

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Комп'ютерно-інтегрованих технологій
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему:

**Розробка автоматизованої системи керування
біореактором.**

Виконав(ла): студент(ка) IV курсу, групи КТ-41
спеціальності 151 – Автоматизація та комп'ютерно-
інтегровані технології

(шифр і назва спеціальності)

(підпис)

Хортик Д. Ю.

(прізвище та ініціали)

Керівник

(підпис)

Стухляк Д.П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Чихіра І.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Микитишин А.Г.

(прізвище та ініціали)

Рецензент

(підпис)

Медвідь В.Р.

(прізвище та ініціали)

Тернопіль
2023

АНОТАЦІЯ

Кваліфікаційна робота складається з пояснювальної записки та графічної частини.

Об'єм графічної частини кваліфікаційної роботи становить __ слайдів. Об'єм пояснювальної записки складає __ друкованих сторінок формату А4 (210×297), об'єм додатків – __ друкованих сторінок формату А4.

У цьому проекті було розроблено керуючу програму для мініатюрної системи біореактора, яка виробляє біогаз за допомогою відкритої електронної платформи Arduino. Розроблена програма керування підходить для візуалізації та моніторингу відповідних параметрів процесу. Однак слід зробити незначні адаптації, щоб зробити його придатним для керування мініатюрною системою біореактора. По-перше, контрольна програма виявилася ефективною для моніторингу визначених параметрів. Контроль відповідних параметрів процесу був математично прийнятним. Тим не менш, хімічна точка зору припускає, що PID базового насоса слід відрегулювати, щоб він був достатнім для контролю значення рН у системі біореактора.

Візуалізація відповідних параметрів за допомогою розробленого графічного інтерфейсу виявилася успішною. Крім того, представлена система керування є легкою для створення та є економічно ефективною альтернативою порівняно з іншими системами керування, які доступні для біореакторів звичайного масштабу. Остаточна система керування Arduino була розроблена відповідно до кількох етапів, а саме: визначення налаштувань системи, апаратного забезпечення Arduino, програмного забезпечення Arduino та графічного інтерфейсу користувача.

Графічний інтерфейс користувача не має бездротового підключення до контролера Arduino. Подальші дослідження повинні вказати, чи є зчитування значень з IP-адреси та їх передача до обробки цінним майбутнім вдосконаленням дизайну.

Ключові слова: КОНТРОЛЬ, БІОРЕАКТОР, КЕРУВАННЯ, РІВЕНЬ.

ЗМІСТ

ВСТУП	6
1. АНАЛІТИЧНА ЧАСТИНА	7
1.1. Постановка завдань розробки.....	7
1.2. Матеріали та методи розробки системи керування.....	10
1.3. Стандартні пристрої для реалізації проекту.....	16
2. ПРОЄКТНА ЧАСТИНА.....	19
2.1. Аналіз вхідних параметрів для системи керування.....	19
2.1.1. Значення рН.....	20
2.1.2. Значення температури.....	21
2.1.3. Швидкість потоку біогазу.....	23
2.2. Аналіз вихідних параметрів системи.....	24
2.2.1. Перистальтичний базовий насос.....	25
2.2.2. Активація/деактивація сигналізації про зміну температури.....	26
2.2.3. Швидкість перемішування.....	27
2.2.4. Корм.....	29
2.2.5. Програмне забезпечення Arduino.....	30
2.3. Розрахунки базового насоса.....	33
3 СПЕЦІАЛЬНА ЧАСТИНА.....	37
3.1 Розробка графічного інтерфейсу користувача.....	37
3.2. Розробка програми системи керування.....	38
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ.....	55
4.1 Долікарська допомога при шоку.....	55
4.2 Розробка, оформлення кімнати для психологічного розвантаження працівників ..	57
ВИСНОВКИ.....	61
БІБЛІОГРАФІЯ.....	62

ВСТУП

З моменту розробки біореакторів вчені знайшли багато способів застосування технології в кількох галузях техніки. Приклади таких галузей включають очищення стічних вод, клітинну культуру та тканинну інженерію в охороні здоров'я, виробництво цінних фармацевтичних препаратів і хімічних речовин у промислових біотехнологіях і навіть вирощування водоростей для отримання кисню під час дослідження космосу. У зв'язку з постійним розвитком біотехнологічної сцени попит на різноманітні програми активно збільшує прогрес у конструкціях біореакторів.

Однією з останніх розробок біореакторних конструкцій є мікробіореакторна система. Ці мікробіореакторні системи (MBR) набагато менші за загальновідомі біореактори. Мікробіореактор — це зменшений біореактор, який має ємність від приблизно 0,1 мл до приблизно 100 мл.

MBR спочатку були розроблені для досягнення прискорення швидкості обробки шляхом використання високопродуктивної здатності, оскільки висока пропускна здатність дає можливість виконувати багато клітинних культивувань паралельно. Група продуктів і процесів для біотехнології (GPPB) Університету Гронінгена розробила власну систему біореактора, яка містить чотири багатопаралельні мініатюрні операції біореактора.

1. АНАЛІТИЧНА ЧАСТИНА

1.1. Постановка завдань розробки

Робота біореактора буде використовуватися для вивчення моделювання поведінки клітинних культур процесами мікробної ферментації. У цих процесах швидкість росту та виробництва клітин сильно залежить від кількох параметрів процесу, а саме: рН, швидкості перемішування та температури.

Конструкцію цієї системи біореактора можна побачити на рис 1.1.

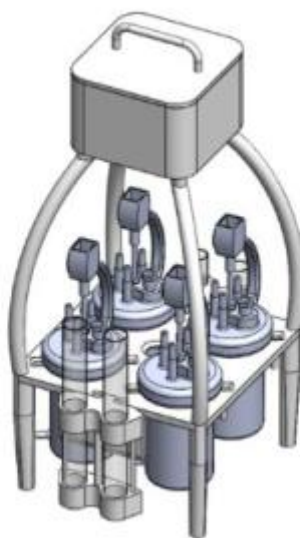


Рисунок 1.1 - Чотири багатопаралельні мініатюрні біореакторні системи.

GPPB хоче працювати з 96 операціями одночасно. Це означає, що система, яка зображена на рис. 1.1, буде використана 24 рази. Показана система, за оцінками, має продуктивність приблизно 1–40 мл біогазу на день.

Ці параметри можна контролювати та спостерігати в системі мікробіореакторів. У цьому проекті основна увага буде зосереджена на одній мініатюрній системі біореактора з відповідними параметрами процесу.

Для контролю та моніторингу відповідних параметрів процесу GPPB хоче розробити програму автоматизованого керування, яка є менш дорогою

порівняно з системами керування, які частіше використовуються для біореакторів звичайного масштабу.

Окрім зацікавленості GPPB у цьому проекті, інші інститути та університети також можуть бути зацікавлені в можливо розробленій програмі керування для своїх власних майбутніх досліджень. Проблему відносно високих витрат, які виникають для звичайних систем керування, можна вирішити, використовуючи Arduino як мікроконтролер. Arduino — це друківана плата, спеціально розроблена для того, щоб бути дешевою та простою у використанні.

Крім того, плата ARDUINO також має багато інших електронних компонентів, які необхідні для функціонування мікроконтролера або для розширення його можливостей. Мова програмування, яка використовується в Arduino, є спрощеною мовою C++ в інтегрованому середовищі розробки (IDE), доступ до якого легко отримати з Інтернету.

Arduino використовує файли програмного коду з іменами (*.ino), які називаються ескізами, які можна скопіювати та завантажити на друківану плату за допомогою USB-кабелю, який з'єднує друківану плату Arduino з пристроєм, на який завантажено IDE. Системи керування міні-біореакторами є дефіцитними продуктами. У всесвітній мережі можна знайти біореактор Томмі Лінтілава.

Біореактор у його процесі контролює швидкість перемішування для біореактора ротаційної системи культивування клітин. З використанням Arduino вартість цього RCCS становить приблизно 65 євро. Однак RCCS контролює лише один параметр у цьому проекті буде контролюватися кілька параметрів.

Таким чином, можна зробити оцінку, що витрати на цей проект перевищать 65 євро, які використовуються для RCCS.

Цінова оцінка важлива для можливих майбутніх інвесторів, таких як Applikon Biotechnology, яка сприяє фінансуванню розробки мініатюрних біореакторів. Applikon Biotechnology є світовим лідером у розробці та

постачанні сучасних біореакторних систем для промислових біотехнологій. Applikon може направляти клієнта від початкового перегляду до повномасштабного виробництва, використовуючи ту саму платформу.

Таким чином, програма, яка могла б контролювати міні-біореактори, може бути цікавою як майбутня інвестиція для Applikon. Використання плати Arduino для контролю та моніторингу відповідних параметрів процесу роботи біореактора має більше переваг, ніж її простота та відносно недорогі компоненти. По-перше, використання комп'ютерів (тобто через пристрій/карту збору даних) у процесах біореактора надає можливість зберігати дані та перетворювати їх у формат, який можна використовувати в електронних таблицях або інших програмних пакетах для детальнішого аналізу.

Іншою перевагою є отримання даних у реальному часі для цілей моніторингу та контролю з можливою реалізацією повністю автоматизованої установки мікробіореактора, що може полегшити майбутні дослідження процесів біореактора.

Arduino вже використовується як контролер для біореакторів. Хусейн наводить гарний приклад використання Arduino у сцені з біореакторами. Він спеціально розробив контролер для мініатюрного біореактора, який є таким самим у цій роботі. Результати показують, що розробка програми на основі симулятора Arduino досяжна. Кінцевою метою цього проекту є розробка програми Arduino, яка може імітувати культуру клітин. Після цього програма повинна мати можливість повертати відповідні параметри процесу з культури клітин. Крім того, програма також повинна забезпечувати графічний інтерфейс користувача, де можна вказати відповідні параметри процесу, а поведінку моделювання можна контролювати під час роботи мікробіореактора.

1.2. Матеріали та методи розробки системи керування.

Щоб вирішити, які компоненти Arduino слід використовувати для керування системою, необхідно отримати інформацію про розроблений мініатюрний біореактор та його відповідні параметри процесу. У наступних двох розділах буде розглянуто налаштування системи та відповідні параметри з їх робочими діапазонами.

Налаштування системи.

Перш ніж розпочати пояснення щодо налаштування роботи мініатюрного біореактора, слід пояснити, як використовуються біореактори та який із цих процесів використовується у випадку мініатюрного біореактора, розробленого та виготовленого GPPB. В даний час біореактори використовуються для різних процесів. Яскравими прикладами цього є культивування клітин ссавців (таких як СНО), мікробних клітин (таких як кишкова паличка), дріжджів або малих рослинних клітин (таких як мох).

Принцип цих процесів полягає в тому, що клітини виробляють бажану сполуку шляхом перетворення поживних речовин у високоцінні біофармацевтичні продукти. Перетворення поживних речовин відбувається шляхом бродіння. Процес бродіння можна проводити двома способами, а саме анаеробним і аеробним. Найбільша різниця між ними полягає у використанні кисню. При анаеробних процесах процес відбувається без використання кисню.

У випадку мініатюрного біореактора використовується анаеробний процес. Перевага анаеробного процесу полягає в тому, що він вимагає низьких витрат енергії. Крім того, буде утворюватися невелика кількість побічних продуктів (шламу). Розроблений мініатюрний біореактор використовує у своїй конструкції процес анаеробного зброджування.

Поперечний розріз і вид зверху мініатюрного біореактора можна знайти нижче на рис. 1.2. Важливі частини позначені стрілкою та описом. На рисунку можна виділити чотири параметри, які необхідно враховувати при

проекуванні міні-системи керування. Параметрами є рН, температура, перемішування та швидкість потоку біогазу. Мініатюрний біореактор спеціально розроблений для вимірювання цих параметрів.

Візуалізовано з видом зверху, гайку та порт для зонда рН і мішалки можна легко отримати для контролю. Крім того, для вимірювання та контролю витрати біогазу використовується вихід газу. Цей газовий вихід буде вести до колби з водою. Завдяки тиску газу вгорі вода буде виходити з колби. Різниця у вазі колби згодом може вказувати, скільки газу було утворено під час роботи біореактора. На рис. 1.3 показано комбінацію процесу біореактора та колби для вимірювання витрати біогазу.

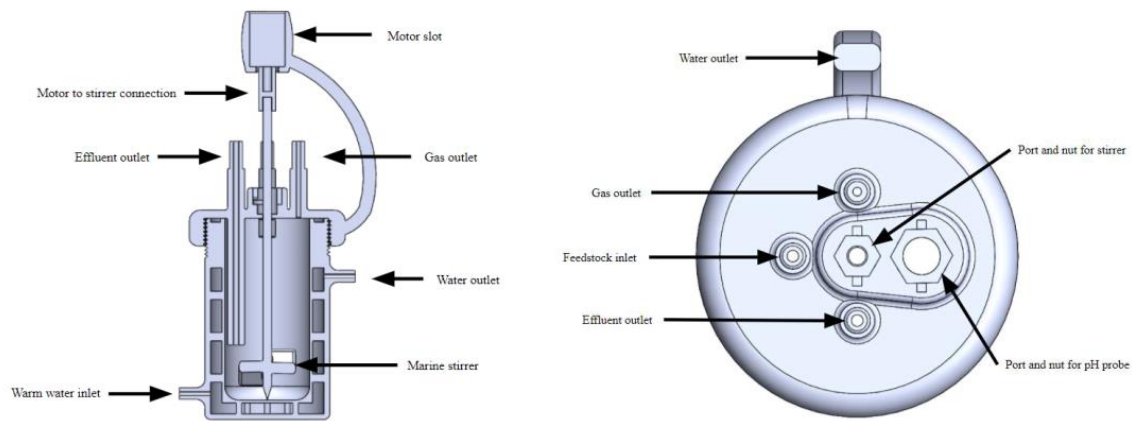


Рисунок 1.2 - Поперечний розріз і вид зверху розробленого мініатюрного біореактора.

Параметри.

Під час роботи біореактора одним із важливих завдань є контроль ключових параметрів. Ключові параметри можуть оптимізувати процес, підтримуючи клітини в потрібному фізичному та хімічному середовищі. Параметри, які необхідно контролювати та/або вимірювати, це рН, температура, перемішування та швидкість потоку біогазу. Кожен із цих параметрів розглядатиметься відповідно до їх важливості та робочих діапазонів.

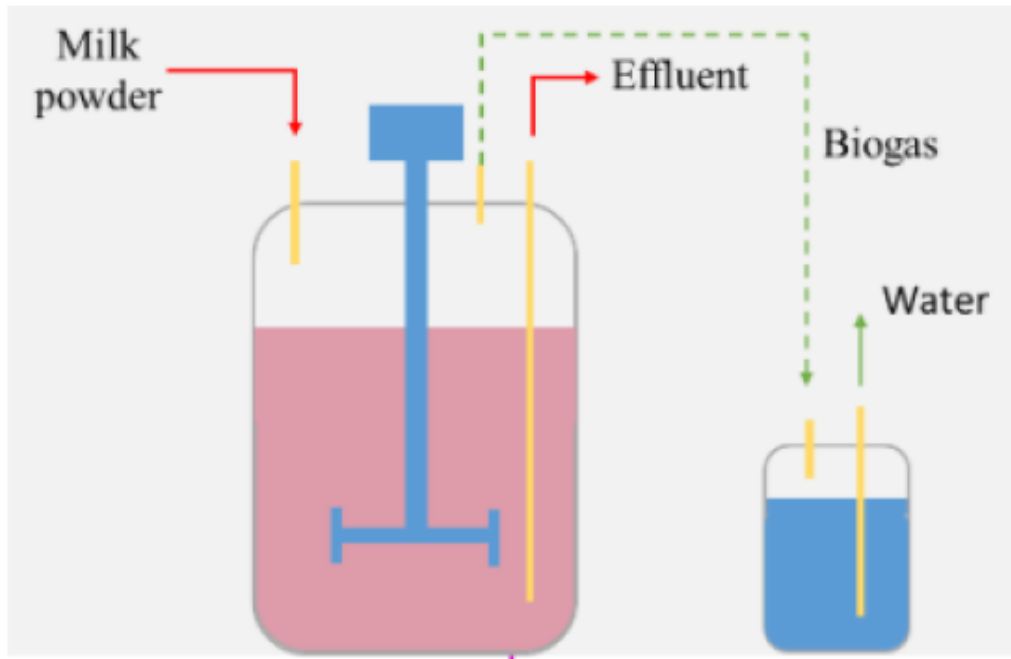


Рисунок 1.3 - Регулювання витрати біогазу.

pH Рівень.

pH є одним із найважливіших параметрів, який необхідно контролювати. Без контролю pH ріст клітин може бути уповільнений або навіть повністю зупинений через надто кисле середовище вирощування.

