

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Розробка автоматизованої системи моніторингу та управління
виробничими процесами на підприємстві
(тема)

Виконав:
студент 2 курсу, групи КТРСм-21-1
Бондарев А. М.
(прізвище, ініціали)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми Освітньо-професійна

Освітня програма Комп'ютеризовані та
робототехнічні системи
(повна назва освітньої програми)

Керівник доцент Максимова С. С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологійКафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатронікиРівень вищої освіти другий (магістерський)Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Комп'ютеризовані та робототехнічні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2022 р.

ЗАВДАННЯ**НА КВАЛІФІКАЦІЙНУ РОБОТУ**студентові Бондареву Андрію Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка автоматизованої системи моніторингу та управління виробничими процесами на виробництві
затверджена наказом університету від 30.11.2022 р. № 740 Ст.
2. Термін подання студентом роботи до екзаменаційної комісії 12.12.2022 р.
3. Вихідні дані до роботи _____
- 3.1 Апаратно-програмний комплекс на базі Arduino UNO
- 3.2 Мова програмування елементів Arduino C++
- 3.3 Серве AWS IoT, технологія MQTT, нереляційна база даних DynamoDB
- 3.4 Мова програмування для вебдодатку Python, фреймворк Flask
- 3.5 Хмарний сервіс Amazon Web Services
4. Перелік питань, що потрібно опрацювати в роботі
- 4.1 Вступ
- 4.2 Аналіз середовищ та інструментів розробки
- 4.3 Проектування автоматизованої системи моніторингу
- 4.4 Розробка автоматизованої системи моніторингу та управління виробничими процесами на підприємстві
- 4.5 Охорона праці
- 4.6 Висновки
- 4.7 Додатки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
 демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.pptx) – 12 с.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання до кваліфікаційної роботи	10.11.22	виконано
2	Вступ	11.11 – 11.11.22	виконано
3	Аналіз технічного завдання	11.11 – 13.11.22	виконано
4	Аналіз інструментів розробки та аналіз технічного завдання	13.11 – 15.11.22	виконано
5	Проектування системи	15.11 – 15.11.22	виконано
6	Розробка автоматизованої системи моніторингу та управління виробничими процесами на підприємстві	15.11 – 26.11.22	виконано
7	Охорона праці	27.11 – 27.11.22	виконано
8	Висновки	27.11 – 28.11.22	виконано
9	Подання роботи на перевірку Інтернет-сервісом Unichек	02.12 – 02.12.22	
10	Оформлення пояснювальної записки	03.12 – 04.12.22	виконано
11	Подання роботи на рецензію	05.12 – 05.12.22	
12	Подання роботи на підпис зав. кафедри	07.12 – 07.12.22	
13	Подання кваліфікаційної роботи в ЕК	12.12.22	

Дата видачі завдання _____ 10.11.2022 р.

Студент _____
 (підпис)

Керівник роботи _____ доц. Максимова С. С.
 (підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 104 с., 0 табл., 28 рис., 4 дод., 32 джерел.

АВТОМАТИЗОВАНА СИСТЕМА МОНІТОРИНГУ, СЕНСОРИ, ПЛАТФОРМА ARDUINO, ХМАРНІ ОБЧИСЛЕННЯ, AMAZON WEB SERVICES, МОВА ПРОГРАМУВАННЯ PYTHON, ФРЕЙМВОРК FLASK.

Мета роботи – скорочення часу на обробку даних, підвищення рівня контролю важливих показників та ефективності підприємства, можливість контролювати стан робочих об'єктів, виробничих процесів, що виготовляють матеріальну продукцію на підприємстві.

Об'єкт розробки – процес взаємодії апаратно-програмного комплексу на базі платформи Arduino з програмним забезпеченням, розгорнутим у хмарному середовищі Інтернету речей Amazon Web Services за допомогою мови програмування Python і фреймворку Flask.

Предмет розробки – автоматизована система моніторингу інформації з датчиків на базі платформи Arduino та інтеграція з вебдодатком на базі фреймворку Flask для обробки даних з датчиків за допомогою середовищу Інтернету речей Amazon Web Services на підприємстві.

Для розробки автоматизованої системи моніторингу та управління виробничими процесами на підприємстві було проаналізовано технічне завдання, обрано та обґрунтовано програмно-апаратні засоби. Розроблено схеми взаємодії з компонентів та технологій обробки інформації. Були проаналізовані вже існуючі системи моніторингу на підприємствах. Були досліджені датчики, важливість і актуальність збору метрик.

Продемонстровано роботу обробки і розгорнення показників системи моніторингу виробничих процесів за допомогою хмарного сервісу Amazon Web Services, мови програмування Python і фреймворку Flask.

ABSTRACT

Explanatory note: 104 pp., 0 tabl., 28 fig., 4 adj., 32 sources.

AUTOMATED MONITORING SYSTEM, SENSORS, ARDUINO PLATFORM, CLOUD COMPUTING, AMAZON WEB SERVICES, PYTHON PROGRAMMING LANGUAGE, FLASK FRAMEWORK.

The purpose of the work – to reduce the time for data processing, increase the level of control of important indicators and the efficiency of the enterprise, the ability to monitor the state of work objects, production processes that manufacture material products at the enterprise.

The object of development – the process of interaction of the hardware and software complex based on the Arduino platform with software deployed in the Amazon Web Services cloud environment of the Internet of Things using the Python programming language and the Flask framework.

The subject of development – an automated system for monitoring information from sensors based on the Arduino platform and integration with a web application based on the Flask framework for processing data from sensors using the Amazon Web Services Internet of Things environment at the enterprise.

To develop an automated system for monitoring and managing production processes at the enterprise, the technical task was analyzed, software and hardware tools were selected and substantiated. Schemes of interaction of components and information processing technologies have been developed. Already existing monitoring systems at enterprises were analyzed. Sensors, importance and relevance of metrics collection were explored.

The work of processing and deploying indicators of the production process monitoring system using the Amazon Web Services cloud service, the Python programming language, and the Flask framework is demonstrated.

ЗМІСТ

	с.
Перелік скорочень	8
Вступ.....	9
1 Аналіз актуальності розробки та технологій систем	11
1.1 Актуальність розробки системи	11
1.2 Аналіз сучасних систем та компонентів моніторингу	12
1.2.1 Датчики збору зовнішніх показників і модулі інтеграції.....	13
1.2.2 Апаратно-програмні платформи та мікроконтролери	15
1.2.2.1 Платформа Arduino	15
1.2.2.2 Платформа Raspberry Pi.....	18
1.2.3 Програмне забезпечення обробки інформації.....	20
1.2.4 Проєктування розгорнення системи у мережі Інтернет	21
1.3 Висновки до першого розділу.....	23
2 Проєктування структури та розробки автоматизованої системи моніторингу	24
2.1 Проєктування інфраструктури системи моніторингу	24
2.2 Послідовність і принцип дії автоматизованої системи	26
2.3 Проєктування програмного забезпечення та формування необхідних засобів для обробки даних.....	30
2.4 Аналіз особливостей розгортання додатку у AWS	31
2.4.1 AWS IoT.....	32
2.5 Висновки до другого розділу	36
3 Розробка і розгортання програмного та апаратного забезпечення	37
3.1 Створення загальної архітектури автоматизованої системи моніторингу....	37
3.2 Розробка апаратного забезпечення на платформі Arduino	38
3.2.1 Під'єднання компонентів до Arduino	38
3.2.2 Середовище розробки апаратного забезпечення і поеднання з AWS IoT.....	43

	7
3.3 Середовище та розробка вебдодатку на Flask та Python.....	47
3.4 Висновки до третього розділу.....	53
4 Охорона праці	54
4.1 Організація охорони праці на підприємстві.....	55
4.1.1 Охорона праці на підприємстві	55
4.2 Обов'язки роботодавця	56
4.3 Служба охорони праці на підприємстві.....	57
4.4 Відповідальність за порушення вимог законодавства про охорону праці....	58
4.5 Правила роботи за комп'ютером.....	59
4.6 Розрахунок освітленості робочого приміщення і робочого місця для забезпечення безпечних умов роботи	60
4.7 Висновки до четвертого розділу.....	62
Висновки	63
Перелік джерел посилання	65
Додаток А Лістинг коду для платформи Arduino	69
Додаток Б Лістинг коду програмного забезпечення на мові програмування Python та фреймворку Flask	79
Додаток В Демонстраційний графічний матеріал	98
Додаток Г Відомість кваліфікаційної роботи.....	104

ПЕРЕЛІК СКОРОЧЕНЬ

ІТ – інформаційні технології;

ІЧ – інфрачервоне;

МК – мікроконтролер;

МП – мікропроцесор;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

API – application programming interface – прикладний програмний інтерфейс;

AWS – Amazon Web Services – платформа хмарних обчислень;

IoT – internet of things – інтернет речей;

SQL – Structured Query Language – мова структурованих запитів;

MQTT – Message Queuing and Telemetry Transport – черга повідомлень;

WSS – Websockets Secure – захищений протокол вебсокетів;

SDK – Software development kit – набір для розробки ПЗ;

IDE – Integrated Development Environment – система для розробки ПЗ;

HTTP – Hypertext Transfer Protocol – протокол передачі даних;

WPAN – wireless personal area network – бездротові персональні мережі;

WLAN – wireless local area network – бездротова локальна мережа.

ВСТУП

Останнім часом відбувся значний прогрес у системі моніторингу в реальному часі на основі Інтернету речей (IoT). Рівень використання Інтернету речей зростає експоненціально через його величезне застосування в різних сферах, і багато з них ще належить дослідити. Використання інформації про критичні показники для виконання робочих процесів не є чимось новим для будь-яких підприємств. Цей процес створювався і підтримується для оптимізації як фінансових, так і виробничих витрат.

Також важливо бути інтегрованим чи здатним до міграції до сервісів хмарних обчислень. AWS пропонує сервіси та рішення Інтернету речей (IoT), які допомагають підключатися до мільярдів пристроїв та керувати ними. Існує можливість збирати, зберігати та аналізувати дані IoT для робочих навантажень у промисловості, сфері споживчих товарів, комерційній та автомобільній галузях. Також хмарні обчислення – це завжди про безпеку даних, стабільність робочих систем.

Автоматизована система моніторингу та управління виробничими процесами на підприємстві виконує наступні основні функції:

- збір критичних показників для робочих процесів на виробництві;
- візуалізація інформації показників і управління пороговими значеннями;
- виявлення та фіксація критичних ситуацій для робочих процесів на виробництві.

Виходячи з перерахованої інформації, з використанням наведеної автоматизованої системи моніторингу ефективність виробництва і його ймовірна збитковість на далекому можливому прогнозі буде зручніше відстежуватися по критичним показникам для роботи підприємства. Наявність моніторингу буде запобігати поганим наслідкам повного або часткового виходу із строю елементів вироблення матеріального продукту підприємства, що є частиною фінансової

бізнес-моделі. Промисловий IoT допомагає масштабувати ресурси, збирати та відправляти інформацію, дистанційно керувати процесами, автоматизувати виробництво.

Таким чином, актуальною задачею стає розробка автоматизованої системи моніторингу та управління виробничими процесами на підприємстві.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати вже існуючі системи моніторингу на підприємствах, системи на базі датчиків, що збирають інформацію про критичні показники для виконання робочих процесів;
- дослідити датчики, які можуть використовуватися на підприємствах і важливість тих чи інших показників, актуальність збору метрик;
- розглянути мікроконтролерні рішення у вигляді платформ для апаратного забезпечення;
- обґрунтування вибору апаратних і програмних середовищ для розробки апаратного і програмного забезпечення;
- побудувати інфраструктуру обробки і розгорнення показників системи моніторингу виробничих процесів за допомогою хмарного сервісу Amazon Web Services, мови програмування Python і фреймворку Flask;
- розробити автоматизовану систему моніторингу, використовуючи всі досліджені технології.

Звіт з кваліфікаційної роботи виконано згідно з [1-4].

Результати роботи пройшли апробацію на 26-му Міжнародному молодіжному форумі “Автоматизовані системи та комп’ютеризовані технології радіоелектронного приладобудування”, а також на 17-тій Міжнародній науково-практичній конференції “Мультидисциплінарні наукові записки. Теорія, методологія і практика”. Тези були опубліковані у збірнику студентських наукових та на міжнародній конференції: “Автоматизація та приладобудування ADED-2022” [5], “Виробництво & Мехатронні Системи 2022” [6].

1 АНАЛІЗ АКТУАЛЬНОСТІ РОЗРОБКИ ТА ТЕХНОЛОГІЙ СИСТЕМИ

1.1 Актуальність розробки системи

Моніторинг інформаційно-технологічних систем є складовою управління інформаційної інфраструктури підприємства, що полягає у постійному спостереженні та періодичному аналізі компонентів робочих процесів з відстеженням динаміки що відбуваються із нею змін. Ключовим завданням систем моніторингу є отримання, збереження та аналіз інформації про стан підконтрольних елементів структури компанії.

Необхідність шукати методи виявлення проблем на виробництві під час виконання робочих процесів є важливим компонентом та показником продуктивності бізнес-структури. Для необхідної результативності треба мати можливість своєчасно помічати та вирішувати подібні проблеми, щоб вони не допускали негативних перспектив, критичних інцидентів та не тягнули за собою матеріальні втрати у компанії. Актуальність теми моніторингу у технологічній сфері це особливо актуально, бо неякісне використання виробничої потужності або матеріалів може потягнути за собою тенденцію на негативну ефективність виробництва і його збитковість на далекому можливому прогнозі. Тоді коли наявність навіть самого простого моніторингу може запобігти ці погані наслідки.

Промисловий IoT виділяється масштабами і призначенням. Можливість збирати та відправляти інформацію, дистанційно керувати процесами, автоматизувати виробництво. Власнику або директору виробництва не прийдеться витратити кошти на утримання додаткового персоналу, який буде виконувати ті ж самі завдання, з якими зможе впоратись автоматизована система на основі модуля обробки інформації. Отримана інформація може бути використана для запобігання позаплановим простоям, поламкам устаткування, скороченню позапланового техобслуговування та збоям в управлінні ланцюжками поставок, тим самим

дозволяючи підприємству функціонувати більш ефективно. Дані будуть передаватися у хмарний сервіс, наприклад Microsoft Azure, де висококваліфіковані спеціалісти подбають про захист даних.

Актуальність системи контролю і моніторингу важливих показників під час робочих процесів на виробництві є також важливим пунктом у питанні безпеки, але більше користі буде від безперебійної роботи бізнесу. Розроблювальний вебдодаток дозволяє оперативно відреагувати на проблему в роботі сервісів, а також ефективно запобігати виникненню неполадок. Тому і була висунута ідея на створіння подібної системи, яка буде являти собою автоматизованої системи моніторингу та управлінням робочими процесами на виробництві. Під робочими процесами мається на увазі створення матеріального продукту, необхідного для існування і розвитку виробництва як бізнес-структурі, а моніторинг необхідна частина для покращення ефективності створення цього продукту.

1.2 Аналіз сучасних приладів та систем моніторингу зовнішніх показників

Під час проектування та розробки систем моніторингу, ядром яких є інформаційно-вимірювальні системи, одним з основних питань є обґрунтування множини діагностичних ознак. Аргументовано вибрати такі ознаки дозволяє побудова і аналіз математичних моделей об'єктів моніторингу та діагностики або фізичних процесів, що супроводжують роботу цих об'єктів.

Моніторинг на основі аналізу випадкових процесів та полів, включаючи і шумову діагностику, базується, в першу чергу, на відповідним чином побудованих математичних моделях, які встановлюють зв'язок між фактичним технічним станом об'єкта і деякими характеристиками та параметрами, отриманими в результаті опрацювання вимірювальної інформації. Ця математична модель є випадковим процесом або випадковим полем з деякого класу, якій притаманні певні властивості, що дозволяють оцінити дану модель за допомогою статистичних чи інших методів на основі вимірювання та опрацювання однієї або більше реалізацій процесу (поля). При побудові імовірнісних математичних моделей фізичних процесів у вузлах теплоенергетичного обладнання найзручніше використовувати класи лінійних

випадкових процесів. Такі моделі є конструктивними, тобто кожен їх параметр має конкретний фізичний зміст. Крім того, властивості лінійних випадкових процесів дозволяють отримувати аналітичні вирази для моментів розподілу досліджуваного процесу, будувати його функції розподілу або характеристичні функції. На основі аналізу таких моделей можна теоретично обґрунтувати діагностичні ознаки, визначати допустимі межі їх зміни (діагностичні простори), будувати ефективні вирішальні правила щодо виявлення та класифікації дефектів. Крім того такі моделі дозволяють скористатися методами імітаційного комп'ютерного моделювання. Такий підхід дозволяє скоротити витрати часу і коштів на проведення фізичного моделювання процесів моніторингу або експериментальних досліджень на реальному обладнанні.

1.2.1 Датчики збору зовнішніх показників і модулі інтеграції

Існує багато різних типів датчиків для промислового та комерційного використання. Датчики використовуються майже в усіх галузях промисловості для додатків, з якими ми стикаємося щодня, а також для більш промислових і спеціалізованих застосувань [5]. Розгортання датчиків у виробничій промисловості продовжує суттєво зростати – завдяки підвищенню обізнаності про те, що вони можуть, у поєднанні зі все більш доступною ціною.

Датчик використовується для вимірювання, вимірювання буде оброблено, а результат процесу буде надано на виході. Тоді результат спричинить зміни чи переміщення. Простий приклад – датчик температури в термостаті. Датчик температури постійно контролює температуру, як тільки проведені вимірювання досягають бажаної температури, вимірювання обробляється, і вихід викликає вимикання котла.

Датчики, які використовуються у промисловості, контролюють продуктивність різних процесів і аспектів роботи машини, збираючи дані для визначення нормального базового рівня роботи, а також виявляючи навіть наймізерніші коливання продуктивності. Потім ці дані можуть увімкнути прогнозне технічне

обслуговування – більш ефективний і результативний спосіб уникнути простоїв і підтримувати обладнання в робочому стані. Останньою частиною сенсорної головоломки є зв'язок і підключення в режимі реального часу з використанням сучасних високошвидкісних мережевих можливостей мережі Інтернет, щоб забезпечити моніторинг даних і обробку даних у режимі реального часу.

Існує багато типів виробничих датчиків і засобів керування процесами, кожен з яких призначений для моніторингу та збору даних про різні процеси, вектори та показники продуктивності обладнання. Кожен із типів датчиків, що використовуються в промисловій автоматизації, служить певній меті. Не всі датчики будуть потрібні для всіх застосувань – реалізація повинна визначатися обладнанням, що використовується, а також цілями стратегії прогнозованого технічного обслуговування. На рисунку 1.1 можна побачити інформаційні моніторингові панелі, розгорнуті на різних пристроях:



Рисунок 1.1 – Структурна схема інформаційних моніторингових панелей

Триосьові датчики акселерометра контролюють вібрацію на обертовому обладнанні та проводять вимірювання, включаючи швидкість, прискорення та переміщення. Ці дані корисні для відстеження узгодженості піків і спадів у вібрації

обладнання. Якщо вібрація виникає передбачуваним чином, можна вважати, що обладнання працює належним чином. Якщо спостерігаються коливання або відхилення від постійного зчитування вібрації, це майже завжди означатиме, що обладнання не працює належним чином і що необхідне подальше дослідження (і обслуговування).

