

Куликовська Н.А.

Національний університет «Запорізька політехніка»

Тіменко А.В.

Національний університет «Запорізька політехніка»

Костецький Д.В.

Національний університет «Запорізька політехніка»

АВТОМАТИЗОВАНА СИСТЕМА КОНТРОЛЮ ТЕМПЕРАТУРИ В ПРИМІЩЕННІ

Стаття присвячена інформаційним технологіям у сфері Інтернету речей (IoT), які дозволяють з'єднувати різні пристрої та сенсори через мережу Інтернет. Але для комфортної та ефективною взаємодії клієнта з IoT-системами необхідні розумні та природні інтерфейси користувача, які враховують контекст та потреби людини. У статті розкрито використання чатботів, які можуть забезпечити користувачам зручну та продуктивну підтримку, відгукуючись на їх питання, безпосередньо вирішуючи їх проблеми. Крім того, визначено, що чатботи можуть обробляти великі масиви даних, які генеруються сенсорами та системами IoT, та візуалізувати їх у доступному для розуміння форматі. У даній роботі розглянуто IoT-систему, яка складається з датчиків, хмарної платформи для зберігання та обробки даних, бази даних та інтерфейсу чатбота для взаємодії з користувачем. Автоматизована система контролю температури в приміщенні використовує датчики температури, підключені до IoT або мікроконтролерів, які регулярно вимірюють температуру і передають її в хмарну платформу. Хмарна платформа обробляє, аналізує і зберігає дані про температуру у базі даних за допомогою Python скриптів. Користувачі можуть взаємодіяти з IoT системою через інтерфейс чатбота, який розуміє їх команди і реагує відповідно. Чатбот дозволяє запитувати поточну температуру, історичні дані про температуру, встановлювати порогові значення температури або налаштовувати інші аспекти IoT системи. Тестування чатбота проводилось на прикладі де встановлені датчики температури, кондиціонери, датчики відкриття/закриття вікон, датчики відкриття/закриття дверей, система вентиляції. Бот може працювати автономно підтримуючи температуру. Чатбот має адаптивний інтерфейс та реагує на текстові команди від користувача для зміни температури до певного значення. В системі передбачені критичні ситуації. Визначені механізми захисту, такі як автоматична зміна температури та повідомлення про виникнення проблеми. В результаті дослідження перевірені функції чатбота в реальному часі для надання користувачам можливості встановлювати бажану температуру в приміщеннях, переглядати поточний стан датчиків температури та кондиціонерів, отримувати сповіщення про критичні ситуації та умови включення датчиків та кондиціонерів. Завдяки використанню інтерфейсу месенджера отримано людину орієнтовану автоматизовану систему.

Ключові слова: IoT, чатбот, моделювання, контроль температури, критичні ситуації, модель, система, датчики.

Постановка проблеми. Інтернет речей (IoT) є одним з найперспективніших напрямків розвитку інформаційних технологій, який передбачає зв'язок між різноманітними пристроями та сенсорами через мережу Інтернет. IoT системи можуть збирати, обробляти та аналізувати великі обсяги даних, що генеруються цими пристроями, та використовувати їх для покращення якості життя людей, ефективності бізнесу та безпеки середовища. Однак для того, щоб клієнти могли легко та зручно взаємодіяти з IoT системами, потрібні інтелектуальні та природні інтерфейси

користувача. Чатботи є одним з таких інтерфейсів, які використовують штучний інтелект (ШІ) та обробку природної мови (ОПМ) для симуляції розмови з людьми або іншими чатботами за допомогою тексту або голосових повідомлень [1, с. 1]. Чатботи можуть надавати користувачам комфортну та ефективну допомогу, відповідаючи на їх запитання, надаючи їм більш цікаві відповіді, безпосередньо реагуючи на їх проблеми. Крім того, чатботи можуть аналізувати великі обсяги даних, що виробляються сенсорами та системами IoT, та представляти їх у форматі,

який легко сприймається та діє користувачами [2, с. 100006].