Зміни значень pH можуть негативно вплинути на вихід продукту в процесі експлуатації. Таким чином, підтримання pH у правильному робочому діапазоні впливатиме як на життєздатність клітин, так і на якість продукту. Рішенням згаданих проблем може бути встановлення попередньо визначеного значення pH. Зазвичай біореактор повинен мати робочий діапазон значення pH між 6,7 і 7,4, тоді як метаногенна активність припиняється, коли значення pH знижується нижче 6,5.

Щоб підтримувати попередньо визначене задане значення, слід використовувати зонд pH. Цей pH-зонд має вимірювати значення pH у мініатюрному біореакторі. Якщо pH має тенденцію бути занадто кислим, слід додати основний розчин, щоб стабілізувати pH.

Температура.

Інший важливий параметр, який потрібно контролювати, це температура. В анаеробних процесах можна виділити дві істотні температурні зони. Причиною різних температурних зон є різниця в типі мікроорганізмів, з якими працюють. Мікроорганізми можна розділити на мезофільні та термофільні організми.

Мезофільні організми мають оптимальну швидкість роботи при температурі приблизно 35 °С, тоді як оптимальна швидкість роботи термофільних організмів знаходиться при температурі близько 55 °С. У випадку мініатюрного біореактора резервуар нагрівається до 35 °С. Контроль температури важливий з кількох причин. Температура вище 37 °С може швидко пошкодити життєздатність клітин, тоді як нижча температура може призвести до сповільнення клітинного метаболізму. Крім того, під час роботи біореактора слід прагнути до однорідної та постійної температури. В іншому випадку процес не може бути проведений ефективно.

Перемішування.

Перемішування в біореакторі підвищує ефективність контролю температури та рН. Система перемішування біореактора складається з робочого колеса, приводного механізму і двигуна. Мета системи перемішування полягає в тому, щоб забезпечити введення потужності в культуральне середовище шляхом змішування речовини в біореакторі для отримання однорідного розподілу рН, температури та належної швидкості потоку біогазу.

Ефективність розподілу тісно залежить від розміру крильчатки, типу та розташування через гідродинаміку та аерацію та їх потенційний вплив на клітини та процес. Однак дослідження показує, що, якщо метантенк постійно перемішувати, різна інтенсивність змішування лише мінімально впливає на біореактор. Єдиний ефект можна знайти при початковому запуску. Інтенсивне змішування під час початкового запуску шкодить продуктивності.

Щоб запобігти виходу з ладу варочного котла на початковому етапі запуску, слід уникати інтенсивного перемішування.

Потік біогазу.

Продуктом процесу є виробництво біогазу. Для розрахунку ефективності процесу необхідне вимірювання кількості виробленого біогазу. Вироблений біогаз може бути джерелом енергії, яке можна використовувати як електроенергію, тепло, біопаливо або вводити в мережу природного газу. Основним складом біогазу, який утворюється в результаті анаеробного збродження, є метан і вуглекислий газ 50%~70% і 25%~50% відповідно.

Однак біогаз також містить інші сліди газів, вологу, тверді частинки (PMs) і забруднювачі, такі як леткі органічні сполуки (VOCs), сполуки сірки, силосани та аміак. В даний час GBBP використовує газову хроматографію для визначення складу біогазу. Біогаз відбирають з біореактора за допомогою шприца.

Після цього біогаз досліджують шляхом розділення хімічних компонентів за допомогою носіїв для виявлення та визначення присутності чи відсутності та/або кількості необхідних компонентів.

Годування.

Для ефективного виробництва біогазу необхідним є регулювання умов вирощування. Виснаження поживних речовин разом із накопиченням метаболічних відходів є основними факторами старіння та метаболічних змін, які знижують якість виробленого біогазу. Використання систем годівлі може усунути коливання рівня глюкози та покращити виробництво біогазу. Цього можна досягти двома діями. По-перше, шляхом подачі ключових поживних речовин у мініатюрний біореактор, щоб біологічний реактор усередині біореактора не застоювався через низьку кількість бажаних поживних речовин.

По-друге, шляхом віднімання метаболічних відходів і непридатної біоматерії. Оптимальна стратегія живлення в біореакторах залежить від структури кінетики реакції та взаємодії між різними реакціями. Щоб

регулювати подачу в мініатюрний біореактор, було визначено, що вхідний і вихідний отвір складатиметься з 2 мл на день. Крім того, до складу корму входить сухе молоко на дистильованій воді.

Апаратне забезпечення Arduino.

Параметри та налаштування системи тепер визначено. Таким чином, датчики та виконавчі механізми можуть бути призначені для кожної функції та параметра. Щоб почати з розподілу датчиків і виконавчих механізмів, слід вибрати стандартні частини, необхідні для роботи з Arduino. Після цього необхідно визначити, які додаткові компоненти Arduino потрібні. Крім того, за допомогою Fritzing і Circuito будуть надані схематичні огляди схем Arduino.

Щоб досягти цього, необхідно розрізнити входи, процеси трансформації та виходи. Процеси трансформації можуть бути повністю заповнені за допомогою правильного кодування. Однак входи та виходи мають виконуватися компонентами Arduino.

Вхідні дані:

1. Значення рН.
2. Значення температури.
3. Об'єм біогазу.
4. Корм.

Процеси перетворення корму:

1. Аналізуйте значення рН: якщо воно нижче 6,7, запустіть двигун, щоб додати основу до біореактора.

2. Проаналізуйте значення температури: якщо нижче 34, запустіть сигнал тривоги, доки працівник лабораторії не припинить сигнал тривоги після перевірки нагрівача. Якщо температура вище 36, включити сигналізацію, поки лаборант не вимкне сигналізацію після перевірки нагрівача.

3. Газ, що утворюється, виштовхує із системи такий же об'єм води. Після цього обчисліть об'єм біогазу, використовуючи різницю у вазі. Наприклад, 1 грам дорівнює 1 мл, тобто 1 см^3 .

Виходи:

1. Запуск базового насоса.
2. Увімкнення/вимкнення сигналу тривоги, щоб повідомити, коли нагрівач не повністю заповнює виробничі стандарти.
3. Швидкість перемішування.
4. Запуск насосів для віднімання та додавання корму в мініатюрну систему біореактора.
5. Графічний інтерфейс користувача.

1.3. Стандартні пристрої для реалізації проекту.

Щоб створити робочу систему керування, необхідно додати базові частини, необхідні для кожного контролера Arduino. У цьому розділі розглядатимуться основні частини з коротким описом того, що містять у собі основні частини. Плата Wi-Fi Arduino Uno Rev 2: плата Wi-Fi Arduino Uno Rev 2 є одна з найбільш використовуваних плат Arduino при роботі з підключенням до Інтернету. Arduino Uno Wi-Fi функціонально такий самий, як Arduino Uno Rev3, але з додаванням Wi-Fi/Bluetooth та деякими іншими вдосконаленнями.

Він включає в себе 8-розрядний мікроконтролер ATmega4809 від Microchip і має вбудований IMU (інерційний вимірювальний блок) LSM6DS3TR. Він має чотирнадцять цифрових входів/виходів (з яких шість можна використовувати як ШІМ-виходи), шість аналогових входів, шістнадцять МГц керамічний резонатор (CSTCE16M0V53-R0), USB-з'єднання, роз'єм живлення, роз'єм ICSP і кнопку скидання.

Плату можна легко підключити до пристрою за допомогою USB-кабелю принтера або за допомогою Інтернету. Кожен компонент Arduino, з

яким буде працювати, повинен бути підключений до джерела живлення (5 В або 12 В) плати Arduino. Крім того, кожен ланцюг повинен закінчуватися в порту GND, щоб замкнути ланцюги.

Макетна плата.

Однією з важливих частин при прототипуванні нової системи керування Arduino є макетна плата, яку можна використовувати для прототипування ваших схем. Можливість тимчасово використовувати компоненти, а не приєднувати їх до схеми та спаювати на місці, особливо корисна на етапі проектування проекту. Налаштування макетних плат є звичайним. Зовнішня частина складається з пластикового корпусу з рядами та колонками отворів. Під цими рядками та стовпцями можна знайти мідні доріжки, які дають можливість електрично з'єднувати компоненти один з одним.

Мультиметр.

Мультиметр - це пристрій, який допомагає з внутрішнім розумінням схеми Arduino, з якою працюється. Мультиметр - це вимірювальний прилад, який вимірює вольти, ампери та опір. Він може показати електричні значення різних компонентів і те, що відбувається з ланцюгом. За допомогою цього пристрою можна перевірити, чи належним чином вони працюють.

Дроти для переходу.

Дроти для переходу використовуються в поєднанні з макетною платою. Ізольований дріт можна використовувати для підключення плати Arduino до компонентів і макетної плати. У випадку Arduino найчастіше використовується ізольований провід обладнання. Цей тип дроту зазвичай використовується в невеликих масштабах для низьковольтних електричних застосувань. Проводи для перемикачів можна розділити на два різновиди: одножильні та багатожильні.

Одножильні дроти для перемикачів - це один шматок дроту, який здебільшого використовується в ситуаціях, коли дріт акуратно покладено на макетну плату, не зміщуючи його занадто сильно. Багатожильний дріт

складається з більшої кількості маленьких дротів, скручених разом, що надає багатожильному дроту більшої міцності та стійкості до згинання, ніж одножильний дріт. Це призводить до можливості частіше переміщати багатожильні дроти. У цьому проекті будуть використовуватися багатожильні дроти.

Екран реле/Mosfet.

Екран реле використовується не в усіх системах Arduino. Однак у системі, яка розроблена в цьому проекті, релейний екран має важливе значення. Arduino Relay Shield - це рішення для керування навантаженнями високої потужності, якими неможливо керувати цифровими ІО Arduino через обмеження струму та напруги контролера. Прикладами цього є двигун мішалки та базовий насос.

Екран реле може керувати навантаженнями високої потужності, розділяючи вхідну напругу на дві секції. Перша секція - це секція команд, яка живиться лише від 5 вольт, які можуть перемикати реле між двома контактами перемикання полюсів (нормально відкритим і нормально закритим). Друга секція подає 12 В, необхідне для активації насоса або двигуна.

MOSFET, що означає Metal Oxide Semiconductor Field Effect Transistor, - це напівпровідниковий пристрій, який можна використовувати для посилення електронних сигналів в електронних пристроях. Тому MOSFET можна використовувати як екран реле. MOSFET - це пристрій із чотирма клемми, який містить клему для витоку, затвора, стоку та корпусу. MOSFET зазвичай розглядається як транзистор і використовується як в аналогових, так і в цифрових схемах. MOSFET можна використовувати в поєднанні з приводами, які необхідні в системі Arduino, яка буде розроблена.

2. ПРОЄКТНА ЧАСТИНА

2.1. Аналіз вхідних параметрів для системи керування.

Вхідні дані – це інформація, отримана з мініатюрного біореактора, яка буде зчитана платою Arduino. До вхідних портів Arduino Uno необхідно підключити кілька датчиків. Кожен із цих датчиків використовує напругу для перекладу отриманої інформації на мову, зручну для Arduino Uno. Крім того, рівень напруги також використовується для визначення точності датчиків.

Точність датчика.

Точність кожного датчика можна обчислити за допомогою мілівольт і бітів, до яких відносяться мілівольти. Мікроконтролер всередині Arduino Uno має схему, яка називається аналого-цифровий перетворювач. Зміна напруги, яка виникає під час використання Arduino Uno, може бути зчитана АЦП, який потім перетворює зміну напруги в число від 0 до 1023.

Діапазон 0 – 1023 можна пояснити за допомогою бітів. Arduino Uno має роздільну здатність 10 біт. Це призводить до двійкового числа від 00 0000 0000 до 11 1111 1111, яке вказує на десяткове число 0 -1023, як було сказано раніше. Це опосередковано означає, що 0 вказує на наявність 0 В на виводі Arduino Uno, а 1023 означає, що на виводі Arduino Uno є напруга 5 В.

У діапазоні від 0 до 5 В `analogRead()` повертає число від 0 до 1023, пропорційне величині напруги, що подається на контакт Arduino Uno. Маючи ці знання, можна пояснити точність самої Arduino Uno. Відповідно до таблиці даних Arduino Uno, точність АЦП в Arduino Uno становить близько $\pm 2 \cdot \text{LSB}$, що означає $5 \text{ В} / 1023 \cdot 2 = 9,8 \text{ мВ}$.

Порівняно з бітами, помилка $\pm 2 \cdot \text{LSB}$ становить 2 біти (4 десяткові) у 10 бітах (1023 десяткових), що є максимальною похибкою у випадку точності конвертера $4/1023$ або приблизно $1/256$, що дорівнює 0,25 %. Маючи ці

знання, можна розрахувати точність кожного датчика, щоб перевірити, чи можуть датчики ефективно вимірювати відповідні параметри процесу.

2.1.1. Значення рН.

Для зчитування рН буде використовуватися рН-електрод від VWR international. Електрод рН підключено до приймача Arduino BNC, який може надсилати виміряні значення на аналоговий PIN 2 Arduino Uno. Після цього аналогові показання можна перевести в значення рН від 1 до 14.

Крім того, тепер можна розрахувати точність датчика рН. Якщо загальне значення рН (14) розділити на 256 (межа похибки), можна отримати точність +/- 0,055. Електрод рН необхідно відкалібрувати за допомогою триточкового калібрування. Причина цього полягає в тому, щоб перевірити, чи електрод рН є точним між кожною з цих точок. З правильним кодом рН-електрод потрібно помістити в розчин із значенням рН 7. Код потрібно налаштувати таким чином, щоб виміряне значення рН відповідало розчин з рН 7. Потім те ж саме потрібно зробити з розчином зі значенням рН 4 і розчином зі значенням рН 10.

Якщо виміряні значення не відрізняються на +/- 0,1 від даних розчинів, рН електрод відкалібрований належним чином. Компоненти, які використовуються для вимірювання значення рН, можна знайти в описі матеріалів нижче. Коли ці предмети придбано, можна створити схему Arduino на малюнку 4. На малюнку 4 видно, що приймач BNC підключений до джерела живлення та порту GND Arduino UNO.

Перелік матеріалів рН-електроду

BreadBoard - Half Size	1x
Arduino Uno	1x
USB Cable A to B	1x
pH electrode	1x
BNC receiver Arduino	1x
Jumper Wires Pack - M/M	6x

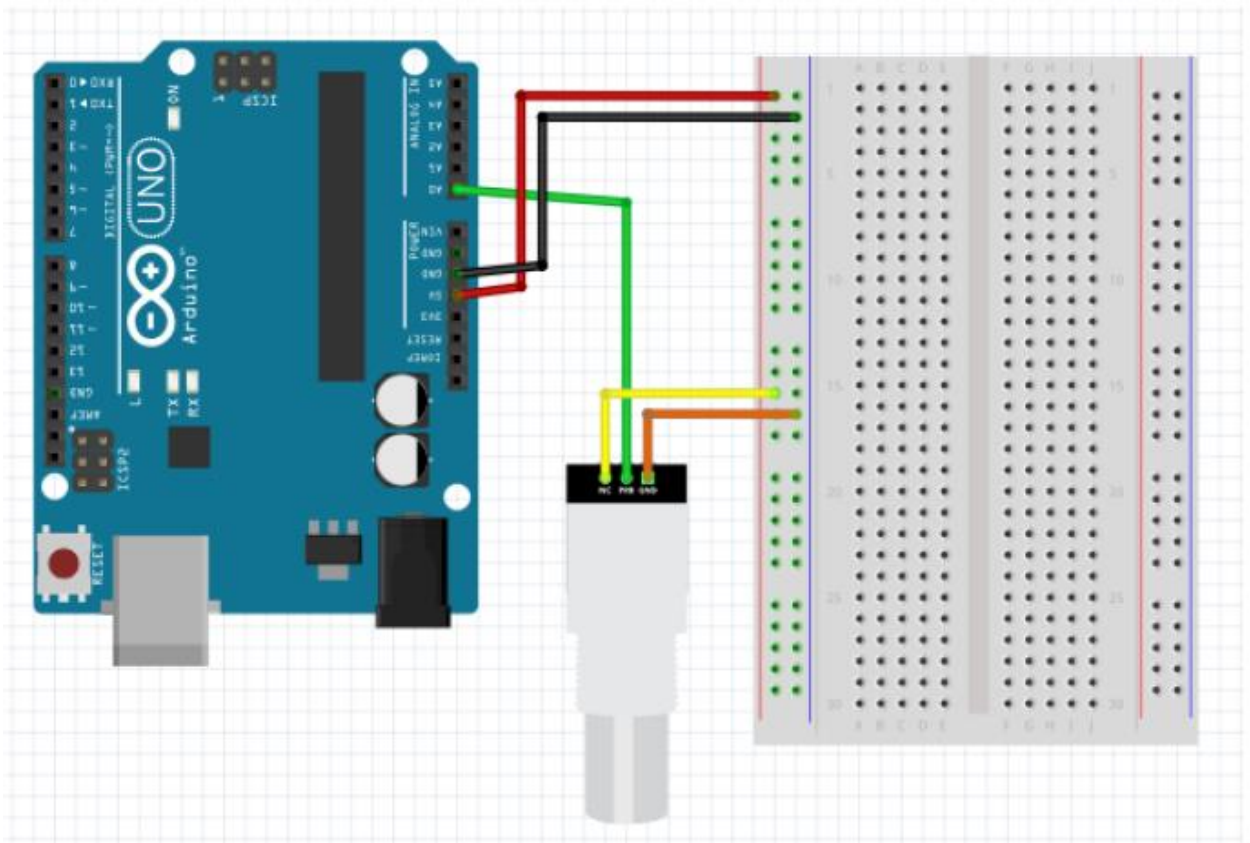


Рисунок 2.1 - Схема Arduino рН-електрода.

