

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Київський національний університет будівництва і архітектури

## **ОСНОВИ ПРОМИСЛОВОГО ІНТЕРНЕТУ РЕЧЕЙ**

Методичні вказівки

до виконання лабораторних та практичних робіт  
для здобувачів першого (бакалаврського) рівня вищої освіти  
спеціальності 174 «Автоматизація, комп'ютерно-інтегровані  
технології та робототехніка» всіх форм навчання

Київ 2024

УДК 681.513.674

О-75

Укладач М. І. Самойленко, асистент

Рецензент Б. О. Трощинський, канд. техн. наук, доцент

Відповідальний за випуск А. В. Запривода, канд. техн. наук,  
доцент

*Затверджено на засіданні кафедри автоматизації  
технологічних процесів, протокол № 6 від 25 вересня 2024 року.*

В авторській редакції.

**Основи** промислового інтернету речей [Електронний ресурс] :  
О-75 методичні вказівки / уклад. Самойленко М. І. – Київ : КНУБА,  
2024. – 16 с.

Містять зміст, порядок виконання лабораторних робіт.

Призначено для здобувачів першого (бакалаврського) рівня  
вищої освіти спеціальності 174 «Автоматизація, комп'ютерно-  
інтегровані технології та робототехніка» всіх форм навчання.

© КНУБА, 2024

## **Загальні положення**

Інтернет речей або IoT (Internet of Things) – система фізичних об'єктів («речей»), взаємопов'язаних між собою за допомогою вбудованих датчиків, програмного забезпечення та/або інших технологій. Мета IoT полягає в тому, щоби передавати дані на інші пристрої в системі або в інші системи через інтернет. Простіше кажучи, фізичні об'єкти виходять в інтернет, щоби відправити інформацію чи прийняти її.

Промисловий Інтернет Речей – це система об'єднаних комп'ютерних мереж і підключених до них промислових (виробничих) об'єктів з вбудованими датчиками і програмним забезпеченням для збору та обміну даними, з можливістю віддаленого контролю й управління в автоматизованому режимі, без участі людини.

Нині інтернет речей та промисловий інтернет речей стрімко розвиваються, охоплюючи собою всі сфери життєдіяльності людини.

Метою вивчення дисципліни є набуття достатнього рівня компетенції у галузі розробки систем Інтернету речей, зокрема, з погляду використання засобів інженерії програмного забезпечення для розробки систем різного призначення.

## **Короткі теоретичні відомості**

Для виконання лабораторних та практичних робіт використовується онлайн платформа Adafruit з хмарним MQTT брокером та пакет «потокowego програмування» Node-red.

Інтернет платформа Adafruit надає сервіси для навчання, розробки та тестування пристроїв інтернету речей. Є безкоштовні та платні сервіси з аналогічним функціоналом, різниця лише в кількості пристроїв/публікацій, що підтримується.

Для використання сервісів достатньо зареєструватись на сайті Adafruit: заповнити поля «ім'я користувача», «прізвище», «електронна пошта», «пароль» та обрати «план». Реєстрація безкоштовна, «безкоштовний план» цілком достатній для навчальних проєктів.

Після реєстрації заходимо у власний акаунт, обираємо вкладку ІО, за бажанням сервіс Google перекладає сторінку українською без втрати функціоналу (рис. 1).

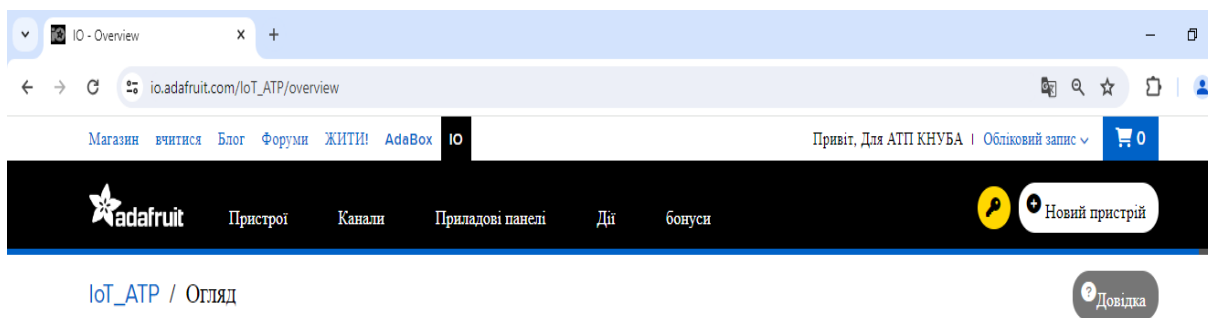


Рис. 1. Вкладка ІО сервісу Adafruit

Натискаємо символ «ключ» (жовтий кружечок на рис. 1) та отримуємо ключі доступу до публікацій та підписки, ці ключі будуть використані під час створення пристроїв.