Температура є одним із найнадійніших індикаторів працездатності обладнання. Загалом можна очікувати, що частина обладнання працюватиме при постійній температурі з охолоджуючими елементами (такими як охолоджуюча рідина, потік повітря або вода), якщо це необхідно, щоб полегшити дотримання послідовності. Таким чином, несподівані та неконтрольовані температурні коливання або стрибки, ймовірно, означають, що щось не працює з максимальною продуктивністю в частині обладнання.

Датчики крутного моменту також використовуються з обертовим обладнанням [5]. У поєднанні з датчиками вібрації персонал з технічного обслуговування може отримати безпрецедентну інформацію та практичні дані для раннього виявлення та вирішення потенційних проблем, максимізуючи час безвідмовної роботи обладнання та майже усуваючи незаплановані простої. Це пов'язано з моніторингом моторних функцій у машинах, що є одним із найважливіших застосувань машинних датчиків. Монітори стану двигуна [6] є життєво важливими для вимірювання загального стану обладнання та виявлення ранніх попереджувальних ознак загрозливих поломок. Завдяки цим можливостям, доданим до обладнання, стає легше діагностувати проблеми, перш ніж вони матимуть можливість вплинути на продуктивність.

1.2.2 Апаратно-програмні платформи та мікроконтролери

1.2.2.1 Платформа Arduino

Плата мікроконтролера відома як невеликий комп'ютер, побудований на мікросхемі напівпровідника з оксиду металу. Спільним для всіх типів

мікроконтролерів є те, що вони мають однакові основні будівельні частини, як-от центральний процесор (CPU), вхід/вихід (I/O), пам'ять і периферійні пристрої (програмовані). Окрім однієї з найважливіших частин будь-якого комп'ютера, мікроконтролери є дуже цікавими речами, які навіть не можна собі уявити, і вони також дуже корисні в багатьох аспектах [7].

Мікроконтролери набули великої популярності завдяки своїй функціональності в усьому світі, і сьогодні вони регулярно використовуються майже в усіх автоматично керованих продуктах і пристроях, таких як пристрої керування, автомобільні двигуни, пульти дистанційного керування, прилади, електроінструменти, іграшки, імплантовані медичні пристрої та інші вбудовані системи. Вони привертають першу увагу інженерів, ентузіастів і програмістів і є першим вибором для створення проєктів своїми руками та навчання.

Мікропроцесор можна визначити як тип мініатюрного електронного пристрою, розміщеного на материнській платі комп'ютера, який містить арифметику, логіку та схеми керування, необхідні для виконання функцій центрального процесора цифрового комп'ютера. Це важливий електронний компонент комп'ютера, який дозволяє комп'ютеру ефективно виконувати свої завдання.

Далі його можна описати як центральний процесор, інтегрований в одну інтегральну мікросхему, що містить мільйони дрібних компонентів, як-от резистори, транзистори та діоди, які працюють разом. Для імплементації обробки сигналів з датчиків і передачі метрик до вебдодатку для обробки і візуалізації даних будуть розглядатися мікроконтролерні платформи Arduino і Raspberry PI, як найпростіші рішення для реалізації.

Проєкт Arduino було розпочато в 2005 році. Мета полягала в тому, щоб використовувати його як інструмент для студентів Інституту проєктування взаємодії Івреа, Італія. Іншим мотивом було надати недорогий і простий спосіб для початківців і професіоналів створювати такі пристрої, які взаємодіють з навколишнім середовищем за допомогою датчиків і приводів.

Arduino відомий своїм простим інструментом для створення прототипів електроніки, який має апаратне та програмне забезпечення з відкритим кодом.

Arduino – це тип плати для розробки мікроконтролерів, який використовується для багатьох цілей, як-от прийом вхідних даних від кнопок, блимання світлодіодів, читання даних із датчиків, керування двигунами та багато інших завдань, пов'язаних із «мікроконтролером».

Найпопулярніша Arduino UNO [8] є однією з найпоширеніших плат Arduino, доступних на ринку. Це плата мікроконтролера з відкритим кодом, яка базується на мікроконтролері ATmega328P і розроблена Arduino.cc. Тут відкритий вихідний код означає, що файли дизайну та вихідний код для програмного забезпечення та бібліотек є у вільному доступі, тому будь-хто, хто цікавиться програмуванням, може безкоштовно використовувати їх як довідник для створення власної плати Arduino. Плата оснащена наборами контактів цифрового та аналогового входу/виводу (I/O), які можна сполучати з різними платами розширення (щитами) та іншими схемами.

Його зручні функції, як вбудований світлодіод, великі роз'єми з кутом 2,54 мм для підключення зовнішніх пристроїв, вбудоване керування живленням (наприклад, зовнішній роз'єм постійного струму) і великий роз'єм USB B для підключення до ПК, підвищують його популярність (рис.1.2).

Arduino UNO – плата мікроконтролера на основі ATmega328P. Він має 14 цифрових входів/виходів (з яких 6 можна використовувати як ШІМ-виходи), 6 аналогових входів, керамічний резонатор 16 МГц, USB-з'єднання, роз'єм живлення, роз'єм ICSP і кнопку скидання. Містить все необхідне для підтримки мікроконтролера; просто підключіть його до комп'ютера за допомогою кабелю USB або живіть його за допомогою адаптера змінного струму в постійний струм або акумулятора, щоб почати [9]. Є можливість виконувати певні дії з Arduino UNO, не турбуючись про те, що буде зроблено щось не так. У найгіршому випадку можна замінити чіп за кілька доларів і почати заново. На відміну від усіх попередніх плат, що використовували FTDI USB мікроконтролер для зв'язку USB, новий Arduino UNO використовує мікроконтролер ATmega8U2.

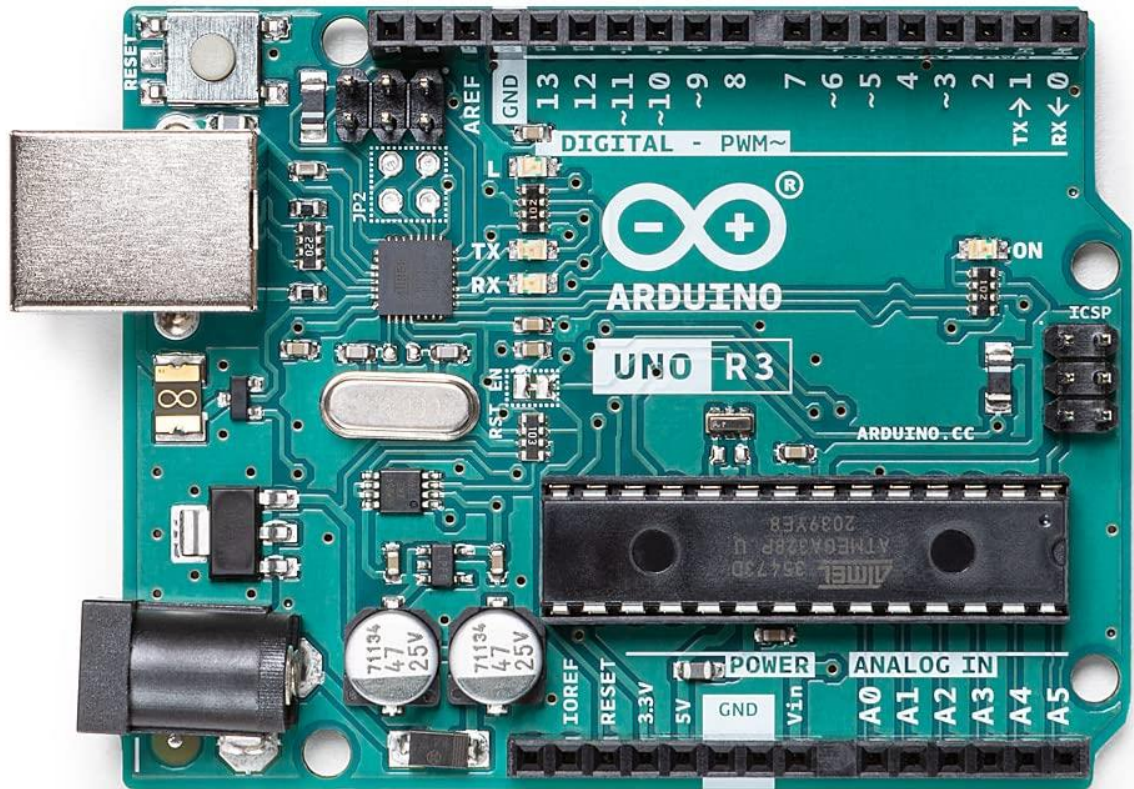


Рисунок 1.2 – Arduino UNO REV3

1.2.2.2 Платформа Raspberry Pi

Raspberry Pi – добре відома назва серії одноплатних комп'ютерів, розроблених британською благодійною організацією Raspberry Pi Foundation. Метою його створення є навчання людей комп'ютерній техніці та надання легшого доступу до комп'ютерної освіти в усьому світі. Слід зазначити один важливий момент: Arduino – це плата розробки на основі мікроконтролера, тоді як Raspberry Pi – це плата на основі мікропроцесора (зазвичай серії ARM Cortex-A), яка працює як комп'ютер [10].

Raspberry Pi має такі ж функції, як настільний комп'ютер, де багато периферійних пристроїв, як-от монітор (через HDMI або AV-порт), клавіатура та миша (через USB), а також камера (через спеціальний інтерфейс камери) та Інтернет (через Ethernet або Wi-Fi), можна підключити (рис. 1.3).

Найдешевший Raspberry Pi – це Zero W, який продається за 10 доларів і забезпечує повноцінний комп'ютер Linux із Wi-Fi і Bluetooth і доступом до важливого GPIO (хоча вам доведеться самостійно спаяти контакти). Використання Pi Zero W як вбудованого пристрою є недорогим способом створення проєкту IoT. Найдорожчим Raspberry Pi є Pi 4 8GB, який продається за 75 доларів, але щоб отримати максимальну віддачу від цієї плати, вам потрібно буде придбати додаткові аксесуари та NAT-плати.

Вартість Arduino може коливатися від кількох доларів за плату клонування до майже 100 доларів за офіційну Portenta H7. ATtiny85, недорогий мікроконтролер із лише шістьма контактами GPIO, можна придбати менш ніж за 2 долари та забезпечує достатню потужність для робототехнічних проєктів. Клони Arduino UNO можна купити відносно дешево і вони забезпечують досить хорошу сумісність у порівнянні з офіційними платами.

Отже, існує широкий вибір плат контролерів, які можна використовувати для апаратних проєктів підприємства [11]. Дві найпопулярніші серед них: Arduino і Raspberry Pi. Arduino базується на сімействі ATmega і має відносно простий дизайн і структуру програмного забезпечення. Raspberry Pi, по суті, є одноплатним комп'ютером. Обидва вони мають центральний процесор, який виконує інструкції, таймери, пам'ять і контакти введення/виведення. Ключова відмінність між ними полягає в тому, що Arduino, як правило, має потужні можливості вводу-виводу, які безпосередньо керують зовнішнім обладнанням. У той час як Raspberry Pi має слабкий вхід-вивід, який потребує транзисторів для керування обладнанням. Давайте подивимося на різницю між Arduino та Raspberry Pi. Arduino є явним переможцем за найнижчою вартістю плати. Але тут є кілька застережень. Для програмування Arduino знадобиться комп'ютер, а також компоненти.

1.2.3 Програмне забезпечення обробки інформації

Arduino можна програмувати за допомогою мов програмування C або C++. Ця платформа на базі мікроконтролера використовує власну мову програмування,

схожу на C++. Однак можна використовувати Arduino з Python або іншою мовою програмування високого рівня. Насправді такі платформи, як Arduino, добре працюють з Python, особливо для програм, які потребують інтеграції з датчиками та іншими фізичними пристроями.

Загалом, Arduino та Python можуть створити ефективне навчальне середовище, яке спонукає розробників до розробки електроніки. Якщо вже є знання основ Python, тоді є можливість розпочати роботу з Arduino, використовуючи для керування нею Python [12].

Платформа Arduino включає як апаратні, так і програмні продукти і у даній роботі будуть використовуватися апаратне забезпечення Arduino та програмне забезпечення Python, основні схеми, а також цифрові та аналогові входи та виходи. Мова програмування Python чудово підходить для побудови відкразостійкої автоматизованої системи моніторингу. З ним і його фреймворком Flask можна буде працювати з AWS IoT, що підходить концепції архітектури.

1.2.4 Розгорнення системи у мережі Інтернет

Хмара дає легкий доступ до широкого спектру технологій, щоб була можливість втілити різні інфраструктурні рішення. Ви можете швидко нарощувати ресурси в міру потреби – від інфраструктурних послуг, таких як обчислення, зберігання та бази даних, до Інтернету речей, машинного навчання, аналітики даних та багато що ще. Завдяки можливостям AWS реалізація на кілька порядків швидше, ніж раніше і це дає свободу експериментувати, тестувати нові ідеї, щоб відрізнити досвід клієнтів і трансформувати свій бізнес.

AWS створена як найбільш гнучка та безпечна серед доступних сьогодні хмарних обчислень [13]. Основна інфраструктура AWS створена, щоб задовольнити вимоги безпеки для військових, глобальних банків та інших високочутливих організацій. Це підкріплено широким набором інструментів хмарної безпеки з 230 службами та функціями безпеки, відповідності та управління. AWS підтримує 90 стандартів безпеки та сертифікатів відповідності, а всі 117 служб AWS, які зберігають дані клієнтів, пропонують можливість шифрувати ці дані. Це свідчить

про те, що використання AWS зробить розроблювальну систему моніторингу більш захищеною.

Основна проблема безпеки у хмарі – старіння ключів доступу. Провайдери всіляко наполягають на тому, щоб проводилася постійна ротація ключів, але це не форсують. Очевидно, потрібний інструмент для примусової ротації. Більше того, інструмент повинен задовольняти політикам конфіденційності та технічним нормам конкретної організації [14]. Але з іншої сторони, це і робить систему безпечніше.

Завдяки хмарним обчисленням не доведеться наперед виділяти зайві ресурси, щоб в майбутньому впоратися з піковими рівнями ділової активності. Замість цього надається необхідна кількість ресурсів. Є можливість збільшити або зменшити ці ресурси, щоб миттєво збільшувати та зменшувати потужність у міру зміни потреб вашого бізнесу. Наприклад, контейнерні та оркестрові послуги AWS полегшують керування базовою інфраструктурою, як локально, так і в хмарі. Це дозволяє зосередитися на інноваціях та потребах свого бізнесу. Майже 80 відсотків усіх контейнерів у хмарі сьогодні працюють на AWS. Такі клієнти, як Samsung, Expedia, GoDaddy і Snap, вибирають запускати свої контейнери на AWS для безпеки, надійності та масштабованості.

Також як практичний приклад використання інфраструктури AWS і, як наслідок, довіри серед державних органів, можна розглянути як 15 липня, у Лондоні, міністр цифрової трансформації Михайло Федоров та Директор з цифрових трансформацій у державному секторі Amazon Web Services (AWS) Ліам Маквелл підписали Меморандум про взаєморозуміння щодо прискорення цифрової трансформації та впроваджень інновацій в Україні. «Протягом багатьох років Amazon (AWS) мав сильну, віддану і зростаючу спільноту клієнтів в Україні. Ми вітаємо підписання цього Меморандуму про взаєморозуміння з українським Урядом і сподіваємося на спільну роботу в напрямі цифрового розвитку держави», – Ліам Маквелл, Директор з цифрових трансформацій у державному секторі, Amazon Web Services в Європі.

Завдяки використанню запропонованої автоматизованої системи моніторингу з використанням хмарного вебсервісу AWS можна буде слідкувати за необхідним освітленням, температурою та вологістю, шумом і вібрацією. Контролювання цих

показників допоможе тримати виробництво у ефективному та відмовостійкому стані. Автоматизована система моніторингу охоплює важливі показники під час робочих процесів на виробництві та займається обробкою даних з показників за допомогою сенсорів та сучасних інформаційно-технічних технологій для стабілізації покращення ефективності створення матеріального продукту, необхідного для існування і розвитку виробництва як бізнес-структурі. Також завдяки використанню хмарних технологій вебпровайдера AWS розроблювальна система буде більш захищеною методами шифрування і безпеки інфраструктури хмарного провайдера та більш еластичною у доробляння функцій, впровадження нових можливостей [15].

Таким чином, автоматизація процесу контролю показників підвищить швидкість реагування на проблеми під час виробництва, дозволить якісно проводити аналізи станів та дозволить запобігти критичним ситуаціям на підприємств.

1.3 Висновки до першого розділу

У першому розділі була проведена робота над аналізом розробки автоматизованої системи моніторингу та управління виробничими процесами на підприємстві, а саме була проаналізована актуальність розробки, були наведені приклади вже існуючих систем та проаналізовані використовуємі технології.

Був проведений аналіз сучасних систем та компонентів моніторингу, а саме детчиків збору зовнішніх показників і модулів інтеграції. Були порівняні апаратно-програмні платформи і обрана для проєктування Arduino. Також був проведений аналіз хмарної вебплатформи Amazon Web Services.

На основі зібраної і проаналізованої інформації було схематично змодельовано розгортання програмного забезпечення системи обробки інформації у мережі Інтернет.

2 РОЗРОБКА СТРУКТУРИ ТА РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ

2.1 Проектування інфраструктури системи моніторингу

Під час проектування інфраструктури системи моніторингу за мету було взято створення автоматизованої системи моніторингу важливих показників під час робочих процесів на виробництві за допомогою сенсорів та сучасних інформаційно-технічних технологій. Під робочими процесами мається на увазі створення матеріального продукту, необхідного для існування і розвитку виробництва як бізнес-структурі, а моніторинг необхідна частина для покращення ефективності створення цього продукту.

Для прикладу можна розглянути підприємство, яке виробляє матеріальний продукт, і для якого важливо дотримання температурного режиму для матеріалів різного роду – це можуть бути продукти хімічної промисловості, з електричною провідністю. При порушенні кліматичного режиму під час обробки і виробництва з таких товарів може вести до значного показника браку у кінцевому варіанті. Щоб запобігти цьому можна налаштувати на робочих ділянках моніторинг на основі датчику вологості та температури, який буде передавати показники у відповідну директорію. Таких датчиків може бути стільки, скільки необхідно, а розраховується ця необхідність тими самими показниками, які можуть впливати на якість виробленого продукту.

Актуальною задачею є створення автоматизованої системи моніторингу освітлення, температури і вологості, шуму та вібрації – важливих показників під час робочих процесів на виробництві за допомогою сенсорів та сучасних інформаційно-технічних технологій з використанням AWS. Під робочими процесами мається на увазі створення матеріального продукту, необхідного для існування і розвитку виробництва як бізнес-структурі, а моніторинг необхідна частина для покращення ефективності створення цього продукту [16]. У той час AWS є важливою частиною в даній системі тому, що саме у вебхмарі буде відбуватися розгортання виконаного

додатку. Крім того, що вебсервіс AWS добре виконує своє завдання, також варто врахувати без того найкращі сторони даного хмарного провайдера.

