
INFORMATION TECHNOLOGY

DOI 10.20535/2411-1031.2023.11.2.293493

УДК 004

АНДРІЙ ГОЛЯТКІН,
АЛІНА МОШИНСЬКА

РОЗШИРЕННЯ ФУНКЦІОНАЛУ ПРИСТРОЇВ ІОТ В УМОВАХ НАДЗВИЧАЙНИХ СИТУАЦІЙ

Стаття присвячена розширенню функціоналу пристроїв Інтернету речей (ІоТ) з метою підвищення безпеки населення в умовах надзвичайних ситуацій, зокрема в контексті природних катаклізмів та воєнних загроз. Провівши аналіз останніх досліджень та публікацій, зроблено висновок про актуальність розвитку технологій ІоТ в контексті сучасних викликів для безпеки громадян.

Спеціальний акцент робиться на використанні інноваційних підходів для вирішення проблем безпеки населення, зокрема для осіб з обмеженими можливостями, таких як особи з вадами слуху чи літні люди. В статті описано реалізацію програмного забезпечення на мові програмування Python для візуального відображення статусу ракетної загрози чи артилерійських обстрілів за допомогою розумних ламп Yeelight, які індикацією кольорового світла передають інформацію про наявність чи відсутність повідомлень про небезпеку. Запропоновані рішення включають синхронну роботу пристроїв в реальному часі, підключення до надійних джерел повідомлень та забезпечення стабільного зв'язку між пристроями та сервером.

В проєкті продемонстровано інноваційний підхід до вирішення проблем безпеки в умовах кризових ситуацій, зокрема воєнного стану, сприяючи швидкому і точному реагуванню на потенційні загрози та захисту населення.

Ключові слова: Інтернет речей, програмування пристроїв ІоТ, Python, розумна LED-лампа, інформування населення під час війни.

Постановка проблеми. В сучасних умовах, коли людство стикається зі зростаючим ризиком природних катастроф та геополітичних конфліктів, забезпечення ефективного захисту населення стає пріоритетом. Внаслідок природних катаклізмів та геополітичних напружень, таких як вторгнення російської федерації в Україну у 2022 році, стає очевидною необхідність вдосконалення систем безпеки та ефективного реагування на надзвичайні ситуації.

Додатковою проблемою є вразливість певних груп населення, зокрема осіб з вадами слуху та літніх людей, перед ризиками та необхідністю швидкої реакції на загрози. Традиційні методи інформування, такі як сигнали сирени та звукові повідомлення, можуть бути неефективними для цих категорій населення. Озвучена проблема є ключовою передумовою для розробки та впровадження ефективних рішень на основі технологій Інтернету речей, спрямованих на забезпечення безпеки та ефективного інформування населення в умовах надзвичайних обставин.

Аналіз останніх досліджень і публікацій. Останнім часом Інтернет Речей (ІоТ) здобуває зростаючу актуальність у контексті реагування на надзвичайні ситуації та планування дій в умовах кризи. Наступні аспекти демонструють головні переваги використання ІоТ при реагуванні на надзвичайні ситуації.

Однією із найсуттєвіших переваг використання Інтернету Речей у контексті реагування на катастрофи є можливість отримання даних в режимі реального часу. За допомогою

підключення пристроїв IoT до різних систем та датчиків, групи, що координують реагування на надзвичайні ситуації, мають можливість збирати дані у реальному часі. Це забезпечує більш детальне розуміння наслідків катастрофи й покращує ефективність та швидкість реакції на подію. Наприклад, системи на основі мікроконтролерів ESP8266 або ESP32, камери з тепловізійними можливостями можуть використовуватися для моніторингу температури та вологості в будівлях критичної інфраструктури і вчасно виявляти пожежі. М. Трінат Басу, Р. Картік, Дж. Махіта і В. Локеш Редді пропонують використання озвученого мікроконтролера ESP8266 на базі Arduino IDE для вчасного виявлення лісових пожеж та відповідного реагування на них [1].

Ще однією важливою перевагою використання Інтернету Речей у сфері реагування на катастрофи є можливість автоматизувати різноманітні завдання. За допомогою підключення пристроїв IoT до різних систем, можна здійснити автоматичне виконання дій, таких як надсилання невідкладних сповіщень, автоматичне включення різноманітних систем та збір даних. Це сприяє не лише спрощенню завдань для груп реагування на надзвичайні ситуації, але й дозволяє звільнити ресурси, які можуть бути спрямовані на більш важливі задачі [2], [3].

