищі AWS.

Висновки. В роботі продемонстровано, що запропонована система забезпечує ефективний та гнучкий підхід до автоматизації лабораторних експериментів. Вона дозволяє значно розширити можливості досліджень у STEM-дисциплінах, зменшуючи витрати на обслуговування обладнання та покращуючи доступність лабораторних ресурсів. Запропоноване рішення може бути масштабоване для використання в інших наукових сферах, де необхідний дистанційний доступ до експериментального обладнання та безперервний збір даних.

Ключові слова: автоматизація, опрацювання даних, Інтернет речей, спектроскопія, AWS, хмарні обчислення



ВСТУП

Робота в наукових, технологічних, інженерних та математичних дисциплінах (STEM - Science, Technology, Engineering, Mathematics) значною мірою залежить від різноманітних видів лабораторної діяльності. Як правило, критичним є доступ до приміщень, де фізично розташоване обладнання, аби мати можливість проводити експерименти з реальними приладами. Ефективність та результативність таких досліджень обмежена, у тому числі й через логістичні фактори. У результаті, значна частина експериментального обладнання використовується значно менше, ніж це передбачено у рамках життєвого циклу приладів. Ці проблеми підкреслюють неефективність традиційних лабораторних практик. Крім того, надзвичайні ситуації, такі як пандемія COVID-19, ще більше ускладнюють регулярний доступ до лабораторних ресурсів [1].

У цій роботі представлено систему автоматизації експерименту з використанням спектрометра StellarNet та Amazon Web Services (AWS). Система дозволяє збирати дані зі спектрометра в режимі реального часу та надсилати їх в хмарне середовище. Там дані зберігаються в базі даних, що важливо для архіву експерименту. Дані опрацьовуються за правилами, заданими користувачем, і виводяться на окрему веб-сторінку для моніторингу в режимі реального часу. Передбачено завантаження архівних даних з бази за певний період для часового аналізу, агрегації та усереднення. Система гнучка та може розширюватись додатковими пристроями IP. Потенційно така система зможе об'єднати широкий набір давачів та пристроїв і забезпечити автоматизацію широкого спектру експериментів з постійним моніторингом результатів.

МАТЕРІАЛИ ТА МЕТОДИ

Зазвичай, розробка віддаленої лабораторії включає застосування персонального комп'ютера разом із пов'язаним інтерфейсом і технологіями веб/хмарного хостингу. Це передбачає значні інвестиції для введення в експлуатацію та подальшого обслуговування. У роботі [1] автор виступив з ініціативою щодо використання вбудованого процесора для розробки віддаленої лабораторії. При такому розташуванні функції персонального комп'ютера та сервера бере на себе вбудований процесор. У цьому проекті одноплатний мікрокомп'ютер Raspberry Pi використовується як системний контролер, а також як веб-сервер. Raspberry Pi, невелика обчислювальна система з власною операційною системою, замінила повноцінний комп'ютер/сервер, тим самим зменшивши вартість і складність проектування віддаленої лабораторії. До розроблених експериментів можна отримати віддалений доступ через Інтернет за допомогою відповідного графічного інтерфейсу користувача (GUI - Graphic User Interface). Графічний інтерфейс супроводжується трансляцією відео в реальному часі за допомогою камери Raspberry Pi, тому користувач може переглядати відео пов'язаних з експериментом процесів у реальному часі. Система містить функцію, завдяки якій користувачі можуть завантажувати дані у локальну систему для автономного аналізу.

У роботі [2] завдання автоматизації експерименту вирішується за допомогою системи збору даних, реалізованою на основі платформи LabVIEW (Laboratory Virtual Instrument Engineering Workbench) у поєднанні з платою Arduino. Ця запропонована система є універсальною та може бути застосована поза лабораторіями, у сонячних фотоелектричних установках, сільському господарстві та екологічних дослідженнях. Важливість цього програмного каркасу полягає у використанні апаратного забезпечення з відкритим вихідним кодом, який можна налаштувати відповідно до конкретних потреб різних застосувань.

У статті [3] представлено архітектуру IP, розроблену для потреб наукової галузі клітинної біології. Продемонстровано її застосування у лабораторних експериментах,

які стосувалися електрофізіології, мікроскопії та флюїдики. Розроблені авторами [3] рішення використовують одноплатні комп'ютери Raspberry Pi із спеціалізованим програмним забезпеченням, що спрощує впровадження на нових лабораторних пристроях. Ця система IP дозволяє вченим контролювати та відстежувати експерименти в режимі реального часу за допомогою онлайн веб-інструменту, автоматизувати дослідницькі завдання та отримувати поточні оновлення щодо статусу експериментів. Ця архітектура розширює дослідницькі можливості та може бути промасштабована на інші види лабораторій.

Прототип віддаленої лабораторії, представлений у праці [4], дозволяє студентам дистанційно керувати промисловим пілотним заводом. Ця функціональність досягається за допомогою трьох ключових компонентів: програмного клієнта викладача, програмного клієнта студента та центрального брокера. Обидва програмні клієнти реалізовані у Node-RED і спілкуються через протокол MQTT (Message Queuing Telemetry Transport). Програмний клієнт інструктора має прямий доступ до заводу та відповідає за керування учнями та системами. Студентські програмні клієнти, які можуть бути встановлені в лабораторії або на персональних комп'ютерах студентів, служать інтерфейсом для виконання студентами практичних завдань.

