

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРЕНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ.СТЕПАНА ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**на тему: «Автоматизація процесу поливу рослин із
використанням фотоелектричних панелей»**

за спеціальністю 174 «Автоматизація, комп'ютерно-
інтегровані технології та робототехніка»

Виконавець роботи: студент групи АКТ-42сп

Гречин Андрій Романович

Науковий керівник
доц. Боярчук О.В.
(науковий ступінь, вчене звання, прізвище
та ініціали)

Дубляни-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВЕТЕРЕНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ
ІМ.С.З.ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

перший (бакалаврський) рівень вищої освіти
ОС «Бакалавр» за спеціальністю – 174– «Автоматизація, комп'ютерно-інтегровані
технології та робототехніка»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 2025р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Гречин Андрій Романович

Тема роботи «Автоматизація процесу поливу рослин із використанням
фотоелектричних панелей»

Керівник роботи

к.т.н., доц., Боярчук О.В.

затверджені наказом по університету 172 /к-с від 25. 02. 2025

- 1 Строк подання студентом роботи** 16.06. 2025 р.
- 2 Вхідні дані до роботи** Власні зібрані транзакції, реальні або анонімізовані дані з фінтех-платформ, публічні датасети.
- 3 Перелік графічного матеріалу** (з точним зазначенням обов'язкових креслень) теоретичні основи та обґрунтування проєктного рішення, розробка
принципових та електричних вузлів та блоків пристрою, розробка друкованої
плати та програмного забезпечення, тестування та налагодження пристрою,
охорона праці та безпека в надзвичайних ситуаціях, висновки, додаток а лістинг
програмного продукту

5. Перелік презентаційного матеріалу (з зазначенням обов'язкових елементів): _____

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4, 6	Боярчук О.В., доцент кафедри інформаційних технологій		
5	Городецький І.М., доцент кафедри управління проектами та безпеки виробництва		

7. Дата видачі завдання 25.02.2025.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання роботи	Примітка
1	Написання першого розділу та означення головних завдань роботи	25.02.2025 – 01.04.2025	
2	Виконання другого розділу та формування головних показників для розрахунків	01.04.2025 – 01.05.2025	
3.	Виконання третього розділу та формування початкових даних	01.04.2025 – 01.05.2025	
4.	Виконання четвертого розділу та узагальнення отриманих результатів магістерської роботи	01.04.2025 – 01.05.2025	
5.	Написання розділу: «Охорона праці»	01.05.2025 – 01.06.2024	
6.	Завершення роботи в цілому	01.06.2025 – 09.06.2025	

Студент _____ Гречин А.Р.
(підпис)

Керівник роботи _____ Боярчук О.В.
(підпис)

УДК 631.674:621.311.243:004

Дипломна робота: 95 с., 25 рис., 7 табл., 2 дод., 14 джерел.

Метою роботи є автоматизація ефективної, енергоощадної системи поливу рослин, яка функціонує автономно завдяки використанню сонячної енергії та реалізує алгоритмічне управління подачею води залежно від стану ґрунту та зовнішніх умов.

Автоматизація поливу, розумне зрошення, фотоелектричні панелі, мікроконтролери, сенсори вологості, Arduino,

Об'єктом дослідження є системи автоматизованого управління процесами поливу в аграрному секторі.

Предмет дослідження охоплює технологічні рішення, що поєднують фотоелектричне живлення, сенсорні модулі та мікроконтролери для оптимізації водоспоживання.

Актуальність теми зумовлена потребою в підвищенні ефективності використання водних ресурсів у сільському господарстві, особливо в умовах кліматичних змін і обмеженого доступу до електромереж.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1	
ТЕОРЕТИЧНІ ОСНОВИ ТА ОБГРУНТУВАННЯ	
ПРОЕКТНОГО РІШЕННЯ	7
1.1 Аналіз існуючих аналогів.....	7
1.2 Постановка задач для проектного пристрою.....	8
1.3. Аналіз технологій розробки та вибір засобів	
реалізації проекту.....	10
РОЗДІЛ 2	
РОЗРОБКА ПРИНЦИПОВИХ ТА ЕЛЕКТРИЧНИХ	
ВУЗЛІВ ТА БЛОКІВ ПРИСТРО.....	13
2.1. Розробка алгоритма та структурної схеми	13
2.2. Дослідження та аналіз сфери розробки	16
2.3. Проектування функціональної та принципової схеми	19
РОЗДІЛ 3	
РОЗРОБКА ДРУКОВАНОЇ ПЛАТИ ТА ПРОГРАМНОГО	
ЗАБЕЗПЕЧЕННЯ	28
3.1. Дослідження лазерно-утюжної технології	28
3.2. Програмування Ардуіно	32
3.3. Виявлення синтаксичних помилок у середовищі Arduino IDE	34
РОЗДІЛ 4	
ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ ПРИСТРОЮ.....	36
4.1. Огляд функціональності Debugger в Arduino IDE 2.....	36
4.2. Підключення апаратних засобів налагодження	37
4.3. Практика роботи з Debugger в Arduino IDE 2.....	39
РОЗДІЛ 5	
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	
5.1. Структурно-функціональний аналіз технологічного процесу	
5.2. Розрахунок освітлення приміщення комп'ютерного кабінету	
5.3. Безпека в надзвичайних ситуаціях	
ВИСНОВКИ.....	41
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	42
ДОДАТОК А «ЛІСТИНГ	
ПРОГРАМНОГО ПРОДУКТУ».	45

ВСТУП

У сучасному світі, де люди стають усе більш зайнятими, а ритм життя постійно пришвидшується, автоматизація багатьох процесів переходить із категорії зручності до категорії необхідності. Це повною мірою стосується й догляду за рослинами — як кімнатними, так і садовими. Автоматичний полив рослин — це спеціалізована система, що дозволяє економити час і фізичні зусилля, водночас забезпечуючи оптимальні умови зволоження для рослин у будь-який час, навіть за відсутності людини.

Найголовніше, що потрібно рослинам у період відсутності доглядача — це стабільне забезпечення водою. Існує два глобальні підходи до вирішення цього завдання. Перший полягає у максимальному збереженні вологи, яку рослина отримала під час останнього поливу. Такий варіант підходить для нетривалої відсутності, наприклад, для двотижневого від'їзду, і в деяких випадках — навіть до трьох тижнів. Його основна перевага полягає в тому, що він практично не потребує фінансових витрат, однак ефективність обмежена й залежить від виду рослин і умов утримання.

Інший підхід — створення системи, яка забезпечить рослини водою автоматично, без участі людини. Сьогодні існує безліч рішень для поливу кімнатних і садових рослин, зокрема автоматичні системи, які є незамінними у випадках, коли рослини мають виживати без людського втручання протягом тривалого часу — трьох тижнів і більше. Однією з ключових переваг таких систем є мінімальні зусилля, необхідні для їхньої експлуатації. Автоматичний полив може бути реалізований у формі дощування, крапельного або внутрішньогрунтового поливу, кожен з яких має свої особливості та сфери застосування.

Основною функцією автоматичного поливу є точне дозування води для кожної окремої рослини відповідно до її потреб. Таким чином, автоматичний полив — це ефективне, надійне та сучасне рішення для підтримання здоров'я і життєздатності

зелених насаджень

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ТА ОБҐРУНТУВАННЯ ПРОЕКТНОГО РІШЕННЯ

1.1. Аналіз існуючих аналогів

Системи поливу — це сукупність інженерно-технічних рішень, призначених для забезпечення зрошення визначеної ділянки. Сьогодні існує велика кількість компаній, що спеціалізуються на розробці та впровадженні автоматизованих систем поливу рослин. Однією з таких є компанія «GARDENA» [1], яка пропонує широкий спектр садового інструменту та обладнання для автоматизованого догляду за рослинами. Їхні системи автоматичного поливу дозволяють зрошувати рослини в саду або будинку згідно з наперед заданим графіком.



Рисунок 1.1. Логотип компанії GARDENA

Наведемо кілька основних характеристик систем автоматичного поливу від компанії Gardena:

Автоматизація процесу: Управління поливом здійснюється автоматично завдяки вбудованим таймерам або датчикам вологості ґрунту. Користувач має змогу задати розклад поливу відповідно до індивідуальних потреб ділянки.

Гнучкість до умов навколишнього середовища: Деякі моделі мають здатність реагувати на погодні зміни — наприклад, знижувати інтенсивність або повністю вимикати полив у разі дощу чи високої вологості ґрунту.

Рациональне водоспоживання: Системи розроблені з акцентом на економне використання водних ресурсів, забезпечуючи рівномірний розподіл вологи між рослинами.

Модульна архітектура: Конструкція таких систем дозволяє їх масштабувати залежно від розмірів території чи кількості рослин, що потребують зрошення.

Зручність користування: Системи Gardena прості в монтажі та налаштуванні, тому підходять як для досвідчених користувачів, так і для новачків.

Таким чином, продукція Gardena орієнтована насамперед на зрошення прибудинкових ділянок або великих рослинних композицій у приміщенні. Для типових домашніх умов подібна система часто виявляється надмірно складною в інсталяції та надто дорогою, що знижує доцільність її використання в побуті.

1.2. Постановка задач для проектного пристрою

Автоматизовані системи поливу функціонують із мінімальним втручанням користувача. Вони можуть працювати за заданим розкладом як протягом кількох годин, так і протягом кількох місяців безперервно, що дозволяє лише періодично змінювати налаштування відповідно до пори року або тривалості відсутності людини.

Серед основних методів автоматизованого поливу виділяють такі:

– Ґрунтовий метод: базується на використанні датчиків вологості, які визначають рівень зволоження ґрунту. Після досягнення критичної позначки датчик надсилає сигнал на контролер, що активує подачу води. Метод доступний і нескладний у впровадженні, але в горшкових умовах вода може розподілятися нерівномірно.

– Листковий метод: передбачає розпилення води на стебла та листя, імітуючи природне дощування. Такий спосіб доречний для рослин, які

потребують підвищеної вологості повітря. Водночас неправильне застосування може призвести до перезволоження ґрунту або розвитку грибкових захворювань.

– Крапельний полив: полягає в доставлянні води безпосередньо до зони коріння за допомогою трубок або крапельниць. Це економічний та ефективний метод, який зменшує втрати води, пригнічує ріст бур'янів і підтримує стабільний рівень вологості. Однак налаштування такої системи може вимагати більше часу та ресурсів.

– Часовий метод поливу: передбачає подачу води у заздалегідь визначений час незалежно від фактичної вологості ґрунту. Це простий у реалізації спосіб, але його недоліком є ймовірність надлишкового або недостатнього зволоження, оскільки система не враховує фактичні потреби рослин.

Таким чином, кожен із методів має свої особливості, переваги та недоліки, і вибір конкретного варіанта залежить від типу рослин, умов середовища та технічних можливостей користувача.



Рисунок 1.2. Система автоматизованого поливу

Отже, після аналізу різних методів автоматичного поливу рослин можна зробити висновок, що ґрунтовий метод є найбільш точним у визначенні потреби рослини у волозі, однак він підвищує загальну вартість обладнання і не

гарантує рівномірного розподілу води в межах горщика. Метод поливу по листю виявляється доцільним лише на відкритих просторах або під накриттям, оскільки у приміщенні він може створювати надмірну вологість повітря, що спричиняє утворення грибка і шкодить як рослинам, так і здоров'ю людини. Крапельний полив вважається дуже ефективним, оскільки волога подається безпосередньо до кореневої системи, проте налаштування системи потребує точності для уникнення нерівномірності у подачі води. Метод часового поливу, незважаючи на свій головний недолік — відсутність зворотного зв'язку з вологістю ґрунту, може виявитися ефективним у стабільних умовах, наприклад, у приміщеннях з постійною температурою. У такому випадку можна заздалегідь розрахувати оптимальний інтервал та об'єм води для кожної рослини. Таким чином, саме часовий метод поливу може бути найкращим компромісом у кімнатних умовах за умови правильного налаштування.