2.1.2. Значення температури.

Для вимірювання значення температури водного басейну буде використовуватися цифровий термодатчик однопровідної шини DS18B20. Одна дротова шина означає, що тепловому датчику потрібна лише одна лінія даних (і GND) для зв'язку з платою Arduino. Термальний датчик може живитися від плати Arduino завдяки діапазону живлення від 3,0 В до 5,5 В.

Крім того, робочий діапазон теплового датчика становить від -55°C до 125°C . Це означає, що він може легко витримувати температуру, яку потрібно виміряти $\sim 35^{\circ}\text{C}$. Точність теплового датчика при температурі близько 35°C знову можна розрахувати, поділивши загальний температурний діапазон (180°C) на 256, що в результаті дасть точність $\pm 0,7^{\circ}\text{C}$.

Однак у літературі зазначено, що термодатчик є точнішим при температурі близько 35°C порівняно з точністю навколо границь термодатчика. Таким чином, можна оцінити, що точність у випадку цього проекту становить $\pm 0,5^{\circ}\text{C}$. Матеріали, необхідні для роботи термодатчика, наведено в описі матеріалів нижче. Окрім підключення теплового датчика до 5 В і порту GND, його також потрібно підключити до будь-якого цифрового або аналогового контакту.

Таблиця 2.2

Перелік матеріалів датчик температури DS18B20

BreadBoard - Half Size	1x
Arduino Uno	1x
4.7K Ohm Resistor	1x
DS18B20 temperature sensor	1x
Jumper Wires Pack - M/M	5x

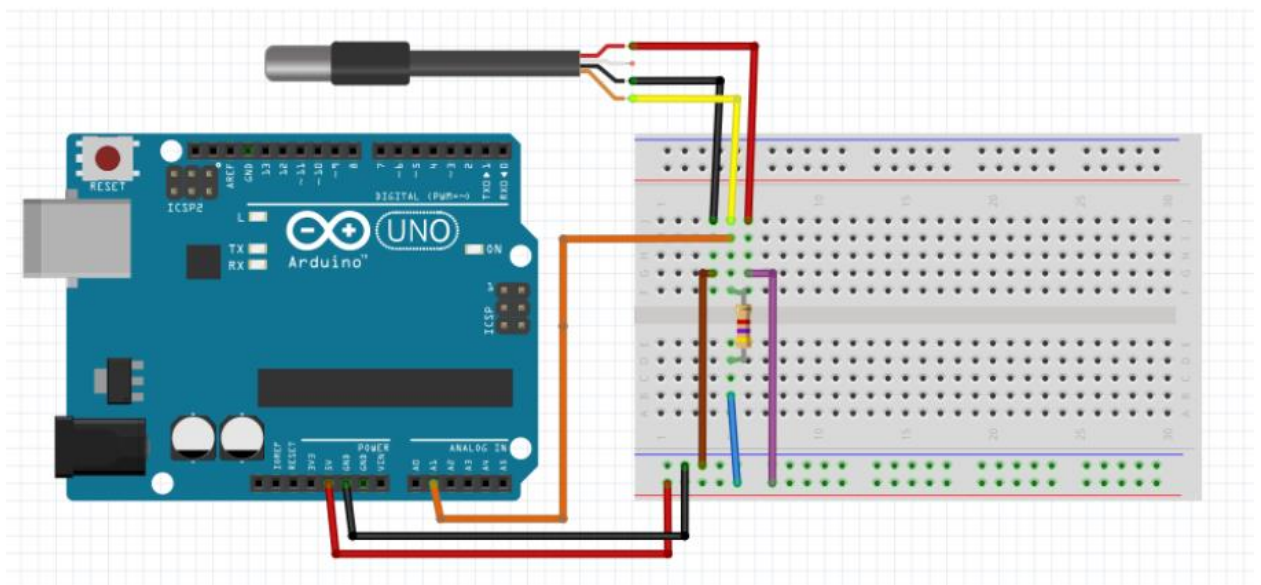


Рисунок 2.2 - Датчик температури Arduino Circuit DS18B20.

2.1.3. Швидкість потоку біогазу.

Швидкість потоку біогазу буде вимірюватися відповідно до різниці у зваженій воді в системі. Компонент Arduino, який використовується для цього вимірювання, буде тензодатчиком Arduino 100 г у поєднанні з датчиком навантаження SparkFun HX711 Підсилювач. Ці частини вибрано для достатнього управління діапазоном виробництва біогазу, який становить приблизно 1 мл – 40 мл на день.

У пошуках дійсного датчика нормальні датчики витрати газу були виявлені недостатньо точними для виробничого діапазону. Діапазон технологічних процесів датчиків витрати, які були знайдені, потребує мінімум 0,25 л/хв для функціонування. У цьому проекті біогаз має приблизний діапазон від 0,04 мл/год до 1,6 мл/год. Це набагато менше, ніж мінімум для датчиків витрати газу.

Тому було прийнято рішення використовувати тензодатчик з максимальною вагою 100 г, тоді як датчик вимірювання навантаження може точно вимірювати вагу від 0 г до 100 г. Розрахунок точності можна здійснити, розділивши загальний діапазон ваги (100 г) на 256, що дасть точність приблизно +/- 0,4 грама. Таким чином, точність тензодатчика достатньо точна, щоб виміряти біогаз, який виробляється протягом одного дня.

Для перекладу вимірювань тензодатчика буде використано SparkFun HX711. Для роботи SparkFun HX711 потрібні два цифрові контакти. Схема Arduino потребує наступних частин, наведених у таблиці 2.3. Для ефективного використання проводів переходу для з'єднання до SparkFun HX711 необхідно під'єднати пакет штекерних роз'ємів Break-Away. Схематичний огляд схеми Arduino датчика навантаження можна знайти на рис. 2.3.

Перелік матеріалів Швидкість потоку біогазу

BreadBoard - Half Size	1x
Arduino Uno	1x
USB Cable A to B	1x
SparkFun HX711 - Load Cell Amplifier	1x
Load Cell Bar 100 g	1x
Jumper Wires Pack - M/M	6x
Male Headers Pack- Break-Away	3x

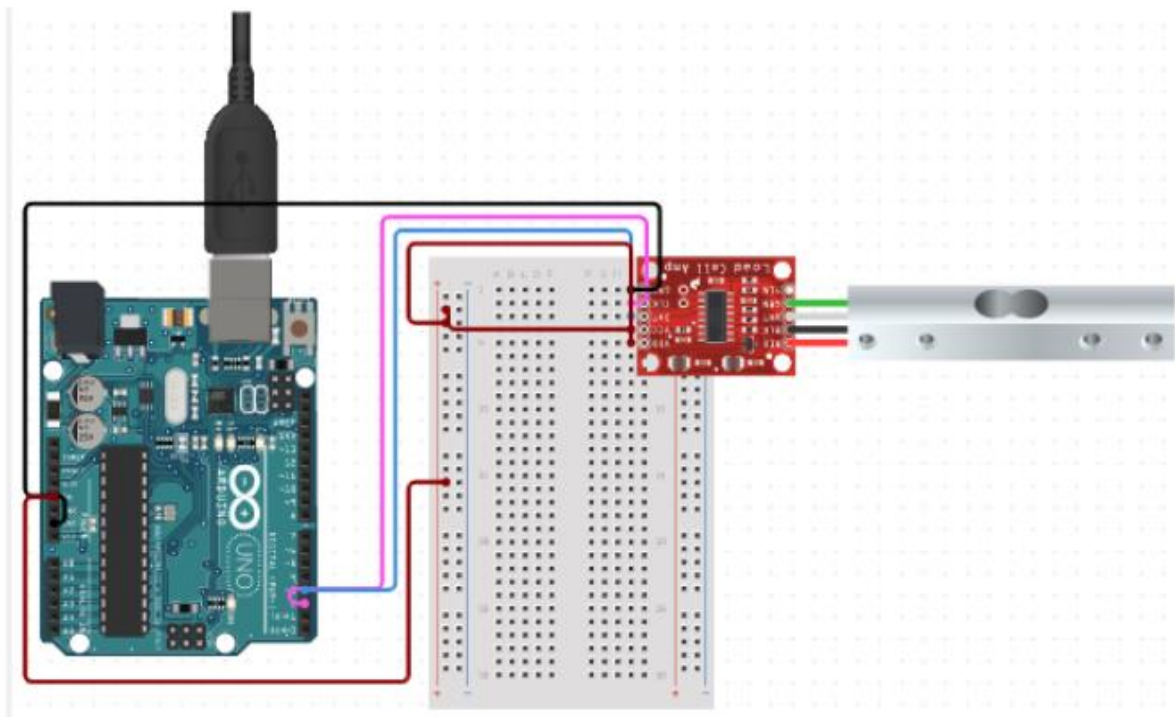


Рисунок 2.3 - Тензодатчик схеми Arduino та SparkFun HX711.

2.2. Аналіз вихідних параметрів системи.

Вихід — це інформація/команди, що надсилаються з плати Arduino до приводів, які повинні виконувати команди. Кожен привод повинен бути підключений до вихідного порту. Команди, яким потрібен насос або двигун (наприклад, базовий насос), також потребують екрану реле або MOSFET для обробки навантажень високої потужності.

2.2.1. Перистальтичний базовий насос.

Щоб підтримувати рівень рН на бажаному значенні, буде використовуватися перистальтичний насос, який може закачувати основу в мініатюрний біореактор. Як видно з таблиці 2.5 і рис. 2.5, перистальтичний базовий насос потребує додаткових компонентів для роботи. N-канальний MOSFET 60V 30A вже згадувався раніше. Однак настінний адаптер живлення – 12VDC 2A не згадується раніше.

Потрібен адаптер живлення, тоді як Arduino UNO/Arduino mega може подавати лише 5 В живлення в систему. 5 В недостатньо для живлення перистальтичного насоса, щоб він працював на повну потужність. Настінний адаптер живлення може гарантувати достатню напругу для перистальтичного насоса. Використовуватиметься перистальтичний насос із мікродвигуном 12 В. Цей насос має максимальну продуктивність 2-3 літри за хвилину і максимальний напір 2 метри. Перелік матеріалів і схему Arduino можна знайти в таблиці 4 і малюнку 7 відповідно. Схема Arduino схожа на схему Arduino двигуна постійного струму, що використовується для перемішування. Це буде показано на рис. 2.6.

Таблиця 2.4

Перелік матеріалів перистальтичний базовий насос

BreadBoard - Half Size	1x
Arduino Uno	1x
Peristaltic Pump 12V Micro Motor Pump	1x
Wall Adapter Power Supply - 12VDC 2A	1x
N-Channel MOSFET 60V 30A	1x
10K Ohm Resistor	1x
Jumper Wires Pack - M/M	6x

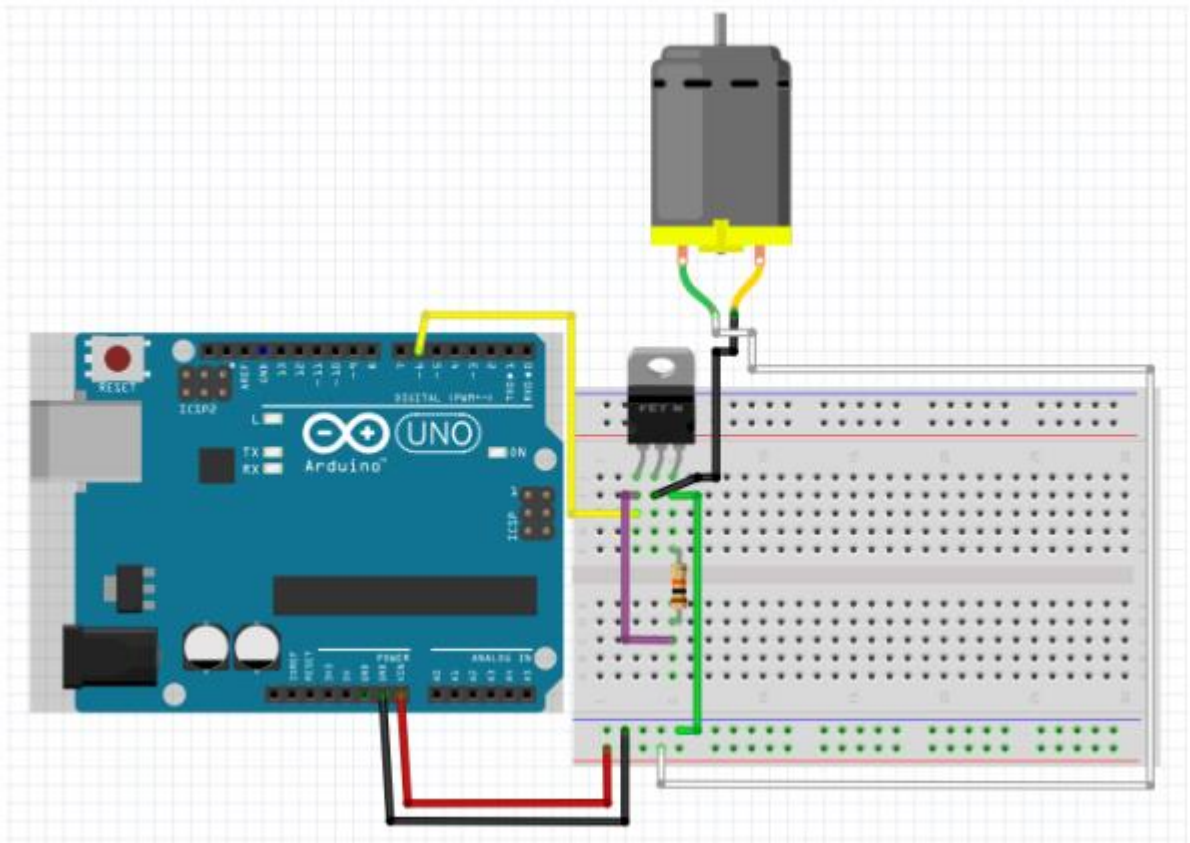


Рисунок 2.4 - Перистальтичний насос схеми Arduino

2.2.2. Активація/деактивація сигналізації про зміну температури.

Температурою керуватиме нагрівач, який потребує взаємодії з працівником лабораторії. Цей нагрівач є окремим ПД-регулятором, який керує температурою водяної бані. Однак, коли обігрівач виходить з ладу або він раптово зупиняється, лаборант повинен подавати тривогу. Щоб сповістити лаборанта, якщо обігрівач не працює належним чином, слід подати сигналізацію.

Сигналізація, яка буде розроблена, заснована на п'єзо. П'єзо — це звуковий пристрій, який можна запрограмувати на створення різних шумів за різних обставин. Piezo можна легко підключити до плати Arduino за допомогою одного з цифрових контактів. Після цього, за правильного програмування, п'єзогенератор може видавати шум, якщо температура виходить за межі його специфікації.

Деталі, необхідні для підключення п'єзо, можна знайти в описі матеріалів нижче. Разом з цим схема Arduino, необхідна для роботи п'єзо, показана на рис. 2.5.

