

Автоматизація збору метеоданих з IoT-сенсорів за допомогою Python та MQTT

Злата Волохова,

здобувачка вищої освіти спеціальності 122 Комп'ютерні науки

Миколаївський національний аграрний університет

м. Миколаїв, Україна

Анотація: *в роботі розглядається автоматизація збору метеорологічних даних як важливий напрямок розвитку сучасних екологічних моніторингових систем. Інтеграція IoT-сенсорів з протоколом MQTT дозволяє передавати дані в режимі реального часу, що забезпечує оперативність та точність вимірювань. Використання мікроконтролерів ESP8266, ESP32 та Raspberry Pi у поєднанні з мовою програмування Python спрощує розробку таких систем..*

Ключові слова: *автоматизація, метеодані, IoT-сенсори, Python, MQTT, моніторинг навколишнього середовища, калібрування сенсорів, обробка даних, візуалізація даних.*

Автоматизація збору метеоданих є ключовим елементом сучасних систем моніторингу навколишнього середовища. Вона дозволяє значно підвищити точність прогнозування погодних умов, оперативно реагувати на зміни клімату та ефективно використовувати отримані дані у різних сферах діяльності. Впровадження IoT-сенсорів у метеорологічні системи дає змогу створювати розподілені мережі збору інформації, які працюють у режимі реального часу, передаючи дані через надійні протоколи комунікації, такі як MQTT. Python є ідеальним інструментом для розробки таких систем, оскільки він пропонує широкий вибір бібліотек для роботи з мікроконтролерами, обробки даних та інтеграції з хмарними сервісами. Поєднання цих технологій забезпечує можливість створення ефективних, масштабованих та економічно вигідних рішень для екологічного моніторингу та прогнозування змін клімату.

IoT-сенсори поділяються на кілька категорій залежно від типу вимірюваних параметрів. Сенсори температури, такі як DHT22 та DS18B20, забезпечують точні вимірювання температури, при цьому DS18B20 працює в широкому діапазоні температур. Сенсори вологості, наприклад SHT31, використовуються для точного визначення рівня вологості повітря. Барометричні сенсори, такі як BMP280 і BME680, дозволяють вимірювати атмосферний тиск та навіть рівень якості повітря. Анемометри допомагають визначати швидкість і напрямок вітру, що важливо для метеорологічного моніторингу. Інфрачервоні сенсори можуть використовуватися для вимірювання рівня сонячного випромінювання. Ці сенсори підключаються до мікроконтролерів, таких як ESP8266, ESP32 або Raspberry Pi, які забезпечують первинну обробку даних та їх передачу на центральні сервери або у хмарні сервіси для подальшого аналізу та візуалізації.

MQTT – це легковаговий протокол обміну повідомленнями, що ідеально підходить для IoT-застосувань завдяки низькому використанню ресурсів та

ефективній передачі даних навіть у мережах з обмеженою пропускнуою здатністю. Він використовує архітектуру «публікація-підписка», що дозволяє різним пристроям ефективно обмінюватися даними через MQTT-брокер (наприклад, Mosquitto, HiveMQ або EMQX). У цій архітектурі сенсори виступають у ролі видавців (publishers), які передають дані на певні теми, а отримувачі (subscribers) можуть підписуватися на ці теми для отримання інформації. Крім того, MQTT підтримує якість обслуговування (QoS) на трьох рівнях, що забезпечує гнучкість та надійність у передачі даних. Завдяки вбудованим механізмам збереження повідомлень (retain message) та підтримці TLS/SSL для захисту даних, цей протокол широко використовується не тільки в метеостанціях, але й у розумних містах та енергетичних системах.