Аналіз останніх досліджень і публікацій. Останні дослідження та публікації за цією темою свідчать про високу актуальність та перспективність чатботів як інтерфейсу користувача для IoT систем. Наприклад, автори статті [3, с. 2930] пропонують міждисциплінарну наукову програму для майбутнього розвитку чатботів, в якій вони виділяють шість тематичних напрямків:

- користувачі та наслідки;
- досвід користувача та дизайн;
- фреймворки та платформи;
- чатботи для співпраці;
- демократизація чатботів;
- етика та приватність.

Для кожного з цих напрямків вони надають короткий огляд сучасного стану, обговорюють ключові наукові виклики та пропонують перспективні напрямки для майбутнього дослідження.

У статті [4, с. 136] автори розглядають застосування чатботів для вирішення складних бізнес-проблем за допомогою ШІ та IoT. Вони показують, як чатботи можуть забезпечити швидкий і зручний доступ до великих обсягів даних, що генеруються пристроями IoT, а також аналізувати

ці дані для покращення продуктивності та якості послуг. Вони також описують деякі сценарії використання чатботів у розумних фабриках, розумних будинках та розумних медичних системах.

У статті [5, с. 10] автори описують історію, технологію та застосування чатботів як природних діалогових систем. Вони розглядають розвиток чатботів в контексті розвитку ШІ, ОПМ та взаємодії людини з комп'ютером. Вони також аналізують сучасні типи чатботів за їх функціональністю та архітектурою. Вони показують, як чатботи можуть бути корисними у різних галузях, таких як освіта, охорона здоров'я, бізнес та розваги.

Метою статті є дослідження роботи чатбот для контролю температури в приміщеннях за допомогою IoT-систем. Аналіз сучасних мов програмування IoT-систем. Перевірка функцій чатбота для надання користувачам можливості встановлювати бажану температуру в приміщеннях, переглядати поточний стан датчиків температури та кондиціонерів, отримувати сповіщення про критичні ситуації та умови включення датчиків та кондиціонерів.

Виклад основного матеріалу. При виборі мов програмування для систем IoT, існує кілька доступних варіантів, кожен зі своїми перевагами та недоліками та сферами застосування. В таблиці 1

Таблиця 1

Аналіз популярних мов програмування

Мова програмування	Аналіз застосування мови програмування для IoT систем
C/C++	Широко використовуються в IoT завдяки її ефективності, низькому рівню керування та сумісності з мікроконтролерами та вбудованими системами. Вона забезпечує прямий доступ до обладнання, що робить його придатним для пристроїв з обмеженими ресурсами. C/C++ зазвичай використовуються в платформах IoT, розробці програмного забезпечення та драйверах пристроїв [6, с. 1].
Python	Це універсальна та проста у вивченні мова, відома своєю доступністю та читабельністю. Вона пропонує широкі бібліотеки та фреймворки для розробки IoT, що робить її популярною для швидкого створення прототипів, аналізу даних та хмарної інтеграції. Хоча Python може бути не таким ефективним, як C/C++, він часто використовується для додатків IoT вищого рівня, обробки даних і сценаріїв [7, с. 8175].
Java	Вона зазвичай використовується в шлюзах IoT, серверних програмах і пристроях IoT на базі Android. Надійність, функції безпеки та масштабованість Java роблять її придатною для широкомасштабного розгортання IoT та корпоративних рішень [8, с. 3].
Rust	Це мова системного програмування, відома своєю зосередженістю на безпеці, продуктивності та паралелізмі. Вона забезпечує гарантії безпеки пам'яті та надійну типізацію, що робить її придатною для низькорівневого програмування в пристроях Інтернету речей і мікропрограмному забезпеченні. Rust часто використовується в критично важливих для безпеки програмах або там, де ефективність використання ресурсів має вирішальне значення [9, с. 55].
Lua	Це легка мова сценаріїв, яка часто використовується у вбудованих системах і пристроях Інтернету речей. Вона легко масштабується та забезпечує хорошу продуктивність. Lua зазвичай використовується для програмування сенсорних вузлів, шлюзів Інтернету речей і створення спеціального програмного забезпечення [10, с. 200].
Мова програмування	Аналіз застосування мови програмування для IoT систем
Swift	Swift в основному асоціюється з розробкою iOS і macOS, але вона набуває популярності в IoT завдяки своїм функціям безпеки, виразності та сумісності з інфраструктурою Apple HomeKit. Swift підходить для розробки додатків IoT, які інтегруються в екосистему Apple [11, с. 22].