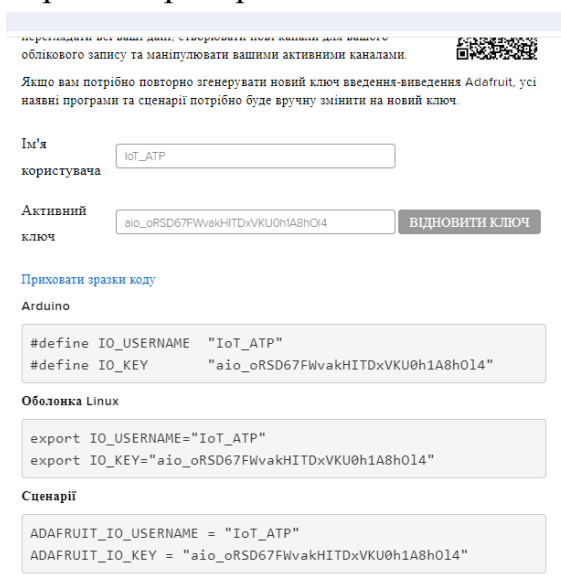


Рис. 2. Ключі доступу до публікацій даних та до підписки

### Створення пристрою інтернету речей

Пристрій можна створити на базі багатьох готових плат, що підтримують комунікації, частіше використовують плати на базі модулів ESP8266 чи ESP32. Плати можна програмувати в різних IDE: Eclipse, ESPHome, Arduino та інші, список доступних IDE розширюється. Платформа Adafruit передбачає також можливість створення пристрою інтернету речей **без написання коду**, скористаємось цим сервісом.

На вкладці сервісу Adafruit рис. 1 натискаємо «Новий пристрій», та виконуємо покрокові інструкції:

1. Обираємо «дошку» (так Adafruit назвав плату), на вибір є плати на базі модулів ESP32, ESP8266 різних модифікацій. Обираємо ту, що є у вас, наприклад Huzzah ESP8266 (на базі модуля ESP8266 є в лабораторії);
2. Сервіс Adafruit пропонує створити резервну копію прошивки, що була в модулі, можна пропустити;
3. Вказуємо назву Wi-Fi мережі SSID Wi-Fi та Пароль Wi-Fi;
4. Підключаємо плату до комп'ютера, вказуємо порт (рис. 3);

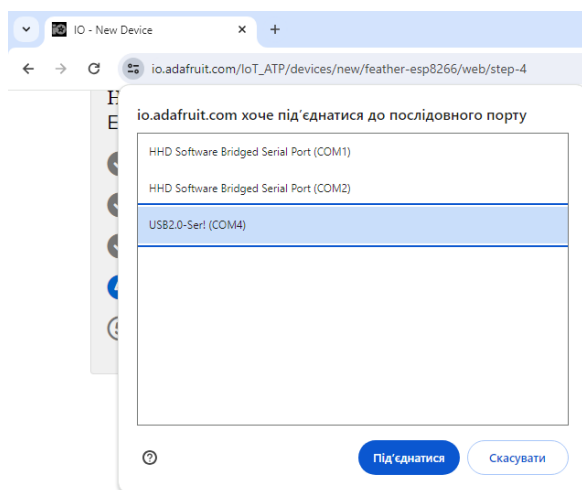


Рис. 3. Вибір порту для підключення плати та завантаження програми

5. Натискаємо «підключитись» та чекаємо поки завантажиться програма в нашу плату;
6. По закінченню загрузки натискаємо кнопку «Reset» на платі та чекаємо підключення пристрою до серверу Adafruit. Після вдалого підключення з'явиться повідомлення (рис. 4).

