

СИСТЕМА ВИЯВЛЕННЯ НЕБЕЗПЕЧНИХ РЕЧОВИН У ПОВІТРІ НА БАЗІ ІНТЕРНЕТУ РЕЧЕЙ

Василь Яцишин

бакалавр кафедри електронних пристроїв та систем

Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», вулиця Академіка Янгеля, 16/9, корпус № 12, Київ, 03056, v.jatsishin@gmail.com

ORCID: 0000-0001-6294-5083

Юлія Ямненко

професор кафедри електронних пристроїв та систем

Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», вулиця Академіка Янгеля, 16/9, корпус № 12, Київ, 03056, yus315171-eds@iill.kpi.ua

ORCID: 0000-0002-9796-6420

Ганна Сарибоба

старший викладач кафедри електронних пристроїв та систем

Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», вулиця Академіка Янгеля, 16/9, корпус № 12, Київ, 03056, sarigana-eds@iill.kpi.ua

ORCID: 0000-0003-0805-7899

Ця стаття присвячена проєктуванню та розробці системи моніторингу вмісту небезпечних речовин – газу та диму (MQ-2) – у системі побутової автоматизації типу «розумний будинок», а також моніторингу й інших параметрів, таких як температура повітря в кімнаті, за допомогою датчика LM35 та його вологість з датчиком DHT11. Програмування такого проєкту було здійснене у середовищі Android Studio та Arduino IDE. Така система дозволяє дистанційно за допомогою телефону відстежувати дані про температуру та вологість у кімнатах, попереджувати користувача про можливу зміну якості повітря в кімнаті. У цьому проєкті використовується база даних на основі Google Firebase, позаяк таку платформу надзвичайно просто інтегрувати в додатки, які розробляються для операційної системи Android. До такої системи моніторингу було додано можливість реєстрації та авторизації у власний кабінет користувача. Розроблено мобільний додаток під операційну систему Android для здійснення моніторингу різних параметрів, в якому передбачено сповіщення про значні зміни температури, вологості, якості повітря (вище заданого порогу). В роботі також описано структурну та принципову схеми, а також алгоритм роботи програмного забезпечення. Було розроблено блок схеми для пояснення принципу роботи програми. Наведено скріншоти емулятора середовища розробки Android Studio. Наведений рисунок, який пояснює принцип роботи такої системи, а саме як користувач взаємодіє з нею. Описані характеристики всіх датчиків, які задіяні у цьому дослідженні. В проєктуванні також було використано мікросхему NodeMCU ESP8266, яка збирає дані з датчика і, крім цього, передає дані до бази даних. Датчики, які не показані на скрінах, підключаються аналогічним способом, як на принциповій схемі.

Ключові слова: система побутової автоматизації, моніторинг вмісту небезпечних речовин у повітрі, Android Studio, Arduino IDE, Google Firebase.

Сучасні системи інтелектуального керування електротехнічними пристроями на базі концепції Інтернету речей (Internet of things – IoT) [1] дозволяють вирішувати складні задачі багато-параметричного керування та прийняття рішень для багатьох застосувань – від побутових користувачів «розумних будинків» до диспетчерських центрів, промислових виробничих ділянок, автономних електротехнічних комплексів та систем [2; 3; 4]. Реалізація концепції IoT дозволяє створити єдину інформаційно-комунікаційну інфраструктуру для забезпечення узгодженості функціонування всіх пристроїв та підсистем за єдиним

критерієм ефективності, яким може виступати енергетична ефективність, вартісні фактори або інша цільова функція [5].

Важливу роль у функціонуванні автоматизованої інтелектуальної системи прийняття рішень та керування відіграє належним чином організований моніторинг параметрів, що впливають на алгоритм керування. Але, окрім важливості для ефективності керування, необхідним є також моніторинг небезпечних речовин у повітрі, зокрема, газу та диму.

