

АНАЛІЗ ЕФЕКТИВНОСТІ СПОСОБІВ ПЕРЕДАВАННЯ ІНФОРМАЦІЇ МІЖ ESP32 ТА RASPBERRY PI

Яковенко О.С.

Науковий керівник – к.т.н., проф. Зубков О.В.

Харківський національний університет радіоелектроніки, каф. МТС

м. Харків, Україна

e-mail: oleksandr.iakovenko@nure.ua

The work analyzed the technical characteristics of ESP32 modules and Raspberry PI single-board computers for the implementation of complex distributed systems for collecting, managing and visualizing information. An analysis of HTTP and MQTT protocols used in such systems was also carried out. An analysis of the advantages and disadvantages of Arduino libraries for implementing data transfer using the HTTP and MQTT protocols was carried out. Programs for transmitting sensor readings and receiving them on Raspberry PI were created and tested. A comparative analysis of the resources consumed by these programs was carried out. Recommendations for their joint use are given.

ESP32 і Raspberry Pi - це два дуже популярних пристрої для реалізації таких проектів, як "розумний будинок", телеметричні системи збору даних і т.д. Кожен з них має свої унікальні характеристики та можливості, і вони можуть використовуватися як окремо, так і в поєднанні для створення різних рішень для автоматизації. ESP32 - це мікроконтролер, який часто використовується для побудови кінцевих "розумних" пристроїв, таких як датчики руху, температурні датчики і т.д. Він має вбудований Wi-Fi та Bluetooth, два ядра, від 2 до 8 Мбайт вбудованої Flash пам'яті, 520кБайт оперативної пам'яті та тактову частоту до 240 МГц, що робить його ідеальним для з'єднання зі смартфонами або іншими мережевими пристроями. Обмеженість внутрішніх ресурсів таких пристроїв обмежує реалізацію голосових асистентів, створення складних сайтів, обробку відеопотоків з відеокамер. Raspberry Pi - це мініатюрний одноплатний комп'ютер, який працює під управлінням операційної системи Linux. Зазвичай, ESP32 використовується для збору даних на місці, а Raspberry Pi - для обробки цих даних, управління та візуалізації інформації, а також для взаємодії з користувачем через веб-інтерфейс або мобільний додаток.

Крім апаратної підтримки передавання інформації не менш важлива програмна реалізація взаємодії між ESP32 та Raspberry PI, бо саме вона визначає швидкість доставки, надійність в умовах низькошвидкісних каналів зв'язку, простоту реалізації кода, а, відповідно, і швидкість роз-

робки проекту. Саме аналіз переваг та недоліків використання існуючих бібліотек для розробки програмного забезпечення (ПЗ) на обох пристроях і є предметом дослідження.

У світі Інтернеті речей найбільш поширеними протоколами є: HTTP (Hypertext Transfer Protocol) і MQTT (Message Queuing Telemetry Transport). Текстовий протокол HTTP працює за принципом клієнт-серверної моделі. Він потребує більшого обсягу заголовків і метаданих у порівнянні з MQTT. Хоча він може бути надійним у більш стабільних мережевих умовах, може виявити проблеми у випадку втрати з'єднання або перебоїв у мережі. MQTT протокол є бінарним, має: менший обсяг накладних витрат, кращу ефективність передачі даних, вбудовану підтримку для управління якістю обслуговування (QoS), що дозволяє гнучко керувати рівнем надійності доставки повідомлень.

Найбільш поширеною бібліотекою клієнту HTTP для модулів ESP32 у середовищі Arduino IDE є HTTPClient. Вона дозволяє організувати з'єднання з сервером, відправку повідомлення у POST запиті і отримання підтвердження про його доставку. Для структурування даних, що відправляються, використовується JSON формат, заголовок повідомлення помічається "Content-Type", "application/json". Данні передаються у вигляді словника, що містить назви інформаційних параметрів та їх значення. Формувати такі словники можна методами обробки рядків мови C або з використанням спеціалізованої бібліотеки ArduinoJson.

У Raspberry PI ПЗ в основному розробляється мовою Python і для прийому пакетів від клієнтів теж підключаються спеціалізовані бібліотеки. Однією з поширених бібліотек Web серверів є Flask, яку можна використовувати для прийому POST запитів різних датчиків і для одночасної реалізації сайту домашньої автоматизації. Наявність в цій бібліотеці модуля json дозволяє одразу інтерпретувати запити і приймати дані у вигляді словників. Реалізація протоколу MQTT на Raspberry PI більш складна, бо потребує спочатку встановлення MQTT брокеру, а після цього у програмі на Python підключення клієнту до цього брокеру. Наприклад, `raho.mqtt.client`.

