

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій

(повне найменування факультету)

Кафедра комп'ютерних систем та мереж

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти)

на тему «Розробка системи дистанційного керування поливом з контролем
стану рослин»

(назва теми)

Виконав: студент 4 курсу, групи КНТ-519

Спеціальності _____

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерна інженерія

ЧОРНИЙ Д.О.

(ПРИЗВИЩЕ та ініціали)

Керівник

СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент

САМОЙЛИК С.С.

(ПРИЗВИЩЕ та ініціали)

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
	СКРУПСЬКИЙ С.Ю., к. т. н., доцент		
Нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання «8» травня 2023року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
	Аналіз технічного завдання	1 тиждень	виконано
	Аналіз структури роботи тепличних господарств	1 тиждень	виконано
	Вибір відповідної технології для створення системи	1 тиждень	виконано
	Підбір комплектуючих елементів для створення системи, схеми підключення	2 тиждень	виконано
	Написання програмного коду	3 тиждень	виконано
	Реалізація проєкту	3 тиждень	виконано
	Оформлення проєкту	3 тиждень	виконано
	Проходження нормоконтролю	4 тиждень	виконано
	Захист роботи	4 тиждень	виконано

Студент(ка)


 (підпис)
Данило ЧОРНИЙ

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)


 (підпис)
Степан СКРУПСЬКИЙ

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 60 с., 15 рис., 3 табл., 12 джерел.

АВТОМАТИЗАЦІЯ, ІОТ, ІНФОРМАЦІЙНА СИСТЕМА, СИСТЕМА БЕЗПЕКИ

В цій дипломній роботі розглянуті питання, пов'язані з автоматизацією процесів, що використовуються в тепличних господарствах. Було створено дослідний зразок системи віддаленого доступу з контролем стану рослин.

У процесі розробки було встановлено, що система, що працює в тандемі Arduino + Shield, обслуговує набір інструментів за протоколами http, тоді як спочатку запланований telegram-бот, який, як я знову нагадаю, не відрізняється інтуїтивно зрозумілим інтерфейсом для непрофесійного користувача, працює із протоколами https. Більше того, навіть якщо систему оптимізувати шляхом прошивки софтом, у процесі налаштування користувач зіткнувся б з протоколами безпеки, які систематично змінюватимуться – це підтверджує довідкова інформація Telegram, де зазначено, що месенджер систематично проводить подібні маніпуляції під час оновлення ключів наскрізного шифрування. Таким чином, система могла б стати вкрай незручною в процесі експлуатації через підвищену увагу з боку споживача.

Проект, який розглядається в цій дипломній роботі, можна назвати домашньою теплицею, і це досить незвичайно розглядати в концепції розумного будинку, оскільки подібні розробки відносно нещодавно порівняно зі звичним розумінням систем розумного будинку з'явилися на ринку.

ЗМІСТ

Вступ.....	6
1 Розвиток технологій у сфері створення автоматизованих систем.....	8
1.1 Поняття та функції розумного будинку.....	8
1.2 Історія створення та розвитку систем розумних будинків.....	13
1.3 Особливості тепличних господарств.....	19
1.4 Розвиток автоматизованих систем тепличних господарств	21
2 Конструкторська частина Розробка системи	25
2.1 Аналіз варіантів елементів приладу та вибір компонентів	25
2.2 Обґрунтування апаратної частини.....	29
2.3 Обґрунтування програмної частини.....	30
2.4 Технічні характеристики обраних елементів	32
3 Моделювання раціонального модельного складу системи захисту інформації	45
3.1 Побудова моделі системи захисту інформації.....	45
3.2 Розробка моделей протидії загрозам інформації безпеки в умовах невизначеності	50
Висновки	55
Перелік джерел посилання	57

Перелік графічного матеріалу:

Пл1 – Схематичне зображення системи;

Пл2 – Графіки функцій належності показника рівня інформаційних ризиків у КІС;

Пл3 – Реалізація схеми системи;

Пл4 – Реалізація системи;

ВСТУП

Життю сучасного суспільства стає все більше завдань у повсякденному житті, все більше потрібно встигнути зробити по роботі, по дому і при цьому ще залишити час для себе, для своїх улюблених занять та для близьких людей, а часу на добу – всього 24 години. Щодня доводиться робити багато рутинних справ, які роблять наше життя в результаті комфортним і звичною, але займають багато часу. Системи «розумного будинку» існують і створюються, удосконалюються для того, щоб спростити наше життя та зменшити кількість неприємних, але потрібних та важливих справ. У систем «розумного будинку» існують різні функції: як створення комфортних умов життя, як безпека людей, як підтримання здорової життєдіяльності за рахунок регулювання мікроклімату квартир і багато інших, залежно від попиту споживача. Телефони в 21 столітті – невід'ємна частина життя, а іноді й людини. Можливості телефону стають дедалі ширшими, вже не тільки дзвінки, спілкування та робота, але й керування простором, в якому ми живемо, навіть на відстані, у тому числі й керування мікрокліматом у домашніх міні-теплицях. Розумні теплиці та різні подібні комплекси для вирощування рослин дають багато можливостей у наш час. Крім того, що жителі міст можуть самі отримувати свою продукцію для домашнього використання, ще можуть бути впевненими в тому, як відбувався процес росту рослин, без шкідливих речовин і хімічних реагентів. людство розвивається, дедалі більше людей беруть під контроль своє харчування, багато хто веде вегетаріанський спосіб життя. ці тенденції тільки посилюються з кожним роком, тому так важливо розуміти, що попит на свої продукти, вирощені правильно, в яких є корисні вітаміни не через хімічні добавки, а через оптимальні умови зростання, тільки збільшуватиметься. як відбувався процес зростання рослин, без шкідливих речовин та хімічних реагентів. людство розвивається, дедалі більше людей беруть під контроль своє харчування, багато хто веде вегетаріанський спосіб життя. ці тенденції тільки посилюються з кожним роком, тому так важливо розуміти, що попит на свої

продукти, вирощені правильно, в яких є корисні вітаміни не через хімічні добавки, а через оптимальні умови зростання, тільки збільшуватиметься. як відбувався процес зростання рослин, без шкідливих речовин та хімічних реагентів. людство розвивається, дедалі більше людей беруть під контроль своє харчування, багато хто веде вегетаріанський спосіб життя. ці тенденції тільки посилюються з кожним роком, тому так важливо розуміти, що попит на свої продукти, вирощені правильно, в яких є корисні вітаміни не через хімічні добавки, а через оптимальні умови зростання, тільки збільшуватиметься.

Проект, який розглядається в цій дипломній роботі, можна назвати домашньою теплицею, і це досить незвичайно розглядати в концепції розумного будинку, оскільки подібні розробки відносно нещодавно порівняно зі звичним розумінням систем розумного будинку з'явилися на ринку.

1 РОЗВИТОК ТЕХНОЛОГІЙ У СФЕРІ СТВОРЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ

1.1 Поняття та функції розумного будинку

Поняття та функції розумного будинку. Розумний будинок - це концепція, заснована на використанні сучасних інформаційних і комунікаційних технологій для автоматизації та управління різними аспектами житлового середовища. Він об'єднує різні пристрої та системи, даючи змогу створити інтелектуальне та зручне середовище для життя. У цій статті ми розглянемо поняття і функції розумного будинку, а також переваги, які він може запропонувати.

Розумний дім - це сучасна концепція, яка об'єднує різні технології для створення інтелектуального та автоматизованого домашнього середовища. У цій статті ми розглянемо поняття і функції розумного будинку, а також розповімо про переваги, які він пропонує власникам.

Визначення розумного будинку. Розумний дім - це будинок, у якому встановлено різні пристрої та системи, здатні взаємодіяти між собою і виконувати різні завдання автоматично або за командою користувача. Вони можуть бути під'єднані до центральної системи управління, яка дає змогу контролювати та керувати різними аспектами домашнього середовища. Основні функції розумного будинку.

Управління освітленням: Розумний будинок дає змогу контролювати освітлення в різних частинах будинку. Це включає можливість регулювати яскравість, вмикати і вимикати світло за розкладом або за певних подій, а також використовувати датчики руху для автоматичного вмикання світла за необхідності.

Керування опаленням і кондиціонуванням повітря: Розумний будинок надає можливість контролювати температуру в різних приміщеннях за допомогою термостатів і датчиків. Це дає змогу оптимізувати використання енергії та створювати комфортні умови для проживання.

Управління безпекою: Розумні будинки забезпечують високий рівень

безпеки завдяки системам відеоспостереження, датчикам руху, датчикам диму і чадного газу, сигналізаціям і системам контролю доступу. Власник може отримувати повідомлення про можливі проблеми і моніторити безпеку свого будинку навіть перебуваючи поза ним.

Керування енергоспоживанням: Розумний дім дає змогу оптимізувати енергоспоживання, даючи змогу власникові контролювати й керувати енергоефективністю різних пристроїв і систем. Наприклад, можна автоматично вимикати розетки або регулювати температуру опалення та кондиціонування повітря залежно від присутності людей у будинку.

Керування медіа-системами: Розумний дім дає змогу інтегрувати різні медіа-системи, як-от аудіо- та відеообладнання, системи домашнього кінотеатру, музичні системи та телевізори. Це дає змогу створити багатокімнатну аудіо- та відеосистему, а також контролювати їхню роботу за допомогою центральної системи управління.

Автоматизація рутинних завдань: Розумний будинок дає змогу автоматизувати рутинні завдання і процеси. Наприклад, можна налаштувати систему так, щоб після повернення додому автоматично вмикалося освітлення, а температура перебувала в комфортному діапазоні. Можна також налаштувати систему для автоматичного поливу рослин, увімкнення та вимкнення пристроїв і багато чого іншого.

Віддалений контроль: Розумний дім дає змогу власнику контролювати та керувати домашнім середовищем навіть перебуваючи поза домом. За допомогою мобільного застосунку або інтернет-порталу можна вмикати і вимикати пристрої, отримувати повідомлення про події та контролювати безпеку.

Переваги розумного будинку. Покращений комфорт: Розумний дім надає власнику комфортні умови проживання, даючи змогу контролювати різні параметри середовища, як-от освітлення, температуру, звук та інші, відповідно до індивідуальних уподобань.

Розумний будинок дає змогу знизити енергоспоживання, оптимізувати використання енергії і тим самим скоротити витрати на електрику та опалення.

Системи безпеки розумного будинку забезпечують захист від вторгнень, пожеж, витоків газу та інших потенційних загроз, а також надають власнику миттєві повідомлення про можливі проблеми.

Зручність і гнучкість: Можливість керувати розумним будинком через центральну систему управління або мобільний застосунок забезпечує власнику гнучкість і зручність в управлінні домашнім середовищем. Підвищена цінність нерухомості: Наявність розумної системи в будинку може підвищити його ринкову вартість і привабливість для потенційних покупців.

Майбутнє розумного будинку. Розумний будинок - це галузь, що швидко розвивається, і майбутнє обіцяє ще більше інновацій і можливостей. З появою нових технологій і розвитком інтернету речей, розумні будинки стають дедалі інтелектуальнішими й автоматизованішими. Очікується, що вони ставатимуть більш доступними та інтегрованими в наше повсякденне життя.

Розумний будинок - це концепція, яка дає змогу власникам контролювати й керувати різними аспектами свого будинку за допомогою технологій і автоматизації. Він пропонує безліч функцій, включно з керуванням освітленням, опаленням, безпекою, енергоспоживанням та іншими аспектами домашнього середовища. Розумний будинок надає зручність, комфорт, безпеку та енергоефективність, роблячи наше життя кращим і більш інтелектуальним.

Розумний будинок - це система, в якій різні пристрої та системи взаємодіють між собою, а також із власником будинку, з метою автоматизації та контролю різних процесів. Він пропонує управління освітленням, опаленням, кондиціонуванням повітря, безпекою, енергоспоживанням та іншими аспектами домашнього середовища. Основна ідея полягає в тому, щоб зробити будинок зручним, безпечним і енергоефективним, а також надати можливість віддаленого управління і контролю.

Функції розумного будинку. Керування освітленням: Розумний будинок дає змогу автоматизувати керування освітленням у будинку. Це може бути здійснено за розкладом, за допомогою датчиків руху або дистанційно через мобільний додаток. Це дає змогу заощадити енергію і створити комфортну атмосферу в

будинку.

Керування опаленням і кондиціонуванням повітря: Розумний будинок дає змогу контролювати температуру в різних приміщеннях, а також налаштовувати режими роботи системи опалення та кондиціонування повітря. Ви можете керувати цими системами з будь-якого місця за допомогою свого смартфона або планшета.

Безпека і відеоспостереження: Розумний дім пропонує різні функції безпеки, включно з системами відеоспостереження, датчиками руху, датчиками диму і витоку газу. Ви можете отримувати повідомлення про можливі загрози і контролювати безпеку вашого будинку, навіть перебуваючи далеко від нього.

Розумні замки і контроль доступу: За допомогою розумного будинку ви можете контролювати доступ до вашого будинку за допомогою розумних замків. Ви можете надавати тимчасові коди доступу для гостей або відкривати двері, використовуючи мобільний додаток.

Керування енергоспоживанням: Розумний дім дає змогу відстежувати й керувати енергоспоживанням у вашому домі. Ви можете контролювати використання енергії різними пристроями, регулювати споживання електроенергії та води, що дає змогу заощадити гроші та бути більш енергоефективним.

Інтеграція з пристроями Інтернету речей (IoT): Розумний будинок може інтегруватися з різними пристроями Інтернету речей, такими як розумні термостати, розумні колонки, розумні праски та інші. Це дає змогу створити єдине управління і контроль за всіма пристроями в будинку.

Переваги розумного будинку. Комфорт і зручність: Розумний будинок забезпечує високий рівень комфорту і зручності. Ви можете керувати різними аспектами вашого будинку за допомогою одного пристрою або додатка, що робить життя більш зручним і ефективним.

Енергоефективність: Розумний будинок дає змогу ефективно використовувати енергію, що призводить до скорочення витрат на електроенергію та зниження негативного впливу на навколишнє середовище.

Безпека: Розумний будинок пропонує розширені функції безпеки, які допомагають захистити ваш будинок від можливих загроз. Ви можете

контролювати доступ і моніторити події в режимі реального часу.

Гнучкість і масштабованість: Розумний будинок пропонує гнучкість у налаштуванні та розширенні системи. Ви можете додавати нові пристрої та функції в міру необхідності, що дає змогу вам адаптуватися до мінливих потреб.

Віддалений доступ: Розумний будинок надає можливість віддаленого управління і контролю за вашим будинком. Ви можете перевіряти статус пристроїв, керувати ними та отримувати повідомлення навіть поза домом.

Підвищена вартість нерухомості: Розумний будинок може збільшити вартість вашої нерухомості, оскільки він пропонує сучасні та затребувані функції, які можуть привабити покупців.

Розумний будинок - це еволюція технологій, яка дає змогу створити інтелектуальне і зручне середовище для життя. Він пропонує широкий спектр функцій, включно з керуванням освітленням, опаленням, безпекою, енергоспоживанням та іншими аспектами домашнього середовища. Розумний дім забезпечує комфорт, енергоефективність, безпеку та гнучкість, роблячи життя більш зручним і сучасним. Ця технологія має великий потенціал і буде продовжувати розвиватися, покращуючи наше життя і задовольняючи наші потреби в майбутньому.

Розумний будинок - це система пов'язаних пристроїв (взаємодіючих між собою або підключених до контролера), яка допомагає людині позбутися рутини за рахунок автоматизації процесів.

Такі системи поділяються на два варіанти управління: інтегровані та автономні, перші являють собою об'єднання кількох приміщень в одну мережу, що управляються з одного контролера, а другі це група індивідуальних рішень для кожного приміщення.

У приміщенні встановлюються різні датчики, які підключені до контролюючого пристрою, і вже залежно від отриманих даних система сама або людина, що має доступ до управління і контролю, регулює отримані показники і процеси, що відбуваються в будинку. Управління зазвичай відбувається з пульта керування, в наш час все частіше стає смартфон.

Найчастіше розумний будинок обмежується кількома основними функціями та керує: світлом, мікрокліматом, безпекою.