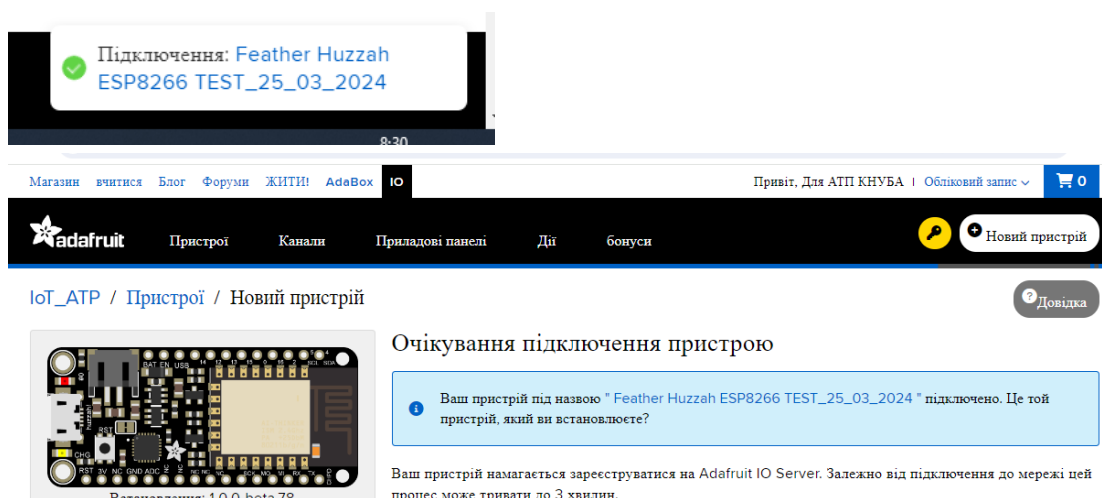


Рис. 4. Повідомлення про вдале підключення пристрою до Adafruit

7. Переходимо до нашого пристрою, подвійний клік мишкою по назві «Feather Huzzah ESP8266 TEST\_25\_03\_2024», у відкритому вікні натискаємо кнопку «новий компонент», в новому вінкю обираємо власне «компонент», наприклад DS18B20 рис. 5.

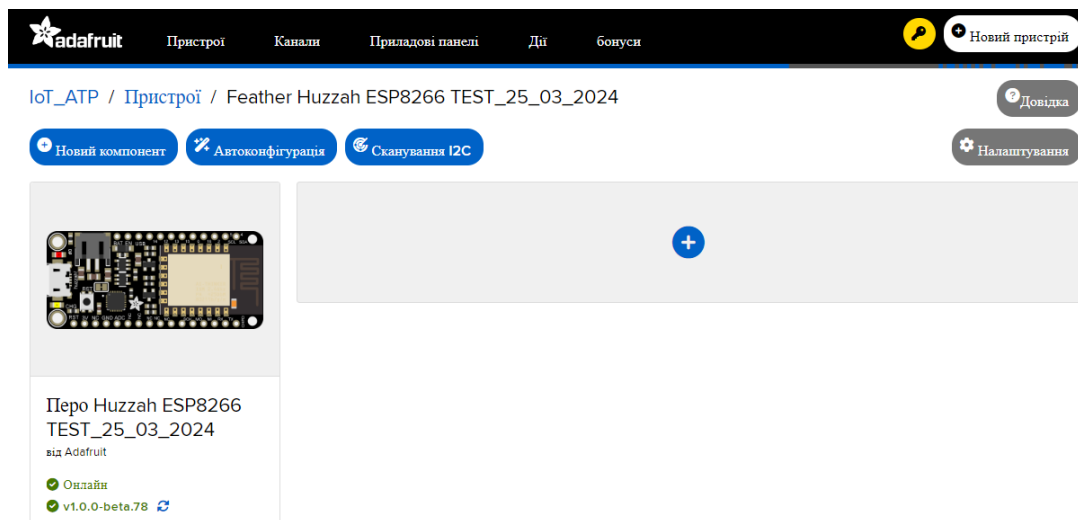


Рис. 5. Вікно конфігурації «Пристрою» та вікно вибору «компоненту»

8. У вікні створення компоненту (рис. 6) налаштовуємо DS18B20:

- обираємо пін, контакт esp8266 до якого буде підключений датчик температури ds18b20, наприклад 0;
- обираємо розподільчу здатність –12 біт;
- за бажанням можна вмикати чи вимикати передачу температури в градусах Цельсія чи Фаренгейта;
- слід обрати періодичність публікацій даних, наприклад 30 с (15 хв за замовчанням довго чекати оновлення);
- натискаємо «створити компонент», автоматично повертаємось на вкладку пристрою та бачимо новий компонент.

9. Переходимо на вкладку канали (рис 7) бачимо наш пристрій - «Перо Huzzah ESP8266 TEST\_25\_03\_2024» та датчик – «DS18B20: Датчик температури (° C)», переходимо до даних, що публікує пристрій (клік по датчику). Дані з датчика, відображаються у вигляді графіка та списком ( рис. 8). Для перевірки можна нагріти датчик рукою.

