



Founders: State University of Economics and Technology

ISSN: 3041-1246

E-mail: [ete@duet.edu.ua](mailto:ete@duet.edu.ua) Journal homepage: <https://ete.org.ua>

JEL: C88

DOI: 10.62911/ete.2024.02.01.07


## Development of an Android application for controlling a smart greenhouse on Arduino

Citation:

Medvediev, D., Khotskina, V., Khodukin, M., Medvedieva, S., & Prykhodko, I. (2024). Development of an Android application for controlling a smart greenhouse on Arduino. Scientific and practical journal "Economics and technical engineering". Vol. 2 No. 1 (2024), 79–92. <https://doi.org/10.62911/ete.2024.02.01.07>

Dmytro Medvediev  
*Assoc. Prof., PhD State University of Economics and Technology, Kryvyi Rih, Ukraine*

e-mail: [medvediev\\_dg@kneu.dp.ua](mailto:medvediev_dg@kneu.dp.ua)

 ORCID iD: 0000-0002-3747-1717


Valentyna Khotskina  
*Assoc. Prof., PhD, State University of Economics and Technology, Kryvyi Rih, Ukraine*

e-mail: [hotskina\\_vb@duet.edu.ua](mailto:hotskina_vb@duet.edu.ua)

 ORCID iD: 0000-0001-8963-4189


Mykola Khodukin  
*Assistant., State University of Economics and Technology, Kryvyi Rih, Ukraine*

e-mail: [hodukin@kneu.dp.ua](mailto:hodukin@kneu.dp.ua)

 ORCID iD: 0000-0001-8577-2541

Svitlana Medvedieva  
*Assistant., State University of Economics and Technology, Kryvyi Rih, Ukraine*

e-mail: [karpiuk\\_sa\\_23848@kneu.dp.ua](mailto:karpiuk_sa_23848@kneu.dp.ua)

 ORCID iD: 0009-0005-9297-2935

Illia Prykhodko  
*Student, State University of Economics and Technology, Kryvyi Rih, Ukraine*

e-mail: [liakhova\\_ia@kneu.dp.ua](mailto:liakhova_ia@kneu.dp.ua)

Received: 11/03/2024

Accepted: 08/04/2024



**Abstract:** The relevance of the topic "Development of an Android application for controlling a smart greenhouse on Arduino" is due to the opportunities offered by the field of automated device development, as well as the attention to this field of IT development, both by specialists and users who have certain tasks in production, in business, at home and other areas whose needs can be solved with the help of these technologies. Therefore, the purpose of the research is development of the device, the software for it, and production of a mobile application for controlling the created device. Using the developed device, software and its application, you can create, for example, an automatic greenhouse, which will save time and resources with maximum return. During the study of the automation of the processes of control and management of environmental parameters, the requirements that must be fulfilled during the automation of the receipt and processing of sensor readings were clearly formulated. On the basis of the analysis of the already conducted research works, the main functional components that should be inherent in any automated control system with the possibility of remote control have been selected. In the process of research, the need to use the integrated development environment Arduino IDE and Android Studio is substantiated. Arduino IDE is designed to write, compile, and load programs into Arduino microcontroller memory, and Android Studio is the official Android application development tool based on JetBrains' IntelliJ IDEA software. The process of creating the device is fully described, with a step-by-step review of each element and the sequence of actions. The program code is provided and the principle of operation of all its components is described. The principle of operation of the Android application is also demonstrated. It contains information about the state of the greenhouse (temperature and humidity) and has the ability to turn on cooling both automatically and manually. Information is presented in the form of graphs that change dynamically.

**Keywords:** Arduino; Android; Bluetooth; data transfer.


JEL: C88

### Development of an Android application for controlling a smart greenhouse on Arduino

Dmytro Medvediev

Assoc. Prof., PhD, State University of Economics and Technology, Kryvyi Rih, Ukraine

e-mail: medvediev\_dg@kneu.dp.ua

 ORCID ID: 0000-0002-3747-1717

Valentyna Khotskina

Assoc. Prof., PhD, State University of Economics and Technology, Kryvyi Rih, Ukraine


e-mail: hotskina\_vb@duet.edu.ua

 ORCID ID: 0000-0001-8963-4189

Mykola Khodukin

Assistant, State University of Economics and Technology, Kryvyi Rih, Ukraine


e-mail: hodukin@kneu.dp.ua

 ORCID ID: 0000-0001-8577-2541

Svitlana Medvedieva

Assistant,, State University of Economics and Technology, Kryvyi Rih, Ukraine

e-mail: medvedieva@kneu.dp.ua

 ORCID ID: 0009-0005-9297-2935

Illia Prykhodko

Student, State University of Economics and Technology, Kryvyi Rih, Ukraine

e-mail: raintoucher@gmail.com

**Abstract:** The relevance of the topic "Development of an Android application for controlling a smart greenhouse on Arduino" is due to the opportunities offered by the field of automated device development, as well as the attention to this field of IT development, both by specialists and users who have certain tasks in production, in business, at home and other areas whose needs can be solved with the help of these technologies. Therefore, the purpose of the research is development of the device, the software for it, and production of a mobile application for controlling the created device. Using the developed device, software and its application, you can create, for example, an automatic greenhouse, which will save time and resources with maximum return. During the study of the automation of the processes of control and management of environmental parameters, the requirements that must be fulfilled during the automation of the receipt and processing of sensor readings were clearly formulated. On the basis of the analysis of the already conducted research works, the main functional components that should be inherent in any automated control system with the possibility of remote control have been selected. In the process of research, the need to use the integrated development environment Arduino IDE and Android Studio is substantiated. Arduino IDE is designed to write, compile, and load programs into Arduino microcontroller memory, and Android Studio is the official Android application development tool based on JetBrains' IntelliJ IDEA software. The process of creating the device is fully described, with a step-by-step review of each element and the sequence of actions. The program code is provided and the principle of operation of all its components is described. The principle of operation of the Android application is also demonstrated. It contains information about the state of the greenhouse (temperature and humidity) and has the ability to turn on cooling both automatically and manually. Information is presented in the form of graphs that change dynamically.