При виборі системи автоматичного поливу важливо враховувати такі фактори, як кількість і види рослин (оскільки вони мають різну потребу у воді), розміри горщиків (що прямо впливає на необхідний об'єм води), умови освітлення та температури (рослини на сонці чи в теплому середовищі потребують більше вологи), а також бюджет користувача, оскільки вартість систем може значно відрізнятися залежно від функціоналу.

Загальний висновок: необхідно розробити пристрій, який дозволяє враховувати кількість рослин і індивідуальну потребу кожної з них у волозі. Користувач має мати можливість самостійно задавати параметри: об'єм води, частоту поливу, інтервали між подачами, а також враховувати розмір горщика. Таким чином, система має забезпечувати гнучке налаштування для кожної рослини окремо, що дозволить досягти максимально ефективного зволоження з урахуванням особливостей догляду в домашніх умовах.

1.3. Аналіз технологій розробки та вибір засобів реалізації проекту.

Розробка автоматизованої системи поливу рослин на перший погляд може здаватися складною, проте за умови ретельного планування та системного підходу її цілком можливо реалізувати власноруч. Основними етапами в такій розробці є аналіз потреб, підбір обладнання та врахування джерел живлення.

1. Оцінка потреб. На цьому етапі слід:– Визначити типи рослин, які потребуватимуть поливу, адже різні види мають різні вимоги до вологості ґрунту;

– Оцінити масштаби системи: площу ділянки або кількість горщиків, що дозволить розрахувати необхідну кількість компонентів;– Визначити джерело водопостачання — це може бути підключення до централізованої системи або використання резервуару з водою;– Встановити бюджет, який вплине на вибір технічних рішень, матеріалів і рівень автоматизації системи.

2. Вибір технічних компонентів. Сюди входять:– Контролер — центральний елемент управління, що відповідає за налаштування часу та частоти поливу. Можна обрати як прості моделі з таймерами, так і сучасні контролери з Wi-Fi-модулем для керування через смартфон;– Датчики — вологоміри ґрунту та сенсори дощу, які дозволяють уникати надмірного поливу та економити ресурси;– Крапельниці або форсунки, які забезпечують подачу води до кожної рослини згідно з потребами;– Трубопровідна система — підбір труб і з'єднувачів відповідного діаметра та матеріалу для транспортування води;

– Насос, який знадобиться, якщо тиск води в джерелі недостатній для повноцінної роботи системи;– Джерело живлення — залежно від умов, це може бути як мережеве підключення, так і акумуляторні батареї. Якщо передбачається автономна робота на тривалий час, варто додатково інтегрувати сонячну панель, яка забезпечить підзарядку акумулятора і зробить систему незалежною від наявності користувача.

Таким чином, при правильному плануванні й налаштуванні, можна побудувати надійну, гнучку і майже повністю автономну систему поливу, яка враховуватиме індивідуальні потреби кожної рослини, економитиме ресурси і забезпечить стабільний догляд навіть за тривалої відсутності власника.

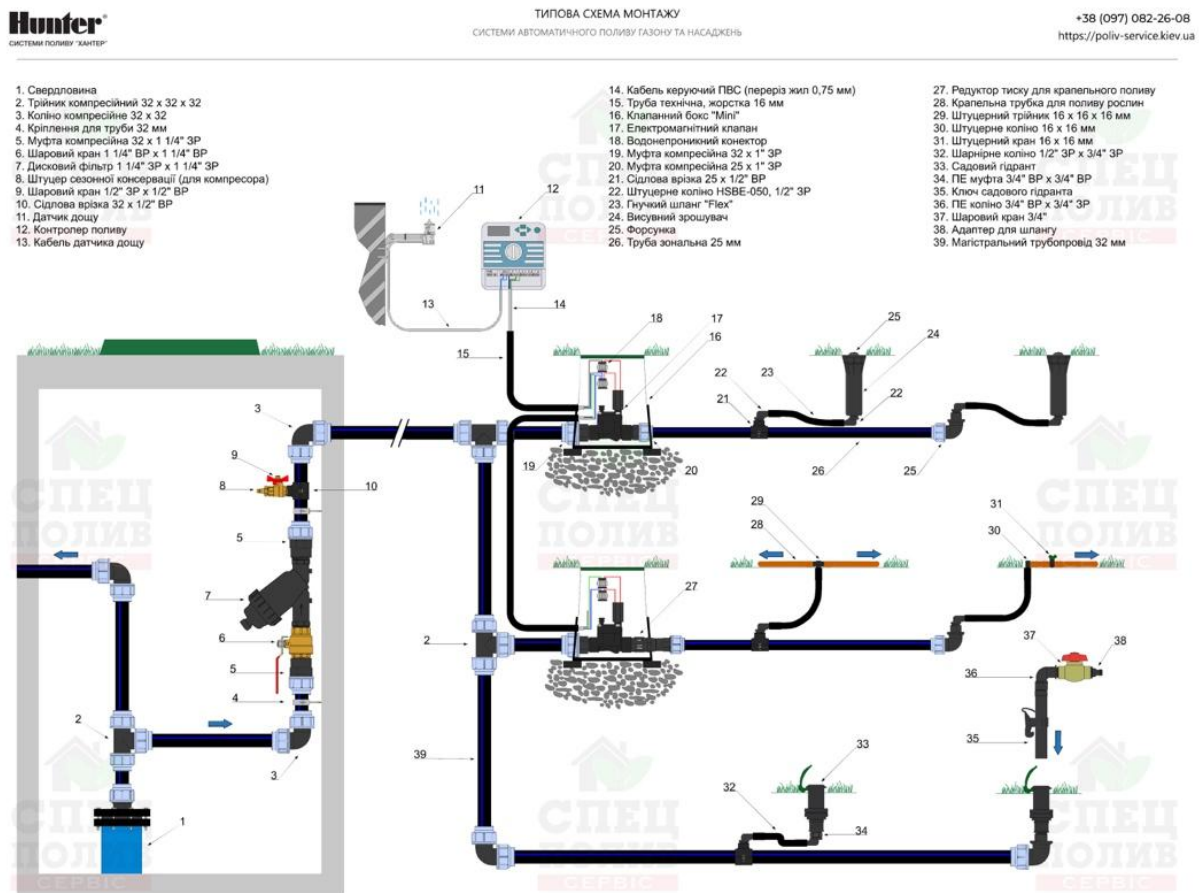


Рисунок 1.3. Схема типової системи автоматичного поливу: контролер, датчики, форсунки, клапани, трубопроводи та насосне обладнання

Проектування системи автоматичного поливу включає кілька ключових етапів. Спершу здійснюється створення плану: рослини розміщуються таким чином, щоб мінімізувати кількість насосів. Завдяки використанню фітінгів можна забезпечити полив кількох рослин одним насосом, особливо якщо їм потрібно більше води або коротший інтервал між поливами. Проводиться розрахунок довжини труб, що дозволяє точно визначити необхідну кількість матеріалів. Далі встановлюється кількість насосів та підбираються їх розміри залежно від характеристик системи і кількості рослин.

На етапі програмування та тестування необхідно запрограмувати контролер згідно з графіком поливу, який відповідає потребам кожної рослини. Після встановлення всієї системи проводиться тестування з метою перевірки її роботи: чи забезпечується подача води в потрібному обсязі, чи всі компоненти функціонують справно. У разі виявлення недоліків виконується коригування — змінюється розклад подачі води або коригується розміщення форсунок.

У цьому розділі було розглянуто предметну область, проаналізовано призначення, сфери застосування та різновиди автоматизованих систем поливу рослин. Було обрано відповідний метод поливу та сформовано базову ідею його технічної реалізації.

РОЗДІЛ 2

РОЗРОБКА ПРИНЦИПОВИХ ТА ЕЛЕКТРИЧНИХ ВУЗЛІВ ТА БЛОКІВ ПРИСТРОЮ

2.1 Розробка алгоритма та структурної схеми

У сучасному світі, де пріоритетами стають швидкість, зручність та ефективність, автоматизація все активніше впроваджується в різні сфери життя, зокрема й у догляд за кімнатними рослинами. Вони не лише прикрашають інтер'єр, а й створюють атмосферу спокою й затишку. Проте забезпечення правильного догляду за ними, особливо регулярного поливу, може бути складним завданням для людей з обмеженим часом або відсутністю відповідного досвіду.

У зв'язку з цим дедалі більшої популярності набувають автоматизовані системи поливу, які орієнтовані на пересічного користувача. Такі рішення базуються на використанні сучасних технологій, зокрема мікроконтролерів та сенсорів, що дозволяє автоматизувати процес зволоження ґрунту та забезпечити стабільні умови для росту рослин без постійного втручання людини.

Узагальнена структура запропонованої системи автоматичного поливу показана на рисунку 2.2. До її складу входять: блок живлення, ручне керування, цифрова індикація, релейний модуль, насос, а також мікроконтролер, який виконує функції управління та відліку часу.

Живленням для системи може слугувати акумулятор або джерело постійної напруги на 5 вольт.

На рисунку 2.1 зображено блок-схему алгоритму роботи пристрою. Функціонування системи організоване за циклічним принципом, що забезпечує її безперервну та автономну роботу.

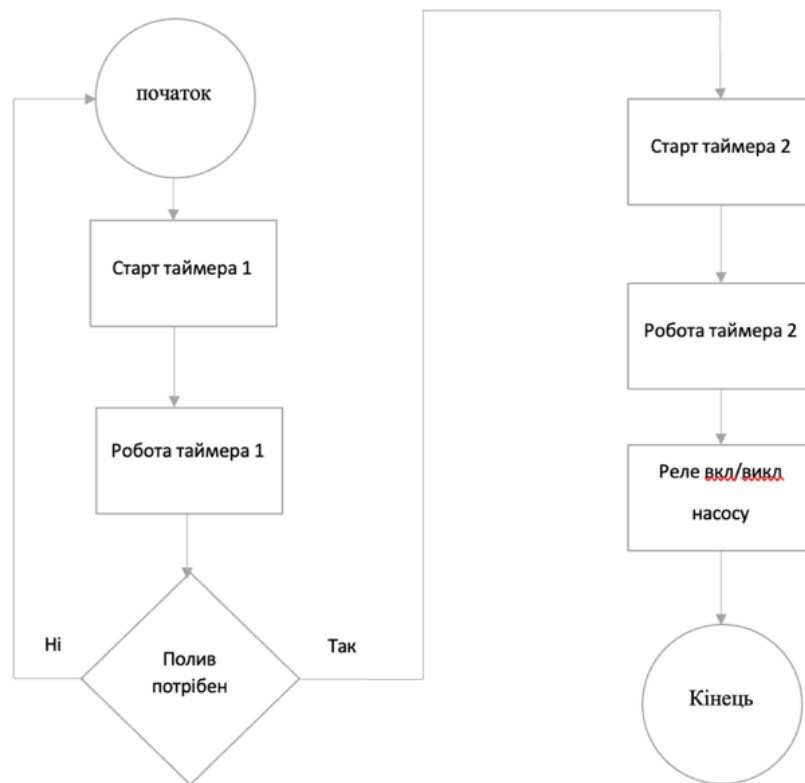


Рисунок .2.1 Алгоритм роботи проектованого пристрою

Після подачі живлення система проходить ініціалізацію, під час якої мікроконтролер встановлює початкові значення для часових параметрів. Далі користувач за допомогою блоку ручного керування може задати тривалість поливу та інтервал між його циклами. Задана інформація виводиться на блок цифрової індикації для візуального контролю.