Хмара дає легкий доступ до широкого спектру технологій, щоб була змога швидше вводити інновації та створювати все, що дозволяють використовувані технології. Можна швидко нарощувати ресурси в міру потреби – від інфраструктурних послуг, таких як обчислення, зберігання та бази даних, до Інтернету речей, машинного навчання, аналітики даних та багато що ще. Завдяки можливостям AWS реалізація на кілька порядків швидше, ніж раніше і це дає свободу експериментувати, тестувати нові ідеї, щоб відрізнити досвід клієнтів і трансформувати свій бізнес.

AWS створена як найбільш гнучка та безпечна серед доступних сьогодні хмарних обчислень [17]. Основна інфраструктура AWS створена, щоб задовольнити вимоги безпеки для військових, глобальних банків та інших високочутливих організацій. Це підкріплено широким набором інструментів хмарної безпеки з 230 службами та функціями безпеки, відповідності та управління. AWS підтримує 90 стандартів безпеки та сертифікатів відповідності, а всі 117 служб AWS, які зберігають дані клієнтів, пропонують можливість шифрувати ці дані. Це свідчить про те, що використання AWS зробить розроблювальну систему моніторингу більш захищеною.

Для дотримання технологічних норм під час виробництва матеріального продукту повинні підтримуватись оптимальні показники, які можуть сприяти у ефективності виробництва. У роботі будуть розглядатися датчики, які зможуть забезпечити інформацією систему моніторингу, а саме:

- датчик освітлення;
- датчик температури та вологості;
- датчик шуму;
- датчик вібрації.

Стає питання у використанні системи, яка буде являти собою процес взаємодії мікроконтролера на базі мікроконтролера з датчиків, звідки інформація буде

відображатися у програмному забезпеченні за допомогою хмарного вебсервісу та бездротових технологій. У даній роботі буде описуватися створення подібної автоматизованої системи моніторингу на виробництві, з якою можна ознайомитися на рис.2.1:

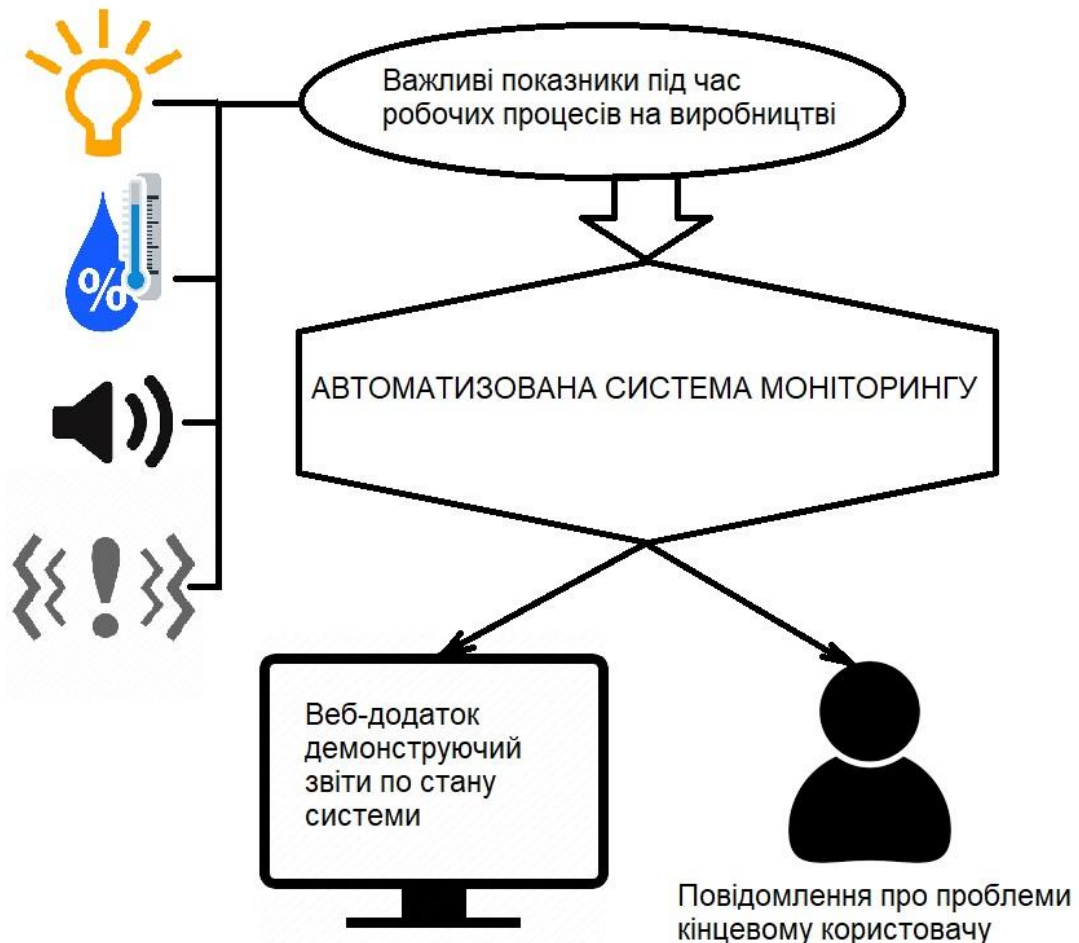


Рисунок 2.1 – Функції автоматизованої системи моніторингу

2.2 Послідовність і принцип дії автоматизованої системи

Для автоматизованої обробки даних з показників датчиків шуму, освітлення, температури і вологості та вібрації пропонується створення системи, яка буде враховувати та вирішувати недоліки сучасних популярних аналогів, беручи за основу існуючі компоненти, та вдосконаливши ефективність [18].

Проаналізувавши матеріали технічних документацій та відгуків про роботу датчиків у виробництві, мікроконтролерних платформ, апаратно-програмних

рішень реалізації взаємодії обробки інформації, було прийнято рішення включити до складу запропонованої до розроблення системи для автоматизації контролю та обліку показників наступні елементи на рисунку 2.2:

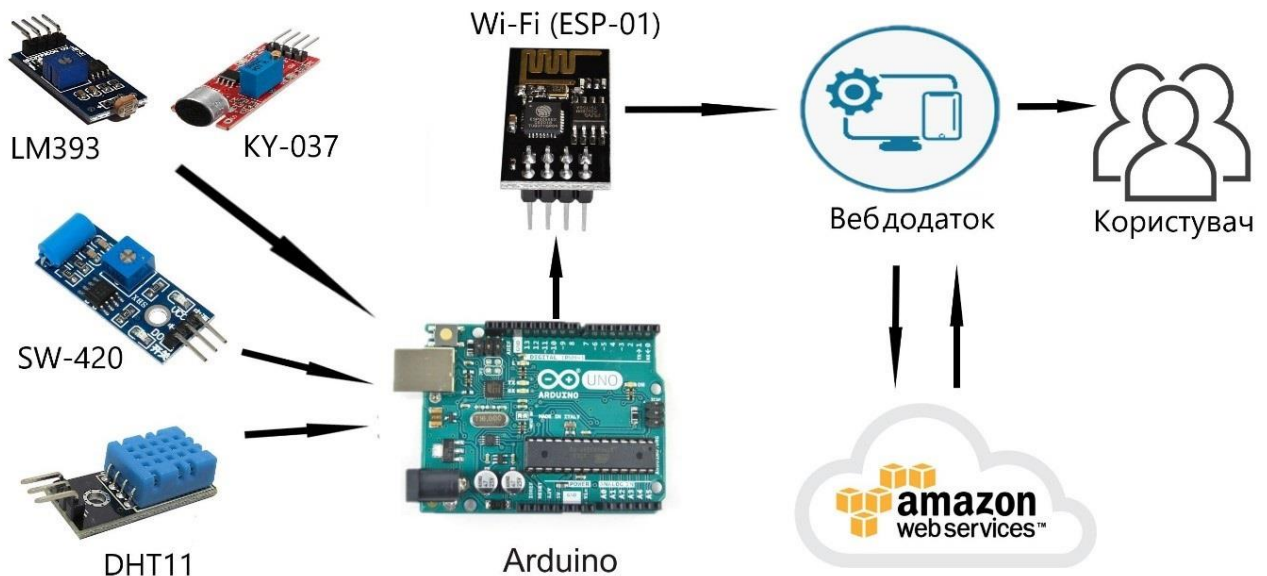


Рисунок 2.2 – Загальна схема роботи автоматизованої системи моніторингу

Wi-Fi модуль ESP-01 – найпопулярніший модуль серії ESP8266. Керуючий пристрій взаємодіє з ESP8266 через UART (Serial-порт) за допомогою набору команд AT. Робота над прийомом і передачі даних виглядає, як взаємодія з TCP-сокетом або з serial-портом комп'ютера. Програмувати та завантажувати прошивки можна через Arduino IDE [19], так само, як при роботі з Arduino. Реакція на AT-команди – це просто функція штатної прошивки, що встановлюється на заводі. Оскільки на платі є 2 порти вводу/виводу, то після прошивки можна обійтися без додаткового контролера, підключаючи периферію безпосередньо до модуля. Для прошивки модуля знадобиться звичайний програматор, як і для деяких плат Arduino. PCB антена модуля забезпечує дальність зв'язку до 400 м на відкритому просторі. Використання Wi-Fi модуля ESP-01 можливе завдяки існуючому та бурхливому програмному забезпеченню. Модуль ESP-01 успішно застосовується в стаціонарних і мобільних приладах у складі систем "розумний будинок", рухомих і пристроях, що носяться. Сучасні побутові прилади завдяки застосуванню ESP-01 стають екосистемою інтернету речей. Бездротовий зв'язок Wi-Fi і функція зниження

електроспоживання особливо актуальні в роботизованих комплексах, що рухаються, з живленням від хімічних джерел струму або від сонячних батарей. Цей модуль ідеально підійде до вирішення задачі з обробки інформації з мікроконтролера до вебдодатку на основі хмарного вебсервісу AWS.

Датчик LM393 здатний визначати відстань до об'єктів та перешкод, а також рівень освітленості в Lux. До складу модуля датчика входять: ІЧ-світлодіод з програмованим драйвером, два фотодіоди для визначення загальної освітленості (Ch0) та освітленості в ІЧ-діапазоні (Ch1), підсилювачі з програмованим коефіцієнтом посилення, МК, АЛУ, АЦП, ОЗУ, контролер шини I2C (контакти SDA и SCL). Датчик освітленості та наближення може використовуватися в Arduino проектах для реалізації механізмів зміни яскравості екранів залежно від освітленості або наближення-видалення предметів.

DHT11 – це цифровий датчик вологості та температури, що складається з термістора та ємнісного датчика вологості. Також датчик містить у собі АЦП для перетворення аналогових значень вологості та температури. Датчик DHT11 не має високу швидкодію і точність, зате простий, недорогий і відмінно підходить для навчання і контролю вологості в приміщенні.

Датчик KY-037 відрізняється високою енергоефективністю в режимі роботи (споживає від 3,1 мА до 6 мА) та автоматичним регулюванням шумів. Остання перевага добре проявляє себе у роботі з прикордонними в діапазоні шумів звуками (тихі звуки посилюються, а гучні – навпаки). Вбудований електричний мікрофон чутливий до звуків у діапазоні частот від 22 Гц до 22 кГц. Стандартне підключення модуля до контролера (наприклад, плати Arduino) виконує за трьома контактами: Vdd (“+”), GND (“-”), OUT (до аналогового входу). За допомогою контакту GAIN можна контролювати значення максимального посилення гучності (на 40 дБ, 50 дБ, або на 60 дБ). Контакт AR застосовується для налаштування часу спрацьовування.

Датчик вібрації SW-420 призначений для фіксації вібрацій у системах на основі Arduino. Модуль знаходить широке застосування від протиугінних систем до створення детекторів землетрусів. Датчик вібрацій SW-420 реагує на удари та вібрації. Чутливість модуля можна налаштувати за допомогою підстроювального резистора. Датчик має невеликі розміри, але укомплектований двома світлодіодами,

один з яких дозволяє контролювати наявність живлення, а другий вмикається під час спрацьовування. На відміну від інших аналогів цей датчик спрацьовує не від постійної вібрації – достатньо одного поштовху.

На основі вище описаних датчиків буде виконуватись рішення задачі для автоматизації обробки даних обраних до задачі показників і створення системи, яка буде працювати на основі мікроконтролера Arduino і будуватимуться інформативні графіки у реальному часі на основі вихідних даних [20]. Зображення буде показуватись у вікнах програмного забезпечення, яке у свою чергу буде розгорнуто через вебсервіс AWS. Моніторинг буде доступний через вебверсію у захищеному середовищі хмарного провайдера то його не треба буде встановлювати на нові комп'ютери або інші цифрові прилади, достатньо буде лише мати доступ до вебверсії та зручно користуватися ефективною системою, яка допоможе підприємству слідкувати за власними важливими робочими процесами.

Завдяки хмарним обчисленням вам не доведеться наперед виділяти зайві ресурси, щоб в майбутньому впоратися з піковими рівнями ділової активності. Замість цього надається необхідна кількість ресурсів. Є можливість збільшити або зменшити ці ресурси, щоб миттєво збільшувати та зменшувати потужність у міру зміни потреб бізнесу підприємства. Наприклад, контейнерні та оркестрові послуги AWS полегшують керування базовою інфраструктурою, як локально, так і в хмарі. Це дозволяє зосередитися на інноваціях та потребах свого бізнесу. Майже 80 відсотків усіх контейнерів у хмарі сьогодні працюють на AWS. Такі клієнти, як Samsung, Expedia, GoDaddy і Snap, вибирають запускати свої контейнери на AWS для безпеки, надійності та масштабованості.

AWS IoT дозволяє вибрати найбільш відповідні та сучасні технології для вашого рішення. Щоб допомогти вам керувати своїми пристроями IoT і підтримувати їх у польових умовах, AWS IoT Core підтримує такі протоколи:

- MQTT (черга повідомлень і транспортування телеметрії);
- MQTT через WSS (Websockets Secure);
- HTTPS (протокол передачі гіпертексту – безпечний);
- LoRaWAN (глобальна мережа дальнього дії).

На рис.3.3 можна ознайомитися із схемою роботи мікроконтролера Arduino у

хмарній системі обчислень з використанням сервісів платформи AWS та стороннім сервісом моніторингу Grafana.

Grafana допоможе візуалізувати метрики, які буде віддавати система під час роботи і з допомогою цього сервісу можна буде комфортніше і ефективніше давати оцінку роботі системи [21]. Також він якісно інтегрований з платформою AWS, що робить використання Grafana дієвим способом реалізації системи моніторингу.

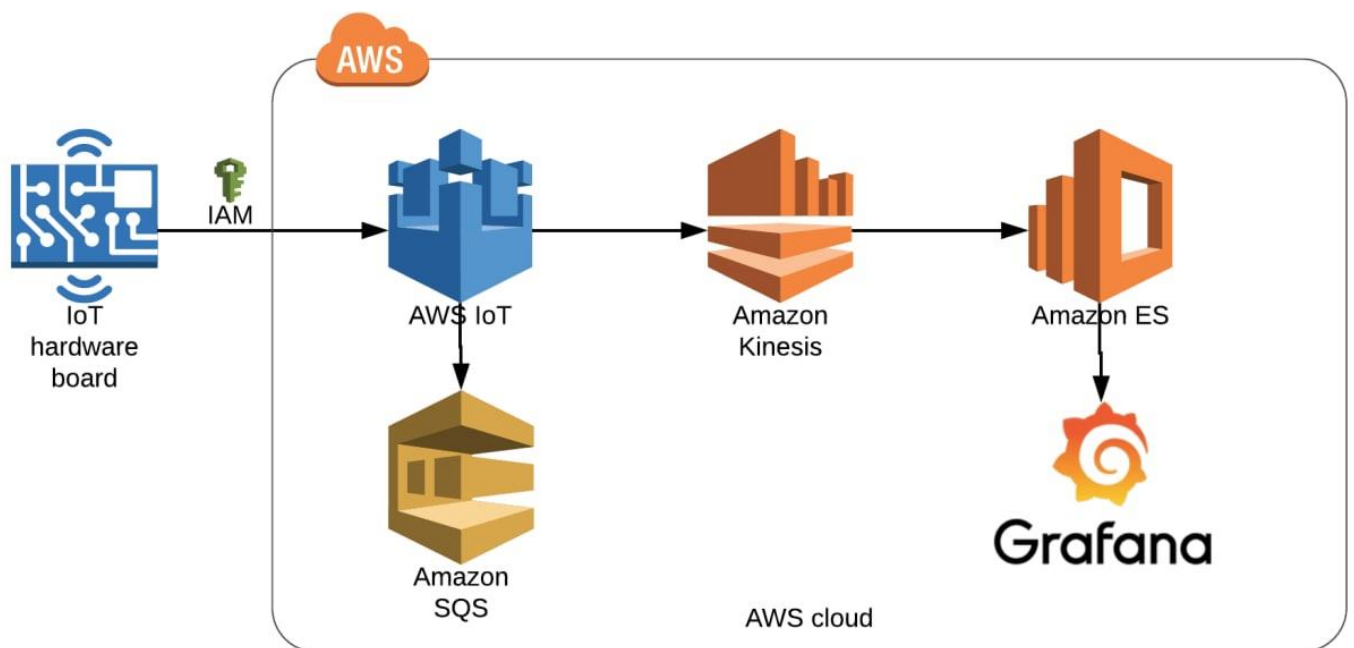


Рисунок 2.3 – Загальна схема роботи автоматизованої системи моніторингу

2.3 Проектування програмного забезпечення та формування необхідних засобів для обробки даних

За останні роки Python став однією з найпопулярніших мов програмування у світі. Його використовують у всьому, від машинного навчання до створення вебсайтів і тестування програмного забезпечення. Його можуть використовувати як розробники, так і не розробники. Python, одна з найпопулярніших мов програмування у світі, створила все, від алгоритму рекомендацій Netflix до програмного забезпечення, яке керує безпілотними автомобілями. Python – це мова загального призначення, що означає, що її розроблено для використання в низці

додатків, включаючи науку про дані, програмне забезпечення та веброзробку, автоматизацію та загалом виконання завдань.

Python часто використовується для розробки задньої частини вебсайту чи програми – частин, які користувач не бачить. Роль Python у веброзробці може включати надсилання даних на сервери та з них, обробку даних і зв'язок із базами даних, маршрутизацію URL-адрес і забезпечення безпеки. Python пропонує кілька фреймворків для веброзробки. Найбільш часто використовувані включають Django і Flask. Web Application Framework або просто Web Framework представляє набір бібліотек і модулів, які дозволяють розробникам вебдодатків писати програми, не турбуючись про деталі низького рівня, такі як протокол, керування потоками тощо [22].

Flask дозволяє легко розробляти вебдодатки. Він має невелике ядро, яке легко розширити: це мікрофреймворк, який не містить ORM (Object Relational Manager) або подібних функцій. Він має багато цікавих функцій, таких як маршрутизація URL-адреси, система шаблонів [23]. Це мікрофреймворк, але це не означає, що весь ваш додаток має бути в одному файлі Python. Ви можете і повинні використовувати багато файлів для більших програм, щоб впоратися зі складністю. Micro означає, що фреймворк Flask є простим, але розширюваним. Flask надає змогу приймати всі рішення: яку базу даних використовувати, використання ORM тощо, Flask доволі гнучкий. Flask є одним із найпопулярніших вебфреймворків, що означає, що він актуальний і сучасний. Ви можете легко розширити його функціональність. Ви можете масштабувати його для складних програм.