Інтернет Речей також може відігравати роль у покращенні зв'язку між групами, що координують реагування на катастрофи. За допомогою підключення пристроїв IoT до систем зв'язку, команди можуть легко обмінюватися даними одна з одною, спрощуючи координацію відповідей та забезпечуючи єдність дій. Ця можливість особливо корисна під час великомасштабних катастроф, коли команди можуть знаходитися на різних місцях. Окрім цього, Інтернет Речей (IoT) також може сприяти удосконаленню процесу планування надзвичайних ситуацій шляхом збору даних з різних джерел. Команди можуть розробляти більш точні моделі можливого розвитку катастроф, що дозволяє передбачати оптимальні реакції на різні сценарії. Це сприятиме глибшій підготовці до можливих критичних ситуацій, забезпечуючи ефективне використання часу та ресурсів [2], [3].

Після стихійного лиха інфраструктура зв'язку може бути зруйнована, що ускладнює оперативну оцінку потреб постраждалих громад. Використовуючи можливості IoT, служби реагування можуть застосовувати інтелектуальні пристрої для отримання даних у реальному часі та прийняття обґрунтованих рішень. Наприклад, підключені датчики можна використовувати для моніторингу рівня води, якості повітря та інших факторів навколишнього середовища. Після цього оброблені дані дозволяють здійснювати прогнозування поширення загрози та визначення найкращих маршрутів евакуації. Поза збором даних, технологія IoT також має потенціал для координації зусиль у наданні допомоги. Смартфони, планшети та інші пристрої можна використовувати для мобілізації волонтерів, координації ресурсів та надання інших видів допомоги. Технологію Інтернету Речей також можна використовувати для моніторингу процесу відновлення. Підключені пристрої можуть відслідковувати розподіл гуманітарних ресурсів та допомагати рятувальникам визначити області, де потрібна додаткова допомога. У майбутньому навіть можливе використання безпілотників з підтримкою Інтернету Речей для доставки медичних засобів та іншої допомоги безпосередньо постраждалим у віддалених районах [2], [3].

Нижче наведена таблиця з переліком деяких пристроїв та технологій IoT, які, так чи інакше, використовуються для завчасного реагування на стихійні лиха [3].

BRINCO

Це перший детектор, сумісний з Інтернетом речей, який сповіщає користувача про потенційний землетрус. Система сенсорів, що використовується в даному пристрої, включає в себе акселерометр, блок звукового сигналу для надання акустичних попереджень та цифровий сигнальний блок для обробки даних.

Маяк передає зібрані дані до Центру даних Brinco (BDC), який функціонує як приватний постачальник хмарних послуг. При виявленні будь-якого руху з боку Землі, цей центр даних аналізує інформацію, що надійшла та інтегрує її з іншими даними, отриманими з сейсмічних датчиків, для формування повноцінного уявлення про потенційну небезпеку. У випадку, якщо

результати оцінки свідчать про високі ризики землетрусу, маяк активує тривожний сигнал та надсилає негайні сповіщення на мобільні пристрої користувачів (які працюють на операційних системах Android або iOS). Крім того, зібрані дані можуть обмінюватися між локальними та глобальними спільнотами за допомогою використання Інтернету [3].

Таблиця 1 – Прилади IoT

Solutions	Cloud enabled	App based	Key sensors	Communications	Application
Brinco	Yes	Yes	Accelerometer	BLE and Wi-Fi	Earthquake and tsunami
Flood Beacon	Yes	No	Ultrasonic range finder	GSM	Flood
Floating sensor network	Yes	Yes	Accelerometer and ultrasonic range finder	GSM and BLE	Flood and tsunami
Lightning detection	Yes	No	Lightning sensor	Radio	Lightning
Alarms	Yes	Yes	Accelerometer	Radio and BLE	Landslide
My shake	Yes	Yes	Accelerometer	CDMA, Wi-Fi, and BLE	Earthquake

Flood Beacon

Дані детектори розроблені для передачі даних про поточний об'єм та напрямок руху води. Зокрема, ця система допомагає відстежувати повені та цунамі практично в реальному часі. Плавучі пристрої відстежують рівень води над поверхнею водойми. Коли показники підняття води перевищують певний поріг, автоматично відправляється сповіщення особі, яка може постраждати. Всі зібрані дані про водні об'єкти обробляються в централізованому хмарному сервісі IoT Xively для подальшого аналізу та обробки [3].

Floating Sensor Network

В Університеті Каліфорнії в Берклі вдалося досягнути значного прогресу у створенні нової моделі датчиків для збору інформації про стан річкового стоку під час повені. Цей компактний комерційний плаваючий девайс оснащений камерою та системами глобального позиціонування (GPS). Він автоматично реєструє різкі зміни або поступові зсуви води та передає цю інформацію місцевим мешканцям через веб-сповіщення [3].