Однак розумний будинок може складатися з комплексу найрізноманітніших систем і пристроїв, крім звичних нам:

- інтелектуальні лічильники, розумні мережі та рішення, інтегровані з ними;
- системи внутрішнього позиціонування; в) системи поливу.система мікроклімату (опалення, вентиляція, кондиціювання, зволоження тощо);

1.2 Історія створення та розвитку систем розумних будинків

Системи розумних будинків - це результат еволюції сучасних технологій, яка дала змогу автоматизувати та керувати різними аспектами домашнього середовища. У цій статті ми розглянемо історію створення та розвитку систем розумних будинків, починаючи з їхніх перших появ до сучасних технологій, які ми бачимо сьогодні.

Перші кроки. Ідея розумного будинку з'явилася ще в 1960-х роках. Перші системи управління були простими і містили використання таймерів для автоматичного вмикання і вимикання освітлення та інших пристроїв. Це був перший крок до автоматизації домашнього середовища.

Системи розумних будинків є інноваційними технологіями, які дають змогу автоматизувати й інтегрувати різні аспекти життя в домі, забезпечуючи комфорт, безпеку, енергоефективність і зручність для його мешканців. У цій статті ми розглянемо історію створення і розвитку систем розумних будинків, починаючи з їхньої появи до сучасних тенденцій і перспектив.

Поява і ранні приклади розумних будинків. Концепція розумних будинків почала розвиватися в 1970-х роках, коли компанії та дослідницькі інститути почали експериментувати з автоматизацією різних систем у будинку. Ранні

прикладі включали автоматичне керування освітленням, опаленням і кондиціонуванням повітря за допомогою примітивних систем управління.

У 1980-х роках з'явилася перша система розумного будинку під назвою "ЕCHO IV", розроблена компанією "Echoflex Solutions". Ця система пропонувала інтеграцію різних систем у будинку, включно з освітленням, опаленням, охоронною сигналізацією та аудіо-відео системам.

Розвиток технологій і розширення функціоналу. З розвитком комп'ютерних технологій, бездротових мереж та Інтернету, системи розумних будинків стали більш доступними і функціональними. У 1990-х роках з'явилися перші приклади систем управління будинком з використанням персональних комп'ютерів і смартфонів.

На початку 2000-х років технології розумних будинків стали більш універсальними і стандартизованими. З'явився стандарт ZigBee, який забезпечував бездротовий зв'язок між різними пристроями в будинку. Це дало змогу створювати мережі розумних будинків, у яких різні пристрої можуть взаємодіяти між собою.

Останніми роками стандарт Wi-Fi став домінуючим у сфері розумних будинків, даючи змогу під'єднувати всі пристрої до мережі Інтернет. Також стали популярними пристрої з голосовим управлінням, як-от Amazon Echo і Google Home, які дають змогу керувати різними системами в будинку за допомогою голосових команд.

Основні функції та можливості розумного будинку. Системи розумних будинків пропонують широкий спектр функцій і можливостей, які покращують якість життя і забезпечують комфорт і безпеку для мешканців. Деякі з основних функцій розумного будинку включають. Керування освітленням: Розумні будинки дають змогу програмувати і контролювати освітлення в різних частинах будинку за допомогою сенсорів і пристроїв керування.

Контроль температури: Системи розумного будинку дають змогу керувати опаленням і кондиціонуванням повітря, забезпечуючи комфортну температуру в будинку.

Розумні будинки пропонують системи відеоспостереження, сигналізації,

контролю доступу та дистанційного моніторингу, забезпечуючи безпеку і захист будинку.

Управління енергоспоживанням: Системи розумного будинку дають змогу оптимізувати енергоспоживання, автоматично керувати енергоощадними пристроями та моніторити споживання енергії.

Автоматизація рутинних завдань: Розумні будинки дають змогу автоматизувати рутинні завдання, як-от полив рослин, прибирання, закриття та відкриття вікон і дверей.

Розумний домашній кінотеатр і розваги: Системи розумних будинків пропонують інтеграцію аудіо-відео систем, мультимедійного контенту та управління домашнім кінотеатром.

Розвиток систем управління. У 1970-х і 1980-х роках з'явилися більш просунуті системи управління, які використовували програмування і контролери для автоматизації різних процесів у будинку. Такі системи давали змогу керувати освітленням, опаленням, кондиціонуванням повітря та іншими аспектами домашнього середовища.

Поява домашніх комп'ютерів. У 1990-х роках із розвитком комп'ютерної технології стало можливим використання домашніх комп'ютерів для управління системами розумного будинку. Було розроблено спеціальні програми та інтерфейси, які давали змогу власникам будинків контролювати різні пристрої та системи через комп'ютер.

Розвиток мережевих технологій. З появою бездротових і мережевих технологій на початку 2000-х років системи розумних будинків стали ще більш доступними і гнучкими. Було розроблено стандарти зв'язку, як-от Zigbee, Z-Wave і Wi-Fi, які давали змогу різним пристроям взаємодіяти між собою і з центральною системою управління.

Поява Інтернету речей (IoT). З розвитком Інтернету речей (IoT) останніми роками системи розумних будинків стали ще більш інтегрованими і розумними. Пристрої та системи в будинку можуть бути під'єднані до інтернету і взаємодіяти один з одним, обмінюючись даними і виконуючи різні завдання. Це дає змогу

власникам будинків контролювати та керувати всіма аспектами домашнього середовища навіть віддалено через мобільні пристрої.

Розвиток і доступність технологій. Сьогодні системи розумних будинків стали більш доступними і широко поширеними. Різні компанії пропонують різноманітні пристрої та системи для розумних будинків, які можуть бути легко встановлені та налаштовані в домашньому середовищі. Багато з цих пристроїв мають інтуїтивно зрозумілі інтерфейси і можуть бути інтегровані з іншими пристроями та сервісами.

Системи розумних будинків пройшли довгий шлях розвитку, починаючи від простих таймерів і програмних контролерів до сучасних мережевих технологій та інтернету речей. Сьогодні розумні будинки пропонують широкий спектр функцій, включно з керуванням освітленням, опаленням, безпекою, енергоспоживанням та іншими аспектами домашнього середовища. Вони стали більш доступними і гнучкими, надаючи зручність, комфорт, безпеку та енергоефективність для власників будинків. Майбутнє розумних будинків світле, і ми можемо очікувати подальшого розвитку та інновацій у цій галузі.

Вважається, що історія таких розумних систем починається з появи перших електричних побутових приладів у 20 столітті: холодильник, праска, пральна машина тощо.

Одним із перших проектів розумного будинку став будинок американського інженера Еміля Матіаса ще 1950 року. Інженер розмістив у своєму будинку з кнопками для їх управління. Він міг у такий спосіб регулювати відкриття дверей гаража та включення радіо дистанційно, встановив осмотичні системи сигналізації.

Це був не єдиний подібний проект у середині 20 століття. Іншим проектом того періоду стало ECHO IV (Electronic Computing Home Operator) у 1960 році Джим Сазерленд. За допомогою цього пристрою інженер міг регулювати температуру в будинку та включати прилади в будинку.

Важливим етапом розвитку автоматизації у будинку стала розробка стандарту для електричної техніки X10 у 1975 році.

1987 року в було опубліковано ідею розумного будинку «СФІНКС». Це був радіокерований комплекс, до якого входили три пристрої видів призначень: що носяться; призначені для дому; призначені для автомобіля.

Наприкінці 20 століття тема розумного будинку стає дедалі популярнішою. У 1999 році студія Disney випустила фільм «Розумний дім», що розповідає про автоматизований будинок на чолі з роботом-покоївкою. У новому тисячолітті ця тема піднімалася в кіно і не лише дедалі частіше.

Тенденції та перспективи. Системи розумних будинків продовжують розвиватися й інновірувати, відкриваючи нові можливості та перспективи для мешканців і розробників. Деякі з поточних тенденцій включають.

Інтеграція зі штучним інтелектом і машинним навчанням: Розумні будинки стають дедалі інтелектуальнішими, завдяки впровадженню штучного інтелекту і машинного навчання. Це дає змогу системам розумного будинку адаптуватися до вподобань і поведінки мешканців, оптимізувати енергоспоживання і пропонувати персоналізовані рекомендації.

Розширення екосистеми розумних пристроїв: Системи розумних будинків стають відкритішими та гнучкішими, даючи змогу інтегрувати різні пристрої та платформи. Це відкриває можливості для розробки екосистеми розумних пристроїв, де пристрої можуть взаємодіяти між собою і пропонувати нові функціональні можливості.

Безпека і приватність даних: Зі зростанням кількості під'єднаних пристроїв у розумних будинках, безпека і захист даних стають важливими аспектами. Розробники та виробники приділяють велику увагу забезпеченню безпеки та захисту персональних даних мешканців.

Розвиток IoT і хмарних технологій: Інтернет речей (IoT) і хмарні технології відіграють важливу роль у розвитку систем розумних будинків. Використання IoT-сенсорів і пристроїв дає змогу збирати дані з різних аспектів будинку, а хмарні технології забезпечують зберігання, аналіз і обробку цих даних.

Стійкість та енергоефективність: Системи розумних будинків стають дедалі більш орієнтованими на стійкість та енергоефективність. Вони дають змогу

оптимізувати використання ресурсів, керувати енергоспоживанням і впроваджувати поновлювані джерела енергії.

Розширення сфери застосування: Системи розумних будинків не обмежуються тільки житловими будинками. Вони знаходять застосування в комерційних будівлях, громадських установах, готелях та інших типах нерухомості. Це відкриває нові ринки і можливості для розвитку і розширення систем розумних будинків.

Історія систем розумних будинків свідчить про постійний розвиток та інновації в цій галузі. З появою нових технологій, стандартів і можливостей, системи розумних будинків стають дедалі доступнішими, зручнішими та інтелектуальнішими. Вони пропонують широкий спектр функцій і можливостей, які покращують якість життя, забезпечують безпеку та енергоефективність. У майбутньому очікується подальший розвиток та інтеграція систем розумних будинків з іншими технологіями, такими як штучний інтелект, IoT і хмарні обчислення, що відкриє нові горизонти для автоматизації та комфорту в домашньому середовищі.

Найбільш активний період розвитку систем автоматизування будинку та різних пристроїв настав у перше десятиліття 21 століття. Це сталося через кілька основних факторів: розвивається світ технологій дедалі більше, штучний інтелект, бездротові зв'язки, сильне поширення смартфонів, вдосконалення побутових пристроїв.

Етапи розвитку розумних будинків можна розділити на:

- впровадження безпроводного зв'язку;
- впровадження технологій штучного інтелекту;
- використання роботів-помічників.

1.3 Особливості тепличних господарств

Щоб зрозуміти актуальність цієї роботи повною мірою, розберемо, навіщо потрібні теплиці, наскільки багато праці вони вимагають, які рослини примхливі і які вже зараз є та йдуть розробки автоматизованих тепличних угідь.

Більшість знаходиться у південно області. Але цього не вистачає. Усього 20% населення можна забезпечити цією продукцією. Наразі держава фінансує 40 проектів із будови тепличних господарств вартістю близько 41 млрд тенге загалом.

Вирощування рослин у теплицях вважається найкращим рішенням для областей, в яких погодні умови не підходять молодій розсаді або в яких погода мінлива. Захищений від впливу зовнішніх факторів тепличний ґрунт допомагає захистити рослини від заморозків, від посухи, від граду та багатьох різних невідповідних погодних умов.

Будь-якій культурній рослині для нормального росту, розвитку та плодоношення потрібні як мінімум три фактори:

- комфортна, відносипостійна температура;
- достатню кількість світла;
- певна вологість ґрунту та повітря;
- добрива.

На жаль, на відкритому ґрунті, на відміну від теплиць, досить важко збалансувати всі ці умови.

Регулювання мікроклімату допомагає отримати високу продуктивність, а також триваліший вегетаційний період.

Однак теплиці будуть помічником, а не шкідником для рослин тільки якщо дотримуватись багато різних умов, починаючи від побудови на правильному місці і закінчуючи ретельним доглядом за культурами, що вирощуються.

Захищений ґрунт вирішує два основні завдання:

- отримати більш ранній урожай;
- забезпечити комфортні умови для зростання та розвитку теплолюбних

культур.

Незважаючи на численні витрати, як людської праці, так і фінансові, одержання в рази вищого врожаю в більш ранні терміни і триваліший час окупає всі праці.

У будівництві теплиці необхідно ставитися відповідально і ґрунтовно: враховувати сторони світла, ґрунт, рельєф території, ґрунтові води.

Якщо температура піднімається чи опускається більше, ніж потрібно, це може негативно позначитися на висаджених рослинах.[3]

Вологість ґрунту та повітря теж сильно впливає на ріст: Щоб рослина нормально росла і розвивалася, клітини її листя мають бути насичені вологою. Якщо її не вистачає листя перегріваються, в'януть, що веде до зниження швидкості фотосинтезу та збільшення витрати речовин на дихання. різні рослини вимагають різної вологості повітря, наприклад, для огірка до плодоношення необхідно підтримувати відносну вологість лише на рівні 75 – 80 %, коли рослина входить у стадію плодоношення – 80 – 85 %; для помідора оптимальна вологість 60 - 70%; для розсади овочевих культур – 60- 65%.

Від вологості повітря залежить дихання рослин, від якого, своєю чергою, – всмоктувальна здатність листя. Якщо відносна вологість повітря низька, транспірація зростає, втрати вологи посилюються, ростові процеси гальмуються, рослина слабшає. Тому при спекотній погоді необхідний рясний полив та підвищення вологості за допомогою розпилення води в теплиці.

Найбільш ефективно себе показують теплиці, в яких ретельно стежать за всіма показниками росту рослин та навколишнього середовища, щоб вони залишалися оптимальними. Набагато простіше це зробити, якщо система автоматизована, а праця людини зводиться до мінімуму.

1.4 Розвиток автоматизованих систем тепличних господарств

Тепличні господарства відіграють важливу роль у сільському господарстві, забезпечуючи вирощування рослин у контрольованих умовах. З появою нових технологій і розвитком автоматизації, тепличні господарства почали використовувати автоматизовані системи для оптимізації процесів вирощування, контролю навколишнього середовища і підвищення врожайності. У цій статті ми розглянемо розвиток автоматизованих систем тепличних господарств, їхні переваги та перспективи.

Історія розвитку автоматизації в тепличних господарствах. Автоматизація в тепличних господарствах почалася з простих систем, як-от автоматичне поливання та керування освітленням. Однак із розвитком електроніки, датчиків і комп'ютерної технології, виникли більш складні системи управління, які дають змогу контролювати й регулювати умови в теплиці, як-от температуру, вологість, освітлення та CO₂-рівень.

В останні десятиліття автоматизація тепличних господарств досягла нових висот завдяки розвитку інтернету речей (IoT) і штучного інтелекту (ШІ). Тепер тепличні господарства можуть бути повністю автоматизованими, з можливістю віддаленого моніторингу та управління, оптимізації ресурсів і прийняття управлінських рішень на основі даних.

Переваги автоматизованих систем тепличних господарств. Збільшення врожайності: Автоматизовані системи дають змогу оптимізувати умови вирощування рослин, забезпечуючи необхідну температуру, вологість, освітлення та живлення. Це сприяє збільшенню врожайності та якості продукції.

Економія ресурсів. Автоматизовані системи дають змогу ефективно використовувати ресурси, такі як вода, енергія та добрива. Вони можуть автоматично регулювати полив, освітлення і системи вентиляції залежно від поточних умов і потреб рослин, що призводить до зниження витрат на ресурси. Покращення якості продукції: Автоматизовані системи забезпечують стабільні

умови в теплиці, унеможлиблюють вплив несприятливих чинників навколишнього середовища, таких як зміни погоди або шкідники. Це допомагає підвищити якість і однорідність продукції.

Зменшення трудовитрат: Автоматизація дає змогу скоротити необхідність ручної роботи та контролю в тепличних господарствах. Багато процесів, як-от полив, підживлення і контроль параметрів середовища, можуть бути автоматизовані, що звільняє час і ресурси для інших завдань.

Управління на основі даних: Автоматизовані системи збирають і аналізують дані про стан теплиці та рослин. Це дає змогу ухвалювати поінформовані управлінські рішення, оптимізувати процеси вирощування та попереджати про можливі проблеми.

Технології, що використовуються в автоматизованих системах тепличних господарств. Датчики: Датчики використовуються для вимірювання різних параметрів, таких як температура, вологість, освітлення, CO₂-рівень тощо. Вони надають дані для контролю та регулювання умов у теплиці.

Актuatorи: Актuatorи використовуються для управління різними системами в теплиці, такими як системи поливу, освітлення та вентиляції. Вони регулюють параметри відповідно до заданих значень і команд.