Апаратне забезпечення системи, яка розглядається у даній роботі, представлене спектрометром VIS-50 та світлодіодним джерелом випромінювання SL1-LED від компанії StellarNet (США), а також міні-ПК Synco PC Box (рис. 1). Світлодіодний випромінювач працює від блоку живлення напругою 5 В. Світло надходить через оптоволоконний кабель на вхід спектрометра. Спектрометр, в свою чергу, передає зчитаний спектр через порт USB 2.0 до міні-ПК, де встановлені необхідні драйвери та програма для зчитування та передачі даних зі спектрометра у хмарне середовище.

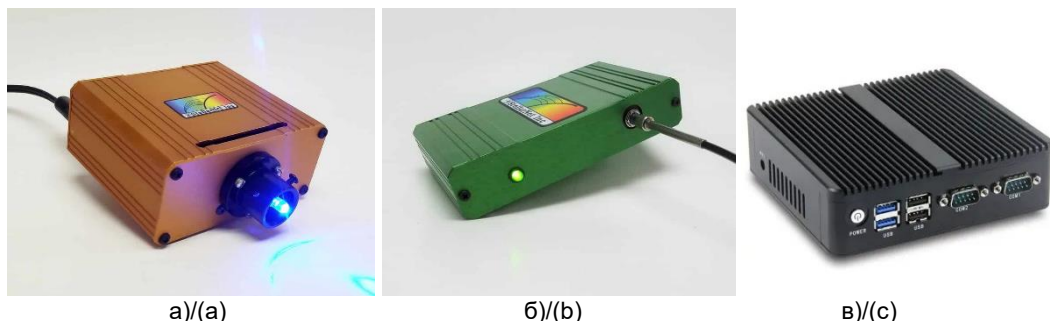


Рис. 1. Апаратне забезпечення системи: а) світлодіодне джерело StellarNet SL1-LED, б) спектрометр StellarNet VIS-50 та в) міні-ПК Synco PC Box.

Fig. 1. Experimental hardware: (a) LED radiation source StellarNet SL1-LED, (b) spectrometer StellarNet VIS-50, (c) mini-PC Synco PC Box.

Спектрометри StellarNet GREEN-Wave — це оптоволоконні прилади з широкою гамою стандартних моделей для вимірювань у діапазоні від ультрафіолету (УФ), через видиме світло і до інфрачервоного (ІЧ) випромінювання, - довжини хвиль 350-1150 нм. Різні моделі пропонують вибір діапазону дифракційної ґратки та роздільної здатності щілин. Нова вдосконалена електроніка з високошвидкісним 16-бітним оцифровувачем забезпечує швидкий збір даних і співвідношення сигнал/шум 400:1. Знімний блок спектрографа та електроніка керування розташовані всередині міцного металевого корпусу. Підтримується послідовне підключення і одночасна робота до 8 пристроїв. Спектрометри GREEN-Wave можна жити безпосередньо від

комп'ютерного порту USB 2.0. Одножильний волоконно-оптичний кабель або зонд забезпечує введення випромінювання через стандартний волоконно-оптичний роз'єм SMA 905 із можливістю вибору довжини кабелю. Оптика спектрометра є міцною та має вібростійку модульну конструкцію без рухомих частин. Передбачена пам'ять із попередньо встановленими калібруваннями та налаштуваннями спектрометра, а також пам'ять моментальних знімків для забезпечення миттєвого спектрального зображення з високочутливої ПЗЗ-лінійки із 2048 елементів.

StellarNet має різноманітні світлодіодні джерела УФ та видимого випромінювання, доступні для застосування в спектроскопії. Використана модель SL1-LED включає носовий конус із набором світлодіодів, що складається з 6 кольорових та одного білого світлодіода. Світлодіоди можна міняти місцями без змін проводки. Кожен світлодіод можна використовувати для збудження флуоресценції у різних експериментах. Серед переваг - тривалий термін служби світлодіодів, гнучкість (для заміни світлодіодів потрібно кілька секунд) та невеликий розмір, що ідеально підходить для компактних спектрометрів.

Спеціалізований системний блок Synco PC Box розроблений для інтенсивного використання у складних умовах. Блок захищений від пилу, вологи та функціонує в широкому температурному діапазоні.

Для підключення та налаштування спектрометра використовується програмний пакет StellarPro, який містить також необхідні драйвери пристрою і має інтерфейс для демонстрації даних в режимі реального часу. Для зчитування сигналу використано Python SDK від StellarNet. Для з'єднання з хмарним середовищем та передачі даних використано AWS IoT Python SDK.