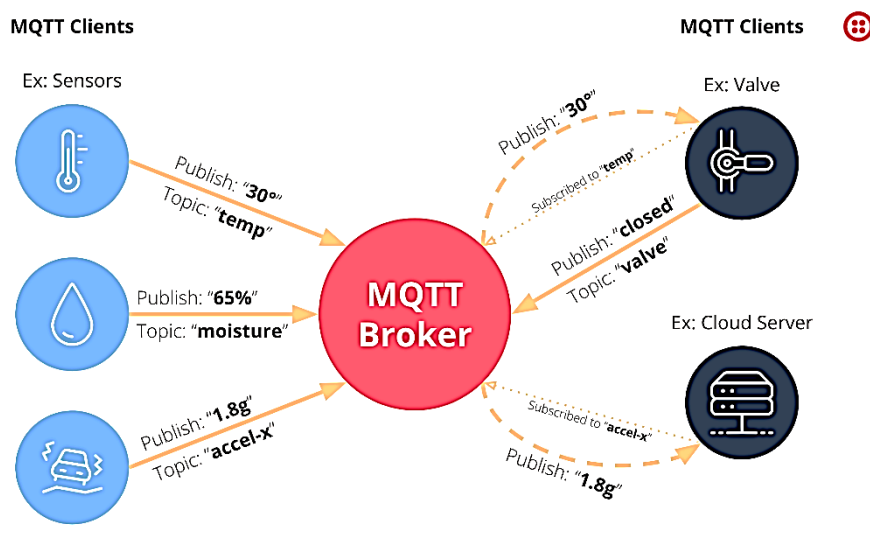


Рис. Архітектура MQTT протоколу

Джерело: <https://www.twilio.com/en-us/blog/what-is-mqtt>

Python є популярною мовою програмування для роботи з IoT, оскільки має безліч бібліотек для роботи з мікроконтролерами, мережевими з'єднаннями та протоколом MQTT. Однією з найпоширеніших бібліотек для роботи з MQTT у Python є `paho-mqtt`. Ось приклад коду для збирання та передачі даних з датчика температури DHT22 за допомогою ESP8266: У цьому коді мікроконтролер зчитує показники температури з датчика DHT22 та передає їх на MQTT-брокер через задану тему. Інші пристрої можуть підписатися на цю тему для отримання даних.

Отримані метеодані можна зберігати у базі даних (наприклад, InfluxDB) або візуалізувати за допомогою графічних бібліотек, таких як Matplotlib або платформи для моніторингу Grafana. Крім того, використання хмарних сервісів, таких як Google Cloud BigQuery або AWS IoT Analytics, дозволяє централізовано зберігати великі обсяги даних і проводити глибокий аналіз за допомогою алгоритмів машинного навчання.

```

import paho.mqtt.client as mqtt
import Adafruit_DHT
import time

BROKER = "mqtt.eclipse.org"
TOPIC = "weather/temperature"
DHT_SENSOR = Adafruit_DHT.DHT22
DHT_PIN = 4

client = mqtt.Client()
client.connect(BROKER, 1883, 60)

while True:
    humidity, temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)
    if temperature is not None:
        client.publish(TOPIC, f"Temperature: {temperature:.2f}C")
        time.sleep(10)

```

Це допомагає виявляти довгострокові тенденції кліматичних змін та створювати прогностичні моделі. Використання бази даних також забезпечує можливість інтеграції з іншими системами, такими як мобільні додатки для моніторингу погоди або автоматизовані системи керування сільськогосподарськими процесами.

Автоматизація збору метеоданих за допомогою IoT-сенсорів, Python і MQTT є ефективним рішенням для екологічного моніторингу. Це дозволяє створювати розподілені системи збору даних, що забезпечують точний і безперервний моніторинг погодних умов. Подальші дослідження можуть бути спрямовані на застосування алгоритмів машинного навчання для прогнозування змін клімату на основі зібраних даних.

Список використаних джерел:

1. <https://mqtt.org/>
2. <https://randomnerdtutorials.com/mqtt/>
3. <https://docs.python.org/3/library/>
4. <https://www.hivemq.com/mqtt-protocol/>

Abstract: *The paper considers the automation of meteorological data collection as an important direction of development of modern environmental monitoring systems. The integration of IoT sensors with the MQTT protocol allows for real-time data transmission, which ensures the efficiency and accuracy of measurements. The use of ESP8266, ESP32, and Raspberry Pi microcontrollers in combination with the Python programming language simplifies the development of such systems,*

Keywords: *Automation, meteorological data, IoT sensors, Python, MQTT, environmental monitoring, sensor calibration, data processing, data visualisation.*

Науковий керівник:

Пархоменко О. Ю.,

*к. ф.-м. н., доцент кафедри економічної кібернетики,
комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет*