

АНАЛІЗ ТЕХНОЛОГІЙ ВІДДАЛЕНОГО ДОСТУПУ ДО ПРИСТРОЇВ НА МІКРОКОНТРОЛЕРАХ

к.т.н., доцент Зубков О.В., викладач Олійник В.В., студентка Павлій С.С.

Харківський національний університет радіоелектроніки,
кафедра медіаінженерії та інформаційних радіоелектронних систем,
м. Харків, Україна
e-mail: oleh.zubkov@nure.ua

Abstract. An analysis of modern remote access technologies for monitoring the status and control of modern electronic devices based on microcontrollers was carried out. The economic and technical infeasibility of deploying embedded servers based on microcontrollers when accessed from the Internet is shown. An analysis of modern Docker technology for applications containerization and their deployment on local or cloud servers was carried out. The effectiveness of the technology has been proven by the example of deploying a web application for monitoring and controlling a home heating system.

Ключові слова: мікроконтролер, віддалений доступ, протокол, мережа.

Вступ. Віддалений доступ до пристроїв на мікроконтролерах стає все більш актуальним і важливим з розвитком Інтернету речей (IoT) та потребою віддаленого керування вбудованими системами [1]. Віддалений доступ особливо актуальний для вирішення наступного кола проблем:

1. Моніторинг та управління постійно зростаючою кількістю підключених пристроїв і систем в рамках Інтернету речей, промисловості, медичних пристроїв та інше.

3. Для віддаленого налагодження, діагностики помилок та підтримки користувачів.

4. Націлений віддалений доступ може забезпечувати захист та безпеку, забороняючи несанкціонований доступ до систем та дозволяючи контролювати права доступу.

5. Можливість отримати швидкий доступ до вбудованих систем для реагування на події або зміну параметрів в реальному часі.

У цілому, актуальність віддаленого доступу до пристроїв на мікроконтролерах стає все більш важливою з поглибленням цифрової трансформації у багатьох галузях, де важливо не лише зібрати дані, а й здійснювати контроль та управління пристроями здалеку.

Основна частина. На апаратному рівні віддалений доступ означає поєднання мережних інтерфейсів з підтримкою сучасних протоколів передавання даних та вбудованих чи хмарних серверних технологій. Хоча сучасні мікроконтролери мають багато інтерфейсів передавання даних (USB, CAN, USART та ін.), але для глобального дистанційного доступу в

основному використовують інтерфейс дротового Ethernet з'єднання або бездротову технологію WiFi. В обох цих інтерфейсах на комунікаційно-програмному рівні передавання даних забезпечують протоколи: HTTP (Hypertext Transfer Protocol), MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), SSH (Secure Shell), Telnet, SNMP (Simple Network Management Protocol) [2]. Найпростішим рішенням дистанційного доступу є розгортання вбудованого у сам пристрій серверу, наприклад WEB серверу. Для популярних мікроконтролерів є багато готових бібліотек, що дозволяють у короткі строки вбудувати у програмний код backend частину веб сайту чи інше програмне рішення. Наприклад, для популярних WiFi модулів ESP32 із вбудованими мікроконтролерами загального призначення широко використовуються такі синхронні та асинхронні сервери, як WebServer та ESPAsyncWebServer [3]. Так можна забезпечити доступ комп'ютерам або смартфонам до веб сайту такого пристрою із локальної мережі і здійснювати локальний моніторинг або керування. При цьому глобальний доступ обмежується архітектурою сучасних мереж. В основі мережних протоколів передачі даних лежить IP адресація електронних пристроїв [1]. Пристрої локальної мережі мають внутрішню IP адресацію, яка невідома для глобальної мережі, а роутер, що поєднує локальну і глобальну мережу зазвичай має динамічну IP адресу, що видає йому провайдер. Навіть, якщо провайдер видає статичну IP адресу, за якою можна зайти до нього із мережі Internet, то для маршрутизації на веб сторінку електронного пристрою необхідно виконати додаткові налаштування роутеру. Користувачі, що купують такі пристрої не мають необхідної кваліфікації для цього, а використання сервісних спеціалістів коштує дорого і економічно необґрунтовано. Саме тому останні роки поширюється використання хмарних технологій і серверів, на яких розміщуються спеціальні додатки, що приймають інформацію із сукупності електронних пристроїв та передають її за запитом до клієнтських додатків на персональні комп'ютери, смартфони і т.д.

На початку 2000х років практично усі фірми виробники обладнання з дистанційним доступом закупали апаратні сервери, адміністрували їх та розташовували один або декілька програмних серверів з їх додатками [4]. Такі крупні компанії як Samsung, Xiaomi і зараз мають власні сервери, а менші компанії намагаються скоротити витрати на покупку обладнання, заробітню платню адміністраторів і віддають перевагу розташуванню серверів з додатками на сторонніх хмарних серверах та оплачують орендну платню за експлуатацію ресурсів сервера.