Стаття присвячена розробці системи виявлення цих небезпечних речовин у рамках системи

побутової автоматизації. Система моніторингу базується на постійному спостереженні за температурою та вологістю в різних приміщеннях, а також реагує на сигнали від датчиків розбиття вікон, датчиків руху, датчиків відкриття для забезпечення охоронної функції. Система також реагує на протікання газу в приміщенні та накопичення диму від горіння відповідно за допомогою датчиків газу та диму. У разі відхилення показників від норми система реагує відповідними діями – певною мірою знешкоджує джерело неполадки та сигналізує про відповідну подію [6].

Метою розробки є створення простого та надійного пристрою моніторингу вмісту газу та диму у системі автоматизації «розумного будинку».

Для початку потрібно розглянути наявні системи, як вони були розроблені та історію їх створення.

Наявні рішення у сфері систем безпеки та моніторингу забезпечують безпечно та комфортно функціонування, але в кожній системі є свої недоліки, такі як відсутність зручного інтерфейсу для користувача, низька енергоефективність та інше [3; 8].

Основною перевагою систем побутової автоматизації є зручність та легкість керування електротехнічними пристроями та підсистемами різного функціонального призначення. Незважаючи на швидкі темпи розвитку автоматизованих систем прийняття рішень, системи побутової автоматизації найчастіше використовують диспетчерське керування та/або передбачають можливість втручання користувача.

Найпростішим способом реалізації такого «втручання» є кнопкові або сенсорні панелі. Іншим способом є пульт дистанційного керування з кнопковим або сенсорним екраном. Деякі моделі пультів керування можуть відображати відео з камер у реальному часі [3].

Третім, найзручнішим та найбільш сучасним способом, є керування за допомогою мобільного додатка. За допомогою підключення до мережі Інтернет забезпечується сповіщення користувача про показники з датчиків, а також керування виконавчими пристроями системи. Також за наявності доступу до мережі є можливість дистанційного перегляду відео з камер спостереження.

Технічна реалізація системи моніторингу базується на використанні датчиків руху, температури та вологості, диму. Функції моніторингу та керування забезпечуються використанням мікроконтролера NodeMCU ESP8266 з інтерфейсом WiFi та розробленого додатка для смартфона [7]. Мікроконтролер підключається до мережі Wi-Fi та встановлює з'єднання TCP/IP.

Показники датчиків синхронізуються з базою даних (БД) у реальному часі, при цьому дані

оновлюються автоматично. Така функція забезпечується базою даних Realtime Database від Firebase та мікроконтролером (МК) з інтерфейсом WiFi від виробника Espressif Systems [8]. На рис. 1. представлена блок-схема, яка ілюструє зв'язок між інтерфейсом користувача, мікроконтролером, базою даних та датчиками системи.

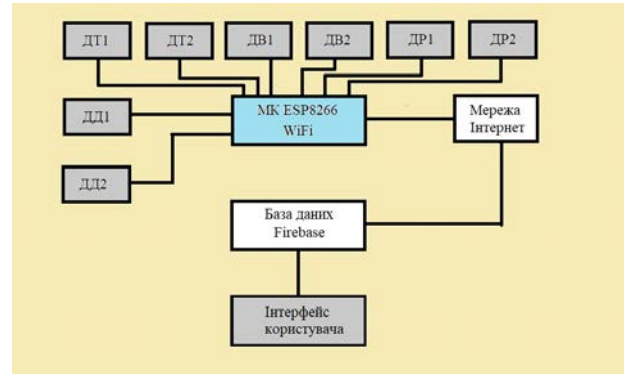


Рис. 1. Блок-схема взаємодії інтерфейсу користувача із мікроконтролером та базою даних

У блок-схемі використовуються вісім датчиків, які передбачаються встановлювати в різних приміщеннях будинку. Датчики ДТ1 та ДТ2 є датчиками температури типу LM35 [9]. Датчики ДВ1 та ДВ2 забезпечують вимірювання вологості в приміщенні. Було досліджено датчики типу DHT11, які можуть вимірювати температуру та вологість, але в датчика DHT11 більша похибка вимірювання, ніж у датчика LM35. Крім того, прецизійний інтегральний датчик температури LM35 має широкий діапазон – він забезпечує вимірювання температури з точністю $\pm 0,25^{\circ}\text{C}$ за кімнатних умов. У середовищі Arduino IDE відбувається перетворення напруги на температуру за допомогою формули (1.1.):

$$Tempin = \left[\frac{V_{out} \text{ in mV} - 500}{10} \right], \quad (1.1.)$$

де V_{out} – вихідна напруга на датчику в мілівольтах.