Після введення параметрів мікроконтролер передає команду лічильнику часу, який запускає таймер. Увесь подальший процес роботи системи керується цим таймером. Коли настає встановлений момент для поливу, активується таймер поливу, який подає сигнал на релейний модуль. Реле, у свою чергу, вмикає насос, що забезпечує подачу води. Після завершення заданого часу поливу таймер надсилає команду на реле для відключення насоса.

Одночасно лічильник передає інформацію до мікроконтролера про завершення поточного циклу. Мікроконтролер, отримавши цей сигнал, ініціює новий цикл, запускаючи зворотний відлік до наступного поливу. У такий спосіб система працює циклічно, забезпечуючи автоматичне зволоження рослин відповідно до заданого користувачем графіка.

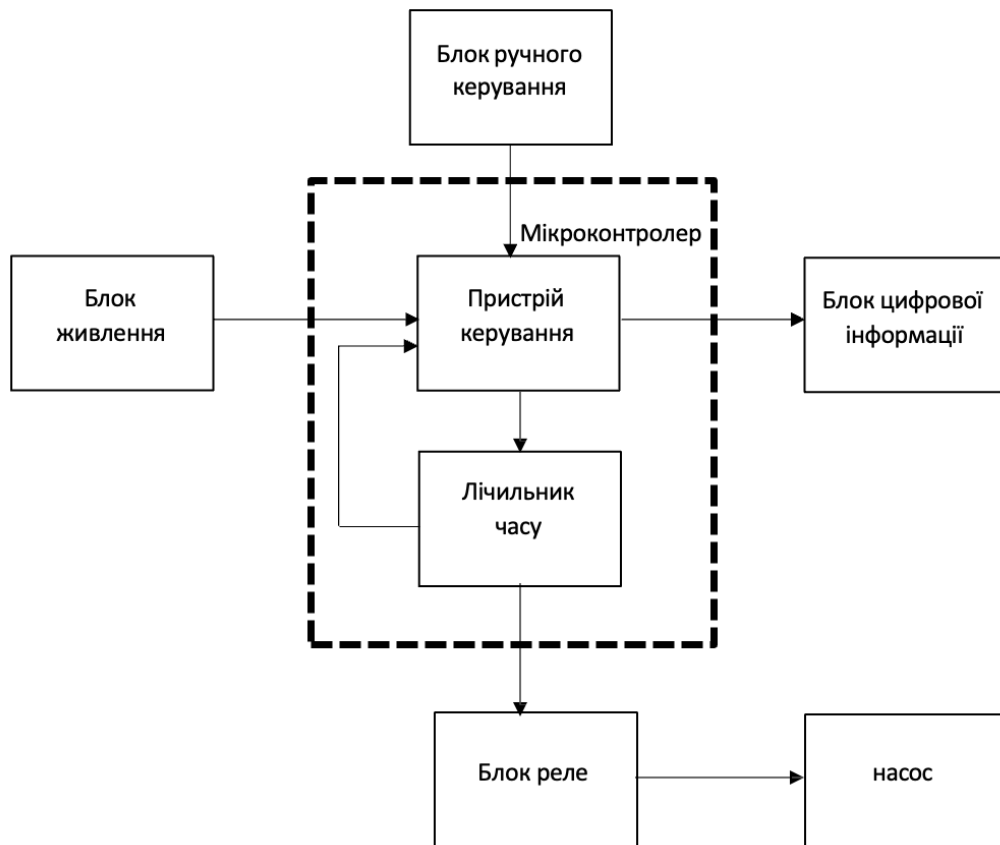


Рисунок 2.2 – Структурна схема проектованого пристрою

На рисунку показано структурну схему роботи системи автоматичного поливу. Основним керуючим елементом є мікроконтролер, який здійснює управління всіма компонентами на основі отриманих вхідних сигналів. Живлення системи забезпечується за допомогою блоку живлення, а взаємодія з користувачем реалізується через блок ручного керування. Для встановлення тривалості поливу використовується таймер або лічильник часу. Передача та отримання інформації, зокрема від датчиків вологості, здійснюється через блок цифрової індикації. Безпосереднє увімкнення насоса виконується за допомогою релейного модуля, який спрацьовує відповідно до алгоритму, закладеного в мікроконтролері. Така схема дозволяє організувати як автоматичний, так і ручний режим роботи, що забезпечує високу адаптивність та стабільність функціонування системи.

2.2 Дослідження та аналіз сфери розробки

У сучасних умовах стрімкого розвитку технологій аналіз та створення принципових схем відіграють ключову роль у впровадженні інновацій в інженерних сферах. Цей розділ присвячено розгляду основ проектування та реалізації принципових електронних схем, що формують основу для сучасних електротехнічних і механічних систем. Під час розгляду акцент зроблено як на теоретичних підходах, так і на практичних реалізаціях із урахуванням сучасних вимог до швидкодії, надійності та енергоефективності. Також розглянуто актуальні напрямки в галузі розробки схемотехнічних рішень і практичне застосування відповідних програмних засобів.

Одним із перших засобів проектування є EAGLE (Easily Applicable Graphical Layout Editor) [5] — популярне програмне забезпечення для створення електронних схем та друкованих плат, розроблене компанією CadSoft. EAGLE надає зручне середовище для схематичного проектування, ручного та автоматичного трасування плат, а також має широкі бібліотеки компонентів із можливістю створення власних. Програма підтримує обмін файлами з іншими САПР та дозволяє переглядати тривимірну модель плати.

AutoCAD Electrical [6] є потужним спеціалізованим інструментом для створення електричних схем і систем автоматизації, розробленим компанією Autodesk. Він містить професійні інструменти для розміщення елементів, автоматичну нумерацію компонентів, створення специфікацій і генерацію звітів, а також підтримує інтеграцію з іншими системами проектування та програмування ПЛК.

Microsoft Visio [7] — це універсальний графічний редактор для створення різноманітних діаграм і структурних схем. Завдяки великій кількості шаблонів і інтеграції з іншими продуктами Microsoft, Visio широко використовується для створення організаційних, мережевих та процесних схем. Підтримується колективна робота в режимі реального часу, що робить програму зручною для спільних проєктів.

EasyEDA [8] — це хмарна платформа, яка надає повноцінне середовище для створення електронних схем і проєктування друкованих плат. Основна перевага EasyEDA полягає в тому, що вона працює безпосередньо у веб-браузері, не потребуючи встановлення на комп'ютер. Завдяки великій бібліотеці компонентів, підтримці імпорту/експорту у різних форматах (Gerber, KiCad, Altium, Eagle) та можливості колективної розробки, ця платформа є зручною як для новачків, так і для досвідчених інженерів.

Провівши аналіз розглянутих програмних засобів для створення принципових схем, можна дійти висновку, що EasyEDA є найбільш зручним інструментом. Це пояснюється її доступністю через браузер, розширеними функціональними можливостями, значною кількістю вбудованих компонентів та простим, інтуїтивно зрозумілим інтерфейсом, який значно полегшує процес проєктування.

2.3 Проектування функціональної та принципової схеми

Робота розробленого пристрою базується на обчисленні часу поливу та інтервалу між циклами зрошення. До основних технічних параметрів системи належать:

- вимірювання та відлік часу, що відводиться на полив, а також паузи між поливами;
- виведення інформації на цифровий дисплей для налаштування режимів роботи;
- наявність релейного модуля, який відповідає за активацію насоса.

Живлення системи здійснюється від джерела з напругою 5В, а рівень споживаного струму визначається характеристиками конкретного типу насоса, що використовується. На рисунку 2.3 подано функціональну схему автоматизованої системи поливу.

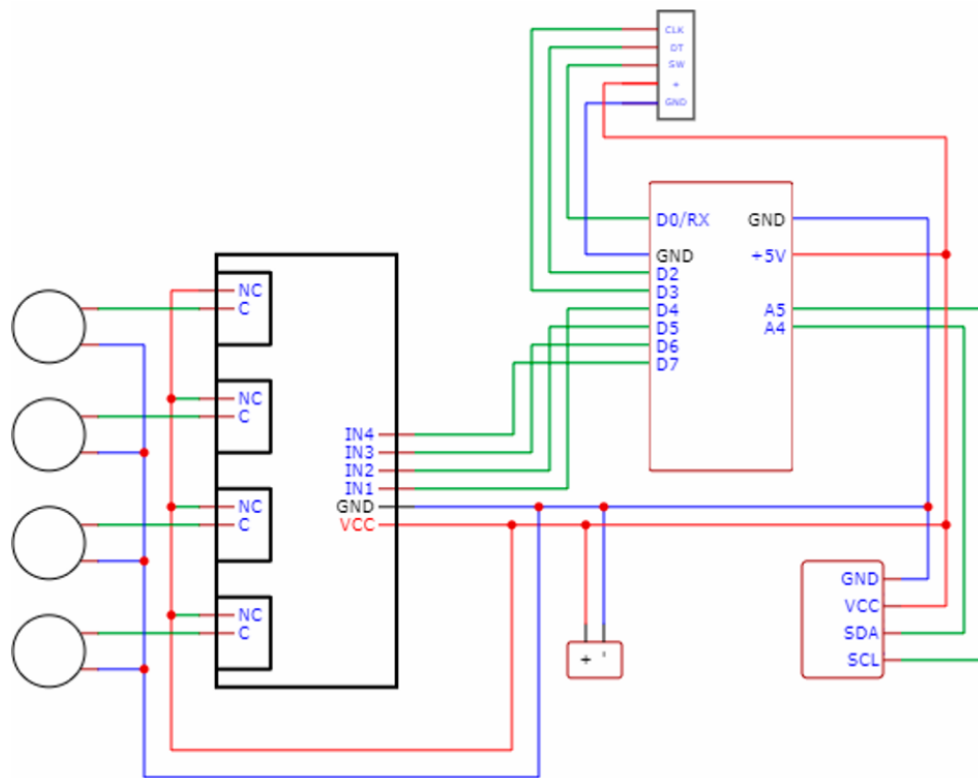


Рисунок 2.3 – Функціональна схема пристрою автоматичного поливу

Для створення принципової схеми спочатку необхідно підібрати всі необхідні компоненти системи, визначити їх технічні характеристики, а також спроектувати друковану плату, на якій вони будуть з'єднані між собою.

Складові системи та схема їх підключення:

1. Arduino NANO [9] використовується як основний елемент керування. Він відповідає за обробку сигналів, отриманих від сенсорів, а також за управління виконавчими пристроями, такими як реле й насос. На рисунку 2.4 представлено вигляд цього мікроконтролера згори.



Рисунок 2. 4 Вид зверху Arduino NANO

Таблиця 2.1 Характеристики Arduino NANO

Модель	Arduino NANO
Мікроконтролер	ATmega328
Робоча напруга	5В
Напруга живлення (рекомендована)	7-12В
Напруга живлення (гранична)	6-20В
Цифрові входи/виходи	14
Аналогові виходи	8
Максимальний струм одного виходу	40 <u>мА</u>
<u>Flash</u> -пам'ять	32 <u>Кб</u>
SRAM	2 <u>Кб</u>
Тактова частота	16МГц
Розміри плати	1,85 см х 4,3 см

АТmega328 [14] — це 8-бітний мікроконтролер, побудований на RISC-архітектурі AVR, розроблений компанією Microchip Technology. Він входить до відомої лінійки мікроконтролерів AVR, яка вирізняється зручністю у використанні, доступністю та широким функціональним потенціалом. Завдяки цим перевагам АТmega328 широко використовується як у навчальних платформах для початківців, так і в складних рішеннях для вбудованих систем. На рисунку 2.5 зображено мікропроцесор Atmega328.

