

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«"Розумний автомат" для розливу води
на основі IoT технології»**

на здобуття освітнього ступеня бакалавра (магістра)
зі спеціальності 126, інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне
джерело*

_____ Кирило ПИЛИПЕНКО
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр.ІСД-43

Кирило ПИЛИПЕНКО

Ім'я, ПРІЗВИЩЕ

Керівник: Др. тех. наук, Ірина СРІБНА

Ім'я, ПРІЗВИЩЕ

Рецензент: _____

Ім'я, ПРІЗВИЩЕ

Київ 2025

6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	13.03.25-24.03.25	
2	Розділ 1. Огляд технічних засобів системи "розливу рідин"	25.03.25-28.03.25	
3	Аналіз доступних деталей та пристроїв для реалізації проекту	29.03.25-09.04.25	
4	Розділ 2. Вибір інструментів для системи "розумний автомат" на основі мікроконтролера arduino	10.04.25-19.04.25	
5	Створення схеми та програмного забезпечення для системи.	20.04.25-25.04.25	
6	Розділ 3. Розробка системи "розумний рідин" на основі мікроконтролера arduino	26.04.25-28.04.25	
7	Висновки	29.04.25-02.05.25	
8	Оформлення дипломної бакалаврської роботи (чистовий варіант)	03.05.25-21.05.25	

Здобувач(ка) вищої освіти _____
(підпис)

Кирило ПИЛИПЕНКО
(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи _____
(підпис)

Ірина СРІБНА
(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра (магістра): 56 стор., 18 рис., 20 джерел.

Об'єкт дослідження – Розумний автомат для розливу води, здатний видавати чітко задану кількість рідини.

Предмет дослідження – Розробка системи автоматичного розливу рідин, що поєднує в собі точність мікроконтролерів Arduino та можливості віддаленого керування на основі IoT технологій.

Мета роботи – Розробка автоматизованої системи дозування рідин на базі мікроконтролера з метою мінімізації втрат та забезпечення високої точності контролю потоку.

Методи дослідження – Аналіз існуючих рішень, розробка програмного забезпечення, тестування та оцінка результатів.

Короткий зміст роботи: У рамках кваліфікаційної роботи було розроблено функціональний пристрій для автоматичного розливу рідин, а також відповідне програмне забезпечення. Ця розробка дозволяє автоматизувати процес розливу, забезпечуючи точне дозування та мінімізацію витрат.

КЛЮЧОВІ СЛОВА: МІКРОКОНТРОЛЕР, РОЗУМНИЙ АВТОМАТ, РОЗЛИВ, РІДИНА, ОПТИМІЗУВАННЯ, СИСТЕМИ, ДОЗУВАННЯ, ARDUINO.

ABSTRACT

Text part of the bachelor level qualification work: 56 pages, 18 pictures, 20 sources.

Object of research - A smart water dispenser capable of dispensing a clearly defined amount of liquid.

Subject of research - Development of an automatic liquid dispensing system that combines the accuracy of Arduino microcontrollers and remote control capabilities based on IoT technologies.

Purpose of the study - Development of an automated liquid dosing system based on a microcontroller in order to minimise losses and ensure high accuracy of flow control.

Research methods - Analysis of existing solutions, software development, testing and evaluation of results.

Summary of work:

Within the framework of the qualification work, a functional device for automatic filling of liquids was developed, as well as the corresponding software. This development allows automating the filling process, ensuring accurate dosing and minimising costs.

KEYWORDS: MICROCONTROLLER, SMART MACHINE, FILLING, LIQUID, OPTIMISATION, SYSTEMS, DOSING, ARDUINO.

ЗМІСТ

	Стор.
ВСТУП.....	9
1. ОГЛЯД ТЕХНІЧНИХ ЗАСОБІВ СИСТЕМИ "РОЗЛИВУ РІДИН".....	11
1.1. Аналіз поширених систем "розливу рідин". Визначення переваг та недоліків.....	11
1.1.1 Аналіз автоматичних системи розливу.....	12
1.1.2 Аналіз неавтоматичних системи розливу.....	15
1.2. Характеристики, технічні можливості мікроконтролерів для розумних систем.....	18
1.3. Огляд компонентів пристрою для розливу рідин.....	22
Висновки до розділу 1.....	24
2. ВИБІР ІНСТРУМЕНТІВ ДЛЯ СИСТЕМИ "РОЗУМНИЙ АВТОМАТ" НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO.....	26
2.1. Огляд основних компонентів системи.....	26
2.1.1 Виділення переваг Arduino Nano.....	26
2.1.2 Опис насосу для рідин та повітря.....	28
2.1.3 Огляд характеристик драйвера двигуна MX1508.....	30
2.1.4 Переваги перемикача з виступом.....	31
2.2 Огляд додаткових компонентів системи.....	32
2.2.1 Переваги використання micro USB на DIP 2.54.....	32
2.2.2 Аналіз модуля дисплея TM1637.....	34
Висновки до розділу 2.....	35
3. РОЗРОБКА СИСТЕМИ "РОЗУМНИЙ РІДИН" НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO.....	37
3.1. Моделювання системи "розливу рідин" на основі мікроконтролера Arduino.....	37
3.2. Розробка програмного забезпечення для Arduino Uno.....	39
3.3. Опис програмного коду Arduino Nano.....	40
Висновок до розділу 3.....	56
ВИСНОВКИ.....	58
ПЕРЕЛІК ПОСИЛАНЬ.....	59
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	61

ВСТУП

Актуальність роботи. В сучасних умовах автоматизація технологічних процесів є ключовим напрямком розвитку промисловості, сервісних компаній та побутових систем. Автоматизовані пристрої дозування рідин широко використовуються в харчовій, фармацевтичній, хімічній та інших галузях промисловості, де важливо забезпечити високу точність дозування та мінімізувати втрати продукту. Використання технологій Інтернету речей та мікроконтролерів, таких як Arduino, для управління процесом розливу дозволяє створювати ефективні та гнучкі рішення, які можуть бути інтегровані в різні виробничі середовища.

Запропонована в цій роботі система автоматичного дозування рідини допомагає підвищити продуктивність, зменшити втручання людини в процес і забезпечує високу точність дозування, що особливо важливо для підприємств з високими вимогами до якості продукції.

Метою роботи є створення ефективної автоматизованої системи дозування рідини на базі мікроконтролера для мінімізації втрат рідини під час розливу та забезпечення високої точності регулювання витрати. Реалізація цієї мети дозволяє підвищити ефективність виробничих процесів та зменшити втручання людини в технологічний процес.

Об'єктом дослідження є інтелектуальний автоматизований пристрій розливу води, який здатний дозувати чітко визначену кількість рідини відповідно до заданих параметрів. Він забезпечує високу точність дозування, що сприяє оптимізації використання рідин та зменшенню їх втрат.

Предметом дослідження є процес розробки пристрою для автоматичного дозування рідини, а також створення механізму управління на основі сучасних IoT-технологій. Основний акцент зроблено на використанні мікроконтролерів

Arduino, які дозволяють точно контролювати об'єм рідини, що видається, а також інтеграцію з іншими системами автоматизації.

Методи дослідження включають аналіз існуючих рішень в області автоматизованих систем розливу, розробку програмного забезпечення для управління пристроєм, проведення серії тестів для визначення точності і стабільності роботи системи, а також оцінку результатів для подальшого вдосконалення конструкції.

Апробація результатів та публікації. Теоретичне, методичне та практичне значення отриманих результатів.

Теоретичне значення полягає в аналізі та узагальненні сучасних підходів до створення автоматизованих систем дозування рідин, розгляді ефективних алгоритмів керування на основі мікроконтролерів та IoT-рішень, які можуть бути використані для розробки подібних систем в майбутньому.

Методичне значення полягає в розробці методів проектування, програмування та тестування автоматичних пристроїв, які можуть бути використані в навчальному процесі для підготовки фахівців в галузі електроніки, автоматизації та інформаційних технологій.

Практичне значення полягає у створенні функціонального автоматизованого пристрою, який може бути впроваджений у виробничі лінії підприємств різних галузей промисловості, забезпечуючи точне дозування рідин, зниження матеріаломісткості та підвищення якості продукції.

Короткий зміст роботи: В ході виконання кваліфікаційної роботи було розроблено та сконструйовано функціональний пристрій для автоматичного дозування рідини. Крім того, створено відповідне програмне забезпечення для управління процесом розливу, що дозволяє точно регулювати об'єм дозованої рідини, оптимізувати витрати та підвищити продуктивність праці. Розроблений пристрій може бути використаний в різних галузях, в тому числі в харчовій, фармацевтичній та хімічній промисловості.

1 ОГЛЯД ТЕХНІЧНИХ ЗАСОБІВ СИСТЕМИ "РОЗЛИВУ РІДИН"

1.1 Аналіз поширених систем "розливу рідин". Визначення переваг та недоліків

Автоматизовані системи розливу рідини відіграють важливу роль у сучасному виробництві, забезпечуючи точне дозування, високу продуктивність і мінімізацію втрат сировини. Вони широко використовуються в різних галузях промисловості, включаючи харчову, фармацевтичну, хімічну та косметичну. Завдяки технологічним розробкам сучасні системи розливу здатні працювати з рідинами різної в'язкості, хімічного складу і фізичних характеристик, що робить їх універсальним рішенням для автоматизації виробництва.

Основне призначення систем розливу - забезпечити точне дозування і рівномірне наповнення продуктом, незалежно від його фізичних властивостей. Вибір відповідної системи залежить від ряду факторів, серед яких:

1. В'язкість рідини (від водянистих розчинів до густих паст і гелів).
2. Наявність твердих включень (наприклад, соуси зі шматочками овочів або креми з частинками трав).
3. Точність дозування (фармацевтична промисловість вимагає високоточного дозування).
4. Продуктивність (високошвидкісне виробництво вимагає безперервного процесу розливу).
5. Гігієнічні вимоги (у медичній та харчовій промисловості системи повинні відповідати санітарним нормам).

За принципом роботи системи розливу можна класифікувати на декілька різних груп зі своїми перевагами та недоліками.

1.1.1 Аналіз автоматичних системи розливу

Автоматичні системи розливу рідини — це механізовані пристрої, що забезпечують точне дозування та видачу рідини в тару без участі людини або з мінімальним втручанням. Вони широко використовуються в харчовій, фармацевтичній, косметичній та хімічній промисловості.

Завдяки своїй ефективності системи автоматичного розливу підвищують продуктивність підприємств, покращують якість продукції та відповідають сучасним вимогам безпеки та гігієни.

Прикладом може слугувати поршневі системи дозування рідини, що працюють на основі поршневого механізму, який здійснює зворотно-поступальний рух всередині циліндра, забезпечуючи точне регулювання подачі рідини. При прямому русі поршня створюється необхідний тиск, який змушує рідину виходити через дозувальний отвір, а при зворотному русі забирається нова порція рідини. Завдяки такому принципу роботи забезпечується висока точність дозування і можлива можливість роботи з рідинами різної в'язкості.