Lighting Detection

Сильні грози можуть серйозно нашкодити життю людей. Щороку близько 24 000 осіб по всьому світу гинуть від ударів блискавки. З метою запобігання цим негодам розробляються детектори блискавки на базі Інтернету речей. Детектор блискавки, наприклад, AS3935 Franklin Lightning Sensor, на платформі Raspberry Pi, обладнаний сенсором блискавки та може реєструвати навіть незначну грозову активність на відстані кількох кілометрів [4].

ALARMS

Британське геологічне дослідження оприлюднило проект під назвою "Оцінка ризиків обвалів за допомогою систем реального моніторингу з використанням акустичних сигналів" (Acoustic Real-time Monitoring Systems, ALARMS), який може надавати відомості про раннє попередження можливих обвалів на територіях в зоні ризику. Для визначення нестабільності схилу пристрій на основі акселерометра встановлюється над областю схилу, де можливий обвал [5].

Myshake

Дана платформа є додатком для моніторингу сейсмічної активності і використовується на мобільних пристроях користувачів. При виявленні сенсором мобільного пристрою поштовхів, характерних для землетрусів, додаток виконує відповідний аналіз. Якщо

результати аналізу вказують на землетрус, інформація надсилається до Берклійської сейсмологічної лабораторії (BSL) для подальшої обробки з використанням останніх координат GPS, отриманих від мобільного пристрою. Цей підхід створив можливість розробити ефективну систему моніторингу сейсмічної активності, яка базується на спільноті користувачів і спростила збір даних про сейсмічну активність у реальному часі [3].

Рисунки наведені нижче демонструють графічне зображення технологій зв'язку Інтернету речей, які використовуються для реагування на природні катастрофи. На графіках представлено вартість та споживану потужність різних технологій [3].

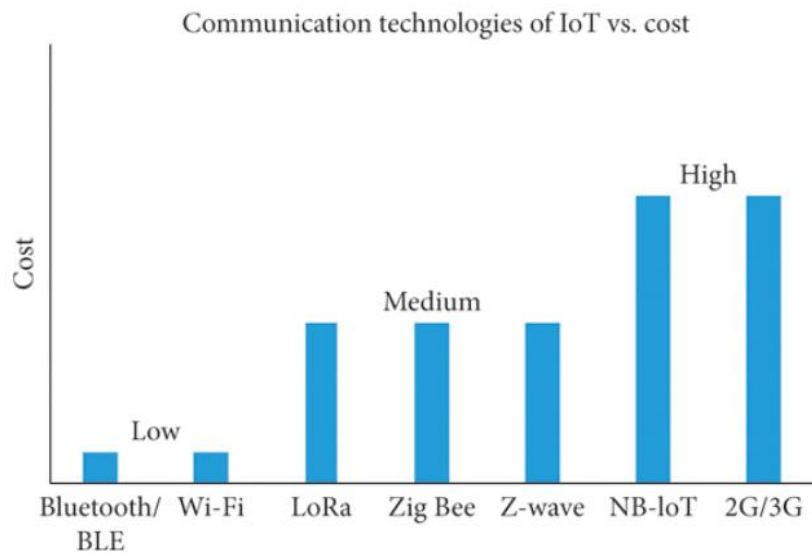


Рисунок 1 – Показники вартості використання різних технологій зв'язку

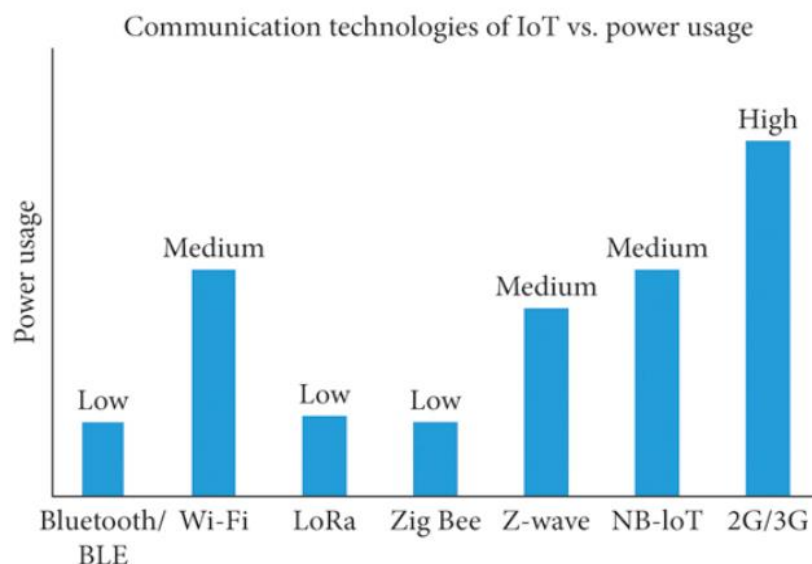


Рисунок 2 – Показники споживаної потужності різних технологій зв'язку

Незважаючи на великий потенціал технології IoT для покращення реагування на стихійні лиха, техногенні катастрофи та воєнні дії, все ж існують численні виклики, які потрібно подолати. Наприклад, багато постраждалих територій може не мати належної інфраструктури для ефективного функціонування пристроїв Інтернету Речей. Крім того, питання безпеки підключених пристроїв є однією з головних труднощів.