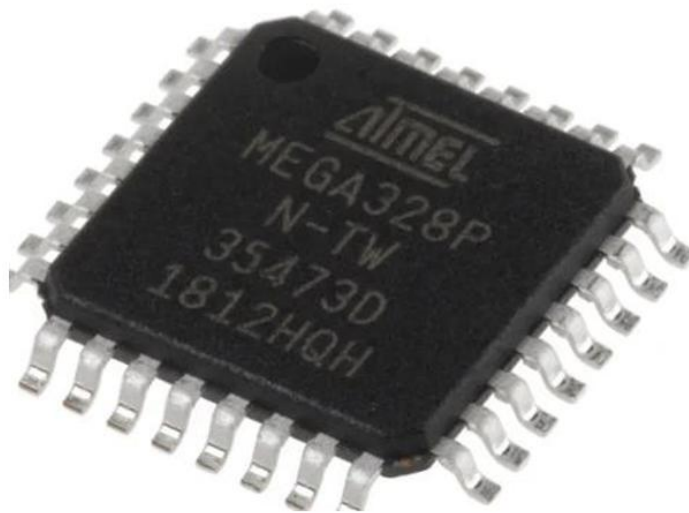


Рисунок 2.5. Процесор «Atmega 328»

Архітектура та основні характеристики

Мікроконтролер АТmega328 побудований на 8-бітній RISC-архітектурі AVR® і може працювати з тактовою частотою до 20 МГц. Він оснащений 32 КБ Flash-пам'яті з підтримкою функції читання під час запису, 1 КБ EEPROM для збереження даних при вимкненні живлення та 2 КБ оперативної пам'яті SRAM, яка використовується під час виконання програми.

У сфері периферії мікроконтролер має 23 універсальні лінії вводу/виводу (GPIO), які можуть бути використані для підключення різноманітних датчиків, пристроїв керування тощо. Також він оснащений трьома 16-бітними таймерами/лічильниками з можливістю порівняння, що

дозволяє реалізовувати точне вимірювання часу або генерацію імпульсів. Шестиканальний 10-бітний АЦП забезпечує перетворення аналогових сигналів у цифрову форму. Для організації послідовного зв'язку в ATmega328 передбачено модулі USART і SPI, а також підтримуються внутрішні й зовнішні джерела переривань.

Мікроконтролер працює в діапазоні напруги живлення від 1.8 до 5.5 В. Можливі варіанти корпусу включають DIP28, TQFP32 та QFN32.

Переваги ATmega328 полягають у його доступності, простоті програмування, широкому функціональному наборі, а також активній спільноті користувачів, що спрощує отримання допомоги. Водночас, до недоліків можна віднести 8-бітну архітектуру, яка може обмежувати продуктивність у складніших застосуваннях, та обмежений обсяг пам'яті, що не завжди підходить для великих проєктів.

ATmega328 знаходить застосування у великій кількості пристроїв, зокрема: у платі Arduino NANO, яка є надзвичайно популярною серед початківців; у робототехнічних системах, де використовується для керування моторами, датчиками та іншими елементами; у вбудованих пристроях (наприклад, термостати, охоронні системи, медичне обладнання); а також у численних аматорських проєктах для створення або модифікації електронних приладів.

Релейний модуль [10] — це компонент, який дозволяє керувати високопотужними пристроями через керуючий сигнал від мікроконтролера Arduino. Він виконує функцію електронного перемикача, що дозволяє замикати або розмикати електричне коло, забезпечуючи керування навантаженням, яке перевищує можливості самого мікроконтролера. На рисунку 2.6 представлено чотириканальне реле.



Рисунок 2.6 «Реле 4-х каналъне»

Модуль містить чотири комутуючі реле з керуванням через оптоізоляцію, що забезпечує захист мікроконтролера від зворотних імпульсів. Керування здійснюється безпосередньо з цифрових виходів більшості мікроконтролерів, зокрема Arduino. Кожне реле здатне комутувати навантаження до 10А при напрузі 250В. Чотириканальний релейний модуль ідеально підходить для керування потужними пристроями в складі автоматизованих систем.

Насос [11] — це компактний занурювальний водяний насос, призначений для перекачування рідини між різними ємностями. Його продуктивність може досягати 120 літрів за годину. Такий насос широко застосовується в автоматизованих системах поливу, акваріумах, міні-фонтанах, а також у сферах гідропоніки та аквапоніки. При використанні в складі системи з сенсорами він дозволяє створити ефективне рішення для автоматичного поливу та підтримки оптимальних умов для рослин. Працює в діапазоні напруги живлення 2.5–6 В, що робить можливим його використання з автономними джерелами, зокрема сонячними панелями. У разі повного

занурення у воду насос потребує надійного кріплення за корпус та забезпечення додаткової герметизації в зоні виходу кабелю.

Характеристики водяного насоса наведено в таблиці 2.2.

Напруга живлення	3 – 6 Вольт
Потужність	0,4 – 1,5 Вт
Швидкість перекачування рідини	До 2 л/хв або 120 л/год
Матеріал корпусу	Пластик, <u>загерметизований</u>
Максимальна висота водяного стовпа	0,4 – 1,1 м
Зовнішній діаметр при виході	7,5 мм
Внутрішній діаметр на виході	4,7 мм
Розмір	24 мм x 45 мм x 33 мм
Може перекачувати масло і воду	
Струм залежить від напруги живлення	

Зображення насоса 2.7 «водяна помпа»



Рисунок 2.7 «Водяна помпа»

Модуль енкодера [12] EC11 призначений для реалізації елементів введення та управління у різноманітних електронних проєктах. Він має зручне підключення без потреби пайки, оскільки на платі вже встановлені підтягувальні резистори для сигналів енкодера, а також передбачене місце для встановлення резистора, пов'язаного з кнопкою. Завдяки компактним розмірам та монтажним отворам, цей модуль зручно використовувати як початківцям, так і досвідченим користувачам Arduino. Дискретність становить 20 кроків на один повний оберт валу.

Модуль має 20 фіксованих положень при повороті, що забезпечує точне зчитування положення. Принцип дії енкодера базується на механічному перемиканні: зміна положення осі змінює стан контактів A і B (CLK та DT), а також C (GND). Вал оснащений кнопкою, яка активується натисканням і може використовуватися, наприклад, для вибору пунктів меню, зміни режимів або чутливості.

Сигнали від енкодера мають низький рівень TTL при замиканні контактів, у такому разі на виходи CLK і DT подається логічний нуль. Подача живлення 5 В генерує високий логічний рівень. Виводи CLK і DT відповідають за передачу даних про напрямок обертання валу — за годинниковою стрілкою або проти неї.

Призначення виводів енкодера:

- CLK — вихід сигналу з контакту A
- DT — вихід сигналу з контакту B
- SW — сигнал від кнопки
- VCC — подача живлення • GND — контакт "землі" (C)

На рисунку 2.8 наведено зовнішній вигляд модуля енкодера.

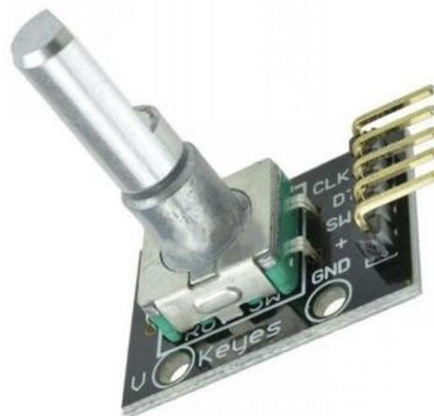


Рисунок 2.8 «Енкодер»

LCD-дисплей 1602 [13] призначений для використання з мікроконтролерами Arduino. Він має два рядки по 16 символів, що дозволяє виводити текстову інформацію на екрані. Дисплей сумісний зі стандартною бібліотекою LiquidCrystal, яка входить до складу Arduino IDE, що спрощує його

підключення та програмування. На рисунку 2.9 зображено дисплей LCD 1602.



Рисунок 2.9 «Дисплей LCD 1602»

Таблиця 2.4 Характеристики Дисплею LCD 1602

Розміри	
Робоча температура	0 – 50 C°
Колір символів	Білий
Підсвітка	Блакитна
Розмір символу	4.35 мм x 2.95 мм
Формат	16 x 2
Розмір крапки	0.5 мм x 0.5 мм
Інтерфейс	HD44780
Видима область	64.5 мм x 13.8 мм
Живлення	5 В

Схема підключення основних компонентів системи має такий вигляд:

Підключення енкодера здійснюється через п'ять контактів: два для живлення та три сигнальні. Контакт GND з'єднується з відповідним контактом GND на платі Arduino, а VCC підключається до виводу 5V. Сигнальний контакт CLK (Clock) з'єднується з цифровим входом D2, DT (Data) — з D3, а контакт кнопки SW — з входом D4. Вибір пінів D2 і D3 пояснюється тим, що вони підтримують зовнішні переривання, необхідні для точного зчитування імпульсів від енкодера.

LCD-дисплей 16x2 з інтерфейсом I2C підключається за допомогою чотирьох контактів: GND — до загальної землі Arduino, VCC — до 5V, контакт SDA — до аналогового входу A4, а SCL — до A5. Завдяки використанню інтерфейсу I2C значно зменшується кількість проводів, що спрощує підключення дисплея.

Для релейного модуля використовуються чотири сигнальні лінії: IN1, IN2, IN3 та IN4 підключаються до цифрових пінів Arduino, наприклад D5, D6, D7 та D8. Живлення подається через контакт 5V, а GND модуля приєднується до GND плати Arduino.

Щодо підключення водяної помпи, то один її контакт під'єднується до джерела живлення 5 В, інший — до виводу реле (COM). Нормально розімкнутий контакт реле (NO) приєднується до загального проводу (GND) цього ж джерела. Така конфігурація дозволяє вмикати насос через реле лише під час поливу, відповідно до заданої програми.

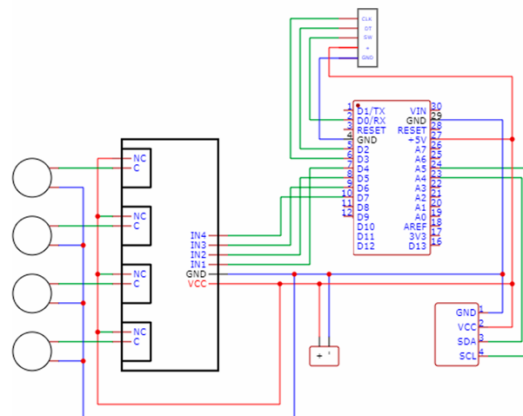


Рисунок 2.10 Принципова схема проектуваного пристрою

Живлення системи організовано наступним чином: плата Arduino отримує енергію через USB або від зовнішнього адаптера з напругою 5 В, підключеного до відповідного роз'єму. Релейний модуль разом із клапанами підключається до окремого зовнішнього джерела живлення з тією ж напругою — 5 В. Принципова електрична схема пристрою зображена на рисунку 2.10.

РОЗДІЛ 3

РОЗРОБКА ДРУКОВАНОЇ ПЛАТИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Дослідження лазерно-утюжної технології

Процес розробки друкованих плат (ДП) є ключовим етапом у створенні будь-якого електронного пристрою. В умовах стрімкого розвитку технологій та постійного вдосконалення електроніки виникає потреба у швидкому та доступному виготовленні плат. Для масового виробництва застосовують професійні методи, зокрема фотолітографію, автоматизоване свердління та машинне встановлення компонентів. Ці технології забезпечують високу точність та надійність, проте вимагають дорогого обладнання та спеціальних умов, що робить їх малопридатними для індивідуального використання чи невеликих проєктних команд.

Одним із найбільш поширених і доступних варіантів виготовлення друкованих плат у домашніх умовах є лазерно-утюжна технологія (ЛУТ). Вона дозволяє створювати якісні плати з мінімальними витратами, без необхідності використовувати складне обладнання. Суть методу полягає в тому, що зображення електричної схеми друкується лазерним принтером на спеціальному глянцево-папері, після чого переноситься на мідну поверхню текстоліту за допомогою звичайної праски. Цей підхід є зручним для виготовлення прототипів або обмежених серій виробів.