Ці системи можуть бути як механічними, так і автоматизованими. Механічні варіанти зазвичай використовуються на невеликих виробництвах або в лабораторних умовах, де потрібне ручне управління процесом дозування. Автоматизовані поршневі системи інтегровані в промислові лінії і можуть працювати з високою продуктивністю, забезпечуючи безперервне наповнення великих об'ємів рідин.

Поршневі системи ідеально підходять для роботи з рідинами середньої та високої в'язкості, такими як йогурти, соуси, креми, шампуні, косметика та фармацевтичні препарати. Завдяки точному регулюванню об'єму вони здатні забезпечувати стабільну якість продукції та мінімізують втрати використаного матеріалу (рис. 1.1).

Серед низки переваг необхідно виділити:

1. Висока точність дозування. Оскільки поршень дозує фіксований об'єм рідини, забезпечується висока повторюваність об'єму при кожному циклі дозування.

2. Гнучкість в налаштуванні. Можливість регулювання робочого об'єму поршня дозволяє працювати з різними об'ємами наповнення.

3. Ефективність при роботі з густими рідинами. За рахунок механічного витіснення поршнем рідина без проблем подається навіть при високій в'язкості.

4. Можливість роботи з неоднорідними рідинами. Системи можуть дозувати продукти з дрібними включеннями, наприклад, соуси з шматочками овочів.

Серед недоліків потрібно зазначити:

1. Складність конструкції. Велика кількість рухомих частин вимагає регулярного обслуговування і періодичної заміни поршневих ущільнень.

2. Низька швидкість наповнення. У порівнянні з системами безперервного перекачування або проточними системами, поршневі системи мають обмежену продуктивність.

3. Дорожча експлуатація. Через необхідність заміни поршневих ущільнень витрати на обслуговування можуть бути високими.

4. Можливість забруднення. Якщо система недостатньо очищується, можуть накопичуватися залишки рідини, що може вплинути на якість продукт.



Рисунок 1.1 Поршнева система розливу рідин

Перистальтичні системи наповнення також є представником автоматизованих систем для розливу рідин та працюють на основі гнучких трубок і роликового механізму, який поступово стискає шланг у певних точках, створюючи хвилеподібний рух і штовхаючи рідину вперед. Такий принцип роботи забезпечує рівномірний потік рідини і точне дозування без необхідності використання клапанів або складних насосних механізмів.

Перистальтичні системи можуть бути автоматизованими. Автоматичні моделі дозволяють точно регулювати швидкість і об'єм подачі, що важливо для безперервного виробництва і мінімізації відходів. Завдяки відсутності ризику зворотного потоку і можливості роботи з в'язкими і агресивними речовинами ці системи є одними з найкращим рішенням для промислового розливу рідин.

Такі системи найчастіше використовуються у фармацевтичній промисловості, медицині, лабораторних дослідженнях, а також у харчовій промисловості для дозування рідин, які вимагають високої стерильності(рис. 1.2).

Найголовніші переваги:

1. Повна стерильність. Оскільки рідина контактує лише з трубкою, ризик забруднення мінімальний.
2. Просте обслуговування. Для очищення системи достатньо замінити трубку, що значно спрощує гігієнічні процедури.
3. Можливість роботи з чутливими рідинами. Такий механізм дозволяє дозувати біологічні та фармацевтичні розчини, які не можна піддавати сильному тиску.
4. Гнучкість у дозуванні. Можливість регулювання швидкості стиснення роликів дозволяє адаптувати систему до різних типів рідин.

Головні недоліки:

1. Обмежена швидкість розливу. Через циклічний характер подачі рідини перистальтичні системи не можуть забезпечити високу продуктивність.
2. Обмежений термін служби трубок. Матеріал трубки зношується під дією механічного стискання, тому її потрібно періодично замінювати.

3. Непридатність для густих рідин. Перистальтичний механізм найкраще працює з рідинами низької та середньої в'язкості.

4. Досить висока вартість. Хоча конструкція системи є простою, спеціалізовані трубки можуть бути дорогими.



Рисунок 1.2 Перистальтична система розливу рідин

Враховуючи все, системи розливу рідин є ключовими компонентами автоматизованого виробництва, що дозволяють значно підвищити ефективність та якість готової продукції. Завдяки широкому спектру технологій, компанії можуть вибрати оптимальну систему, яка відповідає їхнім вимогам, забезпечує стерильність, точність та безперервність виробничого процесу. Але їхня громіздкість та ціна обслуговування не дають використовувати дані системи поза підприємствами.

1.1.2 Аналіз неавтоматичних системи розливу

Неавтоматичні системи наповнення рідиною – це пристрої, які вимагають ручного втручання для дозування та доставки рідини. Зазвичай вони простіші за конструкцією і дешевші, ніж автоматизовані системи. Такі системи використовуються в дрібносерійному виробництві, домашньому використанні або для невеликих партій продукції.

До неавтоматичних систем належать гравітаційні дозатори, ручні насосні системи та рідинні диспенсери. Процес наповнення здійснюється вручну або механічним натисканням насоса або дозатора. Хоча точність дозування може бути нижчою, ці системи забезпечують легкість обслуговування та низькі експлуатаційні витрати. Вони часто використовуються для продукції, яка не потребує високого рівня автоматизації або у випадках, коли обсяги виробництва невеликі.

Дозатор рідини складається з контейнера для зберігання рідини, дозувального механізму та вихідного отвору або насадки для подачі рідини. Неавтоматичний дозатор рідини фактично оснащений насосом, який працює за принципом механічного штовхача або важеля. Користувач натискає на важіль або насос, який активує механізм і завдає шкоди до тих пір, поки не вийде певна кількість рідини. У контексті огляду об'єм рідини, що подається, можна регулювати за допомогою спеціальних налаштувань, таких як різні форсунки або регулятори потоку(рис. 1.3).

Не дивлячись на простоту механізму, у подібних систем є багато переваг:

1. Простота в експлуатації: Неавтоматичні диспенсери прості у використанні, не вимагають складних налаштувань або технічного обслуговування. Все, що потрібно – це ручне натискання на дозатор для подачі рідини.

2. Низька вартість: Ці системи дешевші в порівнянні з автоматизованими аналогами, що робить їх доступними для використання в домашніх умовах, малих підприємствах або для дрібних партій продукції.

3. Мобільність та зручність: Неавтоматичні диспенсери зручні у використанні, оскільки вони можуть бути розміщені в будь-якому місці, де є необхідність дозування рідини. Вони не потребують електричного підключення або складних систем управління.

4. Контроль за об'ємом: Користувач сам контролює обсяг рідини, що видається, що може бути корисно в деяких ситуаціях, коли точність дозування не критична.

Разом з великою кількістю переваг є низка недоліків як необхідно вказати:

1. Низька точність дозування: У порівнянні з автоматичними системами, точність дозування у неавтоматичних диспенсерах може бути нижчою, оскільки залежить від сили натискання користувача, що може призводити до незначних помилок у кількості пролитої рідини.

2. Трудомісткість: Процес розливу потребує участі людини, що може бути менш ефективним при великому обсязі виробництва або у випадках, коли потрібно часто дозувати рідину.

3. Обмежена продуктивність: Неавтоматичні диспенсери не можуть працювати на високих швидкостях, що може бути недоліком у виробничих умовах, де потрібна висока продуктивність.

4. Можливість забруднення: При ручному використанні є ризик забруднення рідини, оскільки користувач може торкатися дозатора, що може призвести до контакту з брудними руками чи іншими забруднюючими агентами.

5. Невеликий обсяг розливу: Такі системи обмежені в обсязі рідини, яку можуть подати за один раз, тому вони не підходять для великих виробничих ліній або місць, де потрібен безперервний розлив.



Рисунок 1.3 Приклад неавтоматизованої системи розливу рідин

У заключенні можна сказати, що неавтоматичний рідинний диспенсер є чудовим варіантом для домашнього використання або малих підприємств, де важлива низька вартість і простота у використанні. Проте, для більших обсягів або високих вимог до точності дозування, цей тип диспенсера може бути менш ефективним порівняно з автоматизованими системами.

1.2 Характеристики, технічні можливості мікроконтролерів для розумних систем

Мікроконтролер — це інтегральна схема, що поєднує процесор, пам'ять та периферійні пристрої на одному кристалі. Його можливості залежать від того, як він запрограмований і які функції має виконувати.

Програмування мікроконтролерів зазвичай здійснюється через спеціалізовані середовища, такі як Arduino IDE, STM32Cube, MPLAB для мікроконтролерів PIC або інші. Вони знаходять застосування в багатьох сферах, таких як вбудовані системи, автоматизація, робототехніка, автомобільна промисловість та багато іншого.

Основне призначення мікроконтролера — керування або обробка даних у різних електронних пристроях та системах. Він виконує певні завдання згідно з програмою, що збережена в його пам'яті. Наприклад, у системі розливу рідин мікроконтролер може відповідати за відстеження тари або подачу рідини.

Основні складові мікроконтролера включають:

1. Процесор (Центральний обчислювальний блок, ЦОБ): Здійснює основні обчислення та координує роботу мікроконтролера.

2. Пам'ять: Призначена для зберігання програм і даних. Існують два основних типи пам'яті — оперативна (RAM), що використовується для тимчасових даних, та постійна (ROM, EEPROM, Flash), де зберігаються програми і постійні дані.

3. Периферійні пристрої: Це можуть бути порти вводу/виводу (GPIO), модулі комунікації (SPI, I2C, UART, USB, Ethernet), аналогово-цифрові перетворювачі (ADC), цифро-аналогові перетворювачі (DAC), таймери, контролери переривань та інші.

4. Шина: Це внутрішня система з'єднань, яка дозволяє різним компонентам мікроконтролера обмінюватися даними та працювати разом.

Arduino Uno — це плата на основі мікроконтролера ATmega328P, який є одним із найпопулярніших у сімействі Arduino завдяки своїй доступності та простоті використання, що робить його чудовим вибором для початківців.

Архітектура Uno є відкритою, тобто її схеми та програмне забезпечення вільно доступні для використання та модифікації. Це дає можливість створювати власні версії дошки за умови наявності відповідних знань.

Arduino Uno має 14 цифрових контактів введення/виведення, які можна використовувати як для введення, так і для виведення. Шість з них підтримують широтно-імпульсну модуляцію (ШИМ). Крім того, плата оснащена шістьма аналоговими входами, 16-метровим кристалічним резонатором, USB-портом для програмування та підключення до комп'ютера, виходом живлення, роз'ємом ICSP і кнопкою скидання.

Одним із головних аспектів Arduino Uno є простота програмування. Плата використовує власну мову програмування, засновану на Wiring, і інтегроване середовище розробки (IDE), що дозволяє легко писати код і завантажувати його на плату. Це середовище також містить багато прикладів коду, щоб ви швидко почали працювати(рис. 1.4).