Проте, в цілому, підхід до реагування на кризові ситуації на основі Інтернету речей пропонує низку вагомих переваг порівняно з традиційними методами. Використовуючи дані та аналітику в режимі реального часу, рішення, засновані на Інтернеті речей, дозволяють швидко та ефективно виявляти, відстежувати та реагувати на катастрофи. Додатково, системи, побудовані на базі Інтернету речей, можуть бути використані для інформування про плани евакуації або сигнали про небезпеку. Використання пристроїв Інтернету речей дозволяє швидше реагувати та ефективніше діяти під час небезпечної ситуації, що сприяє зменшенню кількості постраждалих людей, економічних збитків і втрат.

Формулювання цілей статті. Нажаль, окрім природних катаклізмів існують й інші загрози безпеки людей. Повномасштабне вторгнення російської федерації на територію України 24 лютого 2022 року наочно продемонструвало чому саме в Україні необхідно розвивати та просувати стартапи, пов'язані з технологіями Інтернету речей. В умовах сучасної війни застосування на практиці проектів Інтернету речей може запропонувати інноваційний підхід у вирішенні проблем як військового, так і цивільного населення, безпека якого є дуже важливою в умовах воєнного стану в країні.

Проведення вчасного інформування усіх груп населення України про можливі загрози ракетних або артилерійських обстрілів є обов'язковим заходом, який може врятувати життя людей. Особливої уваги заслуговують особи з певними вадами слуху або літні люди, які чисто фізично не здатні швидко реагувати на сигнали сирени, звукові повідомлення, погано користуються смартфоном та персональним комп'ютером. Вирішення такої проблеми є однією з головних цілей практичного проекту, розробка якого буде описана в даній статті. В результаті роботи створеного програмного забезпечення пристрій, або група пристроїв, мають візуально відображати статус наявності або відсутності повідомлень про небезпеку. Індикатором відображення може слугувати колір розумної лампи: червоний – наявність повідомлення про повітряну тривогу на території області користувача, зелений – відсутність такого повідомлення. Окрім цього, даний пристрій, або група пристроїв, мають синхронно працювати в режимі реального часу, а також бути під'єднані до перевіреного джерела повідомлень про небезпеку. Прикладом надійних джерел можуть бути офіційні канали в соціальних мережах, сповіщення від компанії Google, тощо. Окрім цього необхідно забезпечити стабільний зв'язок LED-ламп одна з одною та з сервером. Також необхідно надати можливість керування пристроями в ручному режимі.

Виклад основного матеріалу дослідження. Орієнтуючись на поставлені завдання щодо розширення функціоналу пристрою IoT розумна LED-лампа, було прийнято рішення обрати Python в якості мови програмування для проекту. Провівши аналіз недоліків та переваг різних мов програмування було виділено ряд факторів, які стали ключовими у виборі Python.

По-перше, простота синтаксису цієї мови програмування дозволяє швидко розробляти якісні прототипи програмного забезпечення на різні платформи.

По-друге, Python має велику кількість різних фреймворків та бібліотек. Така особливість дозволяє досить легко вирішувати проблеми роботи з мережею, керувати пристроям від різних виробників, а також забезпечувати зв'язок з LED-лампам та сервером.

По-третє, Python може легко інтегруватися з різними джерелами повідомлень про небезпеку, такими як соціальні мережі та сповіщення від компаній, завдяки багатьом бібліотекам для роботи з HTTP-запитами. Окрім цього, ця мова програмування дозволяє створювати інтуїтивний і легкий у використанні інтерфейс для керування LED-лампам в ручному режимі.

По-четверте, ця мова програмування підтримує багато різних систем управління базами даних (СУБД), завдяки бібліотекам та модулям. Така особливість допомагає при вирішенні проблеми створення мережі з пристроїв Інтернету речей, в даному випадку розумних LED-ламп [6].

В якості пристрою Інтернету речей, що буде програмуватися було прийнято рішення обрати розумну LED-лампу від виробника Yeelight (дочірня компанія Xiaomi). IoT лампи Yeelight

задовольняють усім вимогам необхідним для реалізації проекту, а саме: можливість програмування пристрою мовою Python, невисока ціна, вільний продаж на території України [7].

Авторизація користувача та підключення розумних ламп було реалізовано через розробку бота в соціальній мережі Telegram. Системою автоматичного реагування пристрою на наявність або відсутність повітряної тривоги на території користувача виконується за допомогою підключення telegram каналів з повідомленнями про небезпеку. Для коректної роботи системи автоматизації зазначений бот має бути доданий з правами адміністратора в необхідні telegram канали.