StellarPro для Windows/Mac/Linux представляє собою поєднання програмного забезпечення SpectraWiz від StellarNet із розширеними інструментами та інтерфейсом прикладного модуля. Вбудовані модулі можуть використовуватися для радіометрії, колориметрії, хімічної абсорбції/кінетики, хеометрії, спектральної ідентифікації та спектроскопії комбінаційного розсіювання. StellarPro також включає інтегровану попередню спектральну обробку, таку як віднімання базової лінії, видалення спайків і перегляд похідних. Спектральне узгодження, розробка моделі PLS Calibration, алгоритм класифікації та розгортання під час виконання також включені в пакет. StellarPro має налаштування дозволів користувача, входу та звітності для експорту всіх даних в одному місці. Також включені додаткові програми для дослідження тонких плівок, мутних середовищ, тощо.

StellarNet надає універсальний набір програмного забезпечення Python SDK для взаємодії з лінійкою спектрометрів, що забезпечує інтеграцію та контроль для різноманітних спектроскопічних програм. Цей SDK підтримує збір даних у реальному часі та забезпечує функції для калібрування, спектрального аналізу та візуалізації даних. SDK містить вичерпну документацію та приклади сценаріїв, щоб полегшити швидку розробку та розгортання спеціальних спектроскопічних рішень. Можливість інтеграції з іншими бібліотеками та інструментами Python додатково розширює можливості спектрометрів StellarNet.

Для інтеграції з хмарною платформою використано AWS IoT Core Python SDK, що надає надійний набір інструментів для підключення, керування та взаємодії з пристроями Інтернету Речей через Amazon Web Services. Цей SDK спрощує інтеграцію пристроїв IoT із службами AWS, забезпечуючи збір, обробку та аналіз даних у реальному часі. Завдяки підтримці протоколів MQTT, HTTP та WebSockets, SDK забезпечує надійний і безпечний зв'язок між пристроями та ядром AWS IoT. SDK пропонує функції для публікації повідомлень і підписки, а також безпечного зв'язку за допомогою TLS/SSL. Ця гнучкість дозволяє розробникам створювати масштабовані та ефективні додатки IoT, використовуючи повний спектр сервісів AWS, таких як AWS Lambda, AWS S3 і Amazon DynamoDB.

РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

Високорівневу архітектуру автоматизованого рішення представлено на рис. 2.

Було розроблено Python скрипт, що інтегрує спектрометр StellarNet із AWS IoT Core, дозволяє отримувати спектрографічні дані в реальному часі та публікувати їх у хмарі. Скрипт призначений для калібрування спектрометра, постійного збору даних спектру та надсилання цих даних до AWS IoT Core для подальшої обробки або аналізу.

Скрипт починається з налаштування необхідних параметрів для підключення до AWS IoT Core. Це включає вказування посилання, ідентифікатора клієнта та шляхів до сертифікатів SSL/TLS, необхідних для безпечного зв'язку. Використовуючи метод `'mqtt_connection_builder.mtls_from_path'` з AWS IoT Core Python SDK, сценарій ініціалізує клієнт MQTT. Потім встановлюється з'єднання з AWS IoT Core, що дозволяє сценарію публікувати дані у вказаному MQTT-топіку. З'єднання MQTT через Transport Layer Security (TLS) забезпечує безпечний і надійний зв'язок між локальним спектрометром і хмарним середовищем AWS.

Дані сценарій ініціалізує спектрометр StellarNet за допомогою бібліотеки `'stellarnet_driver3'`. Спектрометр ініціалізується викликом функції `'sn.array_get_spec(0)'`, яка отримує об'єкт спектрометра та пов'язані дані про довжину хвилі. Після ініціалізації сценарій переходить до калібрування спектрометра. Функція `'calibrate_spectrometer'` встановлює параметри, такі як час інтеграції, кількість сканувань для усереднення, коефіцієнт згладжування та стан синхронізації зовнішнього тригера. Ці параметри мають вирішальне значення для забезпечення точного та узгодженого збору даних про спектр.