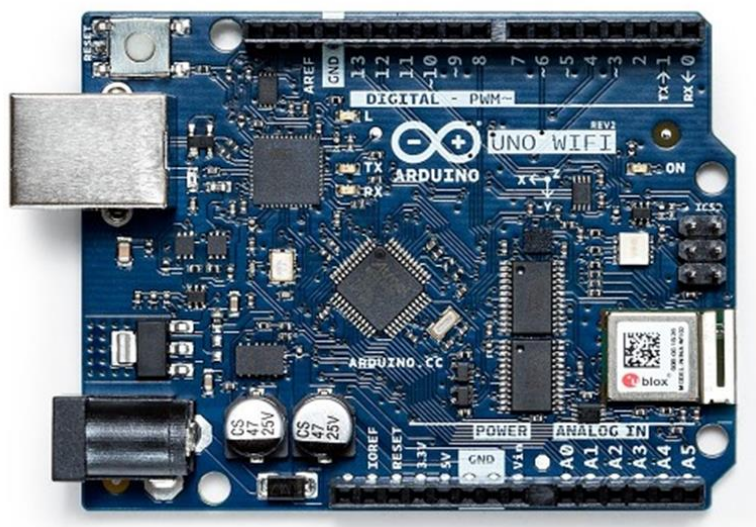


Рисунок 1.4 Мікроконтролер Arduino UNO

Таким чином, Arduino Uno широко використовується для створення різноманітних проектів DIY, таких як роботи, системи автоматизації, музичні інструменти, іграшки, системи безпеки тощо. Його універсальність і простота використання роблять його ідеальним інструментом для ентузіастів електроніки, викладачів, студентів і професіоналів.

Raspberry Pi — це серія одноплатних мікрокомп'ютерів, розроблених Raspberry Pi Foundation у Великобританії. Ці комп'ютери використовуються для навчання інформатики, а також у різноманітних проектах, таких як робототехніка та створення «розумного» будинку.

Остання модель, Raspberry Pi 4, є більш потужною версією серії з низкою функцій, які роблять її ідеальним вибором для розумного будинку. Він оснащений з'єднаннями Ethernet, Wi-Fi, Bluetooth, а також кількома портами USB для підключення різних пристроїв. Однією з ключових особливостей Raspberry Pi 4 є його здатність працювати з іншими операційними системами, такими як Raspbian, Ubuntu і навіть Windows 10 IoT Core. Це дозволяє вибрати операційну систему, яка найкраще підходить для вашого проекту(рис. 1.5).



Рисунок 1.5 Мікроконтролер Raspberry Pi

Raspberry Pi 4 можна використовувати для автоматизації різних систем з розумних пристроїв. Завдяки такому програмному забезпеченню, як OpenHAB, Raspberry Pi 4 може стати головним контролером вашої мережі, координуючи та керуючи всіма підключеними пристроями та системами.

Також варто зазначити, що Raspberry Pi є платформою з відкритим кодом. Це означає, що існує велика спільнота розробників, які активно створюють і діляться ідеями програмного забезпечення та проектів. Це робить Raspberry Pi надзвичайно гнучким інструментом для реалізації різноманітних проектів розумного будинку.

ESP8266 — це високоцінний вбудований мікроконтролер із вбудованим Wi-Fi, розроблений Espressif Systems. Його основною функцією є забезпечення підключення Wi-Fi для різних пристроїв Інтернету речей (IoT) і мікроконтролерів у різних електронних проектах(рис. 1.6).



Рисунок 1.6 Мікроконтролер ESP8266

ESP8266 має вбудований 32-розрядний мікроконтролер, який можна програмувати за допомогою різних інструментів, таких як Arduino IDE і прошивка NodeMCU на основі Lua. Це відкриває великі можливості для розробників. З одного боку, ESP8266 може працювати як модуль Wi-Fi для забезпечення підключення до Інтернету для пристроїв, які не мають вбудованої підтримки Wi-Fi. З іншого боку, він може виступати в якості повноцінного мікроконтролера з Wi-Fi для створення різних пристроїв IoT.

ESP8266 знаходить застосування в різноманітних сферах, таких як розумні домашні пристрої, автоматизовані системи поливу, моніторинг погодних умов, спостереження за навколишнім середовищем та багато інших.

Завдяки своїй здатності забезпечувати з'єднання з Інтернетом або Wi-Fi мережею, ESP8266 є незамінним для проектів, що потребують онлайн-зв'язку. Крім того, цей мікроконтролер має привабливу ціну, що робить його доступним і популярним вибором серед розробників.

1.3 Огляд компонентів пристрою для розливу рідин

Насоси є одними з основних елементів у багатьох технологічних системах, що використовуються для переміщення рідин, газів або сумішей. Вони знаходять застосування в різних галузях, таких як сільське господарство, промисловість,

медичні пристрої, водопостачання, а також у побутових автоматизованих системах. З огляду на широкий спектр застосувань, існує безліч типів насосів, кожен з яких має свої характеристики, переваги та недоліки, що визначають їх ефективність і доцільність у конкретних умовах.

У сучасному світі, зокрема в розробці проектів на платформі Arduino, використання насосів стає важливим для автоматизації різноманітних процесів, таких як полив, охолодження, зрошення, а також для створення систем контролю за середовищем, де потрібна точність і контроль над потоком рідини. Вибір типу насоса залежить від багатьох факторів, таких як тип рідини, обсяг перекачуваного середовища, необхідний тиск, а також специфіка конкретної задачі.

Кожен тип насоса має свої специфікації і особливості роботи, що дозволяє використовувати їх у різних проектах з урахуванням вимог до ефективності, точності та зручності в експлуатації:

1.Мембранний насос використовує еластичну мембрану для перекачування рідини. Коли мембрана деформується, вона створює вакуум, що призводить до всмоктування рідини, а потім виштовхує її через вихідний отвір. Цей тип насоса підходить для роботи з агресивними, хімічно активними або в'язкими рідинами, оскільки мембрана запобігає прямому контакту рідини з механічними частинами насоса.

2.Шестерневий насос використовує два обертових шестерні для перекачування рідини. Цей насос може забезпечити вищий тиск і більш стабільний потік рідини, ніж інші типи насосів. Шестерневі насоси часто використовуються для більш складних завдань, таких як перекачування масла, пального або інших в'язких рідин.

3.Мініатюрний водяний насос - це компактний насос, який працює від постійного струму. Він зазвичай використовується для перекачування невеликої кількості рідини, наприклад, води, для проектів на Arduino. Такий насос можна підключити до Arduino через транзистор або реле для керування включенням і

вимиканням насоса. Це зручний варіант для таких проектів, як автоматизація поливу, охолодження або невеликі акваріуми.

Також одним з важливих елементів пристроїв для розливу води є датчик, що буде подавати сигнал системі для активації насосу. Кожен тип датчика має свої характеристики і застосування. Ось кілька основних типів датчиків, що можуть бути використані для збору даних та взаємодії з Arduino:

1. Датчики руху реагують на рух об'єктів у їхньому полі зору. Вони широко використовуються в системах безпеки, автоматизації будинків, а також для створення інтерактивних проектів.

2. Датчики рівня використовуються для визначення рівня рідин у резервуарах або ємностях. Вони можуть бути як аналоговими, так і цифровими.

3. Кнопка є одним із найпростіших, але водночас важливих компонентів для взаємодії з системою. Вона використовується для передачі сигналу до мікроконтролера, що дозволяє запускати або зупиняти певні процеси, здійснювати увімкнення або вимкнення функцій. Кнопки можуть бути механічними або тактильними.

Висновки до розділу 1

У цьому розділі були розглянуті та проаналізовані автоматичні та неавтоматичні системи для розливу рідин. Було розглянуто фактори для їх класифікації, типи систем, їхні переваги та недоліки та важливі елементи для їхньої роботи.

Поршнева та Перистальтична системи були ретельно проаналізовані як приклади автоматизованих механізмів для роботи з великими об'ємами рідин у промислових масштабах. Не дивлячись на різні переваги чи недоліки, спільною

метою є автоматизація процесу та підвищення точності дозування. Також було розглянуто неавтоматичні системи розливу рідин, що мають свою низку сильних та слабких сторін. Метою цих систем було спрощення процесу розливу рідин на малих підприємствах чи у домашніх умовах з простим використанням та низькою вартістю обслуговування.

Додатково було переглянуто різні мікроконтролери, які є центром систем, типи насосів та датчиків для активації процесу перекачування рідини.

Отже, огляд цих систем і компонентів є важливим для розуміння потенціалу пристроїв для розливу рідин та вибору найкращих рішень для конкретних потреб користувачів. Розвиток технологій продовжує відкривати нові можливості для спрощення процесів у повсякденному житті, і цей огляд є важливим кроком для розуміння та використання цих можливостей.

2 ВИБІР ІНСТРУМЕНТІВ ДЛЯ СИСТЕМИ "РОЗУМНИЙ АВТОМАТ" НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO

2.1 Огляд основних компонентів системи

2.1.1 Виділення переваг Arduino Nano

Для виконання дипломної роботи буде використовуватися плата Arduino Nano. Arduino Nano – це компактна мікроконтролерна плата, яка базується на ATmega328P (або ATmega168 у старших версіях). Вона є зменшеною версією Arduino Uno і розроблена для використання в проектах з обмеженим простором.

Платформа повністю сумісна з середовищем програмування Arduino IDE, що дозволяє використовувати просту мову програмування на основі C/C++. Також дане середовище підтримує багато необхідних стандартних функцій. Це середовище дозволяє швидко і ефективно створювати і завантажувати скетчі в мікроконтролер, спрощуючи процес розробки навіть для початківців.

Завдяки широкій екосистемі бібліотек та інструментів розробники можуть легко реалізувати складні функції, такі як робота датчиків, керування двигунами, бездротовий зв'язок тощо. Arduino Nano підтримує підключення численних модулів і розширень, включаючи Wi-Fi, Bluetooth, RFID, дисплеї, релейні модулі та інші периферійні пристрої. Це робить її ідеальним рішенням для автоматизації, IoT, робототехніки, інтелектуальних систем та інших інженерних проектів.

Arduino Nano має 14 цифрових входів/виходів, з яких 6 можуть працювати в режимі ШІМ-виходів, а також 8 аналогових входів. Плата оснащена кварцевим резонатором на 16 МГц, інтерфейсами I2C, SPI, UART та підтримує програмування через USB (mini-USB або micro-USB залежно від версії). На платі

є вбудований стабілізатор напруги, що дозволяє жити її від джерела з напругою 6-12 вольт з допомогою виводів Vin або використати живлення від USB(рис. 2.1).

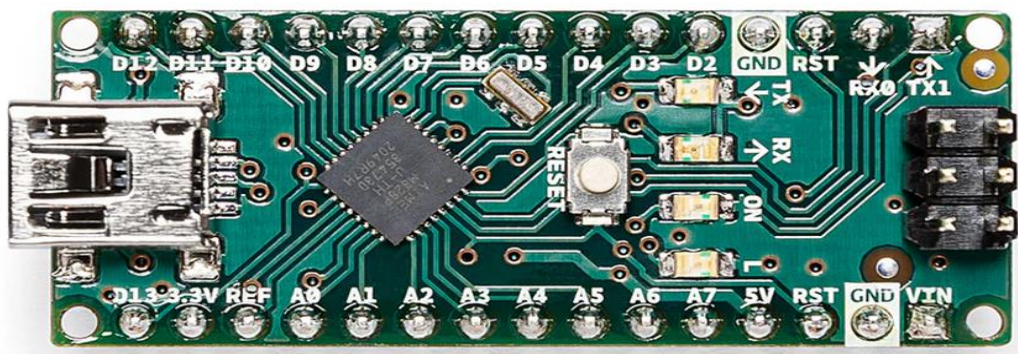


Рисунок 2.1 Плата Arduino NANO

Завдяки невеликим розмірам та енергоефективності вона широко використовується в робототехніці, системах автоматизації побутових і промислових процесів, розумних будинках, пристроях Інтернету речей (IoT), сенсорних системах, а також в освітніх цілях.