Після вибору мови програмування, фізичного обладнання, способів авторизації користувачів та автоматизації процесів, першим кроком в розробці програмного забезпечення буде встановлення необхідних Python бібліотек. В даному проекті використовуються дві бібліотеки, які потребують встановлення, а саме pyTelegramBotAPI(telebot) та Yeelight.

Бібліотека telebot є інструментом для взаємодії з Telegram Bot API за допомогою мови програмування Python. Вона надає зручний інтерфейс для розробки ботів для месенджера Telegram [8].

Бібліотека yeelight є інструментом для взаємодії з розумними лампами Yeelight різних моделей за допомогою мови програмування Python. Вона дозволяє легко керувати функціями такими як вмикання/вимикання, зміна кольору, яскравості, температури кольору тощо [7].

Встановлення зазначених бібліотек виконується за допомогою консольних команд `pip install pyTelegramBotAPI` та `pip install yeelight` відповідно. Після успішного встановлення бібліотеки можна імпортувати до середини розробки, як це продемонстровано нижче. В даному випадку в якості середовища розробки використовувалося Visual Studio Code. Також імпортується модуль `re` - це вбудований модуль Python. Він дозволяє працювати з регулярними виразами та не потребує додаткового встановлення [6].

```
import telebot
import re
from telebot import types
from yeelight import Bulb, BulbException
```

Наступним кроком розробки програми буде реєстрація нового бота в BotFather та отримання спеціального токена. BotFather - це офіційний бот Telegram, який може використовуватися для створення нових ботів та отримання токенів для керування ними.

```
TOKEN = ' '
bot = telebot.TeleBot(TOKEN)
```

В якості простої бази даних, в якій буде зберігатися інформація про кількість підключених ламп та їх IP адреси використовується словник `user_data`. Використання словника - це відмінний спосіб зберігання простих даних невеликого розміру. На даному етапі бот не має великого об'єму даних та додаткових вимог до їх збереження, тому використання словника може бути прийнятним рішенням. Проте, в подальшому, для розширення функціоналу та вдосконалення ПЗ, можливе використання різних сучасних баз даних, таких як SQL Lite, MongoDB та інших.

```
user_data = {}
```

Нижче наведений варіант коду з поясненням, для підключення необхідної кількості розумних ламп користувача.

При отриманні команди `/start` бот вітає користувача та починає процес підключення ламп.

Функції для підключення ламп:

- `ask_lamps_count`: Запитує користувача про кількість ламп для підключення.
- `process_lamps_count`: Обробляє введену користувачем кількість ламп.

– ask_lamp_ips: Запитує IP-адреси ламп для підключення. IP-адреси пристроїв користувач може переглянути в офіційному застосунку Yeelight на своєму смартфоні в розділі «Інформація про пристрій».

– process_lamp_ip: Обробляє введений користувачем IP-адресу лампи та перевіряє її коректність.

– connect_to_lamps: Встановлює з'єднання з усіма лампами, вмикає їх та повідомляє користувача про успішне підключення пристроїв.

```
@bot.message_handler(commands=['start'])
def handle_start(message):
    bot.send_message(message.chat.id, "Вітаю! Я бот для керування та реагування на повітряну тривогу лампи Yeelight.")
    ask_lamps_count(message.chat.id)
```

```
def ask_lamps_count(chat_id):
    msg = bot.send_message(chat_id, "Скільки ламп ви хочете підключити? Відповідь напишіть цифрою. Якщо щось піде не так, напишіть команду /start, щоб заново почати процедуру підключення ламп.")
    bot.register_next_step_handler(msg, process_lamps_count)
```

```
def process_lamps_count(message):
    try:
        lamps_count = int(message.text)
        user_data[message.chat.id] = {'lamps_count': lamps_count, 'bulbs': []}
        ask_lamp_ips(message.chat.id, lamps_count, 0)
    except ValueError:
        msg = bot.send_message(message.chat.id, "Будь ласка, введіть число.")
        bot.register_next_step_handler(msg, process_lamps_count)
```

```
def ask_lamp_ips(chat_id, lamps_count, current_lamp_index):
    user_data[chat_id]['lamps_count'] = lamps_count

    if current_lamp_index < lamps_count:
        msg = bot.send_message(chat_id, f"Введіть IP адресу для лампи {current_lamp_index + 1}:")
        bot.register_next_step_handler(msg, lambda m: process_lamp_ip(m, chat_id, lamps_count, current_lamp_index))
    else:
        connect_to_lamps(chat_id)
```

```
def process_lamp_ip(message, chat_id, lamps_count, current_lamp_index):
    if is_valid_ip(message.text):
        try:
            bulb = Bulb(message.text)
            bulb.get_properties() # Намагаюся отримати властивості лампи, щоб зрозуміти, що вона доступна
            user_data[chat_id]['bulbs'].append(bulb)
            bot.send_message(chat_id, f"З'єднання з лампою {current_lamp_index + 1} встановлено.")
            ask_lamp_ips(chat_id, lamps_count, current_lamp_index + 1)
        except BulbException as e:
            msg = bot.send_message(chat_id, f"Помилка підключення до лампи {current_lamp_index + 1}: {e} Перевірте коректність IP лампи и введіть повторно")
            bot.register_next_step_handler(msg, lambda m: process_lamp_ip(m, chat_id, lamps_count, current_lamp_index))
    else:
        msg = bot.send_message(chat_id, "Некоректний IP адреса. Будь ласка, введіть повторно.")
```

```

        bot.register_next_step_handler(msg, lambda m: process_lamp_ip(m,
chat_id, lamps_count, current_lamp_index))

def is_valid_ip(ip):
    # Перевірка, що строка схожа на формат IP адреси (приклад: "192.168.1.1")
    # В кінцевій версії застосунку, можлива більш складна перевірка
    return True

def connect_to_lamps(chat_id):
    try:
        for bulb in user_data[chat_id]['bulbs']:
            bulb.turn_on()
            bot.send_message(chat_id, "Всі лампи успішно підключені та вимкнені!
Також лампи підключені до телеграм каналу з повідомленнями про повітряну тривогу.
Для ручного керування пропишіть команду /setlamp")
    except BulbException as e:
        bot.send_message(chat_id, f"Помилка при вимкненні ламп: {e}")