Аналогічним чином можна підключати інші компоненти зі списку.

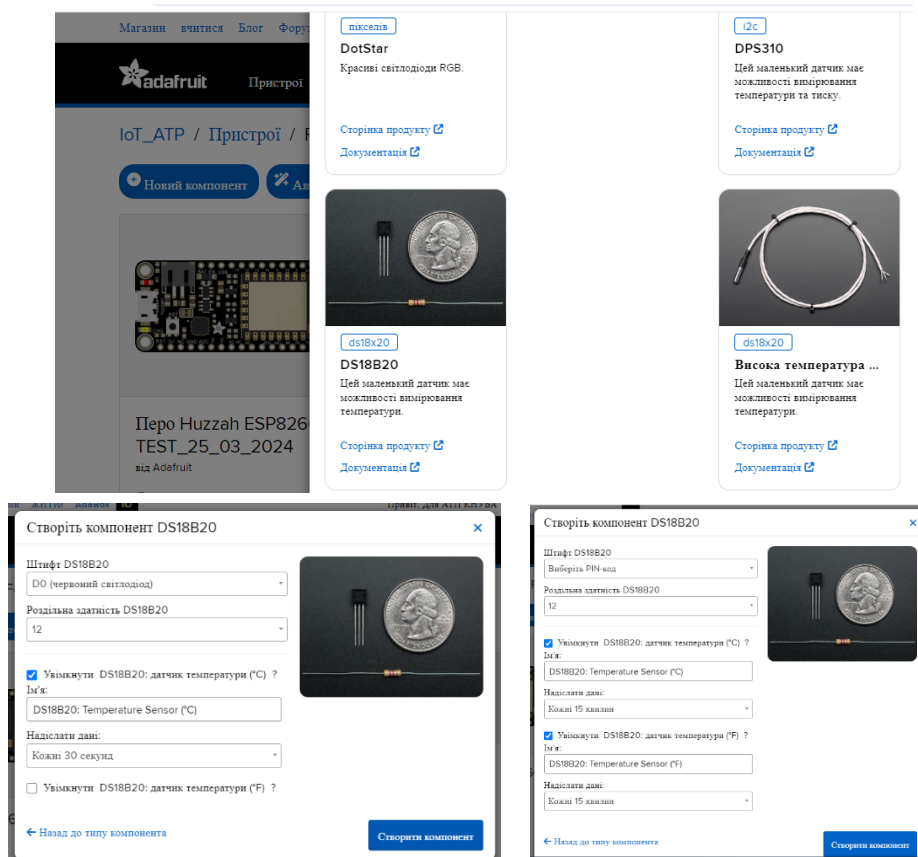


Рис. 6. Вікно створення компонента

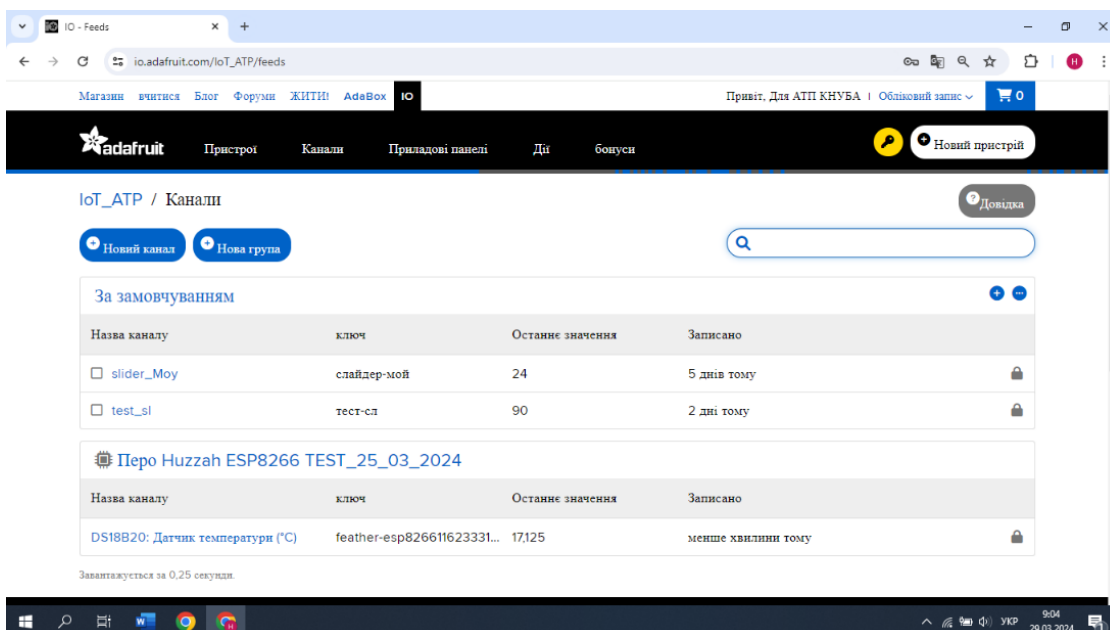


Рис. 7. Новий компонент DS18B20 у складі пристрою

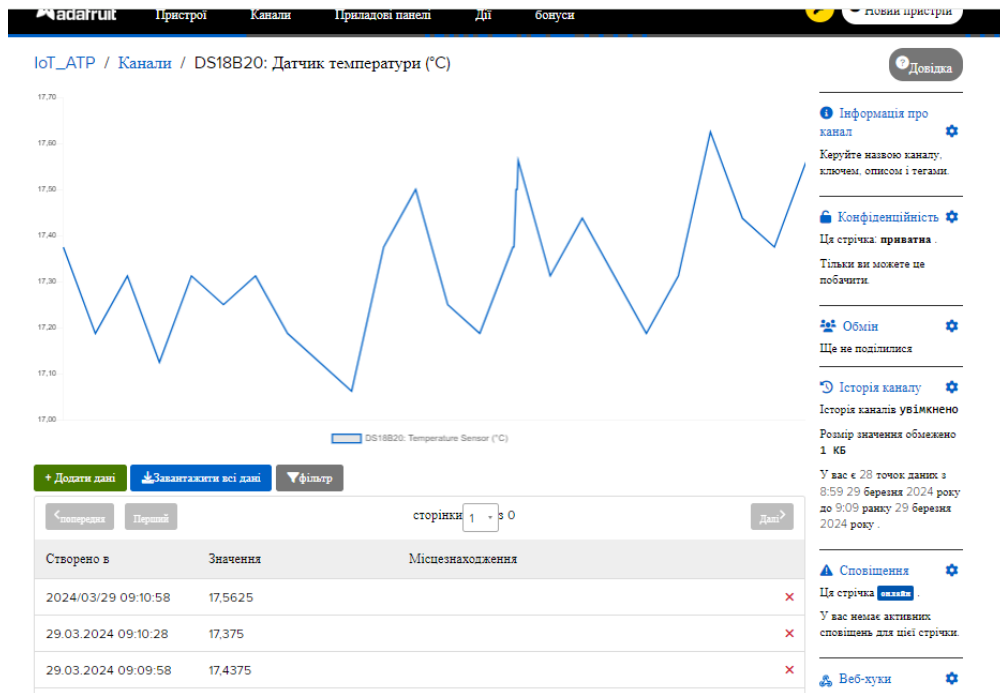


Рис. 8. Дані, що публікує пристрій

### Створення панелі приладів

Переходимо на вкладку «Приладові панелі» (Dashboard), де відображається список вже існуючих панелей, натискаємо «Нова інформаційна панель», вводимо назву панелі, за бажанням – короткий опис.

У списку панелей з'явився новий рядок з назвою щойно створеної панелі, відкриваємо панель та додаємо візуальні компоненти, налаштовуємо компоненти та панель:

- натискаємо кнопку редагування панелі, праворуч вгорі, з зображенням «шестерня», та обираємо «Створити новий блок» (Create New Block) (рис. 9);

- обираємо зі списку потрібний компонент, наприклад круглий показуючий прилад «Gauge», налаштовуємо його: обираємо feed – джерело даних, що будуть відображатись, підпис до приладу, розмір приладу, діапазон, розмірність, значення по замовчанню, по закінченню – натискаємо кнопку створити «Створити» (рис 9). Прилад готовий до роботи і вже відображає дані.



[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json). Якщо ви хочете додати більше дошок, просто розділіть URL-адреси комами (рис. 10).

Переходимо до менеджера плат Board manager та встановлюємо модулі підтримки плати ESP8266, далші – аналогічно встановлюємо бібліотеки підтримки Dallas (ds18b20) та Adafruit MQTT library.

