

Прикарпатський національний університет імені Василя Стефаника

Кафедра комп'ютерної інженерії та електроніки

Дякун Євген Вікторович

Diakun Yevhen

УДК 004:681.5

Спеціальність 123 «Комп'ютерна інженерія»

Кваліфікаційна робота

на здобуття освітньо-кваліфікаційного

рівня бакалавр

Розробка мікроконтролерної системи для вирощування рослин  
методом гідропоніки

Development of the microcontroller system for growing plants using  
the hydroponics method

Науковий керівник:

д. ф.-м. н., проф. В.І. Мандзюк

Рецензент

д. ф.-м. н., проф. І.В. Горічок

Івано-Франківськ

2023

## АНОТАЦІЯ

Пропозиція виконати роботу народилася з ідеї полегшити та здешевити гідропонне вирощування їжі. Гідропоніка – це, по суті, система, де рослини вирощуються без використання традиційного ґрунту, замість цього коріння розміщують у водних розчинах, які містять необхідні розчинені поживні речовини. Гідропонна ферма, зазвичай, використовує спеціально розроблені системи зрошування, контрольоване освітлення та інші технології, щоб оптимізувати ріст та врожайність рослин. Найскладнішою та найдорожчою частиною такої системи є контролер.

Мета дипломної роботи полягала в тому, щоб розробити робочий гідропонний контролер, який є дешевим і досить простим у створенні. Він відстежуватиме та контролюватиме ключові екологічні інгредієнти, необхідні для успішного гідропонного вирощування, головним чином рН, електропровідності (electrical conductivity, EC), повітря та рівні води. Візуальні сигнали будуть активовані, якщо показники перевищуватимуть рівень попередньо визначених діапазонів. Контролер також зможе керувати зовнішнім апаратним забезпеченням (hardware, HW), наприклад, освітлення, водонагрівач і водяний насос.

Розроблено друковану плату, яка буде виступати в якості основи для плати Arduino 2560. Монтажна плата допоможе спростити побудову системи. Це також діє як спосіб підтримувати апаратне забезпечення більш стабільним, оскільки дроти не розв'язуються, а перешкоди будуть зведені до мінімуму.

Було враховано знання про те, що є важливим у гідропонному вирощуванні, і ці фактори використовувалися як вхідні дані для формування вимог до програмного (software, SW) і апаратного (HW) забезпечення. Були проведені дослідження зручності використання інтерфейсу користувача, щоб переконатися, що контролер буде зручним у використанні. Дизайн програмного забезпечення та шаблони програмного забезпечення були використані, щоб зробити програмне забезпечення більш модульним і розширюваним.

У цій дипломній роботі докладно описано апаратне та програмне

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 2    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

забезпечення, а також причини прийняття певних проектних рішень. Також описано дизайн класів SW, а також загальне функціонування SW. Також наведено пояснення способу побудови системи.

### ABSTRACT

The proposal to undertake this work arose from the idea of facilitating and reducing the cost of hydroponic food cultivation. Hydroponics is essentially a system where plants are grown without using traditional soil; instead, their roots are placed in water solutions containing the necessary dissolved nutrients. A hydroponic farm typically utilizes specially designed irrigation systems, controlled lighting, and other technologies to optimize plant growth and yield. The most complex and expensive part of such a system is the controller.

The objective of this thesis was to develop a functional hydroponic controller that is affordable and relatively easy to create. It would monitor and control key environmental factors necessary for successful hydroponic cultivation, primarily pH, EC, air, and water levels. Visual signals would be activated if the readings exceed predefined ranges. The controller would also be capable of managing external hardware such as lighting, water heaters, and water pumps.

A printed circuit board was developed to serve as the foundation for the Arduino 2560 board. The assembly board would simplify the construction of the system and help maintain hardware stability by minimizing loose wires and obstacles.

Knowledge of crucial aspects of hydroponic cultivation was taken into account, and these factors were used as input for defining software (SW) and hardware (HW) requirements. Usability research was conducted on the user interface to ensure that the controller would be user-friendly. Software design and templates were employed to make the software more modular and scalable.

This thesis provides a detailed description of the hardware and software, as well as the rationale behind specific design decisions. It outlines the design of software classes and the overall functioning of the software. Additionally, explanations are provided on the system's construction methodology.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 3    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## ЗМІСТ

|  |    |
|--|----|
| АНОТАЦІЯ .....   | 2  |
| ABSTRACT.....  | 3  |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....                                    | 5  |
| ВСТУП.....   | 6  |
| Розділ 1. Особливості вирощування рослин методом гідропоніки ..... | 8  |
| 1.1. Живильні розчини .....  | 9  |
| 1.2. Рівень рН .....   | 10 |
| 1.3. Електропровідність: рівень ЕС .....                           | 10 |
| 1.4. Температура води.....   | 11 |
| 1.5. Температура повітря .....                                     | 11 |
| 1.6. Рівень освітлення .....                                       | 11 |
| Розділ 2. Вимоги до апаратної частини та її проектування.....      | 13 |
| 2.1 Датчик температури.....  | 13 |
| 2.2 Датчик температури води .....                                  | 14 |
| 2.3 Фоторезистор .....   | 14 |
| 2.4 Годинник реального часу .....                                  | 15 |
| 2.5 TFT-сенсорний екран.....                                       | 15 |
| 2.6 Передавач і розетки .....                                      | 16 |
| 2.7 Схема вимірювання електропровідності .....                     | 17 |
| 2.8 Контур рН.....   | 18 |
| 2.9 Корпус .....   | 19 |
| 2.10 Схема апаратного забезпечення .....                           | 19 |
| Розділ 3. Вимоги до програмної частини та її проектування .....    | 21 |
| 3.1 Загальна діаграма пакета UML .....                             | 21 |
| 3.2 Пакет інтерфейсу користувача.....                              | 22 |
| 3.3 Інтерфейс користувача гідропоніки.....                         | 26 |
| 3.4 Бібліотеки.....  | 27 |
| 3.5 Механізм .....   | 29 |
| Розділ 4. Розробка інтерфейсу .....                                | 41 |
| 4.1. Сценарії використання .....                                   | 41 |
| 4.2 Можливості майбутнього розвитку.....                           | 47 |
| ВИСНОВКИ.....  | 48 |
| СПИСОК ЛІТЕРАТУРИ.....   | 49 |
| ДОДАТКИ.....   | 51 |

|     |      |          |        |      |  |  |  |  |                    |      |
|-----|------|----------|--------|------|--|--|--|--|--------------------|------|
|     |      |          |        |      |  |  |  |  | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |  |  |  |  |                    | 4    |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |                    |      |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

pH– powerofhydrogen(вимірювання кислотності або лужності водного розчину)

EC– electrical conductivity (електропровідність)

HW– hardware (апаратне забезпечення)

SW – software (програмне забезпечення)

EEPROM –electrically erasable programmable read-only memory (тип неволатильної пам'яті, яка може бути програмована та стерта електричним шляхом)

RTC – real time clock

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 5    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## ВСТУП

Значення гідропоніки полягає в тому, що середня людина може вирощувати власну їжу, не потребуючи ґрунту, наприклад, для людей, які живуть у квартирах та міських районах. Правильні рівні рН, ЕС і температури води є критично важливими в гідропоніці. Тому допомога контролера, який стежить за цими факторами, є неоціненною та забезпечить більший успіх та ефективність виробника.

Однак на даний момент такі контролери дорогі, і жоден із них не має відкритого коду. Кінцевою метою проекту є розробка та створення цілої гідропонічної системи з контролером, апаратне та програмне забезпечення якого має відкритий код. Сам контролер керує світлом, водяним насосом і вентилятором, щоб допомогти у вирощуванні їжі. Використання відповідного освітлення дало б можливість людям, які живуть в умовах недостатнього освітлення, наприклад у Фінляндії, вирощувати свіжу їжу цілий рік. Це було б особливо корисно зараз, коли ціни на продукти ростуть.

Загальний огляд гідропоніки та її переваг перед ґрунтовим садінням подано у роботі [1].

Надавши контролер з відкритим вихідним кодом, це дозволить більшій кількості людей отримати доступ до гідропонного вирощування. Це буде можливим, оскільки контролер буде дешевшим, а людям не знадобиться затрачати так багато часу для вирощення їжі, оскільки контролер подбає про більшість залучених аспектів. Наприклад, перевірка правильності рівня рН/ЕС або температури води/повітря.

Оскільки їжа стає дорожчою, це дозволить людям вирощувати власну їжу дешевше, особливо в районах, де ґрунт, світло чи температура можуть бути проблематичними для вирощування.

Згідно [2], переваги гідропоніки полягають у наступному:

- ґрунт для гідропоніки не потрібен;
- вода залишається в системі та може бути використана повторно – таким чином, менша потреба у воді;

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 6    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

- є можливість повністю контролювати рівень поживності – таким чином, знижуючи потребу в поживності;
- жодні забруднення не потрапляють в навколишнє середовище завдяки контрольованій системі;
- стабільна і висока врожайність;
- від шкідників і хвороб позбутися легше, ніж у ґрунті, завдяки ізоляваності системи;
- легкість збору врожаю;
- не потрібно використовувати пестициди;
- рослини ростуть здоровішими;
- така їжа краща для споживання і корисніша для здоров'я.

Сьогодні гідропоніка є усталеною галуззю агрономії. Прогрес був швидким, і результати, отримані в різних країнах, довели його практичність і явні переваги перед звичайними методами садівництва.

Є дві головні переваги безґрунтового вирощування рослин. По-перше, гідропоніка потенційно може дати набагато вищі врожаї. Крім того, гідропоніку можна використовувати в місцях, де неможливо вести наземне сільське господарство або садівництво.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 7    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## Розділ 1. Особливості вирощування рослин методом гідропоніки

Перед розробкою SW та HW було проведено дослідження методів і використання гідропоніки. Таким чином гарантується, що контролер запропонує максимальне використання кінцевому користувачеві.

По суті, гідропоніку можна коротко описати як «садівництво без ґрунту». Замість цього рослини отримують додаткові поживні речовини, які їм потрібні, з розчину на водній основі, який пропускають через їх коріння.

Пропускати воду крізь корінням можна вручну. Однак, як правило, це непрактично, якщо є більше пари рослин. Перевага автоматизованої системи полягає в тому, що насоси та таймери працюють автоматично, мінімалізуючи так званий людський фактор, а також дистанційний догляд за фермою.

Контролер використовується в поєднанні з системою. Система – це фактична структура, в якій будуть посаджені рослини. Існує чотири поширені типи систем:

### Система повеней/дренажу

У цій системі резервуар з водою знаходиться безпосередньо під зоною вирощування. Через певні проміжки часу насос наповнює зону вирощування водою. Коли насос вимкнеться, вода знову опуститься в резервуар.

### Крапельна система

У крапельній системі водяний насос буде накачувати воду над рослинами невеликими краплями, це може бути постійно або через певні проміжки часу. Вода під дією сили тяжіння капатиме на рослини та повертатиметься у резервуар.

### Техніка живильної плівки (NFT)

По суті, це така ж сама система, як і крапельна система, за винятком того, що рослини встановлюються в коритах, а вода закачується в жолоб і безпосередньо до коренів.

### Аeroponічна система

Те саме, що й система NFT, за винятком того, що використовується насос високого тиску з отворами, вирізаними кожні п'ятнадцятьсантиметрів або

|     |      |          |        |      |  |  |  |  |                    |      |
|-----|------|----------|--------|------|--|--|--|--|--------------------|------|
|     |      |          |        |      |  |  |  |  | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |  |  |  |  |                    | 8    |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |                    |      |



близько того в трубці. Таким чином вода розпилюється до коріння.

Аeropонічна система представляє актуальний інтерес, оскільки її можна використовувати для годування людей у космічних місіях. Також було доведено, що вона найкраща з точки зору поживних речовин для вирощування їжі. Згідно [2], дослідження NASA показали, що рослини, вирощені за допомогою aeropоники, мають на 80% більше сухої біомаси (необхідних мінералів) порівняно з рослинами, вирощеними на гідропоніці. Aeropоніка використовує на 65 % менше води, ніж гідропоніка. NASA також дійшло висновку, що рослини, вирощені aeropонікою, потребують  $\frac{1}{4}$  поживних речовин порівняно з гідропонікою. На відміну від рослин, вирощених на гідропоніці, рослини, вирощені на aeropоніці, не зазнають трансплантаційного шоку при пересадці в ґрунт, і дають виробникам можливість зменшити поширення хвороб і патогенів. Aeropоніка також широко використовується в лабораторних дослідженнях фізіології та патології рослин. NASA приділяє особливу увагу aeropонічним методам, оскільки в умовах невагомості з туманом легше працювати, ніж з рідиною.

Усі системи мають ту саму основну передумову, що й їх основна методологія, а саме утримувати коріння вологими та піддаватися впливу повітря. Контролер гідропоніки, який розглядається в цій дипломній роботі, підтримуватиме всі перераховані вище системи, що робить його універсальним для широкого діапазону методів вирощування.

### 1.1. Живильні розчини

Поживні розчини можна купити безпосередньо в магазинах. Однак існують безкоштовні інструменти з відкритим кодом, створені професійними хіміками, які дозволяють робити їх індивідуально. Такі як:

HydroBuddy[10];

HydroCal [11].

Система контролера вимірюватиме наступні властивості навколишнього середовища/поживного розчину, які є важливими при гідропонному

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 9    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

виросуванні рослин. Вони обговорюються в наступних розділах.