Таким чином на хмарних серверах повинні одночасно існувати десятки, сотні, мільйони незалежних додатків. Їх виконання забезпечується завдяки сучасній технології віртуалізації Hyper-V [5]. Спочатку на одному фізичному сервері створювали та запускали декілька віртуальних машин із

своїми операційними системами. Але кожна операційна система потребує багато ресурсів: пам'ять, обчислювальна потужність (ядра) процесора. Тому ця технологія мала високу вартість. Зараз використовують нові підходи до віртуалізації. Docker - це платформа для розробки, доставки та запуску програмного забезпечення відокремленими контейнерами [6]. Ця технологія дозволяє упаковувати програми та їх залежності в стандартизовані контейнери, які можуть бути запуснені на будь-якому комп'ютері, що підтримує Docker, без будь-яких змін. Основними компонентами Docker є: Docker Engine - рішення для створення та управління контейнерами. Він включає в себе сервер, який запускає та управляє контейнерами, та набір інструментів для їх створення та керування; Docker Image - це шаблон або темплейт, який використовується для створення контейнерів. Він містить всі необхідні файли та налаштування для запуску програми або сервісу; Docker Container - це інстанція Docker Image, яка виконується та працює на операційній системі. Контейнер є ізольованим середовищем, в якому виконується програма або сервіс, і містить усі необхідні компоненти для його роботи; Docker Hub - це хмарна платформа для зберігання та обміну Docker Images. Основними переваги використання Docker є: портативність, ізоляція та легковагість (від декількох мегабайт до сотен мегабайт), швидкість розгортання.

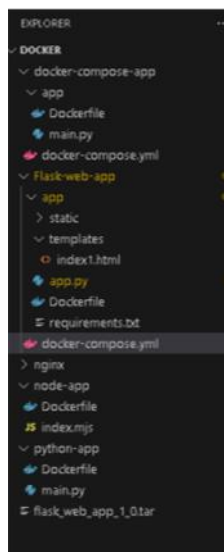


Рисунок 1 – Структура проекту для створення образу Docker

Для проведення дослідження ефективності платформи Docker у якості апаратної платформи був обраний модуль ESP32. До модуля ESP32 було підключено датчик температури DS18B20 та релейний модуль для керування системою нагріву приміщення. Вмикання та вимикання нагрівача здійснюється за відповідними температурними порогоми, початкові значення яких задані у базовій прошивці модуля і можуть бути переналаштовані за допомогою web додатку. Програмне забезпечення було

розроблено мовою C++. Для обміну даними між пристроєм та web додатком використано бібліотеку HTTP клієнту.

Web додаток було реалізовано мовою Python із застосуванням мікрофреймверку Flask для розгортання самого додатку та контенту сайту (web сторінки, графічні файли, css файли). Для створення проекту web додатку використовувалось середовище VS Code 2023. У VS Code було інстальовано розширення Docker. Структура загального проекту наведена на рисунку 1. Папка Flask-web-app містить усі файли web додатку. Файл docker-compose.yaml містить опис процесу створення образу і поєднання порту web додатку у контейнері із зовнішнім портом. Файл Dockerfile описує процес створення контейнеру web додатку у середині образу і інсталяцію додаткових модулів (Python, Flask і т.д.) від яких залежить робота додатка. В подальшому робота додатка була протестована у сукупності із роботою модуля ESP32.

Висновки. В роботі були проаналізовані технічні аспекти проблеми дистанційного доступу до електронних пристроїв, а також шляхи їх вирішення. Аналіз сучасної технології віртуалізації Docker і її тестування з використанням одного із найпоширеніших мікроконтролерів ESP32 показала стабільність роботи із мінімумом використаних апаратних ресурсів сервера (розмір образу 58 Мбайт).

Список використаних джерел.

1. Sachin Kumar, Prayag Tiwari & Mikhail Zymbler Internet of Things is a revolutionary approach for future technology enhancement: a review Journal of Big Data Vol. 6, 2019, pp.1-21
2. Сидни Фейт. TCP / IP. Архітектура. Протоколи. Реалізація. – К. вид-во Лори, 2014, 424 с.
3. Зубков О.В., Зубков А.О. Особливості реалізації web серверів на модулях ESP8266 та ESP32 у Arduino IDE AERT-2022, с.8-11
4. Rob Barrett, Eser Kandogan, Paul P. Maglio, Eben M. Haber, Leila Takayama Field studies of computer system administrators: Analysis of system management tools and practices roceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, CSCW 2004, Chicago, Illinois, USA, November 6-10, 2004
5. Dandu, Sujitha, "Data Center Server Virtualization Solution Using Microsoft Hyper-V" (2017).Culminating Projects in Information Assurance. 23. https://repository.stcloudstate.edu/msia_etds/23?utm_source=repository.stcloudstate.edu%2Fmsia_etds%2F23&utm_medium=PDF&utm_campaign=PDFCoverPages
6. Minh Thanh Chung, Nguyen Quang-Hung, Manh-Thin Nguyen, Nam Thoai Using Docker in High Performance Computing Applications 2016 IEEE Sixth International Conference on Communications and Electronics pp.52-57