Центральні системи управління: Центральні системи управління об'єднують дані від датчиків, керують актуаторами і надають можливість віддаленого моніторингу та управління. Вони також можуть використовувати алгоритми ШІ для аналізу даних і прийняття управлінських рішень.

IoT і хмарні технології: IoT дає змогу пов'язувати всі компоненти автоматизованої системи тепличного господарства в єдину мережу, а хмарні технології забезпечують зберігання та аналіз великих обсягів даних, а також можливість віддаленого доступу та управління.

Виклики та перспективи розвитку автоматизованих систем тепличних господарств. Вартість і доступність: Впровадження автоматизованих систем може бути високим за вартістю, особливо для малих і середніх господарств. Важливо знизити витрати і зробити технології більш доступними для широкого кола

фермерів.

Навчання та підтримка: Впровадження та використання автоматизованих систем вимагає спеціальних знань і навичок. Навчання фермерів і надання технічної підтримки є важливими аспектами успішної реалізації цих систем.

Інтеграція з іншими технологіями: Взаємодія автоматизованих систем з іншими технологіями, такими як системи штучного освітлення, гідропоніка та вертикального фермерства, представляє великі можливості для створення інтегрованих та ефективних систем вирощування.

Інновації та дослідження: Подальші дослідження та інновації в галузі автоматизованих систем тепличних господарств сприятимуть розробці нових технологій, поліпшенню процесів вирощування та підвищенню ефективності систем.

Автоматизовані системи тепличних господарств являють собою важливу технологічну інновацію в сільському господарстві. Вони дають змогу оптимізувати умови вирощування рослин, економити ресурси, покращувати якість продукції та зменшувати трудовитрати. Розвиток автоматизованих систем тепличних господарств триває, і майбутнє обіцяє ще більше інновацій та перспектив у цій галузі. Це представляє великі можливості для підвищення ефективності та стійкості сільського господарства, а також для забезпечення продовольчої безпеки та сталого розвитку.

Зараз йдуть активні розробки розумних теплиць у всьому світі, як у як стартапів (GreenHouse) і перших кроків розробок університетів (томський політехнічний, херсонський державний аграрний університети), так і у вигляді бізнес-проектів, які вже успішно показують затребуваність таких систем на ринку (LG, SmartSunshine, Smart Standart, iFarm).

Такі розумні теплиці допомагають контролювати процес вирощування рослин навіть на відстані за рахунок доступу з мобільного додатка або інтернет-порталу. розглянемо, що пропонують розробники та різні фірми, щоб полегшити трудомісткий процес підтримки необхідного мікроклімату рослин.

У політехнічному університеті ставлять експерименти з поліпшення

теплиць, створюють роботизовані газонокосарки, відчують надійність розумних технологій при вирощуванні рослин і культур, забезпечуючи теплиці автоматичними системами поливу та освітлення, також використовують спеціальні роботизовані системи для запилення рослин та посадки насіння.

У китайської провінції Інтань розумну теплицю підключили до мобільного додатку не тільки для відстеження дій автоматизованих систем, але й для управління ними, наприклад, полив, внесення добрив, контроль температури та вологості повітря. Також смарт теплиця попереджає про наявність шкідників рослин LG Electronics та зайнялася виготовленням розумних теплиць, призначених для використання в квартирі або будинку, воно також здійснює регулювання та контроль температури, освітлення та вологості повітря в приміщенні.

Також вони акцентують увагу на тому, що система ретельно контролює об'єм води, що надходить до рослин, і розподіляє тільки ту кількість, яка потрібна зелені, що дозволяє уникнути неприємних запахів, використовується крапельна система подачі води, а також є примусова вентиляція повітря.

Це може бути хорошим майданчиком для домашнього вирощування овочів та фруктів, без великих зусиль з боку домогосподарки, яка не потребує великих площ землі та власної ділянки.

Унікальність даної теплиці в її конструкції: вона виглядає як міні-холодильник і дозволяє розташовувати розсаду і рослини вертикально, один над одним на полицях. в неї вбудована інтелектуальна система, що відстежує мікроклімат (світло, температуру та вологість) усередині шафи та регулює його під потреби рослин.

У Херсонському державному аграрному університеті розробляють розумну теплицю, всіма процесами в якій буде так само, як і в попередній системі, комп'ютер відстежувати мікроклімат та стан ґрунту всередині теплиці.

Людині залишається часом перевіряти систему та поповнювати запаси добрив, а також збирати врожай.

Smart standard, «розумна теплиця» - також як і розглянуті вже системи

пропонує автоматичний полив рослин, контроль мікроклімату та управління освітленням. дана система надсилає всі дані з датчиків на портал, зміни зберігаються та доступні для аналізу та перегляду.

Система Smart standard «Розумна теплиця» пропонує в комплектації Датчик температури та вологості ґрунту, датчик температури та вологості повітря, Світильник з дистанційним керуванням, Авто-відкривач кватирок з контролером

Конструкції Smart Sunshine, які надають теплиці для цілорічного вирощування фруктових, овочевих, квіткових культур, найбільш повно склали комплектацію на основі високотехнологічного обладнання: крім звичних систем поливу та контролю мікроклімату за допомогою температури та вологості повітря та ґрунту, вони обладнані розподілом та підживленням CO₂, системами туманоутворення та охолодження, системами іригації.

Таким чином, всі існуючі системи, незалежно від методу управління (автоматизований або за допомогою віддаленого доступу з контролем та управлінням), мають у своїй комплектації датчики вологості повітря та ґрунту, датчик температури та систему поливу. У цій дипломній роботі розроблятиметься проект на основі перелічених датчиків.

2 КОНСТРУКТОРСЬКА ЧАСТИНА РОЗРОБКА СИСТЕМИ

2.1 Аналіз варіантів елементів приладу та вибір компонентів

Щоб визначитися з основними характеристиками майбутнього проекту, потрібно проаналізувати методи управління: централізоване та децентралізоване – та методи з'єднання: провідна або бездротова системи –, а також розглянути різні варіанти елементів: які плати, датчики та яке програмне забезпечення краще використовувати.

На рисунках 2.1 і 2.2 представлені схематичне зображення та технологічна схема системи, що розробляється.

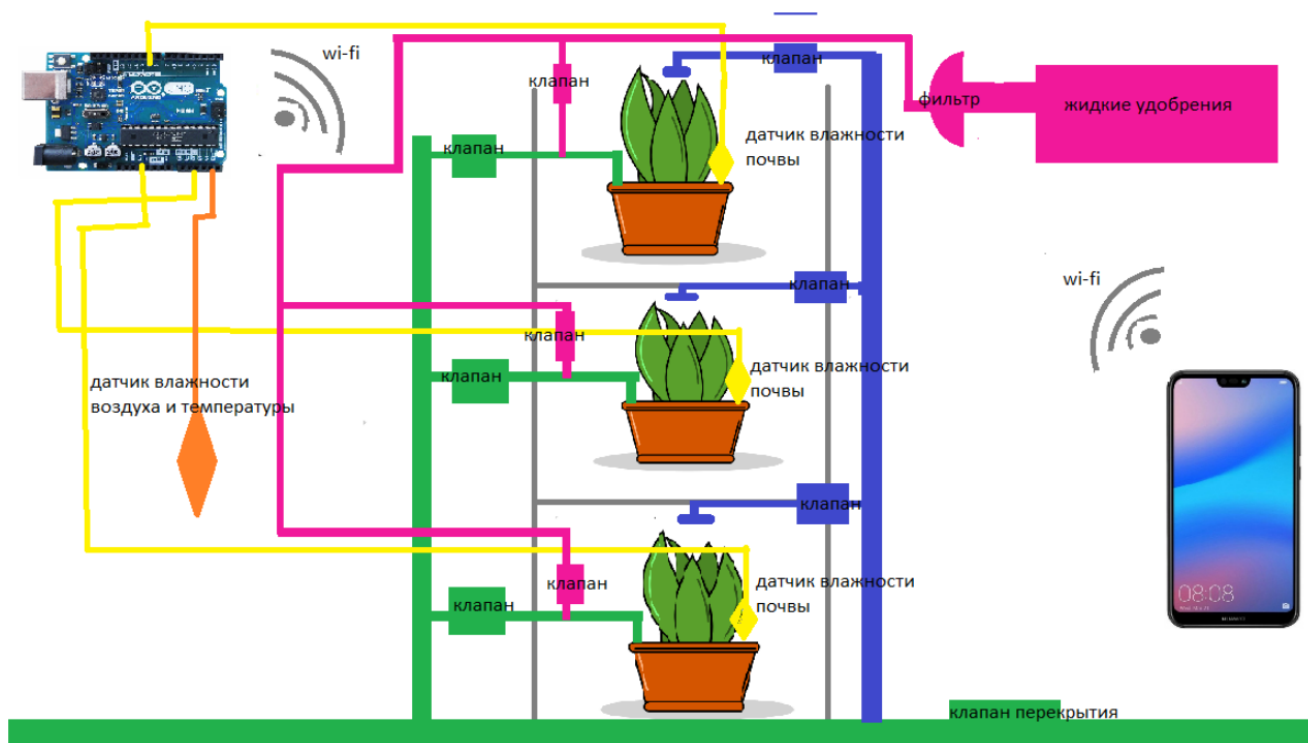


Рисунок 2.1 - Схематичне зображення системи

Цей проект вирішує відразу кілька завдань: веде моніторинг, проводить дистанційне інформування та оптимізує процес формування сприятливого середовища для рослин у тепличних умовах. Для повнішого розуміння роботи системи буде наведено локальний приклад. Ми маємо рослину, яка вимагає догляду. У ґрунт інтегрований датчик вологості ґрунту, який підключений до системи аналізу за допомогою плати Arduino. У реальному часі користувач отримує дані про стан ґрунту. Якщо середовище сприятливе, тобто ґрунт насичений достатньою кількістю вологи, на веб-сервері ми отримуємо дані в діапазоні 0% - 100%, критичні значення визначені коридором 0%-40%. При досягненні критичних показників, оператор отримує сигнал про необхідність провести низку маніпуляцій для повернення до вихідних показників. При допомогою тумблерів-маніпуляторів користувач дистанційно здійснює активацію системи стабілізації, в даному конкретному випадку відкриває клапан для подачі рідини в ґрунт, до необхідного рівня насичення. Досягши оптимальних показників оператор блокує (закриває клапан) систему, що перекриває клапан подачі рідини.

У відповідній графі веб-сервера ми бачимо результат виконаних маніпуляцій: датчик вологості демонструє оптимальні показники, продовжуючи в реальному часі вести мониторинг стану ґрунту.

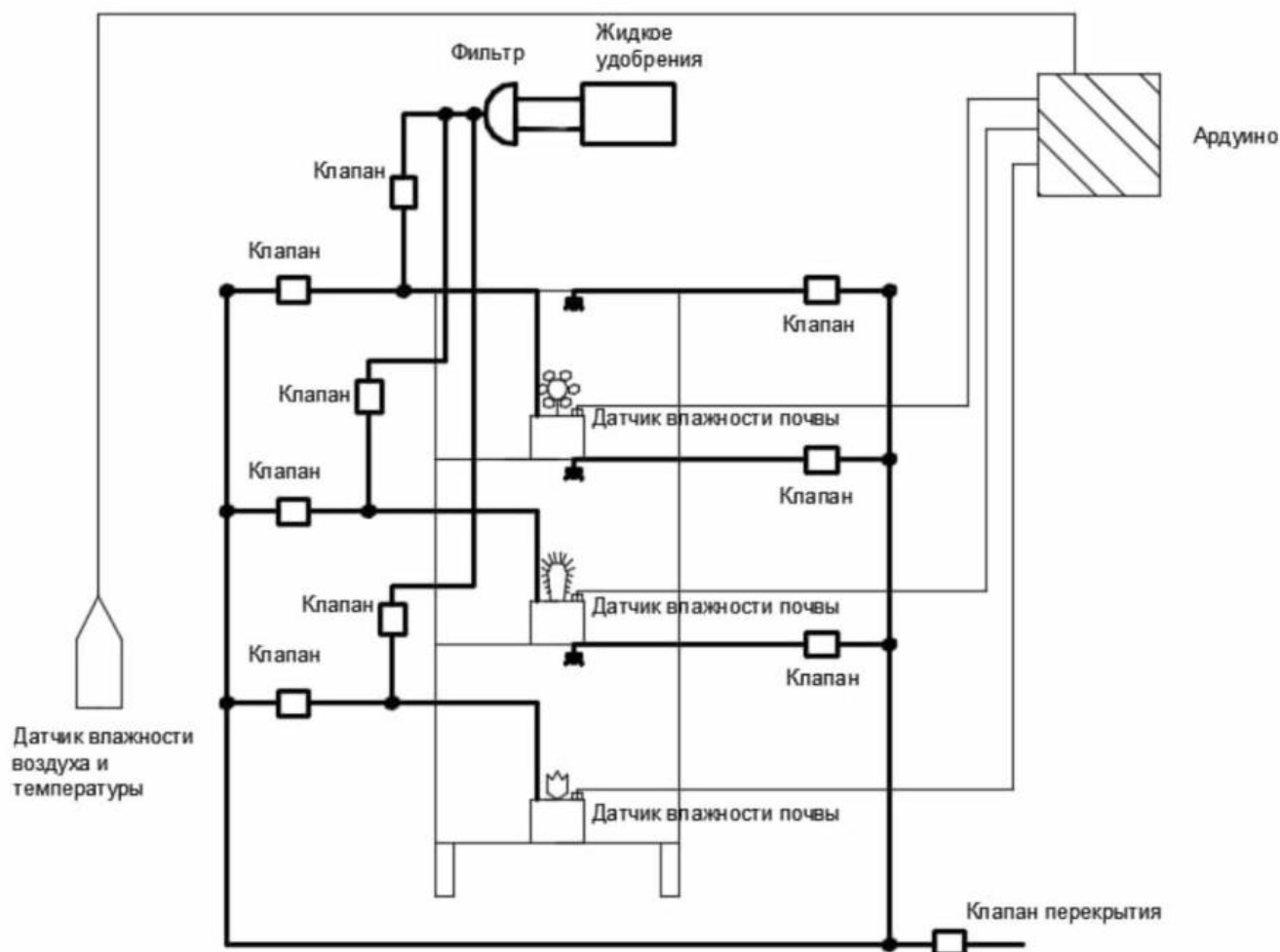


Рисунок 2.2 - Технологічна схема системи

Дротова система будується на тому, що всі пристрої з'єднані єдиною провідною шиною, зазвичай це спеціальні кабелі.

Головні переваги такої системи – це надійність, швидка швидкість відгуку без перебоїв, тривалий термін служби за рахунок відсутності батарейок.

Однак таку систему в будинку потрібно заздалегідь планувати та проектувати. Потрібно багато проводки і потрібно всі керовані пристрої підключати до щита. У результаті районі щита утворюється досить великий пучок проводів. Потрібний щит досить великих розмірів (ширина близько 60 см і висота

від 80 см до 150 см, залежно від розміру автоматизованого об'єкта) Встановлюється така система може тільки на початку ремонту, доки не зроблено основну електропроводку за класичною схемою. У готовому ремонті зробити провідний розумний будинок якщо і вийде, то або доведеться підлаштовуватися під умови, або виглядатиме будинок небезпечно і неакуратно.

Бездротові системи скорочують час на встановлення системи та її монтаж, такі системи можна використовувати не змінюючи відвідування, це і є головною їхньою перевагою. Найчастіше такі системи використовуються на батареях, що ще більше спрощує процес розташування системи.

Один із недоліків і впливає з цього – потрібно стежити за терміном батарейок та вчасно їх міняти. Ще один важливий недолік - можливість нестабільного сигналу зв'язку.

Суть таких систем у тому, що керування всіма пристроями, приладами і датчиками походить з одного контролера. Варто пам'ятати, що доповнити подібні системи буде складніше, оскільки контролер програмується для всіх пристроїв заздалегідь. Велика перевага централізованих систем – можливість керування всім з одного інтерфейсу, наприклад, з телефону. У такому комплексі велика відповідальність лежить на програмісті, який програмує головний контролер, та проблема цих систем у людському факторі. Головний недолік установок з головним контролером у тому, що цей контролер часто єдиний і від нього залежить робота всіх пристроїв, тому якщо з ним щось станеться, вся система зупиниться.

У децентралізованих систем кожен пристрій досить автономний і має свій окремий контролер або мікропроцесор у збиранні. Величезна перевага таких систем - надійність: у разі виходу з ладу одного контролера всі інші прилади можуть продовжувати справно працювати.

