

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-56-10>

УДК 004.72

Багнюк Наталія Володимирівна, к.т.н., доцент

<https://orcid.org/0000-0002-7120-5455>

Лавренчук Світлана Василівна, к.т.н., доцент

<https://orcid.org/0000-0002-5453-3924>

Міскевич Оксана Іванівна, ст. викладач

<https://orcid.org/0000-0002-5009-2391>

Свистун Богдан Васильович, магістрант

Луцький національний технічний університет, м.Луцьк, Україна

РЕАЛІЗАЦІЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» З ВИКОРИСТАННЯМ BLUETOOTH, ARDUINO ТА ANDROID: ВІД ДАТЧИКІВ ДО МОБІЛЬНОГО ІНТЕРФЕЙСУ

Багнюк Н. В., Лавренчук С. В., Міскевич О. І., Свистун Б. В. Реалізація системи «Розумний будинок» з використанням Bluetooth, Arduino та Android: від датчиків до мобільного інтерфейсу. Описана розробка програмного забезпечення для системи «Розумний будинок» на базі платформи Arduino Uno. Основна увага приділяється написанню скетчу для взаємодії з датчиками руху, температури та вологості, які забезпечують моніторинг і контроль умов у приміщенні. Створено мобільний додаток для операційної системи Android, що функціонує як Bluetooth термінал для збору, відображення та аналізу даних, отриманих від датчиків. Реалізовано інтерфейс для зручного налаштування та підключення пристроїв, а також забезпечено можливість візуалізації зібраної інформації в реальному часі. Проведено тестування системи в умовах реальної експлуатації, що підтвердило її стабільну роботу та точність вимірювань. У статті також наведено аналіз можливостей подальшого розширення функціоналу, таких як інтеграція з іншими системами автоматизації та підтримка додаткових типів датчиків. Представлена розробка має практичне значення для побудови інтелектуальних систем управління середовищем у житлових та комерційних приміщеннях.

Ключові слова: Arduino, розумний будинок, Bluetooth, мобільний додаток, датчики, автоматизація, контроль середовища

Bahniuk N., Lavrenchuk S., Miskevych O., Svystun B. Implementation of the «Smart House» system using Bluetooth, Arduino, and Android: from sensors to mobile interface. The article describes the development of software for the Smart Home system based on the Arduino Uno platform. The main focus is on writing a sketch for interaction with motion, temperature and humidity sensors that provide monitoring and control of indoor conditions. A mobile application for the Android operating system has been created that functions as a Bluetooth terminal for collecting, displaying and analysing data received from the sensors. An interface was implemented for easy configuration and connection of devices, as well as the ability to visualise the collected information in real time. The system has been tested in real-world operation, which has confirmed its stable operation and measurement accuracy. The article also analyses the possibilities of further expanding the functionality, such as integration with other automation systems and support for additional types of sensors. The presented development is of practical importance for the construction of intelligent environmental management systems in residential and commercial premises.

Keywords: Arduino, smart home, Bluetooth, mobile application, sensors, automation, environment control

Постановка наукової проблеми. Система «Розумний будинок» набуває все більшої популярності завдяки можливості автоматизації рутинних завдань і підвищенню рівня комфорту та безпеки житла. Одним з важливих аспектів таких систем є моніторинг навколишнього середовища в приміщенні та забезпечення своєчасного реагування на зміни умов, таких як температура, вологість та рух. Існуючі рішення, часто складні та дорогі, вимагають застосування спеціалізованого обладнання та програмного забезпечення. Таким чином, постає наукова задача розробити прототип доступної та простої у використанні системи «Розумний будинок» на базі мікроконтролера Arduino Uno з використанням модуля Bluetooth для передачі даних. Основною метою є створення системи, яка зможе ефективно моніторити рух, температуру та вологість у приміщенні і передавати ці дані на мобільний додаток, забезпечуючи зручний інтерфейс для користувача.

