

## РОЗРОБЛЕННЯ ВЕБ-СТОРІНКИ КЕРУВАННЯ МОБІЛЬНИМ РОБОТОМ ЧЕРЕЗ ПРОТОКОЛ MQTT

**Т.А. Лихо**

Харківський національний університет радіоелектроніки

Україна, 62433, Люботин, вул. Шмідта, 9

E-mail: [tymur.lykho@nure.ua](mailto:tymur.lykho@nure.ua)

**Анотація:** У даній статті розглянуто основні аспекти створення веб-інтерфейсу для керування роботизованою платформою відеонагляду на базі ESP32-CAM. В результаті роботи будемо проведемо тестування взаємозв'язку робота та інтерфейсу користувача через протокол MQTT.

**Ключові слова:** Мобільний Робот, Відеонагляд, Веб-Сторінка, Інтерфейс, Протокол Mqtt, Інновація Процесу

## DEVELOPMENT OF A WEB PAGE FOR CONTROLLING A MOBILE ROBOT VIA MQTT PROTOCOL

**T. Lykho**

Kharkiv National University of Radio Electronics

Ukraine, 62433, Liubotyn, Schmidt Str., 9

E-mail: [tymur.lykho@nure.ua](mailto:tymur.lykho@nure.ua)

**Annotation:** This article explores the main aspects of creating a web interface for controlling a robotic surveillance platform based on ESP32-CAM. As a result of the work, we will conduct testing of the interaction between the robot and the user interface via the MQTT protocol.

**Keywords:** Mobile Robot, Surveillance, Web Page, Interface, Mqtt Protocol, Process Innovation

У наші часи все більше набирають обертів системи охоронної та/або пожежної сигналізації, «розумний дім» і подібні. Альтернативним варіантом системи відеонагляду є мобільний робот з камерою на базі ESP32-CAM.

Для керування таким роботом необхідна зручна веб-сторінка, яка буде мати всі необхідні елементи керування роботом та вікно для відображення відеопотоку з камери, розроблення такої сторінки буде розглянуто у даній роботі.

MQTT - це стандартизований протокол або набір правил для обміну повідомленнями, який використовується для взаємодії між комп'ютерами. Інтелектуальні датчики, носимі пристрої та інші пристрої Інтернету речей (IoT) зазвичай передають та отримують дані через мережі з обмеженими ресурсами та пропускну здатністю. До основних переваг протоколу MQTT відносять його простоту та ефективність, масштабованість, надійність та безпечність.

Протокол MQTT працює на принципах моделі «видавець/підписник». У традиційній мережевій комунікації клієнти та сервери спілкуються один з одним безпосередньо. Клієнти запитують ресурси або дані від сервера, після чого сервер обробляє та відправляє відповідь. Проте MQTT використовує модель видавець/підписник для роз'єднання відправника повідомлення (видавця) від отримувача повідомлення (підписника). Замість цього третій компонент, який називається мережним брокером, відповідає за зв'язок між видавцями та підписниками. Завдання брокера полягає в фільтрації всіх вхідних повідомлень від видавців та їх правильному розподілі підписникам. Брокер роз'єднує видавців та підписників за наступними принципами:

- просторове роз'єднання, тобто видавець та підписник не мають інформації про мережеве розташування один одного і не обмінюються такими даними, як IP-адреси або номери портів;

- часове роз'єднання, за цим принципом видавець на підписник не використовують, або не мають одночасного мережевого з'єднання;

– роз'єднання синхронізації, , тобто видавці та підписники можуть відправляти та отримувати повідомлення без перешкод один одному, що забезпечує асинхронність. [1]

Таким чином протокол MQTT задовольняє всім потребам проекту мобільного робота відеонагляду.

Для керування роботом на отримання відеопотоку з камери робота необхідно розробити веб-сторінку, що буде містити основні елементи керування мобільним роботом, а саме:

- кнопки напрямку руху мобільної платформи;
- регулятор швидкості руху мобільної платформи;
- кнопки керування механізмом нахилу та панорамування;
- регулятор яскравості спалаху камери;
- вікно із відопотоком з камери робота.

За наведеними вимогами до веб-інтерфейсу робота було створено веб-сторінку з використанням HTML та CSS, яку наведено на рисунку 1.