2.4 Аналіз особливостей розгортання додатку у AWS

Звісно можна було б обійтися і без складної системи і обрати прилади заміру показників, які б одразу давали інформацію. А після того як би людина отримувала інформацію про показники – записувала у звичайний журнал ручкою і той журнал треба було вести кожен день у ручному режимі. Звісно це простіше і можливо навіть дешевше, але тут треба ще враховувати наявність людського фактору, який буде відображатися у неповному радіусі та неточності рахувань, так як людина не може

фізично поміститися туди, куди зможе електричний сенсор. Ведення журналу вручну через свою незручність буде мати наслідки у передачі та обміну інформацією, а також у можливості її резервного копіювання.

Таким чином, стає актуальною створення автоматизованої системи моніторингу, яка дозволить вести контроль без участі людини, а уся інформація буде збиратися в цифровому вигляді, що спростить трансфер, обробку, зберігання та взаємодію з даними через хмарний вебсервіс.

2.4.1 AWS IoT

AWS IoT Core – це хмарна платформа, яка дозволяє підключати пристрої в хмарі сервісів AWS. AWS IoT надає інтерфейс, який дозволяє пристроям безпечно та надійно обмінюватися двонаправленим зв'язком із точками дотику AWS, навіть коли пристрої перебувають у режимі офлайн.

AWS пропонує послуги та рішення Інтернету речей (IoT) для підключення та керування мільярдами пристроїв. Ці хмарні служби з'єднують ваші пристрої IoT з іншими пристроями та хмарними службами AWS. AWS IoT надає програмне забезпечення для пристроїв, яке може допомогти вам інтегрувати ваші IoT-пристрої в рішення на основі AWS IoT. Якщо ваші пристрої можуть підключатися до AWS IoT, AWS IoT може підключати їх до хмарних служб, які надає AWS – S3 Bucket's, Lambda, DynamoDB та інші (рис. 2.4).

Amazon DynamoDB – це повністю керована безсерверна база даних NoSQL із ключем-значенням, розроблена для запуску високопродуктивних програм у будь-якому масштабі.

DynamoDB може керувати структурованими або напівструктурованими даними, включаючи документи JSON. SQL є стандартом для зберігання та отримання даних. Реляційні бази даних пропонують багатий набір інструментів для спрощення розробки програм, керованих базами даних, але всі ці інструменти використовують SQL.

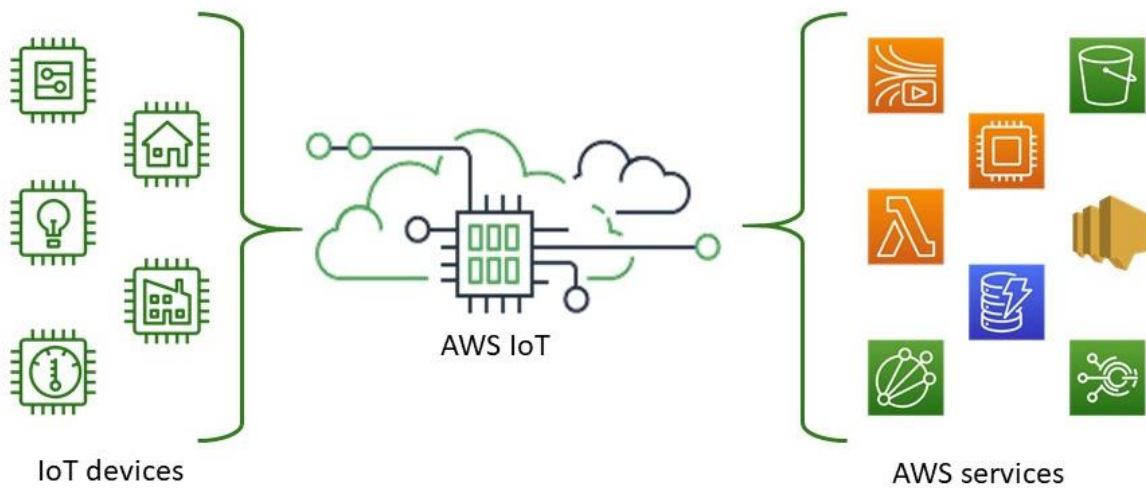


Рисунок 2.4 – Схема роботи AWS IoT

AWS IoT дозволяє вибрати найбільш відповідні та сучасні технології для вашого рішення. Щоб допомогти вам керувати своїми пристроями IoT і підтримувати їх.

Брокер повідомлень AWS IoT Core підтримує пристрої та клієнти, які використовують протоколи MQTT і MQTT через WSS для публікації та підписки на повідомлення. Він також підтримує пристрої та клієнти, які використовують протокол HTTPS для публікації повідомлень.

AWS IoT Core для LoRaWAN допомагає підключати та керувати бездротовими пристроями LoRaWAN (малопотужна глобальна мережа великого радіусу дії). AWS IoT Core для LoRaWAN замінює вам потребу в розробці та експлуатації мережевого сервера LoRaWAN (LNS).

Якщо підприємству не потрібні такі функції AWS IoT, як зв'язок пристроїв, правила або завдання, то можна переглянути AWS Messaging, щоб отримати інформацію про інші служби обміну повідомленнями AWS IoT, які можуть краще відповідати вимогам підприємства.

Сервіси AWS IoT Core підключають пристрої IoT до сервісів AWS IoT та інших сервісів AWS. AWS IoT Core включає шлюз пристрою та брокер повідомлень, які з'єднують і обробляють повідомлення між вашими пристроями IoT і хмарою (рис 2.5).

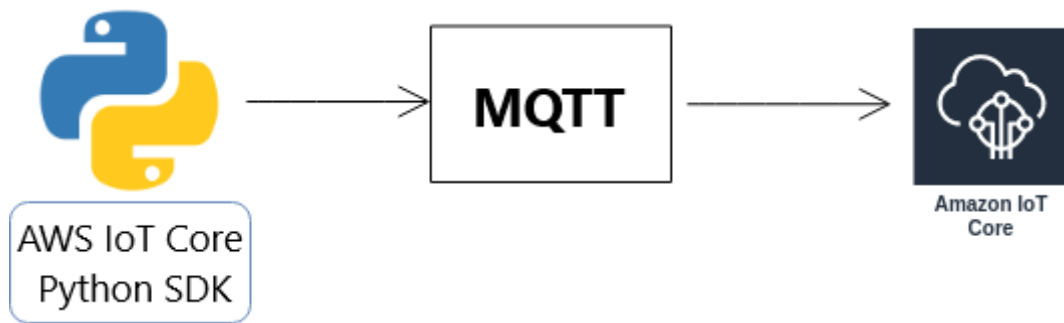


Рисунок 2.5 – Схема публікування MQTT повідомлень в AWS IoT CORE за допомогою Python

Переваги розробки та розгортання змін за допомогою технології AWS IoT досягається тому, що AWS IoT Core – це керований хмарний сервіс, який дозволяє підключеним пристроям легко та безпечно взаємодіяти з хмарними програмами та іншими пристроями. AWS IoT Core може підтримувати мільярди пристроїв і трильйони повідомлень, а також може надійно та безпечно обробляти та направляти ці повідомлення до кінцевих точок AWS та інших пристроїв.

MQTT (телеметричний транспорт черги повідомлень) – це легкий і широко поширений протокол обміну повідомленнями, розроблений для обмежених пристроїв [24]. Підтримка AWS IoT Core для MQTT базується на специфікаціях MQTT v3.1.1 і MQTT v5.0 з деякими відмінностями, як зазначено в відмінностях AWS IoT від специфікацій MQTT. Як остання версія стандарту, MQTT 5 представляє кілька ключових функцій, які роблять систему на основі MQTT більш надійною, включаючи нові вдосконалення масштабованості, покращене звітування про помилки з відповідями коду причини, таймери закінчення повідомлень і сеансу та спеціальні заголовки повідомлень користувача. Для отримання додаткової інформації про функції MQTT 5, які підтримує AWS IoT Core. AWS IoT Core також підтримує перехресну версію MQTT (MQTT 3 і MQTT 5). Видавець MQTT 3 може надіслати повідомлення MQTT 3 передплатнику MQTT 5, який отримуватиме повідомлення про публікацію MQTT 5, і навпаки.

AWS IoT Core підтримує підключення пристроїв, які використовують протокол MQTT і MQTT через протокол WSS і ідентифікуються за ідентифікатором

клієнта. Пакети SDK пристроїв AWS IoT підтримують обидва протоколи та є рекомендованими способами підключення пристроїв до AWS IoT Core. Пакети SDK пристроїв AWS IoT підтримують функції, необхідні для підключення пристроїв і клієнтів до послуг AWS IoT і доступу до них. Пакети SDK пристрою підтримують протоколи автентифікації, які вимагають служби AWS IoT, і вимоги до ідентифікатора з'єднання, які вимагають протоколи MQTT і MQTT через протоколи WSS.

Рекомендується використовувати AWS IoT Device SDK для підключення до AWS IoT, хоча це не є обов'язковим рішенням. Проте, якщо не використовується SDK пристроїв AWS IoT, ви повинні забезпечити необхідну безпеку з'єднання та зв'язку. Клієнти повинні надіслати розширення TLS індикації імені сервера (SNI) у запиті на підключення. Спроби підключення, які не містять SNI, відхиляються. Додаткову інформацію див. у розділі Безпека транспорту в AWS IoT. Клієнти, які використовують користувачів IAM та облікові дані AWS для автентифікації клієнтів, повинні надати правильну автентифікацію підпису четвертої версії (рис. 2.6).

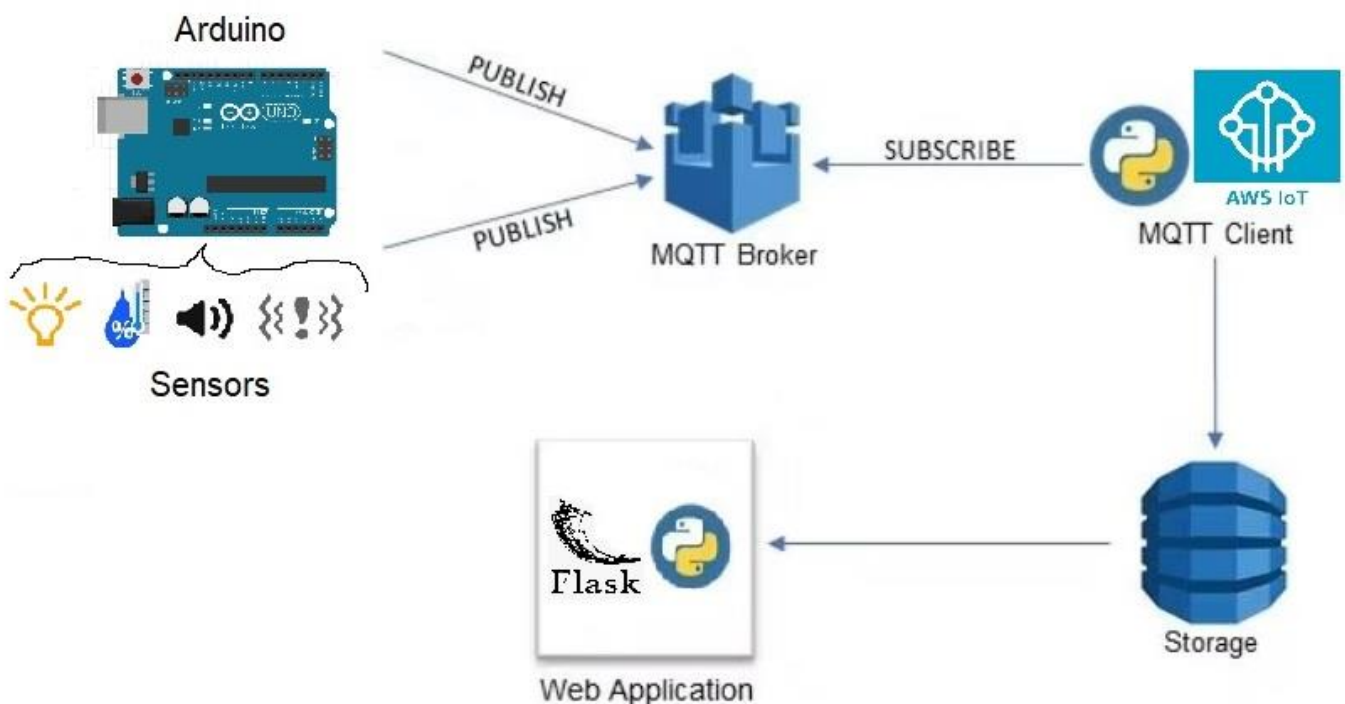


Рисунок 2.6 – Схема роботи системи IoT за допомогою технології MQTT

2.5 Висновки до другого розділу

У даному розділі була виконана робота по проектуванню структури та розробки автоматизованої системи моніторингу. Відповідно була зпроектована хмарна система Інтернету речей, яка збирає інформацію з набору датчиків навколишнього середовища через платформу Arduino за допомогою протоколу MQTT і за допомогою технології AWS IoT. Буде створений вебдодаток (Flask в якості вебфрейворка і логіка на Python) для відображення даних, зібраних датчиками.

Отримані значення публікуються в темі через канал MQTT у вигляді файлів JSON. Від брокера буде ще один скрипт Python (у ролі підписника), який вставляє отримані дані в таблицю EnvironmentalStationDB в системі DynamoDB. Записи з бази даних відображаються на вебсайті. Усі вхідні і вихідні дані будуть інтегровані з хмарним середовищем AWS IoT.

Також як практичний приклад використання інфраструктури AWS і, як наслідок, довіри серед державних органів, можна розглянути як 15 липня, у Лондоні, міністр цифрової трансформації Михайло Федоров та Директор з цифрових трансформацій у державному секторі Amazon Web Services (AWS) Ліам Максвелл підписали Меморандум про взаєморозуміння щодо прискорення цифрової трансформації та впроваджень інновацій в Україні [25]. «Протягом багатьох років Amazon (AWS) мав сильну, віддану і зростаючу спільноту клієнтів в Україні. Ми вітаємо підписання цього Меморандуму про взаєморозуміння з українським Урядом і сподіваємося на спільну роботу в напрямі цифрового розвитку держави», – Ліам Максвелл, Директор з цифрових трансформацій у державному секторі, Amazon Web Services в Європі.

3 РОЗРОБКА І РОЗГОРТАННЯ ПРОГРАМНОГО ТА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Створення загальної архітектури автоматизованої системи моніторингу

Були розглянуті різні типи систем Інтернету речей, які пропонуються або впроваджуються в різних середовищах моніторингу в реальному часі. Були виділені рішення на основі хмарних обчислень їхні спільні риси. Були описані ключові складові системи IoT і пояснили, як вони працюють разом для моніторингу середовища в реальному часі. Необхідна формалізація їх на різних рівнях, щоб сформуванати загальну архітектуру системи моніторингу в реальному часі на базі IoT. На виході загальна архітектура буде складатися з чотирьох рівнів: рівня сприйняття, рівня шлюзу та зв'язку, рівня хмарних обчислень та обслуговування та рівня додатків.

Наразі пропонується загальна архітектура, яка може служити стандартною архітектурою для будь-якого типу рішення моніторингу важливих показників на виробництві. Якщо зобразити це візуально, то вийде наступна схема (рис. 3.1):

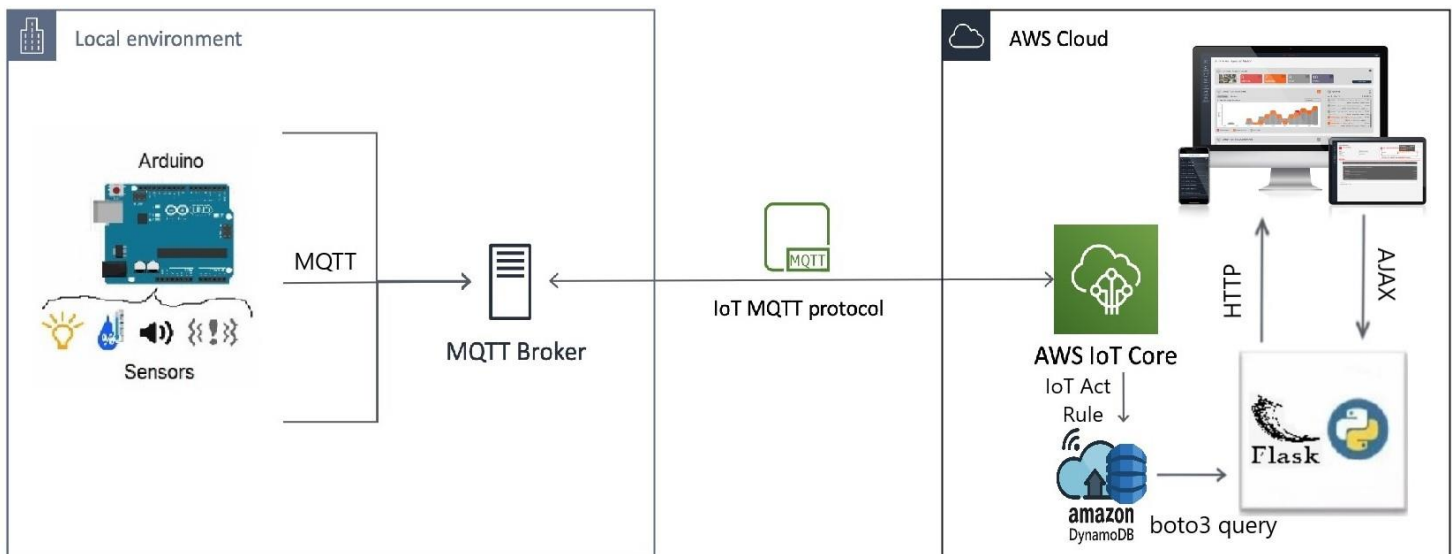


Рисунок 3.1 – Схема архітектури автоматизованої системи моніторингу

На схемі архітектури автоматизованої системи моніторингу продемонстровано

її загальну роботу та інфраструктуру. Є сектор з апаратним забезпеченням, де задіяні сенсори та апаратно-програмна платформа Arduino, завдяки IDE якої є можливість запрограмувати сенсори і відправляти дані через Wi-Fi модуль.

Далі буде використовуватися MQTT брокер повідомлень, який через протокол MQTT буде обробляти дані з платформи Arduino і відправляти з створеного апаратного комплексу до віртуального середовища у AWS, з поєднанням AWS IoT Core.

У створеній хмарі будується логіка вебдодатку, що отримує інформацію з сенсорів і будує графіки та менеджмент. Рушієм є програма на мові Python з використанням фреймворку Flask, яка інтегрована у AWS IoT Core. Для більш стабільної роботи системи буде використовуватися база даних DynamoDB. Її використання придасть низький час відгуку запитів метрик, що зробить систему більш ефективною.

3.2 Розробка комплексу апаратного забезпечення на платформі Arduino

3.2.1 Під'єднання компонентів до Arduino

За основу апаратно-програмного комплексу обробки інформації було взято мікропроцесор ATmega328P на платформі Arduino UNO [26]. Плата оснащена наборами контактів цифрового та аналогового входу/виводу (I/O), які можна підключати до різних плат розширення (щитів) та інших схем. Плата має 14 цифрових контактів вводу/виводу (шість із можливістю виведення ШІМ), 6 аналогових контактів вводу/виводу та програмується за допомогою Arduino IDE (інтегрованого середовища розробки) через кабель USB типу B. Його можна живити від USB-кабелю або зовнішньої 9 В батареї, хоча він приймає напругу від 7 В до 20 В. Він схожий на Arduino Nano і Leonardo. Еталонний дизайн апаратного забезпечення поширюється за ліцензією Creative Commons Attribution Share-Alike 2.5 і доступний на вебсайті Arduino.