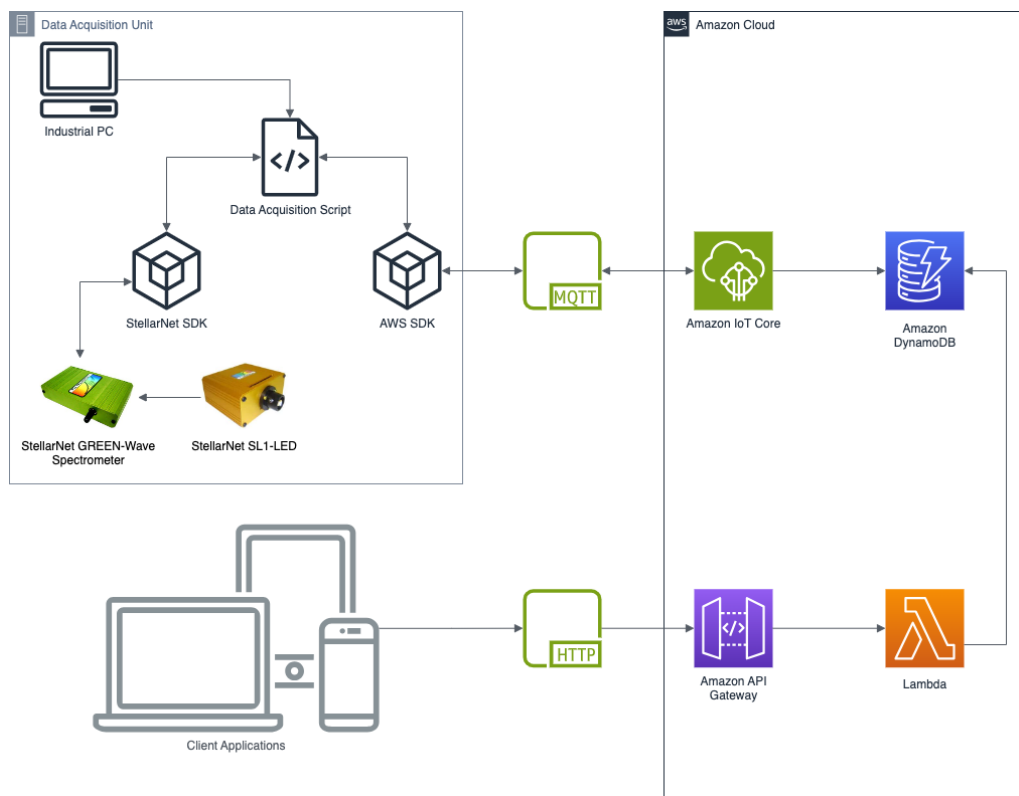


Рис. 2. Архітектура хмарної лабораторії
Fig. 2. Cloud Lab Architecture

Після калібрування сценарій входить у безперервний цикл, де він отримує спектральні дані в реальному часі зі спектрометра. Функція `get_spectrum_data` використовується для отримання спектральних даних - довжини хвилі та відповідних значень інтенсивності. Ця функція викликає метод `sn.array_spectrum(spectrometer, wav)` для отримання поточних даних спектру. Далі дані форматуються у JavaScript Object Notation (JSON) сумісну структуру, включаючи позначку часу (timestamp), що вказує, коли дані були зібрані.

Після отримання спектральних даних вони надсилаються до AWS IoT Core за допомогою функції `publish_to_aws_iot`. Ця функція перетворює дані в JSON і публікує їх у вказаній темі на AWS IoT Core за допомогою методу `mqtt_connection.publish`. Публікуючи дані в режимі реального часу, скрипт дозволяє дистанційно контролювати та аналізувати покази спектрометра. Це налаштування особливо корисно для програм, які вимагають безперервного збору даних і опрацювання в хмарі. Окрім того, скрипт дозволяє підписатись на той же топик AWS IoT Core, що дозволяє перевіряти надіслані дані і відстежувати помилки.

Скрипт починається з калібрування спектрометра, а потім входить у нескінченний цикл, де він отримує та публікує дані спектру кожні 5 секунд (на практиці для конкретних сценаріїв цю затримку можна збільшувати чи зменшувати). Цикл може бути перерваний користувачем (перериванням виконання клавішами Ctrl+C в терміналі), у цей момент сценарій гарантує, що спектрометр належним чином скинуто, а з'єднання MQTT правильно закрито. Це гарантує, що спектрометр залишається в «чистому» стані та те, що немає тривалих підключень до AWS IoT Core, які не використовуються.