```

Наступним кроком буде автоматизація роботи IoT ламп Yeelight, а саме їх автоматичне реагування на сигнали повітряної тривоги та відбою повітряної тривоги. Для цього розроблений телеграм бот необхідно додати з правами адміністратора в спеціалізовані телеграм канали, які займаються інформуванням населення про небезпеку. В результаті роботи програмного забезпечення, бот зможе аналізувати повідомлення в каналі, вмикаючи лампи червоним або зеленим кольором в залежності від знайдених в повідомленні ключових слів "тривога" або "відбій". Дана система може бути вдосконалена, аналізом ключових слів в повідомленні, які б ідентифікували наявність або відсутність повітряної тривоги в конкретній області користувача. В такому випадку необхідно запросити користувача вказати свою область при авторизації.

```

@bot.channel_post_handler(content_types=['text', 'sticker', 'photo'])
def handle_channel_post(message):
    for chat_id, data in user_data.items():
        for bulb in data['bulbs']:
            text = message.text.lower()
            if re.search(r'тривога', text):
                bulb.turn_on()
                bulb.set_rgb(255, 0, 0) # Червоний колір при тривозі
            elif re.search(r'відбій', text):
                bulb.turn_on()
                bulb.set_rgb(0, 255, 0) # Зелений колір при відбої

```

Нижче наведений код, що дозволяє створити базовий інтерфейс користувача, для керування розумною лампою в ручному режимі. Наразі реалізовано 4 кнопки: повітряна тривога – лампа вмикається та горить червоним кольором; відбій повітряної тривоги - лампа вмикається та горить зеленим кольором; ввимкнути лампу – лампа вмикається та горить білим кольором та кнопка вимкнення пристрою.

```

@bot.message_handler(commands=["setlamp"])
def start(m):

    keys =types.InlineKeyboardMarkup()
    but_1=types.InlineKeyboardButton(text="ПОВІТРЯНА ТРИВОГА",
callback_data="tryvoga")
    but_2=types.InlineKeyboardButton(text="ВІДБІЙ ПОВІТРЯНОЇ ТРИВОГИ",
callback_data="vidbiy")
    but_3=types.InlineKeyboardButton(text="ВВИМКНУТИ ЛАМПУ",
callback_data="on_lamp")

```



```
but_4=types.InlineKeyboardButton(text="ВИМКНУТИ ЛАМПУ",
callback_data="off_lamp")
keys.add(but_1, but_2, but_3, but_4)
bot.send_message(m.chat.id, "Вас вітає бот, який дозволяє вам керувати
розумною лампою від компанії Yeelight", reply_markup=keys)
```