Серед переваг лазерно-утюжного методу можна відзначити наступне:

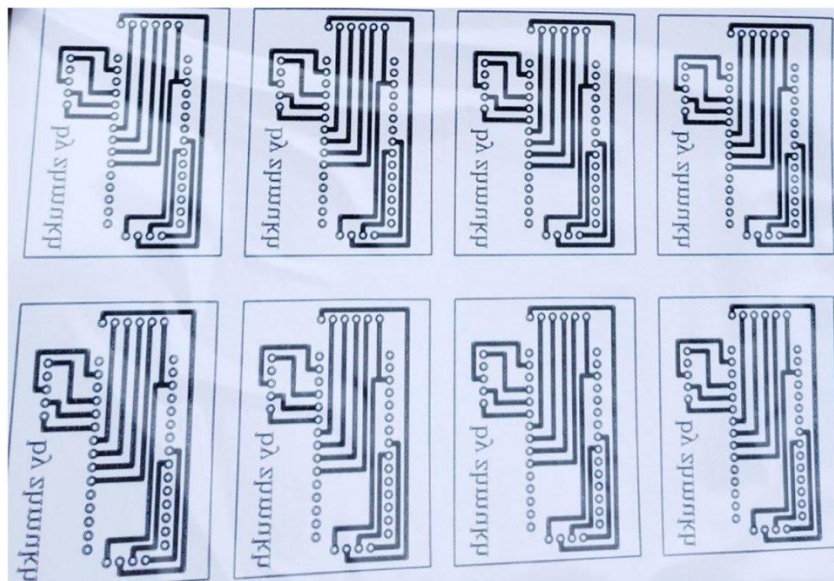
1. Доступність — усі необхідні матеріали (лазерний принтер, праска, мідний текстоліт, засіб для травлення) легко знайти, що робить метод зручним для більшості користувачів.
2. Простота — освоїти технологію можна без спеціальних знань, що робить її привабливою для початківців.

3. Економічність — витрати на виготовлення значно нижчі, ніж при використанні промислових методів, що дозволяє створювати прототипи без великих фінансових вкладень.

4. Швидкість — увесь процес від проектування до готової плати займає всього кілька годин, що ідеально підходить для оперативного тестування нових рішень.

Процедура виготовлення плат за ЛУТ починається з друку схеми на лазерному принтері. Для цього потрібен не звичайний папір, а глянцевий — з журналів, фотопапір або спеціальний термотрансферний папір. Саме глянцеве покриття дозволяє тонеру залишатися на поверхні та легко переноситися на мідь, тоді як звичайний папір вбирає тонер, що унеможлиблює його перенесення. Схему слід друкувати у дзеркальному відображенні, дотримуючись реального масштабу. В налаштуваннях друку потрібно вимкнути економію тонера та встановити найвищу контрастність — це гарантує якість зображення. Приклад правильно надрукованої схеми представлений на рисунку 4.1.

Після друку малюнок переноситься на підготовлений текстоліт. Контур майбутньої плати обводиться, після чого сам текстоліт вирізається за розміром. Для цього бажано використовувати лобзик або ножівку, адже ножиці по металу можуть пошкодити мідне покриття або деформувати матеріал.



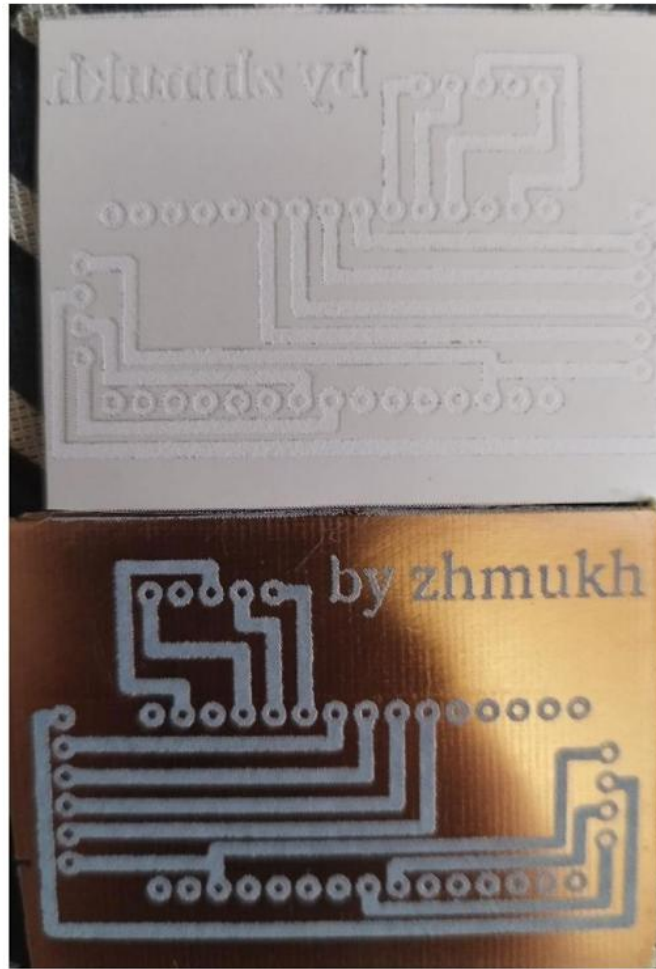
Зображення 4.1 роздрукований рисунок плати

Перш ніж розпочати виготовлення друкованої плати, необхідно підготувати текстоліт до нанесення зображення. Для цього слід ретельно зачистити його поверхню до блиску за допомогою наждачного паперу із зернистістю приблизно 2000. Далі потрібно промити текстоліт мильною водою за допомогою губки під проточною водою, а після цього обов'язково знежирити його спиртом або ацетоном. Це необхідно для усунення жирових або маслянистих слідів, які можуть завадити якісному перенесенню тонера на мідну поверхню.

Наступний етап — перенесення зображення схеми на текстоліт. Для цього слід скористатися праскою, попередньо нагрітою до температури близько 200 градусів. Спочатку кладемо текстоліт на звичайний аркуш паперу, накриваємо ще одним аркушем і прогріваємо приблизно 30 секунд. Це дозволяє текстоліту трохи розширитися, щоб під час нанесення зображення не виникло дефектів.

Після попереднього прогрівання на текстоліт накладаємо надруковану схему й розпочинаємо основний процес перенесення. Протягом двох хвилин рівномірно та з помірним натиском праскою прогріваємо папір, поки тонер не почне плавитися й переноситися на мідну поверхню. Важливо уважно стежити за процесом: потрібно поступово і ретельно прасувати, доки папір не почне злегка жовтіти, що свідчить про достатній нагрів.

Коли перенесення завершено, обережно відокремлюємо папір від поверхні текстоліту. Залишки паперу змиваємо під водою за допомогою ватного тампона або м'якої гумки, очищаючи плату від волокон. Результат переносу схеми на мідну основу можна побачити на рисунку 4.2.



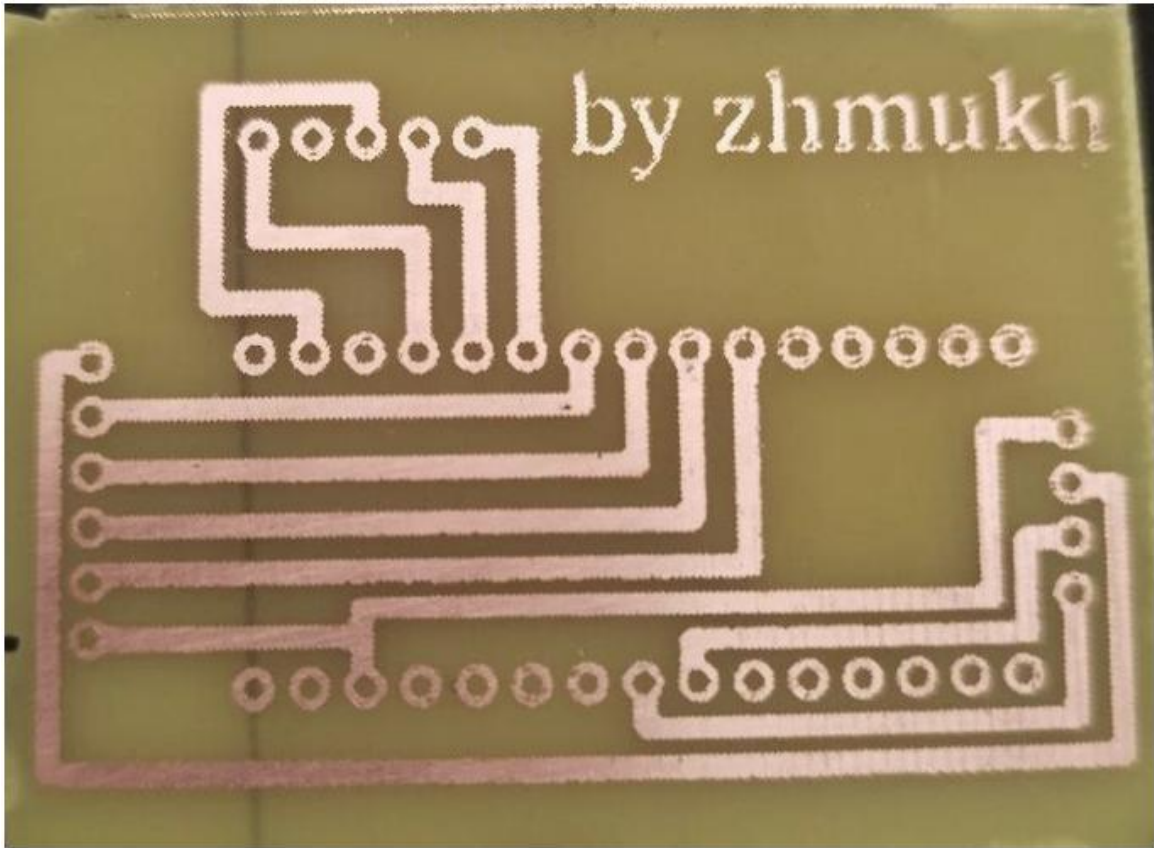
Зображення 4.2 результат термічної обробки

Переходимо до етапу травлення друкованої плати. Раніше для цього широко використовували хлорне залізо, однак сьогодні існують більш прості та доступні альтернативи.

Для приготування розчину, за допомогою якого буде проводитися травлення, необхідно використати звичайні побутові компоненти: аптечний перекис водню, кухонну сіль та лимонну кислоту. Оптимальні пропорції суміші: 30 мл перекису водню, 10 грамів солі та 5 грамів лимонної кислоти. Усі інгредієнти ретельно перемішуються, після чого в отриманий розчин занурюється плата. Процес травлення триває приблизно 10 хвилин.

Після завершення травлення плату промивають під проточною водою, а тонер, який залишився на поверхні, знімають за допомогою ацетону. Далі необхідно просвердлити отвори для встановлення компонентів. Це можна зробити за допомогою бормашини або мінішурупверта. Завершальний етап —

лудіння контактних доріжок, що виконується досить просто й швидко. Після цього можна переходити до монтажу елементів — і плата вважається готовою до використання. Результат виконаної роботи зображено на рисунку 4.3.



Зображення 4.3 результат роботи

Лазерно-утюжна технологія є одним із найефективніших методів виготовлення друкованих плат у домашніх умовах. Вона дає змогу швидко та з мінімальними витратами створювати прототипи, що особливо важливо на етапі розробки й випробування електронних схем. Завдяки простоті, невисокій вартості та легкій доступності матеріалів, цей спосіб набуває все більшої популярності серед студентів, ентузіастів і розробників електроніки

3.2. Програмування Ардуіно

Цей код призначений для керування системою поливу, яка складається з кількох насосів, використовуючи енкодер для налаштувань і ЖК-дисплей для відображення інформації. Повний код знаходиться у Додатку А. Ось основні компоненти та принципи роботи коду:

Wire.h: Бібліотека для роботи з I2C, використовується для підключення дисплея.