Використовуючи децентралізовані системи, є великий ризик: кількість пристроїв у щиті досить велика, тому при виборі сумнівного виробника (а такі є, тому що всі покладають на ринок розумних будинків великі надії) ви ризикуєте зіткнутися з виходом з ладу того чи іншого пристрою, який потрібно замінити .

Таким чином, щоб у результаті проект, який розробляється в цій роботі, можна було пристосувати до будь-якого приміщення та керувати зі смартфона, можна використовувати як провідну, так і бездротову систему, але централізована провідна система автоматизації набагато простіша у розробці.

В цій дипломній роботі розроблятиметься проект на основі централізованих провідних систем.

2.2 Обґрунтування апаратної частини

Arduino і Raspberry pi - це маленькі плати, проте дуже відрізняються. Arduino - проста мікропроцесорна плата, яку можна збирати як конструктор, з'єднуючи додаткові шлейфи і плати, також не вимагає довгого попереднього налаштування (наприклад, операційної системи), в той час як Raspberry pi - повноцінний мінікомп'ютер, для якого потрібна операційна система, він також оснащений різними інтерфейсами для підключення пристроїв Raspberry pi спочатку оснащений usb для підключення клавіатури, мишки, є порт Ethernet, HDMI. Перевага Arduino перед Raspberry pi у її простоті, а також відносної дешевизни. Arduino не чекає запуску операційної системи, завдяки цьому швидше, ніж Raspberry pi. Arduino простіше мову програмування, не потрібний зовнішній програматор. Arduino також більш стійкий до відмови. Arduino споживає менше енергії, ніж Raspberry pi, також може працювати на батарейках або сонячних панелях. Струм на виході у Arduino 40 ма, Raspberry pi не більше 16 ма. Arduino працює на напрузі 5в, Raspberry pi на 3, 5, також часто на платах вбудований стабілізатор напруги. В Arduino упо вбудований запобіжник, що відновлюється, що захищає юсб порт пк від коротких замикань і перевантажень. Незважаючи на те, що у багатьох пк вже стоїть запобіжник, додатковий забезпечує найкращий рівень захисту. Якщо від Usb споживається струм більше 500 мА, запобіжник автоматично розірве з'єднання до усунення причин коротких замикань або

перевантажень. Також зручно, що Arduino можна збирати так, як вам треба, підключаючи плати розширення (шилди). Raspberry pi, звичайно, не настільки потужний, як сучасні комп'ютери, але він чудово підходить для вивчення комп'ютера та основ роботи з ним, для вирішення багатьох практичних завдань, для інтернет-серфінгу, а також для програвання відео та прослуховування аудіофайлів. [5,6]

Датчики бувають аналогові та цифрові. Аналогові сигнали можуть бути спотворені перешкодами, тоді як у цифровому сигналі є перешкоди, він не буде. Таким чином, цифрові сигнали більш точні: якщо сигнал є, то він приходить без спотворень.

Bluetooth- в даній роботі завдання створити дистанційне керування, яке не прив'язане до відстані між телефоном і контролером, тому таке з'єднання не підходить.

Також є варіанти, яку плату підключити до Arduino для роботи Wi-Fi. тут два варіанти: w5100 та esp8266. Тут буде використовуватися w5100, це простіше, ніж esp8266, тому що не потребує додаткових налаштувань wi-fi мережі[7].

2.3 Обґрунтування програмної частини

Щоб працював додаток на телефоні та з нього можна було б керувати контролером, є кілька варіантів:

- створити програму з нуля на Java;
- створити через графічний редактор, який може підключитись до Arduino, додаток (Blynk, RemoteXY).

Розберемо особливості кожного пункту, щоб вибрати найкращий варіант.

RemoteXY пристосований спеціально для створення інтерфейсів для керування зі смартфона різними контролерами. Дуже зручно те, що програма підключається до хмарного сервісу, тому доступ до пристрою буде, якщо є доступ

до інтернету. Однак зараз у цього графічного редактора з'єднання з хмарним сервісом не завжди і можливі перебої. Також у цьому додатку для підключення до контролера клієнту буде складно розібратися, до того ж програма обмежена безкоштовне користування.

Blynk - це подібний сервіс, і на відміну від попереднього є qr код, який допоможе клієнту, не знаючи технічних деталей, користуватися додатком, також, як і в попереднього сервера, дуже обмежене безкоштовне користування.

Java - це мова програмування високого рівня, і, звичайно, щоб програма в результаті працювала коректно, потрібно вивчити мову та проводити тестування після написання програми.

У зв'язку з тим, що розглянуті варіанти або займають дуже багато часу та сил, або дорогі в обслуговуванні, або непостійні та незручні клієнту, було вирішено у проекті дипломної роботи використовувати телеграм бот. До того ж це зменшує ризик того, що людина не зможе встановити «зайву» програму в телефоні, або що програма буде некоректно працювати. Але для того, щоб створити підключення, потрібен протокол https, який не підтримується w5100, якщо для цього використовувати esp8266, то виникає інша проблема - відсутність офіційних бібліотек телеграм бота для arduino ide, а знайти коректну бібліотеку, яка працювала, дуже складно. Таким чином, у цій дипломній роботі буде створюватися локальне підключення по мережі та управління через html сайт реле, підключених до клапанів, і там будуть виводитися показання з датчиків.

Користувач сучасного ПК не замислюється про те, як працюють окремі частини ПК, запускає потрібні програми і працює з ними, тому одним із завдань даного проекту було створити просту в управлінні та установці для користувача систему. У проекті будуть використані елементи: мікроконтролер Arduino mega, wi-fi shield w5100, що підключається до мікроконтролера, цифрові датчики вологості повітря та температури dht23, вологості ґрунту y169, веб-сторінка для підключення до системи.

2.4 Технічні характеристики обраних елементів

Апаратна платформа Arduino Mega базується на мікроконтролер ATmega2560 (технічні специфікації представлені далі). Плата оснащена 16 аналоговими входами, 4 послідовними UART портами, 54 цифровими входами/виходами. Крім іншого, в платформу інтегровані: кнопка перезавантаження, USB-конектор, генератор 16 МГц (кварцовий), а також роз'єми: живлення та ICPS. Живлення на платформу Arduino Mega подається за допомогою USB-кабелю, акумуляторної батареї або адаптера AC/DC.

Примітка. Arduino Mega сумісна з платами розширення Duemilanove та Uno.

Особливості аналогових/цифрових входів та виходів апаратної платформи Arduino Mega. Arduino Mega оснащений 54 цифровими висновками. Останні можна налаштовувати як вихід або вхід, використовуючи функції `pinMode()`, `digitalWrite()`, `digitalRead()`. Кожен висновок передбачає наявність резистора навантаження, відключеного в початковому стані 20-50 кОм (пропускна здатність до 40 мА). Усі висновки працюють при напрузі 5В.

Arduino Mega передбачає додаткову пару висновків платформи:

Reset – перезавантаження мікроконтролера при низькому рівні сигналу на виході. Використовується для підключення кнопки перезавантаження на платі розширення закриваючи доступ до кнопки безпосередньо на платі Arduino Mega.

AREF – Необхідний для обслуговування функції `analogReference()` (Опорна напруга для аналогових висновків).

Ряд висновків передбачає набір особливих функцій:

LED 13 - апаратна платформа Arduino Mega оснащена вбудованим світлодіодом, підключеним до 13 виведення №13. Світлодіод спалахує в тому випадку, якщо значення на виході має високий потенціал.

I2C SDA (20) та SCL (21) – забезпечує зв'язок I2C (TWI) за допомогою бібліотеки Wire. Примітно, що розташування висновків на апаратній платформі

Arduino Mega не відповідає розташуванню Diecimila та Duemilanove.

PWM (2-13 / 44-46) – робота будь-якого виведення забезпечується ШІМ (роздільна здатність 8 біт) за допомогою функції `analogWrite()`.

16 аналогових висновків апаратної платформи Arduino Mega2560 можуть набувати 1024 різних значення, тобто мають роздільну здатність 10 біт. З основні висновки розраховані діапазон виміру до 5 В (щодо землі). Крім іншого, передбачена можливість коригування верхнього межі. Ця операція здійснюється за допомогою функції `analogReference()` та виведення AREF.

Пам'ять апаратної платформи Arduino Mega

На борту встановлено флеш-пам'ять (256 кБ) для зберігання коду програми. На зберігання даних завантажувача відведено лише 4 кБ. Оперативної пам'яті – 8 кБ та ще 4 кБ – EEPROM (читання та запис проводиться за допомогою бібліотеки EEPROM).

Живлення апаратної платформи Arduino Mega

Живлення апаратної платформи Arduino Mega виробляється двома способами. Користувач може використовувати як зовнішнє джерело живлення (вибір проводиться автоматично), так і здійснити подачу напруги за допомогою USB.

Подачу зовнішнього живлення слід проводити через акумулятор або батарею (перетворювач напруги AC/DC). Підключення здійснюється за допомогою роз'єму 2.1 мм (на центральному контакті – позитивний плюс). Проводи від акумулятора слід з'єднувати з висновками Vin і Gnd (роз'єм живлення POWER).

Примітно, що апаратна платформа Arduino Mega2560 не передбачає використання мікроконтролера FTDI USB (використовувався в попередніх версіях). Обмін даних по USB здійснюється мікроконтролером Atmega8U2 (слід програмувати як USB-to-serial конвертер).

Роботу апаратної платформи Arduino Mega слід вибудовувати в діапазоні зовнішнього живлення 6-20 В. Наприклад, при напрузі живлення нижче 7 В, виведення 5V буде здатне видати менше 5 В (за таких значень стабільна робота

платформи не гарантується). Альтернативні характеристики понад 12 В можуть навпаки призвести до перегріву регулятора напруги та завдати критичних ушкоджень платі. Рекомендований діапазон - 7-12 Ст.

Апаратна платформа Arduino Mega передбачає такі види живлення: 5V, VIN, 3V3, GND.

GND – виведення заземлення.

3V3 - забезпечує напередження на виведенні 3.3 У платформі, що генерується мікросхемою FTDI (максимальне споживання струму 50 мА).

VIN – вихід необхідний для живлення від зовнішнього джерела. Слід використовувати за відсутності 5В від роз'єму USB (або відмінного джерела живлення). Через цей висновок проводиться подача напруги живлення Примітно, що через цей вхід можна запитати, якщо живлення подається на роз'єм 2.1mm.

5V є регульованим джерелом напруги. Використовується для живлення компонентів на платі та самого мікроконтролера. Живлення можна подавати від роз'єму USB через регулятор напруги, або іншого (регульованого) джерела напруги 5 В.

Передача даних. Зв'язок з комп'ютером апаратної плати Arduino Mega проводиться за допомогою ряду інтегрованих пристроїв або мікроконтролерів. Наприклад, ATmega2560 підтримує відразу 4 порти, призначені для послідовної передачі даних UART для TTL. Мікросхема ATmega8U2 (встановлена на платформі) здатна надати віртуальний COM порт програм на комп'ютері, направляючи один з інтерфейсів через USB.

Serial Monitor дозволяє відстежувати спеціальну утиліту моніторингу з середовища Arduino. Вона ж забезпечує передачу та прийом текстових даних при підключенні до платформи. Відстежувати роботу системи допомагають світлодіоди RX та TX на платформі. Останні будуть блимати під час передачі даних через мікросхему ATmega8U2 та USB підключення. Додам, що ATmega2560 синхронізується з інтерфейсами TWI (I2C) та SPI. Для зручності використання шини I2C Arduino передбачена бібліотека Wire.

Програмування апаратної платформи Arduino Mega Програмування

апаратної платформи Arduino Mega провадиться за допомогою середовища розробки Arduino. Всі зміни вносяться мовою програмування, що базується на C/C++ - це робить Arduino одним із найпростіших пристроїв у освоєнні на даний момент.

Примітно, що полегшує запис нових програм без використання зовнішніх програматорів мікроконтролерів ATmega2560 із встановленим завантажувачем. Для обмеження зв'язку використовується оригінальний протокол STK500.

У випадку, якщо у користувача немає необхідності використовувати Завантажувач, система представляє можливість запрограмувати мікроконтролер за допомогою внутрішньосхемного програмування (висновки блоку ICSP). Тим часом контролер ATmega8U2 має власний завантажувач (DFU). Активація останнього проводиться шляхом замикання джампера поруч картою Італії (зворотний бік) та функцією reboot-контролера. Для завантаження нової прошивки слід використовувати утиліти DFU (Mac OS X/Linux) або Atmel's FLIP (Windows). Альтернативним варіантом прошивки є використання зовнішнього програматора через вход ISP.

Сумісність апаратної платформи Arduino Mega та фізичні характеристики. Апаратна платформа Arduino Mega сумісна з будь-якими платами розширення сімейства Uno, Diecimila та Duemilanove. Практично всі висновки (0-13, включаючи примикаючі GND і AREF, блок ICPS, аналогові входи 0-5, блоку ICSP, порту послідовної передачі UART (висновки 0 і 1) зовнішнього переривання 0 і 1 (висновки 2 і 3) розташовані аналогічно платам сімейства Diecimila, Duemilanove та Uno. SPI-зв'язок провадиться за допомогою блоку ICSP (аналогічно платформ Mega2560, Duemilanove, Diecimila).

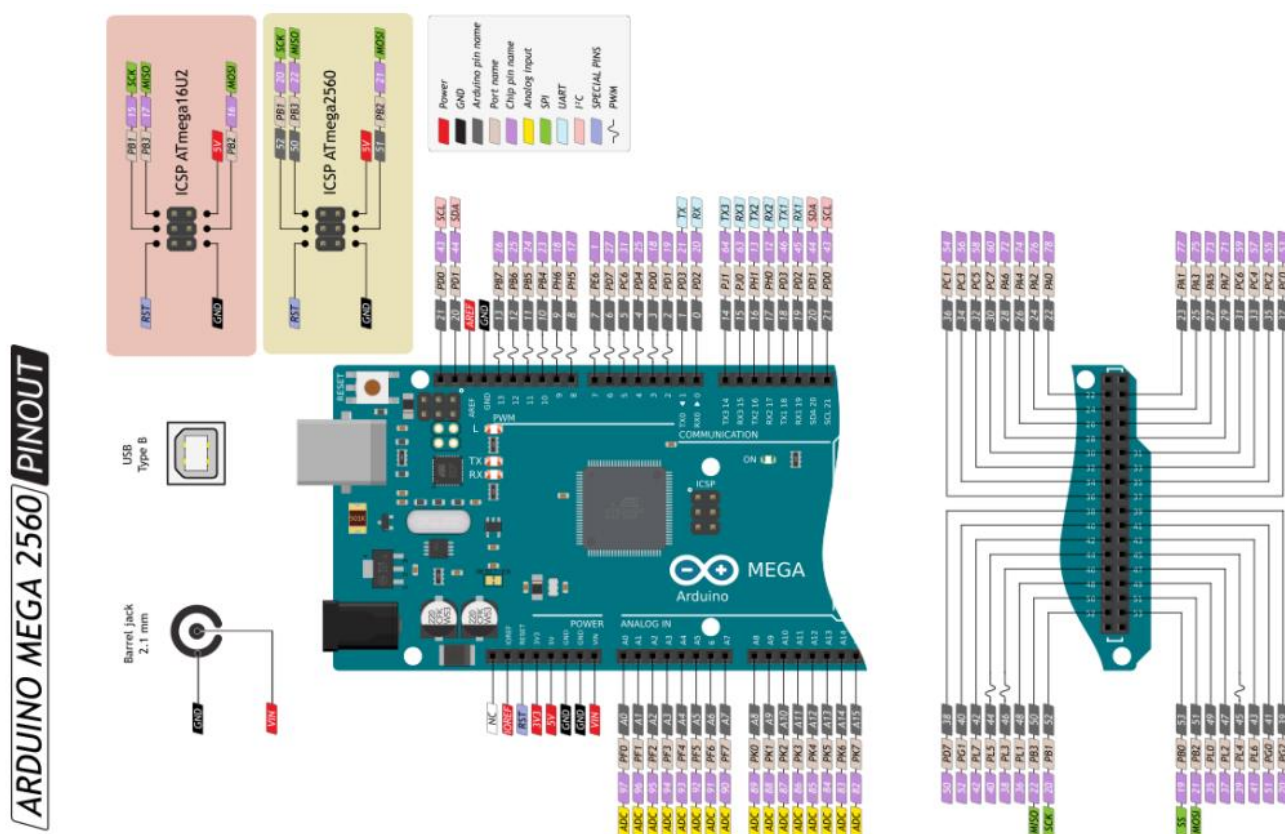
Винятком є розташування висновків (20, 21) зв'язку I2C на платформі Mega – на платформах Duemilanove та Diecimila аналогові виходи 4 та 5 локалізовані інакше.

Друкована плата Mega2560 має такі розміри: довжина 10,2, ширина 5.3 см. За межі зазначених параметрів виходять силовий роз'єм та роз'єм USB. Примітно, що апаратна платформа має три заздалегідь підготовлені отвори, що дозволяють

закріпити плату на потрібній поверхні. Практично всі висновки мають інтервал 0,25 см. Винятки становлять цифрові висновки, що дозволяє їх легко ідентифікувати – висновки 7 та 8 (0,4 см).

Переваги роз'єму USB в Arduino Mega2560. Використання роз'єму USB в Arduino Mega2560 передбачає наявність струмового захисту. У випадку конкретної апаратної платформи використовується плавка вставка, що захищає комп'ютерний USB порт від надструмів і струмів короткого замикання. Запобіжник виконує роль допоміжного (додатковий бар'єр), тому практично всі сучасні комп'ютери мають альтернативні (або аналогічні) системи захисту. Обмін даних переривається запобіжником у момент проходження струму понад 500 мА через USB.

На рисунку 2.3 представлена схема висновків, яку використовують при програмуванні та з'єднанні.



Модуль складається з двох частин: контактного щупа YL-69 та датчика YL-38, в комплекті йдуть дроти для підключення. Між двома електродами щупа YL-69 створюється невелика напруга. Якщо ґрунт сухий, опір великий і струм буде меншим. Якщо земля волога - опір менший, струм - трохи більше. За підсумковим аналоговим сигналом можна судити про рівень вологості. Щуп YL-69 з'єднаний з датчиком YL-38 по двох дротах.

У комплекті йде плата, що перетворює аналоговий сигнал датчика на цифровий сигнал (нуль – одиниця), побудована на основі мікросхеми LM393 і потенціометра, яким Управління датчиком здійснюється або від Arduino контролера, або від іншого керуючого мікропроцесорного пристрою за допомогою спеціальних програм. На платі розташований змінний резистор, який використовується для налаштування порога спрацьовування датчика (чутливості).

Датчик вологості ґрунту має два інтерфейси для підключення до живлення та мікроконтролера, для підключення чутливого елемента:

- для підключення до живлення та мікроконтролера використовується 4-контактний штирьовий інтерфейс. Позначення контактів: VCC – напруга живлення; GND - загальний контакт; D0 – цифровий вихід; A0 – аналоговий вихід;
- для підключення чутливого елемента використовується 2-контактний штирьовий інтерфейс.

Живлення датчика здійснюється або від Arduino контролера, або іншого керуючого мікропроцесорного пристрою, або зовнішнього джерела живлення (блоку живлення, батареї). Напруга живлення датчика 3,3- 5В.

Характеристики:

- вимірювальний елемент: металізований щуп (окремий);
- виконання датчика: на мікросхемі LM393 з налаштуванням чутливості;
- виходи датчика: цифровий та аналоговий;
- напруга живлення датчика: 3,3 - 5В;
- габарити модуля: 38 x 15,6 x 8 мм;

- габарити датчика: 63,8 x 20 x 8 мм;
- довжина кабелю: 20 см;
- вага комплекту: 11 г.

Крім контактів з'єднання зі щупом, датчик YL-38 має чотири контакти для підключення до контролера:

- Vcc – живлення датчика;
- GND – земля;
- A0 – аналогове значення;
- D0 – цифрове значення рівня вологості.

Датчик YL-38 має два світлодіоди, що сигналізують про наявність вступника на датчик живлення та рівня цифрового сигналу на виході D0. Наявність цифрового виведення D0 та світлодіода рівня D0 дозволяє використовувати модуль автономно, без підключення до контролера. На рисунку 2.4 представлено зображення датчика YL-69. [8,9]

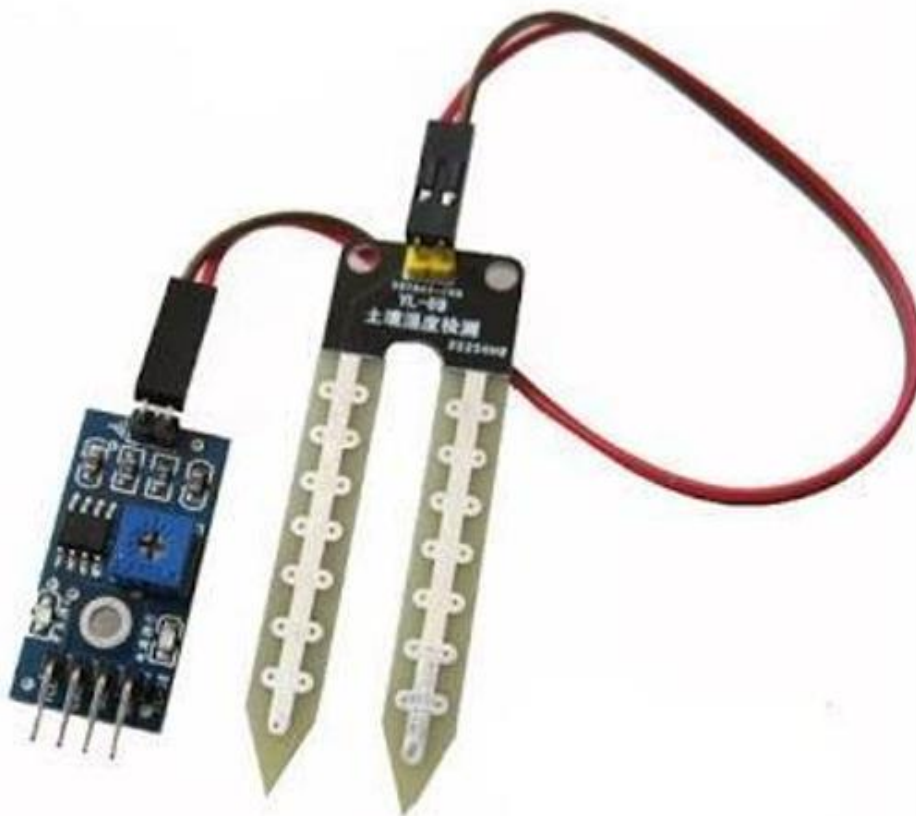


Рисунок 2.4 – датчик вологості ґрунту YL-69 2.4.3 Ethernet Shild W5100

Найбільш популярні Ethernet модулі для Ардуїно сьогодні це мікросхеми на основі wiznet w5100, яка здатна підтримувати обмін даними з постійною швидкістю в 100 Мбіт/сек.

Практична точка зору робота з Ethernet виглядає як можливість з'єднати певним чином кінцеве обладнання з найближчою точкою, що має вихід в інші сегменти мережі. Найчастіше це роутер або маршрутизатор з доступом до інтернету або до інших ресурсів локальної мережі. На рисунку 2.5 представлено зображення підключеного shield до плати.

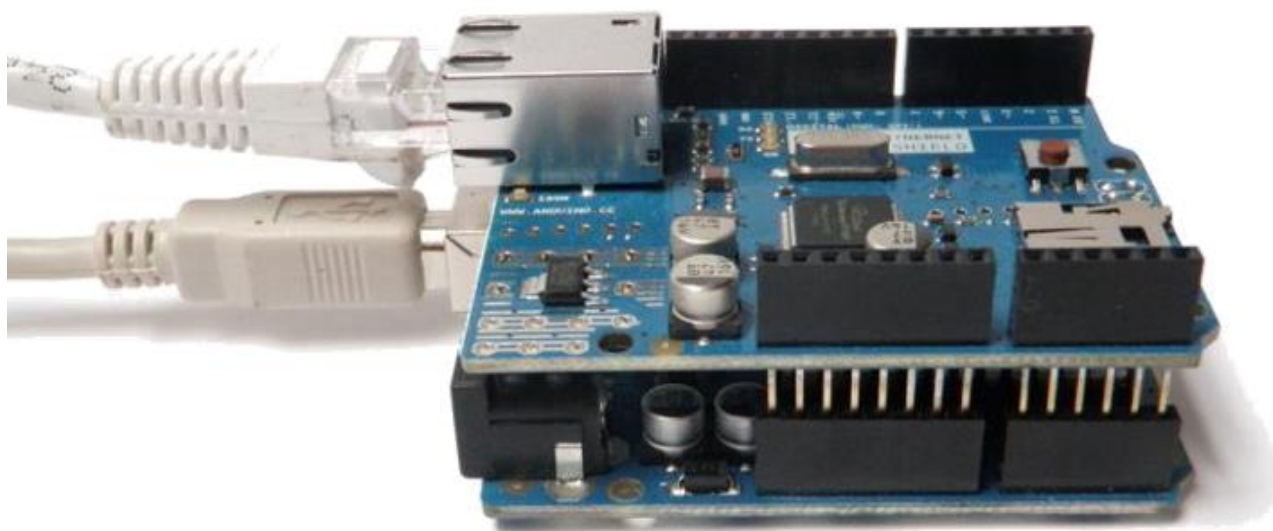


Рисунок 2.5 – підключення shield w5100 до плати

Ключові характеристики модулів на базі W5100:

- робоча напруга - 5 Вольт, підходить живлення з плати Arduino;
 - внутрішній буфер 16 Кб;
 - швидкість з'єднання до 10/100 Мбіт/сек;
 - зв'язок із платою ардуїно здійснюється за допомогою порту SPI; д)
- W5100 підтримує TCP та UDP.[7]

На рисунку 2.6 представлено зображення Shield.



Рисунок 2.6 – shield W5100

Електромагнітний клапан - пристрій, який призначений для керування та регулювання потоків усіх типів рідин та газів. Він складається з корпусу, соленоїда (електромагніту) із сердечником, на якому встановлений диск або поршень, що регулює потік.

На електромагнітну котушку клапана подається електрична напруга, після чого магнітний сердечник втягується в соленоїд, що призводить до відкриття або закриття клапана. Сердечник поміщений усередині закритої трубки котушки соленоїда - це необхідно для герметичності електромагнітного клапана.

На рисунку 2.7 представлено зображення клапана.



Рисунок 2.7 – електронно-магнітний соленоїдний клапан

Датчик вологості повітря та температури DHT22

Загальні показники датчиків.

DHT11:

- дуже низька вартість;
- харчування та I/O: 3–5 В;
- визначення вологості 20-80% з точністю 5%;
- визначення температури 0–50 °C із точністю 2 %;
- частота опитування не більше 1 Гц (не більше одного разу на 1 сек.);
- розміри 15,5 12 5,5 мм.

DHT22:

- низька вартість;
- харчування та I/O: 3–5 В;
- визначення вологості 0-100% з точністю 2-5%;
- визначення температури –40...+125 °C з точністю $\pm 0,5$ °C;

– частота опитування не більше 0,5 Гц (не більше одного разу на 2 сек); е) розміри 15,1 25 7,7 мм.

Сенсор DHT22 має кращі, ніж у DHT11 характеристики, але більш високу вартість, в даному проекті буде використано DHT22.

На рисунку 2.8 представлено зображення датчика DHT22.

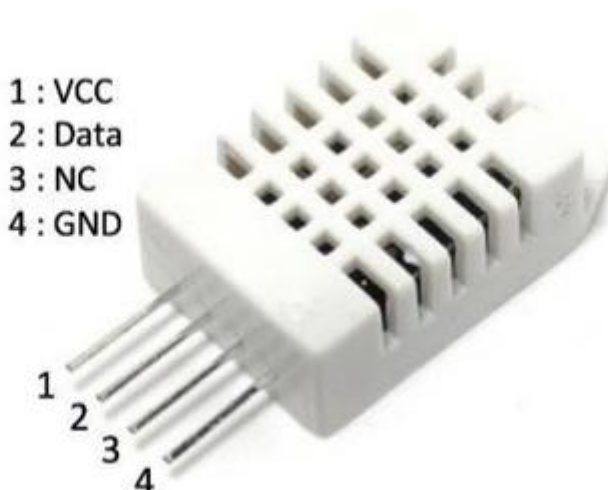


Рисунок 2.8 – Датчик вологості повітря DHT22

Даний модуль реле високого рівня призначений для керування струмом як постійним, так і змінним. Крім самого реле, модуль містить ще й оптоелектронну розв'язку з транзистором, які захищають висновки Ардуїно від стрибків напруги на котушці. На рисунку 2.9 представлено зображення реле. [10]



Рисунок 2.9 – Реле

Для повного розуміння у майбутньому куди підключати, створимо монтажну схему (представлена рисунку 2.10).[11]

На схемі позначення: ДВ1 – датчик вологості ґрунту 1, ДВ2 – датчик вологості ґрунту 2, ДВ3 – датчик вологості ґрунту 3, ДНТ – датчик температури та вологості ґрунту DHT22, к1 – клапан 1, к2 – клапан 2, к3 – клапан 3, к4 – клапан 4, к5 – клапан 5, к6 – клапан 6, р1 – реле 1, р2 – реле 2, р3 – реле 3, р4 – реле 4, р5 – реле 5, р6 – реле 6.

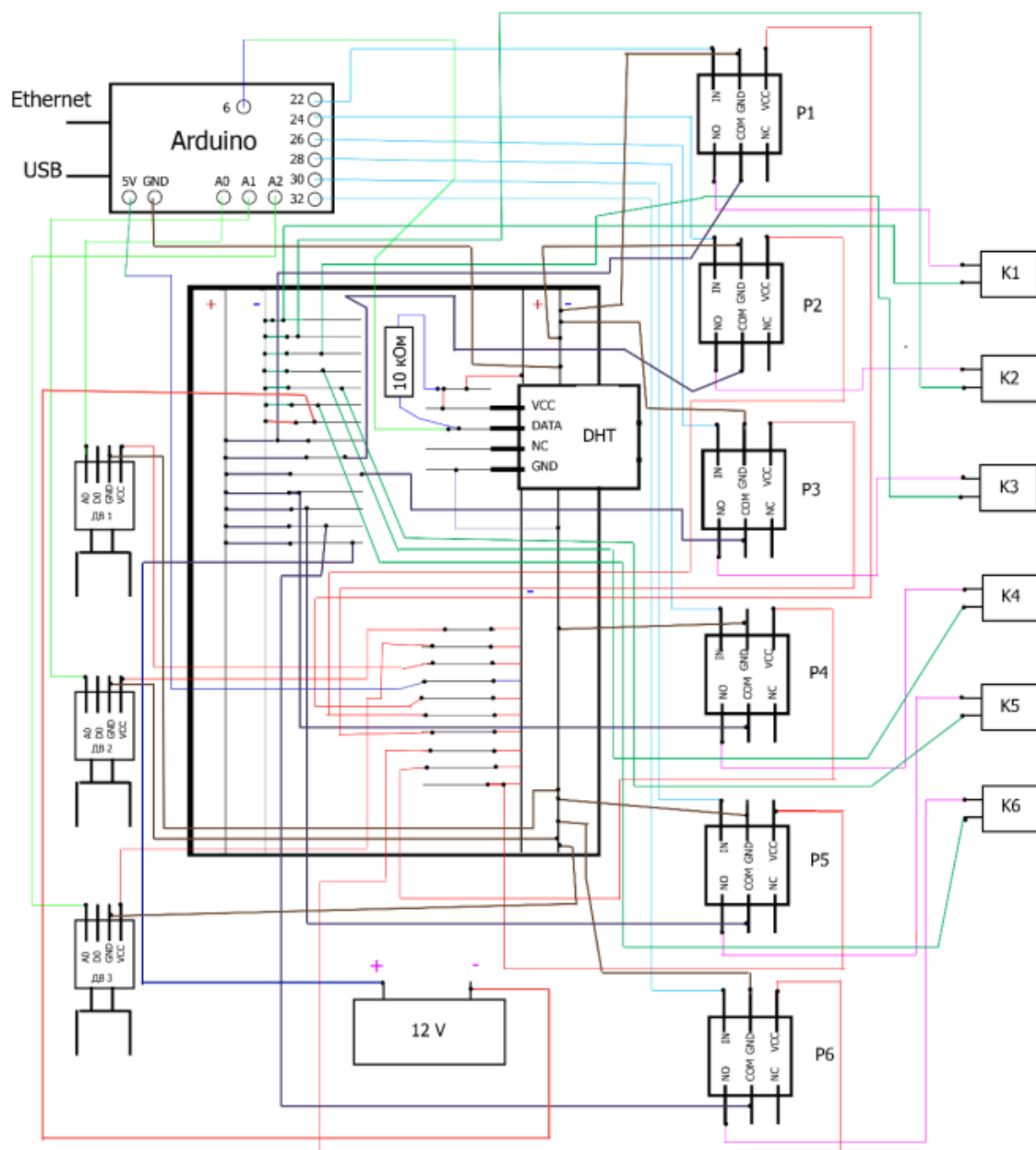


Рисунок 2.10 – Монтажная схема

3 МОДЕЛЮВАННЯ РАЦІОНАЛЬНОГО МОДЕЛЬНОГО СКЛАДУ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ

3.1 Побудова моделі системи захисту інформації

Ця програма варта конструювання схеми. Вона підтримує дві мови програмування FBD та LAD, створювати проект можна на будь-якому з них.