Чк датчики диму ДД1 та ДД2 було вибрано MQ-2 [9], які мають досить високу точність (300-10000 ppm) та середню дальність спрацювання, а також невисоку ціну. Такий датчик має 4 виходи, а саме VCC, GND, A0 та D0. Виходи A0 та D0, аналоговий та цифровий відповідно.

Датчики ДР1 та ДР2 є інфрачервоними (ІЧ) датчиками руху типу HC-SR505. Вони мають високий діапазон робочих температур, дальність спрацювання до 3 м, а також кут огляду близько 120° . Має три виходи VCC, GND, OUT. OUT є цифровим виходом.

Всі ці датчики підключаються до МК NodeMCU ESP8266. MQ-2 підключаємо на анало-

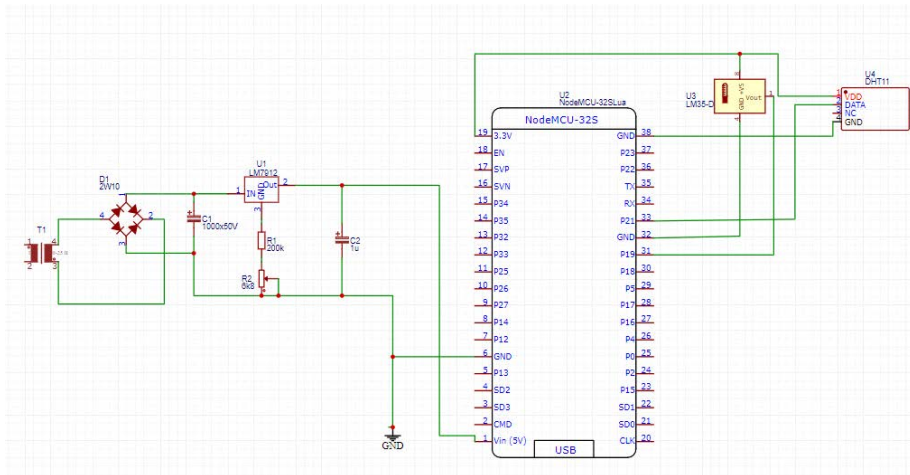


Рис. 2. Схема підключення датчиків та блоку живлення до мікроконтролера

говий порт, LM35 – аналоговий порт, HC-SR505 – цифровий порт, DHT11 – цифровий порт. Кількість портів у цьому мікроконтролері достатня для підключення всіх 4 датчиків одночасно. Зчитані дані відправляються до бази даних Firebase через модуль WiFi.

Програмування мікроконтролерного модуля NodeMCU ESP8266 здійснюється в програмному середовищі Arduino IDE [11]. Схему підключення датчиків, а також блоку живлення зображено на рис. 2. Аналогічно підключаються всі інші датчики.

Для роботи у середовищі Arduino IDE потрібен USB-кабель і плата NodeMCU. Також потрібно встановити бібліотеки для програмування плат [10].

Елемент DHT11 – це цифровий датчик вологості та температури, який складається з емнісного датчика вологості та термістора із вбудованим АЦП. Інфрачервоний датчик руху HC-SR505 дозволяє виявити рух людини або тварини на відстані до 3 м. Тривалість імпульсу у разі виявлення лежить у діапазоні 8–12 с залежно від налаштування. Кут розпізнавання сягає 100°. Робоча температура датчика – від -20 до +80°C, струм споживання – приблизно 50 мА.