## 1.2. Рівень рН

Контроль рН надзвичайно важливий не лише в гідропоніці, але й у ґрунті. Рослини втрачають здатність поглинати різні поживні речовини, коли рН змінюється. Різні рослини мають певний рН, який є найефективнішим для них, хоча для більшості рослин оптимальне слабокисле середовище росту.

Ідеальний рівень рН становить від 5,5 до 7. Занадто швидка зміна рівня рН негативно вплине на рослину, оскільки це надто сильно її напружить.

Перевага контролера полягає в тому, що рівень рН води постійно повідомляється (кожні п'ять секунд). Користувач може встановлювати обмеження для рівнів рН, тому на головному екрані з'являтиметься візуальна підказка, якщо рівень рН коливається за межами попередньо визначених рівнів. Без контролера користувачеві знадобилося б використовувати зовнішній пристрій, який можна використовувати лише кілька разів на день. До цього часу рН може вийти за межі діапазону, що спричинить пошкодження рослин.

## 1.3. Електропровідність: рівень ЕС

Після рівня рН другим найважливішим елементом для вимірювання у водному розчині є рівень ЕС – міраконцентрації поживних речовин у розчині. Баланс речовин в розчині потрібний точний і чіткий. Згідно [4], першорядне значення мають іони, які визначають рН, які мають провідність усотні разів більшу, ніж інші іони.

Тому ЕС слід завжди вимірювати при постійному рН. Контролер дозволяє користувачеві знати рН під час вимірювання ЕС, тому ця вимога завжди виконуватиметься. Причина, чому рівень ЕС такий важливий, є наступною [4]. Електропровідність може сказати, чи розчин втратив поживні речовини або воду через випаровування, якщо вимірювання виконуються при тому самому значенні рН. ЕС слід вимірювати під час приготування розчину та тричі на день після цього. Якщо ЕС розчину стає занадто високим, можна додати води, щоб знизити його до вихідного значення. Якщо ЕС стає занадто низьким (70% від

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 10   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

початкового значення), не слід додавати поживні речовини. Це означає, що рослина суттєво змінила склад розчину, його потрібно утилізувати та приготувати свіжий.

Так само, як і рівень рН, ЕС постійно повідомляється користувачеві (кожні п'ять секунд), з можливістю для нього встановлювати сигнали тривоги на попередньо визначених рівнях. Якщо рівень ЕС виходить за межі попередньо визначеного діапазону, з'являється візуальна підказка. Знову ж таки, без контролера користувачеві знадобився б якийсь зовнішній пристрій, щоб зробити це та переконатися, що це виконується кілька разів на день.

#### **1.4. Температура води**

Щоб поживні речовини добре засвоювалися рослинами, температура води повинна бути в межах 18-26°C. Якщо температура води надто низька, її можна знову підігріти за допомогою водонагрівача.

Контролер повідомляє користувачеві про температуру води в режимі реального часу на головному екрані. Користувач має можливість встановити нижню і верхню межі температури води. Користувач, якщо він використовує водонагрівач, може налаштувати його так, що якщо температура впаде занадто низько, водонагрівач увімкнеться. Він вимикається, якщо вода стає достатньо теплою, забезпечуючи постійну температуру. Ця функція більш корисна, якщо користувач використовує контролер на вулиці, наприклад, у теплиці.

#### **1.5. Температура повітря**

Контролер повідомляє користувачеві температуру повітря в режимі реального часу на головному екрані. Як і в інших звітах головного екрану, якщо значення виходить за межі вказаного користувачем діапазону, на екрані засвітиться червоний індикатор. Якщо температура в межах діапазону, він буде зеленим. Контролер наразі не може контролювати температуру повітря. Можливе розширення проекту полягало б у тому, що вентилятор можна було б контролювати, якщо температура повітря занадто висока.

#### **1.6. Рівень освітлення**

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 11   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Проблема з рівнем освітленості існує не тільки в гідропоніці, але є загальною проблемою виробників. Якщо рівень освітленості занадто низький, можна використовувати контролер, щоб увімкнути лампу для вирощування. Оскільки система керуватиме зовнішніми дистанційними пультами, це означає, що користувач має гнучкість у виборі пристроїв.

Як показано вище, основні переваги контролера полягають у тому, що він дозволяє автоматизувати найважливіші аспекти вирощування та негайно попереджає користувача, якщо якийсь елемент виходить за межі заданого діапазону.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 12   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## Розділ 2. Вимоги до апаратної частини та її проектування

Для того, щоб контролер якісно виконував свої функції, було проведено дослідження того, як люди насправді використовують гідропонні системи, і це будуть вимоги, на яких базуватиметься апаратне та програмне забезпечення. Обґрунтування використання HW та SW описано в цьому та наступному розділах.

Апаратною платформою буде Arduino Mega 2560. Вона береться, оскільки має відкритий вихідний код і забезпечує достатньо контактів для підтримки 3,2-дюймового TFT-екрану та численних датчиків. Крім того, вона забезпечує доступ до контактів, які підтримують переривання, що робить її ідеальним для цього проекту. Доступний простір для програмного забезпечення становить 250 КБ, що більш ніж достатньо для цього проекту. Також вона підтримує зберігання EEPROM, що дає можливість зберігати будь-які налаштування користувача.

### 2.1 Датчик температури

Контролер повинен вміти вимірювати температуру повітря. Для цього буде використовуватися датчик DHT11. Було обрано саме його, оскільки температурний діапазон добре вписується в діапазон, необхідний для вирощування харчових продуктів, який становить 0-50°C. Він також має температурну точність  $\pm 2^\circ\text{C}$ . Однак це можна покращити, використовуючи зсув у програмному забезпеченні, щоб налаштувати його на фактичну температуру за допомогою ртутного термометра. Датчик може отримувати нові дані лише раз на 2 с. Хоча це не повинно бути проблемою для гідропоніки. Імовірність великих коливань температури повітря протягом двох секунд малоімовірна.

Повні технічні деталі датчика згідно [12]:

- працює від 3 до 5 В;
- максимальне використання струму 2,5 мА під час перетворення (під час запиту даних);
- добре підходить для вимірювання температури 0-50°C з точністю  $\pm 2^\circ\text{C}$ ;

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 13   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

- частота дискретизації не більше 1 Гц (раз на секунду).

Тривалий вплив сильного сонячного та ультрафіолетового світла може погіршити ефективність. З цієї причини датчик DHT11 буде знаходитись у самому корпусі. Контур не розсіює багато тепла, а тести, проведені за допомогою ртутного термометра, показують, що температура дорівнює температурі поза корпусом. Таким чином, можна отримати фактичну температуру, а не ту, яка була б вищою, наприклад, якби сонце світило прямо на нього.

При паянні DHT11 до схеми температура повинна бути нижче 260°C. Крім того, датчик не буде добре працювати в умовах роси. Датчик DHT11, який використовується в цьому проекті, поставляється з власними резисторами. Це має перевагу, оскільки займає менше місця на друкованій платі та полегшує створення кінцевим користувачам.

## 2.2 Датчик температури води

Як датчик температури води буде використано DS18B20 [13]. Він буде закритий пластиком, щоб можна було занурюватися у воду на невизначений термін. Вони широко продаються, тому будь-який користувач зможе отримати їх. Багато поставляються з кабелем довжиною 3 метри. Це зручно, оскільки тоді контролеру не потрібно розташовувати безпосередньо біля резервуара для води, що дає користувачеві додаткову гнучкість. Його температурний діапазон становить від -55°C до 125°C, тому більш ніж достатньо для вирощування їжі. Точність становить  $\pm 0,5^\circ\text{C}$  у всьому діапазоні – від 10°C до 85°C.

Роздільна здатність датчика температури налаштовується користувачем на 9, 10, 11 або 12 біт, що відповідає крокам 0,5°C, 0,25°C, 0,125°C і 0,0625°C. У цьому проекті використовується роздільна здатність 9 біт. Для гідропонного вирощування їжі достатньою точністю вважається півградуса.

Датчик підтримує зв'язок через однопровідну шину, тому за визначенням потрібна лише одна лінія даних.

## 2.3 Фоторезистор

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 14   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Для вимірювання кількості доступного світла використовується фоторезистор. Якщо світло занадто слабе, можна подати сигнал про ввімкнення лампи для вирощування. Фоторезистор не буде прикріплений до друкованої плати, а використовуватиме контакти плати Arduino. Причина цього в тому, що приєднання резистора до друкованої плати заощадить місце. Інші переваги включають той факт, що резистор повинен бути на зовнішній стороні корпусу, тому до нього легко просто приєднати кілька проводів.

## 2.4 Годинник реального часу

DS3234 є RTC і керується через інтерфейс SPI. Він оснащений вбудованою схемою визначення живлення, яка виявляє збої живлення та автоматично перемикається на резервне живлення від батареї, що ідеально підходить, якщо систему вимкнено. Годинник є центральним компонентом контролера, оскільки користувач може вмикати насоси через певні проміжки часу та залишати їх увімкненими протягом необхідного часу. Точність таймера становить  $\pm 2$  ppm між  $0^{\circ}\text{C}$  і  $40^{\circ}\text{C}$ . Він також підтримує сигналізацію. Вони не використовуються в цьому проекті, але ця можливість дає додаткову гнучкість під час розширення функцій у майбутньому. Основною причиною вибору цього пристрою була його точність, досягнута за допомогою температурно-компенсованого кристалічного генератора, як зазначено в описі [14]. Під час програмування пристрою дотримувалась інструкція щодо інтерфейсу Arduino SPI [15], щоб забезпечити надійне використання.

У попередній конструкції контролера використовувався дешевший DS1307 RTC та час не зберігався належним чином. Кілька хвилин на день втрачалися нерегулярно, що ускладнювало компенсацію. Оскільки робота таймера важлива для цього пристрою, було вирішено, що краще придбати більш дорогий і надійний. DS1307 RTC мав перевагу в тому, що для керування ним використовувався I2C, що означало, що потрібно було менше контактів, тим більше, що точність таймера є більш важливою.

## 2.5 TFT-сенсорний екран

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 15   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Користувач буде взаємодіяти з контролером через цей сенсорний екран. Для цього був обраний 3,2-дюймовий сенсорний екран Sainsmart. Розмір не надто малий, щоб було важко контролювати, і він не надто великий, щоб зробити сам контролер занадто великим.

Сумісний щит, який постачається разом з уже використаним. Це допомагає в будівництві, оскільки тоді контакти легше з'єднуються з Arduino. Він також підтримує SD-карту, щоб на ній можна було зберігати зображення, щоб інтерфейс виглядав краще. У цьому проекті не використовується SD-карта, але це буде сама те, що буде використано для подальшого вивчення, щоб зробити інтерфейс користувача більш приємним у використанні.

Іншими варіантами були керування системою лише через мобільний телефон чи комп'ютер. Однак це робить його менш незалежним і означає, що багато версій програмного забезпечення потрібно буде написати для багатьох ОС. Завдяки використанню TFT він робить HW доступним для набагато ширшої аудиторії. TFT не надто дорогий, тому в цілому він додає цінності продукту. Однак майбутній розвиток може полягати в тому, щоб дозволити пристрою підключатися до мережі, у якій комп'ютер може використовуватися для аналізу даних, які згенерував контролер. Ним також можна керувати дистанційно, через мережу, коли користувач знаходиться поза домом.

Бібліотеки графіки, які використовуються, надані Henning Karlsen [16]. Ці бібліотеки постачаються з гідною документацією та надають користувачу саме ті функції, які потрібні. Використання зовнішніх бібліотек зменшить час розробки та підвищить якість, оскільки ці бібліотеки використовуються багатьма людьми та є відкритими. Більше інформації про програмні бібліотеки, які використовуються разом із TFT HW, обговорюються в розділі програмного забезпечення.

## 2.6 Передавач і розетки

Для надсилання інформації на бездротові штекери знадобиться передавач. Передавач повинен буде працювати на частоті 433,92 МГц, оскільки це відкрита частота в Європі (а також у багатьох інших частинах світу) для

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 16   |



передачі даних, як зазначено в Рекомендації ЕСС 70-03 [5].

Це також частота, на якій працюють бездротові розетки в Європі, тому для того, щоб контролер був сумісний із найбільшою кількістю розеток, потрібно використовувати цю частоту.

Інші варіанти можна використовувати через передавач і бездротові розетки, наприклад, за допомогою реле. Однак було вирішено не використовувати цей підхід, оскільки він надто складний для простого користувача. Одним із принципів проекту є підвищення рівня використання та успіху гідропоніки. У багатьох випадках люди можуть бути не знайомі з електронікою, тому «возитися» з електромережею може бути надто страшно. Використання бездротових розеток також дає перевагу розміщення самого контролера в більшій кількості місць, тому не обов'язково поруч із самою системою.

У цьому проекті використовуються бездротові розетки NEXA PBR-2300, хоча можна використовувати будь-які самопрограмовані розетки, які працюють на частоті 433 МГц. Вилки будуть підключені до освітлення, водяного насоса та водонагрівача. Контролер увімкне та вимкне їх за потреби залежно від середовища чи налаштувань користувача.

Під час тестування в приміщенні та без приєднаної антени штекери можна вмикати та вимикати на відстані 10 м. За допомогою антени можна досягти відстані до 100 м. Цього більш ніж достатньо для цього проекту. Під час тестування цього проекту антена не використовувалася.