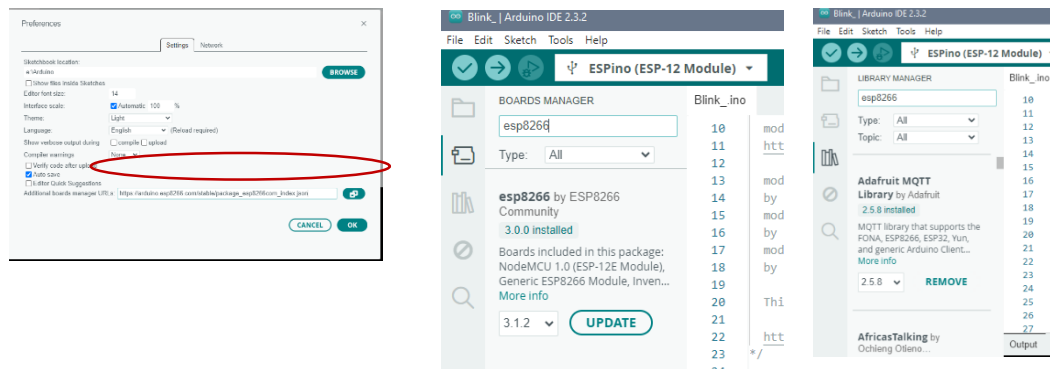


Рис. 10. Налаштування середовища Arduino

По закінченню відкриваємо скейч приклад використання MQTT esp8266, переходимо «File» -> «Exemples» -> «Adafruit MQTT library» -> «MQTT\_esp8266». У налаштуваннях скейчу обираємо плату та порт, у коді прикладу слід вказати назву Wi-Fi мережі SSID Wi-Fi та Пароль Wi-Fi, також ім'я та ключ з серверу Adafruit (див. рис. 2). Зберегти скейч під новим ім'ям, перекомпілювати та завантажити в плату, приклад робочий підключається до мережі Wi-Fi та через інтернет публікує дані на сервер Adafruit.

## Основи роботи з Node-Red

### *Встановлення пакету NODE RED*

Перед встановленням пакету слід встановити платформу Node.JS, для цього:

- завантажити msi-файл Node.JS LTS;
- запустити на виконання msi-файл від імені адміністратора і встановити Node.JS, всі діалогові вікна, всі запити залишати за замовченням.

Встановлення NODE-RED:

в командному рядку набираємо: `npm install -g --unsafe-perm node-red`  
npm це (Node Package Manager)

Запуск NODE-RED:

1. У командному рядку набираємо: node-red.

Хоча краще ще з ім'ям вашого проєкту, node-red fpr («fpr» по типу first project).

2. Відкрийте браузер, перейдіть до редактора Node-Red, вкладка <http://127.0.0.1:1880/>

## **ЛАБОРАТОРНА РОБОТА №1**

**Тема: Створення пристрою «інтернету речей»  
та передача даних хмарному брокеру**

### *Хід роботи*

1. Встановити середовище Ардуіно та бібліотеки роботи з ESP8266, 1-Ware, DS1821.
2. Створити обліковий запис на сервері Adafruit (це безкоштовний брокер).
3. У середовищі Ардуіно відкрити бібліотеку з прикладом роботи з хмарним брокером Adafruit MQTT Library->mqtt\_esp8266, цей приклад можна взяти за основу.
4. Змінити назву Wi-Fi мережі та пароль до мережі на ваші, налаштувати джерела (feeds) для передачі даних на брокер.
5. Перекомпілювати скейч та завантажити скейч в ESP8266.
6. Відкрити інструмент «Монітор порту» та переконатись у вдалому підключення до мережі та брокера.
7. Переглянути дані на брокері.
8. Доповнити скейч бібліотеками роботи з 1-Ware та DS1821, доповнити публікацію температури на брокер.
9. Підключити DS1821 до ESP8266, повторити п. 5-7.
10. Зробити висновки.
11. Оформити звіт. Звіт містить назву роботи, хід роботи, текст програми, скрини екранів та висновки.

### **Контрольні запитання**

1. З якою мінімальною періодичністю можна передавати дані брокеру?
2. Яка кількість каналів доступні в брокері безкоштовно?

3. Скільки датчиків DS18B20 можна підключати до ESP8266?
4. Де в тілі програми дані передаються до брокера?
5. Як реалізована «підписка»?
6. Де в тілі програми реалізовано налаштування датчика температури DS18B20?
7. Де в тілі програми задається періодичність передачі даних до брокера?
8. Як можна візуалізувати дані, отримані з «підписки»?
9. Яка періодичність отримання даних з «підписки»?