LiquidCrystal_I2C.h: Бібліотека для роботи з LCD-дисплеєм через I2C.
Encoder.h: Бібліотека для роботи з енкодером.

lcd: Об'єкт для роботи з LCD-дисплеєм, адреса 0x27, дисплей має 16 символів на рядок та 2 рядки.

myEnc: Об'єкт для роботи з енкодером, підключеним до контактів D2 та D3.

const int relayPins[4] = {4, 5, 6, 7};

Масив контактів для підключення 4 реле до контактів D4, D5, D6 та D7.

wateringTime : Масив для зберігання часу поливу для кожного з 4 реле.

currentRelay: Індекс поточного реле, який обирається для налаштування часу.

timeMode: Режим часу, true означає секунди, false означає години.

oldPosition: Зберігає попереднє значення енкодера для виявлення змін.

lcd.begin(): Ініціалізація дисплея. lcd.backlight(): Увімкнення підсвітки дисплея.

lcd.print("Система поливу"): Виведення повідомлення на дисплей. for (int i = 0; i < 4; i++): Налаштування контактів реле як виходів та вимикання реле.

pinMode(8, INPUT_PULLUP): Налаштування кнопки енкодера як входу з внутрішнім підтягуючим резистором.

delay(2000): Затримка на 2 секунди для відображення початкового повідомлення.

```

updateDisplay(): Оновлення відображення на дисплеї.
long newPosition = myEnc.read(): Отримання поточного положення
енкодера.
if (newPosition != oldPosition): Перевірка, чи змінилося положення
енкодера.
oldPosition = newPosition: Оновлення старої позиції.
wateringTime[currentRelay] = newPosition: Оновлення часу поливу для
поточного реле.
updateDisplay(): Оновлення відображення на дисплеї.
if (digitalRead(8) == LOW): Перевірка натискання кнопки енодера.
delay(500): Затримка для антидребезгу.
if (digitalRead(8) == LOW): Перевірка, чи кнопка все ще натиснута. if
(timeMode): Якщо режим часу - секунди, перемикаємо реле. currentRelay =
(currentRelay + 1) % 4: Перемикання між реле.
else: Якщо режим часу - години, перемикаємо режим часу. timeMode =
!timeMode: Зміна режиму часу. updateDisplay(): Оновлення відображення на
дисплеї.
for (int i = 0; i < 4; i++): Перевірка кожного реле для поливу. if
(wateringTime[i] > 0): Якщо час поливу більше 0. digitalWrite(relayPins[i], LOW):
Вмикання реле.
if (timeMode): Якщо режим часу - секунди. delay(wateringTime[i] * 1000):
Затримка для поливу у секундах. else: Якщо режим часу - години.
delay(wateringTime[i] * 3600000): Затримка для поливу у годинах.
digitalWrite(relayPins[i], HIGH): Вимикання реле.
wateringTime[i] = 0: Скидання часу поливу.
lcd.clear(): Очищення дисплея. lcd.print("Реле "): Виведення тексту
"Реле".
lcd.print(currentRelay + 1): Виведення номера поточного реле.
lcd.setCursor(0, 1): Переміщення курсора на другий рядок. lcd.print("Час: "):
Виведення тексту "Час:". lcd.print(wateringTime[currentRelay]): Виведення часу

```

поливу для

поточного реле.

if (timeMode): Перевірка режиму часу.

lcd.print(" c"): Виведення символу "c" для секунд. else: Якщо режим часу - години.

3.3 Виявлення синтаксичних помилок у середовищі Arduino IDE

Перш ніж розпочати роботу з більш складними інструментами налагодження, необхідно навчитися розпізнавати типові синтаксичні помилки, що виникають під час написання програм для мікроконтролерів. На початковому етапі розробки в середовищі Arduino IDE компілятор автоматично перевіряє синтаксис програми й у разі виявлення помилки видає повідомлення з описом проблеми. Це допомагає швидко визначити місце помилки та внести необхідні виправлення, що є базовим рівнем захисту від некоректної роботи пристрою. На рисунку нижче наведено приклад типової синтаксичної помилки — некоректного виклику функції `delay()`, яка спричиняє помилку під час компіляції.

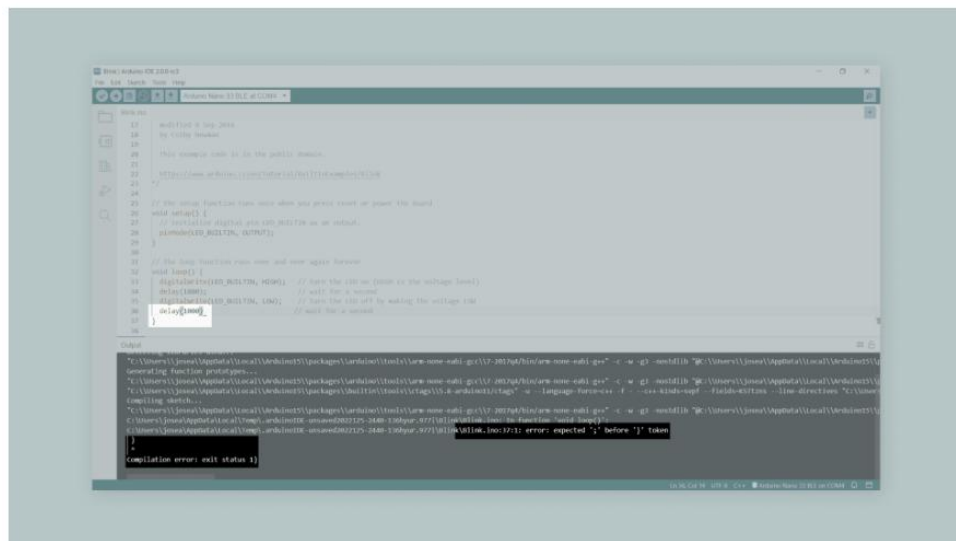


Рисунок 4.1— Повідомлення про синтаксичну помилку в Arduino IDE під час компіляції програми

На рисунку 4.1 представлено приклад виявлення синтаксичної помилки в середовищі розробки Arduino IDE 2.0, що є характерною ситуацією під час створення програмного забезпечення для вбудованих систем на базі Arduino.

Зображення демонструє код, призначений для керування вбудованим світлодіодом плати Arduino Nano 33 BLE, а також виведення повідомлення про помилку в нижній панелі «Output». Програма містить стандартні функції `setup()` і `loop()`, при цьому помилка виникає у виклику функції `delay`: замість `delay(1000);` написано `delay1000;`, через що компілятор повертає повідомлення про помилку: "expected ';' before ')' token", тобто очікувалась крапка з комою перед дужкою. Це типовий приклад синтаксичної помилки.

Такий випадок демонструє важливість своєчасного виявлення помилок на ранніх етапах програмування, що є невід'ємною частиною процесу відлагодження коду. Подібний приклад доцільно включити до підрозділу 4.3 «Тестування та налагодження пристрою» як наочну ілюстрацію поширених помилок та засобів їх діагностики в середовищі Arduino IDE.

РОЗДІЛ 4

ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ ПРИСТРОЮ

4.1 Огляд функціональності Debugger в Arduino IDE 2

Arduino IDE 2 — це оновлене середовище розробки, яке включає в себе новий інструмент для налагодження коду (Debugger). Завдяки цьому інструменту розробники можуть здійснювати покрокове виконання програм безпосередньо на мікроконтролері, що істотно полегшує виявлення логічних помилок, збоїв у роботі або непередбаченої поведінки програмного забезпечення.

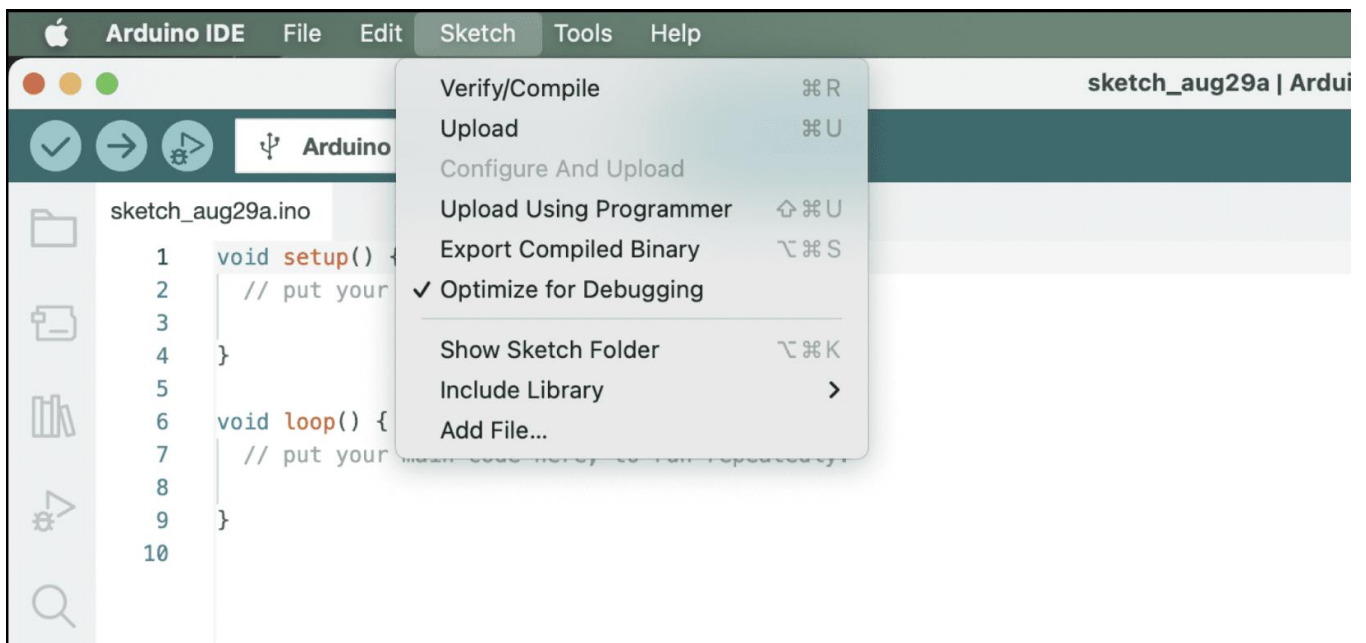


Рисунок 4.1. Інтерфейс налагоджувача в Arduino IDE 2

У вбудованому засобі налагодження (Debugger) реалізовано функції покрокового виконання коду, встановлення точок зупину (breakpoints), перегляду поточних значень змінних, а також навігації по функціях: вхід у функцію (step in), пропуск (step over) та вихід із функції (step out). Доступ до Debugger здійснюється через бічну панель середовища Arduino IDE 2. Варто зазначити, що ця функція підтримується лише на сумісних платах, таких як Arduino Zero, MKR WiFi 1010, Nano 33 IoT та інших моделях на базі мікроконтролерів архітектури SAMD.

Застосування Debugger надає можливість детально дослідити роботу програми під час її виконання та отримати глибше уявлення про внутрішні процеси, які відбуваються на рівні мікроконтролера.

4.2 Підключення апаратних засобів налагодження

Щоб скористатися можливостями Debugger, необхідно підключити відповідний апаратний засіб налагодження. Існує три основні способи реалізації такого підключення. Одним із них є використання плати Arduino Zero, яка вже містить вбудований налагоджувач і підключається через спеціальний програмувальний USB-порт. Це робить її зручною для освітніх цілей та простих проєктів, де не передбачено використання додаткового обладнання.