## 2.7 Схема вимірювання електропровідності

Для вимірювання рівнів ЕС у воді використовувався Atlas Scientificstamp 3.1 [18]. Це використовується, оскільки їхній рівень якості високий і існує багато документації щодо їх використання. Схема невелика і легко вписується в макетну плату або невеликий щит, який використовувався б на Arduino. Є також деякі світлодіоди налаштування, які можна використовувати як індикацію того, що команди справді надсилаються до схеми. Кінцевий користувач не побачить їх, але вони корисні в життєвому циклі розробки.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 17   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Як описано в таблиці 1 Додатку, AtlasScientific продає три різні типи контурів, кожен спеціально розроблений для певних видів води. У цьому проекті використовуватиметься датчик K0.1, оскільки питна вода буде використовуватися для поливу рослин. Використовувані шпильки описані в таблиці 2 (див. Додаток).

Сама схема вимагає лише двох ліній даних і працює при напрузі 5 В, що забезпечує точні показання. Схема може зберігатися при температурі від -20 до 125°C. Важливо зауважити, що під час тестування роз'єму BNC схему потрібно підключати безпосередньо до макетної плати без використання проводів, інакше шум спричинить неточні показання. Оскільки він буде припаяний до друкованої плати, пристрій не повинен страждати від жодної з цих проблем.

## 2.8 Контур рН

Була використана версія 5.0 схеми рН від AtlasScientific [18]. Він підтримує повне вимірювання рН (від 0,01 до 14,00), а точність знаходиться в межах двох значущих цифр. Він також пропонує залежні або незалежні показники температури. Незважаючи на те, що використовуватимуться лише показники температури, оскільки контролер постачається з датчиком температури води. Сам контур має діапазон температур зберігання від -40°C до 125°C. Використовувані шпильки описані в таблиці 2 (див. Додаток).

Швидкість за замовчуванням становить 38400, 8 біт без паритету. Однак схема підтримує вісім різних швидкостей передачі даних:

1. 300
2. 1200
3. 2400
4. 9600
5. 19,2 тис
6. 38,4 тис
7. 57,6 тис
8. 115,2 тис

Схема рН дуже чутлива, і торкання до неї може спричинити неточні

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 18   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

показання протягом приблизно п'яти хвилин. Оскільки контур буде додано до контролера гідропоніки, це не повинно викликати жодних проблем.

## 2.9 Корпус

Корпус призначений лише для зовнішнього вигляду та захисту обладнання від бризок води на контур. У цьому проекті використано базовий пластиковий корпус, у якому спереду та ззаду будуть вирізані отвори для TFT-екрану та будь-яких проводів.

## 2.10 Схема апаратного забезпечення

Було розроблено та виготовлено основу Arduino, який з'єднає всі частини HW разом і міцно утримуватиме їх на місці. Нижче наведено діаграму для цього (рис. 2.1). Все буде розроблено за допомогою Fritzing, оскільки він безкоштовний без обмежень і дозволяє виготовляти друковані плати.

Було виготовлено друковану плату, оскільки це зробить гідропонний контролер більш стабільним і менш ймовірно, що зчитування буде залежати від перешкод. Це також усуває необхідність у безладних проводах і будь-яких помилках HW; наприклад прогорання дроту.

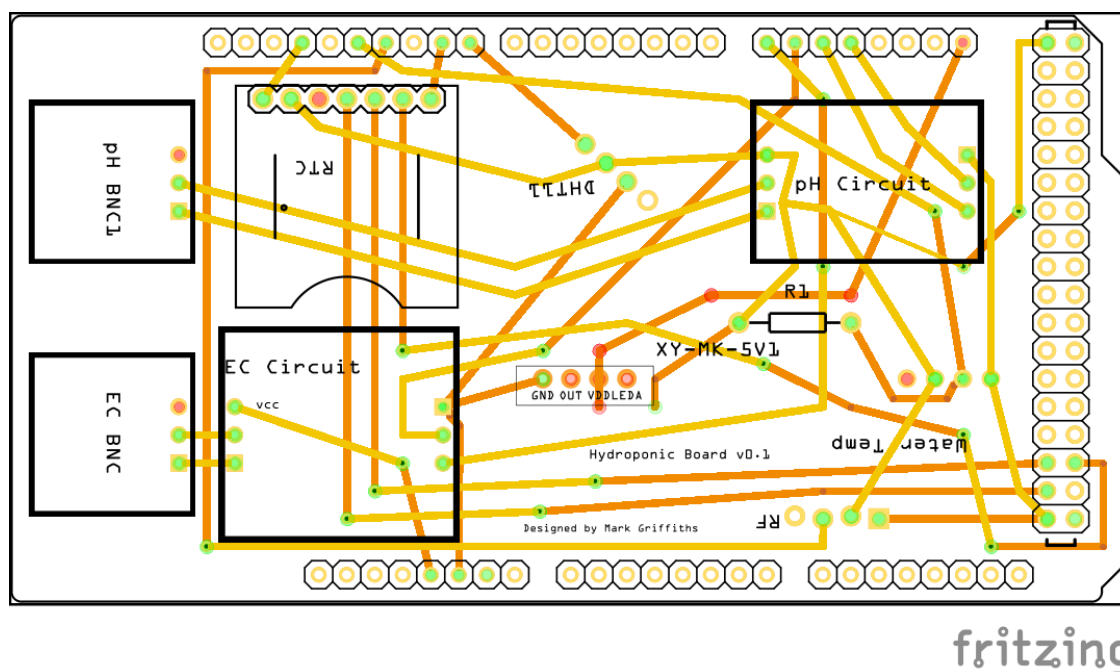


Рисунок 2.1 – Схема гідропоніки

Схема, яка використовується в цьому проекті, показана на рис. 2.1. У ній використовуються два мідні шари з мідним шаром зверху, щоб мінімізувати

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 19   |

ризик перешкод від передавача та приймача. TFT просто підключатиметься за допомогою роз'єму molex до необхідних контактів. Схема була розроблена таким чином, щоб з'єднання не перетиналися занадто сильно і були якомога коротшими. Було також подумано про збірку системи, тому роз'єми рН і ЕС BNC розташовані ззаду. Таким чином, система виглядатиме краще, оскільки всі з'єднання будуть на задній панелі пристрою. Провід живлення для Arduino розміщуватиметься трохи нижче роз'ємів BNC.

Однією з головних проблем при розробці HW був простір. Усі компоненти майже поміщаються на платі. Якщо знадобиться додаткове апаратне забезпечення, знадобиться також більша плата. Це може не викликати особливих проблем, якщо у корпусі достатньо місця для його розміщення. Однак більша плата також буде коштувати дорожче, а однією з цілей проекту було зробити контролер якомога дешевшим. Зараз використовуються майже всі переривання та цифрові контакти Arduino. Будь-яке додаткове розширення апаратного забезпечення вимагатиме перегляду того, який процесор і плата повинні жити пристрій. Загальний вигляд системи показаний на рис. 2.2.

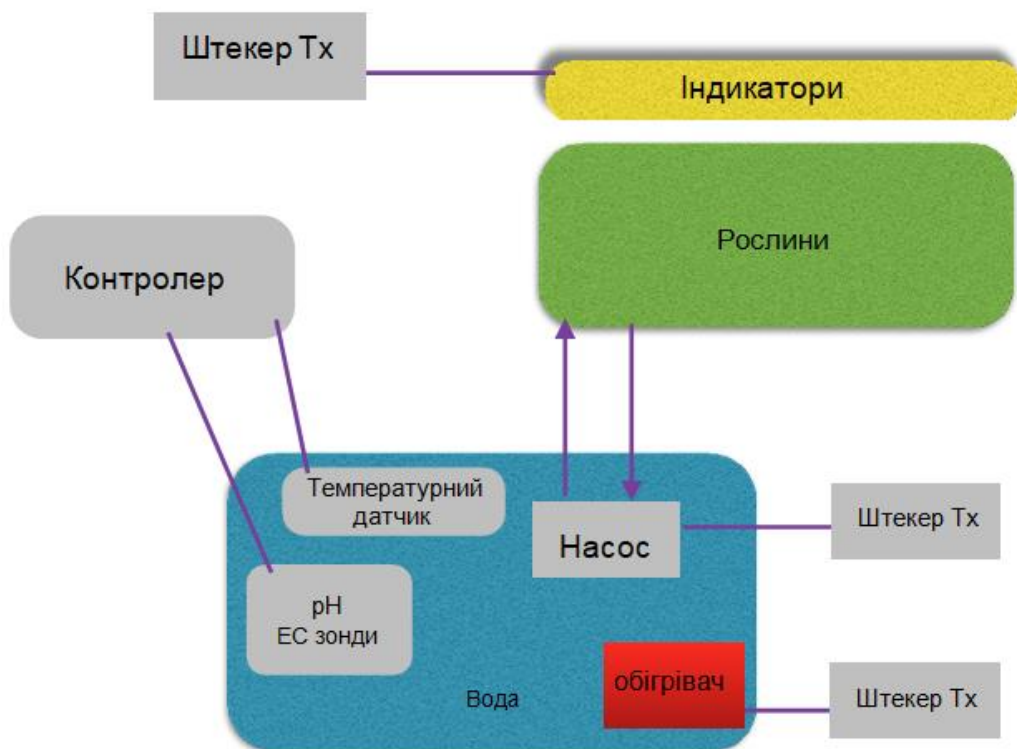


Рисунок 2.2 – Загальний вигляд системи

### Розділ 3. Вимоги до програмної частини та її проектування

Розробка програмного забезпечення була найбільш трудомісткою частиною створення контролера, оскільки саме тут уся логіка. Використана мова C++, а Arduino IDE використовувався як компілятор для перезавантаження програмного забезпечення на апаратне забезпечення. Основну частину програмного забезпечення було створено з нуля, хоча деякі бібліотеки з відкритим кодом використовувалися для взаємодії з апаратним забезпеченням.

Програмне забезпечення має не тільки зчитувати інформацію датчика, але також має використовуватися для калібрування зондів рН і ЕС, а також інших датчиків, щоб надати детальну інформацію користувачеві, щоб функціонувати як попередження для користувача, якщо деякі налаштування, наприклад, надто високого або низького рівня рН, і для керування деякими іншими пристроями, наприклад освітленням і водяними насосами.

Сам контролер працює без допомоги комп'ютера, і всі введення надходять з інтуїтивно зрозумілого сенсорного екрана, який за необхідності видає попередження користувачу. Програмне забезпечення було розроблено таким чином, що кожен компонент HW має власну бібліотеку, а інтерфейс користувача, контролер і механізм були відокремлені один від одного. Контролер – це просто функція циклу, яка є основною функцією Arduino.

Причина цього поділу полягає в тому, що в майбутньому може використовуватися інший TFT-екран. Тоді потрібно буде переписати лише розділ інтерфейсу користувача. Аналогічно для бібліотек HW, якщо в майбутньому використовуватиметься інший пристрій HW, то потрібно буде переписати лише цю бібліотеку. Якщо потрібно, потрібно створити лише оболонку інтерфейсу.

#### 3.1 Загальна діаграма пакета UML

На рис. 3.1 показано бібліотеки, які використовуються в програмному забезпеченні, і їхні зв'язки одна з одною. Пакети Engine, MainLoopController і UI створені самостійно для цього проекту. Усі інші пакети є пакетами

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 21   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

сторонніх розробників, усі зовнішні пакети використовуються для керування певним апаратним забезпеченням. Пакунки обговорюються більш детально в наступних кількох розділах.

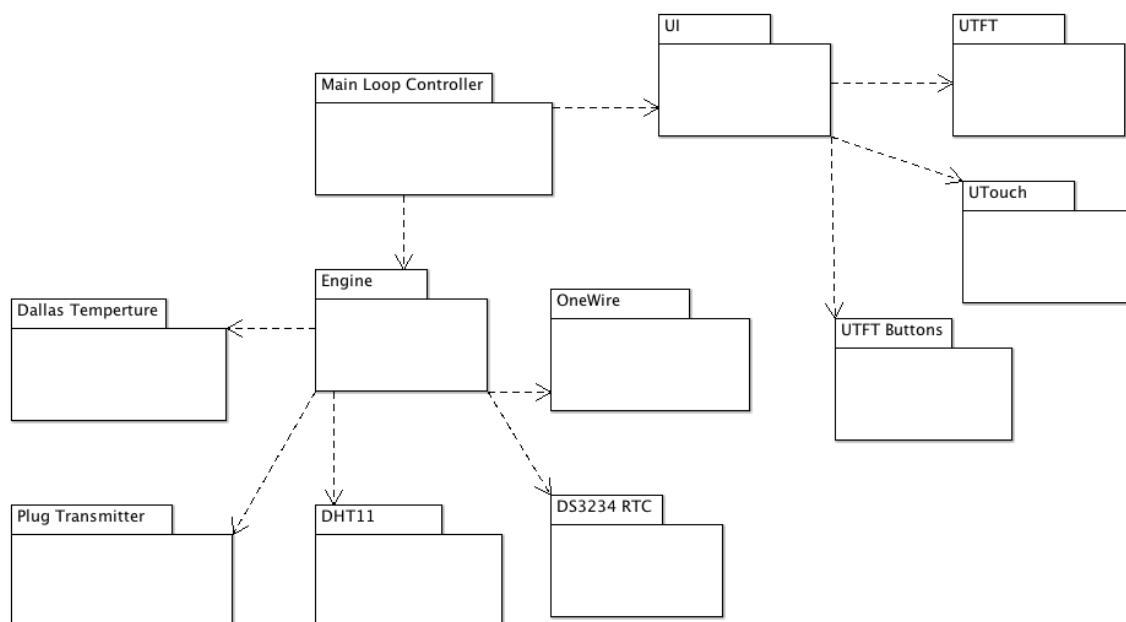


Рисунок 3.1. – Діаграма упаковки

### 3.2 Пакет інтерфейсу користувача

У пакеті інтерфейсу користувача знаходиться більшість коду та логіки для контролера. Цей пакет відповідає за малювання інтерфейсу користувача, а також за фіксацію дотиків до екрана. Якщо деякі значення виходять за межі діапазону, він подає сигнал в інтерфейс користувача. Значення передаються до інтерфейсу користувача через контролер.