До Arduino UNO будуть під'єднанні датчики, які зможуть забезпечити необхідними зовнішніми показниками системи моніторингу, а саме це – датчик

освітлення (модуль на базі компаратора LM393), датчик температури та вологості (DHT11 другої версії) та датчик вібрації (модуль SW-420 датчик вібрації нормально-замкнутий). Також хочу зазначити, що у цьому списку немає датчику шуму (планувалося використовувати модуль мікрофонного підсилювача KY-037) тому, що його не було у наявності. У якості Wi-Fi модулю буде використовуватися модуль Wi-Fi ESP8266/ESP-01 з послідовним інтерфейсом.

Апаратний комплекс виходить немаленьким, тому будуть наведені схеми під'єднання кожного елемента окремо. Почнемо з модулю Wi-Fi ESP8266 / ESP-01, з схемою підключення до Arduino UNO якого можна ознайомитися на рисунку 3.2:

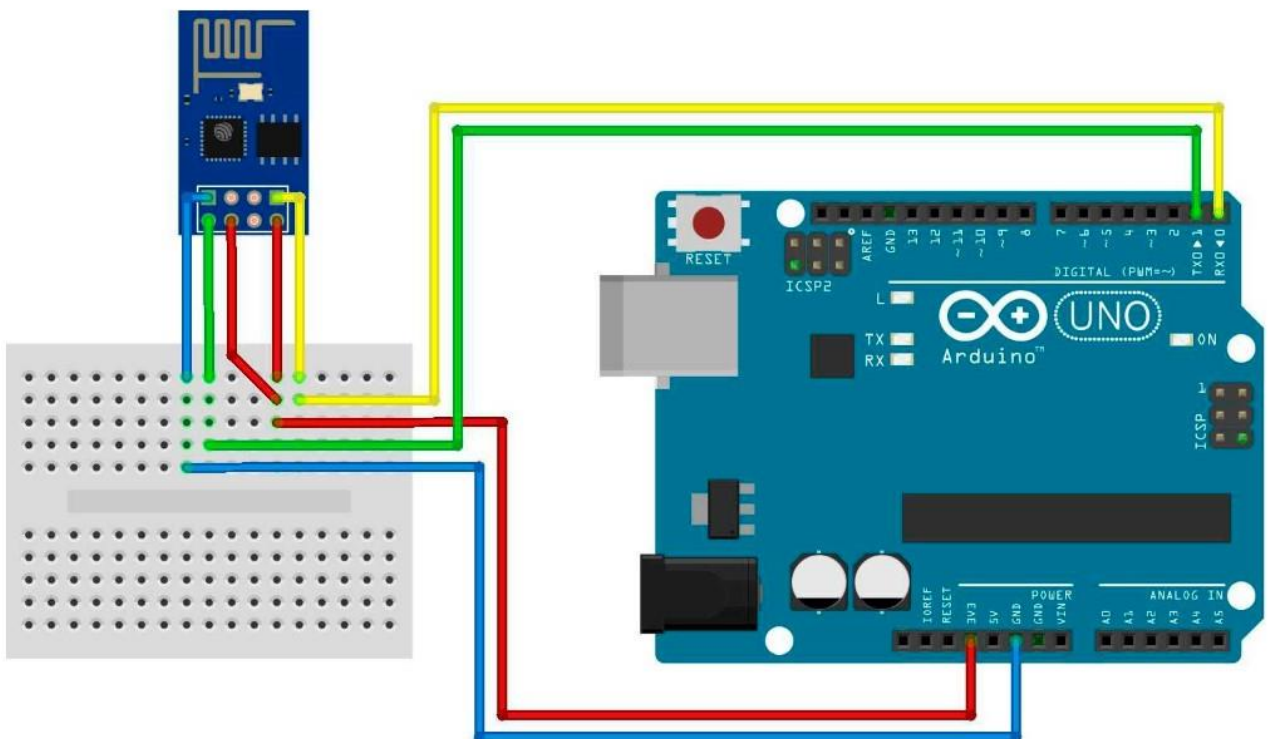


Рисунок 3.2 – Схема підключення Wi-Fi модулю ESP-01

VCC і GND [26] – це живлення модулю. Модуль працює від 3,3 В. Контакт RX підключимо до піну 0 на Arduino. А контакт TX до 1 пін. Також треба перевірити контакти живлення та землі перед включенням. Після підключення живлення на модулі повинна загорітися червона лампочка.

Далі йде підключення датчику температури та вологості DHT11 другої версії, схема представлена на рисунку 3.3:

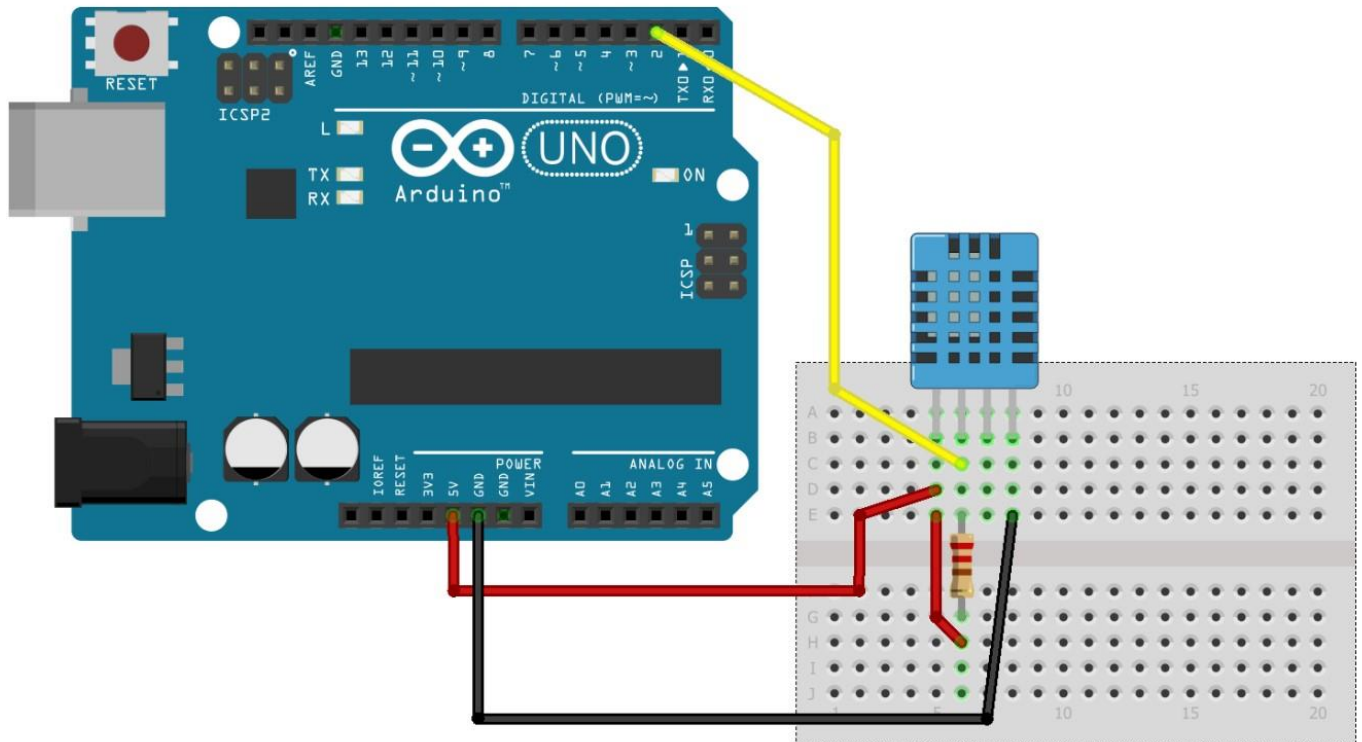


Рисунок 3.3 – Схема підключення модулю DHT11 (v2)

Для під'єднання необхідні резистор 10K і перемички. У нашому випадку використовується 4-х контактна модель, яка під'єднується контактом (жовтий) DHT11 до 3 В Arduino UNO, резистор 10K підключається між контактами 1 і контактами 2 DHT 11, контакт 3 датчика DHT 11 залишається не підключеним і контакт (чорний) DHT 11 підключається до GND на Arduino UNO.

Далі підключається модуль SW-420, який буде служити датчиком вібрації для апаратного комплексу системи моніторингу.

Модуль датчика вібрації SW-420 інтегрований з датчиком вібрації SW-420, компаратором напруги LM393, потенціометром, резисторами обмеження струму, які діють як дільники напруги, отже, контролюють струм і конденсатори як елементи зміщення та для фільтрації шуму. Схема під'єднання до Arduino UNO представлена на рисунку 3.4:

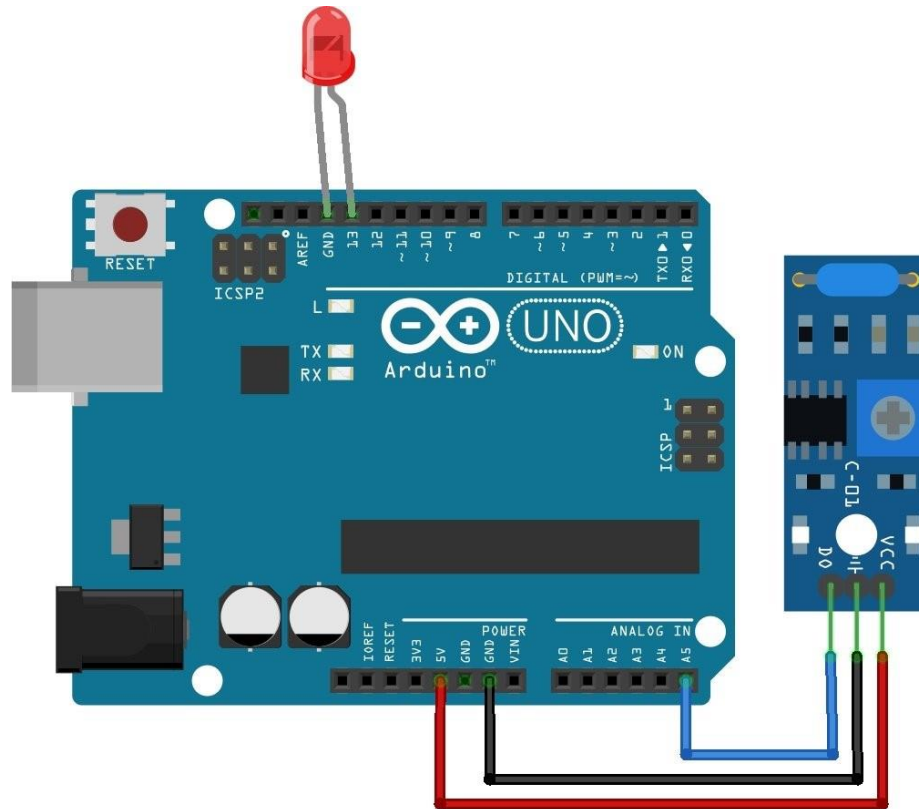


Рисунок 3.4 – Схема підключення модулю вібрації SW-420

Для регулювання чутливості датчика на модулі розміщений потенціометр 10 К. Чутливість можна збільшити або зменшити залежно від вимоги. Це робиться шляхом встановлення попереднього або порогового значення, яке надається LM393 як еталон для порівняння.

Вібраційний перемикач SW-420 визначає величину вібрації в навколишньому середовищі. Він реагує на вібрацію через відкриття або закриття електричного контакту. Тригерний перемикач може бути електромеханічним, релейним або напівпровідниковим компонентом.

Модуль має два світлодіоди. Один з них загоряється, коли модуль подається під напругу, а інший – для вказівки на цифровий вихід. Також для показовості роботи апаратного комплексу був під'єднаний звичайний світлодіод для перевірки працездатності.

Далі буде продемонстровано останній елемент під'єднання до Arduino UNO – це модуль на базі компаратора LM393, що служить датчиком освітлення. Схема під'єднання представлена на рисунку 3.5:

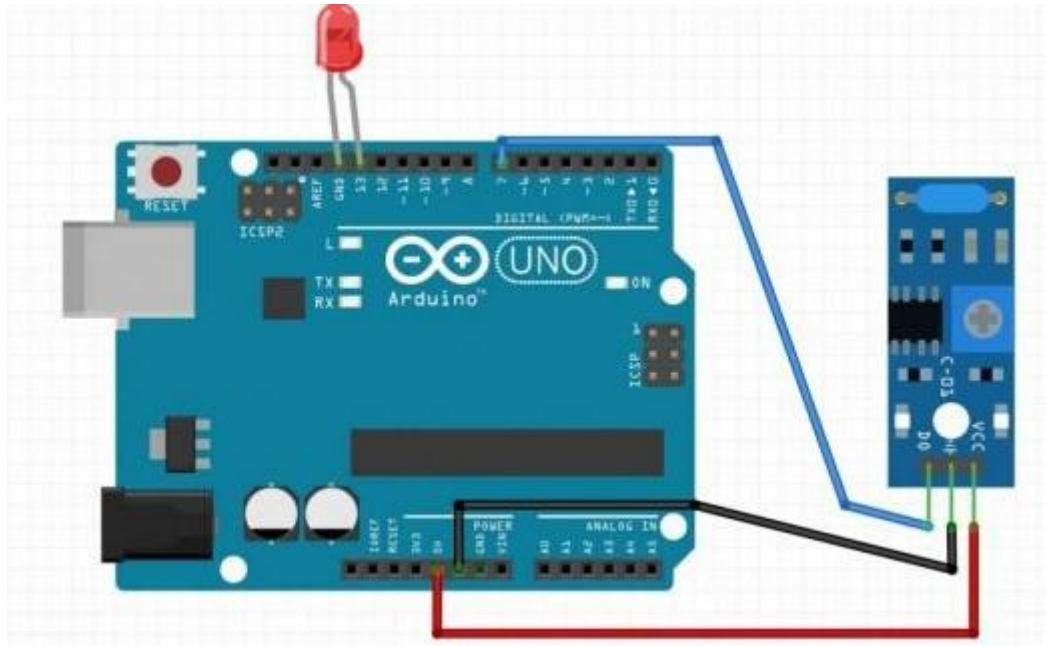


Рисунок 3.5 – Схема підключення модулю освітлення LM393

Цей модуль має високочутливий датчик вібрації та вбудований компаратор напруги для створення цифрового вихідного сигналу. Коли вібраційний перемикач знаходиться в закритому стані провідності, вихідний сигнал низький і зелене світло горить. Коли вібраційний вимикач від'єднано, вихідний сигнал високий, а зелене світло вимкнено. Його можна безпосередньо підключити до мікроконтролера для зчитування вихідного рівня та легкого визначення стану датчика. Вихід датчика вказує, чи була виявлена вібрація в навколишньому середовищі.

Побачити всі елементи Arduino UNO, що були під'єднанні по схемам і інструкціям описаним вище можна на рисунку 3.6:

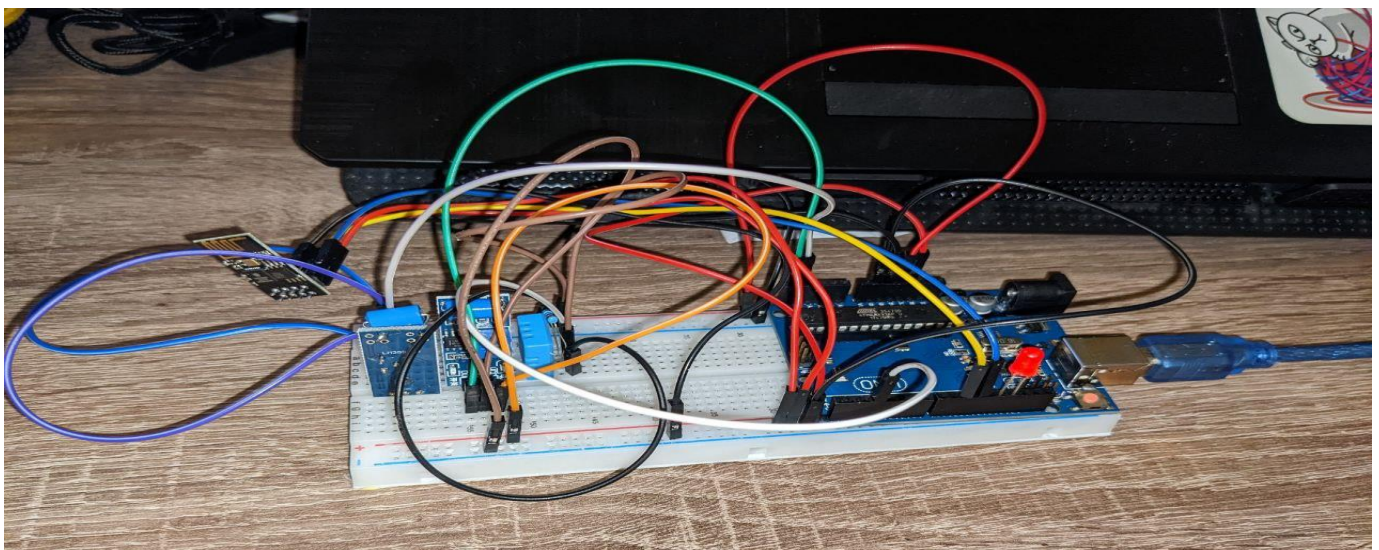
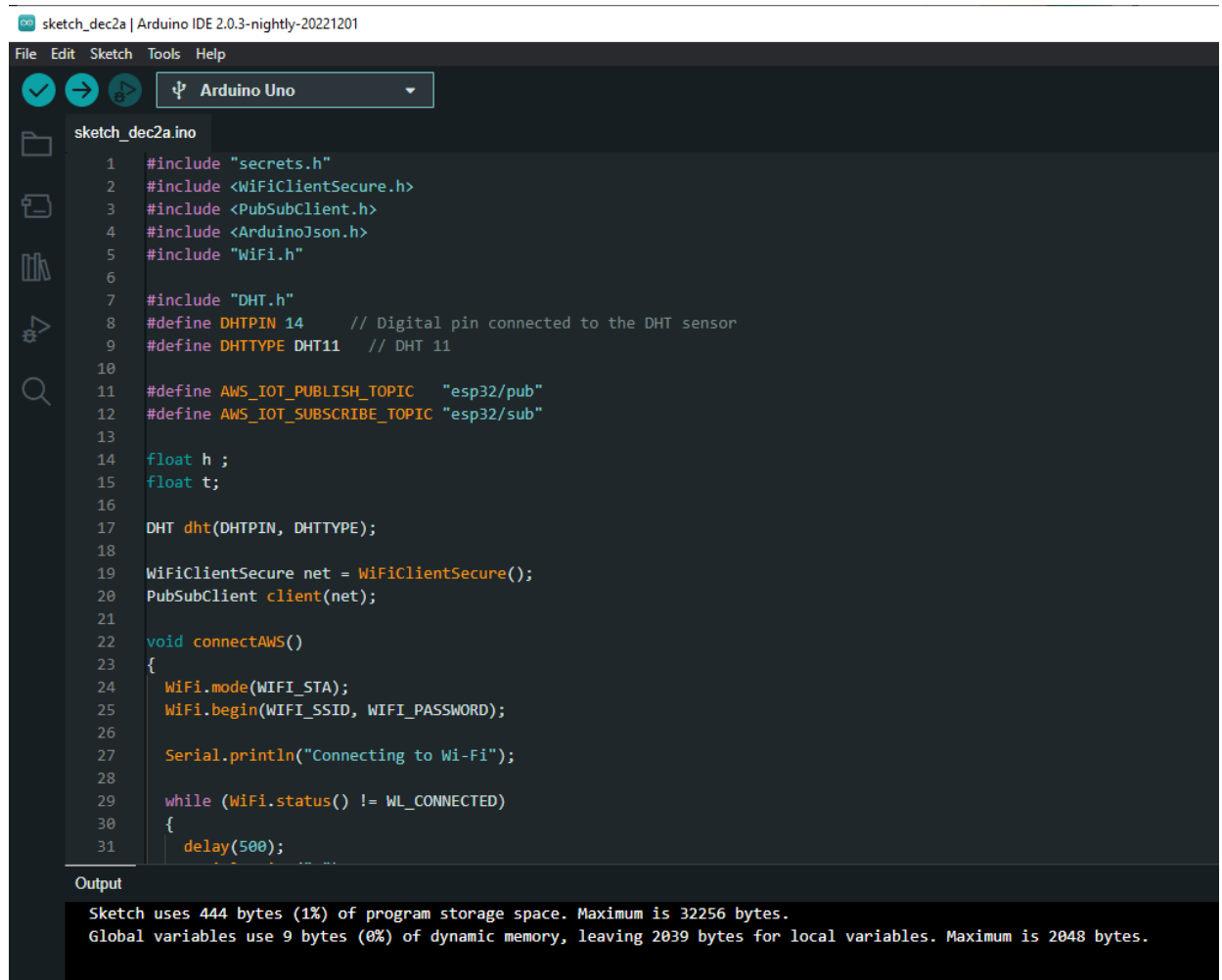