Рисунок 4.2 Підключення відлагоджувача Atmel-ICE до плати MKR WiFi 1010 через SWD-інтерфейс

Segger J-Link — це професійний засіб налагодження, що використовує інтерфейс SWD для підключення. Щоб під'єднати його до плати MKR WiFi 1010, потрібно попередньо припаяти контакти до відповідних виводів для налагодження. Такий підхід забезпечує надійне з'єднання та сумісність з широким набором мікроконтролерів, гарантуючи стабільну роботу системи під час відлагодження.

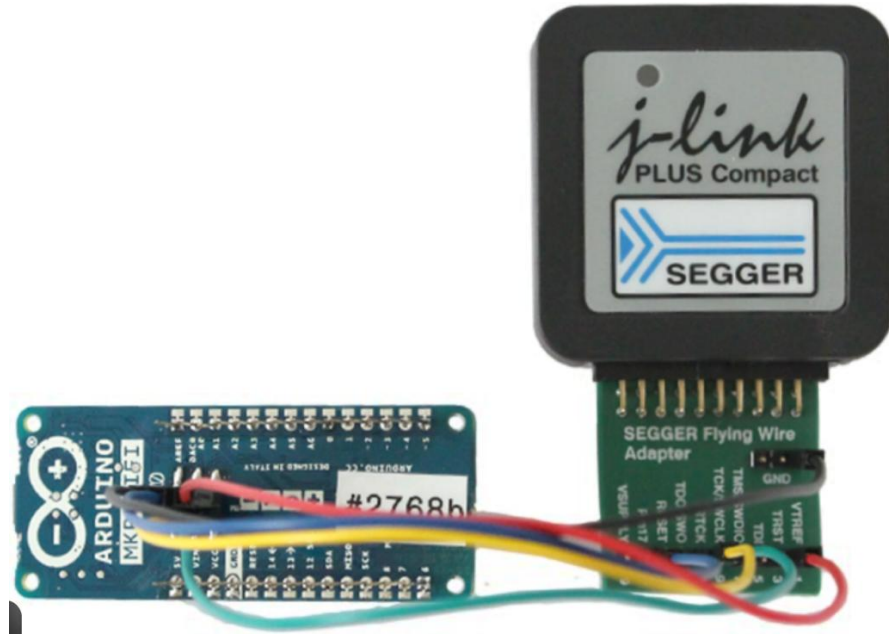


Рисунок 4.3. Підключення відлагоджувача Segger J-Link до плати MKR WiFi 1010 через інтерфейс SWD

Atmel-ICE — це ще один зовнішній засіб налагодження, який, подібно до J-Link, працює через інтерфейс SWD. Він забезпечує апаратне налагодження коду та офіційно підтримується середовищем Arduino IDE 2. Для підключення також необхідно припаяти контакти до відповідних виводів налагодження, як і у випадку з J-Link.

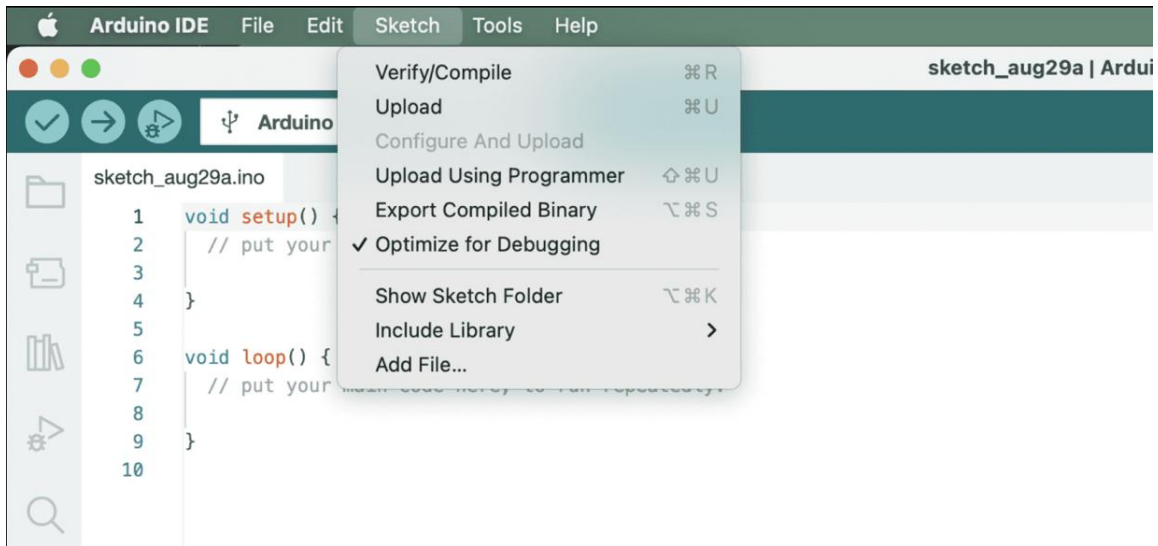


Рисунок 4.4. Arduino Zero з вбудованим відлагоджувачем та активованою опцією Optimize for Debugging в Arduino IDE 2

Усі перелічені способи налагодження дають змогу запускати програму в контрольованих умовах, встановлювати точки зупину, аналізувати значення змінних та виконувати код покроково. Такий функціонал є надзвичайно важливим під час роботи зі складними проєктами, де необхідна точна діагностика та глибоке розуміння логіки виконання програми.

4.3 Практика роботи з Debugger в Arduino IDE 2

Після підключення необхідного апаратного засобу можна переходити безпосередньо до процесу налагодження. Перед завантаженням скетчу потрібно активувати параметр "Optimize for Debugging" у налаштуваннях компілятора. Це дозволить зберегти відлагоджувальну інформацію під час компіляції програми.

Для прикладу можна використати стандартний скетч Blink (File > Examples > Basics > Blink). Після його завантаження на сумісну плату з'являється можливість увімкнути режим Debugger.

На рисунку нижче показано приклад запуску налагодження стандартного скетчу Blink в Arduino IDE 2. У нижній панелі відображено вивід GDB-сервера, який демонструє спробу встановити з'єднання з пристроєм через інтерфейс SWD.

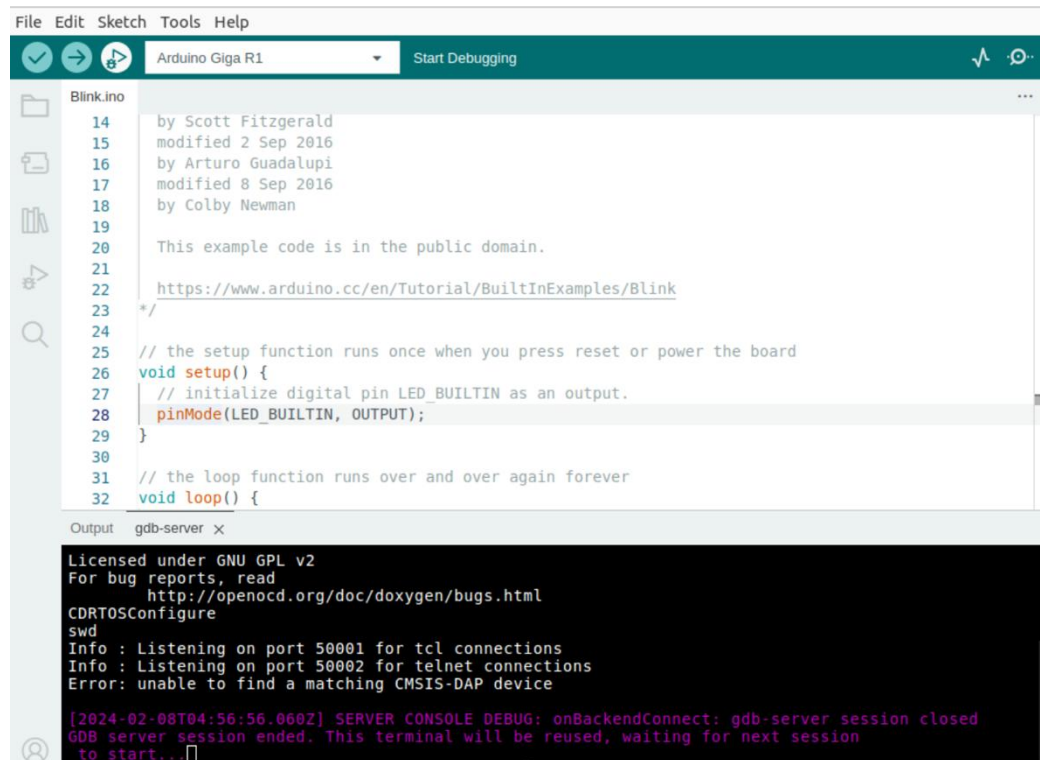


Рисунок 4.4. Інтерфейс Arduino IDE 2 під час запуску налагодження скетчу Blink та виведення повідомлень GDB-сервера

Спочатку в коді встановлюються breakpoints — точки зупину, наприклад, у тих рядках, де здійснюється ввімкнення або вимкнення світлодіода. Під час виконання програма зупиняється в цих точках, що дозволяє проаналізувати поточні значення змінних або перейти до наступного етапу виконання.

Для навігації можна використовувати інструменти: Continue (для продовження виконання до наступної точки зупину), Step Over (перехід до наступного рядка без входу у функцію), Step In (вхід у функцію, наприклад, digitalWrite()) та Step Out (вихід із функції назад до основного коду).

Ці інструменти особливо корисні для детального аналізу складного коду, кращого розуміння структури бібліотек і внутрішніх викликів, що використовуються у вбудованих системах. Завдяки підтримці налагодження в Arduino IDE 2 розробник отримує змогу значно підвищити точність, ефективність і надійність створюваного програмного забезпечення для мікроконтролерів.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Структурно-функціональний аналіз технологічного процесу

Розробка та вживання ефективних заходів запобігання аварійним і травмонебезпечних ситуаціям можливі лише при завчасному виявленні тих небезпек, з яких починаються процеси їх формування. Оскільки небезпечні умови не завжди завчасно можна виявити, а для вивчення небезпечних дій іноді потрібно багато часу, щоб зібрати статичний матеріал, то і методи виявлення цих небезпек повинні бути відповідно диференційовані (табл. 5.1).

Таблиця 4.1. - Моделі формування та виникнення травмонебезпечних і аварійних ситуацій

Вид робіт, виробничий підрозділ, робоче місце, виробниче обладнання, склад агрегату	Виробнича небезпека			Можливі наслідки	Заходи запобігання небезпечним ситуаціям
	Небезпечна умова (НУ)	Небезпечна дія (НД)	Небезпечна ситуація (НС)		
Виконання робіт із електрооблад- нанням	Не вимкнено живлення. Відсутність заземлення.	Нехтування правилами ТБ	Ураження струмом	Травма (Т)	Проведення повторного інструктажу з ТБ. Розробка нових способів захисту. Встановлення заземлення.
НД ↓ НУ —→ НС —→ Т					

Відповідно до аналізу небезпечних умов, які існують у виробничому процесі виокремлено такі наступні за характером дії на працівника їх групи:

- характеризують стан або рівень безпеки обладнання, які використовуються.
- сприяють виникненню технологічних помилок обслуговуючого персоналу впродовж виробничого процесу;

- створювати умови та можливість проникнення працівника в небезпечну зону;
- приводять до виникнення небезпечних дій (внаслідок низького рівня професійної підготовки працівників та організації навчання з охорони праці).

Моделі формування та виникнення травмонебезпечних і аварійних ситуацій в комп'ютерному кабінеті представлено у вигляді моделі формування та виникнення травмонебезпечних і аварійних ситуацій – табл. 5.1.

5.2. Розрахунок освітлення приміщення комп'ютерного кабінету.

Освітленість виробничих приміщень може бути штучною і природною. Природне освітлення при правильному обладнанні найбільш сприятливе для людини. Основні вимоги для освітлення наступні:

- освітлення повинне бути достатнім для швидкого і легкого розпізнання об'єктів роботи;
- освітлення повинно бути рівномірне без різких тіней;
- джерело світла не повинно осліплювати працівника;
- рівень освітленості не повинен обмежуватись часом.