**Keywords:** Arduino; Android; Bluetooth; data transfer

## Розробка Android - додатка для керування розумною теплицею на Arduino

Дмитро Медведєв

доцент, к.т.н., Державний університет економіки і технологій, Кривий Ріг, Україна


e-mail: medvediev\_dg@deut.edu.ua

 ORCID ID: 0000-0002-3747-1717

Валентина Хоцкіна

доцент, к.т.н., Державний університет економіки і технологій, Кривий Ріг, Україна


e-mail: hotskina\_vb@duet.edu.ua

 ORCID ID: 0000-0001-8963-4189

Микола Ходукін

асистент, Державний університет економіки і технологій, Кривий Ріг, Україна


e-mail: hodukin@kneu.dp.ua

 ORCID ID: 0000-0001-8577-2541

Світлана Медведєва

асистент, Державний університет економіки і технологій, Кривий Ріг, Україна

e-mail: medvedieva@kneu.dp.ua

 ORCID ID: 0009-0005-9297-2935

Ілля Приходько

студент, Державний університет економіки і технологій, Кривий Ріг, Україна

e-mail: raintoucher@gmail.com

**Анотація:** Актуальність теми «Розробка Android - додатка для керування розумною теплицею на Arduino» зумовлена можливостями, що пропонує сфера розробки автоматизованих пристроїв, а також увагою до цієї галузі ІТ - розробок, як фахівців, так і користувачів, що мають певні завдання на виробництві, в бізнесі, побуті та інших сферах, чії потреби можна вирішити за допомогою даних технологій. Тому метою дослідження є розробка пристрою, програмного забезпечення до нього та розробка мобільного додатку для керування створеним пристроєм. Користуючись розробленим пристроєм, програмним забезпеченням та додатком до нього, можна створити, наприклад, автоматичну теплицю, що дозволить заощадити час та ресурси з максимальною віддачею. При дослідженні автоматизації процесів контролю та керуванні параметрами середовища були чітко сформульовані вимоги, що повинні виконуватись при автоматизації надходження та обробки показів сенсорів. На основі аналізу вже проведених дослідних робіт виділено основні функціональні компоненти, які повинні бути притаманні будь-якій системі автоматизованого контролю з можливістю дистанційного керування. У процесі дослідження обґрунтовано необхідність у використанні інтегрованого середовища розробки Arduino IDE та Android Studio. Arduino IDE призначено для написання, компіляції та завантаження програм в пам'ять мікроконтролера плат Arduino, а Android Studio – це офіційний інструмент розробки додатків для Android, заснований на програмному забезпеченні IntelliJ IDEA від JetBrains. Повністю описаний процес створення пристрою, з поетапним розглядом кожного елемента та послідовністю дій. Надано програмний код та описаний принцип роботи усіх його компонентів. Також продемонстровано принцип роботи Android - додатка.

**Ключові слова:** Arduino; Android; Bluetooth; передача даних

## Вступ (Introduction)

Розумна теплиця – це теплиця, що має мікроконтролери, сенсори та спеціальний програмний додаток. Також розумну теплицю часто синхронізують і з іншими технічними рішеннями, наприклад, автоматичними системами поливу. Мікродатчики можуть контролювати величезний масив факторів: температуру, вологість та якість ґрунту, освітлення. Додатково з їх допомогою фіксують дані про зростання рослин, кількість пестицидів, використання добрив і т.д. Зібрану інформацію можна надіслати на локальний або хмарний сервер. Консоль веб адміністратора дозволяє налагоджувати параметри системи та синхронізувати її з іншими рішеннями. А додаток на мобільному пристрої збирає звіти про стан культур та умови у теплиці, генерує попередження, якщо якийсь показник не відповідає нормі. Сучасні технології дозволяють створити розумну теплицю з бездротовою передачею даних, що дозволить заощадити на дротовому з'єднанні. «Розумне сільське господарство» ставить перед собою мету максимально автоматизувати сільськогосподарську діяльність, підвищити урожайність і якість продукції (Samoilenko, 2020).

І хоча вирощування різних культур в тепличних комплексах є одним із найбільш енерговитратних процесів у сільському господарстві, але завдяки більш точному контролю умов вирощування рослин, теплиці дозволяють забезпечити більш високу врожайність на квадратний метр та більш якісну продукцію.

Алгоритми функціонування таких комплексів заснований на відхиленні або збуренні регулюючої величини (температура, вологість, освітлення і т.д.). Для регулювання температурного режиму електричних нагрівачів застосовуються в основному системи автоматичної стабілізації в яких регулятор підтримує постійне значення регульованого параметру та системи програмного регулювання, які змінюють параметр по попередньо заданому закону в часі (Baranov, & Bielikov, 2021).

Основним обмеженням при використанні датчиків є складність їх характеристик та калібрування. Це обумовлено гіпереластичними характеристиками матеріалів, що викликають нелінійність, великий гістерезис, повзучість і дрейф, що призводить до генерування несподіваної фізичної поведінки та електричних реакцій датчиків. Ці недоліки ускладнюють використання датчиків. При програмуванні контролера важливо уявляти, що зміна процесу перебуватиме біля межі спрацьовування, неминуче відбуватиметься явище коротких переходів туди й назад за граничну лінію ("брязкот"). Для позбавлення системи від "брязкоти" при зближенні з лінією кордону використовується "Зона нечутливості" або "Гістерезис" (від грец. hysteresis – відставання, запізнення). Тобто при перетині відповідного порога порушення стабільного стану системи (спрацьовування) виникає без затримки, а зникає лише при перетині зони Гістерезису. Інакше кажучи, за одних і тих же зовнішніх умов стан об'єкту дослідження може бути різним, залежним від його стану в попередні моменти часу (Hadetska, Dubnytskyi, Kushneruk, Filatova, & Khodyriev, 2021).