Зокрема, в робототехніці Arduino Nano часто використовується для керування рухом двигунів, зчитування даних з датчиків відстані, кута та освітленості, а також для реалізації простих алгоритмів автономного руху. У системах автоматизації мікроконтролер може керувати освітленням, температурою, вентиляцією або, як у цьому проекті, виконувати точне дозування рідини за допомогою насоса та відповідних датчиків. Вона повністю сумісна з Arduino IDE та підтримує функції `pinMode()`, `digitalWrite()`, `digitalRead()` для роботи з цифровими виводами(рис2.2).

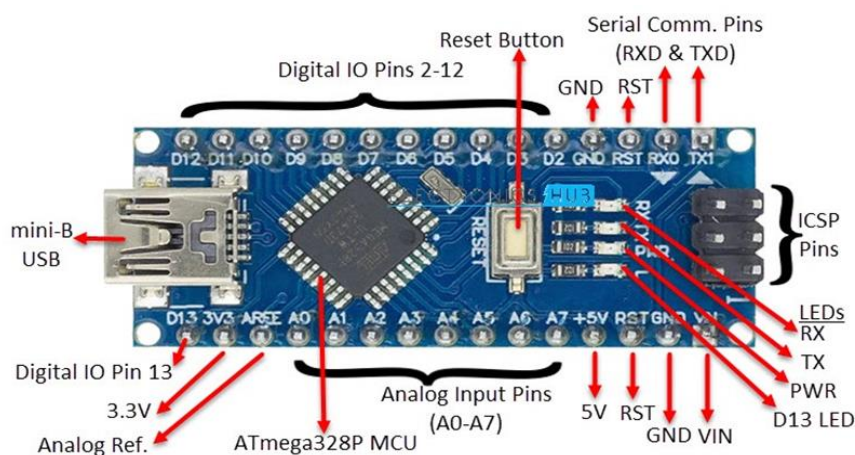


Рисунок 2.2 Виводи Arduino NANO

Обсяг флеш-пам'яті ATmega328 становить 32 КБ (з яких 0.5 КБ використовуються завантажувачем). Мікроконтролер також має 2 КБ пам'яті SRAM і 1 КБ EEPROM (з якої можна зчитувати або записувати інформацію за допомогою бібліотеки EEPROM).

Оскільки Nano не має власного роз'єму живлення, її можна підключити до джерела через виводи GND і Vin. Не зважаючи на це, Arduino Nano – чудовий вибір для розробників, які потребують функціональності Arduino Uno, але в компактному форм-факторі.

2.1.2 Опис насосу для рідин та повітря

Мембранний насос для води та повітря — це електромеханічний пристрій, який використовується для перекачування рідин або газів за допомогою спеціальної мембрани, що коливається під дією електродвигуна. Такий насос працює за принципом створення різниці тиску в камері, що забезпечує всмоктування та виштовхування рідини або повітря.

Основний принцип роботи полягає у зворотно-поступальному русі мембрани, який створює вакуумний ефект. Це дозволяє насосу перекачувати рідини без потреби у великих механічних елементах, як у відцентрових або поршневих насосах.

Даний насос працює від постійної напруги 12В, що робить його енергоефективним і зручним для використання у системах, що живляться від акумуляторів, блоків живлення або сонячних панелей. Він може застосовуватися в різних сферах, включаючи автомобільні системи, акваріуми, системи зрошення, автоматизацію, охолодження, водопостачання та медичне обладнання. Споживана потужність залежить від конкретної моделі та може змінюватися в межах 10-60 Вт, що дозволяє вибрати оптимальний варіант для конкретного застосування.

Продуктивність насоса може варіюватися від 1 до 10 літрів на хвилину, що визначає його здатність ефективно перекачувати рідину або газ у відповідних

умовах експлуатації. Він забезпечує робочий тиск до 3-8 бар, що є достатнім для більшості завдань, зокрема для систем водопостачання, охолодження, зрошення та дозування рідин у промисловості.

Цей тип насоса підтримує роботу як із рідинами, так і з газами, що розширює сферу його застосування. Зокрема, він може перекачувати воду, легкі хімічні розчини, повітря та використовуватися в вакуумних системах.

Конструкція мембрани відіграє важливу роль у довговічності та стійкості до різних середовищ. Вона виготовляється з силікону, гуми або тефлону, що забезпечує високу хімічну стійкість, зносостійкість і надійність навіть у складних умовах експлуатації. Це робить мембранний насос ідеальним вибором для роботи з агресивними або специфічними рідинами, які можуть руйнувати інші типи насосів(Рис. 2.3).



Рисунок 2.3 мембранний насос 12В

Таким чином, мембранний насос - це надійний, компактний і універсальний пристрій, який добре підходить для різних побутових і промислових застосувань.

Завдяки своїй автономності, ефективності та низькому енергоспоживанню він є ідеальним рішенням для перекачування води, хімічних розчинів, повітря або вакуумних систем. Його конструкція дозволяє працювати у важкодоступних місцях, а можливість підключення до автомобільних, сонячних або акумуляторних систем робить його ще більш гнучким у використанні. Висока стійкість матеріалів мембрани забезпечує тривалу роботу навіть у складних

умовах, що робить цей насос незамінним в автономному водопостачанні, зрошенні, охолодженні, дозуванні рідини та багатьох інших сферах застосування.

2.1.3 Огляд характеристик драйвера двигуна MX1508

MX1508 - це компактний і доступний двоканальний драйвер двигуна, який використовується для керування двигунами постійного струму і невеликими кроковими двигунами. Його основне призначення - спростити керування двигунами в мікроконтролерних системах, таких як Arduino, ESP8266, Raspberry Pi та інших.

Драйвер працює на основі Н-мостової схеми, яка дозволяє реверсувати (змінювати напрямок обертання) і регулювати швидкість двигунів за допомогою широтно-імпульсної модуляції (ШИМ). Завдяки вбудованим MOSFET-транзисторам модуль має високий коефіцієнт корисної дії і може ефективно працювати з двигунами малої та середньої потужності.

MX1508 підтримує діапазон робочих напруг від 2 до 10 В, що робить його ідеальним для використання з малопотужними двигунами 3, 5 і 9 В. Максимальний струм навантаження на канал становить до 1,5 А, а короточасні пікові навантаження можуть досягати 2 А (Рис. 2.4).

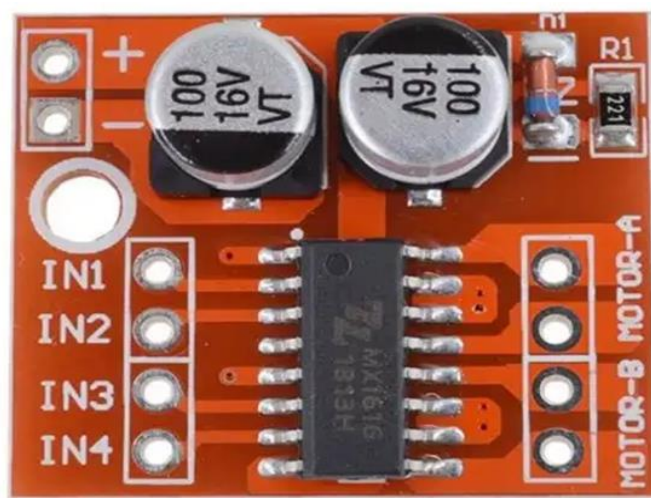


Рисунок 2.4 драйвери двигуна MX1508

Модуль має 4 входи керування, що дозволяє керувати двома двигунами одночасно. Для керування кожним двигуном використовується 2 входи: сигнал на один з них змушує двигун обертатися в одному напрямку, а на інший - у протилежному. За допомогою ШІМ-сигналу можна регулювати швидкість обертання.

Завдяки компактним розмірам MX1508 можна легко інтегрувати в різні проекти: транспортні моделі, автоматизовані механізми та системи «розумного будинку».

Драйвер MX1508 — це чудовий вибір для керування маломощними двигунами, особливо в навчальних або дослідницьких проєктах. Його простота, компактність та доступність роблять його ідеальним невеликих та простих систем.

2.1.4 Переваги перемикача з виступом

Перемикач Snap-Action Switch або перемикач з виступом 18,5 мм - це компактний механічний перемикач, який широко використовується для керування електричними ланцюгами в автоматизованих системах, електроніці та механічних пристроях. Його конструкція передбачає миттєве перемикання контактів при досягненні певного положення, що забезпечує високу точність і надійність роботи (Рис. 2.5).

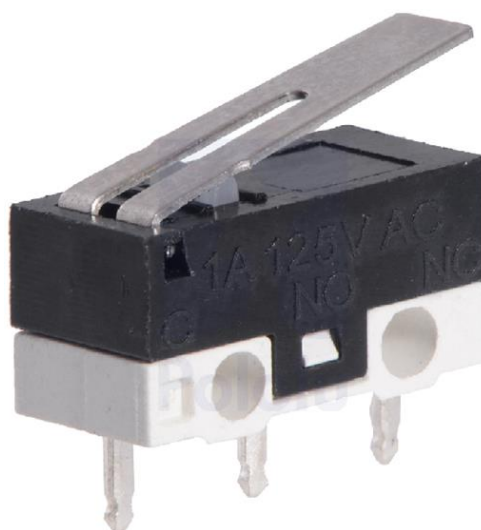


Рисунок 2.5 перемикач з виступом 18,5 мм

Головною особливістю цього вимикача є спеціальний виступ (важіль) довжиною 18,5 мм, який виконує роль чутливого елемента, що реагує на механічний вплив. Це дозволяє використовувати його в пристроях, де необхідно точно контролювати рух або положення деталей.

Перемикач Snap-Action працює за принципом миттєвого замикання або розмикання контактів, коли на нього діє зовнішня сила (наприклад, тиск або механічний вплив). Виступ 18,5 мм забезпечує чутливу реакцію на зміну положення механізму, завдяки чому вимикач може активувати або деактивувати електричний ланцюг з мінімальною затримкою.

Коли навантаження знімається, механізм повертається у вихідне положення під дією пружини або магнітної сили, вбудованої в конструкцію елемента. В результаті контакти відновлюють свій початковий стан, замикаючись або розмикаючись відповідно до конструкції пристрою. Такий принцип роботи особливо важливий у системах, які потребують швидкого реагування та високої повторюваності перемикачів.

Таким чином, перемикач з виступом - це ефективний і компактний механічний вимикач, який забезпечує точне спрацювання в автоматизованих і електронних системах. Завдяки своїй високій чутливості та швидкості, він широко використовується в різних системах і побутових пристроях. Хоча він має певні обмеження по навантаженню, його надійність, простота підключення і довговічність роблять його чудовим вибором для різноманітних застосувань автоматизації.