Таким чином, скрипт надає комплексне рішення для інтеграції спектрометра StellarNet з AWS IoT Core. Він охоплює ініціалізацію, калібрування, збір даних у реальному часі та публікацію в хмарі. Використовуючи можливості StellarNet SDK та AWS IoT Core SDK, сценарій забезпечує плавний і ефективний моніторинг даних спектрометра, що дозволяє автоматизувати експерименти у лабораторії з підключеними пристроями та спостерігати за результатами у реальному часі. Приклад роботи скрипта показано на рис. 3.

```
Spectrometer connected!
set_ext_trig True
setParam inttime=50 scansavg=1 smooth=0 xtiming=3
setTempComp True
Connection Successful with return code: 0 session present: False
Connected to AWS!
Subscribing to topic 'spectrum'...
Subscribed with QoS.AT_LEAST_ONCE
Received message from topic 'spectrum': b'{"timestamp": 1721052925, "wavelength": [339, 356, 374, 391, 409, 428, 446, 464, 483, 501, 520, 539, 559, 578, 597, 617, 637, 657, 677, 697, 717, 738, 759, 780, 801, 822, 843, 865, 886, 908, 930, 952, 974, 997, 1019, 1042, 1065, 1088, 1111, 1134, 1158], "intensity": [827, -4, -4, 9, -6, 2, -14, 67, -36, -1, 22, -4, -6, -16, 3, -16, -17, 9, -11, -14, -16, -8, 1, -16, -4, -3, -22, 7, 18, 36, 32, 8, 8, -70, -6, -3, 7, -16, -20, -6, 16]}'
Received message from topic 'spectrum': b'{"timestamp": 1721052930, "wavelength": [339, 356, 374, 391, 409, 428, 446, 464, 483, 501, 520, 539, 559, 578, 597, 617, 637, 657, 677, 697, 717, 738, 759, 780, 801, 822, 843, 865, 886, 908, 930, 952, 974, 997, 1019, 1042, 1065, 1088, 1111, 1134, 1158], "intensity": [853, -4, 8, 12, 20, 15, 13, -40, 28, -36, -23, 12, 12, -16, -11, 37, 2, -15, -23, -6, 19, -8, -36, 17, -5, 7, -24, -26, -19, -12, -5, 52, -33, 9, 25, 17, 30, 54, 45, -17, -11]}'
Received message from topic 'spectrum': b'{"timestamp": 1721052936, "wavelength": [339, 356, 374, 391, 409, 428, 446, 464, 483, 501, 520, 539, 559, 578, 597, 617, 637, 657, 677, 697, 717, 738, 759, 780, 801, 822, 843, 865, 886, 908, 930, 952, 974, 997, 1019, 1042, 1065, 1088, 1111, 1134, 1158], "intensity": [825, 40, 20, 4, 42, 2, 77, 607, 11295, 54686, 16874, 3848, 853, 261, 60, 49, 33, 49, 9, 6, 45, -27, -2, 13, -21, 0, 13, 67, -4, 0, 42, 12, 17, -10, 26, 52, 48, 18, -23, -31, 16]}'
Received message from topic 'spectrum': b'{"timestamp": 1721052941, "wavelength": [339, 356, 374, 391, 409, 428, 446, 464, 483, 501, 520, 539, 559, 578, 597, 617, 637, 657, 677, 697, 717, 738, 759, 780, 801, 822, 843, 865, 886, 908, 930, 952, 974, 997, 1019, 1042, 1065, 1088, 1111, 1134, 1158], "intensity": [845, 16, 40, 10, 45, 79, 60, 586, 11214, 54250, 17151, 3947, 901, 212, 92, 12, -23, 4, 25, -23, 49, -5, 64, 12, -15, -39, -3, 28, -34, -54, 18, 11, 41, 5, 61, 45, 33, 22, 11, -8, -10]}'
Disconnecting from AWS...
Connection closed
Disconnected!
Releasing spectrometer...
Released!
```

Рис. 3. Приклад роботи скрипта з вимкненим та увімкненим LED джерелом
Fig. 3. Example of script operation with LED source turned off and on

Розглянемо реалізацію рішення зі сторони хмарного середовища. Для передачі даних використовується протокол MQTT, який є легким протоколом обміну повідомленнями, розробленим для малопотужних пристроїв та ненадійних мереж. Він широко використовується в системах IP для забезпечення ефективного, простого та надійного зв'язку між пристроями. Кожен підключений пристрій може використовувати одну чи декілька тем (topics) для публікування/підписки на повідомлення.

Дані передаються напряму в AWS IoT Core. Це керована хмарна служба від Amazon Web Services, яка дозволяє підключеним пристроям легко та безпечно взаємодіяти з хмарними додатками та іншими пристроями. Вона забезпечує інфраструктуру, необхідну для побудови додатків IP, які збирають, обробляють, аналізують і реагують на дані, згенеровані підключеними пристроями, без необхідності управління базовою інфраструктурою.

У межах сервісу створено Річ (Thing) під назвою "stellarnet". Річ – це представлення та запис фізичного пристрою в хмарному середовищі. Фізичному пристрою потрібен запис про річ, щоб працювати з AWS IoT. Типовий варіант використання пристрою передбачає використання назви речі як ідентифікатора клієнта MQTT за замовчуванням. У роботі використано той самий Client ID "stellarnet" у скрипті, який відправляє дані в хмару.

Рис. 4 показує відображення даних про статус під'єднання та авторизації спектрометра у хмарному середовищі, а також статистику щодо опублікованих повідомлень по темі.

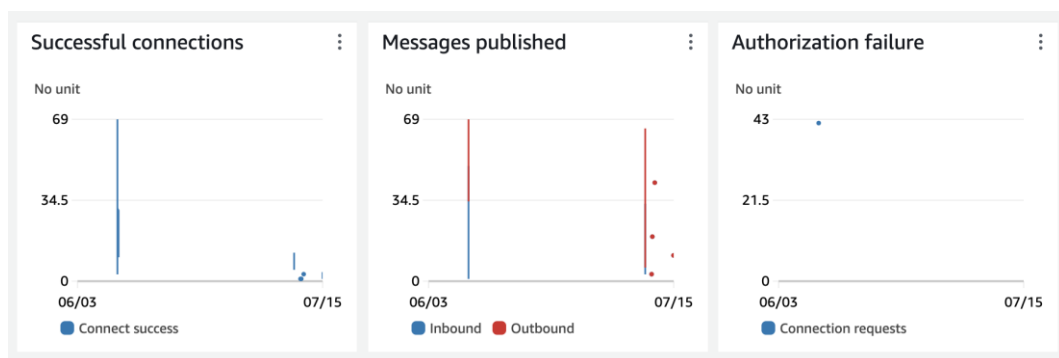


Рис. 4. Приладова панель отриманих та надісланих повідомлень в AWS IoT Core
Fig. 4. Dashboard of received and sent messages in AWS IoT Core

Також в межах AWS IoT Core використовуються правила (Rules). Правило - це компонент AWS IoT Core, який дозволяє автоматично обробляти повідомлення, що надходять від під'єднаних пристроїв. Використовуючи правила, ми можемо фільтрувати, перетворювати і маршрутизувати ці повідомлення до інших сервісів AWS або виконувати інші дії на їхній основі. Зокрема, створено правило під назвою "save_to_db", яке підписується на топик "spectrum", що використовується для публікації даних зі скрипта, описаного вище. Правило вибирає усі дані з повідомлення за допомогою SQL-виразу та передає через конектор до бази даних на AWS DynamoDB.