Аналіз останніх досліджень і публікацій. Існуючі дослідження у сфері систем «Розумний будинок» демонструють широкий спектр технологій та підходів до автоматизації домашнього середовища [1]. Більшість комерційних рішень включають складні системи з багатьма датчиками та контролерами, що взаємодіють через різні протоколи зв'язку, такі як Wi-Fi, ZigBee, або Z-Wave [2-3]. Проте ці рішення часто є дорогими та вимагають спеціалізованих знань для їх встановлення та налаштування. Arduino Uno, у поєднанні з доступними датчиками та модулями зв'язку, пропонує економічно ефективну та гнучку платформу для побудови системи «Розумний будинок». Використання модуля Bluetooth HC-05 для передачі даних забезпечує простоту підключення та управління пристроями за допомогою мобільних додатків [4]. Попередні дослідження показали, що

такий підхід дозволяє створювати функціональні прототипи систем моніторингу та управління домашнім середовищем з мінімальними витратами та складністю.

Виділення невирішених раніше частин загальної проблеми. Попри численні дослідження в області автоматизації та контролю, існує низка невирішених питань, що стосуються інтеграції платформи Arduino з різноманітними датчиками та передачею даних у реальному часі через Bluetooth-з'єднання. Однією з основних проблем є забезпечення стабільної і надійної передачі даних між датчиками, встановленими в системі «Розумний будинок», і мобільним додатком, який виконує роль терміналу для їхнього моніторингу. Традиційні рішення для систем домашньої автоматизації часто вимагають складних і дорогих апаратних компонентів або мережевих протоколів, таких як Wi-Fi або Ethernet. Однак вони можуть бути надмірними для невеликих систем контролю, де головна мета полягає в моніторингу декількох параметрів навколишнього середовища. Водночас використання Bluetooth для передачі даних у таких системах залишалося недостатньо дослідженим через обмежену дальність дії і складнощі забезпечення безперервного зв'язку в умовах перешкод і обмежень енергоспоживання.

Мета дослідження полягає у розробці та впровадженні прототипу системи «Розумний будинок» на базі плати Arduino Uno з використанням стандарту зв'язку Bluetooth для моніторингу та передачі даних про рух за допомогою інфрачервоного датчика руху SR312, температуру та вологість за допомогою датчика DHT11 [5-6]. Система повинна забезпечувати збір даних, надсилання зібраних даних через модуль Bluetooth на мобільний додаток, розроблений на платформі Android, відображення та інтерпретація даних в читабельному вигляді в додатку для користувача [7].

Основна частина дослідження. Для написання прошивки для плати Arduino Uno було використано інтегроване середовище розробки Arduino IDE. Це середовище спеціально створене для написання скетчів для плат сімейства Arduino і містить вбудований редактор коду, консоль, бібліотеки тощо. Процес написання програмного коду для мікроконтролера Arduino, який виконує роль головного модуля системи, починається з підключення бібліотек та створення констант. Бібліотека <DHT.h> необхідна для роботи з датчиком вологості і температури DHT11. Бібліотека <SoftwareSerial.h> потрібна для реалізації послідовного зв'язку на цифрових контактах плати Arduino Uno R3. Процес підключення бібліотек показано на рисунку 1.

```
#include <SoftwareSerial.h>
#include <DHT.h>
#include <DHT_U.h>
```

Рис. 1. Підключення бібліотек у середовищі розробки Arduino

Після підключення бібліотек необхідно ініціалізувати змінні з певними значеннями, а саме, присвоїти їм номери пінів на платі Arduino Uno R3, які будуть використовувати в роботі для підключення елементів системи до мікроконтролера, об'єкти класів для використання елементів, таких як датчики та адаптер стандарту зв'язку Bluetooth. Ініціалізуються пини RX та TX для Bluetooth-адаптера для реалізації обміну даними з іншими пристроями. RX та TX пини відповідають за прийом та відправку даних відповідно. Після цього створюється клас для Bluetooth-модуля. Це видно на рисунку 2.

```
5 // Визначення пінів і типу датчика
6 #define DHTPIN 4 // Пін, до якого підключений датчик DHT11
7 #define DHTTYPE DHT11 // Визначення типу датчика DHT11
8 #define MOTION_SENSOR_PIN 2 // Пін, до якого підключений датчик руху
9 #define LED_PIN 13 // Вбудований LED пін
10
11 // Визначення пінів для Bluetooth HC-05
12 #define BLUETOOTH_TX_PIN 0 // Пін для передачі даних Bluetooth
13 #define BLUETOOTH_RX_PIN 1 // Пін для прийому даних Bluetooth
14
```