Таблиця 2.5

Список матеріалів П'єзо сигналізація

BreadBoard - Half Size	1x
Arduino Uno	1x
USB Cable A to B	1x
Piezo	1x
1K Ohm Resistor	1x
Jumper Wires Pack - M/M	4x

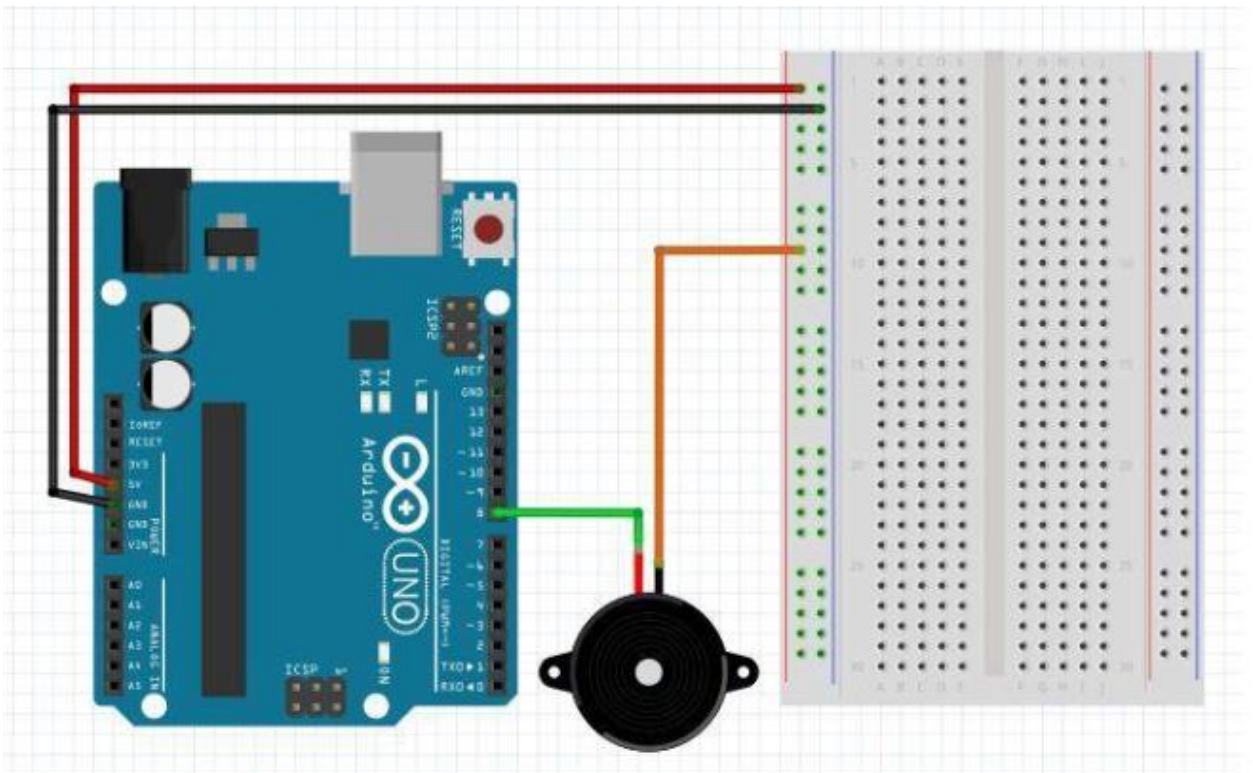


Рисунок 2.5 - П'єзосигналізація схеми Arduino.

2.2.3. Швидкість перемішування.

Швидкість перемішування, якою керуватиме плата Arduino, виконується двигуном-редуктором Micro Spur (6-12 V). Цей двигун ідеально вписується у верхню частину мініатюрної системи біореактора, яка вже розроблена. Для роботи двигуна знову потрібне живлення від настінного

блоку живлення, який використовується для перистальтичного насоса. Крім того, двигуну також потрібен MOSFET для посилення живлення від адаптера. Мотор-редуктор Micro DC вибрано за його розмір.

Таблиця 2.6

Перелік матеріалів прямокутний двигун постійного струму

BreadBoard - Half Size	1x
Arduino Uno	1x
Micro DC Gearmotor - (6-12V)	1x
Wall Adapter Power Supply - 12VDC 2A	1x
N-Channel MOSFET 60V 30A	1x
10K Ohm Resistor	1x
Diode Rectifier - 1A 50V	1x
Jumper Wires Pack - M/M	9x

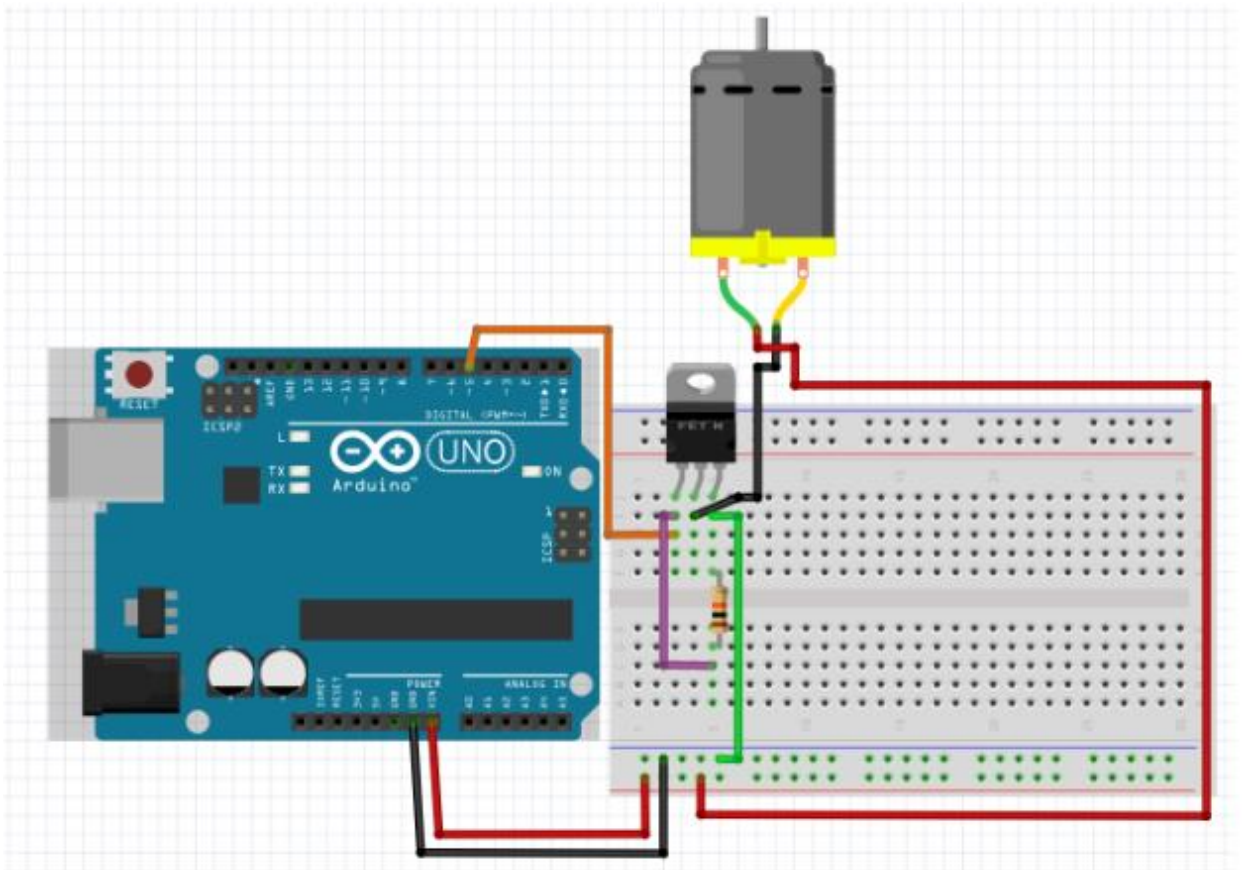


Рисунок 2.6 - Коло постійного струму перемішувального двигуна Arduino.

Розміри двигуна постійного струму складають 10x12x24 мм з віссю 20 мм, що дозволяє розмістити двигун постійного струму на мініатюрному біореакторі. Двигун постійного струму може забезпечувати швидкість від 100 до 150 обертів за хвилину, яка може бути адаптована платою Arduino.

Перелік матеріалів наведено в таблиці 2.6. З цими матеріалами схема Arduino на рис. 2.6. Крім того, двигун постійного струму та перистальтичний насос мають однакові схеми.

2.2.4. Корм.

Для того, щоб регулювати корм, 2 мл/день потрібно закачувати в систему та виводити з неї, щоб додати основні поживні речовини та витягти непридатну біоматеріал.

Таблиця 2.7

Подача номенклатури матеріалів

BreadBoard - Half Size	1x
Arduino Uno	1x
Peristaltic pump 12V	2x
Wall Adapter Power Supply - 12VDC 2A	1x
Relay shield 2 channel 5V	1x
Jumper Wires Pack - M/M	6x
Alligator wire clip	2x

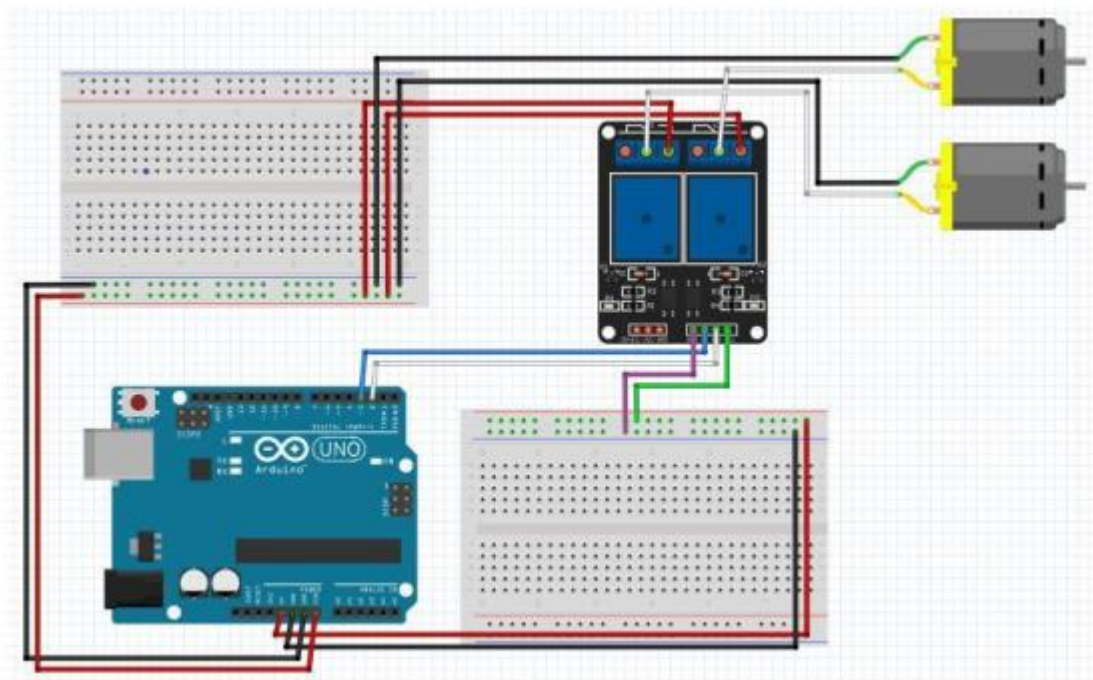


Рисунок 2.7 - Схема перистальтичних насосів Arduino та відповідного екрану реле.

Для цього будуть використані два ідентичних перистальтичних насоса. Так само, як насос і двигун постійного струму, два ідентичні насоси потребують джерела живлення 12 В для роботи. Крім того, робоче середовище насосів має бути від 0 до 40 градусів Цельсія з відносною вологістю повітря.

2.2.5. Програмне забезпечення Arduino.

Після того, як апаратне забезпечення зібрано, необхідно визначити необхідне програмне забезпечення. У Arduino IDE ескізи з кодами програмування можна завантажувати на мейнфрейм Arduino. Ці ескізи складаються з основних команд і операторів, які необхідно зрозуміти, щоб ескізи працювали. У наступному розділі пояснюються основні команди та оператори, які використовуються в коді, необхідному для використання системи керування.

Різні основні команди та оператори можна розділити на чотири окремі розділи, а саме: Коментарі, Оголошення, Void Loop і Void Setup. Коментарі: Коментарі в Arduino можна розділити на два різні розділи, які є багаторядковими та однорядковими коментарями. Багаторядковий коментар зазвичай починається на початку коду. Коментар обмежується символами /* і */. У більшості випадків багаторядковий коментар використовується для надання пояснення або коментаря до коду перед його відображенням.

Коментарі повністю ігноруються програмним забезпеченням під час компіляції та завантаження ескізу. Однорядковий коментар працює майже так само, як і багаторядковий. Однак ці коментарі можна знайти всередині функцій налаштування та циклу. Однорядковий коментар ініціалізується символами // і використовується для пояснення частин коду, які важче зрозуміти самі по собі. Будь-який написаний код після символів // також ігнорується під час завантаження коду на материнську плату Arduino [5].

Оголошення: оголошення в кодї `arduino` — це значення, які зберігаються для подальшого використання програмою.

Оголошення, які використовуються для розробки системи керування, та їхня мета:

- Визначення (`#define`) дозволяє програмісту дати назву постійному значенню перед компіляцією програми. Компілятор може замінити посилання на призначені константи визначеним значенням.

- Включити (`#include`) використовується для включення зовнішніх бібліотек у ваш ескіз. Зовнішні бібліотеки — це попередньо написані фрагменти коду, які допоможуть виконати код за допомогою плати `arduino`.

- Цілі числа (`int`) є основним типом даних для зберігання чисел. `Arduino Uno` може зберігати 16-бітове значення в ціле число. Після цього, викликаючи це ціле число на наступній фазі коду, `Arduino` використовуватиме збережене 16-бітне значення.

- Константа (`const`) — це кваліфікатор змінної, який може змінювати поведінку згаданої змінної, роблячи змінну «тільки для читання», що означає, що змінна не може бути змінена під час виконання коду. `Const` і `#define` можна використовувати з тією ж метою.

- `Float`, (`float`) — це тип даних для чисел із плаваючою комою, тобто чисел із десятковою комою. Функція `float` часто використовується замість функції `integer` через більшу роздільну здатність, яку має функція `float` порівняно з функцією `integer` [5].

Після того, як оголошення були зроблені, код входить у функцію налаштування

`Void Setup.`

Функція `void setup` запускається лише один раз. Метою налаштування `void` є підготовка плати `Arduino`, присвоєння платі значень і властивостей, які не змінюються під час її роботи. Функція налаштування використовує фігурні дужки, `{ та }`, щоб вмістити свій вміст.

Для кожної функції в Arduino потрібен відповідний набір цих дужок. Інакше, коли кількість фігурних дужок не збігається з кількістю функцій, видається помилка. Однією з функцій, які можна знайти в налаштуваннях void, є функція `pinMode`, яка налаштовує вказаний пін для введення або виведення.

Функція `pinMode` містить два параметри, а саме `pin` і `Mode`. `Pin` позначає номер контакту, режим якого ви бажаєте встановити, а `Mode` означає, якщо цей контакт потрібно використовувати як ВХІД або ВИХІД. Ще одна функція, яка використовується в налаштуваннях void, — `Serial.begin`, яка встановлює зв'язок між Arduino та монітором Serial. `Serial.begin` передає визначене значення параметру швидкості.

Після цього зв'язок між Arduino та Serial Monitor обмінюватиметься повідомленнями зі швидкістю передачі даних x біт на секунду. Найчастіше використовується значення 9600 біт на секунду.

Void Loop.

Void Loop схожий на налаштування void, але замість того, щоб запустити один раз, цикл працює безперервно, доки не буде натиснуто кнопку скидання на платі Arduino або вимкнеться живлення.

У циклі void можна вказати фактичні команди, які згодом будуть виконуватися платою Arduino. Однією з цих команд є `digitalWrite`, яка надсилає цифрове значення на PIN-код. Цифрові контакти мають лише два стани: увімкнено або вимкнено. У випадку `digitalWrite` використовується HIGH або LOW. Функція містить два параметри, а саме: пін, номер пін-коду, режим якого потрібно встановити, і вибір між HIGH і LOW.

Іншою функцією, яка багаторазово використовується в циклі void, є затримка. Назва функції говорить сама за себе. Затримка зупиняє програму на певний час у мілісекундах. За цей час нічого не відбувається. Два інших оператори, які можна знайти в сценарії, необхідні для роботи системи керування, це оператор «if» і оператор «for».