Для зберігання оброблених даних спектрометра обрано сервіс Amazon DynamoDB завдяки його високій продуктивності та масштабованості. DynamoDB — це повністю керована служба бази даних NoSQL, яка забезпечує гнучке зберігання даних із низькою часовою затримкою. Створено таблицю з атрибутами 'timestamp', 'wavelength' та 'intensity' (кожен наступний пристрій, відповідно, матиме свою таблицю з іншими полями). Завдяки використанню можливостей автоматичного

масштабування DynamoDB рішення може обробляти великий обсяг вхідних даних без ручного втручання, гарантуючи, що рівень зберігання залишається надійним навіть за великих навантажень. Типові записи у базі показано на рис. 5.

timestamp (Number) ▾	intensity ▾	wavelength ▾
1720739443	[{"N": "998"}, {"N": "-10"}, {"N": "-1...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720777172	[{"N": "876"}, {"N": "31"}, {"N": "10" ...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720777618	[{"N": "871"}, {"N": "9"}, {"N": "-8"}, ...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720777212	[{"N": "897"}, {"N": "4"}, {"N": "3"}, {...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720731060	[{"N": "997"}, {"N": "11"}, {"N": "34" ...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1721052936	[{"N": "825"}, {"N": "40"}, {"N": "20" ...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720777608	[{"N": "888"}, {"N": "24"}, {"N": "-13"...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720777587	[{"N": "885"}, {"N": "22"}, {"N": "-5" ...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720739446	[{"N": "1030"}, {"N": "-2"}, {"N": "-2"...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720777536	[{"N": "899"}, {"N": "34"}, {"N": "25" ...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720739459	[{"N": "1039"}, {"N": "46"}, {"N": "28"...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720739458	[{"N": "1040"}, {"N": "-19"}, {"N": "1"...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...
1720739453	[{"N": "1038"}, {"N": "19"}, {"N": "-3"...	[{"N": "339"}, {"N": "356"}, {"N": "374"}, {"N": "...

Рис. 5. Таблиця з даними спектрометра в AWS DynamoDB
Fig. 5. Spectrometer data table in AWS DynamoDB

Дані, які успішно збережено в базу, мають візуалізуватися. Amazon має кілька сервісів для візуалізації, наприклад QuickSite або SiteWise. AWS QuickSight — це хмарна служба бізнес-аналітики, яка дозволяє не лише візуалізувати дані, а й отримувати аналітичні інсайти. AWS IoT SiteWise — це керована хмарна служба AWS, яка допомагає збирати, організовувати та аналізувати дані з виробничого обладнання. Обидві служби дають можливість створювати дашборди з графіками, але, на жаль, не пристосовані для відображення даних в реальному часі, оскільки спрямовані на роботу з аналітикою та агрегацію даних за певний період. Тому було вирішено розробити окремий додаток для візуалізації даних з DynamoDB.

Для доступу до даних використано сервіс AWS API Gateway. Це керована служба, яка дозволяє розробникам створювати, публікувати, підтримувати, моніторити та захищати API (Application Programming Interfaces) будь-якого масштабу. Сервіс також дозволяє створювати та керувати RESTful або WebSocket API для застосунків, що взаємодіють із бекенд-сервісами. Було створено REST API під назвою "spectrum" з ендпоинтом GET /latest. Згенеровано URL адресу, за допомогою якої можна звертатись до цього ендпоинта. Ендпоинт GET /latest, у свою чергу, прив'язаний до інтеграції з AWS Lambda function "spectrum".

AWS Lambda - це serverless обчислювальна служба, яка дозволяє запускати код без необхідності створення сервера на віртуальній машині. AWS Lambda виконує код тільки тоді, коли це потрібно, і масштабується автоматично від декількох запитів на день до тисяч на секунду. Було створено функцію "spectrum", яка перевіряє вхідний routeKey і обробляє GET /latest запит з API Gateway. Функція отримує дані з бази, вибирає останній елемент і перетворює дані з формату DynamoDB у JSON, який відправляється у відповідь на запит (рис. 6).


```

import json
import boto3
from decimal import Decimal

client = boto3.client('dynamodb')
dynamodb = boto3.resource("dynamodb")
table = dynamodb.Table('spectrum')
tableName = 'spectrum'

def lambda_handler(event, context):
    print(event)
    body = {}
    statusCode = 200
    headers = { "Content-Type": "application/json" }
    try:
        if event['routeKey'] == "GET /latest":
            latest = table.scan()["Items"][-1]
            intensity = []
            for i in latest['intensity']:
                intensity.append(int(i))
            wavelength = []
            for i in latest['wavelength']:
                wavelength.append(int(i))
            body = {
                'timestamp': int(latest['timestamp']),
                'intensity': intensity,
                'wavelength': wavelength
            }
        except KeyError:
            statusCode = 400
            body = 'Unsupported route: ' + event['routeKey']
    body = json.dumps(body)