Датчик диму MQ-2 використовується для розпізнавання горючого газу та диму. Він має широкий діапазон розпізнавання, короткий час реакції, високу чутливість.

Першим етапом роботи програмного забезпечення є ініціалізація, що містить три етапи. Першим етапом є ініціалізація бібліотек, що потрібні для правильного виконання всіх команд, а також введення параметрів бази даних та мережі WiFi – host, унікального ключа мережі, SSID та пароля доступу. Другий етап містить

ініціалізацію даних – встановлюється, до якого піна мікроконтролера підключений вихід датчика, змінні, які відповідають за вимірювання, а також змінні, які виконують передачу даних у БД. На третьому етапі здійснюється перевірка наявності підключення до мережі WiFi. У разі успішного підключення відбувається перехід до основного етапу програми.

На основному етапі проводиться вимірювання даних з датчиків та конвертація сигналу. Усі вище наведені датчики ініціалізовані та готові до роботи після підключення.

У заключному блоці проводиться відправка даних через мережу за вказаним шляхом до БД Firebase [7].

Під час створення інтерфейсу для користувача було вибрано середовище програмування Android Studio, мова програмування – Java [11; 12; 13]. Головний екран програми зображений на рисунку 3.

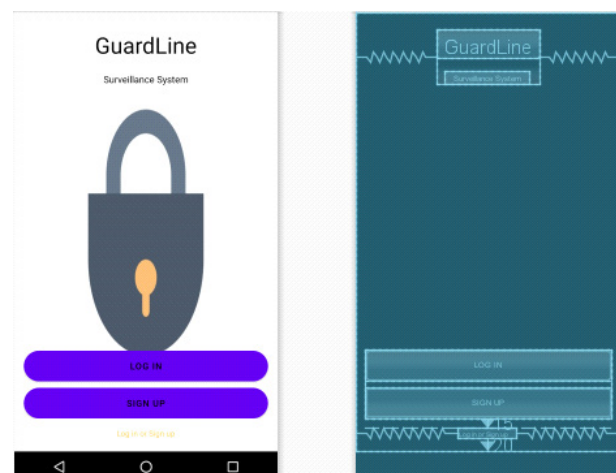


Рис. 3. Головний екран програми

У додатку була реалізована можливість реєстрації та входу до облікового запису (рис. 4.). Під час реєстрації потрібно заповнити такі дані, як email, password, phone number, first and second name.

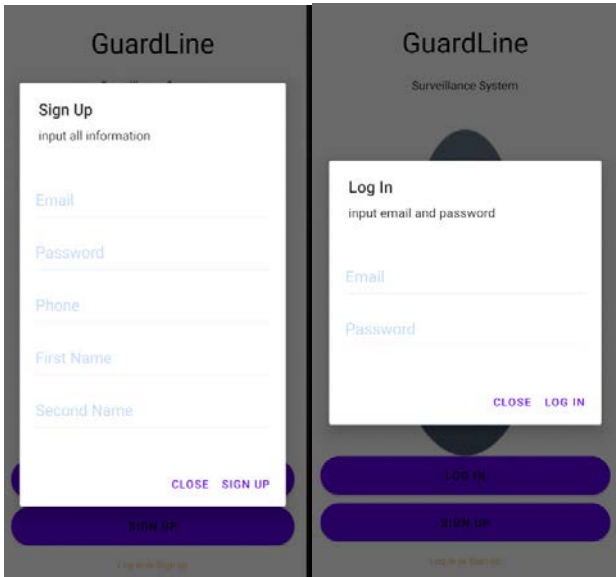


Рис. 4. Діалогові вікна реєстрації та авторизації користувача

Після успішної реєстрації потрібно авторизуватись у системі. Коли авторизацію пройдено, відкривається нове вікно користувача, в якому є кнопки, кожна з яких відповідає своєму параметру (рис. 5).