Рисунок 3.6 – Демонстрація під'єднання всіх елементів Arduino

3.2.2 Середовище розробки апаратного забезпечення і поєднання з AWS IoT

Після підключення всіх елементів та самого мікроконтролера до ПК, необхідно загрузити програму до Arduino. Це було зроблено через офіційну середу розробки Arduino (рис. 3.7).



```
sketch_dec2a.ino
1  #include "secrets.h"
2  #include <WiFiClientSecure.h>
3  #include <PubSubClient.h>
4  #include <ArduinoJson.h>
5  #include "WiFi.h"
6
7  #include "DHT.h"
8  #define DHTPIN 14 // Digital pin connected to the DHT sensor
9  #define DHTTYPE DHT11 // DHT 11
10
11 #define AWS_IOT_PUBLISH_TOPIC "esp32/pub"
12 #define AWS_IOT_SUBSCRIBE_TOPIC "esp32/sub"
13
14 float h ;
15 float t;
16
17 DHT dht(DHTPIN, DHTTYPE);
18
19 WiFiClientSecure net = WiFiClientSecure();
20 PubSubClient client(net);
21
22 void connectAWS()
23 {
24     WiFi.mode(WIFI_STA);
25     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
26
27     Serial.println("Connecting to Wi-Fi");
28
29     while (WiFi.status() != WL_CONNECTED)
30     {
31         delay(500);
```

Output

Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

Рисунок 3.7 – Робоча програма у IDE Arduino

Щоб отримати дані з датчика, була використана бібліотека Adafruit DHT, відповідно до своїх потреб системи моніторингу. Інші модулі працювали без особливих налаштувань.

Далі необхідно налаштувати відправлення та обробку повідомлень до MQTT брокеру повідомлень. Щоб налаштувати наш брокер MQTT для датчиків, потрібно зробити налаштування пристроїв у IoT Core, дотримуючись інструкцій на інформаційній панелі AWS IoT (рис.3.8).

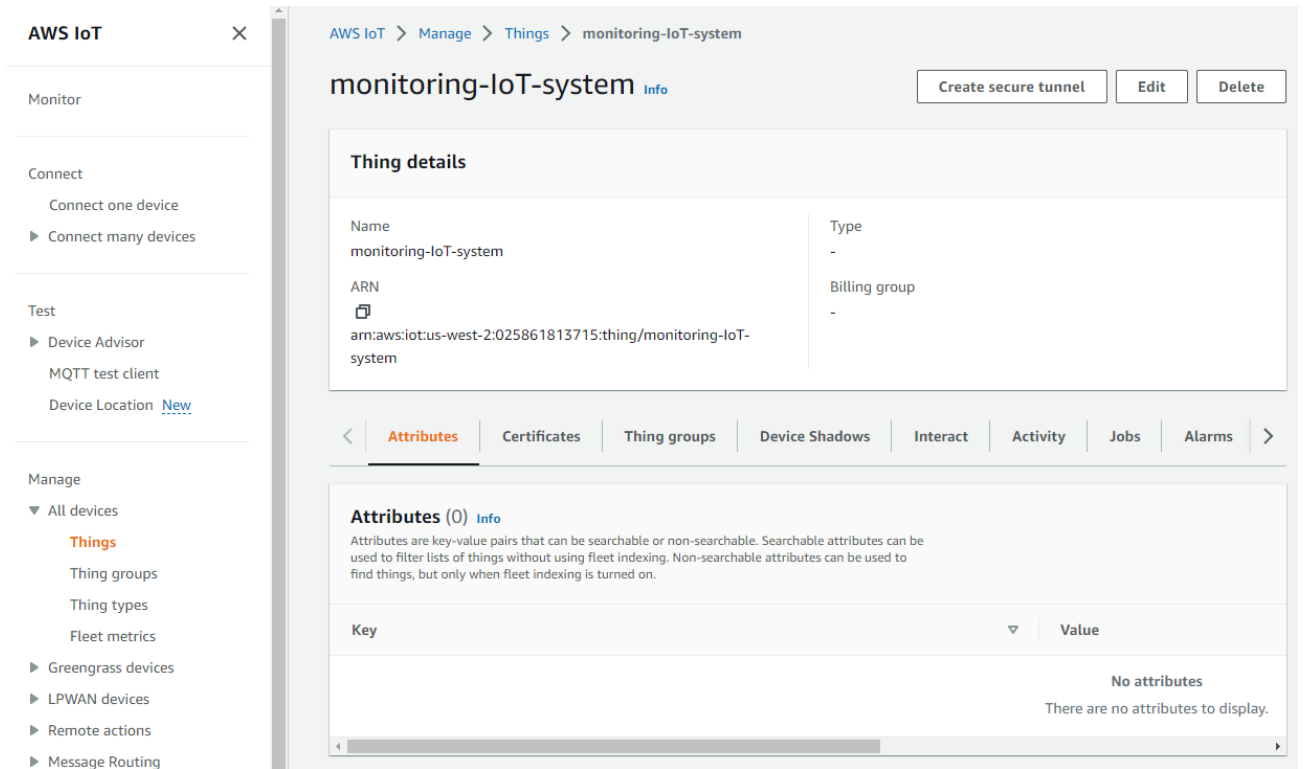


Рисунок 3.8 – Інформаційна панель AWS IoT Core

Щоб взаємодіяти з пристроями, потрібно буде створити нову політику, вибравши `Secure > Policy > New`. Ця політика повинна мати доступ IoT і посилатися на ARN – імена ресурсів Amazon унікально ідентифікують ресурси AWS. Коли буде потрібно однозначно вказати ресурс у всіх AWS, наприклад у політиках IAM, тегах і викликах API, використовується ARN. На рисунку 3.9 була створена проста політика взаємодії між пристроями:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:*",
      "Resource": "*"
    }
  ]
}

```

Рисунок 3.9 – Політика взаємодії AWS IoT Core

Але було б ще краще обмежити ресурси та дії, якщо це можливо, наприклад, вказавши “Thing ARN” у полі ресурсів. Необхідно завантажити приватний і публічний ключі. Тепер, щоб використовувати політику, нам потрібно створити сертифікат і приєднати його до політики, до AWS IoT. Сертифікат має бути активним, щоб пристрій міг підключитися до AWS IoT. Наведені файли використовуються для підключення до брокера AWS MQTT за допомогою сертифікатів TLS. На рисунку 3.10 можна побачити поле для завантаження сертифікатів:

Device certificate
cbbfc1885d8...te.pem.crt

Deactivate certificate Download

Key files
The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

⚠ This is the only time you can download the key files for this certificate.

Public key file
cbbfc1885d85b0890ffe898...d4997e6-public.pem.key

Download
Key downloaded

Private key file
cbbfc1885d85b0890ffe898...4997e6-private.pem.key

Download
Key downloaded

Root CA certificates
Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1

Download

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3

Download

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

Рисунок 3.10 – Поле завантаження сертифікатів і ключів для AWS IoT

Наступним кроком є налаштування Wi-Fi та частину взаємодії з AWS, оскільки підтримки ESP8266 на Arduino і AWS для Arduino є частковими. Тому були ризики зіткнутися зі зламаними бібліотеками. Було прийнято рішення підключитися вручну брокеру MQTT із сертифікатами клієнта. Було інтегровано наведений

фрагмент із бібліотеки на ресурсі GitHub ESP8266 безпечно з'єднання MQTT з автентифікацією сертифіката клієнта, завантажив сертифікати у флешпам'ять за допомогою плагіну із ресурсів GitHub [27] завантажувач файлової системи Arduino ESP8266.

Слід пам'ятати про те, що нам потрібно жорстко закодувати відбиток TLS-сертифіката сервера в ESP. Це робиться, щоб запобігти атакам типу “man-in-the-middle” (людина посередині), і зазвичай це наполегливо обробляється комп'ютерами, тобто перший відбиток пальця зберігається, а наступні мають збігатися з ним. Захист від атак типу MITM вимагає кількох дій, кожна з них має важливе значення. Не дозволяйте комп'ютерам або мобільним пристроям автоматично підключатися до Wi-Fi мереж, переконайтеся, що вони підключаються тільки до відомих та перевірених мереж Wi-Fi.

Далі йде інструкція і використання нереляційної бази даних DynamoDB (рис. 3.11). Щоб додати рівень стійкості була виконана спроба використати вбудовані функції Amazon, який дозволяє створювати тригери для отриманих повідомлень, які взаємодіятимуть з іншими службами AWS, серед яких є DynamoDB.

Insert a message into a DynamoDB table
DYNAMODB

The table must contain Partition and Sort keys.

*Table name
Measurements

*Partition key	*Partition key type	*Partition key value
sensor	STRING	\${topic(3)}
Sort key	Sort key type	Sort key value
timestamp	STRING	\${timestamp0}

Write message data to this column
message

Operation

Рисунок 3.11 – Створення таблиці DynamoDB у AWS

Спочатку були виконані операції з DynamoDB [28], створена нова таблиця з основним індексом (також званим індексом хешування) під назвою “сенсор” і вторинним (для сортування) індексом під назвою “мітка часу”.

Тоді настає необхідність в налаштуванні правила, це як окреме завдання, оскільки більшість помилок безперервно фіксуються, і з'ясувати, що не так, залишається на вміннях програміста вгадувати. Завдяки бібліотекам AWS вийшло реалізувати зпроектовану архітектуру системи. Amazon пропонує мову на основі SQL для аналізу json у вхідних повідомленнях MQTT, у цьому прикладі була використана бета-версію діалекту SQL, оскільки деякі функції, як topic(3), аналізують третій розділ URL-адреси теми (тобто назва датчика) - були недоступні або несправні в старих версіях.

Amazon DynamoDB – це повністю керована власна служба бази даних NoSQL, яка підтримує структури даних «ключ–значення» та документи та пропонується Amazon.com як частина портфоліо Amazon Web Services. DynamoDB надає подібну модель даних і походить від Dynamo, але має іншу базову реалізацію.

Таким чином, кортежі бази даних будуть упорядковані за датчиком і міткою часу, а повідомлення матиме на рисунку 3.12 наступну форму з поєднанням SQL-запиту:

SQL	JSON
<pre>SELECT timestamp() AS timestamp, topic(3) AS sensor, state.desired.temperature AS temperature, state.desired.humidity AS humidity, state.desired.vibration AS vibration, state.desired.noise AS noise, state.desired.light AS light FROM '\$aws/things/+/shadow/update/accepted'</pre>	<pre>{ "sensor" : { "S" : "test" }, "humidity" : { "N" : "73.886108" }, "vibration" : { "N" : "18.182373" }, "temperature" : { "N" : "-30.273438" }, "timestamp" : { "N" : "1585125773702" }, "noise" : { "N" : "82.2052" }, "light" : { "N" : "100.9643544" } }</pre>

Рисунок 3.12 – Опис SQL-запиту та залежностей JSON

3.3 Середовище та розробка вебдодатку на Flask та Python

За основу інтерфейсу було обрано Flask, фреймворк для Python. Щоб створити сервер, спочатку був створений клас для взаємодії з DynamoDB [28], і був використаний boto3 для цієї мети (рис. 3.13).

Однією з найбільших переваг Boto3 є те, що ця технологія автоматично шукає облікові дані AWS, згенеровані AWS-CLI, тому немає потреби явно надавати будь-який маркер доступу. Є можливість встановити інтерфейс командного рядка Amazon, запустити AWS Configure і вставити наш AWS ID/ключ і зона.

Використовується AWS SDK для Python (Boto3) для створення, налаштування та керування службами AWS, такими як AWS IoT. SDK надає об'єктно-орієнтований API, а також низькорівневий доступ до сервісів AWS. Дотримуючись матеріалів з офіційного джерела Boto3, було виявлено, як використовувати ресурси `DynamoDB.ServiceResource` і `DynamoDB.Table`, щоб створювати таблиці, записувати елементи в таблиці, змінювати існуючі елементи, отримувати елементи та запитувати/фільтрувати елементи в таблиці.

Boto3 поставляється з “очікувачами”, які автоматично опитують попередньо визначені зміни статусу в ресурсах AWS. Наприклад, ви можете запустити екземпляр Amazon EC2 і за допомогою очікування зачекати, поки він досягне стану «запущено», або ви можете створити нову таблицю Amazon DynamoDB і зачекати, доки вона стане доступною для використання. Boto3 має очікування для клієнтських і ресурсних API.

```
class DB:
    # Query db for the last 20 measures of a sensor
    def retrieve_values(sensor, metric=None):
        table = dbm.Table('Measurements')
        response = table.query(
            KeyConditionExpression=keyword('sensor').eq(sensor),
            ScanIndexForward=False,
            Limit=2)
        items = []

        for i in response[u'Items']:
            # Transforming timestamp to readable hour and minutes
            # in one line
            i['message']['timestamp'] = datetime.fromtimestamp(int(i['timestamp'][:-3])).strftime('%H:%M')

            if _MetricsDict is not None:
                #if metric is set return only that value
                i = i['message'][metric]
            else:
                # otherwise return all metrics for this sensor
                i = i['message']