У цій роботі використовуватиметься FBD. Спочатку потрібно вибрати контролер (Arduino Mega).

Робоче вікно програми FLProg мовою FBD складається з кількох полів: основне меню програми, дерево проекту, дерево встановленого обладнання (у ньому представлене обладнання, яке використовується в проекті, новий проект в ньому присутні тільки входи та виходи контролера), бібліотека блоків (у ній знаходиться обладнання, яке можна застосувати у проекті). Схема в FLProg є набір плат з обладнанням.[12] На рисунку 3.1 представлений скріншот із програми, на якому видно додані входи.

Arduino Mega є однією з платформ мікроконтролера Arduino, призначеною для розроблення та прототипування різноманітних електронних проектів. Вона є розширеною версією плати Arduino Uno і пропонує вищу продуктивність та більшу кількість входів/виходів.

Ось основні характеристики та особливості Arduino Mega:

- мікроконтролер: Arduino Mega заснована на мікроконтролері ATmega2560. Це 8-бітний мікроконтролер із тактовою частотою 16 МГц і 256 кілобайтами флеш-пам'яті. Він забезпечує досить високу продуктивність для виконання складних завдань і керування великою кількістю периферійних пристроїв;

- входи/виходи: Arduino Mega має 54 цифрові входи/виходи, з яких 15 може використовуватися як виходи з можливістю генерації ШІМ-сигналу. Крім того, вона має 16 аналогових входів для підключення датчиків та інших аналогових пристроїв;

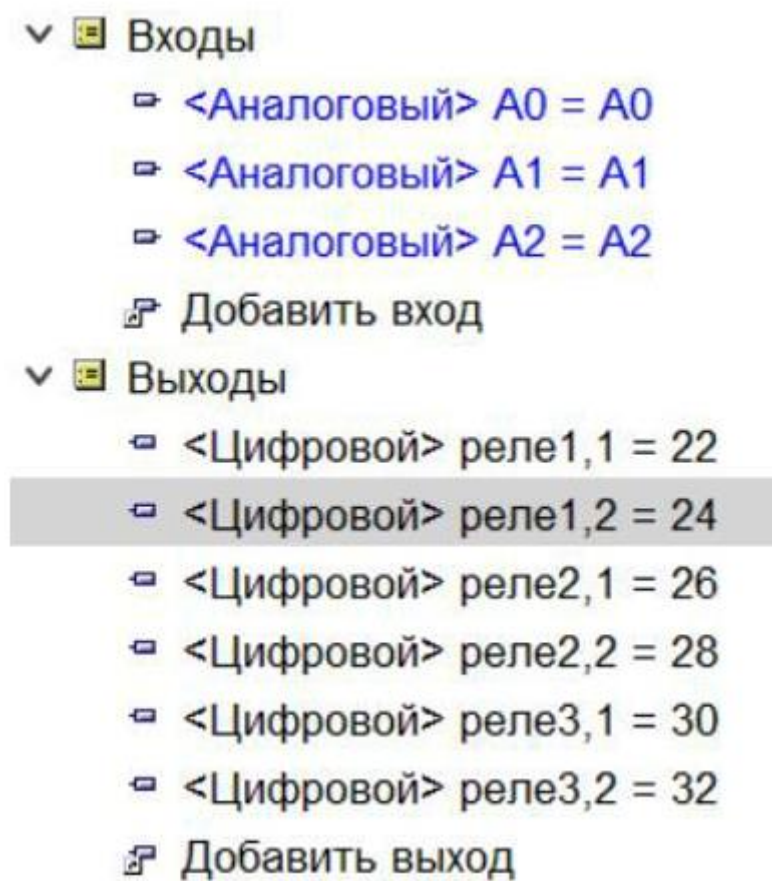


Рисунок 3.1 – цифрові та аналогові входи

Для підключення реле створюємо цифрові виходи.

Масштаб необхідний перетворення значень 1023-0 в 0-100% (скриншот цієї частини програми представлений рисунком 3.2).

Для роботи з веб-сторінкою необхідно знати IP адресу W5100, його можна дізнатися за допомогою базового прикладу програми в Arduino IDE. У разі цього проекту IP 192.168.0.115 його потрібно внести на адресу сторінки. Частина програми з адресою представлена рисунку 3.3.

Для створення веб-сторінки потрібно в тіло сторінки вставити код HTML (рисунок 3.4).

При натисканні на кнопку сторінка переходить на /1 і тд залежно від номера кнопки (рисунок 3.5).

Для того щоб при натисканні на кнопку перемикання реле (відкриття/закриття клапана) сторінка поверталася на вихідну, потрібно прописати в тілі сторінки «<meta http-equiv="refresh" content="0;URL=http://192.168.0.115">».

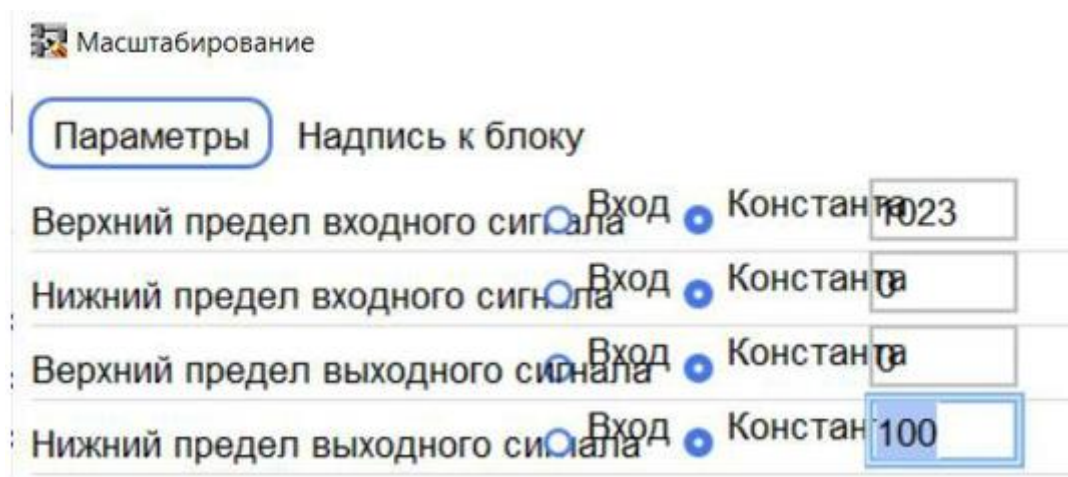


Рисунок 3.2 – Перетворення значень у програмі за допомогою блока scale

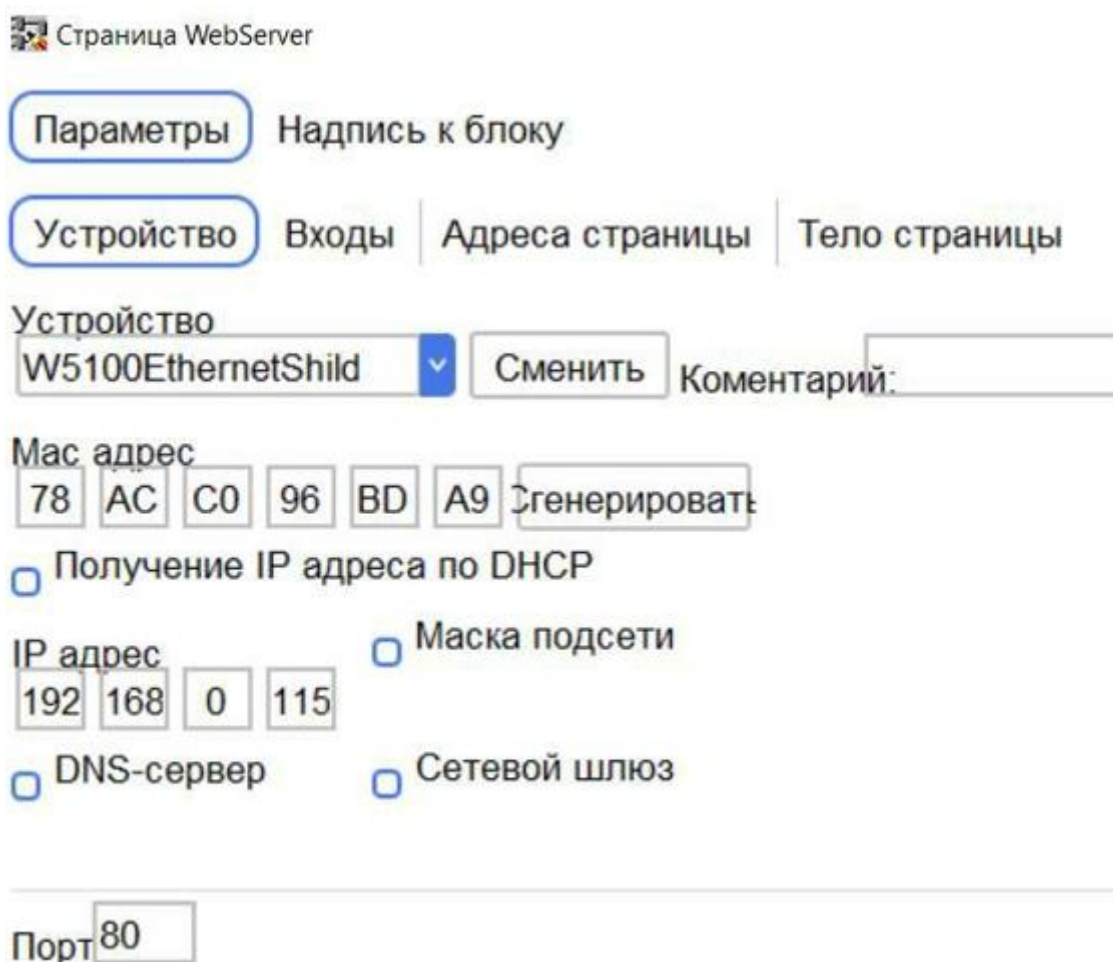


Рисунок 3.3 – Скріншот програми

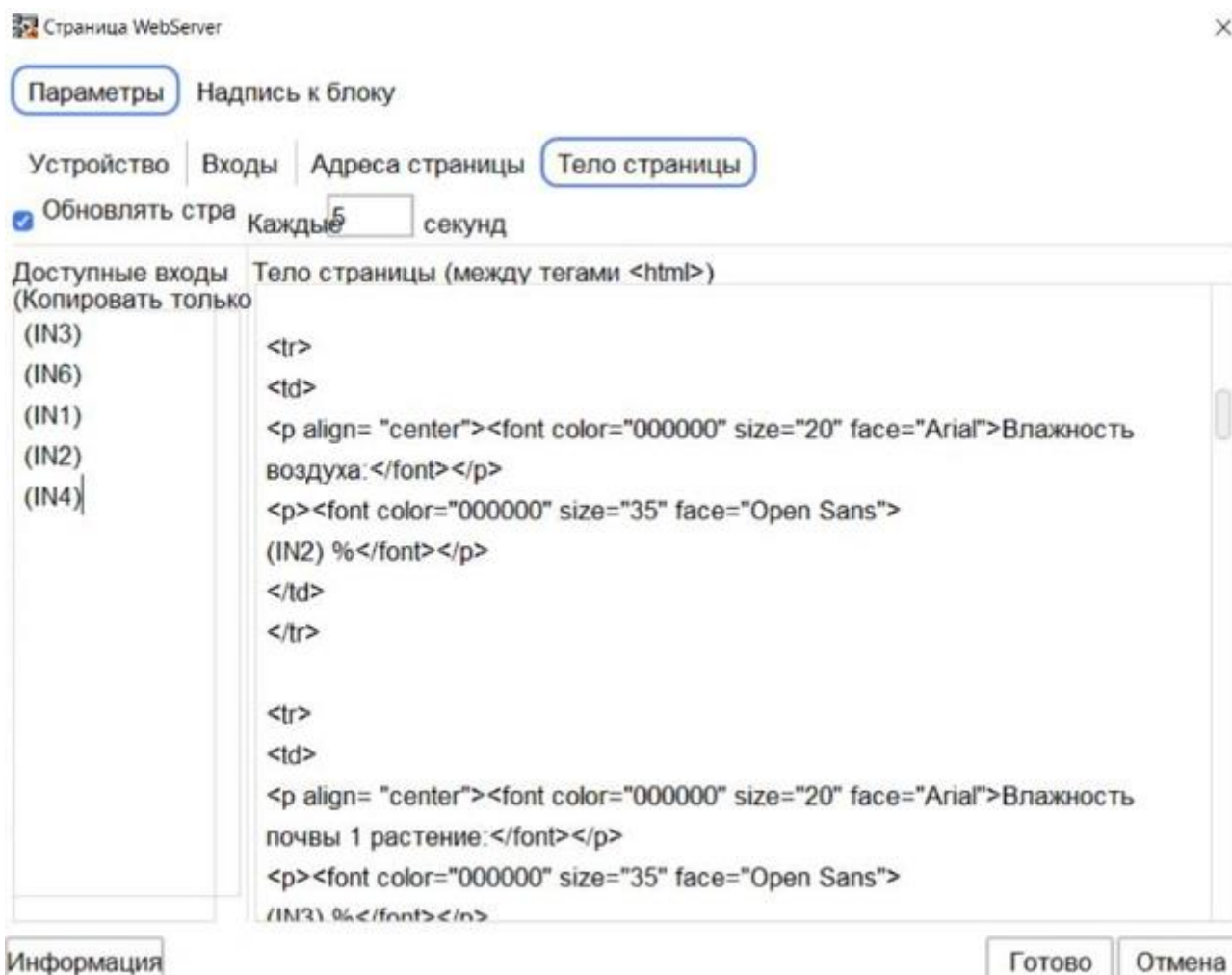


Рисунок 3.4 – HTML код у програмі

HTML (HyperText Markup Language) - це мова розмітки, яка використовується для створення веб-сторінок. HTML-код складається з різних елементів, які визначають структуру та зміст сторінки.

Код HTML може бути розширений і містити безліч інших елементів і атрибутів для створення складніших та інтерактивніших веб-сторінок. HTML працює в парі з CSS (Cascading Style Sheets) для оформлення та стилізації елементів сторінки, а також із JavaScript для додавання інтерактивності та динамічної поведінки.

