

РЕАЛІЗАЦІЯ ПРОТОТИПУ СЕРВЕРУ БІОМЕТРИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Коростельов А. С., магістр, e-mail: xom9k123@gmail.comВадурін К. О., аспірант, e-mail: kir3337@gmail.comГученко М. І., д.т.н., проф., e-mail: mykolaguchenko@gmail.com

Кременчуцький національний університет імені Михайла Остроградського

Актуальність дослідження. У кабінах літаків і гвинтокрилів, а також у відповідних симуляторах, існують багато приладів для відстеження різних параметрів польоту та стану літальних апаратів або їх моделей. Проте, відсутні ефективні засоби для діагностики концентрації та біологічних показників людей-операторів, як це має місце в інших системах, де рішення приймає людина-оператор. Вивчення поточних показників операторів дозволить надавати їм правильний біологічний зворотний зв'язок під час навчальних сеансів і контролювати їх стан та запобігати нещасним випадкам, пов'язаним з людським фактором, під час реальних польотів та інших процесів. Тому актуальним завданням є відстеження стану людей-операторів в реальному часі. Для ефективного збору, початкової обробки, структурування та збереження біометричних даних операторів, розроблено структуру спеціалізованої інформаційної системи, яка враховує різні сценарії використання, складність біометричного комплексу і задовольнятиме потреби як звичайного користувача, так і вимоги дослідника [1].

Мета досліджень. Метою даного етапу досліджень є реалізація прототипу центрального серверу біометричної інформаційної системи за восьми сценарієм передачі даних (напряму до центрального сервера), що дозволить зібрати первинний датасет з біометричного бездротового наручного датчика.

Основні матеріали досліджень. У попередніх дослідженнях розроблено структуру інформаційного обміну у схемі наручного бездротового датчика біометричного комплексу [2] та спроектовано за нею електричну принципово схему [3], за якою побудовано базовий прототип.

У даній роботі прототип пристрою запрограмовано тестовою програмою, що постійно надсилає випадкові значення на сервер для запису у базу даних у форматі JSON.

Тестова програма пристрою реалізована у Arduino IDE та працює за алгоритмом:

1. Включення бібліотек та налаштувань: ESP32WiFi, ESP32HTTPClient та TimeLib; вказівка інформації про Wi-Fi мережу (SSID та пароль); вказівка адреси сервера, до якого відправляються дані (serverAddress) та сервера, який повертає дату та час (dateServer).

2. Оголошення змінних:

- jsonData – рядок, який буде містити JSON-дані для відправки на сервер;
- currentDateTimeVar – рядок для збереження отриманої дати та часу з сервера;
- wifiClient – об'єкт клієнта Wi-Fi.

3. Функція getTimeFromServer(): 1) створюється об'єкт HTTP-клієнта (http); 2) виконує HTTP GET-запит на сервер для отримання дати та часу (dateServer); 3) перевіряється відповідь сервера та отримується дата та час; 4) результат зберігається в currentDateTimeVar.

4. Функція setup(): 1) ініціалізується з'єднання з Wi-Fi мережею; 2) викликається функція randomSeed(analogRead(0)) для ініціалізації генератора випадкових чисел.

5. Головний цикл loop(): 1) викликається функція sendDataToServer() для відправлення даних на сервер; 2) Затримка на 5 секунд для періодичного відправлення даних.

6. Функція sendDataToServer(): 1) викликається функція getTimeFromServer() для отримання поточної дати та часу; 2) створюється об'єкт HTTP-клієнта (http) для відправлення даних на сервер;

3) встановлюються заголовки HTTP-запиту для вказівки формату даних (JSON); 4) створюється рядок jsonData, який містить JSON-дані для відправки, включаючи інформацію

про `currentDateTimeVar`; 5) відправляється POST-запит на сервер із JSON-даними; 6) результат HTTP-запиту виводиться у консоль; 7) завершується HTTP-з'єднання.

При цьому алгоритм роботи спроектованого сервера наступний:

1. Встановлення необхідних бібліотек:

– у коді використовуються бібліотеки `Express`, `bodyParser`, `mysql` та `fs` (файлова система);

– параметри для підключення до бази даних та інші налаштування визначаються.