наведено аналіз популярних мов програмування, які використовуються в розробці IoT:

Зрештою, вибір мови програмування для системи IoT залежить від таких факторів, як цільове обладнання, обмеження ресурсів, вимоги до масштабованості, швидкість розробки та наявний досвід. Системи IoT зазвичай використовують комбінацію мов на різних рівнях, таких як мікро-програмне забезпечення, шлюз, хмарні служби та веб/мобільні інтерфейси.

У даній роботі ми розглядаємо IoT-систему, яка складається з датчиків, хмарної платформи для зберігання та обробки даних, бази даних та інтерфейсу чатбота для взаємодії з користувачем (рис. 1).



Рис. 1. IoT-система

Для пристроїв IoT обрані плати Arduino. Мова розробки Python використовується завдяки широким бібліотекам, підтримці аналізу даних і простоті інтеграції з пристроями IoT. Для зберігання даних про температуру застосована реляційна система керування базами даних PostgreSQL. Ця технологія дає надійну підтримку запитів SQL і може обробляти великі набори даних. Для підключення до бази даних із хмарної платформи застосована бібліотека psycopg2.

Чатбот написаний мовою Python. Для модулю ОПМ використана бібліотека spaCy, яка пропонує розширені функції для токенизації, розпізнавання іменованих об'єктів і аналізу синтаксичних залежностей. Python також надає попередньо підготовлені моделі для різних мов, що дозволяє швидше розробляти додаток. spaCy можна використовувати для аналізу запитів користувачів, вилучення важливих об'єктів (наприклад, значень температури, часу, місця розташування) і розуміння контексту для формулювання значущих відповідей.

Алгоритм роботи автоматизованої системи контролю температури в приміщенні наступний:

1. Датчики підключені до пристроїв IoT або мікроконтролерів, періодично збирають дані про температуру.

2. За допомогою спеціальних бібліотек дані датчиків зчитуються та зберігаються в хмарі.

3. Хмарна платформа отримує дані про температуру від пристроїв IoT через AWS IoT Core.

Скрипти Python обробляють вхідні дані, за потреби виконують аналіз і зберігають їх у базі даних.

4. Інтерфейс чатбота надає користувачам розмовний інтерфейс для взаємодії з системою IoT. Чатбот розуміє команди користувача, отримує дані про температуру з бази даних і реагує відповідно. Клієнти можуть запитувати поточну температуру, історичні дані про температуру, встановлювати порогові значення температури або налаштовувати інші аспекти системи IoT через інтерфейс чатбота.

Блочна структура модулів представленого IoT-бота для температурного контролю в приміщенні надана на рисунку 2.

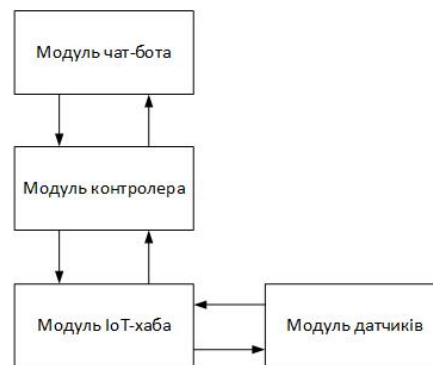


Рис. 2. Блочна структура модулів IoT-бота

Розглянемо детальніше функції кожного блоку:

– модуль чатбота є інтерфейсом між людиною та IoT-системою. Це програмне забезпечення, яке дозволяє користувачеві встановлювати бажану температуру в кожному приміщенні окремо, або встановлювати глобальні параметри для всього будинку;

– модуль контролера аналізує дані від IoT-хаба та приймає рішення про керування пристроями для контролю або зміни температури;

– модуль IoT-хаба збирає всі дані від різних датчиків та передає їх до модуля контролера;

– модуль IoT-датчиків включає в себе сенсори вимірювання температури, роботи кондиціонера, відкритих вікон, вентиляційних решіток. Ці сенсори підключені до мережі Інтернет, щоб надсилати дані в модуль IoT-хаба.

Тестування чатбота проводилось на прикладі приміщення ангара з площею 800 м² де встановлені датчики температури, кондиціонери, датчики відкриття/закриття вікон, датчики відкриття/закриття дверей, система вентиляції. Бот може працювати автономно підтримуючи температуру за допомогою кондиціонерів не вище 32 градусів (рис. 3а). Чатбот має користувацький інтерфейс та реагує на текстові команди від користувача для зміни температури до певного значення (рис. 3б). В кожному режимі є контроль закритих вікон та

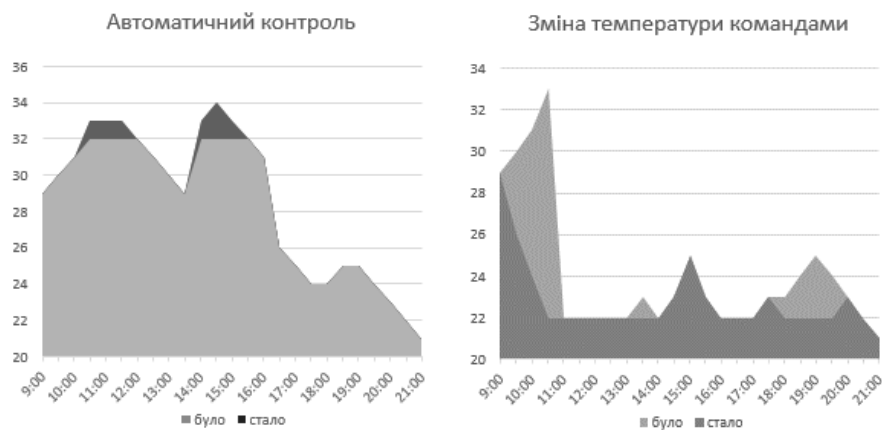


Рис. 3. Результати тестування IoT-бота:
а) робота чатбота в автономному режимі; б) керування температурою згідно команд користувача

дверей, включення системи вентиляції. Показники температури вимірюються кожні півгодини.

В системі передбачені критичні ситуації. Наприклад, збій IoT-датчика температури або помилкова команда користувача, яка може призвести до небезпечної зміни температури в приміщенні. Тому реалізовано механізми захисту, такі як автоматична зміна температури та повідомлення про виникнення проблеми.

Висновки. У даній роботі ми розглянули IoT-систему, яка складалася з датчиків, хмарної платформи для зберігання та обробки даних, бази даних та інтерфейсу чатбота для взаємодії з користувачем. Автоматизована система контролю температури в приміщенні використовувала датчики температури, підключені до IoT або мікроконтролерів, які регулярно вимірювали температуру і передавали її в хмарну платформу через AWS IoT Core. Хмарна платформа обробляла, аналізувала і зберігала дані

про температуру у базі даних за допомогою Python скриптів. Користувачі могли взаємодіяти з IoT системою через інтерфейс чатбота, який розумів їх команди і реагував відповідно. Чатбот дозволяв користувачам запитувати поточну температуру, історичні дані про температуру, встановлювати порогові значення температури або налаштовувати інші аспекти IoT системи. Використання IoT чатбота для контролю температури в приміщенні дозволяє забезпечити комфортні умови для людей та обладнання, зменшити витрати енергії та покращити безпеку. Така автоматизована система є перспективною для розвитку індустрії, оскільки вона може застосовуватися в різних сферах, таких як сільське господарство, медицина, складське зберігання. Наступні етапи дослідження в цій темі можуть бути пов'язані з розширенням функціоналу IoT чатбота, покращенням його інтелектуальних здібностей та адаптацією до різних мов.