Ці оператори також використовуються в інших мовах програмування завдяки їх чіткій зручності. Оператор `if` перевіряє умову та виконує наступний оператор або набір операторів, якщо умова «істинна». Оператор `if` корисний, коли інформацію, необхідну з виміряних параметрів, потрібно перетворити в дію від насоса та п'єзосигналу. Оператор `for` можна використовувати для повторення блоку операторів. У системі керування розробкою оператор `for` можна, наприклад, використовувати для вимірювання різних значень рН зразків, щоб згладити значення знайденого рН.

2.3. Розрахунки базового насоса.

Слід окремо згадати код, необхідний для встановлення ПІД-контролера для базового насоса. Однак спочатку слід пояснити, що таке ПІД-регулятор. PID є аббревіатурою пропорційного, інтегрального та похідної. ПІД-регулятор — це регулятор, який містить елементи з цими трьома функціями. Це передбачає метод керування зі зворотним зв'язком, який використовує ПІД-регулятор як основний інструмент.

Щоб розробити ПІД-регулятор, слід виконати розрахунки щодо регулювання рН базовим насосом. Основою, яку використовує насос, є розчин NaOH (гідроксид натрію). Для розрахунку кількості NaOH, яку потрібно додавати через заздалегідь визначені проміжки часу, потрібно виконати кілька кроків.

Ці кроки:

1. Обчисліть початкову та кінцеву кількість молей іонів водню в резервуарі.
2. Від початкової кількості молей відніміть кінцеву, щоб отримати кількість молей іонів водню, нейтралізованих гідроксид-іонами.
3. Оскільки молярне відношення іонів водню до іонів гідроксиду та гідроксиду натрію становить 1:1:1 при нейтралізації, кількість молей

гідроксид-іонів і, таким чином, NaOH дорівнює різниці, обчислений на кроці 2.

4. Помножте кількість NaOH на його молярної маси, щоб отримати її масу.

5. Розділіть масу NaOH на концентрацію основи, щоб знайти кількість мл, яку потрібно додати до системи біореактора.

6. Розділіть мл, який потрібно додати, на швидкість потоку основного насоса, щоб визначити кількість секунд, протягом яких насос повинен бути увімкненим, щоб накачати кількість мл у систему.

Таблиця 2.8

Результати базових розрахунків насоса

pH	pH diff	g of NaOH	mL to add	Sec pump on
7	0	0	0.00	0.00
6.95	0.05	1.46E-08	0.29	0.18
6.9	0.1	3.11E-08	0.62	0.37
6.85	0.15	4.95E-08	0.99	0.59
6.8	0.2	7.02E-08	1.40	0.84
6.75	0.25	9.34E-08	1.87	1.12
6.7	0.3	1.19E-07	2.39	1.43
6.65	0.35	1.49E-07	2.97	1.78
6.6	0.4	1.81E-07	3.63	2.18
6.55	0.45	2.18E-07	4.36	2.62
6.5	0.5	2.59E-07	5.19	3.11
6.45	0.55	3.06E-07	6.12	3.67
6.4	0.6	3.58E-07	7.15	4.29

Наведені вище кроки виконуватимуться для різних інтервалів.

Припустимо, що об'єм рідини в мініатюрному біореакторі становить 30 мл. Це дорівнює 0,03 дм³ рідини в біореакторі. Крім того, молярну масу NaOH можна знайти за такою інформацією. $M_{m.Na} = 23$, $M_{m.O} = 16$, $M_{m.H} = 1$.

Тоді загальна молярна маса дорівнює 40. Швидкість потоку базового насоса дорівнює 1,67 мл/с. Таблиця 2.8 відображає результати, які були знайдені.

Кількість NaOH, яку потрібно додати, можна розрахувати, виконавши перші чотири кроки, зазначені вище. Спочатку обчисліть початкову та кінцеву кількість молей іонів водню в резервуарі.

Початкові молі водню можна розрахувати за такою формулою, де $in.pH$ — початковий pH, а V — об'єм мініатюрного біореактора:

$$\text{Initial moles of hydrogen} = 10^{-in.pH} * V$$

Крім того, кінцеві молі водень можна обчислити за тією ж формулою. Однак початковий pH потрібно змінити на кінцевий pH ($fin.pH$):

$$\text{Final moles of hydrogen} = 10^{-fin.pH} * V$$

Другим кроком є віднімання початкових молей водню від кінцевих, щоб отримати кількість молей іонів водню, нейтралізованих гідроксид-іонами. Для цього просто поєднайте формули 1 і 2.

$$\text{Neutralised hydrogen ions} = (10^{-fin.pH} * V) - (10^{-in.pH} * V)$$

Третій крок розрахунків полягає у визначенні молей NaOH, необхідних для стабілізації системи біореактора. Оскільки молярне співвідношення іонів водню та гідроксид-іонів до гідроксиду натрію становить 1:1:1 під час нейтралізації, кількість молей гідроксид-іонів i , таким чином, NaOH дорівнює різниці, обчисленій на кроці 2.

$$H \text{ (Hydrogen)} : OH \text{ (Hydroxide)} : NaOH \text{ (Natrium Hydroxide)} = 1 : 1 : 1$$

Четвертий крок полягає в обчисленні маси NaOH, необхідної для зміни pH системи до бажаного значення. Це можна зробити, помноживши кількість NaOH на його молярну масу, щоб отримати його масу. Молярна маса NaOH позначається символом $M_{m.NaOH}$ і дорівнює 40 g/mol

$$\text{Mass of NaOH} = ((10^{-fin.pH} * V) - (10^{-in.pH} * V)) * M_{m.NaOH}$$

П'ятий крок обчислень полягає в тому, щоб знайти кількість мл основи, яку потрібно додається для стабілізації системи біореактора. Для цього масу

NaOH потрібно розділити на концентрацію основи. Однак необхідно визначити концентрацію основи. Щільність, необхідна для створення логічної кількості мл, становить $5 * 10^{-5} \text{ g/L}$. Концентрація вказується C_{base} .

$$\text{mL of base to add} = \frac{((10^{-\text{fin.pH}} * V) - (10^{-\text{in.pH}} * V)) * M_{\text{m.NaOH}}}{C_{\text{base}}}$$

Після цього можна виконати шостий крок, щоб визначити кількість секунд насос повинен бути включений. Розділивши доданий мл на швидкість потоку основного насоса, можна знайти кількість секунд. Швидкість потоку вказується Q_{pump} .

$$\text{Seconds pump needs to be on} = \frac{((10^{-\text{fin.pH}} * V) - (10^{-\text{in.pH}} * V)) * M_{\text{m.NaOH}}}{C_{\text{base}} * Q_{\text{pump}}}$$

Результати, отримані за допомогою вказаних кроків, візуалізовано на рис. 2.8. Однак, перш ніж ці дані можна буде впровадити в необхідний код для базового насоса, слід виконати тести, щоб визначити шум рН-електрода. Після визначення шуму код може врахувати шум, щоб зробити PID більш точним. На малюнку нижче можна визначити, в яких точках і як довго базовий насос має бути активним.

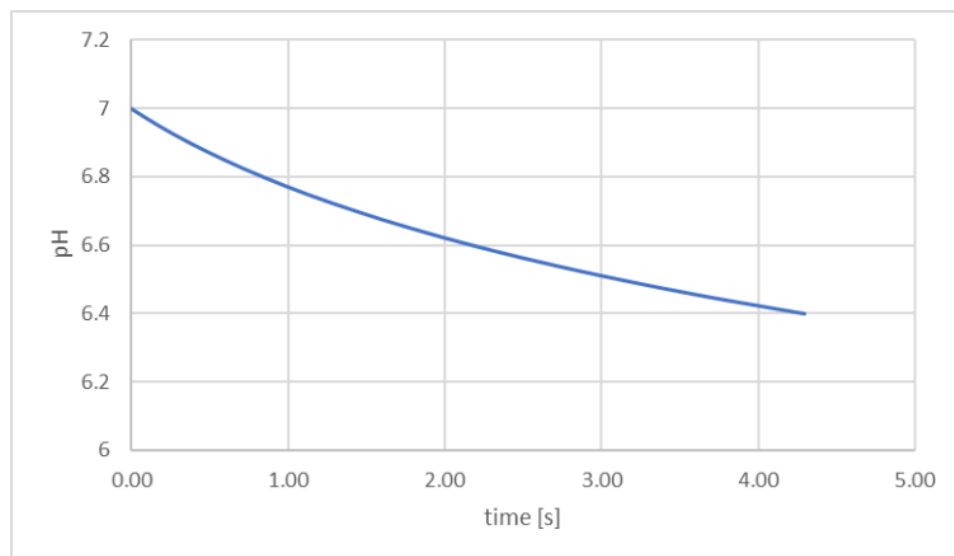


Рисунок 2.8 - Час, коли основний насос має бути включений, щоб стабілізувати MBR.

3 СПЕЦІАЛЬНА ЧАСТИНА

3.1 Розробка графічного інтерфейсу користувача

Для створення графічного інтерфейсу користувача використовується програмне забезпечення Processing. Processing — це гнучке програмне забезпечення, яке, як і Arduino, використовує ескізи та мову C++ для навчання програмуванню в контексті образотворчого мистецтва. Крім того, Processing можна безкоштовно завантажити майже на будь-якій пристрій. Processing і Arduino сумісні один з одним, що дозволяє передавати виміряні значення з плати Arduino в командне вікно Processing.

У Processing значення можна перевести в регульовані вимірювачі, що можна досягти, завантаживши бібліотеку вимірювачів, яку можна легко знайти в менеджері бібліотек Processing. На рис. 3.1 показано, як можуть виглядати лічильники за правильного використання. Обробка використовується для кількох цілей. Перша мета — візуалізація вимірюваних значень температури та рН.

Кожну зі змінних (наприклад, тип шрифту, колір або розмір) можна налаштувати відповідно до побажань користувача. Друга мета Processing полягала б у тому, щоб, як і у випадку з Arduino, сповіщати користувача, якщо температура або рН стає занадто низьким або занадто високим. Сигнал створюється за допомогою звукової бібліотеки, яка дає змогу завантажувати та відтворювати будь-які звукові файли mp3 у вікні команди обробки. Однак вибраний звук буде відтворюватися на пристрої, який використовується для моніторингу параметрів процесу.

Таким чином, один сигнал тривоги лунатиме від самої системи керування Arduino, а один – від пристрою з командними вікнами Arduino та Processing. Звуковий код ще не додано до решти програмного забезпечення обробки.

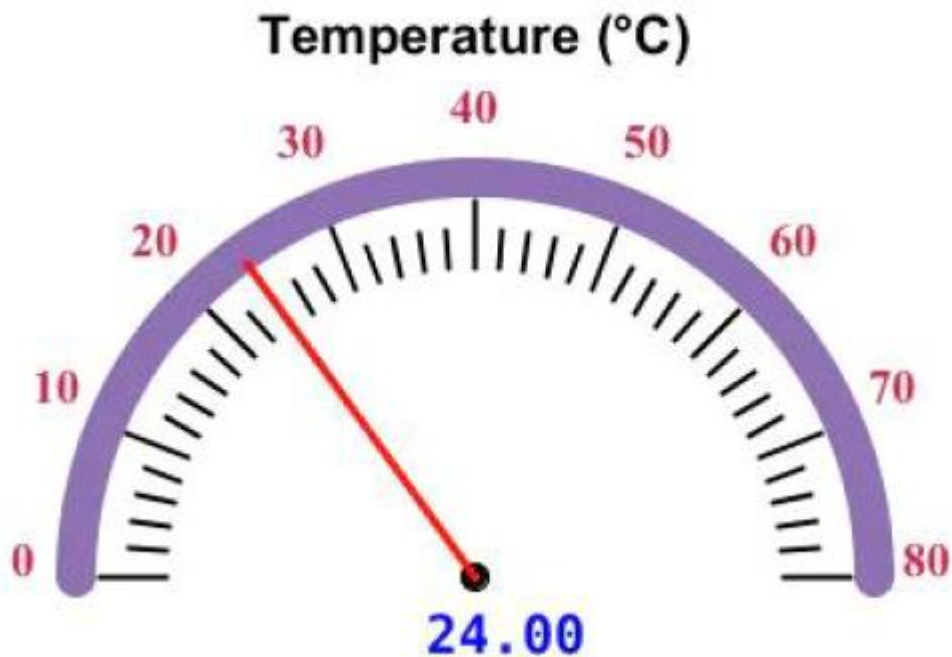


Рисунок 3.1 - Приклад вимірювача температури, виготовленого за допомогою Processing.

3.2. Розробка програми системи керування.

Arduino складається з усіх наданих раніше схем arduino. Деякі схеми Arduino можна комбінувати, щоб мінімізувати кількість використовуваних частин. Наприклад, і схема arduino перистальтичного рідинного насоса, і двигун постійного струму використовують настінний адаптер живлення. Тому частини будуть розділені за допомогою двох макетів.

Одна макетна плата керуватиме компонентами, які потребують додаткової напруги для роботи, а інша макетна плата керуватиме датчиками та п'єзосигналом. Компоненти, необхідні для створення системи керування Arduino, можна знайти в таблиці 3.1. Після цього, за допомогою Fritzing, схематичний огляд того, як побудована система керування, наведено на рис. 3.2.

Слід зазначити, що всі виконавчі механізми (двигун постійного струму, живильні насоси та базовий насос) візуалізуються як двигуни постійного

струму. Щоб уникнути плутанини, до схематичного огляду додано назви виконавчих механізмів і давачів.

Таблиця 3.1

Повний опис матеріалів системи керування

Arduino Uno	1x
Breadboard - Half Size	2x
USB Cable A to B	1x
Wall Adapter Power Supply - 12VDC 2A	1x
N-Channel MOSFET 60V 30A	2x
10K Ohm Resistor	2x
4.7K Ohm Resistor	1x
1K Ohm Resistor	1x
Relay shield 2 channels 5V	1x
DS18B20 temperature sensor	1x
Micro DC motor 100 - 1000 RPM (6-12V)	1x
Piezo	1x
Peristaltic Liquid Pump	3x
Silicone Tubing	6x
pH electrode	1x
BNC receiver Arduino	1x
Jumper Wires Pack - M/M	n
Jumper Wires Pack - M/F	n
Jumper Wires Pack - F/F	n
Male Headers Pack- Break-Away	n

Щоб побудувати наведений вище схематичний огляд, потрібно виконати багато кроків. Для виконання цих кроків у чіткому та логічному порядку було створено посібник, у якому чітко вказано, які кроки необхідно виконати, щоб зібрати всі окремі частини в одну робочу систему керування.

Компоненти та етапи їх складання розділені за макетною платою, на якій вони використовуються. Крім того, деякі частини довелося припаяти, щоб забезпечити кращий зв'язок між проводами та приводами та датчиками. Компоненти, які потребують спаювання, це три насоси, двигун постійного струму та термометр.

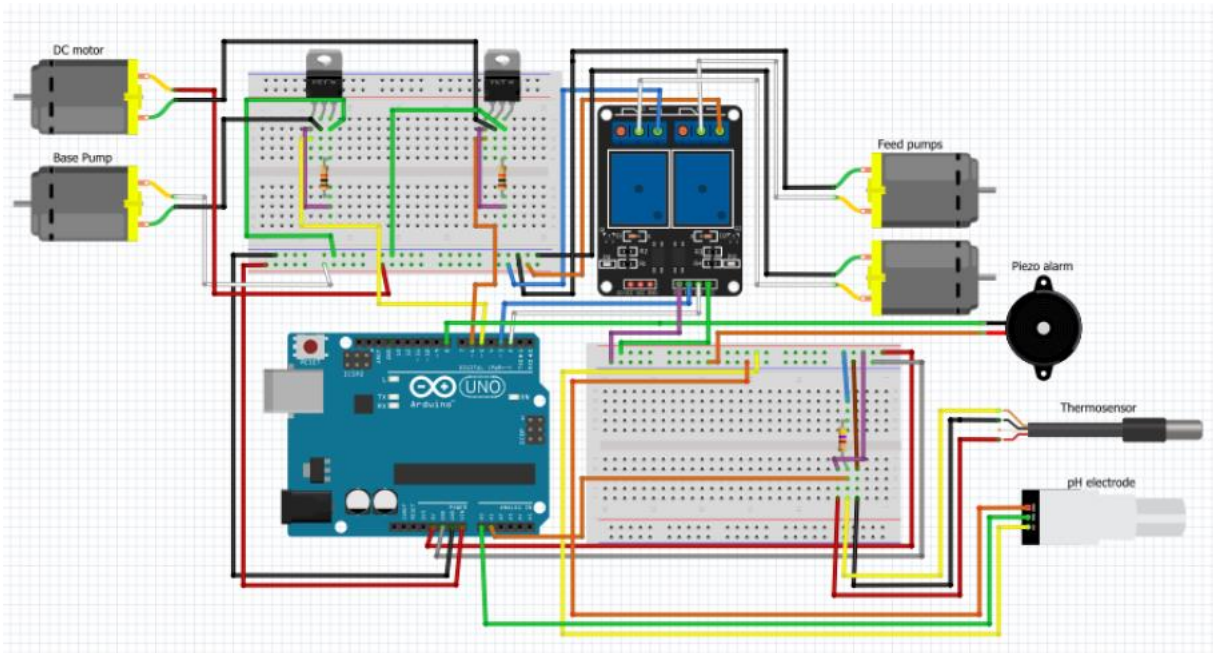


Рисунок 3.2 - Схематичний огляд остаточної схеми Arduino.