Рис. 2. Присвоєння пінів для датчиків та адаптера Bluetooth

Функція `setup()` виконується один раз під час запуску програми на платі Arduino. Вона використовується для налаштування початкових параметрів та ініціалізації обладнання. Функція `setup()` у коді виконує наступні дії:

- Встановлює пін для зовнішнього світлодіода як вихідний. Встановлює пін для датчика руху як вхідний.
- Ініціалізує серійне з'єднання для моніторингу.
- Ініціалізує датчик DHT11 для зчитування температури та вологості. Ініціалізує програмний серійний порт для Bluetooth з'єднання.
- Вимикає вбудований світлодіод.

Ці кроки забезпечують правильну роботу всіх компонентів системи перед початком виконання основного циклу програми. Функцію `setup()` можна побачити на рисунку 3.

```
18
19 void setup() {
20     pinMode(MOTION_SENSOR_PIN, INPUT); //
21     pinMode(LED_PIN, OUTPUT); //
22     Serial.begin(9600); //
23     bluetoothSerial.begin(9600); //
24     dht.begin(); //
25 }
```

Рис. 3. Функція `setup()`

Після налаштування необхідних компонентів у функції `setup()`, програма переходить до виконання основної логіки у циклі `loop()`. Під час кожного циклу виконуються кілька ключових завдань. Спочатку зчитується стан датчика руху за допомогою функції `detectMotion()`, яка визначає наявність руху у приміщенні. Якщо рух виявлено, вбудований світлодіод на платі Arduino вмикається, якщо ні – вимикається. Далі зчитуються значення вологості та температури з датчика DHT11 за допомогою функцій `readHumidity()` та `readTemperature()`. У разі успішного зчитування даних формується повідомлення, яке містить інформацію про стан руху, температуру та вологість, за допомогою функції `createMessage()`. Це повідомлення виводиться на серійний монітор для відлагодження та передається через модуль Bluetooth на мобільний додаток. Наприкінці циклу встановлюється затримка на 6 секунд перед початком нового циклу для уникнення помилок спрацювання. Функція `loop()` наведена на рисунку 4.

```
void loop() {
    // Зчитування значень датчика руху
    int motionState = digitalRead(MOTION_SENSOR_PIN);

    if (motionState == HIGH) { // якщо датчик виявляє рух
        digitalWrite(LED_PIN, HIGH); // вмикаємо вбудований LED
        //Serial.println("Motion detected!"); // виводимо повідомлення в серійний монітор
    } else {
        digitalWrite(LED_PIN, LOW); // вимикаємо вбудований LED
        //Serial.println("No motion."); // виводимо повідомлення в серійний монітор
    }

    // Затримка для стабілізації датчика DHT11
    delay(2000);

    // Зчитування значень датчика DHT11
    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature();

    // перевірка на наявність помилок зчитування
    if (isnan(humidity) || isnan(temperature)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    // формування повідомлення для виведення в серійний монітор і передачі через Bluetooth
    String message = "Motion: ";
    if (motionState == HIGH) {
        message += "Motion detected, ";
    } else {
        message += "No motion, ";
    }
    message += "Humidity: " + String(humidity) + " %, Temperature: " + String(temperature) + " *C";
    Serial.println(message);
    bluetoothSerial.println(message); // відправка повідомлення через Bluetooth

    delay(6000); // невелика затримка для уникнення помилкових спрацювань
}
```

Рис. 4. Функція `loop()`

Після завершення написання коду прошивки для плати Arduino, на основі якої побудовано програмну систему «розумного будинку», наступним кроком є створення мобільного додатку для операційної системи Android. Для розробки мобільного додатку використано середовище Android Studio та мову програмування Java. Перед розглядом деталей процесу, визначається загальна структура мобільного додатку.