            items.append(i)
        return ItemsView
```

Рисунок 3.13 – Таблиця Boto3 для взаємодії з DynamoDB

Вото3 має два різних рівні API. Клієнтські (або «низькопівневі») API забезпечують однозначне зіставлення базових операцій HTTP API. API ресурсів приховують явні мережеві виклики, але замість цього надають об'єкти ресурсів і колекції для доступу до атрибутів і виконання дій.

Після відсортування ключів з'являється змога отримати останні повідомлення, просто зробивши запит до таблиці з первинним ключем, `KeyConditionExpression = Key('sensor').eq(sensor)`, `аргументScanIndexForward = False`, необхідний для зміни порядку та `Limit = 20` дає мені результати за останню годину. Тому що всі датчики надсилають повідомлення кожні 3 хв.

Наступним і останнім кроком у розробленні автоматизованої системи моніторингу на підприємстві є використання фреймворку для Python – Flask. Це фреймворк для опрацювання HTTP/AJAX запитів і фронтенд частини вебдодатку, де буде відбуватися надсилання та отримування даних з ESP-01 у свою програму Flask. Виконання цієї задачі відбувається у середі AWS IoT Core, де і будуть опрацьовуватися запити через протокол MQTT [29]. Тобто датчики, які надсилають дані, які потім відображаються в додатку Flask, і є додаток Flask, який надсилає повідомлення в IoT [30] (рис. 3.14).

Для початку необхідно зробити сторінку для авторизації у вебдодатку для доступу до необхідної інформації. Для цього використовується стандартна бібліотека для будь якого додатку на Flask – Flask-Flask. Flask є типом даних класу Python [31]. Іншими словами, Flask – це прототип, який використовується для створення екземплярів вебдодатків (рис. 3.15).

Flask Blueprints інкапсулюють такі функції, як перегляди, шаблони та інші ресурси. Кожен проект Flask – це об'єкт, який працює дуже подібно до програми Flask. Blueprints може мати ресурси, такі як статичні файли, шаблони та представлення, які пов'язані з маршрутами. Вони записують операції, які будуть виконані пізніше, коли будуть зареєстровані у програмі. Наприклад, коли відбувається прив'язка подання з маршрутом у Flask Blueprint, він записує цей зв'язок, щоб зробити його пізніше в програмі, коли Blueprint буде зареєстровано.

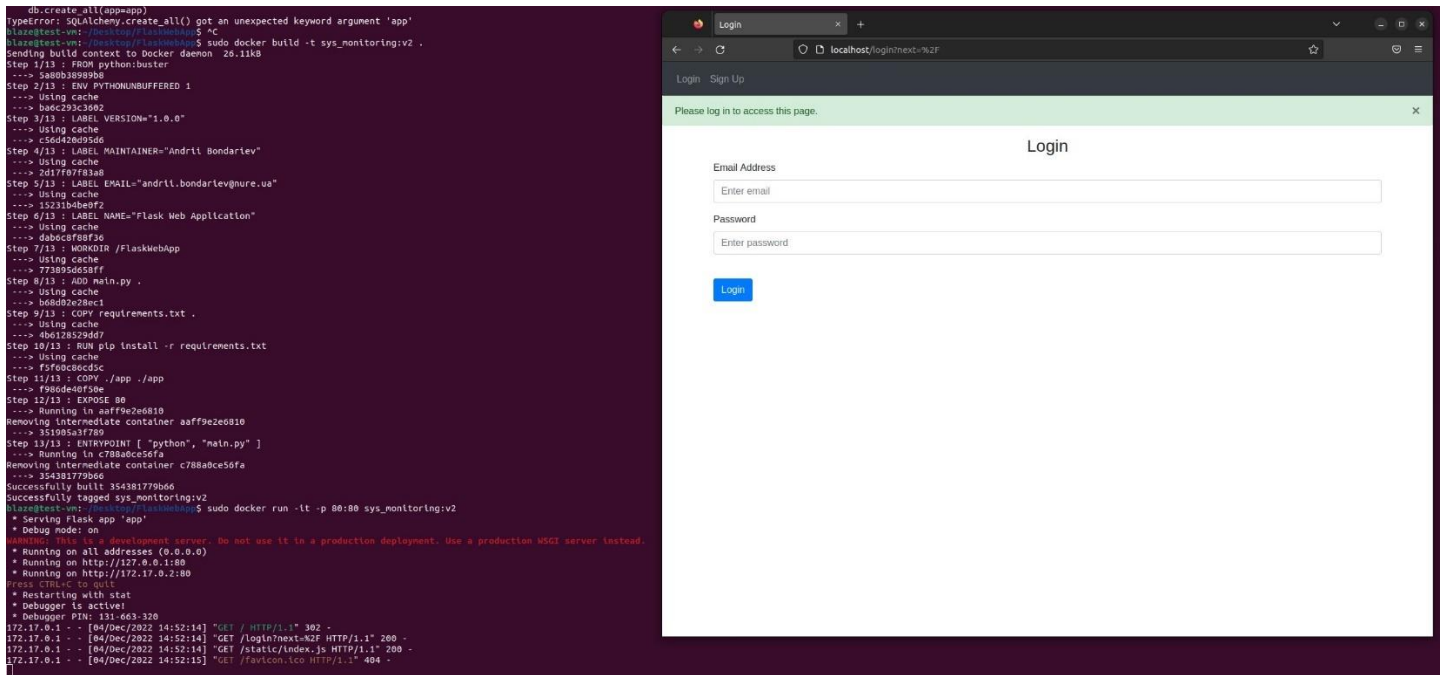


Рисунок 3.14 – Демонстрація роботи сторінки авторизації

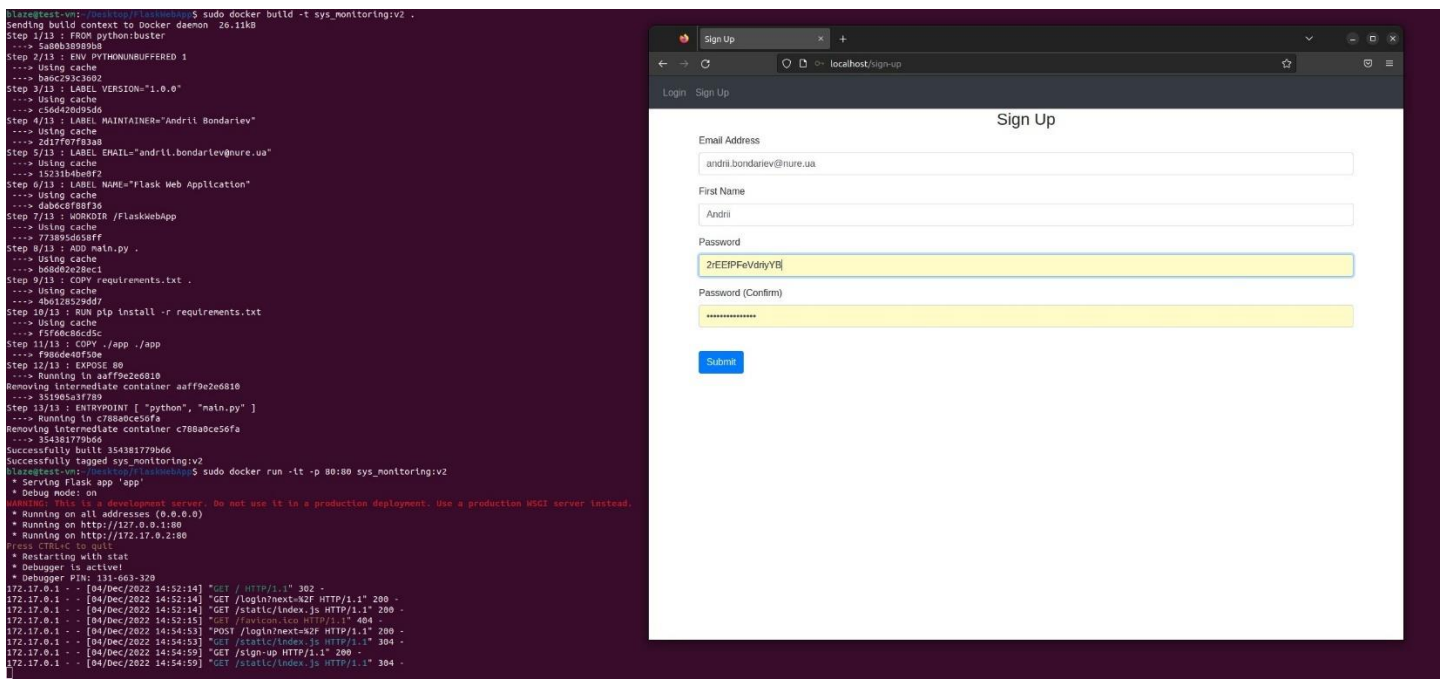


Рисунок 3.15 – Демонстрація роботи сторінки реєстрації

Дозвіл користувачам входити у вебдодаток є однією з найголовніших функцій, які можна додати до вебдодатку автоматизованої системи моніторингу на підприємстві. Додається автентифікація до програми Flask за допомогою пакета Flask-Login.

Також треба зазначити технології, які були використані для оптимізації структури процесів вебдодатку. Це контейнеризація – сервіс Docker та система контролю версій – Git. Це допоміжні системи, які спрощують процес розробки і роблять ефективнішим продукт, що розробляється. Процес розробки відбувається у редакторі коду – Visual Studio Code (рис. 3.16).

Docker – це платформа, яка дозволяє запакувати в контейнер додаток з усім оточенням та залежностями, а потім доставити та запустити його в цільовій системі. Програма, упакована в контейнер, ізолюється від операційної системи та інших програм, що зменшує навантаження на сервер і в результаті отримується краща продуктивність вебдодатку системи моніторингу.

Git – розподілена система контролю версій, яка дає можливість розробникам відстежувати зміни у файлах та працювати над одним проектом спільно з колегами.

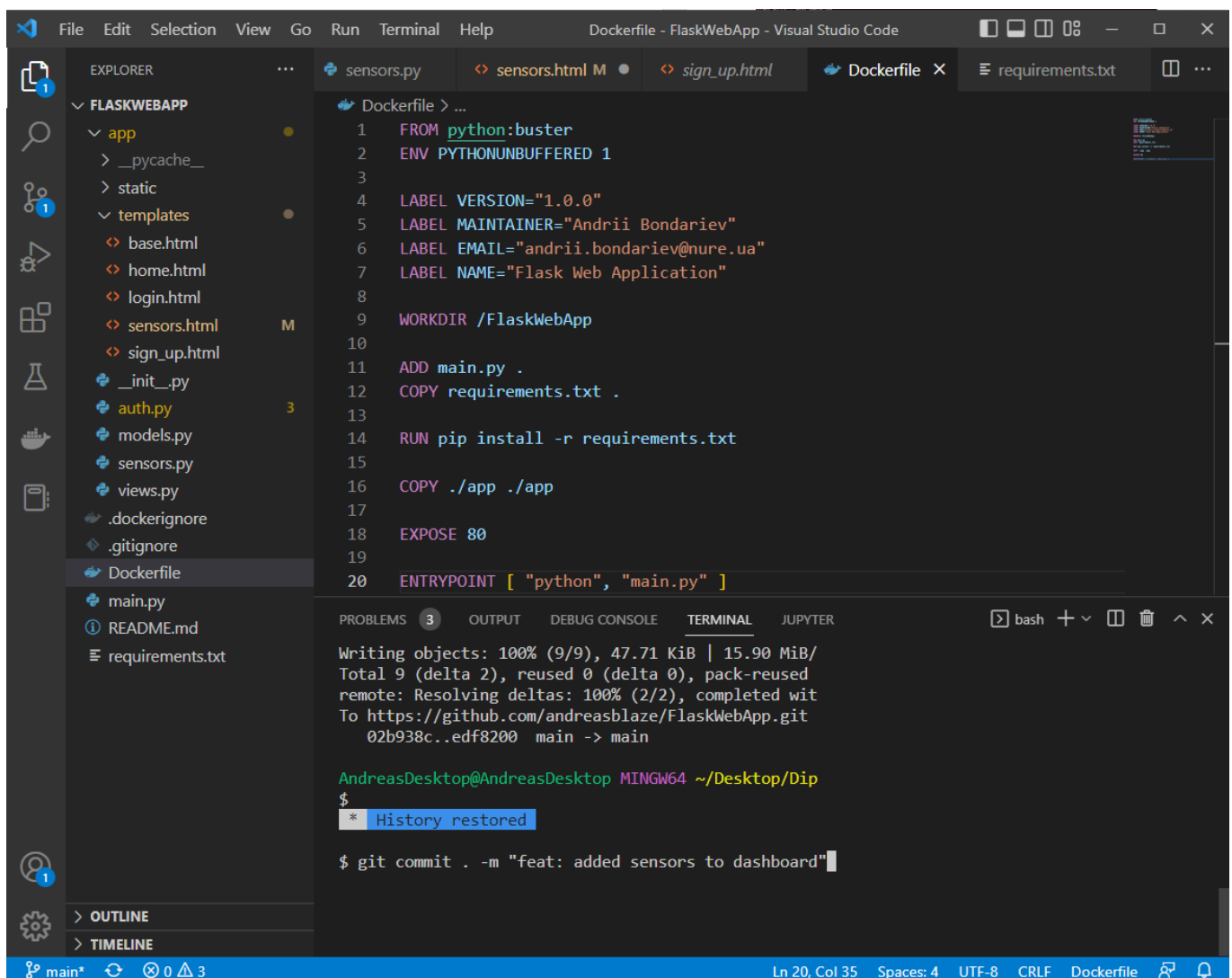


Рисунок 3.16 – Демонстрація процесу розробки у Visual Studio Code

Запускається вебдодаток Flask у локальному докер-контейнері, використовуючи середовище AWS IoT Core для роботи з інформацією з датчиків та апаратно-програмного комплексу на платформі Arduino. На рівні Flask були додані обробники інформації з сенсорів. Показники датчиків в нормі і не перевищують норми.

Були використані шаблони верстки, які використовують звичайний HTML, jQuery та Bulma (фреймворк, який надає готові до використання інтерфейсні компоненти, які можна легко комбінувати для створення адаптивних вебінтерфейсів) як структуру CSS і на рисунку 3.17 можна побачити кінцевий результат:

The screenshot shows a web browser window with the address bar at localhost/sensors. The page title is 'Sensors Dashboard'. Below the title, there are navigation links for 'Login' and 'Sign Up'. The main content area displays sensor data for an 'Air-sensor' (with a 'Test' button). The data is presented in a table with columns for Temperature, Humidity, Vibration, Noise, and Light. The current values are: Temperature: 24.816894, Humidity: 28.97644, Vibration: 37.295532, Noise: 299.2895496, and Light: 45.727539. Below this, a detailed table shows a history of readings with columns for time, temperature, humidity, vibration, noise, and light.

Sensors Dashboard					
Air-sensor Test					
TEMPERATURE	HUMIDITY	VIBRATION	NOISE	LIGHT	
24.816894	28.97644	37.295532	299.2895496	45.727539	
time	temperature	humidity	vibration	noise	light
11:51	24.816894	28.97644	37.295532	299.2895496	45.727539
11:43	-19.491578	84.664916	92.984008	139.7680632	23.571777
11:42	42.218017	46.377563	54.696655	1.9335924	4.4265745
11:41	20.211791	24.371337	32.690429	282.7111788	43.4249875
21:00	32.733154	36.8927	45.211791	327.7880856	49.6856685
20:57	38.809204	42.96875	51.287841	349.661862	2.7236935
20:54	10.137939	14.297485	22.613525	246.4343244	38.3865355
20:52	-47.561646	56.5979	64.91394	38.7158184	9.536743
20:45	-39.541626	64.617919	72.937011	67.598874	13.5482785
20:45	-17.181397	86.978149	95.297241	148.095702	24.7283935
20:44	27.716064	31.872558	40.19165	309.7155744	47.175598
20:43	8.480834	12.64038	20.959472	240.4797336	37.559509
20:43	28.067016	32.226562	40.545654	310.9899888	47.3526

Рисунок 3.17 – Результат розгортки Flask додатку

3.4 Висновки до третього розділу

У даному розділі була виконана робота з розробки і розгортання програмного та апаратного забезпечення автоматизованої системи моніторингу та управління виробничими процесами на підприємстві.

Була створена загальна архітектура автоматизованої системи моніторингу, зроблений опис кожної використаної технології. Були пророблені операції з підключення всіх компонентів до платформи Arduino UNO і відповідно було розроблено апаратно-програмне забезпечення.

Були виконані операції з налаштування середовища розробки апаратного забезпечення і поєднання з хмарним середовищем AWS IoT. Була виконана задача з розробки програмного забезпечення вебдодатку на Flask, фреймворку мови програмування Python. Було продемонстровано роботу автоматизованої системи моніторингу та управління виробничими процесами на підприємстві.

4 ОХОРОНА ПРАЦІ

Охорона праці і техніка безпеки на підприємстві включають в себе комплекс заходів, метою яких є забезпечення безпеки і збереження здоров'я працівників, зайнятих виконанням своїх трудових обов'язків. Рекомендовані практики розроблено для використання в різноманітних умовах малого та середнього бізнесу. Рекомендовані практики представляють поетапний підхід до впровадження програми безпеки та здоров'я, побудованої навколо семи основних елементів, які складають успішну програму.

Основна мета програм безпеки та гігієни праці полягає в запобіганні травмам на робочому місці, захворюваннях і смертях, а також стражданнях і фінансових труднощах, які ці події можуть спричинити для працівників, їхніх сімей і роботодавців. Традиційні підходи часто є реактивними, тобто проблеми вирішуються лише після того, як працівник отримав травму або захворів, опубліковано новий стандарт чи положення або зовнішня інспекція виявила проблему, яку необхідно вирішити. Ці рекомендовані практики визнають, що виявлення та усунення небезпеки до того, як вони спричинять травму чи хворобу, є набагато ефективнішим підходом. Зосередження на досягненні цілей, моніторингу продуктивності та оцінці результатів, ваше робоче місце може прогресувати на шляху до вищого рівня безпеки та здоров'я.

Роботодавці побачать, що впровадження цих рекомендованих практик також приносить інші переваги. Програми безпеки та здоров'я допомагають підприємствам [32]:

- запобігання травмам і захворюванням на виробництві;
- поліпшити дотримання законів і правил;
- зменшити витрати, включаючи значне скорочення компенсаційних премій працівникам;
- залучайте працівників до відпочинку від роботи;
- підвищення їх цілей соціальної відповідальності;
- підвищити продуктивність і покращити загальну бізнес-операцію.

4.1 Організація охорони праці на підприємстві

Обов'язок роботодавця – затвердити документи, передбачені статтею 13 Закону «Про охорону праці». Організація охорони праці на підприємстві – це цілісна система прав, обов'язків і повноважень суб'єктів виробничого процесу, процедур, спрямованих на підтримання безпечного рівня виробництва, правил і нормативних вимог, що регулюють питання найманої праці.

4.1.1 Охорона праці на підприємстві

Перед початком роботи нового працівника роботодавець зобов'язаний проінформувати його під розписку про умови праці, наявні на його робочому місці. В тому числі, про всі небезпечні або шкідливі виробничі фактори, які ще не усунуто, і про можливі наслідки їх впливу на здоров'я працівника, а також про можливі пільги та компенсації за роботу в таких умовах. потенційні нещасні випадки та травми на виробництві можна і потрібно запобігати. Для цього необхідно подбати про створення належних і безпечних умов праці на підприємстві.

Складові (напрями) охорони праці регламентуються відповідними нормативними документами або системою нормативних документів. Підприємство зобов'язано укласти договір про умови проведення робіт на території підприємства. Роботодавець повинен ознайомитися з даним документом і підписати його. За недотримання умов договору про охорону праці покарання нестимуть обидві сторони.

Обмін відповідною інформацією про безпеку та здоров'я з працівниками зміцнює довіру та допомагає організаціям розвиватися більш обґрунтовані рішення щодо безпеки та здоров'я. Роботодавець повинен надати працівникам можливість брати участь у всіх аспектах програми, включаючи, але не обмежуючись, допомогою. Будь-яка програма безпеки та гігієни праці потребує суттєвої участі працівників та їхніх представників.

4.2 Обов'язки роботодавця

Організація охорони праці на підприємстві залежить від роботодавця. Завданням роботодавця є також забезпечення дотримання прав працівників, передбачених нормативно-правовими актами з охорони праці.

З метою створення безпечних і здорових умов праці роботодавець виконує, зокрема, такі функції:

- утворює відповідні відділи та призначає уповноважених осіб для контролю за дотриманням вимог охорони праці, затверджує правила внутрішнього розпорядку, технологічні карти та стандарти;
- затверджує колективний договір і вживає вичерпних заходів щодо підтримання та підвищення рівня охорони праці;
- розробляє програму оптимізації виробництва, впроваджує новітні технології та наукові досягнення;
- відповідає за справний стан виробничих будівель, приміщень, виробничого обладнання та машин;
- вживає невідкладних заходів щодо допомоги потерпілим, організовує виплату компенсації таким особам;
- ініціює неупереджене та об'єктивне розслідування нещасних випадків, вивчає причини, що призвели до нещасних випадків, та затверджує перелік профілактичних заходів, спрямованих на усунення ризиків виникнення подібних причин у майбутньому;
- несе персональну відповідальність за рівень охорони праці та порушення її вимог іншими особами;
- контролює дотримання працівниками технологічних процесів, встановлених правил поведінки та режиму праці.

Встановлюючи конкретні цілі та завдання, керівництво визначає очікування для керівників, керівників і працівників, а також для програми в цілому.

4.3 Служба охорони праці на підприємстві

Створення окремої служби охорони праці на підприємстві є обов'язковим при чисельності працюючих 50 і більше осіб. Для невеликих підприємств допускається передача функцій служби охорони праці за сумісництвом особам, які мають відповідну підготовку, або залучення сторонніх спеціалістів на договірних засадах. Служба охорони праці підпорядковується безпосередньо роботодавцю.

Спеціалісти мають право:

- видавати приписи про усунення допущених недоліків, отримувати від відповідальних осіб інформацію, документацію та пояснення з питань охорони праці;

- вимагати відсторонення від роботи працівників, які не пройшли обов'язковий медичний огляд, навчання, інструктаж, атестацію знань і не мають допуску до відповідної роботи;

- за наявності загрози життю та безпеці працівників – призупинити виробничий процес;

- ініціювати питання про притягнення винних осіб до відповідальності.

Контроль за додержанням законодавства про охорону праці здійснюють професійні спілки та їх об'єднання в особі виборних органів і представників.

Профспілки здійснюють громадський контроль за дотриманням законодавства про охорону праці, створенням безпечних і нешкідливих умов праці, належних виробничо-санітарних умов, забезпеченням працівників спецодягом, спецвзуттям, іншими засобами індивідуального та колективного захисту.

У разі виникнення загрози життю або здоров'ю працівників профспілки можуть вимагати від роботодавця негайного припинення роботи на робочих місцях, виробничих ділянках, у цехах та інших структурних підрозділах взагалі на строк, необхідний для усунення загрози.

У разі відсутності на підприємстві профспілки громадський контроль за додержанням законодавства про охорону праці здійснює уповноважена працівниками особа.

На рисунку 4.1 продемонстровано те, як повинні виглядати робітники на території підприємства де є загроза травмуванню і життєдіяльності людини:



Рисунок 4.1 – Вигляд працівників на підприємстві

4.4 Відповідальність за порушення вимог законодавства про охорону праці

За порушення законодавства про охорону праці та невиконання розпоряджень посадових осіб виконавчої влади юридичні та фізичні особи, які відповідно до законодавства використовують найману працю, несуть відповідальність у встановленому законодавством порядку. органи з нагляду за охороною праці.

Сплата штрафу не звільняє юридичну або фізичну особу, яка відповідно до закону використовує найману працю, від усунення виявлених порушень у зазначений строк. Максимальний розмір штрафу не може перевищувати п'яти відсотків середньомісячного фонду заробітної плати за попередній рік юридичної або фізичної особи, яка відповідно до закону використовує найману працю.