```

Рис. 6. AWS Lambda функція для зв'язування API Gateway та DynamoDB
Fig. 6. AWS Lambda function for linking API Gateway and DynamoDB

Для реалізації додатку візуалізації даних було використано Flutter. Це відкритий фреймворк для створення кросплатформних мобільних, веб та десктопних додатків, розроблений компанією Google. Він дозволяє розробникам створювати додатки для iOS, Android, веб та настільних платформ, використовуючи єдину базу коду. Flutter використовує власний графічний рушій Skia для візуалізації, що забезпечує високу продуктивність та швидкість роботи додатків. А завдяки функції "Hot Reload" розробники можуть миттєво бачити зміни у своєму коді без необхідності перезавантажувати додаток, що значно прискорює процес розробки.

У рамках роботи розроблено додаток "spectroscopy_app" (рис. 7), що отримує дані з AWS API Gateway, описаного вище, і візуалізує їх за допомогою бібліотеки FL Chart (https://pub.dev/packages/fl_chart). FL Chart — це бібліотека для побудови графіків та діаграм для Flutter із широкими можливостями налаштування, яка підтримує лінійну діаграму, гістограму, секторну, точкову та радарну діаграми. Застосунок надсилає GET /latest запит на Web API кожні 5 секунд, після чого оновлює отримані дані на графіку.

В подальшому планується також розробка графіка залежності домінантної частоти від часу. Сюди ж будуть додаватись графіки та приладові панелі (dashboards) з інших підключених пристроїв IP.

ВИСНОВКИ

Автоматизація експериментів з використанням описаного хмарного рішення не тільки підвищує ефективність, але й забезпечує вищу точність і відтворюваність у наукових дослідженнях. Зводячи до мінімуму ручне втручання та використовуючи обробку даних у реальному часі та розширену аналітику, дослідники можуть зосередитися на розробці експериментів та інтерпретації результатів, зрештою прискорюючи темп відкриттів та інновацій.

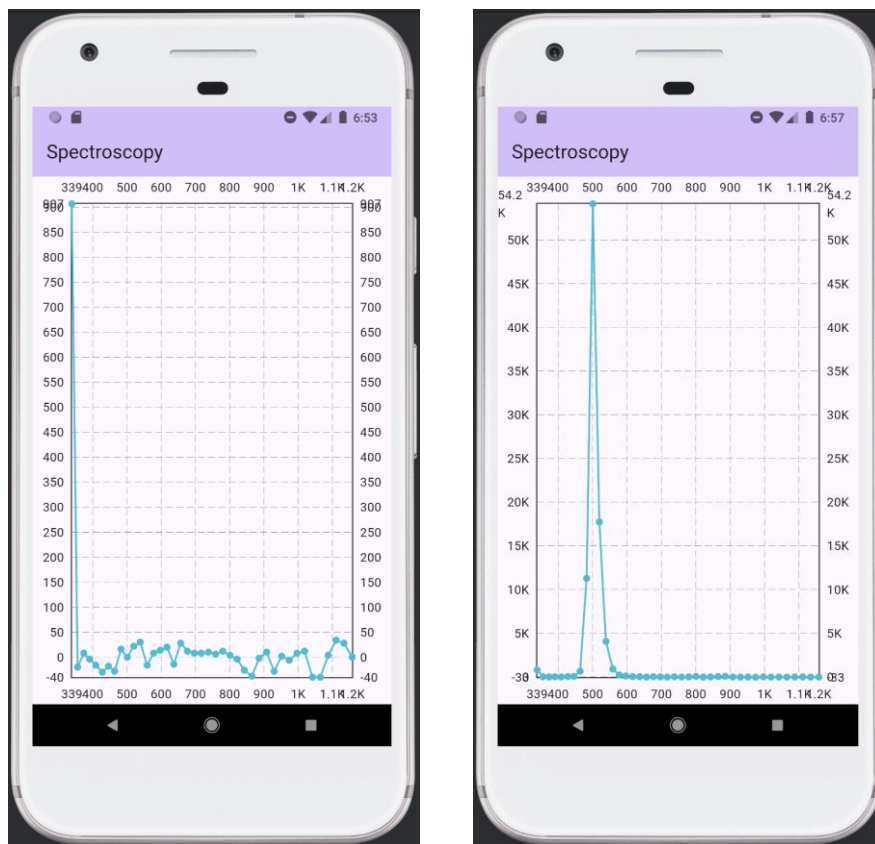


Рис. 7. Візуалізація спектра в мобільному додатку з вимкнутим та увімкнутим LED-джерелом.
Fig. 7. Spectrum visualization in a mobile application with the LED source turned off and on.