## **ЛАБОРАТОРНА РОБОТА №2**

### **Тема: Створення панелі інструментів для візуалізації даних на хмарному брокері**

#### *Хід роботи*

1. Перейти на сторінку «Панель інструментів» хмарного брокера Adafruit IO.
2. Створити нову панель, (для зручності перевірки панель назвати своїм прізвищем) на панелі розмістити візуальні прилади наприклад “Gauge” налаштувати feed – дані що слід візуалізувати.
3. Налаштувати діапазон даних, колір та інші параметри, переконатись коректній роботі панелі приладів.
4. Аналогічно створити та розмістити інші доступну компоненти візуалізації даних.
5. Зробити висновки.
6. Оформити звіт. Звіт містить назву роботи, хід роботи, скрини екранів та висновки.

#### **Контрольні запитання**

1. Скільки візуальних приладів можуть відображати дані з одного каналу (feed)?
2. Які з візуальних компонентів можуть лише відображати дані?
3. Які з візуальних компонентів можуть лише можуть змінювати дані?

4. Які з візуальних компонентів можуть і змінювати і відображати дані?
5. Яка налаштовуються розміри, форма та колір візуальних?
6. Які з компонентів призначені для роботи лише з дискретними даними, а які лише з безперервними?
7. Які налаштування має компонент «Лінійна діаграма»?

### **ЛАБОРАТОРНА РОБОТА №3**

#### **Тема: Знайомство з додатковим функціоналом платформи Adafruit. Створення пристрою «інтернету речей» засобами Adafruit**

##### *Хід роботи*

1. Створити обліковий запис на сервері Adafruit (це безкоштовний брокер).
2. На сервері Adafruit натиснути кнопку «Новий пристрій» та, виконуючи інструкції, створити пристрій.
3. Доповнити пристрій «новим компонентом», наприклад цифровим датчиком температури DS18B20 та налаштувати їх спільну роботу.
4. Перейти на вкладку «Канали» та переконатись в коректній роботі пристрою.
5. Додати пристрій до панелі інструментів, налаштувати візуалізацію отриманих даних.
6. Доповнити пристрій «новим компонентом», наприклад світлодіод та налаштувати їх спільну роботу.
7. Зробити висновки.
8. Оформити звіт. Звіт містить назву роботи, хід роботи, скрини екранів та висновки.

##### **Контрольні запитання**

1. З якою мінімальною періодичністю можна передавати дані брокеру?
2. Яка кількість каналів доступні в брокері безкоштовно?
3. Скільки датчиків DS18B20 можна підключати до пристрою на базі ESP8266?
4. Які параметри налаштування датчика температури DS18B20?
5. Як реалізується «підписка» (на прикладі світлодіоду)?
6. Яка періодичність отримання даних з «підписки»?

## ЛАБОРАТОРНА РОБОТА № 4.0

**Тема: Основи роботи з Node-RED. Знайомство з середовищем.  
Перша програма**

### *Хід роботи*

1. Встановити Node-RED.
2. Створити першу програму у складі вузлів Inject та Debug.
3. Переконайтесь в коректній роботі програми.
4. Доповнити програму вузлом «Функція», в тілі вузла «Функція» написати програму перетворення мітки часу в формат «дата» та «час».
5. Переконайтесь в коректній роботі та зробити висновки.
6. Оформити звіт. Звіт містить назву роботи, хід роботи, скрини екранів та висновки

### **Контрольні запитання**

1. Які налаштування має вузол Inject?
2. Які налаштування має вузол Debug?
3. Який формат посилки з модуля Inject?
4. Куди можна перенаправляти повідомлення з вузла Debug?
5. Поясніть синтаксис програми у вузлі «Функція»?

## ЛАБОРАТОРНА РОБОТА № 4.1

**Тема: Основи роботи з Node-RED. Підключення до хмарного брокера.  
Підписка на отримання даних. Створення панелі приладів**

### *Хід роботи*

1. Скласти програму в складі вузлів «MQTT In» та «Debug».
2. Налаштувати вузол «MQTT In»: сервер, ім'я користувача, назву «feed», на вкладці «Безпека» вказати ім'я користувача та пароль (взяти з «Брокера» My Key).
3. Переконайтесь в коректній роботі програми. Якщо правильні налаштування, програма підключиться до брокера та почине виводити дані у вікно Debug.
4. Доповнити Node-Red палітрою вузлів Dashboard (якщо їх ще не встановлено).
5. Доповнити програму вузлами з панелі приладів, наприклад вузлом «Gauge».