Бот обробляє зворотні виклики від кнопок та надсилає відповідні команди лампам.

```
@bot.callback_query_handler(func=lambda call: True)
def inlin(call):
    chat_id = call.message.chat.id # Исправлено: добавлено получение chat_id
    if call.data == 'tryvoga':
        bot.send_message(chat_id, "ПОВІТРЯНА ТРИВОГА! ЛАМПА ГОРИТЬ ЧЕРВОНИМ")
        for bulb in user_data[chat_id]['bulbs']:
            bulb.turn_on()
            bulb.set_rgb(255, 0, 0)
    elif call.data == 'vidbiy':
        bot.send_message(chat_id, "ВІДБІЙ ПОВІТРЯНОЇ ТРИВОГИ! ЛАМПА ГОРИТЬ
ЗЕЛЕНИМ")
        for bulb in user_data[chat_id]['bulbs']:
            bulb.turn_on()
            bulb.set_rgb(0, 255, 0)
    elif call.data == 'on_lamp':
        bot.send_message(chat_id, "ЛАМПУ ВВИМКНЕНО")
        for bulb in user_data[chat_id]['bulbs']:
            bulb.turn_on()
            bulb.set_rgb(255, 255, 255)
    elif call.data == 'off_lamp':
        bot.send_message(chat_id, "ЛАМПУ ВИМКНЕНО")
        for bulb in user_data[chat_id]['bulbs']:
            bulb.turn_off()
```

Наступна строка означає, що бот постійно опитує сервер Telegram про отримання нових повідомлень.

```
bot.polling(none_stop=True, interval=0)
```

Також, розроблене програмне забезпечення може бути зкомпільоване у файл формату .exe за допомогою Auto PY to EXE. Встановити дану утиліту можна за допомогою консольної команди pip install auto-py-to-exe. По закінченню встановлення запуск Auto PY to EXE проводиться також консольною командою auto-py-to-exe. Після чого, у вікні, яке відкрилося, можна легко перетворити файл формату .py у формат .exe [9].

Висновки. Розроблений проект є досить доступним у застосуванні та наочно показує, яким чином програмування пристроїв IoT може бути ефективно використано в умовах надзвичайних ситуацій та сучасної війни. Розробка програми з розширення базового функціоналу розумної лампи може слугувати гарним прикладом та основою для майбутніх проектів з використанням інших пристроїв та технологій IoT для покращення безпеки населення в умовах воєнного стану.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] M. Trinath Basu, R. Karthik, J. Mahitha, and V. Lokesh Reddy, "IoT based forest fire detection system", *International Journal of Engineering and Technology* (UAE), vol. 7, no. 2, pp. 124-126, 2018. [Online]. Available at: https://www.researchgate.net/publication/324054113_IoT_based_forest_fire_detection_system. Accessed on: Sep 10, 2023. doi: <https://doi.org/10.14419/ijet.v7i2.7.10277>.

- [2] K. Sharma, D. Anand, M. Sabharwal, P. Tiwari, O. Cheikhrouhou, and T. Frikha, "A Disaster Management Framework Using Internet of Things-Based Interconnected Devices", *Mathematical Problems in Engineering (Special Issue: Meta-Heuristic Techniques for Solving Computational Engineering Problems 2021)*, vol. 2021, pp. 1-21, 2021. [Online]. Available at: <https://www.hindawi.com/journals/mpe/2021/9916440>. Accessed on: Sep 19, 2023. doi: <https://doi.org/10.1155/2021/9916440>.
- [3] A. Sinha, P. Kumar, N. Rana, R. Islam, and Y. Dwivedi, "Impact of internet of things (IoT) in disaster management: a task-technology fit perspective", *Annals of Operations Research*, 283 (1-2), pp.759-794, 2017. [Online]. Available at: <https://doi.org/10.1007/s10479-017-2658-1>. Accessed on: Oct. 03, 2023. doi: <https://doi.org/10.1007/s10479-017-2658-1>.
- [4] S. Moessner, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): a literature review", *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164-173, 2015. [Online]. Available at: <https://www.scirp.org/journal/PaperInformation?PaperID=56616&#abstract>. Accessed on: Oct. 05, 2023. doi: <https://doi.org/10.4236/jcc.2015.35021>.
- [5] M. U. Farooq, and M. Waseem, "A Review on Internet of Things (IoT)", *International Journal of Computer Applications*, vol. 113, pp. 1-7, 2015. [Online]. Available at: https://www.researchgate.net/publication/273693976_A_Review_on_Internet_of_Things_IoT. Accessed on: Oct. 15, 2023. doi: <https://doi.org/10.5120/19787-1571>.
- [6] The official website of the Python programming language. [Online]. Available at: <https://www.python.org>. Accessed on: Oct. 12, 2023.
- [7] The official Python documentation of the YeeLight library. [Online]. Available at: <https://yeelight.readthedocs.io/en/latest>. Accessed on: Oct. 12, 2023.
- [8] The official Python documentation of the pyTelegramBotAPI. [Online]. Available at: <https://pypi.org/project/pyTelegramBotAPI>. Accessed on: Oct. 15, 2023.
- [9] The official Python documentation for the Auto PY to EXE. [Online]. Available at: <https://pypi.org/project/auto-py-to-exe>. Accessed on: Oct. 15, 2023.

