

УДК 004.4

Куликовська Н.А.¹, Тіменко А.В.², Трубичин Д.С.³

¹ старш. викл. НУ «Запорізька політехніка»

² викл. Запорізький авіаційний фаховий коледж ім. О.Г. Івченка

³ студ. гр. КНТ-512 НУ «Запорізька політехніка»

ВИКОРИСТАННЯ PYTHON ДЛЯ РОЗРОБКИ ТА МОДЕЛЮВАННЯ ІОТ-СИСТЕМ

З огляду на стрімкий розвиток Інтернету речей (IoT) як ключового напрямку цифрової трансформації в промисловості, розумних містах та побутових системах, виникає потреба у дослідженні гнучких інструментів програмування для створення, моделювання та тестування IoT-рішень, тому вивчення можливостей мови програмування Python у цьому контексті є надзвичайно актуальним. Серед існуючих рішень виділяються такі програмні платформи, як MicroPython для мікроконтролерів, бібліотеки paho-mqtt для роботи з протоколом MQTT, flask для розгортання серверів REST API, а також фреймворки для швидкого прототипування IoT-додатків, зокрема ThingsBoard [1]. Останні дослідження підтверджують високу ефективність Python у розробці як крайових, так і серверних компонентів IoT-інфраструктури завдяки його простоті, великій кількості готових бібліотек та

підтримці асинхронних обчислень [2]. Разом з тим невирішеною залишається проблема оптимізації продуктивності Python-додатків у обмежених обчислювальних середовищах, таких як IoT-пристрої на базі ESP32 або Raspberry Pi Zero, що визначає напрям подальших досліджень.

Метою роботи є розробка і моделювання експериментальної IoT-системи із застосуванням мови Python для реалізації повного циклу обробки даних від збору через сенсори до візуалізації інформації у вебінтерфейсі, з акцентом на використанні протоколу MQTT для комунікації. Основним завданням дослідження є вивчення ефективності Python-бібліотек для збору, передачі, обробки і візуалізації даних, аналіз обмежень продуктивності та вироблення рекомендацій для розробників IoT-додатків.

У роботі використано метод експериментального моделювання із застосуванням бібліотек MicroPython для мікроконтролерів ESP32, paho-mqtt для налаштування обміну даними через MQTT-брокер Eclipse Mosquitto, а також flask для розгортання вебсерверу для візуалізації даних [1–4]. Для оцінки ефективності були використані метрики швидкості обробки повідомлень, часу реакції на події, навантаження на пристрої та енергоспоживання у реальному середовищі.

Практичне дослідження здійснювалось через створення невеликої IoT-мережі, що включала сенсор температури DHT22, підключений до мікроконтролера ESP32, на якому працював код на MicroPython для збору та передачі даних кожні 10 секунд через MQTT-протокол до брокера Mosquitto, встановленого на сервері на базі Raspberry Pi 4. Дані, що надходили на сервер, оброблялись скриптом на Python з використанням бібліотеки paho-mqtt, після чого результат виводився у вебінтерфейс через локальний flask-сервер. В рамках експерименту досліджувалися час затримки між передачею даних і їхньою візуалізацією, частота втрачених пакетів при зміні умов мережі та навантаження на процесор ESP32 і Raspberry Pi 4, результати чого наведено в таблиці 1.

Таблиця 1 - Параметри роботи експериментальної IoT-системи на Python

Параметр	Значення
Середня затримка доставки даних	120 мс
Втрата повідомлень при нестабільній мережі	2%
Середнє навантаження на ESP32	40% CPU
Середнє навантаження на Raspberry Pi 4	15% CPU

Модель функціонування IoT-системи, розробленої на Python, передбачала послідовне проходження даних від сенсорного вузла до брокера MQTT і далі до вебсерверу для обробки та відображення даних. Процес

обробки даних включав періодичний збір показників температури, їхню упаковку у JSON-формат, передачу через MQTT-брокер, обробку повідомлень скриптом Python на сервері і передачу на вебінтерфейс через API.

Мова програмування Python є надзвичайно ефективним інструментом для швидкої розробки, тестування та розгортання прототипів IoT-систем, особливо в поєднанні з протоколом MQTT для організації надійного обміну даними, однак при розробці необхідно враховувати обмеження ресурсів на пристроях та оптимізувати код для зменшення затримок і споживання енергії.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MicroPython – Python for microcontrollers [Electronic resource]. – Mode of access: <https://micropython.org/>
2. Eclipse Paho MQTT Python Client [Electronic resource]. – Mode of access: <https://www.eclipse.org/paho/index.php?page=clients/python/index.php>
3. Flask – Python web framework [Electronic resource]. – Mode of access: <https://flask.palletsprojects.com/>
4. MQTT Essentials – A Lightweight IoT Protocol [Electronic resource]. – Mode of access: <https://www.hivemq.com/mqtt-essentials/>