Рисунок 1 – Вигляд веб-інтерфейсу роботизованої платформи відеонагляду

Розглянемо елементи що містяться на сторінці керування роботом. Посередині маємо вікно, у якому буде відображатись відеопотік з камери робота, у разі, якщо зображення не надходить буде виводитись картинка «Немає зв'язку», як показано на рисунку 1. З лівої сторони бачимо клавіші керування механізмом нахилу та панорамування, а саме кнопки нахилу «вверх» та «вниз», кнопки повороту камери «в ліво» та «в право», а також центральну кнопку, що встановлює центральне положення камери. У правій частині сторінки бачимо кнопки для керування роботизованою платформою, а саме кнопки «вперед», «назад», «в ліво», «в право» та «стоп». Також було додано регулятор яскравості спалаху камери, регулятор швидкості руху платформи та випадючий список, за допомогою якого користувач буде мати можливість змінювати роздільну здатність відеопотоку, що у ситуаціях поганого Інтернет-з'єднання дозволить отримати більшу кількість кадрів.

1. Основні функції для керування роботом з веб-інтерфейсу через протокол MQTT.

Після розробки інтерфейсу робота відеонагляду постає задача як передавати та отримувати данні з веб-сторінки до робота і навпаки. Для вирішення цієї задачі у першій частині цієї статті було обрано протокол MQTT через його зручність та простоту у використанні.

Для написання логічної частини веб-сторінки будемо використовувати мову програмування JavaScript, бо ця мова програмування дозволяє легко реалізувати інтерактивні і динамічні елементи інтерфейсу, при чому логічна частина може знаходитись прямо у HTML-документі, що дозволяє використовувати веб-інтерфейс без розгортання на сервері.[2]

Тож тепер можна перейти до розгляду функцій, що реалізують передачу даних між інтерфейсом та роботом.

Для початку необхідним є підключення до MQTT-брокера, це можна реалізувати, наприклад, з використанням популярної бібліотеки Paho, щоб все працювало коректно її необхідно підключити до нашого HTML-документу за допомогою такої конструкції:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js" type="text/javascript" ></script>. [3]
```

Тепер необхідно написати код підключення до MQTT-брокера, підписку на необхідні «topics» та відслідковування помилок підключення, для цього було написано такий код:

```
var client = new Paho.MQTT.Client("broker.mqttdashboard.com", 8000, "clientId" + new Date().getTime());
client.onConnectionLost = function(responseObject) {
    if (responseObject.errorCode !== 0) {
        console.log("Connection lost:", responseObject.errorMessage);
    }
};
client.connect({
    onSuccess: function() {
        console.log("Connected to MQTT server");
        client.subscribe("camera/test");
    }
});
```

Передача відеопотоку у даному випадку відбувається за допомогою кодування кожного кадру у формат BASE64, що дозволяє перетворити зображення у текстовий вигляд і передати через протокол MQTT. Для того щоб отримати і відобразити потік кадрів на сторінці необхідно додати таку функцію:

```
client.onMessageArrived = function(message) {
    var imageData = message.payloadString;
    if (imageData !== undefined && imageData.length > 0) {
        $("#streamedImage").attr("src", imageData);
    }
};
```

Ця функція перевіряє чи містяться данні у топіку «camera/test», якщо так, тоді данні передаються у HTML-елемент `img` з ідентифікатором «streamedImage».

Завдяки тому, що браузері можуть автоматично розкодувати формат BASE64 немає необхідності у написанні додаткових функцій декодування кадрів.

Також є необхідність відправляти дані роботу, для цього зручним є використання такої функції:

```
function sendCommand(topic, command) {
    var message = new Paho.MQTT.Message(command);
    message.destinationName = topic;
    client.send(message);
};
```

Таким чином ми можемо скоротити написання коду, наприклад, виклик команди на переміщення робота або управління механізмом нахилу та панорамування буде виглядати так:

```
$(".button, .buttonDrive").mousedown(function() {  
    var command = $(this).data("command");  
    sendCommand(keyboardTopic, command);  
});
```

Функція працює таким чином: коли користувач натискає на одну з кнопок відправляються відповідні дані у топик «robot/keyboard», який було збережено у окрему змінну keyboardTopic для читабельності коду, а коли користувач відпускає кнопку виконується функція, яка зупиняє рух робота автоматично:

```
$(".buttonDrive").mouseup(function() {  
    sendCommand(keyboardTopic, "driveStop");  
});
```

За таким самим принципом реалізовано і функції для керування швидкістю переміщення робота та яскравістю спалаху камери:

```
$("#speedRange").change(function() {  
    var speed = $(this).val();  
    sendCommand(keyboardTopic, "speed:" + speed);  
});  
$("#brightnessRange").change(function() {  
    var brightness = $(this).val();  
    sendCommand(flashTopic, "brightness:" + brightness);  
});
```

Наведені вище функції відслідковують положення регулятора з відповідним ідентифікатором та передають значення положень у випадку, коли положення регулятора змінюється.

Зміну роздільної здатності відеопотоку реалізовано наступною функцією:

```
$(".dropdown-item").click(function() {  
    var quality = $(this).data("quality");  
    sendCommand(quantityTopic, "quality:" + quality);  
});
```

## 2. Тестування взаємозв'язку інтерфейсу та робота

Для проведення тестування роботи інтерфейсу користувача для управління основними функціями роботизованої платформи відеонагляду плату ESP32-CAM було підключено до COM-порту комп'ютеру. На рисунку 2 наведено вигляд робочого інтерфейсу та монітор порту у який виводяться данні про натиснуті кнопки.

В результаті проведення дослідів можна зробити висновок, що взаємозв'язок між інтерфейсом та роботизованою платформою відеонагляду за допомогою протоколу MQTT реалізовано коректно і задовольняє потребам проекту.

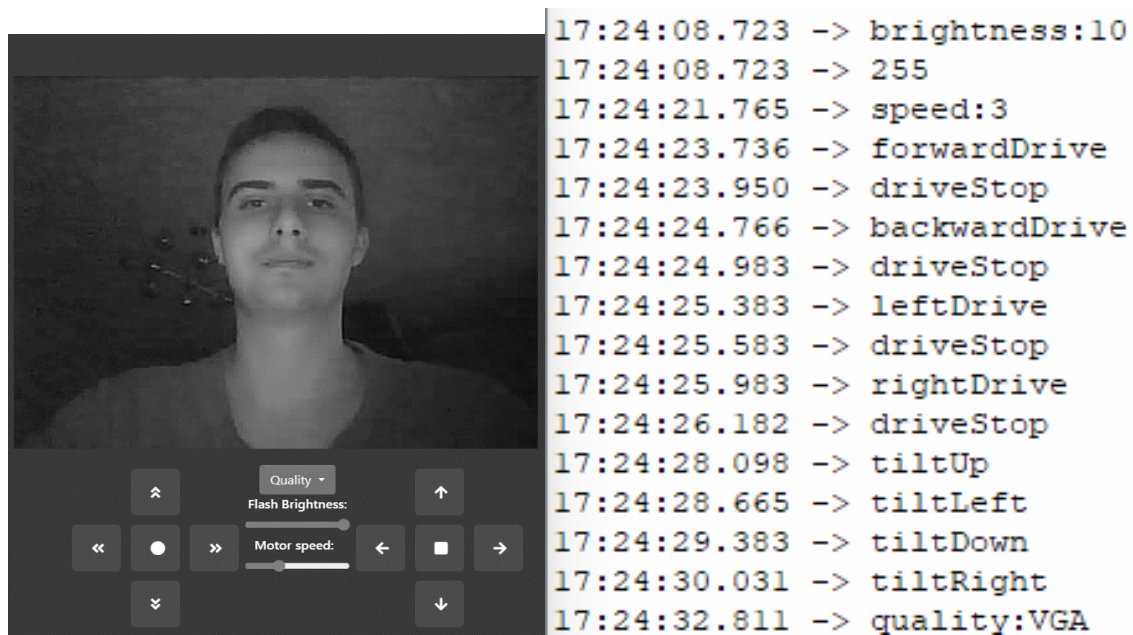


Рисунок 2 – Результат проведення тестування роботи веб-інтерфейсу

ВИСНОВКИ. Основні результати досліджень:

- визначено основні переваги MQTT-протоколу для використання у даному проєкті;
- розроблено інтерфейс користувача для керування роботизованою платформою відеонагляду за допомогою HTML, CSS;
- використовуючи мову програмування JavaScript реалізовано логіку інтерфейсу;
- проведено тестування роботи розробленого інтерфейсу та зв'язку з роботизованою платформою.

Практична цінність:

- застосування мобільного робота відеонагляду актуально в умовах війни, коли безпека стає пріоритетом, і багато людей покидають свої будинки;
- використання дистанційного керування та можливості перегляду зображення у реальному часі робить систему зручною та функціональною.

Подальші перспективи:

- можливість розширення функціоналу веб-застосунку, наприклад, додавання нових роботів, реєстрація та авторизація, можливість зберігати певні кадри з камери робота, тощо;
- інтеграція з системами «розумного дому» для більш широкого застосування в повсякденному житті.

Загалом, розробка мобільного робота для відеонагляду має перспективи в різних сферах, привертає увагу своєю ефективністю та можливістю використання в різноманітних сценаріях.

## ЛІТЕРАТУРА

1. What is MQTT? - MQTT Protocol Explained - AWS. Amazon Web Services, Inc. URL: [https://aws.amazon.com/what-is/mqtt/?nc1=h\\_ls](https://aws.amazon.com/what-is/mqtt/?nc1=h_ls) (date of access: 25.03.2024).
2. Ivashchuk I. Чому JavaScript – перспективна мова програмування? Поради початківцям. Dou.ua. 2021. URL: <https://dou.ua/forums/topic/35184/> (дата звернення: 26.03.2024).
3. Eclipse paho | the eclipse foundation. Projects Gateway | The Eclipse Foundation. URL: <https://eclipse.dev/paho/index.php?page=clients/js/index.php> (date of access: 26.03.2024).
4. Akopov, M., Maksymova, S., & Yevsieiev, V. (2023). Choosing a Camera for 3D Mapping. Journal of Universal Science Research, 1(11), 28-38.
5. Nikitin, V., Maksymova, S., & Yevsieiev, V. (2023). Traffic Signs Recognition System Development. Multidisciplinary Journal of Science and Technology, 3(3), 235-242.

6. Nevliudov, I., Ponomaryova, G., Bortnikova, V., Maksymova, S., & Kolesnyk, K. (2018). MEMS accelerometer in hexapod intellectual control. In 2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH) (pp. 146-150). IEEE.