Список літератури:

1. How AI and Chatbots with IoT can help in solving tricky business. URL: <https://www.capgemini.com/insights/expert-perspectives/how-ai-and-chatbots-with-iot-can-help-in-solving-tricky-business-problems/> (date of access: 01.05.2023).
2. Adamopoulou E., Moussiades L. Chatbots: History, technology, and applications. *Machine Learning with Applications*. 2020. Vol. 2. P. 100006. URL: <https://doi.org/10.1016/j.mlwa.2020.100006> (date of access: 10.05.2023).
3. Future directions for chatbot research: an interdisciplinary research agenda / A. Følstad et al. *Computing*. 2021. Vol. 103, no. 12. P. 2915–2942. URL: <https://doi.org/10.1007/s00607-021-01016-7> (date of access: 11.04.2023).
4. McTear M. Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots. *Synthesis Lectures on Human Language Technologies*. 2020. Vol. 13, no. 3. P. 1–251. URL: <https://doi.org/10.2200/s01060ed1v01y202010hlt048> (date of access: 10.03.2023).
5. Castellucci P., Gomelino E. Chatbot. *Digitalita*. 2021. Vol. 16, no. 2. P. 9–24. URL: <https://doi.org/10.36181/digitalita-00034> (date of access: 01.02.2023).
6. Why Chatbots Can Be Used As Internet of Things (IoT) Interface. URL: <https://www.ameyo.com/blog/why-chatbots-can-be-used-as-internet-of-things-iot-interface> (date of access: 31.05.2023).

7. Controlling Devices by IoT. International Journal of Recent Technology and Engineering. 2019. Vol. 8, no. 4. P. 8173–8176. URL: <https://doi.org/10.35940/ijrte.d8777.118419> (date of access: 01.02.2023).
8. Kulykovska N., Kudermetov R. Improving the Productivity of Distributed Computer Systems Through the Use of Knowledge Engineering. 2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Dortmund, 17–18 September 2020. pp. 1–4. URL: <https://doi.org/10.1109/idaacs-sws50031.2020.9297082> (date of access: 04.03.2023).
9. Тіменко А.В., Шкарупіло В.В., Смолій В.В. Нейромережева модель контролю апаратної сумісності компонентів IoT-системи Вісник Запорізького національного університету фізичних і математичних наук. 2021. № 2. С. 52–59. URL: <https://doi.org/10.26661/2413-6549-2020-2-07> (дата звернення: 21.03.2023).
10. Python Programming Recipes for IoT Applications / M. G. Lanjewar et al. Springer, 2023.
11. Abdul Kadhar K. M., Anand G. Basics of Python Programming. Data Science with Raspberry Pi. Berkeley, CA, 2021. P. 13–47. URL: https://doi.org/10.1007/978-1-4842-6825-4_2 (date of access: 13.05.2023).

Kulykovska N.A., Timenko A.V., Kostetsky D.V. AUTOMATED ROOM TEMPERATURE CONTROL SYSTEM

The article is devoted to information technologies in the field of the Internet of Things (IoT), which allow connecting various devices and sensors via the Internet. But for a comfortable and effective interaction of the client with IoT systems, smart and natural user interfaces that take into account the context and human needs are necessary. The article reveals the use of chatbots that can provide users with convenient and productive support by responding to their questions and solving their problems directly. In addition, it has been determined that chatbots can process large amounts of data generated by sensors and IoT systems and visualize them in an easy-to-understand format. This work examines an IoT system consisting of sensors, a cloud platform for data storage and processing, a database, and a chatbot interface for user interaction. An automated indoor temperature control system uses temperature sensors connected to IoT or microcontrollers that regularly measure the temperature and transmit it to the cloud platform. The cloud platform processes, analyzes and stores temperature data in a database using Python scripts. Users can interact with the IoT system through a chatbot interface that understands their commands and responds accordingly. The chatbot allows you to query the current temperature, historical temperature data, set temperature thresholds, or configure other aspects of the IoT system. The chatbot was tested on an example where temperature sensors, air conditioners, window opening/closing sensors, door opening/closing sensors, and a ventilation system are installed. The bot can work autonomously maintaining the temperature. The chatbot has an adaptive interface and responds to text commands from the user to change the temperature to a certain value. The system provides for critical situations. Protection mechanisms are defined, such as automatic temperature change and notification when a problem occurs. As a result of the study, the chatbot functions were tested in real time to enable users to set the desired temperature in the rooms, view the current state of temperature sensors and air conditioners, receive notifications about critical situations and the conditions for turning on sensors and air conditioners. Thanks to the use of the messenger interface, a human-oriented automated system is obtained.

Key words: *IoT, chatbot, simulation, temperature control, critical situations, model, system, sensors.*