Діаграма класів UML пакета інтерфейсу користувача показана на рис. 3.2. Кожен з класів, крім базових класів, представляє один екран. Тому він має одну конкретну роботу, пов'язану з функціональністю. Класи інтерфейсу користувача описані більш детально нижче. Є два базових класи. MinMaxScreen і Screen.

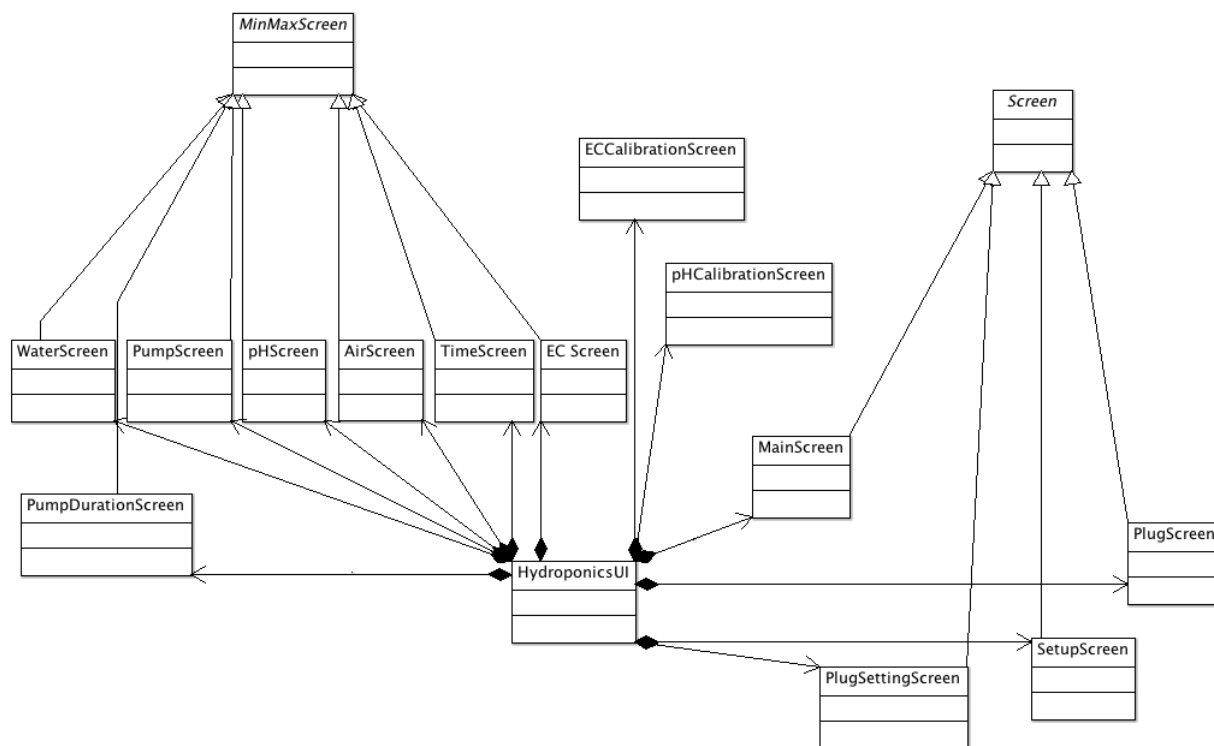


Рисунок 3.2 – Діаграма класів UI

### MinMaxScreen

Це базовий клас для багатьох інших екранів. Його можна використовувати, коли на екрані, який буде використовуватися, потрібно вказати лише мінімальне та максимальне значення. Екрани в цих випадках є досить загальними, і саме ці загальні дані та макет інкапсульовані в цьому класі. Тоді похідні класи мають надати лише певну інформацію, наприклад текст, який має бути на екрані.

### Screen

Цей базовий клас інкапсулює логіку обробки кнопок екранів, які оброблятимуть налаштування або на яких багато кнопок відобразатиметься на екрані. Це полегшує похідним класам обробку логіки кнопки.

Наступні класи представляють екран і малюватимуть на екрані та виконуватимуть деякі функції.

### Водяний екран

Цей екран дозволяє користувачеві встановлювати нижній і вищий порогові значення температури води. Після виходу з екрана мінімальне та максимальне значення зберігаються в EEPROM.

## **Повітряний екран**

Цей екран дозволяє користувачеві встановити нижній і вищий пороги для температури повітря. Після виходу з екрана мінімальне та максимальне значення зберігаються в EEPROM.

## **Екран рН**

Цей екран дозволяє користувачеві встановлювати нижній і вищий пороги для рівнів рН. Після виходу з екрана мінімальне та максимальне значення зберігаються в EEPROM. На цьому екрані також відображається кнопка калібрування рН. Коли користувач натискає цю кнопку, він відкриває рНCalibrationScreen.

## **Екран ЕС**

Цей екран дозволяє користувачеві встановлювати нижній і вищий пороги для рівнів ЕС. Після виходу з екрана мінімальне та максимальне значення зберігаються в EEPROM. На цьому екрані також зображено кнопку ЕС-калібрування. Коли користувач натискає цю кнопку, він відкриває EC CalibrationScreen.

## **Екран часу**

На цьому екрані користувач може змінити час. Їм надано два варіанти, годину та хв. Таким чином, це подібно до мінімального та максимального. Це причина, чому цей екран походить від MinMaxScreen. Після виходу значення не зберігаються в EEPROM, але час зберігається в схемі DS3234.

## **Екран насоса**

Користувач встановлює на цьому екрані інтервал, через який вмикатиметься насос. Знову можливі години та хвилини. Якщо користувач вибере 0 годин і 8 хвилин, це означатиме, що насос вмикатиметься кожні 8 хвилин.

## **Екран тривалості насоса**

Користувач встановлює на цьому екрані інтервал, через який насос буде працювати. Знову можливі години та хвилини. Якщо користувач вибере 0 годин і 4 хвилини, це означатиме, що насос залишатиметься увімкненим протягом

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 24   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |



4 хвилин. Якщо період тривалості перевищує інтервал, насоси ніколи не вимкнуться.

### **Екран калібрування рН**

Цей екран допоможе користувачу відкалібрувати схему рН. Їм даються інструкції для кожного етапу. Кожен етап займає кілька хвилин, і для цього використовується бібліотека simpleTimer. Екран асинхронний і не блокує оновлення інших апаратних пристроїв, наприклад, увімкнення насоса.

### **Екран калібрування ЕС**

Цей екран допоможе користувачеві виконати калібрування ЕС-ланцюга. Їм даються інструкції для кожного етапу. Кожен етап займає кілька хвилин, і для цього використовується бібліотека simpleTimer. Екран асинхронний і не блокує оновлення інших апаратних пристроїв, наприклад, увімкнення насоса.

### **PlugScreen**

Цей екран дозволяє користувачеві відкалібрувати розетки для освітлення, насоса чи обігрівача. Після натискання однієї з кнопок користувачеві пропонується натиснути кнопку на пульті дистанційного керування. Потім пристрій захопить ідентифікатор розетки та використає його під час передачі інформації до розетки для ввімкнення або вимкнення. Користувач також може отримати доступ до екрана PlugSettingsScreen.

### **Екран налаштувань розетки**

Цей екран дозволяє користувачеві вручну вмикати насос, освітлення та обігрівач. Це корисно, якщо користувач хоче вимкнути світло, незважаючи на те, що світла замало тощо. Це лише спосіб надати користувачеві додатковий контроль над пристроєм.

### **Головний екран**

Саме тут контролер проводитиме більшу частину свого часу. На головному екрані відображається інформація про навколишнє середовище, як-от рівні ЕС та рН, а також температура повітря та води. Тут також відображаються сповіщення, наприклад, якщо один із рівнів виходить за межі діапазону. Також відображається час. Звідси користувач може отримати доступ до SetupScreen, WaterScreen, AirScreen, ECScreen або pHScreen.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 25   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## Екран налаштування

Екран налаштування – це просто екран меню, де користувач може переходити до інших екранів. Він отримує доступ до всіх вищезгаданих екранів.

### 3.3 Інтерфейс користувача гідропоніки

Цей клас є мозком пакета інтерфейсу користувача. Тут пропускаються знімки сенсорного екрана, щоб активувався правий екран. Контролер ініціалізує це під час завантаження, і, у свою чергу, цей клас ініціалізує всі інші класи в пакеті.

Наступний фрагмент вихідного коду показує, як активується кожен екран, `CheckForScreenTouch` викликається з контролера кожні 100 мс. Бібліотека `UTouch` використовується для виявлення дотику до TFT-екрану.

```
voidHydroponicsUI::CheckForScreenTouch()
{
    If (myTouch.dataAvailable())
    {
        switch (whichScreen)
        {
            case MAIN_SCREEN:
            {
                handleButtons( iMainScreen->handleScreen() );
                break;
            }
            case ABOUT_SCREEN:
            {
                handleButtons( iAboutScreen->handleScreen() );
                break;
            }
        }
    }
}
```

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 26   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

```

    }

    case SETUP_SCREEN:

    {

    handleButtons( iSetupScreen->handleScreen() );

    break;

    }

```

у наступній функції малюється фактичний екран.

```

voidHydroponicsUI::handleButtons( intaHandleWhichButton)
{
    switch(aHandleWhichButton)
    {
    case AIR_BUTTON:

        {

        iAirScreen->drawScreen(whichScreen);

        whichScreen = AIR_SCREEN;

        break;

        }

    case PH_BUTTON:

        {

        ipHScreen->drawScreen(whichScreen);

        whichScreen = PH_SCREEN;

        break;

```

### 3.4 Бібліотеки

#### Бібліотека UTFT

Для розміщення тексту та графіки на екрані використовувалася вільно доступна бібліотека UTFT з відкритим кодом [5].

|     |      |          |        |      |                    |            |
|-----|------|----------|--------|------|--------------------|------------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк.<br>27 |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |            |

Із точки зору ПЗ, ми використовуємо деякі глобальні змінні в усьому коді. Для TFT- екрана вони визначаються як:

```
//Global definition for whole of the UI
UTFT myUTFT(ITDB32S, 38,39,40,41);
UTouchmyTouch(6,5,4,3,2);
UTFT_ButtonsmyButtons(&myUTFT, &myTouch);
```

Ці цифри позначають шпильки, які використовуються на екрані. Після цього бібліотеки для TFT-екрану готові до використання.

За допомогою цієї бібліотеки можна ініціалізувати екран, установити його колір, очистити та заповнити його.

У наступному прикладі коду показано, як використовується ця бібліотека ПЗ.

```
myUTFT.clrScr();
myUTFT.setBackColor(0, 0, 225);
myUTFT.print("Reset", 55, 30);
```

У наведеному вище прикладі екран було очищено від усього іншого тексту та зображень, колір фону встановлено на синій, а текст «Скинути» було надруковано в заданих координатах x, y.

### **Бібліотека UTouch**

Бібліотека UTouch [19] відповідає за сенсори. Прикладом налаштування екрану є

```
myTouch.InitTouch();
myTouch.setPrecision(PREC_MEDIUM);
```

Тут ініціалізовано об'єкт myTouch, а точність сенсорного екрана встановлено на середню. Після цього бібліотека UTouch використовується не безпосередньо, а через бібліотеку UButtons, як описано нижче.

### **Бібліотека UButtons**

Бібліотека UButtons допомагає спростити код інтерфейсу користувача, легко малюючи кнопки на екрані. Він сповіщає про натискання певної кнопки, повертаючи ідентифікатор кнопки назад. Він може вмикати/вимикати кнопки та

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 28   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

встановлювати для них текст і піктограми. Це найбільш використовувана бібліотека в кодї інтерфейсу користувача. Прикладом може бути:

```
resetButton = myButtons.addButton( 10, 10, MAIN_BUTTON_X,  
MAIN_BUTTON_Y, mainButton);
```

Цей код створює кнопку на x 10 і у 10 розміром MAIN\_BUTTON і використовує зображення mainButton для створення кнопки. resetButton – це ідентифікатор цієї кнопки, який можна використовувати, щоб знати, коли кнопку було натиснуто.

```
intpressed_button = myButtons.checkButtons();  
if (pressed_button==backButton)  
{  
    handleExitScreen();  
    повернути ЕС_BUTTON;  
}  
If (pressed_button==resetButton)  
{  
    factoryReset();  
}
```

У наведеному вище фрагменті коду ми просто перевіряємо, яка кнопка була натиснута, і виконуємо певну дію на основі цього. Використання цієї бібліотеки значно покращило читабельність і зручність обслуговування коду.