Дослідження та їх результати. Для досліджень були використані: модуль ESP-WROOM-32 із Flash пам'яттю 8 Мбайт та Raspberry PI 4 із ОЗУ 4Гбайт. Для модуля ESP32 біло написано 3 програми, що опитують однакові датчики температури та вологості і передають дані на сервер у вигляді Raspberry PI. В першій програмі було розроблено функцію для формування рядка із назвами датчиків і їх показаннями методами мови C відповідно до стандарту JSON (наприклад,

```
"{"sensor":"DS18B20","value":"21.38","sensor":"BME280",  
  "value1":"60.54","value2":"954.14"}").
```

Друга програма використовувала бібліотеку ArduinoJson і формувала словник стандартними методами цієї бібліотеки (наприклад, `doc["DS18B20"] = temperatureC; BME280.add(60.54);`).

Третя програма використовувала протокол MQTT та передавала дані у вигляді топиків `/myhome/bedroom/DS18B20`. Результати компіляції розроблених програм наведені у таблиці 1

Таблиця 1 – Результати компіляції програм для модуля ESP32

Варіант програми	Об'єм flash пам'яті, байт	Об'єм оперативної пам'яті, байт
Без додаткових бібліотек	716818	38664
З бібліотекою ArduinoJson	719006	38664
З MQTT клієнтом	707454	38460

Для прийому JSON запитів у Raspberry PI було встановлено бібліотеку Flask v2.3, а для обміну даними за MQTT протоколом встановлено mosquitto брокер. Для Flask серверу різниці в прийомі запитів, що були створені за допомогою спеціалізованої бібліотеки та без неї немає, бо в обох випадках сервер приймає запити, відповідні стандарту. Для прийому даних за допомогою протоколу MQTT у Python також було встановлено raHo-mqtt клієнт v1.6.1. Результати встановлення бібліотеки та серверу наведені у таблиці 2

Таблиця 2 – Характеристики встановлено у Raspberry PI ПЗ

Програмне забезпечення	Об'єм flash пам'яті, кбайт
Flask v2.3	2290
mosquitto брокер	37800
raHo-mqtt клієнт	416

Висновки. Передавання даних за допомогою протоколу MQTT у модулях ESP32 потребує мінімального об'єму оперативної та flash пам'яті. Виграш в потрібній кількості flash пам'яті у порівнянні з протоколом HTTP дорівнює 0,22% від загального об'єму, а в оперативній пам'яті – 0,04%. При цьому найбільшу простоту формування структури даних надає бібліотека ArduinoJson.

Відносний виграш використання протоколу MQTT незначний. Тому все залежить від необхідності створення на Raspberry PI веб серверу. При

потребі реалізації веб інтерфейсу спроектованої системи немає сенсу у додатковому протоколі MQTT, що потребує запуск додаткового сервісу на апаратній платформі Raspberry PI.

Список використаних джерел:

1. Neil C. Electronics Projects with the ESP8266 and ESP32: Building Web Pages, Applications, and WiFi Enabled Devices 1st ed. Edition/ C.Nail – New York: Apress, – 2020. – 714 p.
2. Monk S. Raspberry Pi Cookbook. Software and Hardware Problems and Solutions/ S.Monk – Manchester: O'Reilly Media, – 2022. – 618 p.
3. Platform for hands-on remote labs based on the ESP32 and NOD-red / N. Abekiri [та ін.] // Scientific African. - 2023. - № 19. - С. 1-14.
4. Architecture Analysis, Cloud-Based Using MQTTProtocol for Braille Literacy / G. Josimar dos santos [та ін.]; Federal University of Sergipe. Communication Papers of the of the 18th Conference on ComputerScience and Intelligence Systems - Warsaw, 2023. - 7 с.
5. Зубков О. В. Особливості реалізації Web серверів на модулях ESP8266 ТА ESP32 У Arduino IDE / О. В. Зубков, А. О. Зубков // IV форум «Автоматизація, електроніка та робототехніка. Стратегії розвитку та інноваційні технології» AERT-2022. – Харків, ХНУРЕ, 2022. – С. 54-55.