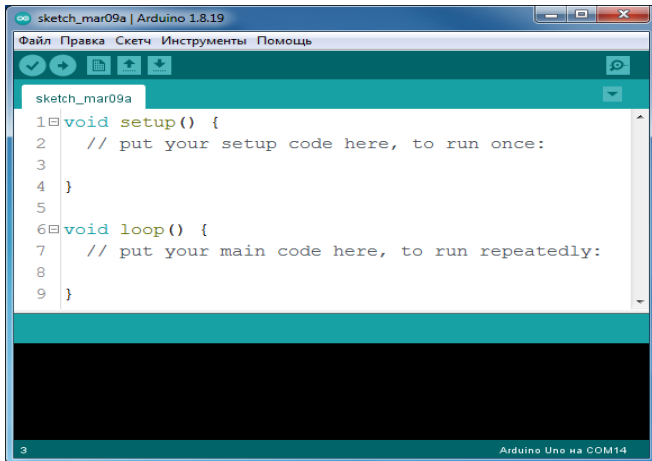
Одним з великих недоліків є неправильне налаштування системи, через яку можна втратити весь урожай. Тому до автоматизації потрібно підходити дуже відповідально (Vilous, 2017). У будь-який момент часу користувач повинен мати можливість перевести управління процесом в ручному режимі, тобто взяти управління процесами на себе. Ручне керування датчиками, незважаючи на велику трудомісткість у порівнянні з повністю автономним керуванням, виявляється іноді більш ефективнішим за рахунок того, що система управління не завжди здійснює правильні рішення.

## Матеріали та методи (Materials and Methods)

Створення проєкту потребує використання середовищ розробки Android studio та Arduino IDE, а також плати Arduino з елементами: датчик температури та вологості DHT11, дисплей LCD1602, Bluetooth модуль HC-06. Побутова техніка - нагрівач та вентилятор будуть підключатися через одноканальні модулі реле (RELE-1x5V).

## Результати (Results)

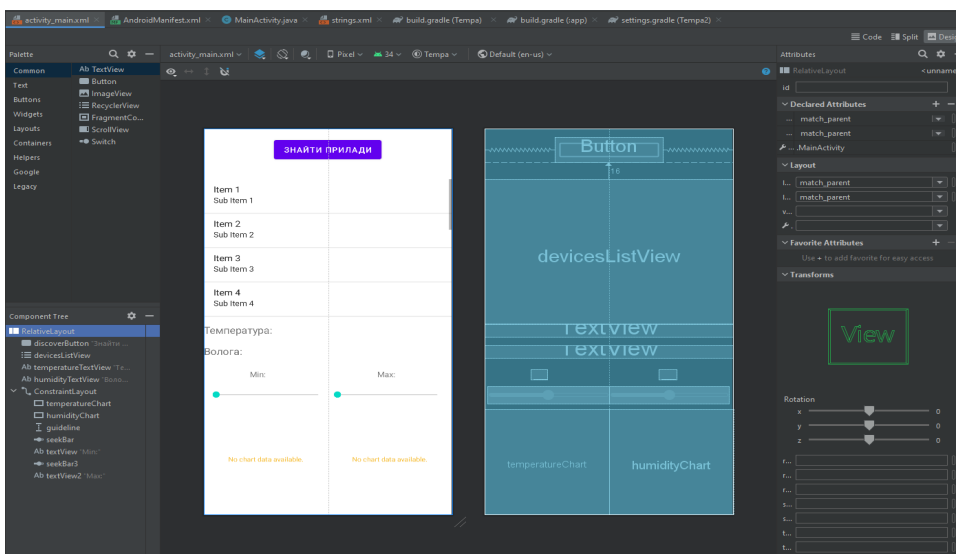
**Інтегроване середовище розробки Arduino IDE** (Рисунок 1) призначено для написання, компіляції та завантаження власних програм в пам'ять мікроконтролера, що встановлений на платі Arduino-сумісного пристрою. Основою середовища розробки є мова Processing/Wiring – це, фактично, звичайний C++, доповнений простими і зрозумілими функціями для керування введенням/виведенням на контактах (Kryvonos, O.M., Kuzmenko, Y.V., & Kuzmenko, S.V. 2016). Існують версії середовища для операційних систем Windows, Mac OS та Linux, а останню версію середовища Arduino IDE завжди можна завантажити зі сторінки офіційного сайту.



**Рисунок 1.** Вікно Arduino IDE

Можливості середовища програмування Arduino може бути значно розширено за допомогою використання бібліотек. Бібліотеки розширюють функціональність програм через реалізацію деяких додаткових функцій, наприклад, роботи з апаратними засобами, функції з обробці даних тощо. Ряд бібліотек встановлюється автоматично разом із середовищем розробки, також можна завантажувати або створювати власні бібліотеки.

**Android Studio** – безкоштовне середовище розробки на основі IntelliJ IDEA (international data encryption algorithm), що надають інтегровані інструменти для розробки та налагодження додатків для платформи Android (Рисунок 2).



**Рисунок 2.** Інтерфейс розробки середовища програмування

Android Studio було представлено 16 травня 2013 року на конференції Google I/O, а після релізу прийшло на зміну плагіну ADT для платформи Eclipse. Android Studio містить: Android SDK (software development kit), інструменти для розробки дизайну, тестування і налагодження. В основу робочого процесу Android Studio закладений концепт безперервної інтеграції, що дозволяє відразу виявити наявні проблем (Krasnorutskyi, Zhuk, Kazmirov, & Martyniuk, 2022). Для розробки програмного забезпечення використовуються мови Kotlin та Java. Мова Java набуває популярності та часто використовується при розробці застосунків для Windows Mobile, Android, Bada, Palm OS і BlackBerry OS (Ichanska, & Ulko, 2020).

Розробка візуальної складової програми відбувається в редакторі файлу формату ".xml", що є файлом розмітки для виведення інформації на екран смартфона. У даному редакторі є можливість перемикання між "Режимами" розробки: Code, Split, Design. "Code" є текстовим редактором даного файлу, в якому позначені розміщені на конструкторі об'єкти (їхні властивості, координати і т.п.). "Design" є візуальним редактором, де розміщення та зміна об'єктів відбувається "на око", тим самим представляючи те, яким у майбутньому буде інтерфейс користувача. "Split" - це комбінований режим, що розділяє екран навпіл, одну частину якого займає режим "Code", а іншу - "Design", надаючи функціонал обом.

**Компоненти апаратної частини.** Передача значень з датчика (температури та вологості) безпосередньо в Андроїд-додаток виконується через Bluetooth-модуль (Medvediev, Khotskina, Khodukin, Dashko, & Kravchuk, 2023)

### LCD1602

Символьні дисплеї використовуються в системах домашньої безпеки, робототехніці, багатофункціональних пристроях, торгових автоматах і турнікетах громадського транспорту.

Символьний дисплей LCD1602 (Рисунок 3) використовується для відображення інформації від контролера та сенсорів у текстовому форматі. Дисплей складається з матриці рідкокристалічного дисплея (LCD) і відображає два рядки по 16 символів білого кольору на синьому підсвічуванні. Матриця дисплея підключена до контролера Hitachi HD44780, здатного відображати ряд цифр, латинських символів і прошиті друковані знаки. У пам'яті дисплея можна створити до восьми спеціальних символів. Кожен символ складається з квадрата розміром 5 x 8 пікселів. При розмірі модуля 80x36x15 мм розмір видимої області складає 64.5 x 14.5 мм. Напруга живлення модуля 5 вольт, а робоча температура 0 ~ 50°C. Працює зі стандартною бібліотекою *LiquidCrystal*.

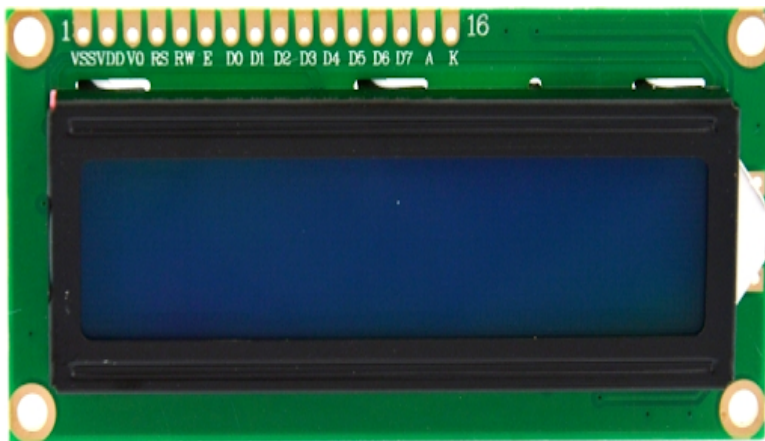


Рисунок 3. Дисплей LCD1602

**Модуль реле.** Модуль реле одноканальний – 5V, 10A, 250V (Рисунок 4) – це одноканальний модуль реле на 5В, що керується за допомогою контролера Arduino. Цей релейний модуль призначений для керування як постійним, так і змінним струмом великої сили. Окрім самого реле, модуль також містить оптоелектронне рішення з транзистором, який захищає клеми Arduino від стрибків напруги на котушці.



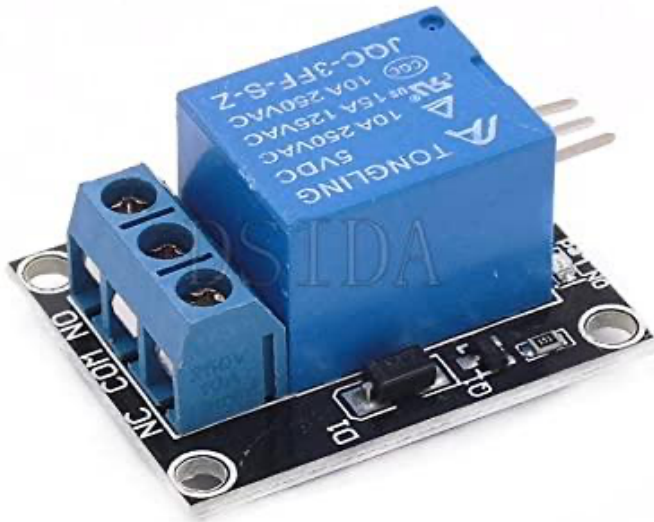


Рисунок 4. Модуль реле

Характеристики:

- Напруга живлення: 5 В
- Струм споживання: 15 мА ... 20 мА
- Сигнал вмикання: 0 В (низький рівень)
- Кількість реле: 1 шт.
- Тип реле: електромеханічний
- Номінальний струм навантаження: 10 А
- Комутована напруга: 250VAC, 30VDC

Розміри: 43 x 20 x 18 мм

Робоча напруга в 5 вольт дозволяє вільно підключати модуль до Arduino, а керований перемикач, що за сигналом Arduino перемикає середній контакт між двома крайніми розмикає або замикає ланцюг. Для підключення до Arduino використовуються 3 контакти: два контакти живлення (5В та Gnd) та контакт управління, який підключається до цифрового виводу на платі. На самому реле з іншої сторони є ще три контакти, але для підключення навантаження використовуються два з них, а інший залишається вільним (всередині реле пов'язаний із заземленням). Тому у разі увімкненого реле відбувається замикання контактів COM (загальний) та NC (нормально замкнений) і споживач вмикається, а вимкненого реле замикаються інші контакти – COM (загальний) та NO (нормально розімкнений).

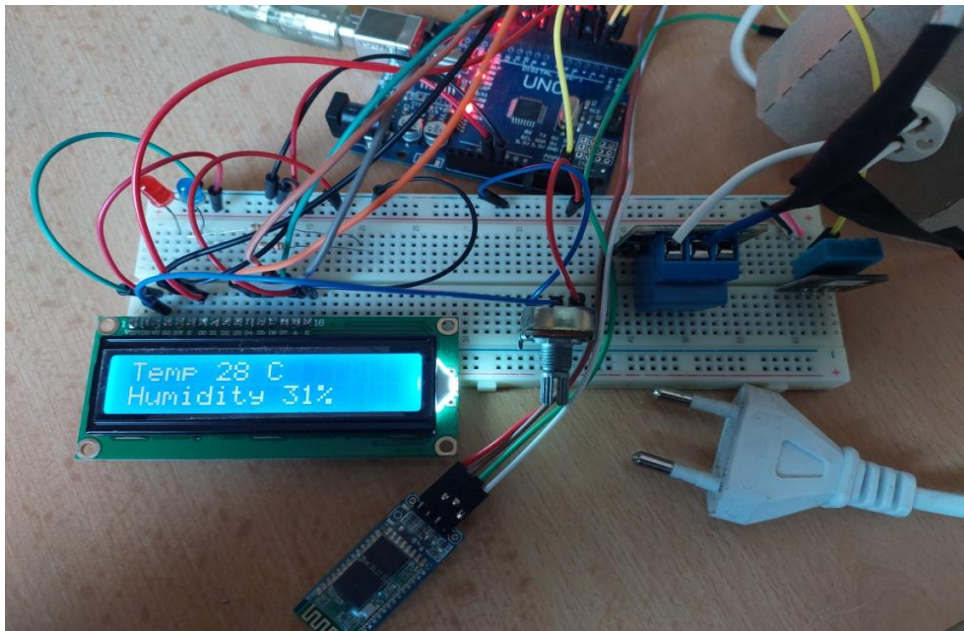
До мінусів даного типу реле можна віднести велике споживання струму та малу живучість у разі великих навантажень.

**Реалізація частини проекту на Arduino.** Вмикання активує процес ініціалізації. Після її завершення – датчик температури та вологості (DHT11) збирає дані у навколишньому середовищі та відправляє їх до плати Arduino UNO, де відбувається обробка отриманої інформації та подальше відправлення її через Bluetooth модуль на Android додаток. Також приймається рішення, щодо встановлення потрібних значень сигналів на піни, що керують робочими елементами (нагрівач та кулер). Відповідно до отриманих сигналів робочі елементи перемикаються між їх робочим та неробочим станом. Ці вказівки можуть бути відредаговані користувачем через Android додаток та після чого відправлені через Bluetooth до плати Arduino UNO. Поточні покази температури та вологості також відображаються на LCD дисплеї (Рисунок 5).

Для релейного управління найпростіший алгоритм передбачає лише два стана («увімкнено» та «вимкнено»):

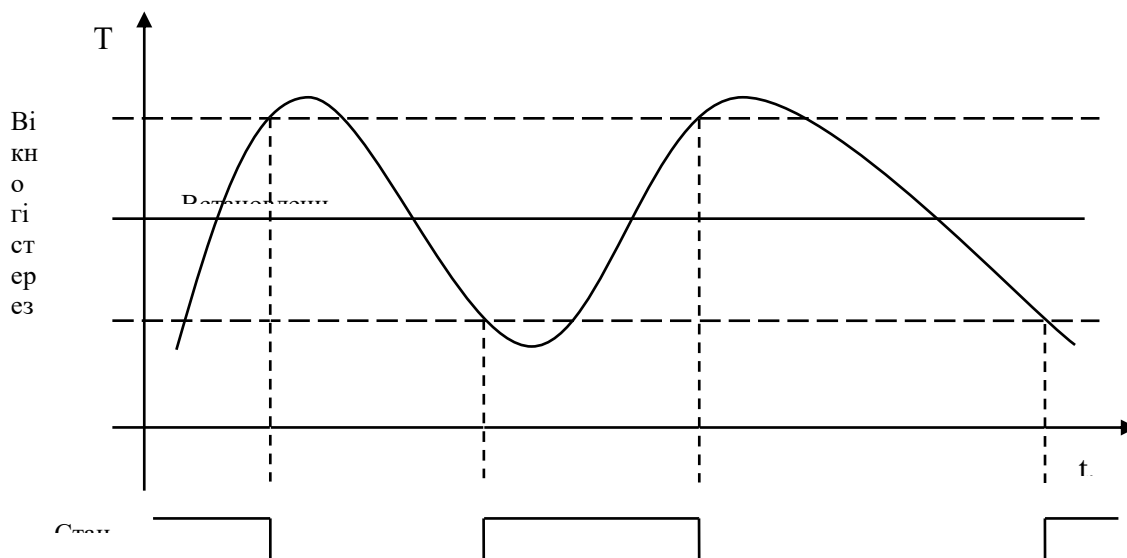
```
if (temp < 24.0) digitalWrite(Pin_rele, 1);
else digitalWrite(Pin_rele, 0);
```

Цей код просто включає реле, коли умовна температура нижче 24 градусів, і вимикає, коли вона вища. Якщо викликати цей код без затримки або таймера, то можна отримати ефект «брязкоту» в момент увімкнення та вимкнення реле, оскільки шуми вимірювань постійно змінюватимуть результат умови. Тобто значення треба буде фільтрувати, щоб збільшити стабільність системи.



**Рисунок 5.** Апаратна частина пристрою у зборі

Найпростішим способом створення нормального релейного регулятора є період роботи регулятора. Його можна реалізувати як затримкою, так і таймером. Інший спосіб – гістерезис (Рисунок 6), що дозволяє ще більше зменшити кількість перемикань реле і навіть позбутися опитування по таймеру, що підвищує реакцію системи на зміни, зберігши при цьому хорошу стійкість до перешкод.



**Рисунок 6.** Встановлення реле

Гістерезис поділяє «встановлення» на два (трохи менше і трохи більше), що утворюють розмір вікна гістерезису, або значення гістерезису ( $\Delta$ ). Зазвичай значення гістерезису



температури встановлюється в діапазоні від плюс 3 °С до плюс 5 °С. (Yevseienko, & Kachanov, 2022). Логіка роботи полягає в тому, щоб утворити область, всередині якої система рухається за інерцією від останнього перемикання і переходить в новий стан тільки при виході з цієї області (включати реле на нагрівання нижче нижньої лінії, і вимикати тільки вище верхньої). Але слід врахувати і те, що додавання гістерезису зменшує не тільки кількість перемикань реле, а й точність. У коді гістерезис можна реалізувати, наприклад, таким чином:

```
#define PIN_rele 2
float controlPoint = 24.0; // встановлення
float hyster = 2; // ширина вікна гістерезису
// .....
static bool releState = false;
if (temp < (controlPoint - hyster)) releState = true;
else if (temp > (controlPoint + hyster)) releState = false;
digitalWrite(PIN_rele, releState);
```

**Реалізація візуальної складової на Android.** Розглянемо розділ з елементами розташованими на екрані – “*Component Tree*” (Рисунок 2). Можна бачити перелік наступних компонентів:

1. *RelativeLayout* – це простір для майбутнього головного меню, макет, в якому можна порядкувати представлення/віджети у відповідності з положенням інших представ/віджетів. Вона не залежить від горизонтального та вертикального представлення, і це можна упорядкувати за поглядом розробника.

2. *Button* – це інтерфейсний компонент, кнопка, який надає користувачеві можливість виконати певну дію, коли він взаємодіє з ним методом натискання, що призводить до запуску закладеної функції.

*DiscoverButton* – цей елемент керування являє собою кнопку, що виконує функцію оновлення списку існуючих пристроїв для підключення по *Bluetooth* поблизу компонент *DeviceListView*.

3. *ListView* – це частина інтерфейсу, яка дозволяє відображати перелік елементів на екрані пристрою. Ці елементи можуть бути текстовими рядками, зображеннями та іншими видами контенту. Користувач може прокручувати цей список вгору та вниз, щоб переглянути всі елементи. Цей компонент дозволяє створювати динамічні списки, які можуть бути заповнені даними з масиву чи бази даних. Крім того, ви можете налаштувати зовнішній вигляд кожного елемента списку, щоб зробити його зручним та привабливим для користувача.

*DeviceListView* – цей компонент є динамічним списком, що надає інформацію про пристрої *Bluetooth*, до яких можна з'єднатися. Список оновлюється за допомогою *DiscoverButton*.

4. *TextView* – це компонент інтерфейсу, який відображає текст на екрані пристрою. Він розміщується на просторі та відіграє роль текстового покажчика.

*TemperatureTextView* – відображає температуру у поточному часі.

*HumidityTextView* – відображає вологість у поточному часі.

*TextView1* – відіграє роль вказівника мінімальної температури.

*TextView2* – відіграє роль вказівника максимальної температури.

5. *ConstraintLayout* – це спосіб організації елементів інтерфейсу у додатку, який дозволяє створювати гнучкі та адаптивні макети. Він використовує обмеження (*constraints*) для визначення положення та розміру кожного елемента щодо інших елементів на екрані. Це дозволяє легко створювати макети, які можуть адаптуватися до різних розмірів екранів та орієнтацій пристроїв. Цей компонент забезпечує гнучкість і зручність у проектуванні інтерфейсів для додатків. У разі він грає роль масштабування графіків.

6. *GuideLine* – це елемент макету, який використовується в *ConstraintLayout* для керування розташуванням інших елементів. Він є невидимою лінією, яку можна розмістити на екрані і прив'язати до певного положення або елемента. Виконує роль створення межі між компонентами.

7. *MPAndroidChart* - це частина інтерфейсу користувача, яка використовується для вбудовування графіків і діаграм, створених за допомогою бібліотеки *MPAndroidChart*. Цей компонент дозволяє відображати графіки та діаграми прямо на екрані пристрою, забезпечуючи візуалізацію даних для користувача.

*TemperatureChart* – відображає графік температури у поточному часі.

*HumidityChart* – відображає графік вологості у поточному часі.

8. *SeekBar* – це елемент інтерфейсу користувача, який являє собою повзунок, який користувач може перетягувати вздовж горизонтальної осі. *SeekBar* зазвичай використовується для вибору числових значень у заданому діапазоні, таких як гучність, яскравість, час і т. д. Користувач може перетягувати повзунок вліво або вправо для зміни значення, яке можна використовувати в додатку.

*SeekBar* – встановлює мінімальне значення температури.

*SeekBar2* – встановлює максимальне значення температури.

При створенні програми були також використані сторонні бібліотеки, які не є стандартною версією середовища програмування. Йдеться про *MPAndroidChart* – це бібліотека, яка дозволяє створювати красиві та інтерактивні графіки та діаграми у додатках. Вона надає різні типи графіків, такі як лінійні, кругові, стовпчасті та інші, а також можливості налаштування їхнього зовнішнього вигляду, кольорів, масштабування та інтерактивності. Завдяки цій бібліотеці можна візуалізувати дані програми так, щоб користувачі могли легко зрозуміти та аналізувати інформацію.

Крім можливостей, присутніх IntelliJ IDEA, в Android Studio реалізовано кілька функцій, таких як нова уніфікована підсистема складання, тестування і розгортання застосунків, заснована на інструментарії Gradle і підтримуюча використання засобів без перервної інтеграції (Kozub H, Kozub Yu., Mohylnyi, & Zhukov, 2021).

Взаємодія з бібліотеками відбувається через систему *Gradle* - це інструмент автоматизації збирання та управління залежностями у проєктах. Він дозволяє керувати процесом складання проєкту, включаючи компіляцію, тестування, пакування та розгортання. *Gradle* використовує декларативний DSL (Domain-Specific Language) для опису складальних скриптів. Він також забезпечує зручне управління проєктами та їх залежностями. Коли додається стороння бібліотека до проєкту через *Gradle*, потрібно вказати її ім'я та версію у файлі сценарію складання проєкту, який називається "build.gradle". *Gradle* потім завантажує цю бібліотеку з центрального сховища, що називається репозиторієм, і додає її до вашого проєкту. Це відбувається автоматично, коли запускається процес складання проєкту або синхронізації *Gradle* у середовищі розробки. Після цього можна використовувати функціонал цієї бібліотеки у проєкті, викликаючи її методи та класи у коді.

**Алгоритм роботи Android додатку.** Переходячи до розробки додатку потрібно звернути увагу на його алгоритм розробки, послідовність в якій будуть додаватися елементи та встановлення їх взаємодії між собою. Першим елементом буде *RelativeLayout* – фундамент, що представляє собою простір для майбутнього розміщення компонентів. Перед тим, як розміщувати елементи для майбутньої програми, потрібно написати код, що включає надання дозволу додатку для взаємодії з *Bluetooth*, це викликано тим, що з метою безпеки дані дозволи не надаються за замовчуванням, що вимагає створення певного класу, в якому будуть прописані дані дозволу, що включають такі процеси взаємодії як перевірку на присутність *Bluetooth* на смартфоні та його статус, пошук девайсів, підключення, обробка, отримання і відправлення даних. Після чого вже посилатися на даний клас при написанні коду взаємодії елементів, пов'язаних з необхідністю використовувати дані функції.

Так само, перед роботою програми потрібно написати код, який буде проводити перевірку *Bluetooth* на його статус (on/off), і якщо на пристрої він вимкнений, то на екрані користувача з'явилось повідомлення з пропозицією включити *Bluetooth*, якраз тут вже знадобився клас дозволу згаданий вище.

Далі вже йде розміщення об'єктів у просторі. Першими компонентами в цьому будуть *DeviceListView*, який надаватиме користувачеві доступні поблизу пристрою *Bluetooth*,

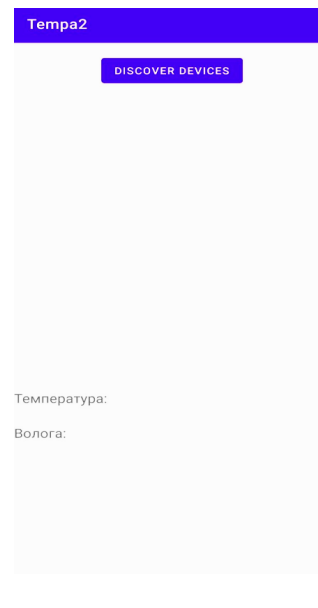
показуючи, зручний для взаємодії список, а за ним - компонент *DiscoverButton*, кнопку, яка оновлюватиме цей список.

Наступними компонентами будуть виступати *TemperatureTextView* та *HumidityTextView*, які, отримуючи дані *Bluetooth* з пристрою автоматичної теплиці, відобразять інформацію про температуру та вологість на поточний момент.

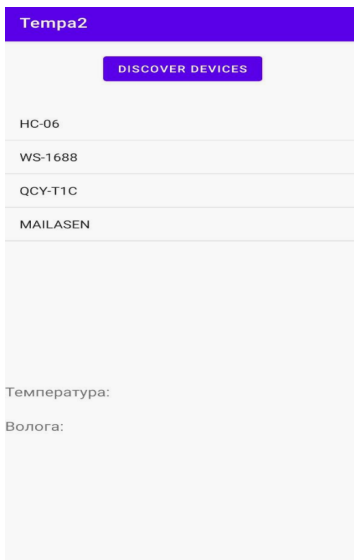
Після цього йде розміщення *ConstraintLayout*, блоку, всередині яких будуть розташовані графіки зміни параметрів з плином часу, а саме *TemperatureChart* – для температури і *HumidityChart* – для вологості, а також *GuideLine* для запобігання накладенню один на одного двох попередніх компонентів.



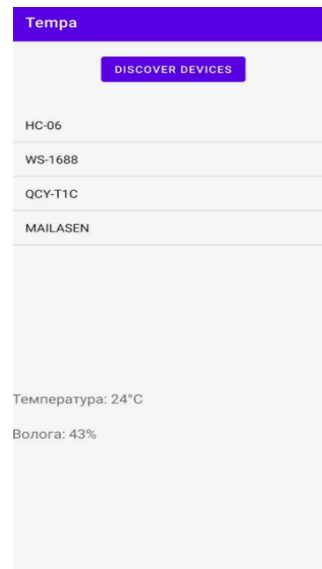
а)



б)



в)



г)

**Рисунок 7.** Етапи роботи з Android додатком

Останніми компонентами будуть *SeekBar*, один з яких відповідає за задану мінімальну температуру, а інший – за максимальну, який із повзунків за що відповідає буде вказувати компоненти *TextView*, що над ними (*Min* і *Max*).

При відкритті програми, якщо на пристрої не запущено *Bluetooth*, з'явиться повідомлення з пропозицією увімкнути цю функцію на смартфоні (Рисунок 7а). Після згоди – перед

користувачем з'являється головне меню програми, де розташований один компонент – це кнопка оновлення пошуку пристроїв (Рисунок 7б). Після натискання на неї з'являється список з наявними поблизу пристроїв з увімкненим *Bluetooth* адаптером (Рисунок 7в). При виборі потрібного, а саме теплиці, за натисканням відбудеться підключення до пристрою, що супроводжується сірим підсвічуванням вибраного пристрою в процесі з'єднання. Після успішного встановлення з'єднання, на екрані смартфона через компонент *TextView* відображатимуться дані про температуру і вологість (Рисунок 7г).

Динаміку зміни температури та вологості всередині приміщення в поточному часі демонструють графіки в нижній частині екрана. (Рисунок 8).



**Рисунок 8.** Графік зміни температури та вологості

## Висновки (Conclusions)

Результатом виконання роботи є працездатний пристрій на базі плати Arduino Uno, програмне забезпечення до нього та мобільний Android - додаток. Ці компоненти взаємодіють між собою: здійснюють обмін даними та обробляють їх для подальшого відправлення певних команд робочим елементам в автоматичному режимі.

Для реалізації задачі та створення взаємодії між платформами було використано Bluetooth - зв'язок за допомогою модуля HC-06 та інтегрованого модуля у смартфоні користувача. Також для створення самого пристрою знадобилися плата Arduino Uno, датчик

вологи та температури DHT11 для збирання даних, LCD дисплей для виведення інформації, вентилятор для охолодження, та нагрівач.

Було створено пристрій, який можна інтегрувати у приміщення невеликого об'єму, що дозволяє регулювати внутрішню температуру та вологість за потреби користувача.

Також одним з результатів стала розробка мобільного додатку на платформі Android за допомогою Android studio. Додаток дозволяє виводити на екрані мобільного пристрою користувача інформацію про поточні покази сенсорів, та змінювати параметри за потреби в ручному режимі.

Для розробки комплексу були обрані платформи Arduino IDE та Android Studio. У першому створюється програмне забезпечення до плати Arduino, а в другому – додаток Android. Детально описаний алгоритм розробки, дії приладу на базі Arduino та мобільного Android - додатку. Принцип взаємодії обох платформ подано в опису таблиці, що демонструє основи, які дозволяють працювати як приладу, так і додатку завдяки Bluetooth - модулю.

Продемонстровано принцип роботи Android - додатка, його вигляд: інтерфейс та його компоненти.

Таким чином, система за типом - розумна теплиця. підтримує оптимальні параметри в приміщенні за можливість в будь-якій час дистанційно відстежувати покази сенсорів та коригувати граничні рівні температури та вологості.

### **Конфлікт інтересів (Conflicts of interest)**

Автори заявляють про відсутність конфлікту інтересів.

### **Фінансування (Funding)**

Це дослідження не отримало зовнішнього фінансування.

### **Внесок авторів (Authors contribution)**

Концептуалізація, М.Д.; методика, Х.М., М.С., та П.І.; програмне забезпечення П.І.; перевірка, М.С. та М.Д.; аналітичні дані, М.С. та П.І.; візуалізація, П.І.; нагляд, Х.В.; адміністрація проекту, М.Д.. Усі автори прочитали та погодились з опублікованою версією рукопису.

### **Література (References)**

Arduino <https://www.arduino.cc/>

Baranov, V.V & Bielikov, K.O. (2021). Porivnialnyi analiz system avtomatychnoho rehulivannia temperatury. *Innovatsii molodi v mashynobuduvanni*. 253-258. <https://doi.org/10.20535/2708-3926.2021.3.231646>

Bilous, O. (2017). Avtomatyzovani systemy upravlinnia klimatom teplytsi. *Ukrainian Journal of Educational Studies and Information Technology*, 5(1), 40–43. <https://doi.org/10.32919/10.32919/uesit.2017.01.40-43>

Bluetooth Tutorial Tutorial-Reports.com. <http://www.tutorial-reports.com/wireless/bluetooth/tutorial.php>

Bluetooth® Technology Website. <https://www.bluetooth.com/>

Different Types of Wireless Communication Technologies. <https://www.watelectronics.com/different-types-of-wireless-communication-technologies/>

Hadetska, S., Dubnytskyi, V., Kushneruk, Yu., Filatova, L., & Khodyriev, O. (2021). *Heometrychni kharakterystyky s-podibnykh (lohistychnykh) kryvykh, shcho zastosovuiutsia pry modeliuvanni yavlyshcha histerezysu. Systemy obrobky informatsii*, (2(165), 14-27. <https://doi.org/10.30748/soi.2021.165.02>



- HC 06 Bluetooth module pinout, features & datasheet. <https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet>
- Ichanska, N., & Ulko, S. (2020). Main aspects of creating mobile applications and selecting the instruments for their development. *Control, Navigation and Communication Systems. Academic Journal*, 1(59), 74-78. <https://doi.org/10.26906/SUNZ.2020.1.074>
- Kozub H. O., Kozub Yu. H., Mohylnyi H. A., & Zhukov A. V. (2021). Clean architecture mobile android supplement with strained principles. *Visnik of the Volodymyr Dahl East Ukrainian National University No. 5 (269)*, 5-10. <https://doi.org/10.33216/1998-7927-2021-269-5-5-10>
- Krasnorutskyi, A., Zhuk, V., Kazmirov, I., & Martyniuk, A. (2022). Development of a mobile app for troubleshooting and troubleshooting of onboard VHF command radio stations. *Systems of Arms and Military Equipment*, 1 (69), 17-25. <https://doi.org/10.30748/soivt.2022.69.02>
- Kryvonos, O. M., Kuzmenko, Y. V., & Kuzmenko, S. V. (2016). Survey and prospects of arduino nano 3.0 platform use in high school. *Information Technologies and Learning Tools*, 56(6), 77-87. <https://doi.org/10.33407/itlt.v56i6.1506>
- Lubko, D., & Lytvyn, Y. (2017). Methodology of Arduino designing as a Web-client and Web-server with the use of the DHT11 sensor and their comparative characteristics, *Bulletin of the National Technical University «KhPI» Series: New Solutions in Modern Technologies*, (32(1254)), 62–67. <https://doi.org/10.20998/2413-4295.2017.32.10>
- Medvediev, D., Khotskina, V., Khodukin, M., Dashko, A., & Kravchuk, M. (2023). Development of an Android application for controlling the device on Arduino. (2023). *Scientific and practical journal «Economics and Technical Engineering»*, 1(2), 98-114. <https://ete.org.ua/index.php/journal/article/view/91>
- Samoilenko M.Yu. (2020). Principles of application of the internet of things technology in the modern world of technical devices. «*Scientific Notes of Taurida V.I. Vernadsky University*», series «*Technical Sciences*» Volume 31 (70). № 6, Part 1, 2020 142-148. <https://doi.org/10.32838/TNU-2663-5941/2020.6-1/24>
- Yevseienko, O.M., & Kachanov P.O. (2022). Shopping mall premises scada-microclimate control system development. *Herald of Khmelnytskyi National University. Technical sciences*, 3 (309), 168-176. <https://www.doi.org/10.31891/2307-5732-2022-309-3>