Природне освітлення забезпечується обладнанням вікон (бокове освітлення) фонарів і світильних покриттів приміщень (верхнє освітлення). Природне освітлення нормується коефіцієнтом природної освітленості. Коефіцієнт природної освітленості – це процентне відношення фактичної освітленості F_v в будь-якій точці приміщення до освітленості F_n розсіяної світлом небозводу точки, яка лежить на відкритій місцевості. Розрахунок природного освітлення через бокові вікна по нормам освітленості ведеться для самої дальньої від вікон точки, тобто знаходять мінімальне значення стик коефіцієнта природної освітленості:

$$e_{\min} = \frac{F_b}{F_n} \cdot 100. \quad (4.1)$$

Значення коефіцієнта природної освітленості визначається не менше чим в п'яти точках. Значення коефіцієнта природної освітленості для

сільськогосподарських виробничих приміщень в даному випадку ремонтній майстерні, беремо $e_{min} = 5\%$

Розрахунок природного освітлення зводиться до визначення площі світлових променів.

Сумарну площу світлових променів $\sum F_o (m^2)$ по коефіцієнту природної освітленості для бокових променів визначаємо по формулі:

$$\sum F_o = \frac{F_n \cdot e_{\min} \cdot r_o \cdot K}{100 \cdot \tau \cdot \Gamma_1}, \quad (4.2)$$

де F_n – площа підлоги; $e_{\min}$ – величина мінімального коефіцієнта природного освітлення; τ – загальний коефіцієнт світловикористання віконного отвору із врахуванням його забруднення, $\tau = 0,25$; r_o – світлова характеристика вікна, $r_o = 9,5$; Γ_1 – коефіцієнт, який враховує підвищення освітленості за рахунок світла, яке відбивається від стін і стелі, $\Gamma_1 = 1,2$; K – коефіцієнт, який враховує затінення вікон сусідніми приміщеннями і загорожею, $K = 1$.

$$\sum F_o = \frac{36 \cdot 0,5 \cdot 9,5 \cdot 1}{100 \cdot 0,25 \cdot 1,2} = 5,7 \text{ м}^2.$$

Кількість світлових променів визначимо:

$$N = \frac{\sum F_o}{F_o}, \quad (4.3)$$

де F_o – площа вікна згідно стандарту, м^2 .

$$N = \frac{5,7}{6} = 0,95.$$

Приймаємо кількість вікон – одне вікно.

При розрахунку природного освітлення найбільш поширеним і простим є метод світлового потоку. При цьому методі розраховуємо світловий потік F_l (Лк), який повинна випромінювати кожна лампа (при заданій кількості ламп).

$$F_l = \frac{k \cdot S_n \cdot E}{n_l \cdot \eta \cdot r^2}, \quad (4.4)$$

де k – коефіцієнт запасу, $k = 1,3$; S_n – площа підлоги, м^2 ; $S_n = 36 \text{ м}^2$.

E – нормативна освітленість, $E = 300 \text{ Лк}$; n_l – кількість встановлених ламп, $n_l = 6$ од; η – коефіцієнт використання світлового потоку, $\eta = 0,25$; r – коефіцієнт нерівномірності освітленості, $r = 0,545$.

Коефіцієнт запасу (K) враховує можливість забруднення світильників пилом, що залежить від характеру виробництва.

Розрахунок штучного освітлення починаємо із визначення висоти розташування світильника і їх кількості. Висоту h_n (м) розташування світильників над робочим місцем знаходимо за формулою:

$$h_n = H - (h_1 + h_2), \quad (4.5)$$

де H – висота приміщення, м; h_1 – віддаль від підлоги до освітлювальної поверхні, м; h_2 – віддаль від стелі до світильника, м.

$$h_n = 4,5 - (2,2 + 1,5) = 0,8 \text{ м.}$$

При симетричному розміщенні світильників по вершинах квадратів їх кількість визначається за формулою:

$$n_c = \frac{S_n}{l^2}, \quad (4.6)$$

де l – віддаль між світильниками, м.

Підставивши значення отримаємо:

$$n_c = \frac{36}{9} = 4 \text{ од.}$$

Тоді світловий потік буде становити

$$F_{\text{л}} = \frac{1,3 \cdot 36 \cdot 300}{4 \cdot 0,25 \cdot 0,545} = 2576,2 \text{ Лк.}$$

При світловому потоці 2576,2 Лк для заданої лампи вибираємо тип і потужність.

Вибираємо тип лампи – люмінесцентну, потужністю 40Вт.

5.3. Безпека в надзвичайних ситуаціях

Забезпечення захисту населення і території у разі загрози і виникнення надзвичайних ситуацій є одним з найважливіших завдань держави.

Захист населення є системою загально-державних заходів, які реалізуються центральними і місцевими органами виконавчої влади, виконавчими органами влад, органами управління з питань надзвичайних ситуацій та цивільного захисту

населення, підпорядкованими їм системами, та підприємств, що забезпечують виконання організаційних, інженерно – технічних, санітарно – гігієнічних, проти епідемічних та інших заходів у сфері запобігання та ліквідації наслідків надзвичайних ситуацій.

Загрози життєво важливих інтересів громадян, держави, суспільства поділяють на зовнішні та внутрішні, виконують під час надзвичайних ситуацій техногенного та природного характеру та воєнних конфліктах.

Принципи захисту впливають з основних положень Женевської конвенції щодо захисту жертв війни та додаткових протоколів до неї, можливого характеру воєнних дій, реальних можливостей держави щодо створення матеріальної бази захисту. З метою захисту населення, зменшення втрат та шкоди економіці в разі виникнення надзвичайних ситуацій має право проводитись спеціальний комплекс заходів.

Оповіщення та інформування, яке досягається завчасним створенням і підтримкою в постійній готовності загально-державної, територіальних та об'єктивних систем оповіщення населення.

ВИСНОВКИ

У результаті виконання дипломної роботи було реалізовано повний цикл створення вбудованої системи: від аналізу існуючих аналогів і вибору оптимальної архітектури до проектування принципової схеми, програмування пристрою та впровадження сучасних методів відлагодження з використанням середовища Arduino IDE 2. Під час проектування було обґрунтовано вибір мікроконтролера з урахуванням специфіки задачі, а також досліджено можливості підключення зовнішніх модулів та датчиків.

Особливу увагу приділено налагодженню програмного забезпечення, оскільки воно є критично важливим етапом у забезпеченні стабільності та передбачуваної поведінки системи. Завдяки використанню інструменту Debugger, інтегрованого в Arduino IDE 2, вдалося дослідити внутрішню логіку роботи програми, проаналізувати перебіг виконання коду, знайти та усунути помилки. Практичне застосування методів трасування, встановлення точок зупину (breakpoints), покрокового виконання (step over, step in, step out) дозволило отримати глибше розуміння процесів, що відбуваються в мікроконтролері під час роботи системи.

Було протестовано кілька варіантів підключення апаратних засобів налагодження, зокрема Arduino Zero з вбудованим дебагером, Segger J-Link та Atmel-ICE. Це дало змогу обґрунтовано підійти до вибору найдоцільнішого інструменту в залежності від складності проєкту, бюджету та необхідної глибини контролю.

Загалом виконана робота підтвердила важливість поєднання апаратної та програмної частини вбудованої системи, а також продемонструвала ефективність використання сучасних відкритих інструментів і платформ для розробки та налагодження. Отримані результати можуть бути використані для створення аналогічних пристроїв у навчальних, дослідницьких або комерційних цілях.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dalal S., Seth B., Radulescu M., Secara C., Tolea C. Predicting Fraud in Financial Payment Services through Optimized Hyper-Parameter-Tuned XGBoost Model // *Mathematics*. – 2022. – Vol. 10, No. 24. – P. 4679. – DOI: 10.3390/math10244679.
2. Lindholm A., Wahlström N., Lindsten F., Schön T.B. Supervised Machine Learning: Lecture Notes for the Statistical Machine Learning Course. – Uppsala: Uppsala University, Department of Information Technology, 2019. – P. 111–112.
3. Sharma V. Survey of Classification Algorithms and Various Model Selection Methods // *Journal of Machine Learning Research*. – 2000. – Vol. 1. – P. 1–48.
4. Hyeoun-Ae P. An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain // *J Korean Acad Nurs*. – 2013. – Vol. 43, No. 2. – P. 154–164. – DOI: 10.4040/jkan.2013.43.2.154.
5. Rokach L., Maimon O. The Data Mining and Knowledge Discovery Handbook. – New York: Springer, 2005. – P. 165–192. – DOI: 10.1007/0-387-25465-X_9.
6. Entropy and Information Gain to Build Decision Trees in Machine Learning [Електронний ресурс]. – Режим доступу: <https://www.section.io/engineering-education/entropy-information-gain-machine-learning/> (дата звернення: 20.04.2023).
7. Decision Trees Explained — Entropy, Information Gain, Gini Index, CCP Pruning [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/decision-trees-explained-entropy-information-gain-gini-index-ccp-pruning-4d78070db36c> (дата звернення: 22.04.2023).
8. Cutler A., Cutler R.D., Stevens R.J. Ensemble Machine Learning: Methods and Applications. – New York: Springer, 2011. – P. 157–176. – DOI: 10.1007/978-1-4419-9326-7_5.
9. Wang Y., Pan Z., Zheng J., Qian L., Li M. A Hybrid Ensemble Method for Pulsar Candidate Classification // *Astrophysics and Space Science*. – 2019. – Springer Nature B.V. – DOI: 10.1007/s10509-019-3602-4.
10. Is Machine Learning Beneficial in Detecting Fraud? [Електронний ресурс]. – Режим доступу: <https://medium.com/mlearning-ai/how-beneficial-is-detecting-fraud-using-machine-learning-b00089d84930> (дата звернення: 28.04.2023).

11. XGBoost: its Genealogy, its Architectural Features, and its Innovation [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/xgboost-its-genealogy-its-architectural-features-and-its-innovation-bf32b15b45d2>(дата звернения: 28.04.2023).

12. 24 Evaluation Metrics for Binary Classification (And When to Use Them) [Электронный ресурс]. – Режим доступа: <https://neptune.ai/blog/evaluation-metrics-binary-classification> (дата звернения: 01.05.2023).

13. Amaratunga D., Cabrera J., Lee Y.-S. Enriched Random Forests // *Bioinformatics*. – 2008. – Vol. 24, No. 18. – P. 2010–2014. – DOI: 10.1093/bioinformatics/btn356.

14. Vujovic D.Z. Classification Model Evaluation Metrics // *International Journal of Advanced Computer Science and Applications*. – 2021. – Vol. 12, No. 6.

ДОДАТОК А
ФРАГМЕНТ ЛІСТИНГУ ПРОГРАМНОГО ПРОДУКТУ

```
import pandas as pd
import numpy as np
import google.oauth2.credentials
import getpass
import Levenshtein
import pycountry

from google.cloud import bigquery
from sklearn.preprocessing import OneHotEncoder
from sklearn.cluster import AgglomerativeClustering as HCLUST

import warnings
warnings.filterwarnings('ignore')

token = getpass.getpass()
creds_bq = google.oauth2.credentials.Credentials(token=token)

### Count distinct card holders
LEVENSTEIN_RATIO = 0.2

def levenstein_nunique(card_holders):
    card_holders = [str(x) for x in card_holders]
    if len(card_holders) > 1000:
        return -1
    ds = np.zeros((len(card_holders), len(card_holders)))
    if len(card_holders) < 2:
        return len(card_holders)
    else:
        for i1 in range(len(card_holders)):
            for i2 in range(i1, len(card_holders)):
                d =
Levenshtein.distance(card_holders[i1].lower(),card_holders[i2].lower())
```