Розроблена система керування була побудована навколо Arduino UNO rev 2 Wi-Fi, яка, як було зазначено раніше, забезпечує 14 контактів цифрового входу/виводу та 6 контактів аналогового входу. Програма використовує 5 цифрових входів/виходів для керування трьома насосами (один для основи та два для подачі), п'єзосигналом сигналізації та мікродвигуном постійного струму.

Крім того, 2 аналогових входи використовуються для зчитування значень, які вимірюють датчик температури та електрод рН. Однак вага для визначення швидкості потоку біогазу не вимірюватиметься. Це більше не впливає на інші компоненти контрольної програми. Причиною невимірювання ваги в системі остаточного контролю є те, що тензодатчик не може бути прочитаний і відкалібрований належним чином. Тому програмне забезпечення та код не будуть включені до системи кінцевого контролю.

На рис. 3.3 показано зображення контрольної програми, зібраної в переносній коробці.

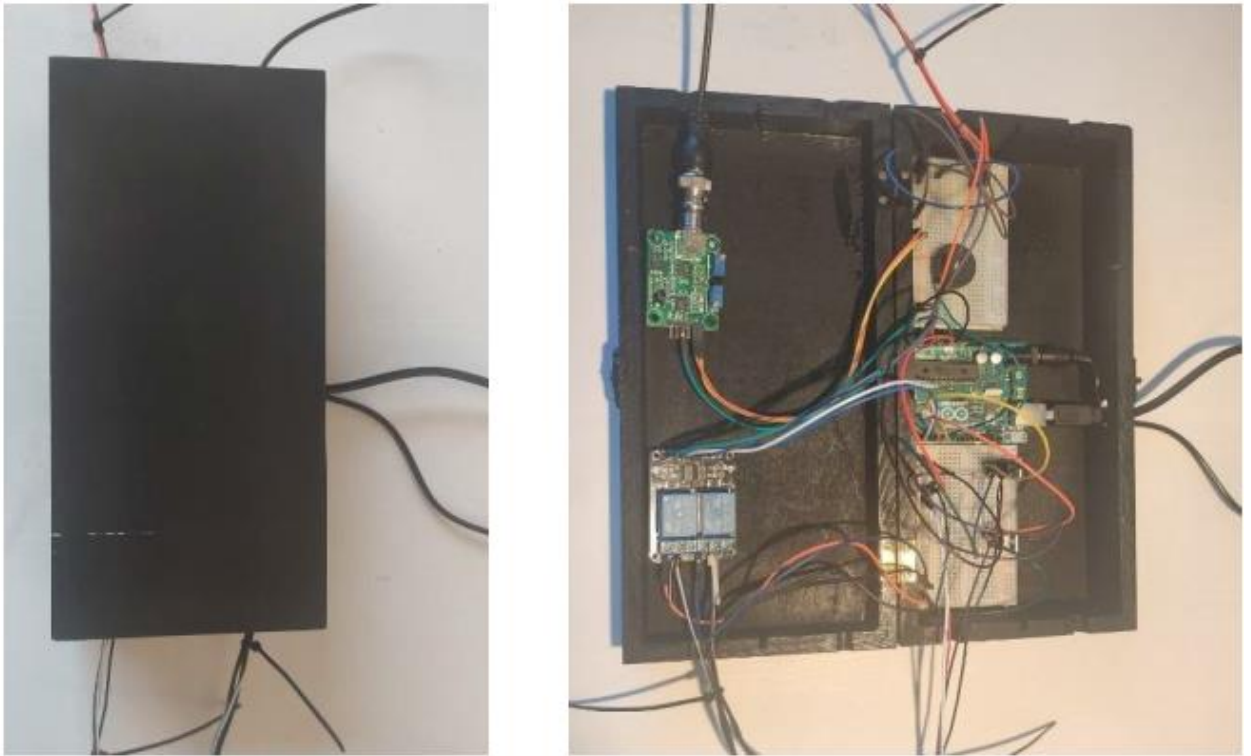


Рисунок 3.3 – Вигляд розробленої системи на базі Arduino.

Щоб створити графічний інтерфейс користувача, Arduino Uno безпосередньо підключається до Processing. Командне вікно обробки може відображати потрібні лічильники, пояснені в попередньому розділі. Відображаються такі параметри, як рН, температура та стан увімкнення/вимкнення двигуна. Остаточний макет візуалізації можна знайти на рис. 3.4. Модуль WiFi на Arduino Uno rev 2 Wi-Fi був призначений для використання для завантаження та керування системою Arduino без використання кабелю USB-B. Однак Arduino Uno rev 2 Wi-Fi використовує лише інтернет-модуль для надсилання даних на заздалегідь визначену IP-адресу.

Не вдалося знайти інформацію для зв'язку вхідних даних у веб-браузері IP-адреси з обробкою. Тому в майбутньому розвитку системи керування було прийнято рішення використовувати бездротове з'єднання.

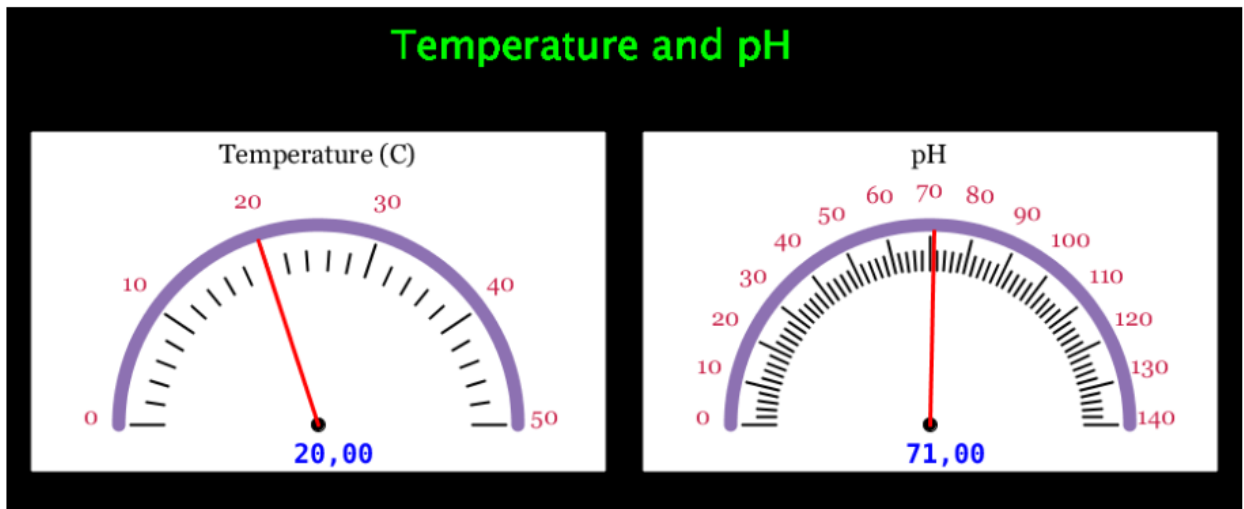


Рисунок 3.4 - Візуалізовані параметри в обробці.

Незважаючи на те, що спільнота Arduino мала багато добре задокументованих прикладів коду з відкритим кодом, написання та тестування можливих комбінацій коду було трудомістким процесом. Поточна функціональна версія містить 102 рядки коду, включаючи коментарі та анотації. Код розроблений на основі кодів, які були знайдені при роботі з кожним компонентом.

Однак поєднати кожен фрагмент коду в одному зразку виявилось складно. Це призвело до кількох непередбачених наслідків, про які йтиметься в обговоренні.

Щоб визначити, чи повною мірою програма керування відповідає потребам у вимірюванні та контролі відповідних параметрів процесу, було проведено моделювання. Термодатчик і рН-електрод тестуються з кількох аспектів. Кожен із цих аспектів перевіряється протягом п'яти хвилин (300 секунд). По-перше, були проведені тести на точність двох датчиків. У випадку з термосенсором вимірювання проводилися на воді, яка досягала кімнатної температури. Воду міряли тричі, щоб зменшити зміну незручностей. Таким чином, на малюнку 16 видно, що значення температури саморегулюється з різницею приблизно 0,065 °C.

Кроки, які можна побачити, викликані мілівольтами, які додаються до заздалегідь визначеної кількості бітів, поясненої раніше. Ці кроки можна скоротити, взявши середнє значення, наприклад, десяти вимірних значень і побудувавши отримані середні значення в залежності від часу. Приклад цього можна знайти пізніше, аналізуючи точність рН-електрода.

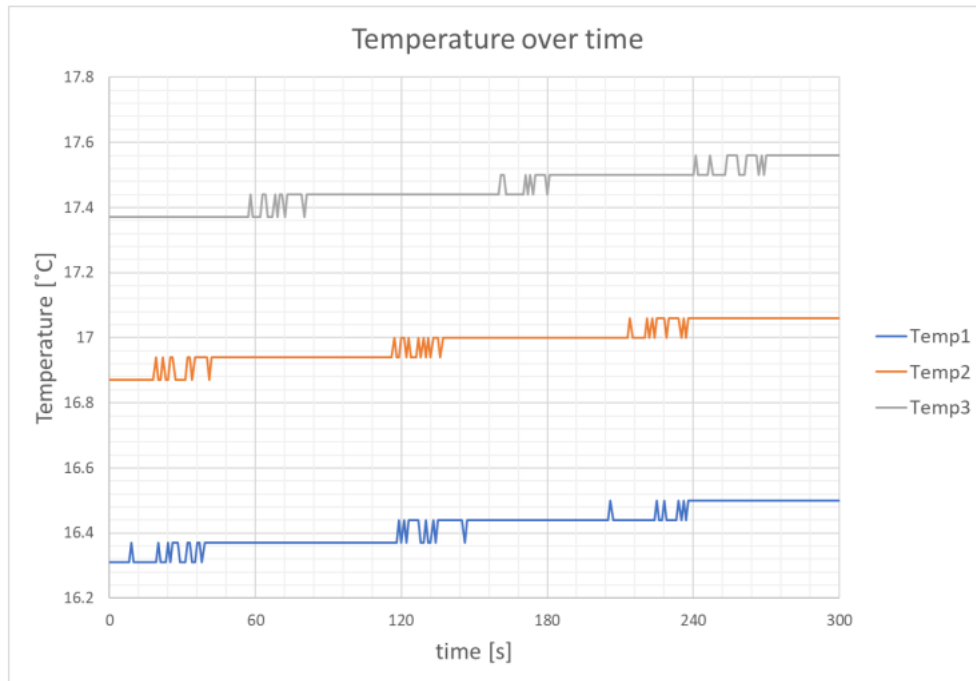


Рисунок 3.5 - Виміряна температура води, що пристосовується до кімнатної температури.

Окрім вимірювання зміни температури води, також було виміряно та проаналізовано вплив різкої зміни температури води. У двох випадках через одну хвилину 100 мл теплої води та 100 мл холодної води додавали в колбу, яка містила 200 мл води, щоб виміряти раптове підвищення та зниження температури. Після додавання води можна проаналізувати, наскільки швидко термометр може ефективно виміряти нову температуру води. На малюнку 17 показано збільшення та зниження температури відповідно. З результатів видно, що датчик температури здатний виміряти нову температуру після затримки в 30 секунд.

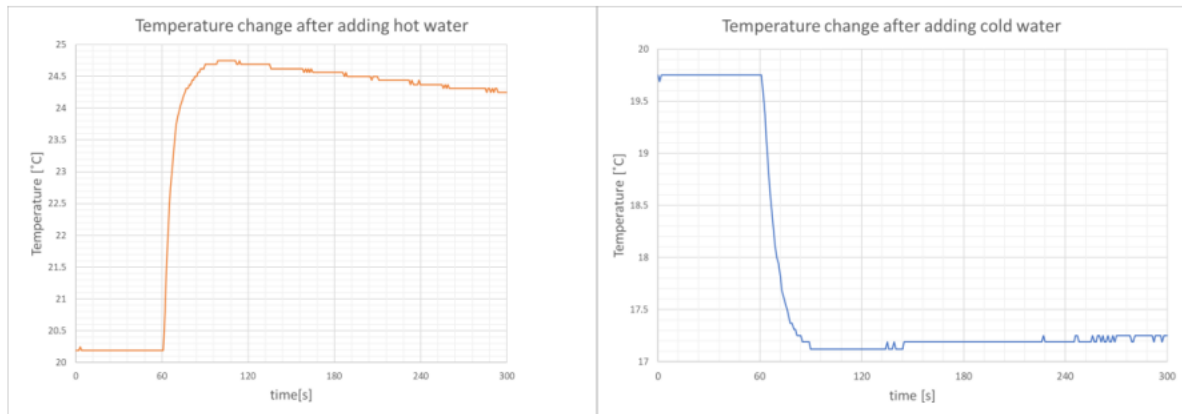


Рисунок 3.6 - Підвищення та зниження температури води.

Значення рН буде вимірюватися за допомогою програми, в якій використовується цикл «for – next». Це твердження for приймає десять вимірних значень і поміщає їх в одне середнє значення, щоб згладити вимірювання, згадані в попередньому параграфі. Перші два вимірювання рН проводили на нейтральному розчині з рН 7.

Першою ситуацією було вимірювання значення рН при безперервному перемішуванні рідини. Друга ситуація полягала у вимірюванні значення рН, коли рідину не перемішували. Як видно на малюнку 18, вийшло два різні результати. Перший тест, який виконував вимірювання під час безперервного перемішування рідини, має відносно постійне значення приблизно на рН 7. Однак другий тест, який проводив вимірювання без перемішування рідини, показує зниження виміряного значення рН протягом час. Зниження виміряного рН, коли рідина не перемішується, спричинене концентрацією H^+ навколо рН-електрода, яка стає вищою, коли рідина не перемішується.

Значення, знайдені під час перемішування рідини, можна додатково проаналізувати шляхом створення співвідношення сигнал/шум (SNR) протягом певного часу. На рисунку 19 показано SNR, яке вказує на те, наскільки вимірне значення відхиляється від того, що повинно було бути. Що можна побачити, це те, що шум не перевищує більше 1% від бажаного вихідного сигналу. Крім того, він навіть показує, що 80% вимірних значень навіть не перевищують 0,5% від бажаного результату.

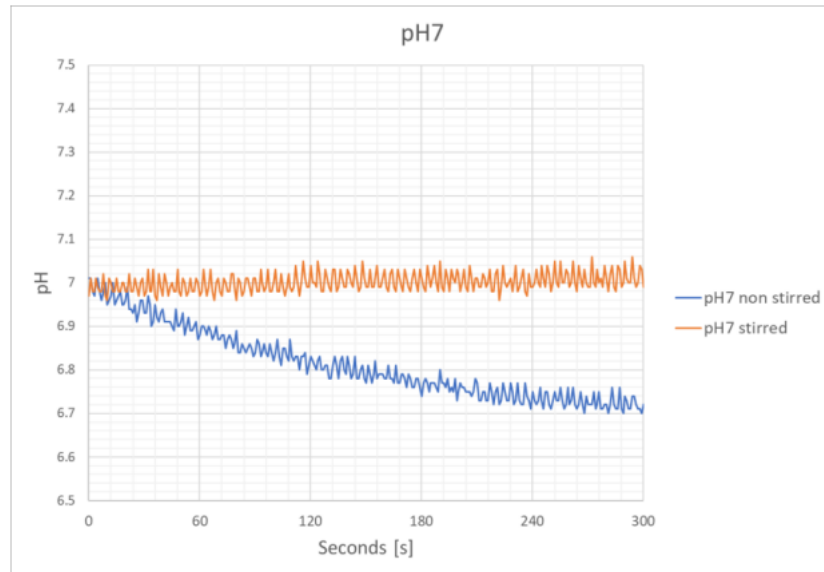


Рисунок 3.7 - Вимірювання рН розчину, що перемішується та не перемішується.