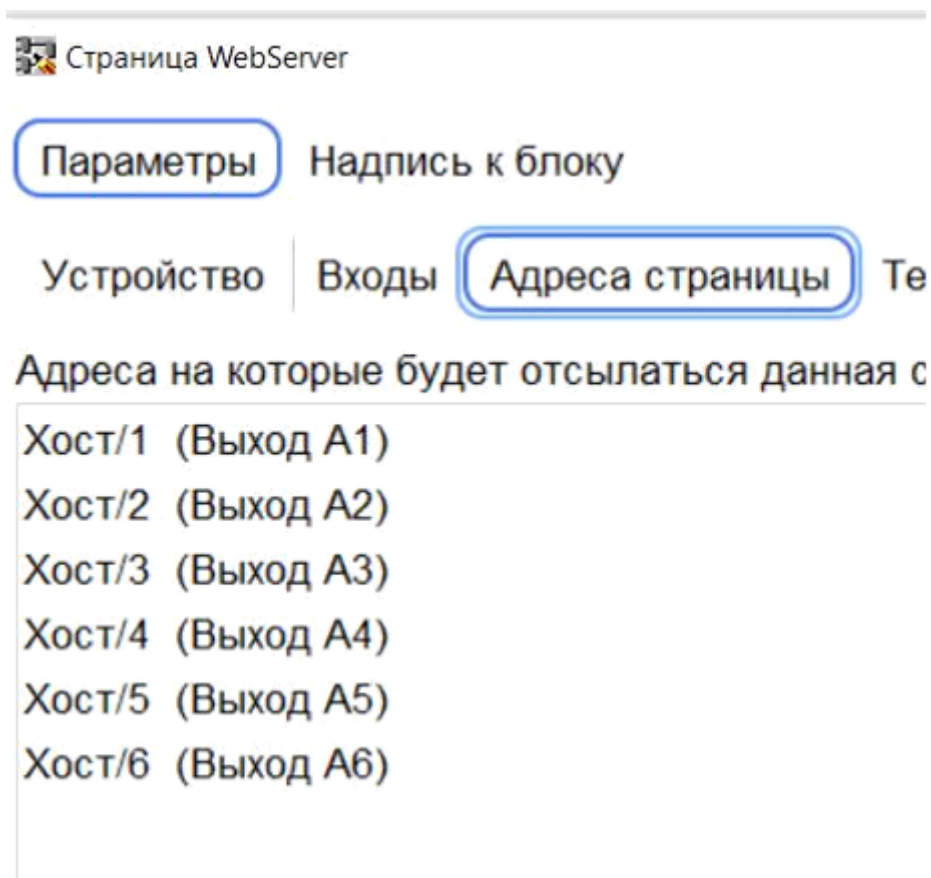


Рисунок 3.5 – Скріншот частини програми з хостами

Нам потрібні будуть тригери rs, tt.

RS тригер: Вхід S встановлює вихід тригера Q одиничний стан, вхід R скидає вихід тригера Q в нульовий стан.

T-тригер - це лічильний тригер. Цей тригер має лише один вхід. Після надходження на вхід T імпульсу, стан тригера змінюється прямо протилежне.

Блок конвертації рядків служить для перетворення чисел на рядки.

На рисунку 3.6 представлено підсумкову зібрану схему.

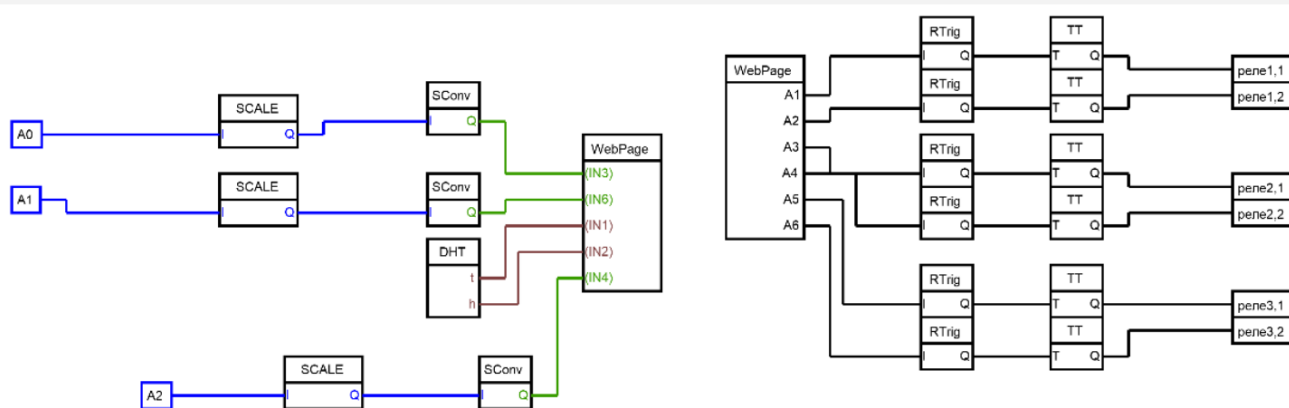


Рисунок 3.6 – Схема

Зібравши та з'єднавши систему частинами, потрібно запустити компіляцію, і програма створить програму в Arduino IDE, в якій перед завантаженням у контролер потрібно внести деякі доповнення. На рисунку 3.7 представлений код програми Arduino IDE, який створюється програмою FLProg.

3.2 Розробка моделей протидії загрозам інформації безпеки в умовах невизначеності

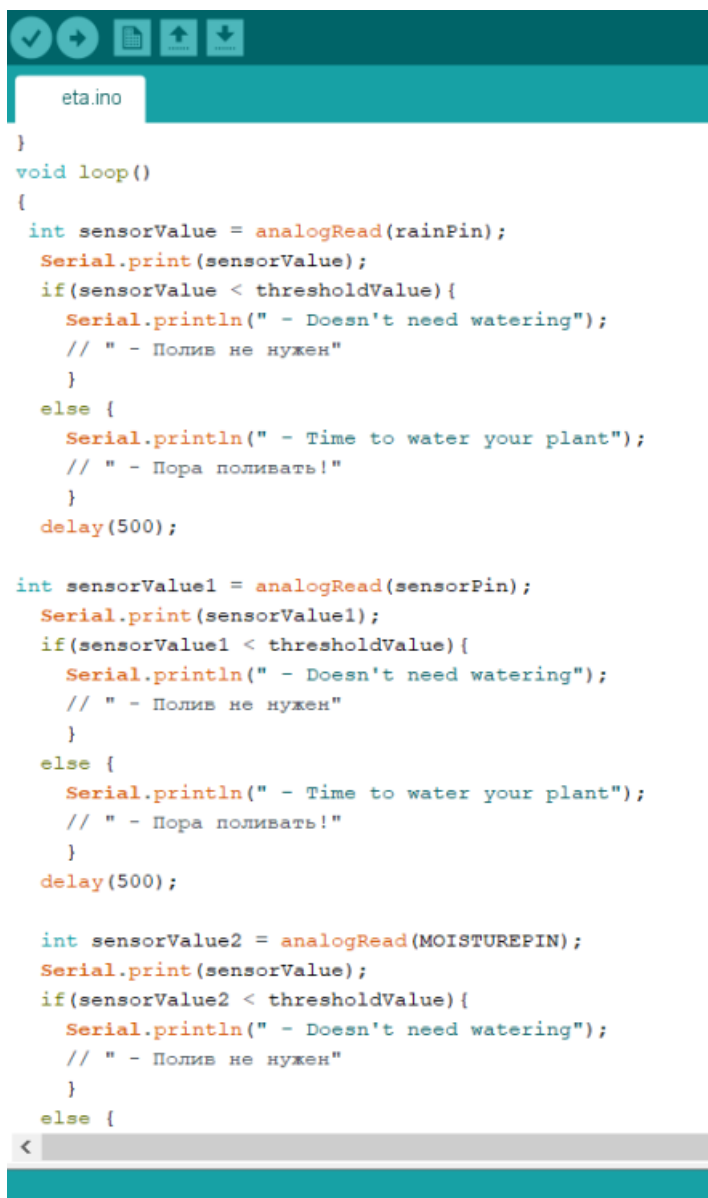
В умовах невизначеності та постійно мінливої загрозової обстановки розробка моделей протидії загрозам інформаційної безпеки є критично важливим завданням. Такі моделі дають змогу організаціям ефективно реагувати на нові загрози, що розвиваються, вживати необхідних заходів для захисту інформації та мінімізації потенційних ризиків.

Основна мета розроблення моделей протидії загрозам інформаційної безпеки в умовах невизначеності - це забезпечення стійкості та захисту інформаційних ресурсів організації.



Рисунок 3.7 – Відкриття та створення коду в Arduino IDE

У fprog відсутній датчик вологості ґрунту, він замінений аналоговим входом, Але щоб показання правильно передавалися, потрібно прописати у програмі: частина датчика вологості ґрунту (рисунок 3.8)



```

}
void loop()
{
  int sensorValue = analogRead(rainPin);
  Serial.print(sensorValue);
  if(sensorValue < thresholdValue){
    Serial.println(" - Doesn't need watering");
    // " - Полив не нужен"
  }
  else {
    Serial.println(" - Time to water your plant");
    // " - Пора поливать!"
  }
  delay(500);

  int sensorValue1 = analogRead(sensorPin);
  Serial.print(sensorValue1);
  if(sensorValue1 < thresholdValue){
    Serial.println(" - Doesn't need watering");
    // " - Полив не нужен"
  }
  else {
    Serial.println(" - Time to water your plant");
    // " - Пора поливать!"
  }
  delay(500);

  int sensorValue2 = analogRead(MOISTUREPIN);
  Serial.print(sensorValue);
  if(sensorValue2 < thresholdValue){
    Serial.println(" - Doesn't need watering");
    // " - Полив не нужен"
  }
  else {

```

Рисунок 3.8 – Частина коду, що бракує

Показчик
Температура: (IN1) C
Влажность воздуха: (IN2) %
Влажность почвы 1 растение: (IN3) %
Влажность почвы 2 растение: (IN6) %
Влажность почвы 3 растение: (IN4) %

Рисунок 3.9 – Вид веб-сторінки, перша частина – показання датчиків

На рисунку 3.9 представлений вид сторінки без показань, замість назв входів при включеній системі відображається значення в діапазоні від 0 до 100%.

Після всіх перелічених дій можна підключати шилд до мережі та відкривати сторінку (рисунки 3.9, 9.10).



Рисунок 3.10 – вид веб-сторінки, частина друга – управління клапанами

На рисунку 3.11 представлено фотографію зібраної системи.



Рисунок 3.11 – Зібраний дослідний зразок

ВИСНОВКИ

У пошуках оптимального рішення поставленого завдання було придбано, протестовано, забраковано або власноручно вдосконалено близько 7 систем. Свої корективи зробив практичний аналіз. Система, яка спочатку передбачала використання Telegram-бота в тандемі зі зв'язкою Arduino + ESP8266, виявила ряд недоліків. Більшість їх пов'язана з практичним досвідом експлуатації продукту кінцевим споживачем. Незважаючи на масове поширення месенджера Telegram, інтерфейс бота, що надається системою, не здатний задовольнити запити обивателя. Більше того, інтеграція Arduino з ESP передбачає низку складних, фізичних втручань для кінцевого споживача, що в сучасному часі викликає відторгнення. Сучасний споживач завжди віддає перевагу Plug and Play рішенням.

Окрім іншого, озвучена зв'язка могла б не тільки позбавити споживача інтуїтивно зрозумілого (френдлі) інтерфейсу, а й обмежити його право вибору, зокрема, у можливості інтеграції системи дистанційного керування зі стабільною локальною мережею за допомогою мережевого кабелю. Опитування представників сфери підтвердило мою теорію про те, що Wi-Fi з'єднанню, в умовах тепличних проєктів, довіряють менше через потенційно можливі перешкоди та обмежену зону покриття.

У процесі розробки було встановлено, що система, що працює в тандемі Arduino + Shield, обслуговує набір інструментів за протоколами http, тоді як спочатку запланований telegram-бот, який, як я знову нагадаю, не відрізняється інтуїтивно зрозумілим інтерфейсом для непрофесійного користувача, працює із протоколами https. Більше того, навіть якщо систему оптимізувати шляхом прошивки софтом, у процесі налаштування користувач зіткнувся б з протоколами безпеки, які систематично змінюватимуться – це підтверджує довідкова інформація Telegram, де зазначено, що месенджер систематично проводить подібні маніпуляції під час оновлення ключів наскрізного шифрування. Таким

чином, система могла б стати вкрай незручною в процесі експлуатації через підвищену увагу з боку споживача.

Крім зручного інтерфейсу, головною відмінністю є незалежність від стороннього продукту зарубіжного виробництва. Апелюючи простими термінами – при закритті або збоях у системі Telegram, система дистанційного керування з контролем стану рослин одномоментно могла б вийти з ладу, втративши віддалений центр управління завданнями, моніторинг та аналіз.

Центр, побудований на власному веб-сервісі, не має таких недоліків. Більш того, він надає необмежений спектр можливостей у налаштуванні, оптимізації, брендування, наповненні віджетами, плагінами та скриптами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Полякова Галина Вікторівна, Чудо-врожай. Велика енциклопедія саду та городу // URL:<http://www.maxima-library.org/redkollegiya/editors/b/280761> (дата звернення 01.04.23)
2. Вечіріна Олена Юріївна, Велика енциклопедія дачника, URL:<https://e-libra.su/books/518658-bol-shaya-enciklopediya-dachnika.html> (дата звернення 01.02.20)
3. Вирощування овочів у закритому ґрунті, URL:<https://nashausadba.com.ua/ogorod/ovochi/423-zakrytyi-grunt.html> (дата звернення 02.04.23)
4. Arduino [Електронний ресурс] - Вільна енциклопедія - 2018р - URL:<https://wikipedia.org/wiki/Arduino>, (дата звернення 15.04.23)
5. Arduino Uno, URL:<http://arduino.com/Hardware/ArduinoBoardUno> (дата звернення 10.04.23)
6. Arduino Ethernet Shield на базі W5100, URL:<https://arduinomaster.ru/datchiki-arduino/w5100-modul-arduino-ethernet/> (дата звернення 15.02.20)
7. Датчик вологості ґрунту, URL:<https://3d-diy.com/wiki/arduino-datchiki/datchik-vlazhnosti-pochvy-arduino/> (дата звернення 15.04.23)
8. Датчик вологості ґрунту (гігрометр), URL:<https://arduino.ua/prod521-datchik-vlajnosti-pochvi-gigrometr> (дата звернення 15.04.23)
9. 1-канальний модуль реле (5 В) з кріпильними отворами, URL:<https://ampermarket.kz/relay/1-kanalnyj-modul-rele-jqc-3ff-szs-krepezhnymi-otverstiyami/> (дата звернення 15.04.23)
10. Модуль Wi-Fi ESP8266 (ESP-01): підключення, розпинування, прошивка IDE, URL:<http://wiki.esp8266-wifi-module.com> (дата звернення 20.04.23)

11. Мікроклімат виробничих приміщень. Нормовані параметри мікроклімату, URL: http://ohrana-bgd.com/proizv_67.html (дата звернення 11.04.20)

12. Arduino Software IDE // [arduino.cc](https://www.arduino.cc)
URL: <https://www.arduino.cc/en/Guide/PortableIDE> (дата звернення: 15.04.2023)

13. Гуманітарний портал // [gtmarket](https://gtmarket.com/ratings/urbanization-index/info) URL:
<https://gtmarket.com/ratings/urbanization-index/info> (дата звернення 10.05.2023)

14. Arduino // [arduino.cc](https://www.arduino.cc) URL : <https://www.arduino.cc> (дата звернення: 15.04.2023)

15. Вікіпедія - Рослинництво // Wikipedia.сщ URL :
<https://wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%82%D0%B5%D0%BD%D0%B8%D0%B5%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%B2%D0%BE> (дата звернення 11.04.2023)

16. Microsoft Developer: Windows 10 IoT // Microsoft
URL: <https://developer.microsoft.com/en-us/windows/iot> (дата звернення: 16.04.2023)

17. Датчик температури та вологості // Arduino-diy URL: <http://arduino-diy.com/arduinoatchiki-temperature-i-vlazhnosti-DHT11-i-DHT22>. (Дата звернення - 16.04.2023)

18. Фреймворк Arduino // [platformio](https://platformio.org/frameworks/arduino) URL:
<https://platformio.org/frameworks/arduino>. (Дата звернення - 16.04.2023)

19. Встановлення та налаштування Arduino IDE // All-arduino
URL: <https://all-arduino.com/arduino-ide/> (Дата звернення - 16.04.2023)

Додаток А

Лістинг А.1 - Фрагмент коду HTML

```

<table width="100%" height="100%" cellpadding="0"
cellpadding="0" border="0" bgcolor="FFFFFF">
  <tr><td>
    <table width="800px" align="center" border="1" cellpadding="20">
      <tr><td>
        <h3 align="center"><font color="000000" size="20" face="Arial">
Показання</font></h3>
        </td></tr><tr><td>
          <p align="center"><font color="000000" size="20"
face="Arial">Температура:</font></p>
          <p><font color="000000" size="35" face="Open Sans">
(IN1) C</font></p>
          </td></tr><tr><td>
            <p align="center"><font color="000000" size="20"
face="Arial">Вологість повітря:</font></p>
            <p><font color="000000" size="35" face="Open Sans">
(IN2) %</font></p>
            </td></tr><tr><td>
              <p align="center"><font color="000000" size="20"
face="Arial">Вологість ґрунту 1 рослина:</font></p>
              <p><font color="000000" size="35" face="Open Sans">
(IN3) %</font></p>
              </td></tr><tr><td>
                <p align="center"><font color="000000" size="20"
face="Arial">Вологість ґрунту 2 рослина:</font></p>
                <p><font color="000000" size="35" face="Open Sans">
(IN4) %</font></p>
                </td></tr><tr><td>
                  <p align="center"><font color="000000" size="20"
face="Arial">Вологість ґрунту 3 рослина:</font></p>