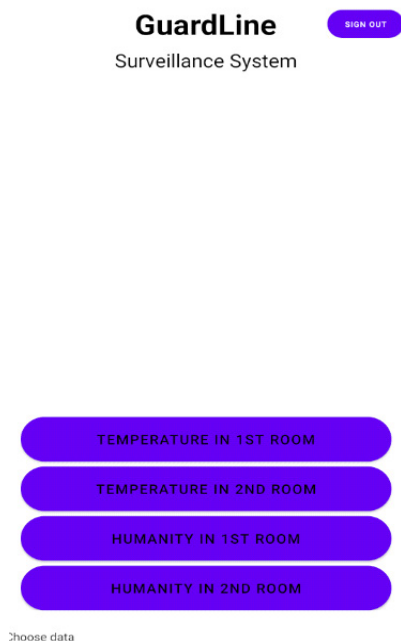


Рис. 5. Головне меню

Список параметрів, які відображаються під час натискання на відповідні кнопки, буде відображатись так, як показано на рис. 6.



Рис. 6. Список даних про температуру

Після завершення розробки системи було виявлено декілька недоліків. Першим і головним недоліком є база даних, адже БД Firebase є безкоштовною та призначена для невеликих проєктів. Вона не розрахована на велику кількість запитів, і у разі суттєвого збільшення їхньої кількості трапляються переривання та «зависання» додатка. Рішенням такої проблеми є створення власної бази даних та написання серверної частини для роботи з додатком. Для створення бази даних потрібно завантажити програму MySQL. Програмування серверної частини дозволяє зберігати інформацію в базі даних та динамічно створювати та повертати HTML та інші типи файлів. Існує можливість отримання даних у форматах JSON та XML.

Другою проблемою є низька точність датчиків. Дійсно, недорогі датчики не можуть забезпечити високу точність, але цю проблему можна частково вирішити програмно.

Третьою проблемою є відсутність автономності системи. У разі відключення джерела енергії система не зможе передавати дані до БД, відповідно, моніторинг припиниться. Рішенням цієї проблеми є додаткове підключення ланцюгу резервного живлення від акумулятора.

Для підвищення надійності та забезпечення безперебійності надходження даних планується також доповнити систему модулем GSM, що

дозволить за відсутності Інтернет-зв'язку забезпечити надходження сповіщень на телефон користувача.

На базі модуля NodeMCU ESP8266 були підключені датчики для моніторингу температури, вологості, повітря та руху. Запрограмований мікроконтролер має можливість відправляти дані з датчиків у базу даних Firebase. За допомогою інтегрованого середовища розробки Android Studio розроблено додаток для смартфона на базі ОС Android. У мобільному додатку реалізована реєстрація користувачів за введеними даними, а саме електронною поштою та паролем.

ЛІТЕРАТУРА

1. Kurniawan A. Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32. Packt Publishing, 2019. 252 с.
2. Ajax Systems. URL: <https://ajax.systems/ru-ua/>.
3. Granzer W.P. Security in building automation systems. Munich : Apress, 2018. 578 с.
4. Sourav K.B. Internet of things: an application based approach using arduino platform and firebase. Independently published, 2018. 109 с.
5. Yamnenko J., Tereshchenko T., Klepach L. and Palii D. "Forecasting of electricity consumption in SmartGrid", 2017 International Conference on Modern Electrical and Energy Systems (MEES), Kremenchuk, Ukraine, 2017, pp. 208–211. DOI: 10.1109/MEES.2017.8248891.
6. Yamnenko J., Globa L., Kurdecha V. and Zakharchuk A. "Data Processing in IoT Systems based on Fuzzy Logics", 2019 Modern Electric Power Systems (MEPS), 2019, pp. 1–4. DOI: 10.1109/MEPS46793.2019.9395055.
7. Smyth N. Firebase Essentials – Android Edition. CreateSpace Independent Publishing Platform, 2017. 534 с.
8. Patrick J.R. Home attitude: everything you need to know to make your home smart. CreateSpace Independent Publishing Platform, 2017. 193 с.
9. Datasheet. URL: <https://www.alldatasheet.com/>.
10. Arduino. URL: <https://www.arduino.cc/en/software>.
11. Lucas L. JavaFx Special Effects. London : Apress, 2017. 432 с.
12. Schildt H. Java: A Beginner's Guide. Eighth Edition, 2018. 720 с.
13. Schildt H. Java: The Complete Reference. McGraw-Hill Education, 2018. 1208 с.

