ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ WEB СЕРВЕРІВ НА МОДУЛЯХ ESP8266 ТА ESP32 У ARDUINO IDE

доцент, к.т.н. Зубков О.В., студент Зубков А.О.
Харківський національний університет радіоелектроніки,
кафедра МТС, м. Харків, Україна
e-mail: oleh.zubkov@nure.ua, artem.zubkov@nure.ua

Abstract. The shortcomings of the Web servers implementation based on ESP8266 and ESP32 modules are considered. The choice of file system for storing Web site content is substantiated. The advantages and disadvantages of using the AsyncWebServer and WebServer libraries to implement Web servers in the Arduino IDE in the ESP8266 and ESP32 modules are determined. Measurements of the loading web pages with content speed for different modifications of modules and using different libraries of Web servers were performed. Practical recommendations are given to the principles of updating information on a Web page to achieve maximum performance. The disadvantages of using the ArduinoJSON library on dual-core processors are revealed.

Вступ. Останні роки характеризуються значним ростом ринку IoT речей та систем автоматизації. Сучасна промисловість може дозволити вкладати великі кошти у створення систем автоматизації на брендовому обладнанні таких виробників, як Siemens, ABB, GE Fanuc і т.д. Однак однією з головних вимог до систем автоматизації приватного житла, малого бізнесу є невелика ціна таких систем. Саме тому все більше проектів реалізується на сучасних WiFi модулях ESP8266 та ESP32 [1, 2]. Обидва модулі мають чіпи з підтримкою сучасних стандартів WiFi і можуть конфігуруватися, як точка доступу або клієнт WiFi мережі на базі роутера, вони мають цифрові лінії для підключення різних датчиків та сухих контактів, багато сучасних інтерфейсів для взаємодії з будь-якою периферією. Однак першій модуль має одноядерний процесор із тактовою частотою 80МГц, а другий двохядерний із частотою 160-240МГц, більшу розрядність аналого-цифрового перетворювача, цифро-аналоговий перетворювач. Для початкового конфігурування модуля, що є складовою частиною автоматизованої системи, або керування за його допомогою цією системою необхідно реалізовувати Web сервер на базі такого модуля. Багато розробників електронної апаратури вважають, що з реалізацією серверу багато складнощей, він повільно працює і відмовляються від використання ESP8266 і ESP32 та використовують більш дорогі рішення. Тому, метою цього дослідження було знаходження рішень, що відповідають сучасним вимогам з розробки Web серверів та забезпечують високу швидкість роботи серверу на процесорах з малою тактовою частотою.

Основна частина. Розробка програмного забезпечення Web серверів складається з двох частин: backend (програмне забезпечення, що запускається на сервері – у нашому випадку у ESP8266 чи ESP32) та frontend (web сторінка, що передається користувачу за його запитом та відображується браузером) [3, 4]. Цю розробку виконують різні спеціалісти-розробники, які, навіть, можуть працювати окремо та незалежно, обговорюючи лише структуру сайту. Тобто контент сайту (web сторінки, графічні файли, css файли) повинен розроблятись і зберігатись окремо від головної керуючої програми серверу. Це скорочує розробку та дозволяє оновлювати будь-яку складову сайту незалежно від інших складових. Але у більшості прикладів проектів для ESP8266 та ESP32 у код backend складової вбудовується код web сторінки для збільшення швидкості роботи сайту і це є помилковим рішенням. Для вирішення цієї проблеми необхідно розділяти загальну пам'ять модулів ESP8266 та ESP32 на дві частини: 1) для зберігання керуючої програми; 2) для складових frontend частини із реалізацією файлової системи. Найбільш популярними файловими системами є SPIFFS, litleFS, FatFS. Хоча файлову систему SPIFFS позиціонують, як спеціально розроблену для використання в ESP8266 та ESP32, але її супровід з боку розробників припинений ще у 2019 році. Експериментальна перевірка показала, що у модулях ESP8266 така система реалізується без проблем, а у останніх версіях модулів ESP32 виникають проблеми із записом файлів у створений розділ SPIFFS. На відміну від SPIFFS файлова система FatFS однаково добре реалізується та функціонує у всіх сучасних модифікаціях ESP8266 та ESP32, що підтверджує експериментальна перевірка. Завантаження файлів у створену файлову систему може відбуватись з використанням спеціалізованих додаткових інструментів (наприклад, ESP32 Sketch Data Uploader у Arduino IDE) або за допомогою вбудованого у серверну програму Ftp серверу.

Найчастіше для розробки програмного забезпечення для ESP8266 та ESP32 використовується середовище Arduino IDE у сукупності із бібліотеками реалізації серверів AsyncWebServer та WebServer. Обидва сервери дозволяють за запитом клієнту відкрити необхідний .html, .css і т.д. файли сайту, що зберігаються у розділі із файловими системами SPIFFS, litleFS, FatFS. Перевагами AsyncWebServer є: обробка запитів клієнтів за подією та можливість заповнювати шаблони html сторінок поточними даними перед відправкою клієнту. Це дає можливість створювати програми великого об’єму у головному нескінченому циклу не затримуючи обробку запитів клієнтів. Але головним недоліком є необхідність пропису реакції на запит завантаження будь-якого файла на рівні керуючої серверної програми. Тобто при зміні контенту web сайту необхідно вносити зміни і в серверну частину для додавання завантаження нових сторінок, графічних файлів і т.д. При використанні бібліотеки WebServer є можливість потокового завантаження будь-якого файлу з