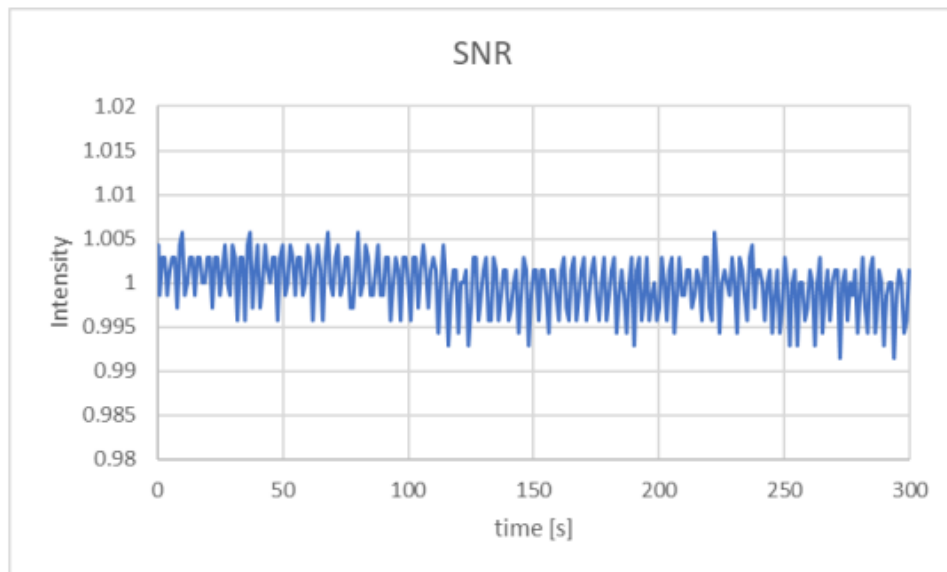


Рисунок 3.8 - Співвідношення сигнал/шум для рН-електрода.

Щоб визначити, скільки часу потрібно рН-електроду після раптового підвищення або зниження рН, було протестовано два сценарії. Перший сценарій полягав у додаванні кислого розчину після однієї хвилини вимірювання, що призвело до лівої сторони рис. 3.9. Можна побачити, що рН швидко падає приблизно з рН 7 до рН 5. Після цього, Електрод рН повільно відрегулюйте до 5,15, де виміряні значення знову стають стабільними. РН-

електроду потрібно приблизно 90 секунд, щоб знову стати стабільним. У другому сценарії буде використано той самий принцип із додаванням базового рішення. Праворуч на малюнку 18 видно, що рН підвищується до 9,15. Після цього вимірне значення стає відносно постійним через 60 секунд приблизно на рівні 9,05.

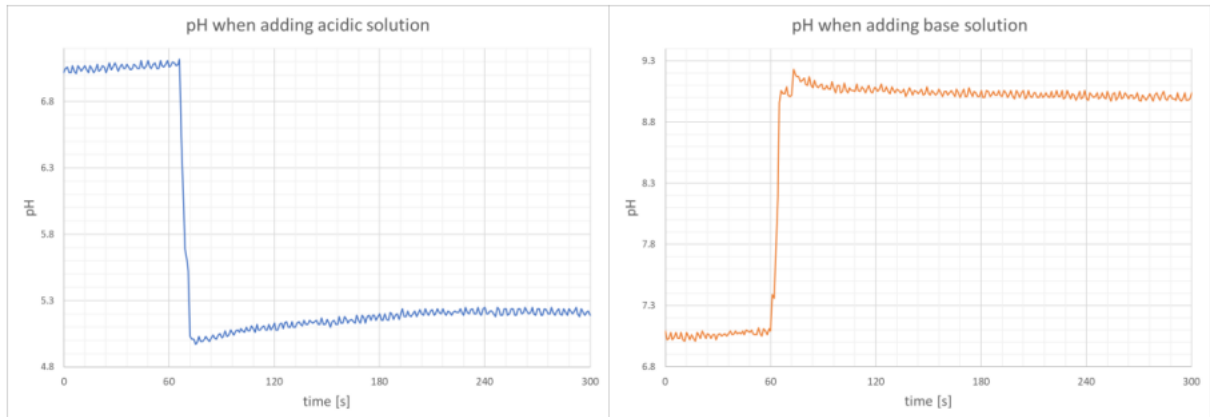


Рисунок 3.9 - Збільшення та зменшення значення рН при додаванні основи та кислоти.

Останнім тестом, який було виконано, було тестування поведінки електрода рН у поєднанні з базовим насосом. Проведено три тести, які складаються з двох звичайних і одного контрольного. По-перше, концентровану основу використовували для збору інформації про те, як швидко підвищиться рН після додавання 1 мл. Як видно на малюнку 19, після додавання 1 мл основи (чорна стрілка) рН одразу піднявся до 9,6. Після цього цю розбавлену основу використовували в наступних двох тестах, які мали показати, як зміниться рН після додавання нової композиції основи. Помаранчева лінія на малюнку 19 показує, що через одну хвилину до рідини додається основа (чорна стрілка).

Що можна побачити, це короткочасне підвищення рН від 6,5 до приблизно 7,1 рН. Через 100 секунд рН знову знизився до 6,8 рН. Знову додають 5 мл розрідженої основи для підвищення рН (помаранчева стрілка). Що можна було побачити після цього полягає в тому, що рН стане відносно постійним через 20 секунд після додавання другого зразка розведеної основи.

Щоб перевірити, чи не було підвищення рН випадковістю, було проведено контрольний тест. Сіра лінія на малюнку 21 вказує на підвищення рН після додавання основи через 60 (чорна стрілка) секунд і 180 секунд (сіра стрілка) відповідно. Можна побачити, що сіра лінія діє за тим же принципом, що й помаранчева лінія. Таким чином, можна стверджувати, що рН-електрод здатний достатньою мірою вимірювати різницю рН після додавання основи.

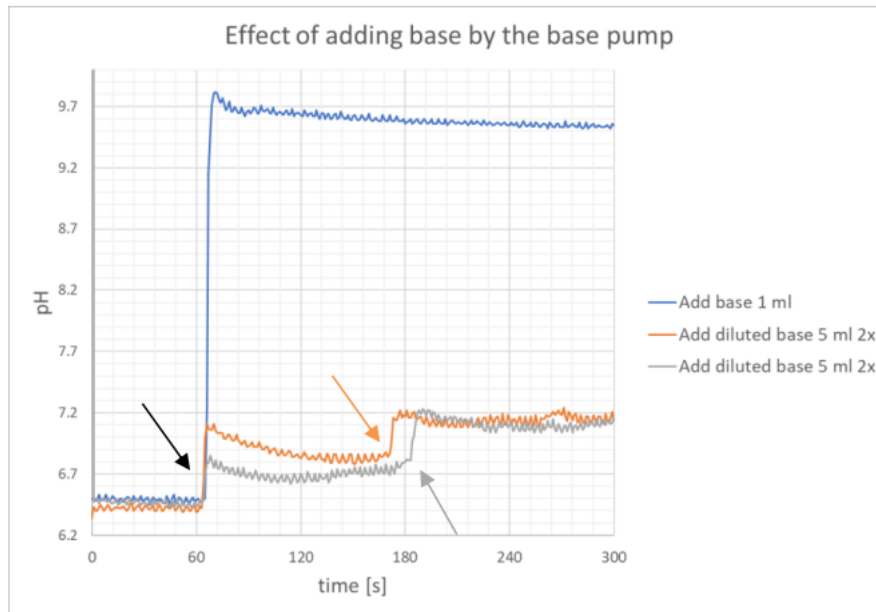


Рисунок 3.10 Тестування основного насоса

Результати цього проекту показують, що Arduino є корисним інструментом для моніторингу, керування та візуалізації відповідних параметрів процесу мініатюрної системи біореактора. Для підтвердження цього аргументу було проведено тестування в змодельованому середовищі. Тому ефективність системи контролю за даних обставин не може бути повністю визначена. Крім того, під час створення контрольного додатку в кожному з розділів було виявлено ряд труднощів і незручностей. По-перше, втрата напруги під час виконання завдань із системою керування Arduino.

Завдяки встановленим з'єднанням відбувається невелика втрата напруги протягом усього застосування керування. Кілька точок контрольної програми перевірено мультиметром. Виміряні значення показали, що деякі з компонентів працюють на рівні напруги, нижчому від очікуваного. Однак

втрати напруги становить лише 2% для джерела живлення 5 В (0,1 В) і приблизно 2,1% для джерела живлення 12 В (0,25 В).

Ці значення можуть незначно змінити вихід вимірних значень. Причиною втрати напруги може бути кілька причин. Першою причиною було б те, що ноутбук забезпечує лише напругу 4,9 В, а блок живлення настінного адаптера лише 11,75 В. Однак ймовірність того, що обидва пристрої мають відхилення приблизно на 2%, малоімовірна. Друга причина може полягати в тому, що резистори, які використовуються, занадто сильні для використання в електричних колах. Це пояснює постійну втрату напруги в контролері.

Останньою причиною можуть бути з'єднання, які виконуються за допомогою перемичок. Перехідні дроти підключаються лише одним контактом/головкою до іншого контакту/головки. Втрати напруги може статися через невеликий неправильно спрямований потік струму. Втрати напруги вимірювали за допомогою мультиметра. Однак слід враховувати і точність мультиметра. По-друге, ще однією проблемою, яка була виявлена під час роботи з апаратною частиною Arduino, був Wi-Fi shield. Як рекламується, Arduino Uno rev 2 Wi-Fi shield має мати можливість бездротового підключення до Інтернету робочого столу, з яким ви працюєте.

Однак під час додаткових досліджень екрану Wi-Fi виявилось, що бездротове з'єднання можна встановити лише в одному напрямку. Wi-Fi shield спочатку потребує підключення до робочого столу, щоб завантажити програмне забезпечення, яке виконується. Після цього Wi-Fi shield може завантажити виміряні значення на визначену IP-адресу.

Це означає, що робочий стіл повинен бути підключений до контролера Arduino в будь-який момент. Інший робочий стіл можна використовувати для читання значень, які надсилаються на визначену IP-адресу. Однак його неможливо (поки що) знайти, якщо присутня можливість перевести значення IP-адреси у вікно команд обробки для візуалізації цих значень. Подальші дослідження це питання слід провести, щоб перевірити, чи можливе

бездротове з'єднання в обох напрямках. По-третє, виникли труднощі при розробці програмного забезпечення Arduino. Як пояснювалося в попередньому розділі, мова Arduino працює з настройками void і циклом void.

Ці команди можна виконати лише один раз. Щоб дозволити системі керування працювати, кожен із окремих фрагментів коду з власною командою циклу void мав бути об'єднаний в один фрагмент коду. Проблеми, які виникли після об'єднання фрагментів коду, полягали у функції затримки. Завдяки функції затримки кілька інших функцій Arduino також затримують виконання своїх завдань. Це побічно означає, що, якщо виміряна температура буде нижче або вище робочого діапазону, відбудеться затримка в 60 секунд, протягом якої рН неможливо виміряти та відрегулювати. Те ж саме стосується і навпаки: якщо вимірне значення рН нижче робочого діапазону, температуру виміряти неможливо. Можливим вирішенням цієї проблеми є налаштування робочих діапазонів таким чином, щоб утворився невеликий буфер. Прикладом цього може бути встановлення робочого діапазону рН від 6,7 до 6,8. Коли температура опуститься нижче 34 градусів за Цельсієм, почнеться хвилина затримки.

До початку затримки рН ще буде вище 6,8. Імовірність того, що рН буде нижче 6,7 після хвилини затримки, дуже мала. Знову ж таки, те саме може працювати навпаки. Таким чином, проблема мінімізована до фактора, який майже не є проблемою для роботи мініатюрного біореактора. Цю проблему можна вирішити кількома способами. Перш за все, до системи можна додати додатковий щит arduino, який дає змогу керувати різними завданнями на різних щитах arduino, щоб завдання не заважали одне одному.

Крім того, можна створити частину інтервального коду, який також використовується для живильних насосів, щоб зробити коди для вимірювання рН і температури більш незалежними від функції контуру. Іншою проблемою, яка виникла при поєднанні коду та апаратного забезпечення arduino, було ПІД-регулювання базового насоса. Результати

розрахунків для PID основного насоса показують, що насос повинен бути активним протягом кількох секунд, щоб додати основу в мініатюрну систему біореактора.

Кількість секунд, протягом яких основний насос має бути активним, становить приблизно від 0 до 4,5 секунд. Однак розраховані значення базуються на концентрації гідроксиду натрію приблизно $5,0 \cdot 10^{-5} \text{ g L}$. Ці розрахункові значення недостатньо описують хімічні можливості, які слід враховувати. Концентрація NaOH до такої міри не може бути використана для цього процесу. Щоб досягти такої низької концентрації, початковий NaOH потрібно розбавити кілька разів, перш ніж концентрація навіть наблизиться до бажаних значень концентрації. Якщо розведення NaOH недостатньо до такої міри, слід знайти інший розчин. Одним із таких рішень була б зміна апаратного забезпечення базового насоса. Базовий насос має швидкість потоку приблизно $1,67 \text{ mL s}$. Якщо швидкість потоку можна зменшити в 1000 разів, рівень концентрації може збільшитися в 1000 разів. Концентрацію $5,0 \cdot 10^{-2} \text{ g L}$ було б набагато легше розвести з розчину NaOH. Крім того, з результатів розрахунків можна побачити, що кількість мл, яку необхідно додати до мініатюрного біореактора, у деяких випадках перевищує 4 мл. Більше 4 мл на загальний об'єм 30 мл може негативно вплинути на роботу біореактора.

Ще одна причина змінити концентрацію, щоб зменшити кількість мл, які потрібно додати до мініатюрного біореактора. Про розрахунки базового насоса також слід зазначити, що вони базуються на водному середовищі. Зазвичай, коли рН регулюється в системі, рідина в системі не на водній основі, а містить буфер із уже використаних хімікатів, який може вплинути на ефект регулювання рН у мініатюрному біореакторі. Таким чином, можна припустити, що для регулювання рН рідини в мініатюрному біореакторі можна використовувати більш високу концентрацію, ніж та, що знайдена в результатах розрахунків базового насоса. Крім того, ще однією проблемою, яка була виявлена під час розробки програмного забезпечення, було

поєднання програмного забезпечення Arduino та програмного забезпечення обробки.

Вимірювання, отримані за допомогою Arduino, повинні бути переведені таким чином, щоб Processing міг зчитувати виміряні значення.

Однак деякі аспекти перекладу повинні бути відомі заздалегідь, щоб він пройшов гладко. По-перше, обробку не можна використовувати, коли монітор в Arduino записує виміряні значення параметрів мініатюрного біореактора. По-друге, в Arduino виміряні значення параметрів мають бути надруковані у вигляді рядкового значення, яке розділяє значення комою, яке потім можна надіслати до обробки. Тоді значення рядка може бути відокремлене в рамках обробки шляхом призначення кожного окремого значення рядка окремому лічильнику. Однак у деяких випадках у Processing виникають труднощі з перекладом рядкового значення, що призводить до виключення індексу масиву за межами.

Ця помилка негайно зупиняє обробку значень з Arduino. Зупинка обробки може призвести до майбутніх помилок у моніторингу відповідних параметрів процесу. Іншою проблемою, яка виникла під час використання обробки, було обмеження команди `m.updateMeter(int(value))`, яка може оновлювати лічильник лише за допомогою цілого значення. Результатом цього є те, що глюкометр відображає лише значення рН і температури як цілі числа. Отже, рН може відображатися лише як 6 або 7.

Для досягнення точного вимірювання рН буде помножено на 10. Таким чином, рН буде показано як ціле число між значеннями 65 і 75, що дуже далеко точніше, ніж лише 6 і 7. У розділі 8. Ринкові опції можна знайти чіткий перелік усіх витрат. Ці результати збігаються з результатами попередніх досліджень. Однак витрати базуються на компонентах, необхідних для одного окремого мініатюрного біореактора. GPPV спрямована на створення системи керування, яка також могла б контролювати роботу чотирьох багатопаралельних мініатюрних біореакторів. Подальший розвиток системи керування для керування чотирма

багатопаралельними роботами біореакторів не впливатиме на ціну лінійно. Причини цього наступні.