### 3.5 Механізм

Основним завданням цього механізму є отримання даних від датчиків обладнання, наприклад, температури води, та оновлення структури датчика. Саме структура датчика передається до користувацького інтерфейсу (UI) через контролер, який використовується при оновленні UI. Структура датчика передається тільки у випадку, якщо є оновлена інформація від обладнання, наприклад, якщо змінилася температура. Контролер перевіряє це застосовуючи описану тут перевірку.

```
if( Hydro_Eng.doesScreenNeedUpdating() )
```

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 29   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

```

    {

    //Оновлення інтерфейсу користувача останньої інформації
    про датчик
    Hydro_UI.updateInfoFromSensors(Hydro_Eng.getSensorInformation(
    ));

    //Оновити екран новою інформацією.

    Hydro_UI.refreshScreen();
    //Інформувати движок, що інтерфейс користувача оновлено.
    Hydro_Eng.screenUpdated();
    }

```

Нижче наведено структуру, що містить оновлені дані датчика.

```

Hydro_UI.updateInfoFromSensors( Hydro_Eng.getSensorInformation());

```

Потім виконується оновлення екрана, щоб інтерфейс користувача завжди відображав найновішу інформацію.

Бібліотека механізму складається лише з одного класу, який взаємодіє з бібліотеками HW. Як видно на рис. 3.3, кожен компонент апаратного забезпечення має свою власну бібліотеку та описаний більш детально в наступних розділах. Кожна з бібліотек є зовнішньою, і всі мають відкритий код [20].

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 30   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

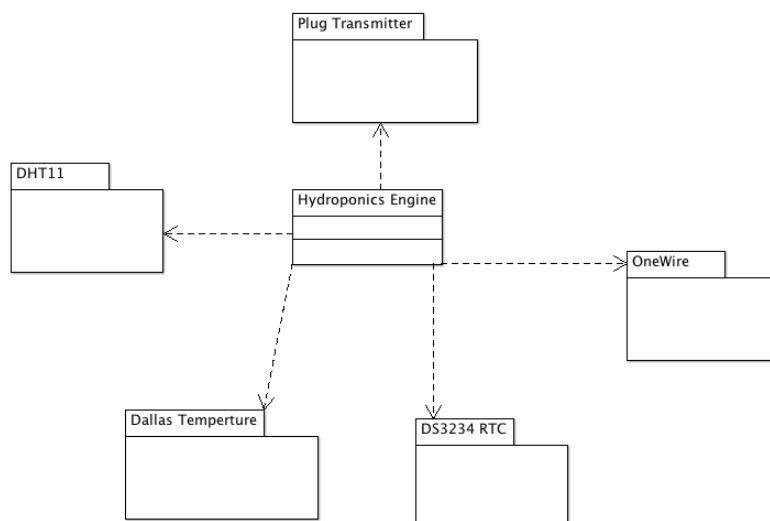


Рисунок 3.3 – Зовнішні бібліотеки механізму

### Бібліотека приймачів і передавачів

Бібліотека, яка використовується для керування приймачем і передавачем, називається 433 МГц для Arduino [21]. Цей пакет містить кілька невеликих бібліотек для Arduino 1.0, які додають можливість зв'язку з деякими радіокерованими (433 МГц / 434 МГц) побутовими приладами. Бібліотеки можна використовувати для легкої домашньої автоматизації за допомогою дешевих готових компонентів.

Ця бібліотека використовувалася для вмикання та вимикання дистанційних розеток. Кожна пробка має бути налаштована на певну функцію, наприклад водяний насос. Це може зробити користувач через інтерфейс користувача.

Щоб отримати деталі розетки, потрібно використовувати пульт від розеток. Коли ви перебуваєте на екрані розетки, вам потрібно просто натиснути кнопку, і пристрій захопить деталі розетки. У коді це робиться через зворотний виклик. Коли передавач отримує інформацію, він захоплює та зберігає її, як показано в прикладі коду.

Спочатку приймач налаштовується на правильний PIN-код і дається зворотний виклик `waitForReceiver`. Номер один, поданий як другий параметр нижче, означає, що це буде зроблено лише один раз. Повтори не допускаються.

```
NewRemoteReceiver::init(RECEIVER_PIN, 1, waitForReceiver);
```

Коли користувач натискає кнопку, викликається зворотний виклик. У зворотному виклику інформація зберігається в EEPROM для подальшого використання.

```
setAddressstoEEPROM(receivedCode.address);
    if (plugDevice.lights)
    {
EEPROM.write(EEPROM_LEARNED_LIGHTS_UNIT,receivedCode.unit);
        plugDevice.lights = false;
    }
    (plugDevice.pump)
    {
EEPROM.write(EEPROM_LEARNED_PUMP_UNIT,receivedCode.unit);
        plugDevice.pump = false;
    }
    if (plugDevice.heater)
    {
EEPROM.write(EEPROM_LEARNED_HEATER_UNIT,receivedCode.unit);
        plugDevice.heater = false;
    }
}
```

### **Бібліотека RTC DS3234**

Використання цієї бібліотеки робить керування часом дуже простим і через інтуїтивно зрозумілий API, який зараз описано. Бібліотека DS3234 [22] ініціалізується шляхом передачі пін-коду SS.

```
DS3234 rtc(RTC_PIN);
```

Після цього час можна встановити на основі введення користувача. Нуль означає, скільки секунд. Користувач не може вибрати це, тому для нього встановлено нульове значення.

```
rtc.setTime(aHour, aMin, 0);
```

Коли запитується час, код виглядає наступним чином.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 32   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |



```

voidHydroponicsEngine::GetTime(int&aHour, int&aMin )
{
    clock = rtc.getTime();
    aHour = clock.hour;
    aMin = clock.min;
}

```

### Далласька бібліотека температури води

Бібліотеку температури води [23] спочатку просять отримати температуру за допомогою такої команди:

```
waterTempSensor.requestTemperatures();
```

Після цього можна дізнатися поточну температуру води за телефоном

```
inttemp = waterTempSensor.getTempCByIndex(0)+ waterTempCal;
```

Тут waterTempCal є калібрувальною константою. Під час тестування я викликав цю функцію та порівняв її з ртутним термометром і помітив, що вона була на градус або два. Більше довіряючи ртутному термометру, я додав до нього константу, яка змінювала б температуру на основі тестів. Для кожного окремого HW, яке використовується, цю константу потрібно буде змінити на основі результатів тестування.

### Бібліотека температури повітря DHT11

Бібліотека DHT11 [24] дуже проста у використанні. Щоб отримати температуру, потрібно просто викликати функцію читання за допомогою контакту, до якого підключено DHT11, як показано у цьому прикладі:

```
intchk = DHT11.read(DHT11_PIN);
```

```
sensorInfo.airTemp= DHT11.temperature();
```

Після цього виклику отримаємо температуру в градусах Цельсія, розташовану в sensorInfo.airTemp, і можемо оновити інтерфейс і будь-що інше за потреби. Якби ми хотіли температуру за Фаренгейтом, ми викликали б функцію

```
DHT11.fahrenheit();
```

Бібліотека також підтримує отримання температури в Кельвінах і

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 33   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

отримання точки роси. Це програмне забезпечення не використовує ці функції.

### ЕС-зонд

Зонд ЕС не використовує власну бібліотеку, оскільки весь зв'язок із ним здійснюється через послідовний порт. По суті, це його бібліотека. Швидкість передачі 38400. Повний посібник і опис доступні в [17].

Наступні команди, наведені в таблиці 3 Додатку, використовуються для керування схемою ЕС. Наприклад, команда **R** вказує схемі ЕС повертати одне показання ЕС. Для виконання цієї інструкції потрібно 1000 мс. Далі вказаний приклад:

```
inputstring = "R\r"; //Команда для читання Serial3.print(inputstring);  
//відправити команду на датчик.
```

Для більшості команд схема відповідає функцією зворотного виклику послідовних портів. Якщо припустити, що ми використовуємо послідовний порт 3, це буде void serialEvent3(). Ця функція зворотного виклику викликається кожного разу, коли функція основного циклу завершується.

Після відправки команди **R**, схема відповідає:

ЕС,TDS,SAL<CR>

де:

ЕС – електропровідність у мкс/см

TDS – загальна кількість розчинених твердих речовин (з посиланням на КСІ)

SAL – це солоність (практична шкала солоності 1978), виражена лише цілим числом.

Як і при надсиланні команди, кожна відповідь завершується поверненням каретки (<CR>). Так програмне забезпечення знає, що отримано повну відповідь.

У функції serialEvent3 отримуємо результати того, що зробила схема. У деяких випадках він дає дані, які запитували, а в інших це лише підтвердження того, що все пройшло так, як очікувалося. Ця функція показана тут:

```
void serialEvent3()
```

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 34   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

```

{
sensostring="";
while(Serial3.available())
    {
    charinchar = (char)Serial3.read();
    sensorstring += inchar;
    if(inchar == '\r') {sensor_stringcomplete = true;}
    }
Serial.print(sensorstring.length());
if(sensorstring.length() == 9) //довжин ЕС а відповіді.
{
    ECLevelString = sensorstring.substring(0,3);
}
//перевірити, які дані ми очікуємо.
if(waiting_for_info&&sensor_stringcomplete)
    {
    drawInfoScreen();
    waiting_for_info = false;
    sensor_stringcomplete = false;
    }
if(waiting_for_single_reading&&sensor_stringcomplete)
    {
    drawReadingScreen();
    waiting_for_single_reading = false;
    sensor_stringcomplete = false
    }
}

```

Контур ЕС повинен бути відкалібрований програмним забезпеченням у такому порядку:

1. Встановіть тип датчика.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 35   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

2. Відкалібруйте для сухого датчика.
3. Виконайте калібрування для високого показання.
4. Виконайте калібрування для низького показання.

Усі дані зберігаються в пам'яті схеми EEPROM і не будуть втрачені під час вимкнення живлення.

Команди, які використовуються лише під час калібрування, перераховані в таблиці 4 Додатку. Під час калібрування ми чекаємо п'ять хвилин, перш ніж надсилати показання в схему. Цей проміжок часу потрібен для стабілізації показників ЕС.

Калібрування схеми ЕС проводиться в класі EC CalibrateScreen. На етапі калібрування потрібен доступ до калібрувальних розчинів.

Очікування п'ятихвилинного періоду є асинхронним з точки зору HW та SW. Протягом цього часу інтерфейс користувача оновлюватиметься кожні три секунди, інформуючи користувача про те, скільки часу залишилося до закінчення процесу калібрування. Користувач не зможе нічого робити, окрім очікування завершення калібрування. Оновлення інтерфейсу користувача зроблено для того, щоб користувач не думав, що пристрій завис.

Проте протягом цього часу програмне забезпечення все ще опитує апаратне забезпечення, щоб отримати рівень ЕС, і воно встановить рівень ЕС для цього калібрування лише після закінчення часу.

### **Зонд рН**

Зонд рН [19], як і зонд ЕС, не використовує власну бібліотеку, оскільки весь зв'язок із ним здійснюється через послідовний порт. Швидкість передачі 38400 бод.

Наступні команди використовуються для керування контуром рН. Наприклад **R** вказує контуру рН повернути одне значення рН. Для виконання цієї інструкції потрібно 378 мс. Наприклад

```
inputstring = "R\r"; //Команда для читання
```

```
Serial2.print(inputstring); //відправити команду на датчик.
```

Схема рН і ЕС виготовлені одним виробником. Як видно з розділу про

|     |      |          |        |      |  |  |  |  |  |                    |      |
|-----|------|----------|--------|------|--|--|--|--|--|--------------------|------|
|     |      |          |        |      |  |  |  |  |  | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |  |  |  |  |  |                    | 36   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |                    |      |

ЕС-зонд, між цими двома зондами багато подібності. Знову ж таки, усі команди мають закінчуватися поверненням каретки. Це повідомляє контуру рН, що ми надіслали нашу команду.

Для більшості команд схема відповідає функцією зворотного виклику послідовних портів. Якщо припустити, що ми використовуємо послідовний порт 2, це буде void serialEvent2(). Це функція зворотного виклику викликається лише кожного разу, коли функція основного циклу завершується.

Після відправки команди Р, схема відповість «XX.XX». Це поточне значення рН, наприклад, 7,5. Якщо коло рН зчитує рН, що виходить за межі діапазону, він відповість повідомленням про помилку «перевірте зонд». Як і при надсиланні команди, кожна відповідь завершується поверненням каретки (<CR>). Таким чином програмне забезпечення знає, що отримано повну відповідь.

У serialEvent2 отримуємо результати схеми. У деяких випадках отримуємо відповідь на запит, а в інших це лише підтвердження того, що все пройшло, як очікувалося. Інші команди та відповіді підсумовано в таблиці 5 в додатках.

Контур рН повинен бути відкалібрований програмним забезпеченням у такому порядку:

1. Калібрування в розчині рН7.
2. Калібруйте в розчині рН4.
3. Калібруйте в розчині рН10.

Усі дані зберігаються в пам'яті схеми EEPROM і не будуть втрачені під час вимкнення живлення.

Команди, які використовуються лише під час калібрування, перераховані в таблиці 6 додатків. Виклик будь-якої з команд, коли датчик рН не занурений у правильний розчин рН, відкалібрує контур рН до довільного значення та може призвести до значних помилок.

Під час калібрування чекаємо три хвилини, перш ніж відправити показання в схему. Цей проміжок часу потрібен, щоб показники рН

|     |      |          |        |      |  |  |  |  |  |      |
|-----|------|----------|--------|------|--|--|--|--|--|------|
|     |      |          |        |      |  |  |  |  |  | Арк. |
|     |      |          |        |      |  |  |  |  |  | 37   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |      |

стабілізувалися.

Калібрування контуру рН проводиться в класі рНCalibrateScreen. На етапі калібрування потрібен доступ до калібрувальних розчинів.