2. Налаштування Express-сервера: 1) створюється об'єкт `Express`; 2) визначається порт (4000), на якому сервер слухатиме запити; 3) визначається об'єкт конфігурації для підключення до бази даних (`dbConfig`); 4) встановлюються парсери для обробки JSON-даних та POST-даних; 5) встановлюється шлях до статичних файлів (HTML, CSS, JavaScript).

3. Обробка POST-запиту: 1) створюється POST-маршрут за шляхом `"/api"`, який очікує POST-запити від клієнта; 2) функція обробки POST-запиту отримує дані з запиту, виводить їх на консоль та викликає функцію `writingToDatabase` для збереження цих даних в базі даних; 3) Відповідь на клієнта містить надіслані дані.

4. Функція `writingToDatabase`: 1) створюється підключення до бази даних з використанням параметрів з `dbConfig`; 2) виконується з'єднання з базою даних; 3) визначається SQL-запит для перевірки наявності запису за ідентифікатором (`id`) та часом (`dtime`); 4) виконується запит до бази даних для перевірки наявності дублікатів за `id` та часом; 5) якщо запис відсутній, то створюється SQL-запит для вставки нового запису в базу даних; 6) виконується вставка запису в базу даних; 7) завершується з'єднання з базою даних.

Висновок. Розроблені програми виконують різні завдання та призначені для різних середовищ. Програма для ESP32, написана у Arduino IDE, призначена для збору даних та відправки їх на сервер. Вона використовує мікроконтролер ESP32 та бібліотеки ESP32WiFi, ESP32HTTPClient для роботи з Wi-Fi та HTTP, а також TimeLib для роботи з часом. Програма зчитує дані, включаючи дату та час, з сервера, і надсилає їх на сервер у форматі JSON. З іншого боку, серверна програма написана на Node.js з використанням фреймворку Express та бази даних MySQL. Її завдання – обробка POST-запитів від клієнта та зберігання надісланих даних в базі даних. Сервер встановлює з'єднання з базою даних, перевіряє наявність записів за ідентифікатором та часом, і вставляє нові дані в базу даних, якщо запис відсутній. Всі дані відображаються на консолі. Після завантаження даних у базу MySQL, вони можуть бути вивантажені за допомогою Workbench, для подальшого аналізу іншими програмними засобами. У подальшому планується розробка користувацького веб-інтерфейсу та реалізація функції вивантаження даних з бази через веб-клієнт.

ПЕРЕЛІК ПОСИЛАНЬ

1. Вадурін К. О., Перекрест А. Л., Кухаренко Д. В. Структура інформаційної системи обробки даних отриманих від біометричного комплексу для моніторингу, прогнозування та підтримки прийняття рішень людини-оператора. XIV Міжнародна науково-практична конференція «Комп'ютерні системи та мережні технології», 13-14 квітня 2023 р., м. Київ. С. 31–33. URL: <https://csnt.nau.edu.ua/files/2023/sbirnyk2023.pdf>

2. Вадурін К. О., Мосьпан Д. В., Юрко О. О. Розробка структури інформаційного обміну у схемі ручного бездротового датчика біометричного комплексу моніторингу стану людини-оператора. XIV Міжнародна науково-практична конференція «Комп'ютерні системи та мережні технології», 13-14 квітня 2023 р., м. Київ. С. 25–27. URL: <https://csnt.nau.edu.ua/files/2023/sbirnyk2023.pdf>

3. Борох К. С., Вадурін К. О., Кухаренко Д. В. Проектування електричної принципової схеми ручного бездротового датчика біометричного комплексу моніторингу стану людини-оператора. XXX Міжнародна науково-практична конференція студентів, аспірантів та молодих учених КрНУ імені Михайла Остроградського «Актуальні проблеми життєдіяльності суспільства», 20-21 квітня 2023 р., м. Кременчук. С. 55–56. URL: http://apgs.kdu.edu.ua/zbirnik_apgs_2023.pdf