файлової системи. Тобто можна створити універсальну функцію потокового зчитування будь-якого файлу контенту web сайту і при додаванні нових файлів чи вилученні існуючих немає потреби вносити зміни у існуючий код серверної частини програмного забезпечення. Головним недоліком бібліотеки WebServer є необхідність створювати нескінчений цикл і у ньому перевіряти – чи не надійшли нові запити від користувачів на завантаження чи оновлення web сторінок або їх контенту. Це призводить к обмеженню об'єму, а відповідно і часу виконання, програми, що розташовується у нескінченому циклу разом із частиною коду перевірки нових запитів від клієнтів. Тобто використання бібліотеки WebServer відповідає сучасним вимогам до розробки backend та frontend частин сайту, але для модулів ESP8266 його використання ефективне лише у випадку невеликого часу (декілька мілісекунд) виконання програми, що розташована у нескінченому циклі. Для модулів ESP32 таких обмежень не існує, бо тут використовується двохядерний процесор, у якому одне з ядер можна задіяти для обробки запитів клієнтів, а інше для реалізації головного керуючого алгоритму. Для експериментальної перевірки ефективності роботи бібліотек AsyncWebServer та WebServer було створено серверну та клієнтську частини сайту із завантаженням не тільки html сторінок, але й з додатковими файлами css, js і завантаженням двох графічних зображень у форматі jpg. При використанні AsyncWebServer та ESP32 час первинного завантаження сторінки був 0,9с, а при використанні WebServer 0,4с. При використанні модулів ESP8266 час збільшився у 1,8 рази.

При розробці frontend частини сайту також слід приділяти значну увагу вимогам до розробки web сторінок. Після першого завантаження сторінки оновлення зображень на сторінці, а також показань датчиків, стану системи можна реалізувати із оновленням всієї сторінки цілком, або лише окремих її елементів. Оновлення усієї сторінки спрощує розробку, але кожне перезавантаження сторінки потребує часу, що відповідає часу початкового завантаження сторінки. Реалізація оновлення стану лише деяких елементів web сторінки потребує написання додаткових скриптів на сторінці і відповідних функцій передавання серверною частиною короткого пакету інформації, але значно зменшує час оновлення, значення котрого не перевищує 0,1с у практичних прикладах. Також при написанні коду html сторінки можна значно скоротити час завантаження зображень, коли, наприклад, декілька зображень відповідають одному елементу сторінки та відображають різні стани цього елементу. Якщо при першому завантаженні сторінки завантажувати графічні зображення, що містять одночасно графічний вигляд усіх станів деякого елементу системи, а потім показувати на сторінці лише частину цього зображення, що відповідає поточному стану елементу сторінки, то немає необхідності знов зчитувати це зображення з серверу, а достатньо користуватись зображенням, що вже

є. Для реалізації такого підходу достатньо прописати клас для елементу сторінки та у java скрипті забезпечити зміну зображення. При передаванні даних для оновлення елементів web сторінки від сервера до клієнта дуже популярною є бібліотека ArduinoJSON. Але, як показали практичні досліди, у випадку використання двохядерних процесорів і одночасного доступу із підпрограм різних ядер до загальної структури JSON, виникають помилки запису і цю бібліотеку не слід використовувати.

Останнім важливим елементом реалізації сайту є вибір між реалізацією синхронних чи асинхронних запитів до серверу. У стандартних сайтах, що не керують автоматизованими системами, використовують асинхронні запити, тобто користувач, наприклад, натиснувши кнопку, не очікує результату а може далі щось робити на сторінці. Такий підхід є невірним для багатьох систем автоматизації, бо лише після виконання однієї команди у автоматизованій системі можна виконувати наступну і користувач повинен очікувати завершення виконання його попередньої команди. Синхронні запити дозволяють вирішити ці запити.

Висновки. Реалізація сучасних web серверів на модулях ESP8266 та ESP32 в середовищі Arduino IDE повинна реалізовуватись з розміщенням контенту сайту у розділі з файловою системою FatFS, що дозволяє реалізувати роздільний підхід до розробки backend та frontend частин програмного забезпечення. Виходячи з цієї умов також слід використовувати бібліотек WebServer для завантаження контенту сайту клієнту. Лише при значному часі виконання головної програми у ESP8266 слід використовувати бібліотеку AsyncWebServer. При оновленні інформації на сторінці у клієнта слід активно використовувати JS, оновлюючи лише стан елементів сторінки.

Список використаних джерел.

1. Yogendra Singh Parihar Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products. ETIR, 2019, Vol.6, Issue 6, pp. 1085-1088.

2. T. Sandeep Rao, Pawan Pranay, Sriman Narayana, Yamunadhar Reddy, Sunil, Pawandeep Kaur ESP32 Based Implementation of Water Quality and Quantity Regulating System. Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security, Vol. 4, pp. 122-129.

3. Dlnya Abdulahad Aziz Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module International. Journal Of Scientific & Engineering Research, 2018, Volume 9, Issue 6, pp. 801-807.

4. P. Machesoa, S. Chisale, C. Dakab, N. Dzupirec, J. Mlathodand, D. Mukanyirigira Design of Standalone Asynchronous ESP32 Web-Server for Temperature and Humidity Monitoring. 7th International Conference on Advanced Computing and Communication Systems, 2021, pp. 76-79.