При переході від одного до чотирьох мініатюрних біореакторів плату Arduino Uno слід замінити на плату Arduino Mega. Плата Arduino Mega має більше цифрових входів/виходів і більше аналогових входів, які можуть легко контролювати та контролювати чотири багатопаралельні біореактори. Плата Arduino Mega дорожча, ніж плата Arduino Uno. Крім того, витрати на рН-електрод і мікродвигун постійного струму можна помножити на чотири, тоді як кожному мініатюрному біореактору потрібен власний рН-електрод і мікродвигун постійного струму.

Однак витрати на термодатчик, п'єзосигналізацію та тензодатчик 100 г залишаються незмінними. Термодатчик і сигналізація контролюють зовнішню водяну ванну. Вода з цієї ванни транспортується до кожного з чотирьох багатопаралельних мініатюрних біореакторів. Крім того, після того, як виробництво біогазу буде виконано, біогаз буде зібрано в колбу над тією ж тензодатчиком, яка потім розраховує загальний вихід продукції. Якщо об'єднати цю інформацію, то можна стверджувати, що система керування, яку можна розробити для роботи чотирьох багатопаралельних мініатюрних біореакторів, є відносно дешевшою, ніж система керування для роботи одного окремого мініатюрного біореактора.

Крім того, коли потрібно більше деталей, з'являється можливість купувати запчастини оптом, що також може призвести до зниження загальних витрат.

У цьому проекті було розроблено керуючу програму для мініатюрної системи біореактора, яка виробляє біогаз за допомогою відкритої електронної платформи Arduino. Розроблена програма керування підходить для візуалізації та моніторингу відповідних параметрів процесу. Однак слід зробити незначні адаптації, щоб зробити його придатним для керування мініатюрною системою біореактора. По-перше, контрольна програма виявилася ефективною для моніторингу визначених параметрів. Контроль

відповідних параметрів процесу був математично прийнятним. Тим не менш, хімічна точка зору припускає, що PID базового насоса слід відрегулювати, щоб він був достатнім для контролю значення рН у системі біореактора. Візуалізація відповідних параметрів за допомогою розробленого графічного інтерфейсу виявилася успішною.

Крім того, представлена система керування є легкою для створення та є економічно ефективною альтернативою порівняно з іншими системами керування, які доступні для біореакторів звичайного масштабу. Остаточна система керування Arduino була розроблена відповідно до кількох етапів, а саме: визначення налаштувань системи, апаратного забезпечення Arduino, програмного забезпечення Arduino та графічного інтерфейсу користувача. Виявилось, що кожен із цих кроків має свій власний рівень складності.

Налаштування системи було відносно легко визначити, тоді як більшість літератури про біореактори та мініатюрні біореактори можна знайти безпосередньо у всесвітній мережі. Крім того, налаштування системи вже були заздалегідь визначені GPPB з використанням інформації про розроблений мініатюрний біореактор. Апаратне та програмне забезпечення Arduino було дещо складніше визначити. У випадку з обладнанням Arduino сталася непередбачена стагнація роботи системи керування через поставку непрацюючих частин та несвоєчасну поставку деталей.

Це досягалося шляхом зосередження на тому, що вже було відомо, а потім підготовки до того, що потрібно було зробити, щоб зробити процес проектування відносно легшим. Після цього було визначено програмне забезпечення Arduino. Кожну частину програмного забезпечення з різних компонентів було знайдено та створено, після чого кожна з цих окремих частин потрібно було об'єднати в один фрагмент коду. Поєднання цих частин призвело до кількох незначних проблем, які вже згадувалися під час обговорення.

Через несвоєчасну/неправильну доставку Інтернет-щит, графічний інтерфейс користувача був визначений на дуже пізній стадії процесу

проектування. Отже, графічний інтерфейс користувача не має бездротового підключення до контролера Arduino.

Подальші дослідження повинні вказати, чи є зчитування значень з IP-адреси та їх передача до обробки цінним майбутнім вдосконаленням дизайну. Крім того, можлива майбутня робота над темою повинна включати кілька додаткових аспектів, один з яких є наступним. Випробування, проведені з рН-електродом, слід повторити. Однак тепер це має бути зроблено в мініатюрному біореакторі, який містить рідину, схожу за характеристиками порівняно з можливою рідиною, яка буде використовуватися для анаеробного зброджування.

Ці тести правильно вкажуть, як рН можна регулювати за допомогою заданої концентрації NaOH (гідроксиду натрію). Крім того, система керування мініатюрним біореактором у його поточному стані (тобто оснащеному вимірюванням температури, рН, швидкості змішування та швидкості потоку біогазу) може мати обмежене застосування. Для розширення функціональних можливостей системи управління можна додати додаткові функції. Прикладами цього є вимірювання щільності клітин, датчик навантаження або навіть рівень DO.

З додатковими функціями, система контролю може бути використана для кількох інших дослідницьких цілей, що в разі збільшення інтересу можливих інвесторів. Другою рекомендацією щодо майбутньої роботи над системою керування буде модернізація кількох датчиків і виконавчих механізмів. Використовувані в даний час датчики і виконавчі механізми повністю виконують ціль з хорошою точністю.

Однак можна використовувати/купити кращі версії кожного компонента, щоб ще більше підвищити точність, що згодом також може призвести до кращого виробництва біогазу. Іншим важливим аспектом проекту було створення системи керування, яка була б відносно дешевшою, ніж системи керування, доступні для систем біореакторів.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Долікарська допомога при шоку

Розглянемо порядок, що визначає механізм надання домедичної допомоги при підозрі на шок не медичними працівниками.

Шок – це стан між життям та смертю; загальний тяжкий розлад життєво важливих функцій організму, спричинений порушенням нервової регуляції життєво важливих процесів; характеризується розладами гемодинаміки, дихання, обміну речовин.

Ознаки шоку у постраждалого:

- бліда, холодна і волога шкіра;
- слабкість;
- неспокій;
- сухість в роті, відчуття спраги;
- часте дихання (більш ніж 20 вдихів за хвилину);
- порушення свідомості; непритомність.
- Причинами виникнення шоку можуть бути:
- зовнішня кровотеча;
- внутрішня кровотеча;
- травми різного генезу;
- опіки;
- серцевий напад тощо.

Послідовність дій при наданні домедичної допомоги постраждалим при підозрі на шок не медичними працівниками:

- 1) переконатися у відсутності небезпеки;
- 2) провести огляд постраждалого, визначити наявність свідомості, дихання;
- 3) викликати бригаду екстреної (швидкої) медичної допомоги;

4) якщо у постраждалого відсутнє дихання, розпочати проведення серцево-легеневої реанімації;

5) усунути причину виникнення шокового стану: зупинити кровотечу, іммобілізувати перелом тощо;

б) надати постраждалому протишокове положення:

а) перевести постраждалого в горизонтальне положення;

б) покласти під ноги постраждалого ящик, валик з одягу тощо таким чином, щоб ступні ніг знаходились на рівні його підборіддя;

в) підкласти під голову постраждалого одяг/подушку;

г) вкрити постраждалого термопокривалом/покривалом;

7) забезпечити постійний нагляд за постраждалим до приїзду бригади екстреної (швидкої) медичної допомоги;

8) при погіршенні стану постраждалого до приїзду бригади екстреної (швидкої) медичної допомоги повторно зателефонувати диспетчеру екстреної медичної допомоги.

4.2 Розробка, оформлення кімнати для психологічного розвантаження працівників

Напружений ритм життя шкільних працівників, інтенсифікація їх праці на тлі низької рухової активності породжують відомий дисонанс між вимогами, що пред'являються до інтелекту, емоційній сфері, і порівняно малої фізичним навантаженням. Робота нервової системи в подібному режимі часто веде до підвищеного напрузі, невміння розслабитися, виходити з напруженого стану, знаходити психічну рівновагу. У більшості випадків у людей, схильних "хворобам століття" - неврозів, гіпертонії та ішемічної хвороби серця, - можна фіксувати підвищену м'язову напруженість, втрату навичку довільного розслаблення м'язів. Крім того, інтенсивне навчання деяких предметів викликає необхідність зняття психічної напруги. Все це ставить перед психологічною службою школи нагальну задачу створення кабінету психологічного розвантаження (КПР).

Кабінет психологічного розвантаження в школі працює в п'яти режимах:

- Психологічна розвантаження співробітників і школярів після напруженої роботи в кінці робочого (навчального) дня або в спеціально відведений для цього час.
- Психологічний настрій (мобілізація) тих співробітників і школярів, які насилу включаються в напружений ритм роботи на початку робочого дня, навчання навичкам мобілізації в стресі (контрольна, іспит і т.п.).
- Зняття психологічного навантаження викладачів і школярів відповідно до курсу, призначеним психотерапевтом.
- Психопрофілактична робота з практично здоровими вчителями та школярами (навчання методам релаксації, медитації, аутогенного

тренування, навичкам безконфліктного спілкування, тренінг спілкування і т.д.).

- Забезпечення процесу інтенсивного навчання, включаючи методи суггестопедии, релаксопедії, гіпнопедії, а також використання кімнати психологічного розвантаження як експериментальної бази для розробки нових методів навчання.

Питання про можливість і необхідність відвідування сеансів психологічного розвантаження вирішується співробітниками психологічної служби на основі даних психодіагностики в залежності від характеру впливу. Для індивідуальної роботи відводиться від 5 до 30 хвилин на одну людину, на групу - 60 хвилин. При наявності в КПП 12-15 місць його пропускна здатність становить 60-80 чоловік у зміну, а курсове лікування можуть отримати одночасно до 200 чоловік, оскільки заняття проводяться два-три рази на тиждень. При проведенні занять інтенсивного навчання пропускна можливість КПП знижується, однак особи, які проходять інтенсивний курс, одночасно випробовують і психопрофілактичний вплив.

До облаштування КПП пред'являються певні технічні вимоги. Кабінет повинен складатися з двох зв'язаних між собою кімнат. Перша кімната є одночасно і робочим кабінетом психологічної служби. Сюди винесена вся апаратура, обслуговуюча сеанси психотерапії і заняття інтенсивного навчання. Крім того, з операторської через спеціальне дзеркальне скло з одного боку можна проводити невиключене спостереження за поведінкою відвідувачів в психотерапевтичному залі. Такий зал обладнується 10-15 м'якими кріслами з високими підголовниками і вмонтованими в них роз'ємами для підключення індивідуальних навушників. Площа залу повинна бути не менше 40 кв. м, стелі повинні бути досить високими, щоб відвідувачі не відчували себе в тісноті і щоб в затемненому залі у них виникало почуття усамітнення.

Інтер'єр кабінету психологічного розвантаження повинен викликати у відвідувачів позитивні емоції, надавати сприятливий вплив на організм

людини. Шумоізолювані стіни КПП повинні бути блакитного або світло-зеленого кольору. В якості будівельного матеріалу використовуються перфопліти або акустична штукатурка, в декоративній обробці застосовуються шкірозамінник, дерматин та інші матеріали, за допомогою яких можна створити затишок, що позитивно впливає на настрій людини.

Психотерапевтичний зал також повинен бути обладнаний автоматичною системою затемнення вікон, екраном, світломузичним пристроєм, акустичними колонками, апаратами для іонізації, зволоження та кондиціонування повітря, великим акваріумом з підсвічуванням і технічними засобами управління станом людини (ТСУС), запропонованими С. М. Зоріна. Будучи головною частиною керованої цветозвукового середовища, ТСУС являє собою поліфункціональну систему для реалізації специфічних аудіовізуальних впливів з метою управління увагою, релаксацією, активізацією, а також для зниження рівня антисуггестивних бар'єрів. У систему ТСУС входить два функціональних блоку:

- Установка керованого колірною клімату. Вона має вигляд рами з алюмінієвого сплаву П-образного профілю, розташованої по периметру кімнати у стелі. П'ять груп ламп накачування, змонтовані на цій рамі, дозволяють здійснювати управління яскравістю і спектральним складом освітлюваної аудиторії. Всі світильники спрямовані в стелю, щоб забезпечувати м'який, розсіяне світло в залі. Освітлення психотерапевтичного залу може змінюватися як вручну оператором, так і автоматично, за заздалегідь розробленою для кожного виду впливу програмою.

- Светодінамічна система (СДС). На відміну від установки керованого колірною клімату, що змінює лише яскравість і кольоровість освітлення, СДС дозволяє здійснити на екрані синтез керованих параметрів світлодинамічних символів. Ці символи, змінюючи за бажанням оператора або за заданою програмою свої обриси, колір, яскравість, насиченість, швидкість і спрямованість руху, можуть з'єднуватися в складні, що

розвиваються у часі динамічні композиції, що мають багатопланове застосування.

Все управління психотерапевтичним сеансом здійснюється з операторської кімнати, де на стелажах розташовані стереомагнітофони з Мікшерський пультом, діа- та кінопроектори, що забезпечують за допомогою спеціально підібраних слайдів, кінозарісовок і музики емоційно-естетичний вплив на людину.

ВИСНОВКИ

У цьому проекті було розроблено керуючу програму для мініатюрної системи біореактора, яка виробляє біогаз за допомогою відкритої електронної платформи Arduino. Розроблена програма керування підходить для візуалізації та моніторингу відповідних параметрів процесу. Однак слід зробити незначні адаптації, щоб зробити його придатним для керування мініатюрною системою біореактора. По-перше, контрольна програма виявилася ефективною для моніторингу визначених параметрів. Контроль відповідних параметрів процесу був математично прийнятним. Тим не менш, хімічна точка зору припускає, що PID базового насоса слід відрегулювати, щоб він був достатнім для контролю значення рН у системі біореактора.

Візуалізація відповідних параметрів за допомогою розробленого графічного інтерфейсу виявилася успішною. Крім того, представлена система керування є легкою для створення та є економічно ефективною альтернативою порівняно з іншими системами керування, які доступні для біореакторів звичайного масштабу. Остаточна система керування Arduino була розроблена відповідно до кількох етапів, а саме: визначення налаштувань системи, апаратного забезпечення Arduino, програмного забезпечення Arduino та графічного інтерфейсу користувача.

Графічний інтерфейс користувача не має бездротового підключення до контролера Arduino. Подальші дослідження повинні вказати, чи є зчитування значень з IP-адреси та їх передача до обробки цінним майбутнім вдосконаленням дизайну.

БІБЛІОГРАФІЯ

1. PowerPoint slides miniaturised bioreactor system and other general information. URL : https://www.sartorius.com/en/products/fermentation-bioreactors/microbial-bioreactors?utm_source=google&utm_medium=cpc&utm_campaign=www_en_search_bioreactor/fermentation_Nonbranded&gad=1&gclid=CjwKCAjw-b-kBhB-EiwA4fvKrEH84TTTRTvqvR6CSyyY9ayMjtsEvc6ZFIGiGKSO30FnIHYtcTzEhRoCmN8QAvD_BwE.
2. Development of novel bioreactor control systems based on smart sensors and actuators. URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7011095/>.
3. Miniature Bioreactors: Current practices and future opportunities. URL : https://www.researchgate.net/publication/7057934_Miniature_Bioreactors_Current_practices_and_future_opportunities.
4. Bioprocess Engineering-Basic Concepts by Shuler and Kargi (PDFDrive). URL : https://www.academia.edu/80127975/Bioprocess_Engineering_Basic_Concepts_by_Shuler_and_Kargi_PDFDrive_.
5. Arduino: a low-cost multi purpose lab equipment. URL : https://www.academia.edu/6819930/Arduino_a_low_cost_multi_purpose_lab_equipment.
6. Open-Source bioreactor controller for bacterial protein expression. URL : https://www.researchgate.net/publication/327277190_Open-Source_bioreactor_controller_for_bacterial_protein_expression.
7. Development of an automation control application (Arduino) for a bioreactor system. URL : https://fse.studenttheses.ub.rug.nl/24020/1/Bachelor_Industrial%20engineering%20and%20Management_2021_JasperSteins3495329.pdf.

8. Pretreatment of organic composite waste mixtures for enhanced biomethanation. URL : https://www.academia.edu/83889720/Pretreatment_of_organic_composite_waste_mixtures_for_enhanced_biomethanation.
9. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 1. [навчальний посібник] (Лист МОНУ №1/11-8052 від 28.05.12р.) - Львів, "Магнолія 2006", 2013. – 256 с.
10. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 2. [навчальний посібник] (Лист МОНУ №1/11-11650 від 16.07.12р.) - Львів, "Магнолія 2006", 2014. – 312 с.
11. Микитишин А.Г., Митник, П.Д. Стухляк. Комплексна безпека інформаційних мережевих систем: навчальний посібник – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. – 256 с.
12. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі : навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 384 с.