DETECTION SYSTEM OF HAZARDOUS SUBSTANCES IN THE AIR BASED ON THE INTERNET OF THINGS

Yatsyshyn Vasyl

Bachelor of the Department of Electronic Devices and Systems

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy avenue, Kyiv, 03056, v.jatsishin@gmail.com

ORCID: 0000-0001-6294-5083

Yulia Yamnenko

Professor of the Department of Electronic Devices and Systems,

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 16/9 Akademika Yangel str., building 12, Kyiv, 03056, yys315171-eds@ill.kpi.ua

ORCID: 0000-0002-9796-6420

Sariboga Hanna

Senior Lecturer at the Department of Electronic Devices and Systems,

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 16/9 Akademika Yangela Street, building 12, Kyiv, 03056, sarigana-eds@ill.kpi.ua

ORCID: 0000-0003-0805-7899

This article describes the development of an indoor monitoring system. Based on the NodeMCU ESP8266 module, sensors connected to monitor temperature, humidity, air and motion. The programmed MCU has the ability to send data from the sensors to the Firebase database. An application for a smartphone based on Android OS developed using the integrated development environment Android Studio. The mobile application implements the registration of users by the entered data, namely email and password.

The main advantages of this system are cheapness, because all the modules and sensors that are used are relatively inexpensive and can be purchased by anyone. The second advantage is the storage of database data, and the ability to track parameters in any place where there is an Internet connection. In performing the tasks of this work, we used a program for programming MCs (Ardiuno IDE), as well as a program for developing a mobile application (Android Studio), a program for creating an electrical circuit (EasyEDA), and programs for creating block diagrams. The system was tested in real conditions and installed in a residential building. As expected, the system can work with a slight delay due to the speed of data transmission over the Internet. But with proper installation, it is possible to reduce this delay in real-time broadcasting.

Key words: advantages, cheapness, sensors, development, application, smartphone.

REFERENCES

1. Kurniawan, A. (2009). Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32. Packt Publishing.
2. Ajax System. (2011). Retrieved from: <https://ajax.systems/ru-ua/>.
3. Granzer, W.P. (2018). Security in Building Automation Systems. Munich: Apress.
4. Sourav, K.B. (2018). Internet of Things: An Application Based Approach Using Arduino Platform and Firebase. Independently published.
5. Yamnenko, J., & Tereshchenko, T., & Klepach, L., & Pali, D. (2017). Forecasting of electricity consumption in SmartGrid. International Conference on Modern Electrical and Energy Systems (MEES), 208–211. DOI: 10.1109/MEES.2017.8248891.
6. Yamnenko, J., & Globa, L., & Kurdecha, V., & Zakharchuk, A. (2019). Data Processing in IoT Systems based on Fuzzy Logics. Modern Electric Power Systems (MEPS), 1–4. DOI: 10.1109/MEPS46793.2019.9395055.
7. Smyth, N. (2017). Firebase Essentials – Android Edition. CreateSpace Independent Publishing Platform.
8. Patrick, J.R. (2017). Home Attitude: Everything You Need To Know To Make Your Home Smart. CreateSpace Independent Publishing Platform.
9. Datasheet. Retrieved from: <https://www.alldatasheet.com/>.
10. Arduino. Retrieved from: <https://www.arduino.cc/en/software>.
11. Lucas, L. (2017). JavaFx Special Effects. London: Apress.
12. Schildt, H. (2018). Java: A Beginner’s Guide. Eighth Edition.
13. Schildt, H. (2018). Java: The Complete Reference. McGraw-Hill Education.

Стаття надійшла 17.02.2023