2.2 Огляд додаткових компонентів системи

2.2.1 Переваги використання micro USB на DIP 2.54

Роз'єм micro USB є важливим комунікаційним елементом для Arduino Nano в системі дозування рідини. Він забезпечить можливість заряджати або передавати живлення між пристроями, а також зв'язок між мікроконтролерами, датчиками та іншими компонентами, які будуть встановлені в пристрої.

Роз'єм micro USB on DIP 2.54 - це стандартний роз'єм micro USB, встановлений на монтажній платі з кроком контактів 2,54 мм (100 міліметрів), що відповідає стандартному розташуванню контактів на макетних і друкованих платах (рис. 2.6).

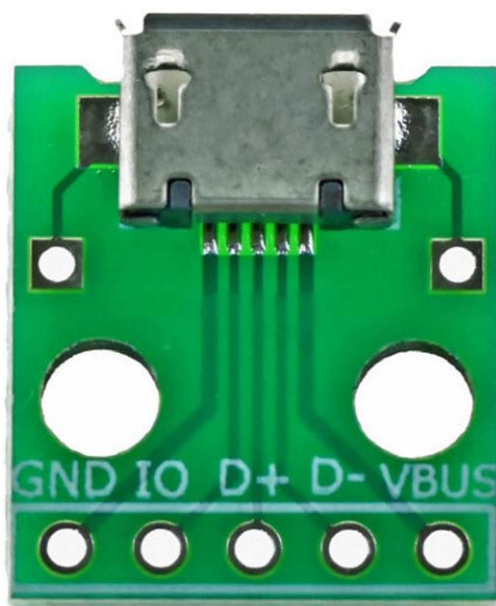


Рисунок 2.6 micro USB на DIP 2.54

Серед переваг цього роз'єму можна виділити наступні:

1. Простота використання
2. Сумісність зі стандартною розводкою
3. Підтримує живлення і передачу даних
4. Довговічність і надійність
5. Компактність.

Роз'єм micro USB на DIP 2.54 є практичним і зручним рішенням для макетування, розробки пристроїв та інтеграції micro USB у проекти на Arduino та інші мікроконтролери. Він дозволяє легко підключати живлення та передавати дані, проте через свою механічну вразливість та застарілість стандарту, поступове

витіснення USB Type-C його використання може бути обмеженим у довгостроковій перспективі.

2.2.2 Аналіз модуля дисплея TM1637

4-розрядний модуль дисплея TM1637 - це компактний і зручний семисегментний світлодіодний індикатор, який широко використовується в електронних проектах і системах відображення інформації. Він складається з чотирьох семисегментних індикаторів і може відображати цифрові значення, години, температуру, таймери та інші показники(рис. 2.7).



Рисунок 2.7 4-розрядний Модуль дисплея TM1637

Модуль базується на мікросхемі TM1637, яка виконує функції контролера дисплея. Це дозволяє спростити керування індикатором через двопровідний інтерфейс, що робить його ідеальним для використання з Arduino, ESP8266, ESP32, Raspberry Pi та іншими мікроконтролерами.

Завдяки мінімальній кількості необхідних проводів, дисплей легко інтегрується в проекти без зайвої складності схеми. Він широко використовується в годинниках, вимірювальних приладах, таймерах, лічильниках, системах контролю температури та розумному будинку.

TM1637 працює через двопровідний інтерфейс, що значно спрощує підключення порівняно з традиційним керуванням семисегментним дисплеєм через окремі виводи GPIO.

Модуль підтримує динамічне керування яскравістю, що дозволяє регулювати рівень підсвічування сегментів відповідно до умов роботи. Доступна готова бібліотека для використання з Arduino та іншими контролерами, що дозволяє легко відображати цифри, символи, керувати яскравістю та анімацією.

Висновки до розділу 2

У цьому розділі було проведено детальний огляд компонентів системи. В ході огляду були проаналізовані та обрані основні та додаткові компоненти. Кожен з розглянутих елементів необхідний для реалізації системи «розливу рідини» та її базового функціоналу.

Розгляд Arduino Nano розпочався з вивчення її архітектури, функціональності та способів підключення. Було зазначено, що ця компактна плата є однією з найпопулярніших у своїй сфері використання. Завдяки невеликим розмірам та широкій сумісності з різноманітними модулями та датчиками, вона ідеально підходить для розробки компактних проектів.

Наступним об'єктом аналізу став мембранний насос 12В, який використовується для перекачування рідин та газів. Було детально розглянуто його конструкцію, принцип роботи, матеріали мембрани, продуктивність та робочий тиск. Крім того, були визначені його основні переваги, серед яких енергоефективність, довговічність і здатність працювати безперервно протягом тривалого часу.

Далі ми проаналізували драйвер двигуна MX1508, який використовується для керування низьковольтними двигунами постійного струму. Були розглянуті його технічні характеристики, підтримка двонаправленого керування та можливість регулювання швидкості за допомогою ШІМ-сигналу. До його переваг можна віднести простоту використання, компактність і низьку вартість, але обмежена максимальна вихідна потужність може бути недоліком для деяких застосувань.

Особливу увагу було приділено Snap-Action Switch - механічному вимикачу з виступом 18,5 мм, який використовується в промислових системах, системах автоматизації та робототехніці. Було розглянуто його принцип роботи, чітке спрацьовування при досягненні певного механічного навантаження та довговічність.

Після аналізу перемикача було розглянуто роз'єм micro USB на DIP 2.54, який використовується для підключення живлення та обміну даними між мікроконтролерами та іншими пристроями. Особливу увагу було приділено його перевагам та недолікам. Було визначено, що цей роз'єм забезпечує надійне з'єднання, компактність та високу популярність в сучасній електроніці, але може мати обмежений термін служби через часті механічні впливи. Незважаючи на свої недоліки, цей micro USB є найкращим варіантом для цього проекту.

Останнім елементом аналізу став 4-розрядний семисегментний дисплей TM1637, який широко використовується для відображення цифрових значень, часу, лічильників та інших параметрів. Було розглянуто принцип роботи контролера TM1637, особливості двопровідного інтерфейсу та можливості регулювання яскравості. До його переваг можна віднести простоту підключення, низьке енергоспоживання, наявність готових бібліотек для Arduino та ESP.

Проведений аналіз дозволив оцінити широкий спектр електронних компонентів, кожен з яких виконує важливу функцію при створенні та реалізації системи. Кожен з проаналізованих компонентів має свої переваги та обмеження, які необхідно враховувати при проектуванні.

3 РОЗРОБКА СИСТЕМИ "РОЗУМНИЙ РІДИН" НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO

3.1 Моделювання системи "розливу рідин" на основі мікроконтролера Arduino

Моделювання системи "розумний автомат для розливу води" було здійснено у програмному середовищі Wokwi.

Wokwi – це сучасне онлайн-середовище для розробки, моделювання та тестування проєктів, що базуються на мікроконтролерах а також на інших електронних компонентах. Дана платформа дозволяє створювати та тестувати схеми та надає можливість перевірити свою ідею перед реалізацією.

Головною особливістю Wokwi є можливість програмувати мікроконтролери у веб-браузері без встановлення додаткового програмного забезпечення. Wokwi є потужним онлайн-інструментом для моделювання схем, програмування та тестування проєктів на базі мікроконтролерів. Завдяки своїй доступності, простоті та гнучкості ця платформа є ідеальним варіантом. Незважаючи на певні обмеження, Wokwi дозволяє швидко та ефективно створити прототип і тестувати код у віртуальному середовищі.

На даному етапі було створено модель "розумного автомату", яка включає в себе мікроконтролер Arduino Nano та декількох компонентів – перемикача з виступом, модуля MX1508, мембранного насосу 12В та сервоприводу.

Перемикач з виступом (рис. 3.1) використовується для подачі сигналу на мікроконтролер про необхідність почати роботу мембранного насосу а також вказати місце до котрого сервоприводу з трубкою необхідно зробити оберт.

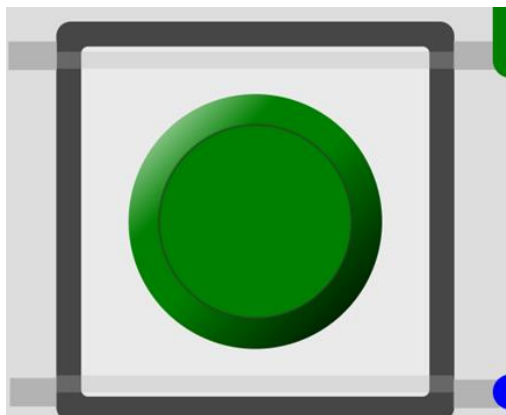


Рисунок 3.1 перемикач з виступом у середовищі Wokwi

Мікроконтролер Arduino Uno відповідає на запит від перемикача та приводить у рух сервопривод(рис. 3.2), який необхідний зворот, тим самим відправляючи трубку від мембранного насосу до тари що необхідно заповнити.

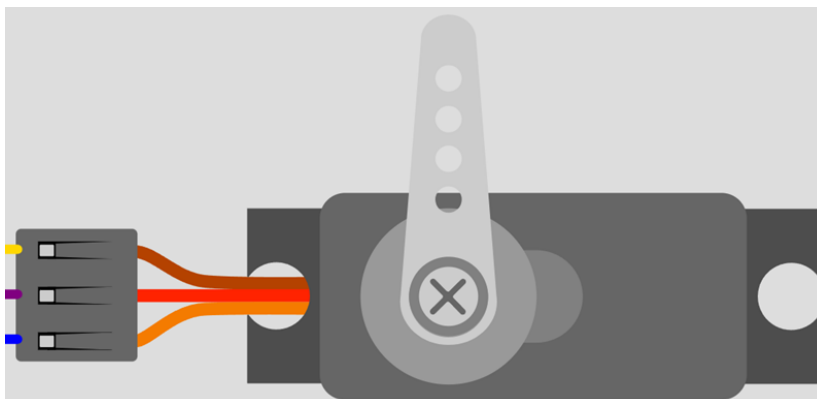


Рисунок 3.2 сервопривод у середовищі Wokwi

Вслід за цим, мікроконтролер подає сигнали на модуль MX1508(рис 3.3), що є компакним драйвером для керування малопотужними колекторними двигунами. Тобто за допомогою даного драйвера мікроконтролер може керувати насосом що дає можливість виконувати головну функцію створюваної системи.

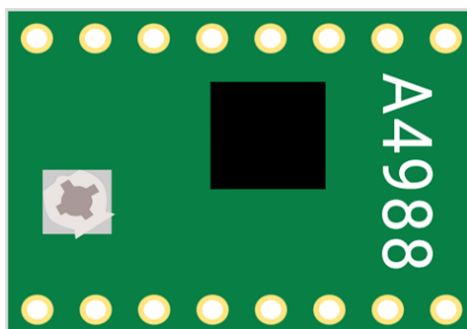


Рисунок 3.3 драйвер для двигунів у середовищі Wokwi

На завершальному етапі роботи система надсилає сигнал з Arduino на мембранний насос (рис. 3.4), який створює тиск і закачує рідину в контейнер через правильно розміщену трубку, запобігаючи протіканню.