Очікування трихвилинного періоду є асинхронним з точки зору HW та SW. Протягом цього часу інтерфейс користувача оновлюватиметься кожні три секунди, інформуючи користувача про те, скільки часу залишилося до закінчення процесу калібрування. Однак користувач не зможе нічого зробити, окрім очікування виконання калібрування. Оновлення інтерфейсу користувача зроблено для того, щоб користувач не думав, що пристрій завис.

Однак протягом цього часу програмне забезпечення все ще опитує HW, щоб отримати рівень рН, і воно встановить рівень рН для цього калібрування лише після закінчення часу.

### **Контролер головного контуру**

Контролер є основним контуром ескізу Arduino. Це діє як спосіб зв'язати інтерфейс і механізм разом. Найкращий спосіб пояснити це, показавши:

```
voidloop()
{
    simpleTimer.run(); loopCounter++;
    //Перевірка інформації датчика кожні 5 с ( 0,1 с*50 )
    if( (loopCounter%50) == 0 )
    {
        loopCounter = 0;
        //Одержання/оновлення інформації про датчики
        Hydro_Eng.pHLevel();
        Hydro_Eng.airTemperatureHumidity();
        Hydro_Eng.dateAndTime();
        Hydro_Eng.waterTemp();
        Hydro_Eng.lightValue();
        Hydro_Eng.ECLevel();
    }
}
```

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 38   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

//Якщо датчики або час показують дані, які не потріні в даний момент

```
//if( Hydro_Eng.doesScreenNeedUpdating() )
{
    //Оновлення інтерфейсу користувача останньої інформації
    про датчик
    Hydro_UI.updateInfoFromSensors(
    Hydro_Eng.getSensorInformation());
    //Оновити екран новою інформацією.
    Hydro_UI.refreshScreen();
    //Інформувати движок, що інтерфейс користувача оновлено.
    Hydro_Eng.screenUpdated();
}
//Будь-яка більша затримка та сенсорний екран не реагує.
//Достатньо швидко.
delay(100); //цикл кожні 100 мс/0,1 с
// при доторканні екрана.
Hydro_UI.CheckForScreenTouch();
}
```

Контролер досить простий. Він отримує останні значення від апаратних датчиків, і якщо вони змінилися з моменту останнього оновлення, інтерфейс користувача буде оновлено відповідно до цієї інформації.

Датчики перевіряються кожні п'ять секунд, оскільки цього часто достатньо, щоб отримати враження про безперервні оновлення в реальному часі. Таймер, який відображається на екрані, має точність до хвилини, тому не становить жодних проблем.

Затримка додається, оскільки це зменшить навантаження на апаратне забезпечення та дещо зменшить вихідну потужність пристрою. Якщо значення більше, інтерфейс користувача помітно впливає на те, що він не реагує на дотики досить швидко.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 39   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Оновлення значень для рівня ЕС і рН надходять не безпосередньо з бібліотек HW, а через зворотний виклик послідовного порту void serialEvent2() для рівня рН і voidserialEvent3() для рівня ЕС.

|            |             |                 |               |             |                    |      |
|------------|-------------|-----------------|---------------|-------------|--------------------|------|
|            |             |                 |               |             | 123. УДК 004:681.5 | Арк. |
|            |             |                 |               |             |                    | 40   |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                    |      |



## Розділ 4. Розробка інтерфейсу

Інтерфейс користувача буде розроблений так, щоб зробити продукт приємним у використанні. Проте графіка буде обмежена через обмеження апаратного забезпечення. Наприклад, використання SD-карти для завантаження більш модної графіки.

Щоб переконатися, що інтерфейс користувача буде простим у використанні, було складено план тестування інтерфейсу користувача, і люди, які нічого не знали про дизайн, були використані для його випробування.

У цьому розділі детально описується оцінка інтерфейсу користувача, який було створено для контролера гідропоніки та потребує перевірки зручності використання. Під час тестування інтерфейс користувача був на ранній стадії розробки, і результати використовувалися, щоб побачити, наскільки легко користувач може виконати два завдання. На основі результатів інтерфейс користувача можна змінити, щоб підвищити зручність використання системи. Два завдання, на яких ми тут зосередимося, це:

- калібрування ЕС (Електропровідність) HW готовий до використання;
- встановлення необхідних рівнів рН.

Ці два завдання було вибрано, оскільки вони дуже точно відповідають іншим операціям, які потрібно виконати на пристрої. Наприклад, налаштування рН HW і налаштування рівнів повітря, води та ЕС майже однакові. Отже, виконання цих двох тестів перевіряє значну частину інтерфейсу користувача.

Кінцева мета цього дослідження – зробити пристрій більш приємним у використанні. Зворотній зв'язок, отриманий під час тестування юзабіліті, буде повернуто до розробки системи. Є надія, що цей цикл триватиме до завершення розробки. Тут детально описано лише один цикл, але методологія залишиться незмінною.

### 4.1. Сценарії використання

У цьому дослідженні зручності використання детально розглянемо два випадки використання. Багато операцій у системі схожі одна на одну, і з цієї

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 41   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

причини наші сценарії використання було вибрано так, щоб охопити якомога більше можливостей використання інтерфейсу користувача.

Як правило, користувач буде використовувати контролер, коли він оновив воду в резервуарі або коли він змінив рослини, які вони вирощують. Знову ж таки, наш вибір варіантів використання для вивчення охоплюватиме обидва ці сценарії та, сподіваємося, охоплюватиме максимальні функції системи.

### Перший тест: відкалібруйте рівень ЕС

У цьому випадку використання користувач щойно змінив воду для використання під час вирощування. Як правило, користувач змінює воду кожні кілька днів, щоб переконатися, що якість залишається достатньою для вирощування рослин. Перший крок – перейти до екрана ЕС і знайти екран калібрування. Ми припускаємо, що в кожному випадку використання користувач розпочне роботу з головного екрана (рис. 4.1).

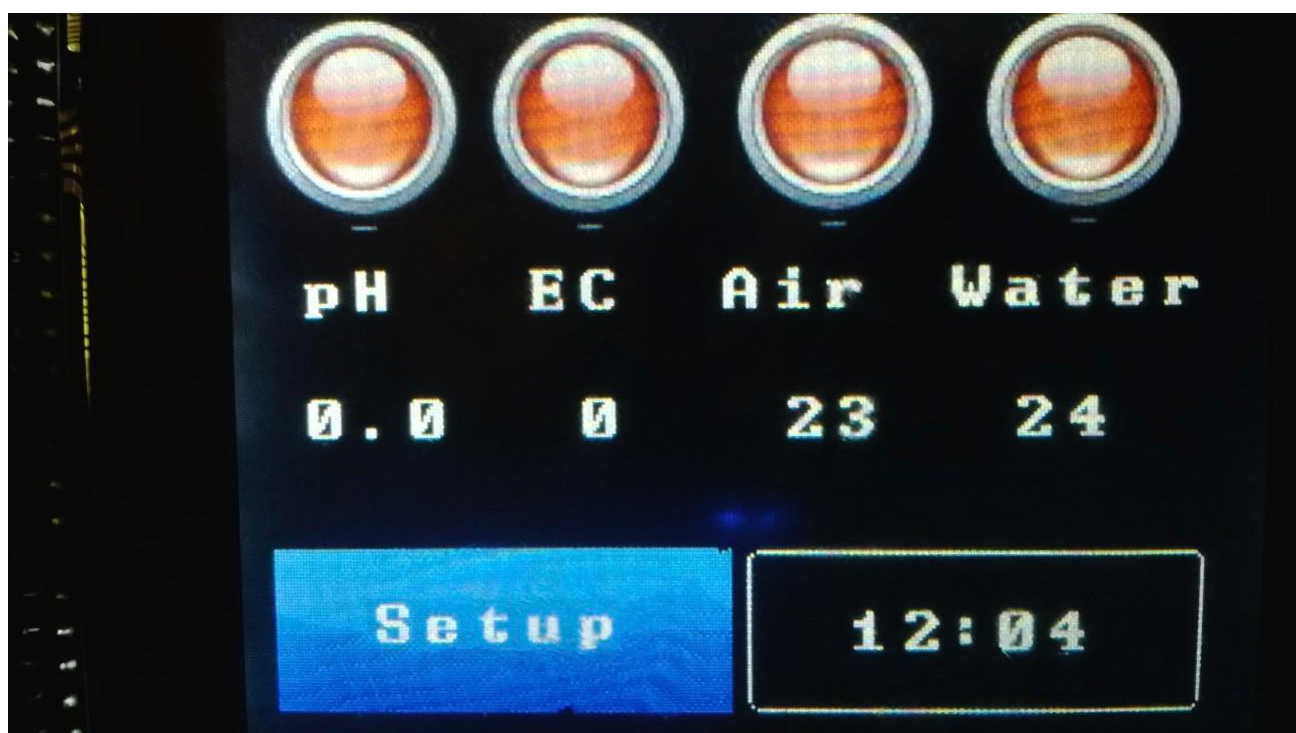


Рисунок 4.1 –Головний екран

Це цілком розумно припустити, оскільки головний екран – це місце, де більшість часу проводитиме пристрій, оскільки це екран, на якому користувач отримує детальну інформацію про поточні рівні та де відобразатимуться попередження, якщо певний рівень стане занадто високим або низьким.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 42   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Як видно з рис. 4.1 є чотири червоні кружечки, які повідомляють користувачеві, що задані рівні виходять за межі встановленого діапазону, а коли вони знаходяться в діапазоні, вони світяться зеленим. Вони також служать кнопками для переходу до екранів налаштування цієї функції. Натиснувши кнопку налаштування, користувач також може перейти до екранів налаштування. Меню налаштування переносить користувача на рис. 4.2, і звідки він може отримати доступ до тих самих функцій, що й на головному екрані, а також до кількох інших.

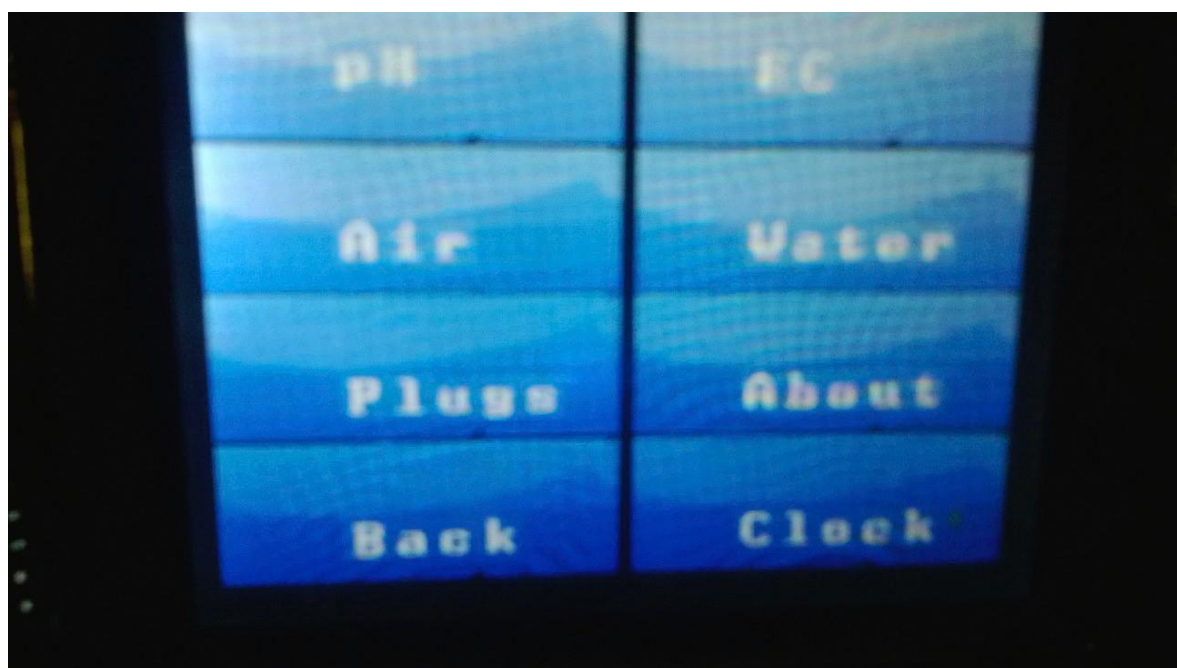


Рисунок 4.2 – Екран налаштувань

Починаючи з головного екрана, користувач має потім перейти до екрана ЕС. Як було сказано, це можна зробити двома способами:

1. Просто натисніть коло ЕС на головному екрані.
2. Натисніть Налаштування, а потім ЕС.

У цьому сценарії побачимо, чи легко користувачеві зрозуміти, що він може натиснути на червоне коло, щоб перейти до екрана ЕС.

Перейшовши на екран ЕС, користувач повинен натиснути кнопку налаштувань (рис. 4.3), а потім кнопку калібрування (рис. 4.4).