6. Налаштувати діапазон даних, колір та інші параметри, переконатись коректній роботі панелі приладів;
7. Аналогічно створити та розмістити інші доступну компоненти візуалізації даних.
8. Переконатись в коректній роботі програми. Зробити висновки.
9. Оформити звіт. Звіт містить назву роботи, хід роботи, скрини екранів та висновки

### **Контрольні запитання**

1. Які налаштування має вузол MQTT In?
2. Які налаштування має вузол Gauge?
3. Як налаштовують розміри візуальних приладів та їх місце розміщення?
4. Яка ієрархія видових вікон в Node-Red?

## **ЛАБОРАТОРНА РОБОТА № 5**

### **Тема: Основи роботи з Node-RED. Підключення до хмарного брокера. Публікація даних**

#### *Хід роботи*

1. Доповнити програму ЛР 3 вузлами «Тумблер», як джерело даних, та «MQTT Out».
2. Налаштувати вузол «MQTT Out»: сервер, ім'я користувача, назву «feed», на вкладці «Безпека» вказати ім'я користувача та пароль (взяти з «Брокера» My Key).
3. Створити на хмарному брокері потрібний feeds (просто новий з унікальною назвою).
4. Розгорнути програму.
5. Перейти на сайт брокера та переконатись в коректній роботі програми. При правильних налаштуваннях програма підключиться до брокера та починає публікувати дані про стан тумблера «істина» чи «фальш».
6. Повторити п.1 -п.5 з використанням джерел даних: слайдер, вікно вводу даних з клавіатури, та інших доступних в Node-Red.
7. Налаштувати на пристрої на базі ESP8266 (л. р. №3) підписку до даних про стан тумблера.
8. Переконатись в коректній роботі програми. Зробити висновки.

9. Оформити звіт. Звіт містить назву роботи, хід роботи, скрини екранів та висновки

### **Контрольні запитання**

1. Які налаштування має вузол MQTT Out?
2. Які налаштування має вузол Slider?
3. Які є обмеження на публікацію даних у безкоштовному пакеті брокера?
4. Які є обмеження на підписку до даних у безкоштовному пакеті брокера?
5. Яка затримка передачі даних підписнику? (орієнтовно в часі)

### **Список рекомендованої літератури**

#### *Базова література*

1. Kurose, James F. Computer networking: a top-down approach / James F. Kurose, Keith W. Ross – 8th edition. – USA: Pearson Education, 2020.
2. Andrew Tanenbaum, David Wetherall Computer Networks, Global Edition. - Pearson Education Limited, 2021.
3. Software Engineering in IoT, Big Data, Cloud and Mobile Computing. Haengkon Kim, Roger Lee. – Springer Nature Switzerland AG, 2021.
4. The internet of things. Samuel Greengard, Cambridge, Massachusetts – The MIT Press, 2021.
5. Practical Node-RED Programming. Taiji Hagino – Packt Publishing Ltd., 2021.
6. Довідник Node-RED з елементами опису Java Script, JSON, JSONata урк.

#### *Інтернет ресурси*

1. Adafruit.com/ - сайт з підтримкою сервісів для навчання, розробки та тестування пристроїв інтернету речей; безкоштовний хмарний MQTT брокер. Режим доступу <https://io.adafruit.com/>
2. Посилання для завантаження node-red: <https://nodejs.org/en/>
3. Інструкції щодо встановлення та налаштування node-red. Режим доступу: <https://nodered.org/docs/getting-started/windows>
4. Повна технічна інформація, приклади та оновлення: <https://nodered.org/>
5. Довідник Node-RED з елементами опису Java Script, JSON, JSONata Режим доступу: <https://drive.google.com/file/d/1tbhv1j-tiUGpIIAO4kWIIInCRXJh0ZIqf/view>

Навчально-методичне видання

## **ОСНОВИ ПРОМИСЛОВОГО ІНТЕРНЕТУ РЕЧЕЙ**

Методичні вказівки

до виконання лабораторних та практичних робіт  
для здобувачів першого (бакалаврського) рівня вищої освіти  
спеціальності 174 «Автоматизація, комп'ютерно-інтегровані  
технології та робототехніка» всіх форм навчання

Укладач **Самойленко** Микола Іванови

Комп'ютерне верстання *А. П. Селівестрової*

Ум. друк. арк. 0,93. Обл.-вид. арк. 1,0  
Електронний документ. Вид № 49/V-24.

Виконавець і виготовлювач  
Київський національний університет будівництва і архітектури  
Проспект Повітряних Сил, 31, Київ, Україна, 03680

Свідоцтво про внесення до Державного реєстру суб'єктів  
видавничої справи ДК № 808 від 13.02.2002 р