4.5 Правила роботи за комп'ютером

Користування ноутбуком або монітором протягом тривалого періоду означає велику кількість дій на зап'ясті. Одним з поширених захворювань, пов'язаних з активним використанням комп'ютера, є остеохондроз. Це захворювання дрібних суглобів, яке часто спричинене сильним напруженням зап'ястка. Тому, щоб уникнути проблем зі здоров'ям, необхідно дотримуватись правил роботи за комп'ютером, які про продемонстровані на рисунку 4.2. Всі ці правила є основними і повинні бути відомі кожному, хто працює за комп'ютером.

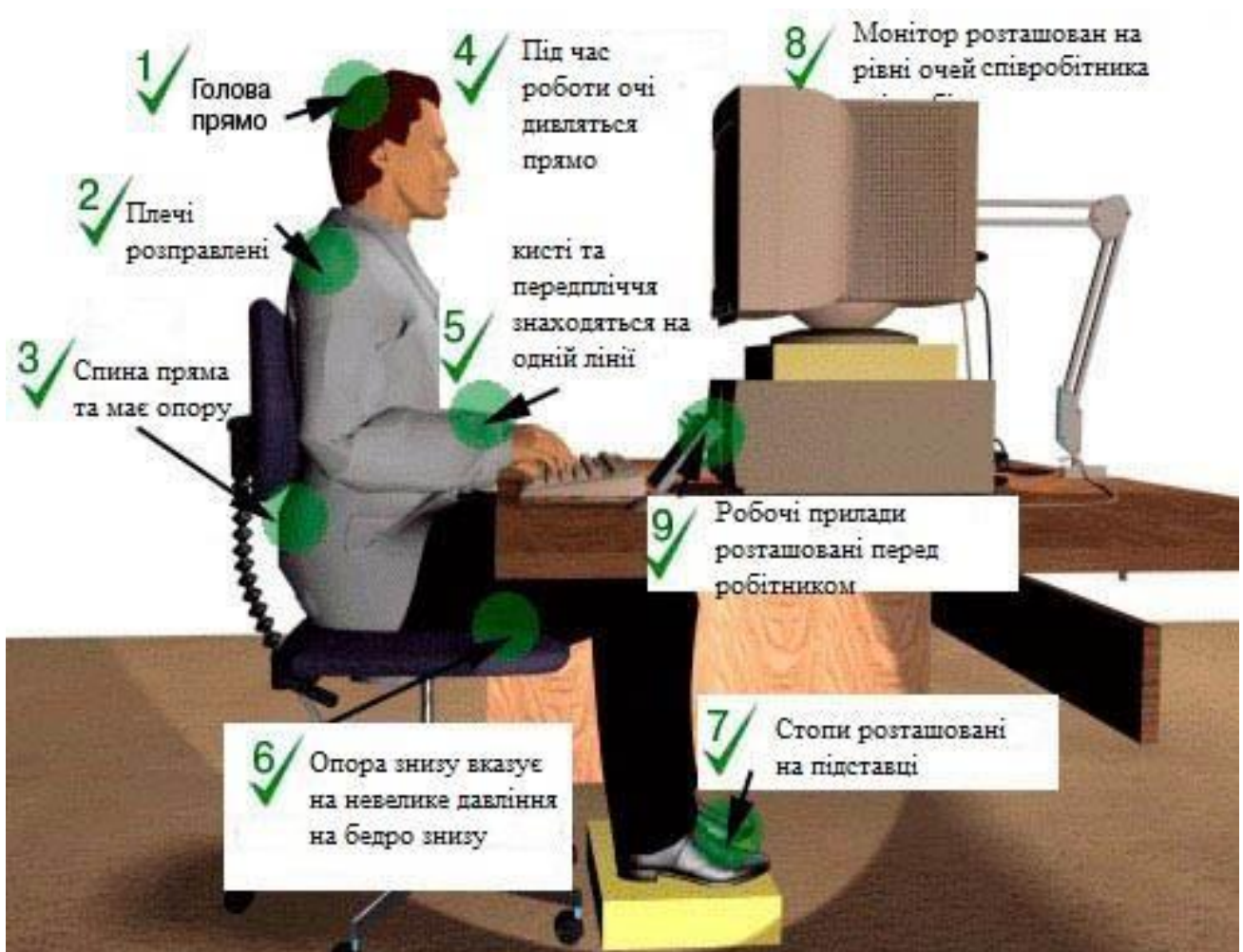


Рисунок 4.2 – Правила роботи за комп'ютером

4.6 Розрахунок освітленості робочого приміщення і робочого місця для забезпечення безпечних умов роботи

У якості основи для розрахунку необхідного освітленості приміщення для неперервної роботи системи, було обрано приміщення з геометричними розмірами 250 м × 150 м × 20 м.

Нормована освітленість для виробництв подібного розміру складає 500 лк. Напруга в мережі підприємства – 220 В, передбачається використовувати світильники люмінесцентні типу ЛПО, для даних світильників коефіцієнт використання світлового потоку – 49 %. Відбивна здатність для різних поверхонь:

- стелі 0,5;
- стін 0,5;
- робочої поверхні 0,3.

Робоча поверхня розміщена на висоті 0,8 м, висота схилю – 0,1 м. Площа ділянки становить 37500 м². Індекс приміщення розраховується формулою (4.1):

$$\varphi = (a \cdot b) / H_p (a + b), \quad (4.1)$$

де a , b – довжина і ширина приміщення;

H_p – висота установки світильників над розрахунковою площиною.

За розрахункову площину було використано висоту письмового столу, тобто

$H_2 = 0,8$ м. Висота установки світильників розраховується за формулою (4.2):

$$H_p = H_1 - H_2, \quad (4.2)$$

У нашому випадку висота установки світильника над рівнем підлоги дорівнює 20 м, тобто $H_1 = 20$ м. З цього можна отримати, що висота установки світильників дорівнює:

$$H_p = 20 - 0,8 = 19,2 \text{ м.}$$

Після знаходження висоти установки світильників, можна розрахувати індекс приміщення, який дорівнює:

$$\varphi = (250 \cdot 150) / 19,2 \cdot (250 + 150) = 4,9.$$

Для знаходженні кількості світильників N , що забезпечують необхідну освітленість, необхідно використовувати цю формулу (4.3):

$$N = E_{cp}Sk / Un\Phi_{л}, \quad (4.3)$$

де $\Phi_{л}$ – світловий потік лампи, є паспортної характеристикою ламп, лм;

n – кількість ламп в світильнику, шт .;

S – площа освітлюваного приміщення, м²;

U – коефіцієнт використання;

E_{cp} – середню освітленість приміщення;

k – коефіцієнт запасу, $k = 1,4$ для сухих чистих приміщень.

Обчислення коефіцієнта використання світильника полягає у вирішенні системи лінійних алгебраїчних рівнянь, складених для всіх поверхонь, що відбивають. Згідно системи лінійних алгебраїчних рівнянь для світильника з відбивачем коефіцієнти використання дорівнює $U = 65$.

Знаючи всі параметри ми можемо обчислити необхідну кількість світильників для забезпечують освітленості приміщення:

$$N = (500 \cdot 37500 \cdot 1,4) / (65 \cdot 6 \cdot 1150) = 58,5 \text{ шт.}$$

Округливши результат, отримаємо необхідну кількість світильників, що дорівнює 59 шт.

4.7 Висновки до четвертого розділу

У даному розділі було опрацьовано охорону праці і техніку безпеки на підприємстві. Були наведені витяги із законодавства і приклади взаємодії із робітниками та роботодавцями.

Були опрацьовані організація охорони праці на підприємстві, обов'язки роботодавця і працівника, служба охорони праці на підприємстві а також відповідальність за порушення вимог законодавства про охорону праці. Окремо були наведені правила роботи за комп'ютером і приклад.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи був пророблений аналіз актуальності розробки та технологій системи моніторингу на підприємстві. Проаналізовано технічне завдання, обрано та обґрунтовано програмно-апаратні засоби. Описано структуру, функціонал та принцип роботи системи. Було проаналізовано вже існуючі системи моніторингу на підприємствах, системи на базі датчиків, що збирають інформацію про критичні показники для виконання робочих процесів. Були досліджені датчики, які можуть використовуватися на підприємствах і важливість тих чи інших показників, актуальність збору метрик. Були розглянуті мікроконтролерні рішення у вигляді платформ для апаратного забезпечення.

За висновками досліджень і аналізів була обрана та спроектована структура розробки автоматизованої системи моніторингу, де були розроблені схеми взаємодії з компонентів та технологій обробки інформації. Була побудована інфраструктура обробки і розгорнення показників системи моніторингу виробничих процесів за допомогою хмарного сервісу Amazon Web Services, мови програмування Python і фреймворку Flask. Продемонстровано роботу компонентів модулю та структуру проекту, розгорнутого у середовищі AWS IoT.

За висновками проектування автоматизованої системи моніторингу був проведений вибір середовищ розробки, був описаний чіткий алгоритм роботи програмного забезпечення, його налаштування і бібліотек, після чого проведена розробка апаратного забезпечення для платформи Arduino, інтеграція з середовищем AWS IoT та розробка вебдодатку на мові програмування Python з використанням фреймворку Flask. Була розгорнута інтеграція апаратно-програмного комплексу на базі платформи Arduino UNO з середовищем Інтернету речей AWS за допомогою технології брокера повідомлень MQTT з використанням Arduino C++. Була побудована взаємодія обробки даних з використанням нереляційної бази даних DinamoDB.

На основі розробленої системи на базі виробництва, а саме на його

контрольно-пропускній частині, був розроблен документ по охороні праці.

У результаті виконання кваліфікаційної роботи продемонстровано роботу:

- апаратно-програмного комплексу з сенсорів та Wi-Fi модулю на базі Arduino UNO;
- інтеграція компонентів Arduino з AWS IoT за допомогою MQTT брокеру повідомлень;
- роботу нереляційної бази даних AWS DynamoDB;
- програмного забезпечення у вигляді вебдодатку для обробки інформації з датчиків на мові програмування Python та фреймворку Flask.

Результати роботи пройшли апробацію на 26-му Міжнародному молодіжному форумі “Автоматизовані системи та комп’ютеризовані технології радіоелектронного приладобудування”, а також на 17-тій Міжнародній науково-практичній конференції “Мультидисциплінарні наукові записки. Теорія, методологія і практика”.

Тези були опубліковані у збірнику студентських наукових та на міжнародній конференціях: “Автоматизація та приладобудування ADED-2022” [5], “Виробництво & Мехатронні Системи 2022” [6].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. К.: ДП “УкрНДНЦ”. 2016. 30 с.
2. Методичні вказівки до підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології” освітньо-професійних програм: “Автоматизоване управління технологічними процесами”, “Комп’ютерно-інтегровані технологічні процеси і виробництва”, “Комп’ютеризовані та роботехнічні системи” / упоряд. : І.Ш. Невлюдов, Р.В. Артюх, В.В. Безкоровайний, Н.П. Демська, В.В. Євсєєв, О.І. Филипенко, О. М. Цимбал. Харків : ХНУРЕ, 2021. 55 с.
3. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 “Автоматизація та комп’ютерно-інтегровані технології”: довід. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. К.: Київ-56, пр. Космонавта Комарова, 1, 2016. 320 с.
4. Положення про протидію академічному плагіату в ХНУРЕ [Електронний ресурс] / nure.ua. – Режим доступу: [www/ URL: https://www.nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf](http://www.nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf) (дата звернення : 16.10.2022).
5. Бондарєв А. М. Розроблення структури автоматизованої системи моніторингу на виробництві // “Автоматизація та приладобудування” ADED-2022, (Харків, ХНУРЕ, 2022 р.). С. 19-24.
6. Бондарєв А. М. Розробка автоматизованої системи моніторингу. – Виробництво та мехатронні системи 2022 (“Manufacturing & Mechatronic Systems 2022”) : матеріали VI-ої Міжнародної конференції, Харків, 21-22 жовтня 2022 р.: тези доповідей / [редкол. І.Ш. Невлюдов (відповідальний редактор)]. Харків: [електронний друк], 2022. С. 15-18.
7. Production Monitoring System and Its Benefits [Електронний ресурс] / techna-tool.com. – Режим доступу: [www/ URL: https://www.techna-tool.com](http://www.techna-tool.com)

tool.com/blog/production-monitoring-system-and-its -benefits/ – 22.09.2022 р. (дата звернення : 16.11.2022).

8. Cloud computing with AWS [Електронний ресурс] / aws.amazon.com. – Режим доступу: [www/ URL::https://aws.amazon.com/what-is-aws/?nc1=f_cc](https://aws.amazon.com/what-is-aws/?nc1=f_cc) (дата звернення : 12.11.2022).

9. Nevlyudov I. Intelligent means in the system of managing a manufacturing agent / I. Nevlyudov, O. Tsymbal, A. Bronnikov // Сучасний стан наукових досліджень та технологій в промисловості. 2018. №1. С. 20–38.

10. PRTG Enterprise Monitor [Електронний ресурс] / paessler. – Режим доступу: [www/URL: https://www.paessler.com/prtg-enterprise-monitor](https://www.paessler.com/prtg-enterprise-monitor) (дата звернення : 16.10.2022).

11. Основи виробництва електронних апаратів / Невлюдов І.Ш. Харків: ТОВ «Компанія СМІТ». 2005. 598 с.

12. Афзель С. Arduino, датчики та мережі для зв'язку пристроїв. – Ефективність інженерних рішень у приладобудуванні: матеріали доповідей XIV Всеукраїнської науковопрактичної конференція студентів, аспірантів та молодих вчених: тез. докл. науч.-практ.конф. / С. Афзель, М. Березанська. Київ: КПІ ім. Ігоря Сікорського; Центр учбової літератури, 2018. С. 16-23.

13. Невлюдов І.Ш. Виробничі процеси та обладнання об'єктів автоматизації: Підручник / Кривий Ріг: КК НАУ. 2017. 444 с.

14. Arduino Documentation [Електронний ресурс] / docs.arduino.cc. – Режим доступу: [www/ URL: https://docs.arduino.cc/hardware/uno-rev3](https://docs.arduino.cc/hardware/uno-rev3) (дата звернення : 16.11.2022).

15. Overview of the Arduino UNO Components [Електронний ресурс] / docs.arduino.cc. – Режим доступу: [www/ URL: https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board](https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board) (дата звернення : 16.11.2022).

16. Невлюдов І.Ш. Виробничі процеси та обладнання об'єктів автоматизації. Збірник задач: Навчальний посібник / І.Ш. Невлюдов, А.О. Андрусевич, Г.В. Пономарьова, А.О. Функендорф. Кривий Ріг: КК НАУ. 2018. 332 с.

17. Oleksandr Tsymbal, Artem Bronnikov, Paolo Mercorelli. Decision-making models for Robotic Warehouse // Proceedings of 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Virtual Meeting, June 24-26, P. 546-551.
18. Невлюдов І.Ш. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипенко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ. 2019. 366 с.
19. Цимбал О.М., Бронніков А.І. Інтелектуальне керування гнучкою роботизовано системою // Матеріали першої міжнародної науково-практичної конференції «Computer and informational systems and technologies». 20-22 квітня 2017 р. Харків: ХНУРЕ, 2017. С. 45-46.
20. Raspberry Pi 4 Tech Specs [Електронний ресурс] / raspberrypi.com. – Режим доступу: [www/ URL: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/](https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/) (дата звернення : 16.10.2022).
21. Igor Nevliudov, Vladyslav Yevsieiev, Murad Omarov, Artem Bronnikov and Viacheslav Liashenko. Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis //International Journal of Emerging Trends in Engineering Research (IJETER), Volume 8 No.10 (October 2020), pp: 7465 – 7473. ISSN 2347 - 3983 (DOI:10.30534/ijeter/2020/1278102020).
22. Industrial Internet of Things [Електронний ресурс] / aws.amazon.com. – Режим доступу: [www/ URL: https://aws.amazon.com/iot/solutions/industrial-iot/?nc1=h_ls](https://aws.amazon.com/iot/solutions/industrial-iot/?nc1=h_ls) (дата звернення : 16.10.2022).
23. Internet of Things (IoT) solutions [Електронний ресурс] / aws.amazon.com. – Режим доступу: [www/ URL: https://aws.amazon.com/marketplace/solutions/IoT?aws-marketplace-cards.sort-by=item.additionalFields.sortOrder&aws-marketplace-cards.sort-order=asc&awsf.aws-marketplace-iot-store-use-cases=*all&awsf.iot-offer-type=*all](https://aws.amazon.com/marketplace/solutions/IoT?aws-marketplace-cards.sort-by=item.additionalFields.sortOrder&aws-marketplace-cards.sort-order=asc&awsf.aws-marketplace-iot-store-use-cases=*all&awsf.iot-offer-type=*all) (дата звернення : 16.10.2022).
24. Невлюдов І.Ш. Техніко-економічне обґрунтування інженерних рішень: Підручник / І.Ш. Невлюдов. Кривий Ріг : Криворізький коледж НАУ. 2019. 448 с.
25. Мінцифра та Amazon (AWS) співпрацюватимуть для прискорення розвитку хмарних технологій в Україні [Електронний ресурс]. – Режим доступу:

www/ URL: <https://thedigital.gov.ua/news/mintsifra-ta-amazon-aws-spiivpratsyuvatimut-dlya-priskorennya-rozvitku-khmarnikh-tekhnologiy-v-ukraini> (дата звернення : 16.11.2022).

26. Датчики та модулі для контролерів Arduino [Електронний ресурс] / arduinka.biz.ua. – Режим доступу: www/ URL: <https://arduinka.biz.ua/uk/datchiki-ta-moduli-c74.html> (дата звернення : 16.10.2022).

27. Flask [Електронний ресурс] / readthedocs.org. – Режим доступу: www/ URL: <https://readthedocs.org/projects/flask/> (дата звернення : 16.10.2022).

28. Create and Query a NoSQL Table with Amazon DynamoDB [Електронний ресурс] / aws.amazon.com. – Режим доступу: www/ URL: https://aws.amazon.com/getting-started/hands-on/create-nosql-table/?ref=docs_gateway/dynamodb/Introduction.html (дата звернення : 16.10.2022).

29. Flask-MQTT's documentation [Електронний ресурс] / flask-mqtt.readthedocs.io. – Режим доступу: www/ URL: <https://flask-mqtt.readthedocs.io> (дата звернення : 16.10.2022).

30. aws-iot-python [Електронний ресурс] / github.com. – Режим доступу: www/ URL: <https://github.com/topics/aws-iot-python> (дата звернення : 16.10.2022).

31. Python documentation. [Електронний ресурс] / docs.python.org. – Режим доступу: www/ URL: <https://docs.python.org/3/> (дата звернення : 16.10.2022).

32. Recommended Practices for Safety and Health Programs [Електронний ресурс] / Occupational Safety and Health Administration. – Режим доступу: www/ URL: <https://www.osha.gov/sites/default/files/publications/OSHA3885.pdf> (дата звернення : 16.10.2022).