Запропоноване інтегроване рішення ефективно використовує потужність сервісів AWS для підвищення функціональності та зручності використання спектрометрів StellarNet. З використанням AWS IoT Core, AWS Lambda, Amazon DynamoDB і Amazon API Gateway створено надійну, масштабовану та ефективну систему для збору, обробки та зберігання даних. Безперебійна взаємодія між цими службами гарантує, що дані спектрометра не тільки збираються та безпечно зберігаються, але й легко доступні для аналізу та прийняття рішень. Додатково розроблений користувацький крос-платформний додаток на Flutter дозволяє переглядати результати експерименту в реальному часі та агрегувати їх за потреби.

Загалом, рішення демонструє потенціал хмарних сервісів у вдосконаленні традиційного лабораторного обладнання. Інтегруючи сучасні хмарні технології з науковими інструментами, дослідники та вчені можуть досягти більшої ефективності, масштабованості та розуміння даних. Цей підхід не тільки оптимізує використання наявних ресурсів, але й прокладає шлях до більш досконалого комп'ютерного опрацювання даних і робочих процесів аналізу в майбутньому.

ВНЕСОК АВТОРІВ

Концептуалізація, [А.К., І.К.]; методологія, [А.К., І.К.]; валідація, [І.К.]; формальний аналіз, [А.К., І.К.]; дослідження, [А.К., І.К.]; написання – підготовка оригінального тексту, [А.К.]; написання – рецензування та редагування, [І.К.]; візуалізація, [А.К.]; нагляд, [І.К.]; адміністративне управління проектом, [І.К.].

Усі автори ознайомилися з остаточною версією рукопису та погодилися з його публікацією.

ДЖЕРЕЛА

- [1] Azad, A. Design and Development of Remote Laboratories with Internet of Things Setting. *Advances in Internet of Things*, 11, 95-112 (2021). DOI: 10.4236/ait.2021.113007.
 - [2] Hairuddin, M., Ashar, N., Abidin, A., Tahir, N. Cost-Effective Interfaces with Arduino-LabVIEW for an IOT-Based Remote Monitoring Application. *LabVIEW - A Flexible Environment for Modeling and Daily Laboratory Use* (2021). doi: 10.5772/intechopen.97784
 - [3] D.F. Parks, et al. IoT cloud laboratory: Internet of Things architecture for cellular biology. *Internet of Things*, 20 (2022), Article 100618. DOI: 10.1016/j.iot.2022.100618
 - [4] M. Domínguez, R. González-Herbón, J.R. Rodríguez-Ossorio, J.J. Fuertes, M.A. Prada and A Morán. Development of a remote industrial laboratory for automatic control based on Node-RED. *IFAC-PapersOnLine* (2020), vol. 53, no. 2, pp. 17210-17215. DOI: 10.1016/j.ifacol.2020.12.1741
-

COMPUTERIZED OPTICAL EXPERIMENTS WITH PORTABLE SPECTROMETERS AND AMAZON WEB SERVICES CLOUD INTEGRATION

Andriy Krupych[✉], **Ivan Karbovnyk**[✉]
Ivan Franko National University of Lviv
Drahomanova Str., 50, 79005 Lviv, Ukraine

ABSTRACT

Background. This paper presents a system for automating optical experiments using the StellarNet spectrometer and AWS cloud services. The relevance of this research is driven by the need for remote access to laboratory equipment, which became particularly critical during the COVID-19 pandemic. The proposed solution enables real-time optical experiments, automates spectral data collection and processing, and ensures secure cloud storage for further analysis. By leveraging cloud technologies, the system enhances research efficiency and accessibility, reducing dependence on physical presence in the laboratory.

Materials and Methods. The study reviews modern approaches to laboratory automation, focusing on both hardware and software solutions. The integration of embedded processors like Raspberry Pi, development environments such as LabVIEW, and IoT-based architectures is considered. A key aspect is the implementation of remote access, allowing seamless operation and real-time monitoring. The system hardware includes the StellarNet VIS-50 spectrometer, the SL1-LED light source, and the Synco PC Box mini-PC. Software tools, such as the StellarNet Python SDK and AWS IoT Core, are used to enable automated data processing and cloud communication.

Results and Discussion. A Python script was developed for integrating the spectrometer with AWS IoT Core, enabling seamless data transmission. The paper describes the setup, calibration, and real-time spectral data collection process. Using the MQTT protocol, experimental data is securely transmitted to the cloud, allowing remote access and analysis. The implementation details, including secure connection protocols,

message publication, and cloud integration, are discussed to highlight system reliability and efficiency.

Conclusions. The proposed system enhances laboratory experiment automation, improving research flexibility and efficiency in STEM disciplines. It reduces equipment maintenance costs and increases accessibility to experimental data. The scalable solution can be adapted to various scientific applications requiring remote equipment access and continuous data collection.

Keywords: Automation, Data Processing, Internet of Things, Spectroscopy, AWS, Cloud Computing