Такий механізм дозволяє автоматизувати систему подачі рідини, роблячи її ефективною, надійною та зручною для різних сфер використання.

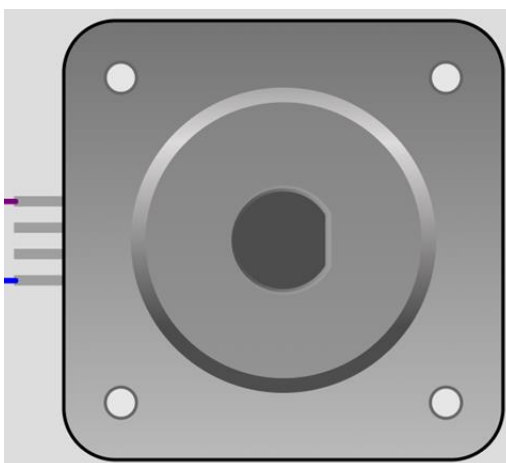


Рисунок 3.4 насос у середовищі Wokwi

3.2 Розробка програмного забезпечення для Arduino IDE

Схема системи "розумний автомат для розливу рідин" була розроблена з урахуванням необхідності забезпечення функціональної безпеки, зручності користувача та максимальної ефективності роботи системи.

Всі частини системи були ретельно протестовані та налагоджені під час моделювання, що забезпечило їх безперебійну роботу в реальних умовах.

Розробка програмного забезпечення для проекту розумного будинку здійснювалася за допомогою Arduino IDE - відкритого середовища розробки, що дозволяє створювати програми для мікроконтролера Arduino (рис. 3.5).

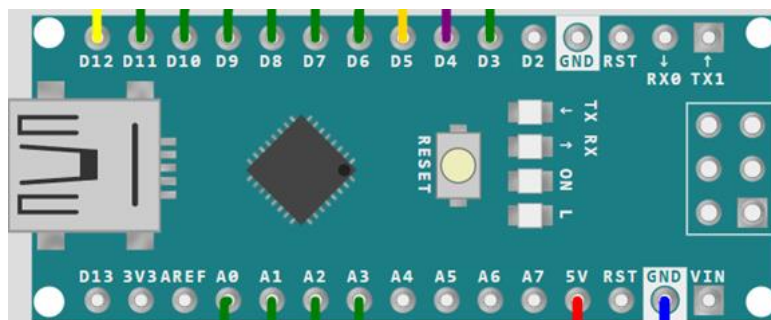


Рисунок 3.5 мікроконтролер Arduino Nano у середовищі Wokwi

Було створено код, що реалізує автоматизовану систему розливу рідини на базі мікроконтролера Arduino. Він визначає основні параметри роботи, зокрема кількість склянок, час наповнення 50 мл рідини, рівень спрацьовування датчиків, можливість інвертування сервоприводу. Він також забезпечує систему підтримки живлення, яка запобігає переходу пристрою в сплячий режим.

Загалом, цей код забезпечує повністю автоматизований процес дозування рідини, гарантуючи точність і стабільність роботи всіх компонентів. Його можна використовувати в різних сферах, зокрема в харчовій промисловості, барах або домашніх автоматизованих пристроях для дозування рідини.

3.3 Опис програмного коду Arduino Nano

Код написано на мові програмування C++, який використовується для створення проектів на базі Arduino - відкритої платформи для вбудованих систем. Ось що робить кожен блок коду.

Лістинг коду:

```
#define NUM_SHOTS
#define TIMEOUT_OFF
#define SWITCH_LEVEL
#define INVERSE_SERVO
const byte shotPos[] = {25, 55, 80, 100, 130, 155};
const long time50ml = 6200;
```

```
#define KEEP_POWER
#define DEBUG_UART 1
#define PUMP_POWER 3
#define SERVO_POWER 4
#define SERVO_PIN 5
#define LED_PIN 6
#define BTN_PIN 7
#define ENC_SW 8
#define ENC_DT 9
#define ENC_CLK 10
#define DISP_DIO 11
#define DISP_CLK 12
const byte SW_pins[] = {A0, A1, A2, A3, A4, A5};
```

Ця частина коду містить оголошення макросів, констант і змінних, які визначають основні параметри системи. Макроси NUM_SHOTS, TIMEOUT_OFF, SWITCH_LEVEL, INVERSE_SERVO задають ключові налаштування, зокрема кількість порцій, час очікування, рівень сигналу для перемикавання та інверсію сервоприводу. Масив shotPos[] зберігає положення сервоприводу для кожної склянки. Константа time50ml задає час, необхідний для наливання 50 мл рідини.

Додатково визначено константу KEEP_POWER, яка відповідає за підтримку живлення, щоб система не вимкнулася. Макрос DEBUG_UART активує режим налагодження через послідовний монітор.

У коді також задаються номери виводів для підключення основних компонентів: PUMP_POWER для насоса, SERVO_POWER і SERVO_PIN для сервоприводу, LED_PIN для світлодіода, BTN_PIN для кнопки, ENC_SW, ENC_DT, ENC_CLK для енкодера і DISP_DIO, DISP_CLK для дисплея. Масив SW_pins[] містить номери аналогових виводів, до яких можна підключити додаткові датчики або перемикачі.

Лістинг коду:

```
#include <GyverTM1637.h>
#include <ServoSmooth.h>
#include <microLED.h>
#include <EEPROM.h>
#include "encUniversalMinim.h"
#include "buttonMinim.h"
#include "timer2Minim.h"
```

Цей фрагмент коду містить підключення бібліотек, необхідних для роботи з різними компонентами системи. GyverTM1637.h використовується для керування 4-розрядним семисегментним дисплеєм TM1637. ServoSmooth.h забезпечує плавне керування сервоприводом, покращуючи його рух. microLED.h відповідає за керування окремими світлодіодами. EEPROM.h дозволяє зберігати і зчитувати дані у вбудованій енергонезалежній пам'яті мікроконтролера.

Також можуть бути підключені бібліотеки користувача: encUniversalMinim.h для роботи з енкодером, buttonMinim.h для обробки натискань кнопок і timer2Minim.h для управління таймерами, які дозволяють точно відстежувати час виконання певних операцій.

Лістинг коду:

```
#define COLOR_DEBTH 2 LEDdata leds[NUM_SHOTS];
microLED strip(leds, NUM_SHOTS, LED_PIN);
GyverTM1637 disp(DISP_CLK, DISP_DIO);
```

Цей код ініціалізує змінні та об'єкти, необхідні для роботи зі світлодіодами та семисегментним дисплеєм.

Змінна COLOR_DEBTH визначає глибину кольору для світлодіодної стрічки, яка може впливати на яскравість і обсяг використовуваної пам'яті. Масив

leds[NUM_SHOTS] створює буфер для даних світлодіодів, де NUM_SHOTS - кількість світлодіодів або зон підсвічування. Об'єкт `microLED strip(leds, NUM_SHOTS, LED_PIN)` відповідає за керування світлодіодною стрічкою, використовуючи вказаний масив та `pin` для підключення.

Крім того, об'єкт `GyverTM1637 disp(DISP_CLK, DISP_DIO)` створено для керування 4-бітним семисегментним дисплеєм TM1637, підключеним до виводів `DISP_CLK` та `DISP_DIO`.

Лістинг коду:

```
#define servoON() digitalWrite(SERVO_POWER, 1)
#define servoOFF() digitalWrite(SERVO_POWER, 0)
#define pumpON() digitalWrite(PUMP_POWER, 1)
#define pumpOFF() digitalWrite(PUMP_POWER, 0)
#if (DEBUG_UART == 1)
#define DEBUG(x) Serial.println(x)
#else
#define DEBUG(x)
#endif
```

Цей фрагмент коду містить макроси, які спрощують керування основними компонентами системи, такими як сервопривід, насос та функція налагодження.

Макроси `servoON()` і `servoOFF()` відповідають за увімкнення та вимкнення сервоприводу шляхом встановлення відповідного рівня сигналу на виводі `SERVO_POWER`. Аналогічно, макроси `pumpON()` і `pumpOFF()` керують роботою насоса, вмикаючи або вимикаючи живлення на піні `PUMP_POWER`.

Додатково передбачено макрос `DEBUG(x)`, який використовується для виведення налагоджувальних повідомлень на серійну консоль. Якщо параметр `DEBUG_UART` встановлено у 1, повідомлення буде виведено на послідовний монітор за допомогою `Serial.println(x)`. Якщо налагодження вимкнено

(DEBUG_UART != 1), то макрос DEBUG(x) не виконує жодних дій, що допомагає оптимізувати програму, коли налагодження не потрібне.

Лістинг коду:

```
void setup() {  
  #if (DEBUG_UART == 1)  
    Serial.begin(9600);  
    DEBUG("start");  
  #endif  
  if (EEPROM.read(1000) != 10) {  
    EEPROM.write(1000, 10);  
    EEPROM.put(0, thisVolume);  
  }  
  EEPROM.get(0, thisVolume);  
  strip.setBrightness(130);  
  strip.clear();  
  strip.show();  
  DEBUG("strip init");  
  pinMode(PUMP_POWER, 1);  
  pinMode(SERVO_POWER, 1);  
  for (byte i = 0; i < NUM_SHOTS; i++) {  
    if (SWITCH_LEVEL == 0) pinMode(SW_pins[i], INPUT_PULLUP);  
  }  
}
```

Ця функція setup() ініціалізує систему при запуску мікроконтролера.

Спочатку, якщо увімкнено налагодження (DEBUG_UART == 1), відкривається послідовний порт зі швидкістю 9600 бод і виводиться повідомлення про запуск. Далі відбувається звернення до EEPROM: якщо за адресою 1000

знайдено значення, відмінне від 10, воно записується разом з початковим об'ємом рідини (thisVolume), який згодом зчитується назад.

Після цього відбувається ініціалізація світлодіодів, встановлення її яскравості, очищення буфера і виведення тестового повідомлення. Далі налаштовуються режими піноутворення: вмикається насос і сервопривід, і якщо логічний рівень перемикачів SWITCH_LEVEL дорівнює нулю, то вони налаштовуються як INPUT_PULLUP для роботи з підтягуючими резисторами.

Дисплей TM1637 очищується, встановлюється його яскравість, а потім генерується налагоджувальне повідомлення про успішну ініціалізацію.

Лістинг коду:

```
disp.clear();
disp.brightness(7);
DEBUG("disp init");
servoON();
servo.attach(SERVO_PIN, 600, 2400);
if (INVERSE_SERVO) servo.setDirection(REVERSE);
servo.write(0);
delay(800);
servo.setTargetDeg(0);
servo.setSpeed(60);
servo.setAccel(0.3);
servoOFF();
serviceMode();
dispMode();
timeoutReset();
TIMEOUTtimer.start();
}
```

Наступний крок - налаштування сервоприводу. Його під'єднують до певного виводу, встановлюють кінцеві положення (600-2400 мікросекунд), а якщо встановлено INVERSE_SERVO, то змінюють напрямок обертання. Далі привід переміщується в початкове положення, встановлюється швидкість і прискорення, а потім вимикається живлення сервоприводу.