Стаття надійшла до редакції 12.11.2023.

REFERENCE

- [1] M. Trinath Basu, R. Karthik, J. Mahitha, and V. Lokesh Reddy, "IoT based forest fire detection system", *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 2, pp. 124-126, 2018. [Online]. Available at: https://www.researchgate.net/publication/324054113_IoT_based_forest_fire_detection_system. Accessed on: Sep 10, 2023. doi: <https://doi.org/10.14419/ijet.v7i2.7.10277>.
- [2] K. Sharma, D. Anand, M. Sabharwal, P. Tiwari, O. Cheikhrouhou, and T. Frikha, "A Disaster Management Framework Using Internet of Things-Based Interconnected Devices", *Mathematical Problems in Engineering (Special Issue: Meta-Heuristic Techniques for Solving Computational Engineering Problems 2021)*, vol. 2021, pp. 1-21, 2021. [Online]. Available at: <https://www.hindawi.com/journals/mpe/2021/9916440>. Accessed on: Sep 19, 2023. doi: <https://doi.org/10.1155/2021/9916440>.
- [3] A. Sinha, P. Kumar, N. Rana, R. Islam, and Y. Dwivedi, "Impact of internet of things (IoT) in disaster management: a task-technology fit perspective", *Annals of Operations Research*, 283 (1-2), pp.759-794, 2017. [Online]. Available at: <https://doi.org/10.1007/s10479-017-2658-1>. Accessed on: Oct. 03, 2023. doi: <https://doi.org/10.1007/s10479-017-2658-1>.
- [4] S. Moessner, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): a literature review", *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164-173, 2015. [Online]. Available at: <https://www.scirp.org/journal/PaperInformation?PaperID=56616&#abstract>. Accessed on: Oct. 05, 2023. doi: <https://doi.org/10.4236/jcc.2015.35021>.

- [5] M. U. Farooq, and M. Waseem, “A Review on Internet of Things (IoT)”, *International Journal of Computer Applications*, vol. 113, pp. 1-7, 2015. [Online]. Available at: https://www.researchgate.net/publication/273693976_A_Review_on_Internet_of_Things_IoT. Accessed on: Oct. 15, 2023. doi: <https://doi.org/10.5120/19787-1571>.
- [6] The official website of the Python programming language. [Online]. Available at: <https://www.python.org>. Accessed on: Oct. 12, 2023.
- [7] The official Python documentation of the YeeLight library. [Online]. Available at: <https://yeelight.readthedocs.io/en/latest>. Accessed on: Oct. 12, 2023.
- [8] The official Python documentation of the pyTelegramBotAPI. [Online]. Available at: <https://pypi.org/project/pyTelegramBotAPI>. Accessed on: Oct. 15, 2023.
- [9] The official Python documentation for the Auto PY to EXE. [Online]. Available at: <https://pypi.org/project/auto-py-to-exe>. Accessed on: Oct. 15, 2023.

ANDRII HOLIATKIN,
ALINA MOSHYNSKA

EXPANDING THE FUNCTIONALITY OF IOT DEVICES IN CONDITIONS OF EMERGENCY SITUATIONS

The article is devoted to the expansion of functional devices of the Internet of Things (IoT) in order to increase the safety of the population in emergency situations, in particular in the context of natural disasters and military threats. After analyzing the latest research and publications, a conclusion was made about the relevance of the development of IoT technologies in the context of modern challenges for the safety of citizens.

Special emphasis is placed on the use of innovative approaches to solving public safety problems, particularly for persons with disabilities, such as the hearing impaired or the elderly. The article describes the implementation of software in the Python programming language to visually display the status of a missile threat or artillery fire using Yeelight smart lamps, which use colored light to convey information about the presence or absence of danger messages. The proposed solutions include synchronous operation of devices in real time, connection to reliable sources of messages and provision of stable communication between devices and the server.

The project demonstrated an innovative approach to solving security problems in crisis situations, in particular martial law, promoting a quick and accurate response to valuable threats and protecting the population.

Keywords: Internet of Things, IoT device programming, Python, smart LED lamp, public awareness during war.

Голяткін Андрій Олексійович, студент магістерської програми, Навчально-науковий інститут телекомунікаційних систем Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”, м. Київ, Україна, ORCID 0009-0008-4955-7818, e-mail: holiatkin_andrii@ukr.net.

Мошинська Аліна Валентинівна, доктор технічних наук, доцент, професор кафедри, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”, м. Київ, Україна, ORCID 0000-0002-6582-9421, e-mail: avmoshinskaya@gmail.com.

Holiatkin Andrii, student of the master's program, Educational and scientific institute of telecommunication systems of the National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.

Moshynska Alina, doctor of technical sciences, associate professor, professor of department, Institute of special telecommunications systems and information protection of the National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.