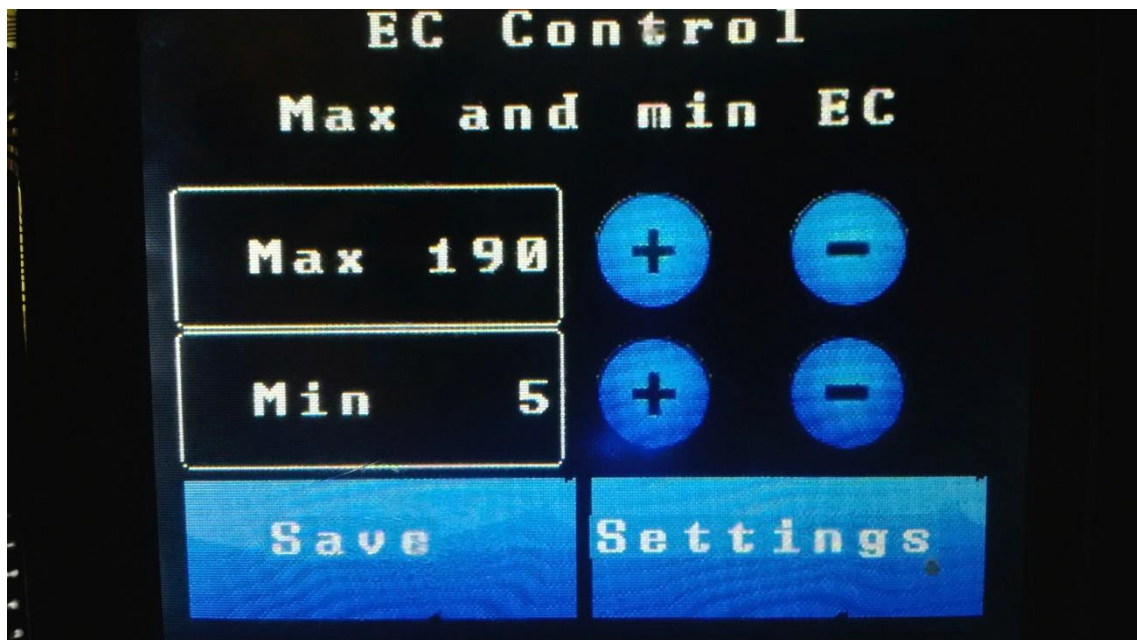


Рисунок 4.3 – Екран ЕС (натисніть Налаштування)

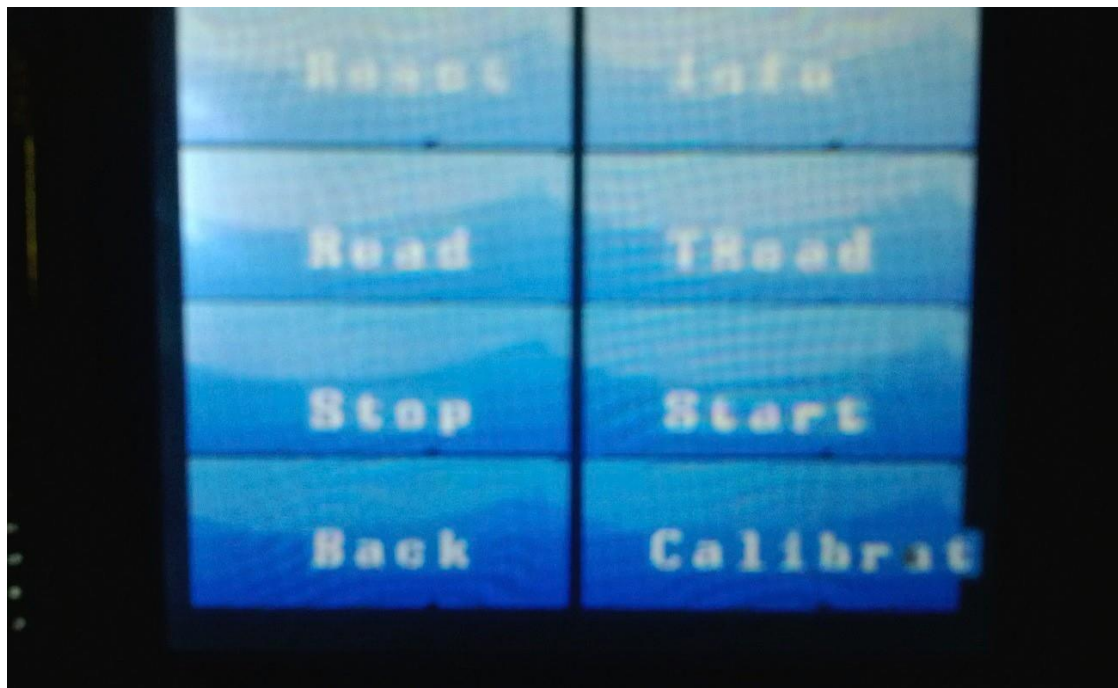


Рисунок 4.4 – Екран налаштувань ЕС – кнопка калібрування розташована  
внизу праворуч

Після цього з'являється екран калібрування з інструкціями щодо того, що потрібно зробити користувачеві, щоб виконати калібрування ЕС. Кожен екран чекатиме, поки користувач не натисне екран, після чого пристрій автоматично збереже дані з датчика ЕС і перейде до наступного етапу. Порядок фаз і час, який користувач повинен чекати, визначаються апаратним датчиком, тому ця

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 44   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

частина тесту є фіксованою та не може бути змінена, навіть якщо це впливає на зручність використання. На кожному екрані буде відображено, скільки часу залишилося до його завершення. Він відображає повідомлення «x seconds left unit completion». Нотатка оновлюється кожні три секунди. Оскільки для стабілізації показань потрібен деякий час, кожна фаза системи калібрування займе певний час, зазвичай близько п'яти хвилин.

Наприкінці процесу користувач отримує повідомлення, і він може вийти. Всього чотири фази, перша фаза показана на рис. 4.5.

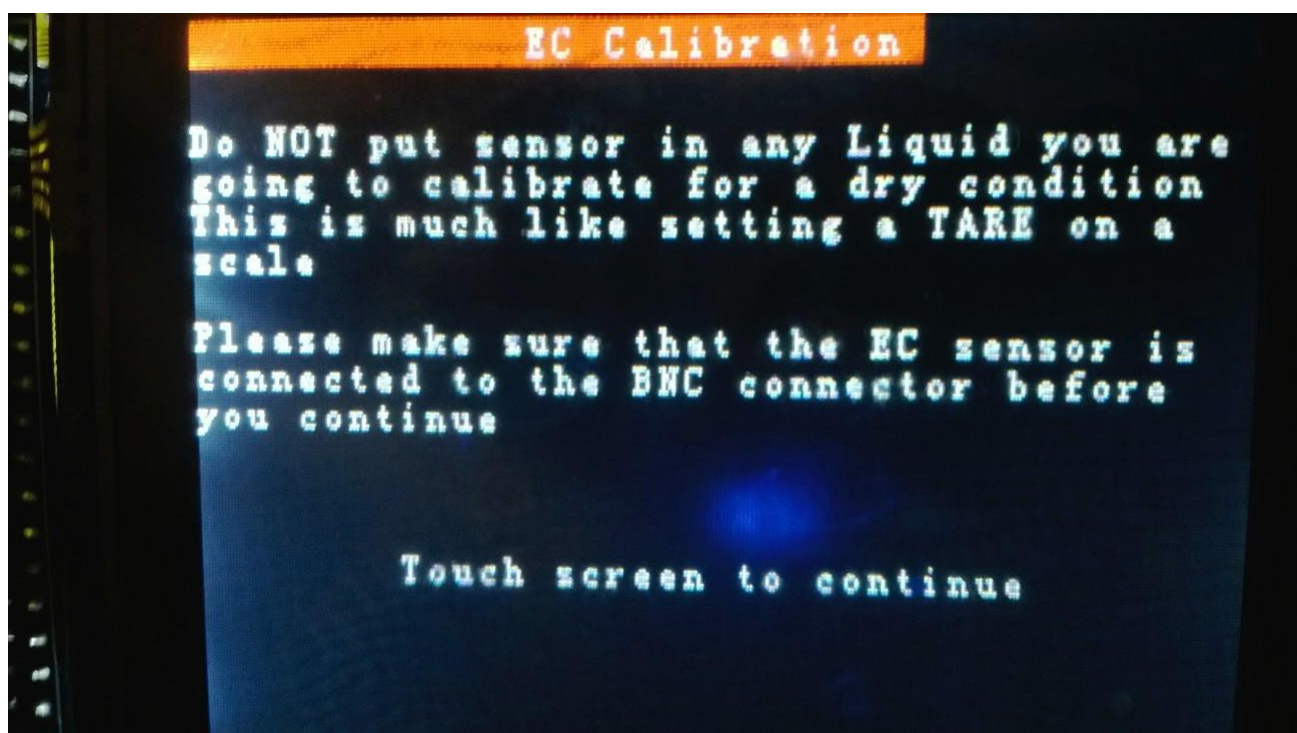


Рисунок 4.5 – Початок калібрування ЕС

Окрім використання пристрою, система надасть інструкції щодо використання датчика HW, щоб забезпечити безперебійне калібрування. У дослідженні зручності використання відстежувалося, наскільки легко користувач може слідувати інструкціям і поставити апаратний датчик у правильне положення.

Це той самий процес, що й під час калібрування рівня рН. Фактичні деталі, наведені на етапах екранів, відрізняються, але процес і схема калібрування з точки зору інтерфейсу користувача однакові. Отже, тут можна побачити, як працюватимуть обидва ці процеси.

|     |      |          |        |      |  |  |  |  |  |                    |      |
|-----|------|----------|--------|------|--|--|--|--|--|--------------------|------|
|     |      |          |        |      |  |  |  |  |  | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |  |  |  |  |  |                    | 45   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |                    |      |

## Тест 2: Встановлення відповідних налаштувань рівня рН

У цьому сценарії використання користувач щойно змінив рослини, які ростуть, і їм потрібен інший рівень рН, ніж той, який був раніше. Щоб змінити рівень рН, користувачеві потрібно перейти до екрана рН (рис. 4.6).

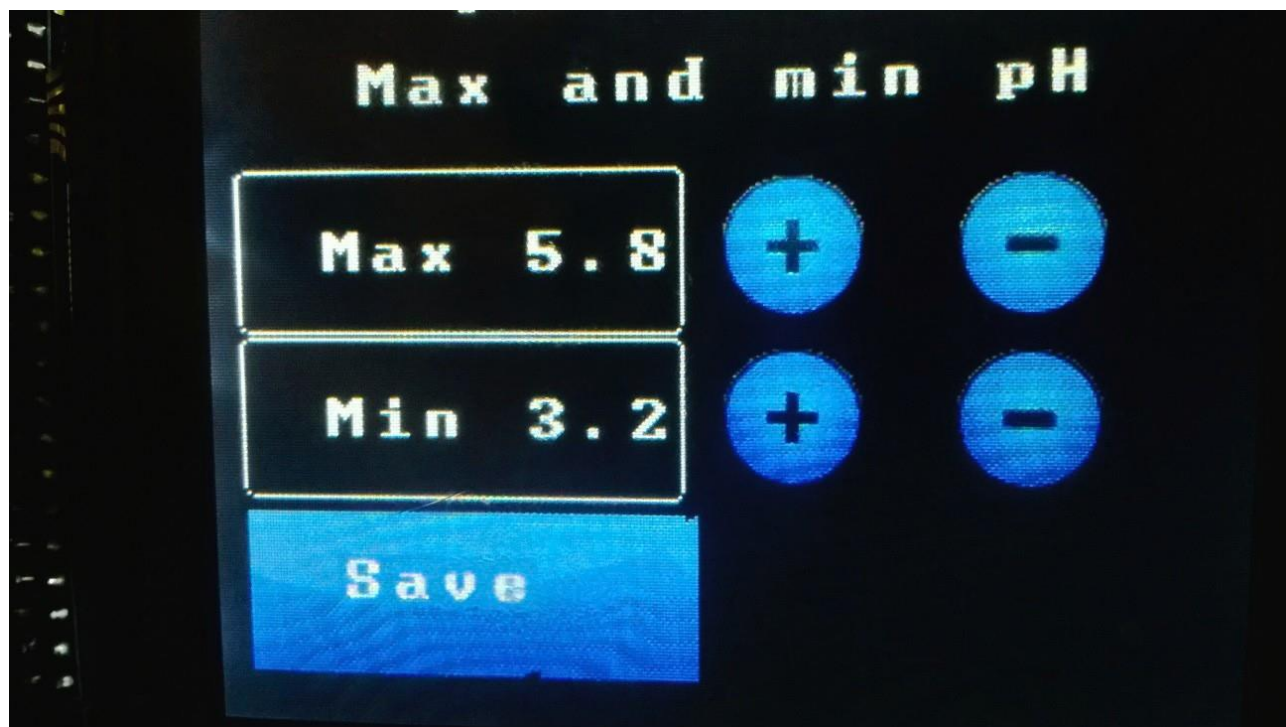


Рисунок 4.6 – Екран контролю рН

Отже, починаючи з головного екрана, користувач має потім перейти до екрана рН. Є два способи зробити це:

1. Просто натисніть коло рН на головному екрані.
2. Натисніть Налаштування, а потім рН.

Після того, як користувач перейде на екран рН, користувач побачить два вікна, в одному з яких написано min, а в іншому – max. У цих вікнах відображається максимальний і мінімальний рівень рН, до якого дозволяється досягати, перш ніж система повідомить користувача про те, що йому потрібно підвищити або знизити рівень рН. Користувач може змінювати рівні, натискаючи піктограми плюс і мінус праворуч. Потім користувач зберігає зміни, натиснувши кнопку «Зберегти».