Наостанок викликаються функції serviceMode() (калібрування системи), dispMode() (виведення стандартних значень на дисплей) і timeoutReset() (скидання таймера очікування), після чого запускається таймер автоматичного вимкнення TIMEOUTtimer.

Лістинг коду:

```
void loop() {  
    encTick();  
    btnTick();  
    flowTick();  
    LEDtick();  
    timeoutTick();  
}
```

Функція loop() є основним циклом програми, який виконується безперервно. Вона викликає декілька допоміжних функцій для обробки подій у системі.

Функція encTick() обробляє події, пов'язані з енкодером, зчитує його стан і визначає, чи відбулося обертання або натискання кнопки. btnTick() перевіряє стан кнопки керування, визначаючи, чи була вона натиснута або утримана. flowTick() відстежує процес протікання рідини та її стан в системі. LEDtick() оновлює стан світлодіодної стрічки, змінюючи її колір або ефекти відповідно до роботи пристрою. timeoutTick() відстежує час бездіяльності та вимикає систему у разі перевищення заданого таймера.

Цей цикл гарантує, що всі основні елементи системи знаходяться в постійній обробці, реагуючи на зміни в реальному часі.

Лістинг коду:

```
#pragma pack(push,1)
typedef struct {
    bool holdedFlag: 1;
    bool btnFlag: 1;
    bool pressF: 1;
    bool clickF: 1;
    bool holdF: 1;
}
buttonMinimFlags;
#pragma pack(pop)
class buttonMinim {
```

Цей код реалізує клас `buttonMinim`, який обробляє натискання кнопки, включаючи короткі та довгі натискання.

Лістинг коду:

```
public:
    buttonMinim(uint8_t pin);
    boolean pressed();
    boolean clicked();
    boolean holding();
    boolean holded();
    private:
    buttonMinimFlags flags;
    void tick();
    uint32_t _btnTimer;
    byte _pin;
};
```

Структура `buttonMinimFlags` використовує побітові поля для зберігання станів кнопки, що дозволяє економити пам'ять. У класі визначено чотири методи: `pressed()`, `clicked()`, `holding()` та `holded()`, кожен із яких відповідає за певний тип взаємодії з кнопкою.

Лістинг коду:

```
buttonMinim::buttonMinim(uint8_t pin) {
    pinMode(pin, INPUT_PULLUP);
    _pin = pin;
}

void buttonMinim::tick() {
    boolean btnState = digitalRead(_pin);
    if (!btnState && !flags.btnFlag && ((uint32_t)millis() - _btnTimer > 90)) {
        flags.btnFlag = true;
        _btnTimer = millis();
        flags.pressF = true;
        flags.holdedFlag = true;
    }
    if (btnState && flags.btnFlag && ((uint32_t)millis() - _btnTimer < 350)) {
        flags.btnFlag = false;
        _btnTimer = millis();
        flags.clickF = true;
        flags.holdF = false;
    }
    if (flags.btnFlag && ((uint32_t)millis() - _btnTimer > 900)) {
        if (!btnState) {
            flags.holdF = true;
        } else {
            flags.btnFlag = false;
        }
    }
}
```

```

        flags.holdF = false;
        _btnTimer = millis();
    }
}
}

```

Конструктор `buttonMinim(uint8_t pin)` ініціалізує кнопку в режимі `INPUT_PULLUP`, що означає, що вона використовується з внутрішнім підтягувальним резистором. Функція `tick()` виконує основну обробку натискань, визначаючи, чи була кнопка натиснута, відпущена або утримувалася протягом певного часу.

Лістинг коду:

```

boolean buttonMinim::pressed() {
    buttonMinim::tick();
    if (flags.pressF) {
        flags.pressF = false;
        return true;
    }
    else return false;
}

boolean buttonMinim::clicked() {
    buttonMinim::tick();
    if (flags.clickF) {
        flags.clickF = false;
        return true;
    }
    else return false;
}

```

Метод `pressed()` повертає `true`, якщо кнопка була просто натиснута. `clicked()` відстежує коротке натискання з подальшим відпусканням. `holding()` визначає, чи кнопка утримується більше 900 мс.

Лістинг коду:

```
boolean buttonMinim::holding() {  
    buttonMinim::tick();  
    if (flags.holdF) {  
        return true;  
    }  
    else return false;  
}  
  
boolean buttonMinim::holded() {  
    buttonMinim::tick();  
    if (flags.holdF && flags.holdedFlag) {  
        flags.holdedFlag = false;  
        return true;  
    }  
    else return false;  
}
```

Метод `holded()` схоже на `holding()`, але додатково запам'ятовує факт утримування та повертає `true` лише один раз за сесію натискання.

Завдяки цьому класу можна легко інтегрувати обробку кнопки в проєкти Arduino, отримуючи різні типи взаємодій із мінімальними затримками.

Лістинг коду:

```
class encMinim  
{
```

```

    public:
        encMinim(uint8_t clk, uint8_t dt, uint8_t sw, boolean dir, boolean type);
        void tick();
        boolean isClick();
        boolean isTurn();
        boolean isRight();
        boolean isLeft();
        boolean isRightH();
        boolean isLeftH();

    private:
        byte _clk, _dt, _sw;
        boolean _type = false;
        boolean _state, _lastState, _turnFlag, _swState, _swFlag, _turnState;
        byte _encState;
        uint32_t _debTimer;
};

```

Цей клас `encMinim` реалізує обробку енкодера з кнопкою для мікроконтролерів типу Arduino.

Також було оголошено наступні методи:

1. `encMinim(...)` - конструктор, що приймає піни енкодера та параметри його роботи.
2. `tick()` - основний метод оновлення стану енкодера та кнопки.
3. `isClick()` - перевіряє, чи була натиснута кнопка енкодера.
4. `isTurn()` - визначає, чи був поворот.
5. `isRight()` та `isLeft()` - визначають, чи був поворот вправо або вліво.
6. `isRightH()` та `isLeftH()` - перевіряють, чи відбулося обертання, якщо була натиснута кнопка.

У приватній частині зберігаються змінні для роботи з енкодером, зокрема налаштування виводу (`_clk`, `_dt`, `_sw`), змінні стану енкодера, прапори обертання та натискання, а також таймер затримки відскоку.

Клас забезпечує ефективну обробку енкодерів на основі швидкості обертання та натискання, що є корисним для навігації по меню або керування параметрами пристрою.

Лістинг коду:

```
#pragma pack(push,1)
typedef struct {
    bool holdedFlag: 1;
    bool btnFlag: 1;
    bool pressF: 1;
    bool clickF: 1;
    bool holdF: 1;
} buttonMinimFlags;
#pragma pack(pop)
class buttonMinim {
public:
    buttonMinim(uint8_t pin);
    boolean pressed();
    boolean clicked();
    boolean holding();
    boolean holded();
```

Цей код реалізує клас `buttonMinim`, який відповідає за обробку кнопок у мікроконтролерних системах.

Лістинг коду:

```
private:
```

```

buttonMinimFlags flags;
void tick();
uint32_t _btnTimer;
byte _pin;
};
buttonMinim::buttonMinim(uint8_t pin) {
    pinMode(pin, INPUT_PULLUP);
    _pin = pin;
}
void buttonMinim::tick() {
    boolean btnState = digitalRead(_pin);
    if (!btnState && !flags.btnFlag && ((uint32_t)millis() - _btnTimer > 90)) {
        flags.btnFlag = true;
        _btnTimer = millis();
        flags.pressF = true;
        flags.holdedFlag = true;
    }
}

```

Структура buttonMinimFlags містить бітові прапори для визначення стану кнопки:

1. heldFlag - чи була кнопка затиснута.
2. btnFlag - поточний стан кнопки.
3. pressF - чи була кнопка натиснута.
4. clickF - чи була кнопка короткочасно натиснута.
5. holdF - чи утримується кнопка.

Клас buttonMinim містить:

1. Конструктор, який ініціалізує кнопку в режимі INPUT_PULLUP.
2. Метод tick(), який оновлює стан кнопки з урахуванням затримки відскоку.

3. Приватні змінні, включаючи таймер `_btnTimer` для відстеження часу між кліками.

При виклику `tick()` перевіряється, чи була кнопка натиснута, утримана або коротко натиснута, і встановлюються відповідні прапори. Це дозволяє ефективно обробляти взаємодію з кнопками без зайвих затримок в основному циклі програми.

Наступний фрагмент коду містить дві функції: `encTick()` і `btnTick()`, які відповідають за взаємодію з енкодером і кнопками.

Лістинг коду

```
void encTick() {
    enc.tick();
    if (enc.isTurn()) {
        volumeChanged = true;
        timeoutReset();
        if (enc.isLeft()) {
            thisVolume += 5;
            thisVolume = constrain(thisVolume, 5, 1000);
        }
        if (enc.isRight()) {
            thisVolume -= 5;
            thisVolume = constrain(thisVolume, 5, 1000);
        }
        dispMode();
    }
}
```

Функція `encTick()` обробляє обертання енкодера. Якщо енкодер обертається, змінна `volumeChanged` набуває значення `true`, а таймер очікування скидається за допомогою `timeoutReset()`. При повороті вліво змінна `thisVolume` збільшується на

5, а при повороті вправо - зменшується. Значення обмежується діапазоном від 5 до 1000 мл за допомогою функції `constrain()`. Після цього дисплей оновлюється викликом `dispMode()`.

Лістинг коду:

```
void btnTick() {
    if (btn.holded()) {
        timeoutReset();
        workMode = !workMode;
        dispMode();
    }
    if (encBtn.holded()) {
        pumpON();
        while (!digitalRead(ENC_SW));
        timeoutReset();
        pumpOFF();
    }
}
```

За обробку кнопок відповідає функція `btnTick()`. Якщо кнопка `btn` утримується натиснутою, відбувається обнулення таймера, перемикання `workMode` і оновлення дисплея. Якщо була натиснута кнопка енкадера (`encBtn`), то насос (`pumpON()`) активується до тих пір, поки кнопка не буде відпущена (`while (!digitalRead(ENC_SW))`). Після цього таймер скидається і насос вимикається (`pumpOFF()`).

Лістинг коду:

```
boolean timerMinim::isReady() {
    uint32_t thisMls = millis();
    if (_status && thisMls - _timer >= _interval) {
        do {
```

```

        _timer += _interval;
        if (_timer < _interval) break;
    } while (_timer < thisMls - _interval);
    return true;
    } else {
    return false;
    }
}

void timerMinim::reset() {
    _timer = millis();
}

```

Метод `isReady()` перевіряє, чи минув заданий інтервал часу з моменту останнього скидання або запуску таймера.

Спочатку зчитується поточний час у змінну `thisMls`. Якщо таймер активний (`_status == true`) і пройшов вказаний `_interval`, то таймер оновлюється.

У середині `do-while` цикл коригує значення `_timer`, додаючи інтервал `_interval` доти, доки значення не буде відповідати актуальному часу. Це запобігає пропуску кроків, якщо виклик таймера був пропущений через затримки в коді. Також є перевірка на переповнення `uint32_t`, що дозволяє уникнути збоїв у підрахунку часу.