7. Yevsieiev, V., Maksymova, S., & Starodubcev, N. (2022). Development of an Algorithm for ESP32-Cam Operation in HTTP Server Mode for Streaming Video. *Collection of scientific papers «ΑΙΟΓΟΣ»*, (July 8, 2022; Paris, France), 177-179.
8. Attar, H., & et al.. (2022). Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. *Computational Intelligence and Neuroscience*, 2022, Article ID 3046116, <https://doi.org/10.1155/2022/3046116>.
9. Al-Sharo, Y., Abu-Jassar, A., Lyashenko, V., Yevsieiev, V., Maksymova, S. A Robo-hand prototype design gripping device within the framework of sustainable development, *Indian Journal of Engineering*, 20 2023 e37ije1673. <https://doi.org/10.54905/disssi.v20i54.e37ije1673>
10. Lyashenko, V., Abu-Jassar, A.T., Yevsieiev, V., Maksymova, S. Automated Monitoring and Visualization System in Production, *Int. Res. J. Multidiscip. Technovation*, 5(6) 2023 09-18. <https://doi.org/10.54392/irjmt2362>
11. Nevliudov, I., & et al.. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis. *International Journal of Emerging Trends in Engineering Research*, 8(10), 7465-7473.
12. Svitlana Maksymova, Vladyslav Yevsieiev, & Amer Abu-Jassar. (2024). Gripping Device Development: Some Aspects. *Journal of Universal Science Research*, 2(1), 150–158
13. Svitlana Maksymova, & Vladyslav Yevsieiev. (2024). Coin Counting Device Kinematic Diagram Development. *Journal of Universal Science Research*, 2(1), 159–168.
14. Svitlana Maksymova, Vladyslav Yevsieiev, & Ahmad Alkhalaileh. (2024). The Monitoring System Architecture Development. *Journal of Universal Science Research*, 2(1), 69–79.
15. Svitlana Maksymova, Vladyslav Yevsieiev, & Amer Abu-Jassar. (2024). Gripping Device Development: Some Aspects. *Journal of Universal Science Research*, 2(1), 150–158.
16. Svitlana Maksymova, & Vladyslav Yevsieiev. (2024). Coin Counting Device Kinematic Diagram Development. *Journal of Universal Science Research*, 2(1), 159–168.
17. Svitlana Maksymova, Vladyslav Yevsieiev, & Amer Abu-Jassar. (2024). The Bipedal Robot a Kinematic Diagram Development. *Journal of Universal Science Research*, 2(1), 6–17.
18. Andrii Bondariev, Svitlana Maksymova, & Vladyslav Yevsieiev. (2023). Automated Monitoring System Development for Equipment Modernization. *Journal of Universal Science Research*, 1(11), 6–16. Retrieved from <https://universalpublishings.com/index.php/jusr/article/view/2484>
19. Mykhailo Akopov, Svitlana Maksymova, & Vladyslav Yevsieiev. (2023). Choosing a Camera for 3D Mapping. *Journal of Universal Science Research*, 1(11), 28–38. Retrieved from <https://universalpublishings.com/index.php/jusr/article/view/2486>
20. Vladyslav Nikitin, Svitlana Maksymova, & Vladyslav Yevsieiev. (2023). TRAFFIC SIGNS RECOGNITION SYSTEM DEVELOPMENT . *Multidisciplinary Journal of Science and Technology*, 3(3), 235–242. Retrieved from <https://mjstjournal.com/index.php/mjst/article/view/225>
21. Yevsieiev, V. Comparative Analysis of the Characteristics of Mobile Robots and Collaboration Robots Within INDUSTRY 5.0. / V. Yevsieiev, D. Gurin // In the VI International Scientific and Theoretical Conference, September 8, 2023. Chicago, USA. P.92-94