Спершу створюється конфігураційний файл для мобільного додатку, який називається «AndroidManifest.xml». Цей файл містить важливу інформацію про додаток та визначає його архітектуру, включаючи назву, іконку, дозволи для використання бібліотек, доступ до Інтернету, камери, Bluetooth тощо. Під час встановлення або роботи додатку користувачам буде показано запит на дозвіл. Маніфест також декларує всі основні робочі директорії додатку, такі як MainActivity, Service, Socket, BroadcastReceiver, BluetoothUnit, DevicesFragment, ContentProvider тощо. Це дозволяє системі Android знати, які компоненти є частиною додатку. Крім того, в маніфесті визначається мінімальна версія Android, на якій буде працювати додаток, інформація про сумісність з різними екранами та загальні метадані. Після створення конфігураційного файлу, створюються з графічних елементів додатку за допомогою вбудованої функції створення UI-елементів в Android Studio. В директорії res є файли формату xml, які описують колір, розмір, властивості UI-елементів. Це видно на рисунку 5.

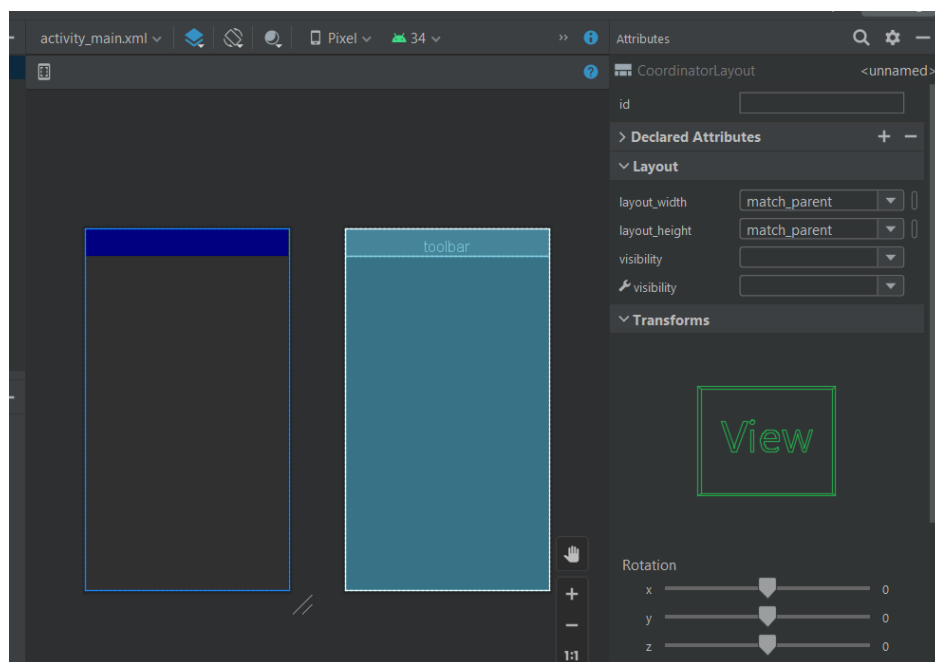


Рис. 5. Створення графічних елементів застосунку для вкладок додатку

Далі переходимо до папки MainActivity. З назви папки стає зрозуміло, що це основна папка нашого проєкту. Код в ній використовує фрагменти різного функціоналу додатку та обробляє навігацію між ними. На рисунку 6 імпортуються необхідні бібліотеки і класи для роботи з фрагментами (FragmentManager), базовою активністю (AppCompatActivity), тулбаром (Toolbar) та іншими компонентами.

```
import android.os.Bundle;
import androidx.fragment.app.FragmentManager;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
```

Рис. 6. Імпорт необхідних бібліотек

Після імпорту бібліотек і класів прописується логіка взаємодії між елементами додатку. Реалізацію логіки для взаємодії елементів зображено на рисунку 7.

```
public class MainActivity extends AppCompatActivity implements FragmentManager.OnBackStackChangedListener {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Toolbar toolbar = findViewById(R.id.toolbar);  
        setSupportActionBar(toolbar);  
        getSupportFragmentManager().addOnBackStackChangedListener(this);  
        if (savedInstanceState == null)  
            getSupportFragmentManager().beginTransaction().add(R.id.fragment, new DevicesFragment(), tag: "d  
        else  
            onBackStackChanged();  
    }  
}
```

Рис. 7. Написання функції onCreate

Клас MainActivity розширює клас AppCompatActivity, забезпечуючи сумісність з попередніми версіями Android без конфліктів між старими і сучасними функціями операційної системи. Крім того, він реалізує інтерфейс FragmentManager.OnBackStackChangedListener, що дозволяє відстежувати зміни у бек-стеку фрагментів. Функція onCreate викликається при створенні активності та встановлює макет, визначений у файлі activity_main.xml, за допомогою setContentView(R.layout.activity_main).

Toolbar налаштовується як ActionBar за допомогою команди setSupportActionBar(toolbar) і додається перехоплювач змін у бек-стеку фрагментів.

Метод onBackStackChanged викликається при зміні бек-стеку фрагментів, в той час як метод onSupportNavigateUp обробляє натискання кнопки «Назад» в ActionBar. методом onBackPressed, який реалізований через булеві значення true і false. Реалізацію цих двох методів можна побачити на рисунку 8.

```
@Override  
public void onBackStackChanged() {  
    getSupportFragmentManager().setDisplayHomeAsUpEnabled(getSupportFragmentManager().getBackStackEntryCount()>0);  
}  
  
1 usage  
@Override  
public boolean onSupportNavigateUp() {  
    onBackPressed();  
    return true;  
}
```

Рис. 8. Написання методів onBackStackChanged та onSupportNavigateUp

У процесі розробки системи «Розумний будинок» на базі плати Arduino Uno з використанням стандарту зв'язку Bluetooth використано два типи Bluetooth терміналів: кастомний термінал, створений власноруч, та стандартний термінал, завантажений із Google Play. Порівняльна характеристика кастомного і стандартного Bluetooth-терміналів наведена в таблиці 1.

Таблиця 1. Порівняльна характеристика кастомного і стандартного Bluetooth-терміналів

Характеристики	Кастомний термінал	Bluetooth-	Стандартний Bluetooth-термінал

Інтерфейс користувача	Призначений для конкретних потреб проєкту, зручний та адаптований для керування системою	Загальний інтерфейс, призначений для роботи з різними пристроями, без можливості кастомізації під конкретний проєкт
Функціонал	Відображення даних з датчиків у реальному часі, можливість зворотнього зв'язку та налаштувань	Підключення до семи пристроїв одночасно, базове відображення даних, без спеціалізованих функцій
Гнучкість	Легке налаштування і розширення функціоналу відповідно до потреб проєкту	Обмежені можливості для кастомізації та розширення, складність внесення змін
Безпека	Реалізовані базові заходи безпеки, можливість додаткового налаштування під конкретні вимоги	Базові заходи безпеки, які можуть бути недостатніми для специфічних потреб
Підключення	Підключення через стандарт Bluetooth	Підключення через стандарт Bluetooth та Bluetooth Low Energy
Взаємодія з пристроями	Підключення до одного пристрою одночасно	Можливість підключення до 7 модулів Bluetooth одночасно

Візуалізацію порівняння характеристик двох терміналів можна побачити на рисунку 9.



