

# РОЗРОБКА ІНТЕРАКТИВНОГО БОТА ДЛЯ КЕРУВАННЯ СИСТЕМОЮ РОЗУМНОГО ОСВІТЛЕННЯ ТА МОНІТОРИНГУ ЕНЕРГОСПОЖИВАННЯ НА БАЗІ PYTHON, SHELLY API ТА MQTT

Макаренко Д.В.

*Київський столичний університет імені Бориса Грінченка, м. Київ*

## ВСТУП

Сучасний розвиток технологій Інтернету речей (IoT), штучного інтелекту та концепції сталого розвитку робить питання енергоефективного управління освітленням та моніторингу енергоспоживання однією з ключових задач як у побутовій, так і в комерційній та муніципальній сферах.

**Актуальність і постановка проблеми.** Актуальність теми зумовлена одночасною дією кількох факторів: необхідністю суттєвого зниження енергоспоживання освітлювальних систем, зростанням попиту на локальні (без постійної хмарної залежності) та приватні рішення, розвитком концепції Energy Management as a Service, а також потребою в доступних, зрозумілих та легко масштабованих системах саме для українського споживача в умовах економічної та енергетичної нестабільності.

**Мета дослідження.** Розробити та дослідити інтерактивний програмний бот для керування системою розумного освітлення та моніторингу енергоспоживання на базі пристроїв Shelly з використанням Python, Shelly API та протоколу MQTT.

**Аналіз останніх досліджень і публікацій.** Ламія Альбрахім та співавтори у статті представляють систему моніторингу енергії на базі smart plug з інтеграцією MQTT для масштабованих IoT-розгортань, демонструючи точність вимірювань у межах  $\pm 5\%$  похибки, що підкреслює практичну ефективність для побутового використання [1]. Мохамед А. Ібрагім Ісмаїл та Мохаммед Аль-Дарааві у своїй роботі пропонують інтеграцію IoT з штучними нейронними мережами (ANN) у мобільний додаток для оптимізації енергоспоживання, з акцентом на сталості та зменшенні екологічного впливу завдяки аналізу даних у реальному часі [2]. Чен І Чен, Шен-Хань Ву, Бо-Вун Хуан, Чао-Хун Хуан та Чен-Фу Ян у статті описують IoT-систему для контролю освітлення та середовища на базі Raspberry Pi з використанням Node-RED та MQTT, що спрощує управління обладнанням і сприяє енергоефективності в лабораторних умовах [3]. Фатеме Дінмохаммаді, Махмуд Шафії та співавтори у публікації своїй презентують реальний IoT-моніторинг на Raspberry Pi для оптимізації енергії в будівлях, інтегруючи сенсори температури, світла та руху, і демонструють потенціал зменшення споживання через автоматизовані алгоритми [4]. Автори комплексного огляду надають

всесторонній аналіз IoT-систем розумного освітлення, акцентуючи на підвищенні енергоефективності, візуального комфорту та адаптивності, з рекомендаціями для майбутніх розробок [5].

### РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розумний будинок — це сукупність датчиків, контролерів та інших систем, здатних забезпечити віддалене керування системами або програмування заданих функцій [6]. Система розумного будинку поєднує різні пристрої та системи, щоб забезпечити автоматизацію та контроль різних аспектів життя в будинку [7].

У системах Інтернету речей (IoT), зокрема в «розумному будинку», для організації ефективного обміну даними між пристроями використовуються спеціалізовані протоколи зв'язку.

MQTT (Message Queuing Telemetry Transport) — це стандартний протокол обміну повідомленнями OASIS для Інтернету речей (IoT). Протокол працює за моделлю publish/subscribe (видавець/передплатник): пристрої (клієнти) публікують дані на певні теми (topics) через центральний брокер, а інші клієнти підписуються на ці теми для отримання повідомлень. Це дозволяє повністю розв'язати відправників і отримувачів [8].

Апаратна частина систем «розумний будинок» включає спеціалізовані IoT-пристрої, такі як реле, димери, лічильники енергії та розумні розетки, які забезпечують керування освітленням та точний моніторинг споживання електроенергії. У проекті використовуються пристрої бренду Shelly [9] — компактні Wi-Fi-модулі з підтримкою MQTT та API, що дозволяють локальне та віддалене керування без залежності від хмарних сервісів виробника.

Середовища проєктування та програмування для систем керування освітленням та моніторингу енергоспоживання в «розумному будинку» базуються на гнучких та ефективних інструментах, таких як мова програмування Python [10], яка є однією з найпопулярніших платформ для IoT-розробки завдяки своїй простоті, потужним бібліотекам та можливості швидкої інтеграції з апаратними компонентами.

Для роботи з MQTT у Python використовується бібліотека paho-mqtt, яка забезпечує підключення клієнта до брокера, підписку на теми, публікацію повідомлень тощо [11].

Для HTTP-взаємодії з API пристроїв, таких як Shelly, використовується бібліотека requests — елегантна та проста HTTP-бібліотека для Python [12].

Для реалізації клієнтської частини системи та створення інтерактивного чат-бота було обрано бібліотеку Aioogram. Це сучасний, повністю асинхронний фреймворк для Telegram Bot API, написаний на Python з використанням asyncio [13].

Серед популярних онлайн-сервісів проєктування та симуляції виділяється Wokwi — платформа для симуляції IoT-проєктів без апаратури, що підтримує Arduino, ESP32 та Raspberry Pi [14]. Він дозволяє симулювати IoT-проєкти в браузері, включаючи датчики, виконавчі пристрої та комунікацію MQTT, що ідеально для тестування логіки бота керування освітленням.

Основні причини вибору чат-бота для керування системою розумного освітлення та моніторингу енергоспоживання:

1. звичне середовище: більшість користувачів щодня використовують месенджери;
2. кросплатформеність: користувачу не потрібно встановлювати специфічне програмне забезпечення для кожного пристрою;
3. економія ресурсів смартфона: відсутність потреби у завантаженні окремих додатків для кожної розумної розетки чи лампи звільняє пам'ять пристрою та спрощує оновлення інтерфейсу (всі зміни відбуваються на сервері бота);
4. дистанційний доступ: завдяки хмарній природі месенджерів, користувач може керувати освітленням з будь-якої точки світу, де є інтернет.

Для реалізації даного проєкту було обрано Telegram, оскільки він забезпечує прямий зв'язок між Python-кодом та MQTT-брокером без зайвих посередників, що гарантує високу швидкість передачі команд та стабільність моніторингу енергоспоживання.

Архітектура розробленої системи базується на принципах хмарної взаємодії компонентів та подійному керуванні (event-driven architecture). Основним завданням архітектурного рішення є забезпечення безперебійного зв'язку між фізичним (або симульованим у Wokwi) пристроєм та мобільним інтерфейсом користувача в Telegram.

Система складається з трьох ключових рівнів:

1. рівень пристроїв (Hardware Level): представлений мікроконтролером ESP32 (симуляція Shelly 1PM у середовищі Wokwi). Він виконує роль виконавчого механізму (реле для освітлення) та сенсорного вузла, що обчислює споживану потужність та енергію;
2. рівень передачі даних (Network Level): використовує протокол MQTT та публічний брокер broker.emqx.io. Цей рівень забезпечує асинхронний обмін повідомленнями за моделлю «видавець-передплатник»;
3. рівень керування та інтерфейсу (Application Level): Серверна частина на базі Python, що об'єднує MQTT-клієнт та Telegram-бота (бібліотека aiogram). Бот виступає «тонким клієнтом», звільняючи користувача від необхідності встановлення локального ПЗ.

Нижче наведено діаграму компонентів, яка демонструє логічну структуру системи та зв'язки між програмними модулями, хмарним брокером та апаратною частиною (див. рис.1).

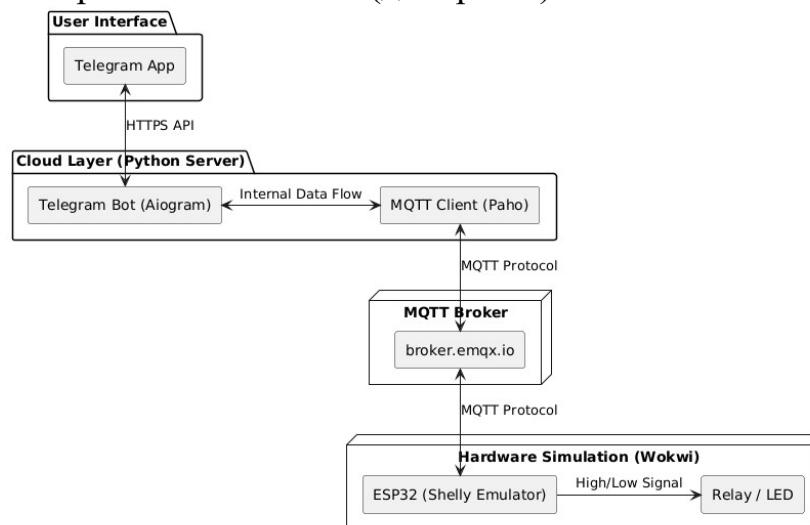


Рис.1 Архітектура системи

Схема призначена для демонстрації керування освітленням за допомогою мікроконтролера ESP32 через реле-модуль (див. рис.2). Вона імітує роботу реального пристрою Shelly, де реле виступає в ролі виконавчого елемента, а світлодіод — як навантаження (імітація лампи).

Склад схеми:

- мікроконтролер ESP32 — центральний елемент, який отримує команди через протокол MQTT і керує роботою реле;
- реле-модуль (5V Relay Module) — електронний ключ, який дозволяє безпечно керувати навантаженням (LED). Реле ізольоване від ESP32, що захищає мікроконтролер від можливих перешкод або вищих напруг;
- світлодіод (LED) — імітує освітлювальний прилад (лампу);
- обмежувальний резистор 220 Ом — захищає світлодіод від надмірного струму, запобігаючи його перегріву або виходу з ладу.

Принцип роботи схеми:

1. Керування реле:

- Сигнал керування надходить з GPIO5 ESP32 на контакт IN реле-модуля.
- Більшість реле-модулів мають низький рівень активації (Low Trigger): коли на IN подається логічний 0 (LOW), реле спрацьовує.
- При спрацьовуванні реле замикається внутрішній контакт між COM і NO (Normally Open).

2. Робота світлодіода:

- Коли реле спрацьовує (IN = LOW), струм проходить по ланцюгу: 5V → COM → NO → Анод LED → Резистор 220 Ом → GND
- Світлодіод загоряється.

– Коли реле відпускає (IN = HIGH), ланцюг розмикається — LED гасне.

### 3. Керування з бота (див.рис.3):

– Python-бот надсилає команди ("on" або "off") через протокол MQTT у топик shelly/light/cmd.

– ESP32, що працює в Wokwi, отримує ці команди і змінює стан GPIO5.

– Зміна стану GPIO5 призводить до спрацьовування/відпускання реле і, відповідно, увімкнення або вимкнення LED.

Режим роботи:

1. При отриманні команди "on" — реле спрацьовує → LED загоряється (див. рис.4).

2. При отриманні команди "off" — реле відпускає → LED гасне (див. рис.5).

Схема працює в реальному часі: команди з бота майже миттєво відображаються на схемі в Wokwi.

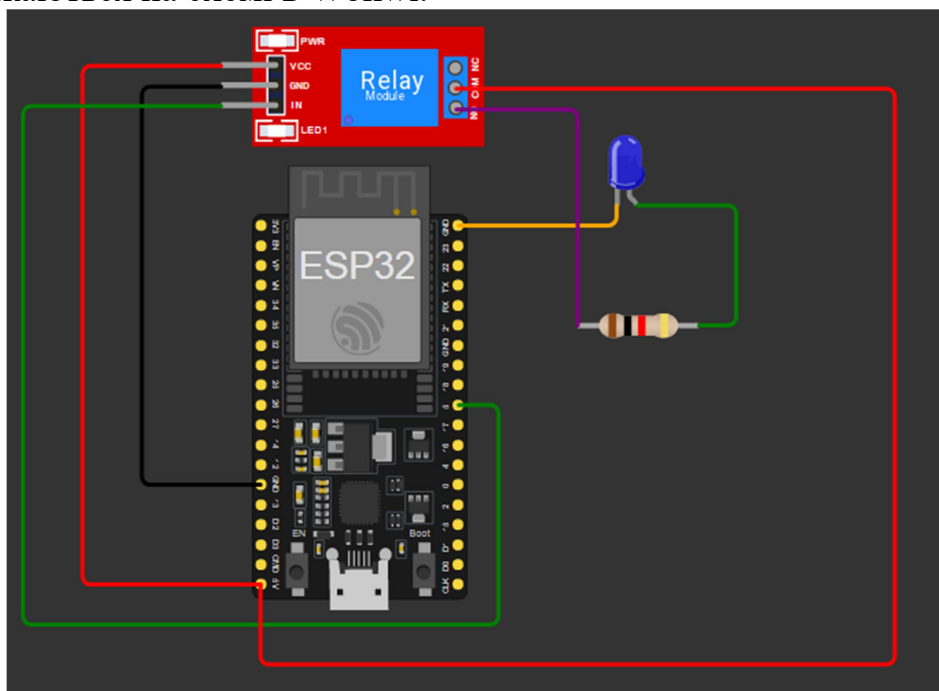


Рис. 2 Схема системи у Wokwi

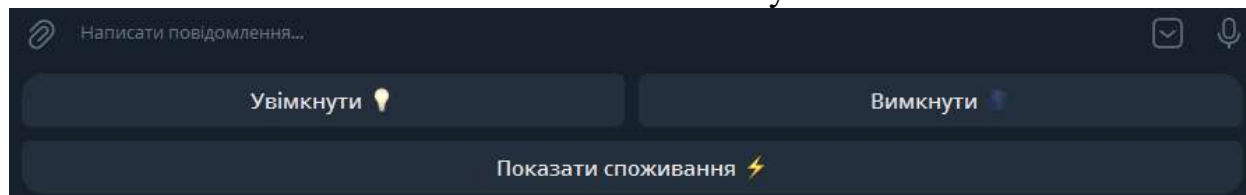


Рис. 3 Кнопки керування у телеграм боті

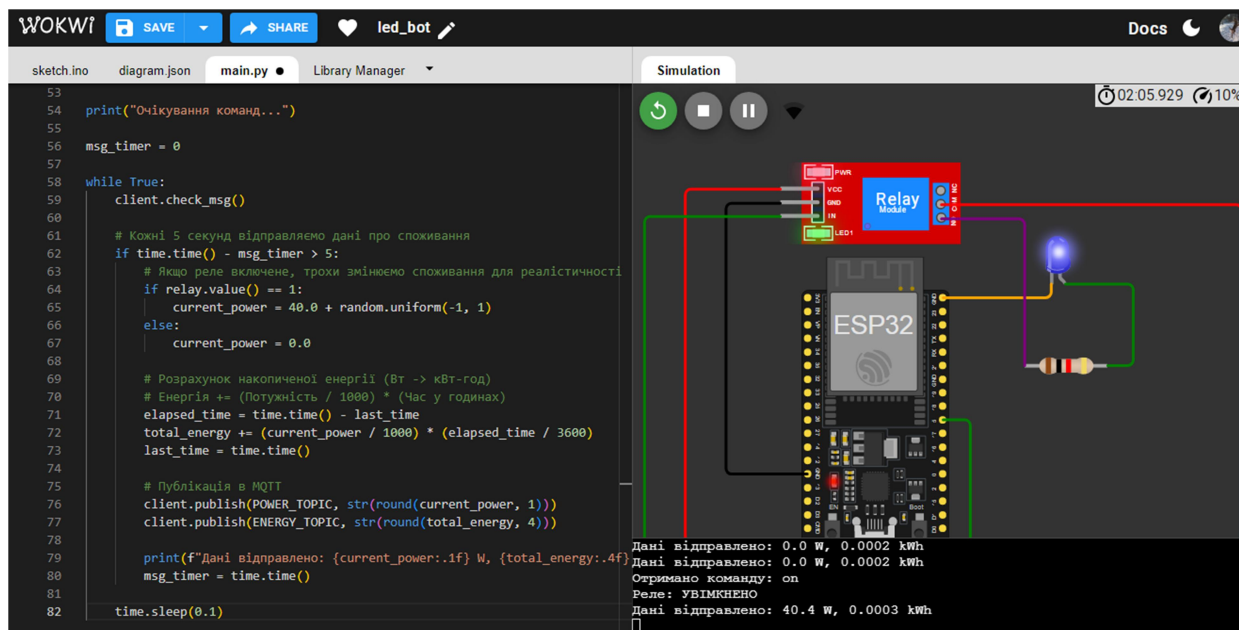


Рис. 4 Увімкнений світлодіод

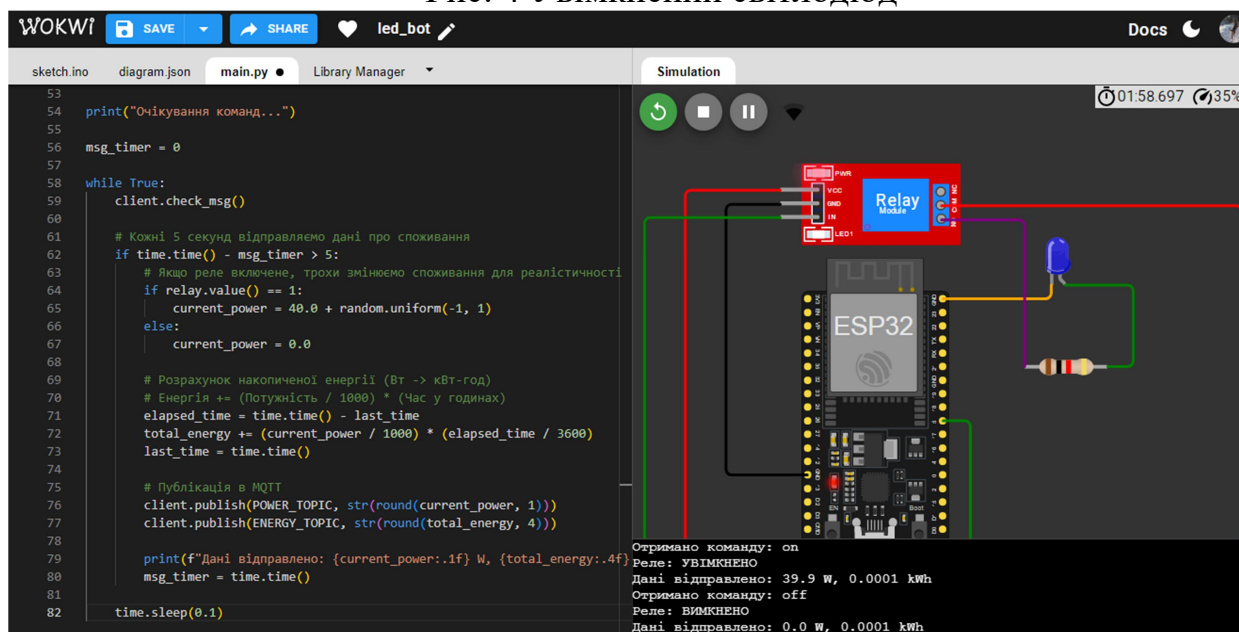


Рис. 5 Вимкнений світлодіод

## ВИСНОВКИ

В роботі досліджено інтерактивний Telegram-бот для керування системою розумного освітлення та моніторингу енергоспоживання на базі пристроїв Shelly з використанням мови програмування Python, асинхронного фреймворку Aiogram та протоколу MQTT.

Практичне значення полягає у створенні мобільної та незалежної від виробника системи керування, яка може бути розгорнута на базі недорогих мікрокомп'ютерів або серверів. Система забезпечує віддалений доступ до моніторингу енергоспоживання з будь-якої точки світу, що є критично важливим для підвищення енергоефективності сучасних житлових та офісних приміщень.

Подальший розвиток включає розширення функціоналу бота (додавання графіків споживання за допомогою matplotlib), інтеграцію алгоритмів машинного навчання для аналізу аномального споживання електроенергії та підтримку голосового керування через API месенджера.

Отримані результати підтверджують ефективність обраного технологічного стеку (Python + Aiogram + MQTT) для створення сучасних, масштабованих та енергоефективних IoT-рішень.

### ДЖЕРЕЛА

1. An iot-based smart plug energy monitoring system / L. Albraheem та ін. International journal of advanced computer science and applications. 2023. Т. 14, № 10. URL: <https://doi.org/10.14569/ijacsa.2023.0141038> (дата звернення: 13.04.2026).
2. Mohamed A., Ismail I., AlDaraawi M. IoT-Driven intelligent energy management: leveraging smart monitoring applications and artificial neural networks (ANN) for sustainable practices. Computers. 2025. Т. 14, № 7. С. 269. URL: <https://doi.org/10.3390/computers14070269> (дата звернення: 13.04.2026).
3. Web-Based internet of things on environmental and lighting control and monitoring system using node-red, MQTT and modbus communications within embedded linux platform / C.-Y. Chen та ін. Internet of things. 2024. С. 101305. URL: <https://doi.org/10.1016/j.iot.2024.101305> (дата звернення: 13.04.2026).
4. Dinmohammadi F., Farook A. M., Shafiee M. Improving energy efficiency in buildings with an iot-based smart monitoring system. Energies. 2025. Т. 18, № 5. С. 1269. URL: <https://doi.org/10.3390/en18051269> (дата звернення: 13.04.2026).
5. K S., M M. S. Illuminating the future: a comprehensive study on IoT-enabled smart lighting systems. Ijcr. 2024. Т. 12, № 4. С. 1–8. URL: <https://ijcr.org/papers/IJCRT2404070.pdf> (дата звернення: 13.04.2026).
6. Гал-Інфо. Система розумного будинку – основні функції та складові елементи | Гал-Інфо. Гал-Інфо. URL: [https://galinfo.com.ua/news/systema\\_rozumnogo\\_budynku\\_osnovni\\_funktsii\\_t\\_a\\_skladovi\\_elementy\\_377240](https://galinfo.com.ua/news/systema_rozumnogo_budynku_osnovni_funktsii_t_a_skladovi_elementy_377240) (дата звернення: 12.04.2026).
7. Розумний будинок: що це, як працює, переваги та недоліки технології. - Dniprobud. URL: <https://dniprobud.com.ua/ua/articles/umnyjj-dom> (дата звернення: 12.04.2026).
8. MQTT - the standard for iot messaging. MQTT - The Standard for IoT Messaging. URL: <https://mqtt.org> (дата звернення: 12.04.2026).
9. API reference. Welcome to Shelly Technical Documentation | Shelly Technical Documentation. URL: <https://shelly-api-docs.shelly.cloud/gen1/> (дата звернення: 16.04.2026).

10. Welcome to python.org. Python.org. URL: <https://www.python.org/> (дата звернення: 20.04.2026).
11. EMQ. MQTT in Python with Paho Client: Beginner's Guide 2025. [www.emqx.com](http://www.emqx.com). URL: <https://www.emqx.com/en/blog/how-to-use-mqtt-in-python> (дата звернення: 20.04.2026).
12. Requests: HTTP for Humans™ – Requests 2.32.5 documentation. Requests: HTTP for Humans™ – Requests 2.32.5 documentation. URL: <https://requests.readthedocs.io/en/latest/> (дата звернення: 20.04.2026).
13. Aiogram. aiogram. URL: <https://aiogram.dev> (дата звернення: 25.04.2026).
14. Wokwi - World's most advanced ESP32 Simulator. Wokwi - World's most advanced ESP32 Simulator. URL: <https://wokwi.com/> (дата звернення: 22.04.2026).
15. Abramov, V., Astafieva, M., Boiko, M., Bodnenko, D., Bushma, A., Vember, V., Hlushak, O., Zhyltsov, O., Ilich, L., Kobets, N., Kovaliuk, T., Kuchakovska, H., Lytvyn, O., Lytvyn, P., Mashkina, I., Morze, N., Nosenko, T., Proshkin, V., Radchenko, S., & Yaskevych, V. (2021). Theoretical and practical aspects of the use of mathematical methods and information technology in education and science. <https://doi.org/10.28925/9720213284km>