Якщо всі умови виконані, метод повертає `true`, інакше — `false`.

Метод `reset()` просто встановлює значення таймера в поточний час, фактично скидаючи його.

Висновок до розділу 3

У даному розділі було розроблено систему та код для автоматизованої системи дозування рідин на базі мікроконтролера Arduino. На початковому етапі

було виконано моделювання системи, у якому враховано ключові компоненти та їх взаємодію, що сприяло розробці ефективного й цілісного плану реалізації.

Надалі було створено код у Arduino IDE, що містить оголошення змінних, макроси, бібліотечні підключення та реалізацію основних функціональних модулів, таких як управління насосом, сервопривід, енкодер, кнопки та світлодіодна індикація.

Структура коду побудована з використанням окремих класів для роботи з енкодером (`encMinim`), кнопками (`buttonMinim`) та таймерами (`timerMinim`), що підвищує зручність та модульність програми. Таймери забезпечують точне відстеження часу, необхідного для роботи насоса та інших елементів. Для збереження налаштувань також передбачена пам'ять EEPROM, що дозволяє відновити параметри після перезапуску.

В цілому, реалізована система є ефективною і добре структурованою рішенням для автоматизованої системи розливу рідини, що забезпечує гнучке налаштування параметрів, стабільну роботу і простоту використання.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було досліджено сучасні системи автоматичного та неавтоматичного розливу рідини, проаналізовано їх переваги та недоліки, а також класифікацію технічних засобів, що використовуються в таких системах. Особливу увагу приділено поршневим і перистальтичним механізмам, мікроконтролерам Arduino, типам насосів, датчикам та іншим ключовим елементам автоматизованих рішень.

На основі проведеного аналізу створено інтелектуальний пристрій для автоматичного дозування рідин на базі мікроконтролера Arduino. Реалізовано програмне забезпечення, що дозволяє здійснювати точне дозування, керувати насосом, сервоприводом та іншими елементами системи. Пристрій також було змодельовано у онлайн середовищі, що дозволило провести ефективне тестування та оптимізацію розробленої системи.

Розроблений пристрій характеризується високою точністю, стабільністю роботи та можливістю використання в різних галузях промисловості, таких як харчова, фармацевтична та хімічна. Робота має як практичне значення для впровадження у виробництво, так і методичне значення для навчального процесу підготовки фахівців у галузі автоматики та електроніки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Arduino URL: <https://www.arduino.cc>
2. Arduino IDE URL: <https://www.arduino.cc/en/software/>
3. Arduino Nano - плата розробки для ATmega328 URL:
<https://itmaster.biz.ua/directory/kits-nabory/arduino-nano.html>
4. Arduino UNO - популярна плата розробки URL:
<https://itmaster.biz.ua/directory/kits-nabory/arduino-uno.html>
5. Встановлення та підключення бібліотек Arduino URL:
<https://bitkit.com.ua/vstanovlennya-ta-pidklyuchennya-bibliotek-arduino>
6. Мікроконтролер ESP8266 URL:
<https://itmaster.biz.ua/directory/microcontrollers/esp8266.html>
7. Основні принципи будови та експлуатації кулера для води URL:
<https://skandinavia.com.ua/ua/blog/osnovni-principi-budovi-ta-ekspluatacii-kulera-dlya-vodi/>
8. Підключення сервоприводу SG90 до Arduino URL:
<https://myproject.com.ua/pidkliuchennia-servopryvodu-sg90-do-arduino.html>
9. Що таке мікрокомп'ютери? Розповідаємо на прикладі Raspberry Pi URL:
<https://e-server.com.ua/uk/poradi/shho-take-mikrokompiuteri-rozpovidajemo-na-prikladi-raspberry-pi>
10. Чим відрізняються Micro USB та USB Type-C? URL:
<https://itedu.center.ua/blog/comparisons/micro-usb-vs-type-c>
11. Автоматична рідинна розливна машина поршневого типу URL:
<https://uk.pistonfillingmachine.com/automatic-piston-type-liquid-filling-machine.html>
12. MX1508 Motor Driver Module – Quick Start URL: <https://www.edn.com/mx1508-motor-driver-module-quick-start/>
13. Наука за точними машинами для розливу рідин [Електронний ресурс] URL:
<https://www.ghfilling.com/uk/blog/the-science-behind-accurate-liquid-filling-machines>
14. Принцип роботи машини для розливу рідини [Електронний ресурс] URL:
<http://uk.lglpak.com/news/principle-of-liquid-filling-machine/>

15. Система наповнення перистальтичного насоса URL:

<https://ua.achievechem.com/chemical-equipment/peristaltic-pump-filling-system.html>

16. TM1637 4-digit 7-segment LED display Arduino tutorial URL:

<https://www.makerguides.com/tm1637-arduino-tutorial/>

17. Характеристики та переваги автоматичної машини для розливу та герметизації рідини URL: <https://ua.trustarpack.com/news/characteristics-and-advantages-of-automatic-li-40820852.html>

18. Принцип роботи машини для розливу води в пляшки URL:

<https://ibottling.com/uk/bottled-water-filling-machine-working-principle/>

19. Як боротися з неточним дозуванням машини для розливу рідини? URL:

<https://ua.dihai-rob.com/news/how-to-deal-with-inaccurate-dosage-of-liquid-67497400.html>

20. Що таке кулер: його основне призначення URL:

<https://mywatershop.com.ua/blog/shcho-take-kuler-yogo-osnovne-priznachennya/>

Мета за завдання роботи

- **Об'єкт дослідження**

Автоматизація процесу розливу рідин.

- **Мета дипломної роботи**

Розробка автоматизованої системи дозування рідин.

- **Завдання роботи**

Аналіз існуючих рішень, вибір необхідних компонентів, розробка програмного забезпечення, тестування та оцінка результатів.

1

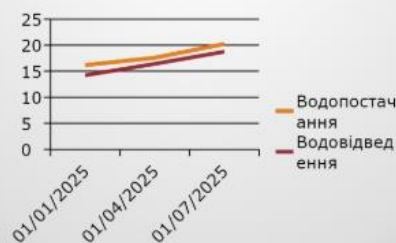
Причини актуальності теми

Чому це актуально?

1. Проблеми з питною водою у містах.
2. Зростання цін на водопостачання.
3. Зростання попиту на автоматизовані рішення в побуті.

Хто користувач?

1. Побутові користувачі (домогосподарства).
2. Малий бізнес.
3. Сфери обслуговування
4. Лабораторії та наукові установи.



2

Розділ 1. Існуючі системи для розливу рідин

- Визначення переваг та недоліків.
- Аналіз сфер застосування.
- Огляд компонентів пристроїв.

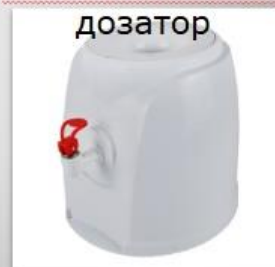
поршнева



перистальтична



неавтоматичний
дозатор



3

Розділ 1. Мікроконтролерів для розумних систем

- Перелік існуючих екземплярів
- Аналіз компонентів мікроконтролерів.
- Виявлення найкращих варіантів використання.

Arduino Uno



Raspberry Pi 4



ESP8266



4

Розділ 2. Основні деталі для "розумного автомату"

- Arduino NANO
- Мембранний насос
- Драйвери двигуна
- Перемикач з виступом



5

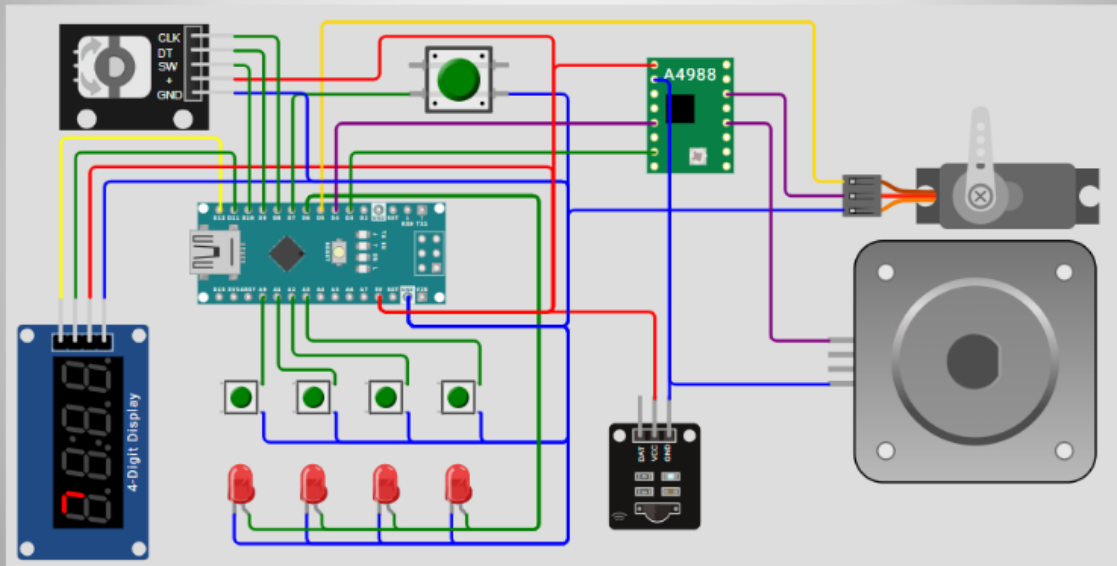
Розділ 2. Додаткові деталі для "розумного автомату"

- Micro USB
- 4-розрядний Модуль дисплея
- Сервопривод



6

Розділ 3. Модель системи



7

Розділ 3. Опис коду

```

WaterDispenser.ino  a_setup.ino  b_loop.ino  buttonMinim.h  c_func.ino  d_control.ino  encUniversalMinim.h  timer2Minim.h

1  |
2  #define NUM_SHOTS 4
3  #define TIMEOUT_OFF 5
4  #define SWITCH_LEVEL 0
5
6  const byte shotPos[] = {25, 60, 95, 145, 60, 60};
7
8  const long time50ml = 5500;
9
10 #define KEEP_POWER 1
11 #define DEBUG_UART 1
12 #define PUMP_POWER 3
13 #define SERVO_POWER 4
14 #define SERVO_PIN 5
15 #define LED_PIN 6
16 #define BTN_PIN 7
17 #define ENC_SW 8
18 #define ENC_DT 9
19 #define ENC_CLK 10
20 #define DISP_DIO 11
21 #define DISP_CLK 12
22 const byte SW_pins[] = {A0, A1, A2, A3, A4, A5};
23

```

8

Висновок

розповсюджені автоматичні системи:

- великі габарити
- велика ціна обслуговування
- сильне забруднення

розроблена автоматична система:

- невеликий розмір
- легка у обслуговуванні
- мале накопичення бурду

9

Апробація

Робота була апробована на VI Науково-технічній конференції «Сучасний стан та перспективи розвитку IoT», яка проходила 15 квітня 2025 року. Тезу на тему " Необхідність автоматизації розливу рідин у системах розумного будинку " було опубліковано у збірнику, присвяченому цій конференції.

10