Той самий процес застосовується під час налаштування температури повітря та води, а також рівня ЕС. Отже, цей один тестовий приклад перевіряє зручність використання всіх цих параметрів.

|     |      |          |        |      |  |  |  |                    |      |
|-----|------|----------|--------|------|--|--|--|--------------------|------|
|     |      |          |        |      |  |  |  |                    | Арк. |
|     |      |          |        |      |  |  |  |                    | 46   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  | 123. УДК 004:681.5 |      |

Знову головна увага тут буде зосереджена на тому, наскільки легко користувач може зрозуміти значення екранів і виконати необхідні завдання.

Після створення схеми можна завантажити програмне забезпечення. Якщо на вашому комп'ютері не встановлено Arduino IDE, завантажте його з веб-сайту Arduino. Далі вам потрібно переконатися, що наведене нижче зовнішні бібліотеки знаходяться в папці ArduinoLibraries [7], [16], [19], [20], [21], [22], [23], [24]

Після цього відкрийте файл HydroControl\_UI.ino і натисніть «Перевірити». Якщо є якісь помилки, швидше за все через відсутність пакета, виправте їх. Після компіляції підключіть Arduino до комп'ютера та натисніть «Завантажити».

Після цього ви повинні побачити головний екран. Першим кроком є встановлення часу та калібрування контурів рН та ЕС.

#### **4.2 Можливості майбутнього розвитку**

Існує кілька можливих вдосконалень як для апаратного, так і для програмного забезпечення. Розглянемо деякі з них:

1. Додати можливість підключення до Інтернету та оновлювати дані на сервері.
2. Розробити комп'ютерну програму, яка дозволить зчитувати дані з апаратного забезпечення та зручно їх відображати користувачу.
3. Розширити функціональність шляхом підтримки Bluetooth, щоб контролер міг взаємодіяти зі смартфонами та комп'ютерами.
4. Поліпшити графічний інтерфейс користувача, зробивши його більш привабливим та зручним у використанні.
5. Додати можливість багатомовної підтримки, щоб програмне забезпечення могло працювати на різних мовах.

Ці пункти можуть бути втілені у подальших версіях апаратного та програмного забезпечення, сприяючи поліпшенню функціональності та задоволенню потреб користувачів.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 47   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## ВИСНОВКИ

1. В цій дипломній роботі досліджено гідропоніку і ключові складові успішного вирощування цим методом. Результати дослідження використовувалися для проектування програмного та апаратного забезпечення контролера.

2. Апаратне забезпечення було вибрано з урахуванням вимог проекту. Наприклад, було перевірено, чи відповідають пристрої необхідним температурам, чи є вони достатньо точними, як з ними взаємодіяти та яка їх ціна. Arduino Mega 2560 був обраний як контролер через свою підтримку, відкритий код, багато контактів і можливостей для під', а також великий простір для програмного забезпечення.

3. Схема контролера була створена після створення прототипу на макетних платах, що допомогло забезпечити надійну систему. Після цього було розроблено програмне забезпечення, яке використовувало функції апаратного забезпечення, наприклад, взаємодію з схемою рН і ЕС через послідовні порти або RTC через інтерфейс SPI. Це приводило до покращення якості та скорочення часу розробки. Код інтерфейсу користувача та апаратного забезпечення були розроблені окремо, щоб забезпечити можливість оновлення апаратного забезпечення з невеликими змінами в інтерфейсі користувача.

4. Звіт про зручність інтерфейсу користувача був проведений для переконання в його простоті та інтуїтивній зрозумілості. Зворотній зв'язок з цього звіту був використаний для вдосконалення програмного забезпечення. Майбутній розвиток проекту може включати подальше поліпшення інтерфейсу з урахуванням отриманого зворотного зв'язку.

5. Останнім етапом проекту було тестування. Основне тестування проводилося шляхом імітації реальних сценаріїв та перевірки пристрою на правильну роботу. Для забезпечення якості коду можливо було б також розробити модульні тести, що охоплюють весь код під час його написання.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 48   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |



## СПИСОК ЛІТЕРАТУРИ

1. Домашня система виробництва овочів /доктор Гектор Муньос, 2010.
2. Гідропоніка: [Веб-сайт]. URL: <http://en.wikipedia.org/wiki/Hydroponics> (дата звернення: 02.03.2023).
3. ЕС порти TDS: [Веб-сайт]. URL: [http://www.hydroponics.net/learn/debate\\_over\\_ec\\_and\\_tds.asp](http://www.hydroponics.net/learn/debate_over_ec_and_tds.asp) (дата звернення: 03.03.2023).
4. FAQ -Електропровідність (ЕС) в гідропоніці / Д. Фернандес, 2009.
5. Пристрої малого радіусу дії: [Веб-сайт]. URL: [https://en.wikipedia.org/wiki/Short\\_Range\\_Devices](https://en.wikipedia.org/wiki/Short_Range_Devices) (дата звернення: 02.03.2023).
6. Бібліотека UTFT / Г. Карлсен, 2014.
7. Вісім золотих правил дизайну інтерфейсу / Б. Шнайдерман.
8. 10 евристик юзабіліті дизайну інтерфейсу користувача / Дж. Нільсен, 1995.
9. HydroBuddy / Д. Фернандес, 2013.
10. ScienceinHydroponics: [Веб-сайт]. URL: <http://scienceinhydroponics.com/> (дата звернення: 11.03.2023)..
11. DHT11 Humidity&TemperatureSensor: [Веб-сайт]. URL: <http://www.micro4you.com/files/sensor/DHT11.pdf> (дата звернення: 11.04.2023).
12. DS18B20 1-провідний цифровий термометр із програмованою роздільною здатністю: [Веб-сайт]. URL: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> (дата звернення: 15.04.2023).
13. Надзвичайно точні RTC шини SPI з вбудованим кристалом і SRAM: [Веб-сайт]. URL: <http://datasheets.maximintegrated.com/en/ds/DS3234.pdf> (дата звернення: 15.04.2023).
14. Короткий вступ до послідовного периферійного інтерфейсу (SPI): [Веб-сайт]. URL: <http://www.arduino.cc/en/Reference/SPI> (дата звернення: 16.04.2023).
15. Бібліотека UTFT / Г. Карлсен, 2014

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 49   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

16. Підсистема моніторингу солоності: [Веб-сайт]. URL: [https://www.atlas-scientific.com/\\_files/\\_datasheets/\\_circuit/EC\\_Circuit\\_3.0.pdf](https://www.atlas-scientific.com/_files/_datasheets/_circuit/EC_Circuit_3.0.pdf) (дата звернення: 13.04.2023).

17. Підсистема моніторингу рНмікрівідбитка: [Веб-сайт]. URL: [https://www.atlas-scientific.com/\\_files/\\_datasheets/\\_circuit/pH\\_Circuit\\_5.0.pdf](https://www.atlas-scientific.com/_files/_datasheets/_circuit/pH_Circuit_5.0.pdf) (дата звернення: 13.04.2023)..

18. Бібліотека UTouch / Г. Карлсен, 2014. .

19. Бібліотека кнопок UTFT / Г. Карлсен, 2014.

20. 433 МГц для Arduino, 2013.

21. Бібліотека SW DS3234 / Г. Карлсен, 2014.

22. Далласька бібліотека контролю температури: [Веб-сайт]. URL: [http://www.milesburton.com/?title=Dallas\\_Temperature\\_Control\\_Library](http://www.milesburton.com/?title=Dallas_Temperature_Control_Library) (дата звернення: 13.04.2023).

23. Клас DHT11 для Arduino / Робтіллаарт, 2013.

24. HydroController / М. Гріффітс, 2014.

25. Дизайн повсякденних речей. Основнікниги / Н.А. Норман, 2002.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | 123. УДК 004:681.5 | Арк. |
|     |      |          |        |      |                    | 50   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## ДОДАТКИ

### Додаток А

Таблиця 1 – Типи електричних датчиків

| Тип датчика | Тип води         | Діапазон              |
|-------------|------------------|-----------------------|
| K0.1        | Чиста питна вода | 11мкС/см – 3000мкС/см |
| K1.0        | Прісна вода      | 1.3 С/см – 40 С/см    |
| K10         | Солона вода      | 36 С/см – 92 С/см     |

Таблиця 2 – Опис контактів ЕС

| Pin     | Опис  |
|---------|---|
| GND     | Земля.  |
| TX      | Вихідні дані поставляються в асинхронному серійному форматі TTL Rs-232. |
| RX      | Вхід отримує асинхронні серійні дані в форматі TTL Rs-232.              |
| VCC     | Працює при напрузі 5В.  |
| PRB     | Підключається до BNC роз'єму для передачі даних зондом.                 |
| PRB GNB | Підключається до BNC роз'єму для заземлення зонда.                      |

Додаток Б

Таблиця 3 – Команди схеми ЕС

| Команда | Функція   | Зворотній зв'язок | Опис  |
|---------|---|-------------------|---|
| R       | Повертає одне значення ЕС/TDS/солоності.                    | ЕС, TDS, SAL      | <p>ЕС означає електричну провідність в мкС/см (мікросіменс на сантиметр).</p> <p>TDS означає загальну розчинену речовину (виражену відносно КСІ).</p> <p>SAL означає солоності (за шкалою практичної солоності 1978 року) і виражається лише як ціле число.</p>   |
| ТТ.Т    | Аналогічно R, повертає температурно скомпенсоване значення. | ЕС, TDS, SAL      | <p>ЕС (Electrical Conductivity) вимірює електричну провідність розчину та виражається в одиницях мікросіменсів на сантиметр (мкС/см).</p> <p>TDS (Total Dissolved Solids) вказує на загальну кількість розчинених речовин у воді і виражається в одиницях, які посилаються на розчин КСІ.</p> <p>SAL (Salinity) вимірюється за шкалою практичної солоності 1978 року і виражається лише цілими числами. Вона вказує на рівень солоності води.</p> |

| Команда | Функція  | Зворотній зв'язок      | Опис   |
|---------|--|------------------------|--|
| C       | Повертає неперервне значення зчитування кожні 1000 мс при попередньо використаній температурі. | ЕС, TDS, SAL           | ЕС представляє собою електричну провідність у одиницях мікросіменсів на сантиметр (мКС/см).<br>TDS вказує на загальну кількість розчинених речовин у воді, виміряну відносно KCl.<br>SAL означає солоність за шкалою практичної солоності 1978 року і виражається лише цілими числами. Вона вказує на рівень солоності води. |
| E       | Зупиняє всі зчитування, і схема переходить в режим очікування.                                 | Відсутній              |  |
| X       | Виконує скидання до заводських налаштувань схеми.  | Заводські налаштування | Схема E.C. (електропровідність) відповість, припинивши передачу даних: "Немає ASCII-відповіді на цю команду".  |
| I       | Повертає інформацію про схему.   | E, V3.0, 4/11          | E = E.C. Схема<br>V3.0 = Версія прошивки<br>4/11 = Дата написання прошивки   |

|     |      |          |        |      |                    |  |      |
|-----|------|----------|--------|------|--------------------|--|------|
|     |      |          |        |      | 123. УДК 004:681.5 |  | Арк. |
|     |      |          |        |      |                    |  | 53   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |  |      |

Додаток В

Таблиця 4 – Команди калібрування схеми ЕС

| Команда | Функція  | Зворотній зв'язок                     | Опис                  |
|---------|--|---------------------------------------|-----------------------|
| P,1     | Повідомляє схемі, що вона є типу 0.1                     | K0.1                                  | Все ОК                |
| Z0      | Інформує Е.С. схему про калібрування для сухого датчика. | Сухе калібрування                     | Калібрування виконане |
| Z30<CR> | Виконує калібрування високого рівня для розчину 3 000.   | Калібрування зі значенням 90000мкС/см | Калібрування виконане |
| Z2      | Виконує калібрування низького рівня для розчину 220.     | Калібрування зі значенням 220мкС/см   | Калібрування виконане |

Додаток Г

Таблиця 5 – Команди контуру рН

| Команда | Функція  | Зворотній з'язок | Опис  |
|---------|--|------------------|---|
| R       | Вказує рН-схемі повернути одне значення рН.  | xx.xx            | Де xx.xx - це деяке значення рН.  |
| ТТ.ТТ   | При передачі температури до рН-схеми буде повернуто значення рН, скомпенсоване за температури.                       | xx.xx            | Де xx.xx - це деяке значення рН. Дані про температуру будуть втрачені, якщо схема втратить живлення.                        |
| С       | ПХ-схема працюватиме в постійному режимі і буде постійно надсилати значення рН кожні 378 мс до отримання команди "Е" | xx.xx            | Якщо рН-схема виявляє, що датчик рН не підключений або пошкоджений, вона відповідь повідомленням про помилку «check probe». |
| Х       | Зупиняє всі вимірювання, а схема переходить в режим очікування (standby mode).                                       | Відсутнє         | рН- схема відповідь, припинивши передачу даних: «Немає ASCII-відповіді на цю команду».                                      |
| Е       | Виконує скидання до заводських налаштувань схеми (factory reset).  | reset            | рН- схема відповідь «reset»   |
| I       | Повертає інформацію в схему  | P, V5.0, 5/13    | P = Схема рН<br>V5.0 = Версія прошивки<br>5/13 = Дата написання прошивки  |

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

Додаток Г

Таблиця 6 – Команди калібрування контуру рН

| Команда | Функція  | Зворотній зв'язок | Опис                  |
|---------|--|-------------------|-----------------------|
| S       | Вказує схемі провести калібрування для розчину рН7.  | 7.00              | Все ОК                |
| F       | Вказує схемі провести калібрування для розчину рН4.  | 4.00              | Калібрування виконане |
| T       | Вказує схемі провести калібрування для розчину рН10. | 10.00             | Калібрування виконане |