Рис. 9. Графік з порівнянням кастомного і стандартного терміналів

Висновки та перспективи подальшого дослідження. У ході виконання роботи успішно реалізовано побудову прототипу системи «Розумний будинок», що підтвердило можливість використання платформи Arduino в поєднанні з датчиками для моніторингу параметрів навколишнього середовища та передачі даних на мобільний додаток через Bluetooth-з'єднання. На основі Arduino IDE створено скетч-програму, яка забезпечує стабільну роботу датчиків і передачу

інформації в структурованому вигляді на мобільний додаток. Подальша розробка включала створення мобільного додатку для операційної системи Android з використанням Android Studio і мови програмування Java. В результаті успішно скомпільовано білд програми, який встановився на телефон і забезпечує коректну взаємодію з платою Arduino, приймаючи дані від датчиків у реальному часі. Порівняння кастомного Bluetooth терміналу, розробленого спеціально для системи «Розумний будинок», зі стандартним терміналом із Google Play показує, що для специфічних проєктів, таких як система «Розумний будинок», кастомний термінал є більш оптимальним вибором завдяки своїй гнучкості, адаптивності та можливості тонкого налаштування під конкретні завдання. Водночас, стандартний термінал може бути корисним для загальних потреб і тих користувачів, які шукають просте рішення без необхідності вносити зміни у функціонал. Перспективи подальшого дослідження включають розширення функціональності системи «Розумний будинок» шляхом інтеграції додаткових датчиків і пристроїв для автоматизації різних аспектів управління середовищем. Також варто дослідити можливості впровадження інших бездротових технологій, таких як Wi-Fi або Zigbee, для розширення радіусу дії та забезпечення більш надійного зв'язку.

Список бібліографічного опису

1. Monk S. Programming Arduino: Getting Started with Sketches, Third Edition, New York : McGraw-Hill, 2023. 176 p.
2. IoT Applications, Security Threats, and Countermeasures / ed. by P. Nayak, N. Ray, P. Ravichandran; Boca Raton : CRC Press, 2021. 277 p.
3. Кіберфізичні системи: багаторівнева організація та проєктування / А. О. Мельник та ін. Київ : Магнолія, 2023. 238 с.
4. Arduino and HC-05 Bluetooth Module Complete Tutorial. *How To Mechatronics*. URL: <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/> (date of access: 06.07.2024).
5. DHT11–Temperature and Humidity Sensor. *Components101*. URL: <https://components101.com/sensors/dht11-temperature-sensor> (date of access: 15.07.2024).
6. SR312 Інфрачервоний Датчик Руху Мій Проєкт. URL: <https://myproject.com.ua/hc-sr312-miniatjurnij-infrachervonij-datchik-ruhu-ua.html> (дата звернення: 23.07.2024).
7. Fazio M. Kotlin and Android Development Featuring Jetpack: Build Better, Safer Android Apps. O'Reilly Media, Incorporated, 2021.

References

1. Monk S. Programming Arduino: Getting Started with Sketches, Third Edition, New York : McGraw-Hill, 2023. 176 p.
2. IoT Applications, Security Threats, and Countermeasures / ed. by P. Nayak, N. Ray, P. Ravichandran; Boca Raton : CRC Press, 2021. 277 p.
3. Кіберфізичні системи: багаторівнева організація та проєктування / А. О. Мельник та ін. Київ : Магнолія, 2023. 238 с.
4. Arduino and HC-05 Bluetooth Module Complete Tutorial. *How To Mechatronics*. URL: <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/> (date of access: 06.07.2024).
5. DHT11–Temperature and Humidity Sensor. *Components101*. URL: <https://components101.com/sensors/dht11-temperature-sensor> (date of access: 15.07.2024).
6. SR312 Інфрачервоний Датчик Руху Мій Проєкт. URL: <https://myproject.com.ua/hc-sr312-miniatjurnij-infrachervonij-datchik-ruhu-ua.html> (дата звернення: 23.07.2024).
7. Fazio M. Kotlin and Android Development Featuring Jetpack: Build Better, Safer Android Apps. O'Reilly Media, Incorporated, 2021.