```

```

<p><font color="000000" size="35" face="Open Sans">
(IN3) %</font></p>
</td></tr></table></td></tr></table>
<table width="100%" height="100%" cellpadding="0"
cellpadding="0" border="0" bgcolor="FFFFFF">
<tr><td>
<table width="1200px" align="center" border="1" cellpadding="0"
cellpadding="20">
<tr>
<td width="60%" valign="top">
<p><font color="000000" size="5" face="Open Sans">Керування
клапанами</font></p>
<table width="60%" cellpadding="0"
cellpadding="20"> <tr><td>
<p><a href="http://192.168.0.115/1"><button>1 рослина:
полив</button></a></p>
</td><td>
<p><a href=""http://192.168.0.115/2"><button>1 рослина:
добрива</button></a></p>
</td><td>
<p><a href=""http://192.168.0.115/3"><button>2 рослина:
полив</button></a></p>
</td><td>
<p><a href="http://192.168.0.115/4"><button>2 рослина:
добрива</button></a></p>
</td><td>
<p><a href=""http://192.168.0.115/5"><button>3 рослина:
полив</button></a></p>
</td><td>
<p><a href=""http://192.168.0.115/6"><button>3 рослина:
добрива</button></a></p>
</td></tr></table></td></table></td></tr></table>

```

Додаток Б

Лістинг Б.1 - Код для Arduino

```
#include <Ethernet.h>
#include <SPI.h>
#include "DHT_NEW.h"
DHT _dht1;
byte ethernet_mac [] = {0x78, 0xAC, 0xC0, 0x96, 0xBD,
0xA9}; IPAddress ethernet_ip(192, 168, 0, 115);
byte ethernet_dns [] = {192, 168, 0, 1};
byte ethernet_gateway [] = {192, 168, 0,
1}; byte ethernet_subnet [] = {255, 255,
255, 0}; EthernetServer
_tspWebServer(80); EthernetClient
_tspWebServer_client; bool _trgt6 = 0;
bool _trgt6I = 0;
bool _trgrt2 = 0;
bool _trgrt2I = 0;
bool _trgt1 = 0;
bool _trgt1I = 0;
bool _trgrt5 = 0;
bool _trgrt5I = 0;
bool _trgt3 = 0;
bool _trgt3I = 0;
unsigned long _dht1LRT = 0UL;
unsigned long _dht1Tti = 0UL;
bool _trgrt4 = 0;
bool _trgrt4I = 0;
bool _trgt5 = 0;
bool _trgt5I = 0;
bool _trgrt1 = 0;
bool _trgrt1I = 0;
bool _trgrt3 = 0;
bool _trgrt3I = 0;
```

```

bool _WSP2_A1 = 0;
bool _WSP2_A2 = 0;
bool _WSP2_A3 = 0;
bool _WSP2_A4 = 0;
bool _WSP2_A5 = 0;
bool _WSP2_A6 = 0;
bool _trgrt6 = 0;
bool _trgrt6I = 0;
bool _trgt2 = 0;
bool _trgt2I = 0;
bool _trgt4 = 0;
bool _trgt4I = 0;
int rainPin = A0;
int rain2Pin = A1;
int rain3Pin = A2;
int thresholdValue = 800;
void setup()
{
  pinMode(rainPin, INPUT);
    pinMode(rain2Pin, INPUT);
    pinMode(rain3Pin, INPUT);
  Serial.begin(9600);
  pinMode(7, OUTPUT);
  digitalWrite(7, 0);
  pinMode(8, OUTPUT);
  digitalWrite(8, 0);
  pinMode(9, OUTPUT);
  digitalWrite(9, 0);
  pinMode(11, OUTPUT);
  digitalWrite(11, 0);
  pinMode(12, OUTPUT);
  digitalWrite(12, 0);
  pinMode(13, OUTPUT);
  digitalWrite(13, 0);

  Ethernet.begin(ethernet_mac, ethernet_ip, ethernet_dns,
ethernet_gateway, ethernet_subnet);

```

```

delay(1000);
_tspWebServer.begin();
_dht1.setup(6);
_dht1LRT = millis();
_dht1Tti = millis();
}
void loop()
{
int sensorValue = analogRead(rainPin);
  Serial.print(sensorValue);
  if(sensorValue < thresholdValue){
    Serial.println(" - Doesn't need watering");
//    - Полив не потрібен
  }
  else {
    Serial.println(" - Time to water your plant");
    // - Пора поливати!
  }
  delay(500);

int sensorValue1 =
analogRead(rain2Pin);
Serial.print(sensorValue1);
if(sensorValue < thresholdValue){
  Serial.println(" - Doesn't need watering");
  // - Полив не потрібен
}
else {
  Serial.println(" - Time to water your plant");
  // - Пора поливати!
}
  delay(500);
int sensorValue2 =
analogRead(rain3Pin);
Serial.print(sensorValue2);

```

```

if(sensorValue < thresholdValue){
    Serial.println(" - Doesn't need
    watering");
    // - Полив не потрібен
}
else {
    Serial.println(" - Time to water your plant");
    // - Пора поливати!
}
delay(500);
_tspWebServer_client =
_tspWebServer.available(); if
(_tspWebServer_client) { boolean
_WSCLineIsBlank = true;
String _WSCReqest="";
bool _WSCIsFirsLine=1;
bool _WSCIsParse=0;
int _WSCPPageNumber=0;
while (_tspWebServer_client.connected())
{ while( _tspWebServer_client.available())
{char _tempWebServerChar = _tspWebServer_client.read();
if(_WSCIsFirsLine) {_WSCReqest+= _tempWebServerChar;}
if (_tempWebServerChar == '\n' && _WSCLineIsBlank)
{_sendWebServerPage(_WSCPPageNumber); break;}
if (_tempWebServerChar == '\n') {_WSCLineIsBlank = true;
_WSCIsFirsLine=0; if (!_WSCIsParse){
_WSCPPageNumber=_parseWebServerReqest(_WSCReqest); _WSCIsParse=1;} }
else if (_tempWebServerChar != '\r') { _WSCLineIsBlank = false;}
}}
if (_WSP2_A6) { if (_trgrt6I) { _trgrt6 = 0;} else {_trgrt6 =
1; _trgrt6I = 1;} } else {_trgrt6 = 0; _trgrt6I = 0;};
bool _tmp6 = _trgrt6;
if (_tmp6) { if (! _trgt6I) _trgt6 = ! _trgt6; }
_trgt6I = _tmp6;
digitalWrite(13, _trgt6);

```

```

    if (_WSP2_A5) { if (_trgrt5I) { _trgrt5 = 0;} else {_trgrt5 =
1; _trgrt5I = 1;} } else {_trgrt5 = 0; _trgrt5I = 0;};
    bool _tmp5 = _trgrt5;
    if (_tmp5) { if (! _trgt5I) _trgt5 = ! _trgt5; }
    _trgt5I = _tmp5;
    digitalWrite(12, _trgt5);
    if (_WSP2_A4) { if (_trgrt4I) { _trgrt4 = 0;} else {_trgrt4 =
1; _trgrt4I = 1;} } else {_trgrt4 = 0; _trgrt4I = 0;};
    bool _tmp4 = _trgrt4;
    if (_tmp4) { if (! _trgt4I) _trgt4 = ! _trgt4; }
    _trgt4I = _tmp4;
    digitalWrite(11, _trgt4);
    if (_WSP2_A3) { if (_trgrt3I) { _trgrt3 = 0;} else {_trgrt3 =
1; _trgrt3I = 1;} } else {_trgrt3 = 0; _trgrt3I = 0;};
    bool _tmp3 = _trgrt3;
    if (_tmp3) { if (! _trgt3I) _trgt3 = ! _trgt3; }
    _trgt3I = _tmp3;
    digitalWrite(9, _trgt3);
    if (_WSP2_A2) { if (_trgrt2I) { _trgrt2 = 0;} else {_trgrt2 =
1; _trgrt2I = 1;} } else {_trgrt2 = 0; _trgrt2I = 0;};
    bool _tmp2 = _trgrt2;
    if (_tmp2) { if (! _trgt2I) _trgt2 = ! _trgt2; }
    _trgt2I = _tmp2;
    digitalWrite(8, _trgt2);
    if (_WSP2_A1) { if (_trgrt1I) { _trgrt1 = 0;} else {_trgrt1 =
1; _trgrt1I = 1;} } else {_trgrt1 = 0; _trgrt1I = 0;};
    bool _tmp1 = _trgrt1;
    if (_tmp1) { if (! _trgt1I) _trgt1 =
! _trgt1; } _trgt1I = _tmp1;
    digitalWrite(7, _trgt1);
    if(_isTimer(_dht1Tti, 1000)) {
    if(_isTimer(_dht1LRT, ( _dht1.getMinimumSamplingPeriod())) {
    _dht1.readSensor();
    _dht1LRT = millis();
    _dht1Tti = millis();
    }
}

```

```

    }
    }
    bool _isTimer(unsigned long startTime, unsigned long period )
    {
        unsigned long currentTime;
        currentTime = millis();
        if (currentTime>= startTime) {return (currentTime>=(startTime +
period));} else {return (currentTime >=(4294967295-
startTime+period));}
    }
    void _sendWebServerPage(int sendPageNumber)
    {_tspWebServer_client.println("HTTP/1.1 200
OK");
    _tspWebServer_client.println("Connection:
close");
    if ( (sendPageNumber==1))
{if(sendPageNumber==1){_tspWebServer_client.println("Refresh: 5");}
    }
    _tspWebServer_client.println();
    _tspWebServer_client.println("<!DOCTYPE HTML PUBLIC \"-
//W3C//DTD HTML 4.01 Transitional//EN\">");
    _tspWebServer_client.println("<html><head>");
    _tspWebServer_client.println("<META content=\"text/html;
charset=utf-8\" http-equiv=\"Content-Type\">");
    _tspWebServer_client.println("</head><body>");
    if (sendPageNumber ==1)
    {_sendWebServerPage1();} if (sendPageNumber
==2) {_sendWebServerPage2();}
    _tspWebServer_client.println("</body></html>
"); delay(1); _tspWebServer_client.stop();}
    int _parseWebServerRequet(String requetAddres)
    {
        int index;
        int result=0;
        index=requetAddres.indexOf("/");

```

```

    requestAddress
    =_stringWithoutCharWithIndex(requestAddress,0,(index));
    index=requestAddress.indexOf(" ");
    requestAddress
    =_stringWithoutCharWithIndex(requestAddress,index,(requestAddress.length
    ()-index));

    if (requestAddress==""){result= 1;}
    if (requestAddress=="1"){_WSP2_A1=1; result= 2;} else
    {_WSP2_A1=0;}
    if (requestAddress=="2"){_WSP2_A2=1; result= 2;} else
    {_WSP2_A2=0;}
    if (requestAddress=="3"){_WSP2_A3=1; result= 2;} else
    {_WSP2_A3=0;}
    if (requestAddress=="4"){_WSP2_A4=1; result= 2;} else
    {_WSP2_A4=0;}
    if (requestAddress=="5"){_WSP2_A5=1; result= 2;} else
    {_WSP2_A5=0;}
    if (requestAddress=="6"){_WSP2_A6=1; result= 2;} else
    {_WSP2_A6=0;}
    return result;
}

String _stringWithoutCharWithIndex(String value, int index,int
count)
{
    String result="";
    for (int i=0; i <= value.length(); i++){
        if((i<index) ||(i>(index+count))){
            result+=value.charAt(i);
        }
    }
    return result;
}

void _sendWebServerPage1(void)
{

```

```

        _tspWebServer_client.println("<table width=\"100%\"
height=\"100%\" cellpadding=\"0\" cellspacing=\"0\" border=\"0\"
bgcolor=\"FFFFFF\" ">");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("<td>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<table width=\"800px\"
align=\"center\" border=\"1\" cellpadding=\"0\"
cellpadding=\"20\">");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("<td>");

        _tspWebServer_client.println("<h3 align=\"center\"><font
color=\"000000\" size=\"20\" face=\"Arial\">");
        _tspWebServer_client.println("Показання</font></h3>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("</td>");
        _tspWebServer_client.println("</tr>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("<td>");
        _tspWebServer_client.println("<p align=\"center\"><font
color=\"000000\" size=\"20\" face=\"Arial\">Температура:</font></p>
");
        _tspWebServer_client.println("<p><font color=\"000000\"
size=\"35\" face=\"Open Sans\">");
        _tspWebServer_client.println("");
        _tspWebServer_client.println(_dht1.temperature);
        _tspWebServer_client.println("C</font></p>");
        _tspWebServer_client.println("</td>");

```

```

_tspWebServer_client.println("</tr>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<tr>");
_tspWebServer_client.println("<td>");
_tspWebServer_client.println("<p align= \"center\"><font
color=\"000000\" size=\"20\" face=\"Arial\">Вологість
повітря:</font></p> ");
_tspWebServer_client.println("<p><font color=\"000000\"
size=\"35\" face=\"Open Sans\">");
_tspWebServer_client.println("");
_tspWebServer_client.println(_dht1.humidity);

_tspWebServer_client.println("%</font></p>");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("</tr>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<tr>");
_tspWebServer_client.println("<td>");
_tspWebServer_client.println("<p align= \"center\"><font
color=\"000000\" size=\"20\" face=\"Arial\">Вологість ґрунту 1
рослина:</font></ ");
_tspWebServer_client.println("<p><font color=\"000000\"
size=\"35\" face=\"Open Sans\">");
_tspWebServer_client.println("");
_tspWebServer_client.println((String((map(( ( analogRead
(0))), (0), (1023), (100), (0))), DEC)));
_tspWebServer_client.println("%</font></p>");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("</tr>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<tr>");
_tspWebServer_client.println("<td>");
_tspWebServer_client.println("<p align= \"center\"><font
color=\"000000\" size=\"20\" face=\"Arial\">Вологість ґрунту 2
рослина:</font></ ");

```

```

    _tspWebServer_client.println("<p><font color=\"000000\"
size=\"35\" face=\"Open Sans\">");
    _tspWebServer_client.println("");
    _tspWebServer_client.println((String((map(( ( analogRead
(1))), (0), (1023), (100), (0))), DEC)));
    _tspWebServer_client.println("%</font></p>");
    _tspWebServer_client.println("</td>");
    _tspWebServer_client.println("</tr>");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("<tr>");
    _tspWebServer_client.println("<td>");
    _tspWebServer_client.println("<p align= \"center\"><font
color=\"000000\" size=\"20\" face=\"Arial\">Вологість ґрунту 3
рослина:</font></ ");
    _tspWebServer_client.println("<p><font color=\"000000\"
size=\"35\" face=\"Open Sans\">");
    _tspWebServer_client.println("");
    _tspWebServer_client.println((String((map(( ( analogRead
(2))), (0), (1023), (100), (0))), DEC)));
    _tspWebServer_client.println("%</font></p>");
    _tspWebServer_client.println("</td>");
    _tspWebServer_client.println("</tr>");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("</table>");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("</td>");
    _tspWebServer_client.println("</tr>");
    _tspWebServer_client.println("</table>");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("<table width=\"100%\"
height=\"100%\" cellpadding=\"0\" cellspacing=\"0\" border=\"0\"
bgcolor=\"FFFFFF\" >");
    _tspWebServer_client.println("<tr>");
    _tspWebServer_client.println("<td>");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("");

```

```

        _tspWebServer_client.println("");
        _tspWebServer_client.println("<table width=\"1200px\"
align=\"center\" border=\"1\" cellpadding=\"0\"
cellpadding=\"20\">");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<td width=\"60%\" valign=\"top\"));
        _tspWebServer_client.println("<p><font color=\"000000\"
size=\"5\" face=\"Open Sans\">Керування клапанами</font></p>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<table width=\"60%\"
cellspacing=\"0\" cellpadding=\"20\">");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("<td>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<p><a
href=\"http://192.168.0.115/1\"><button>1 рослина:
полив</button></a></p>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("</td>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<td>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<p><a

```

```
href=\"\"http://192.168.0.115/2\"><button>1
удобрення</button></a></p>");
```

рослина:

```
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a
```

```
href=\"\"http://192.168.0.115/3\"><button>2 рослина:
```

```
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a
```

```
href=\"\"http://192.168.0.115/4\"><button>2
добрива</button></a></p>");
```

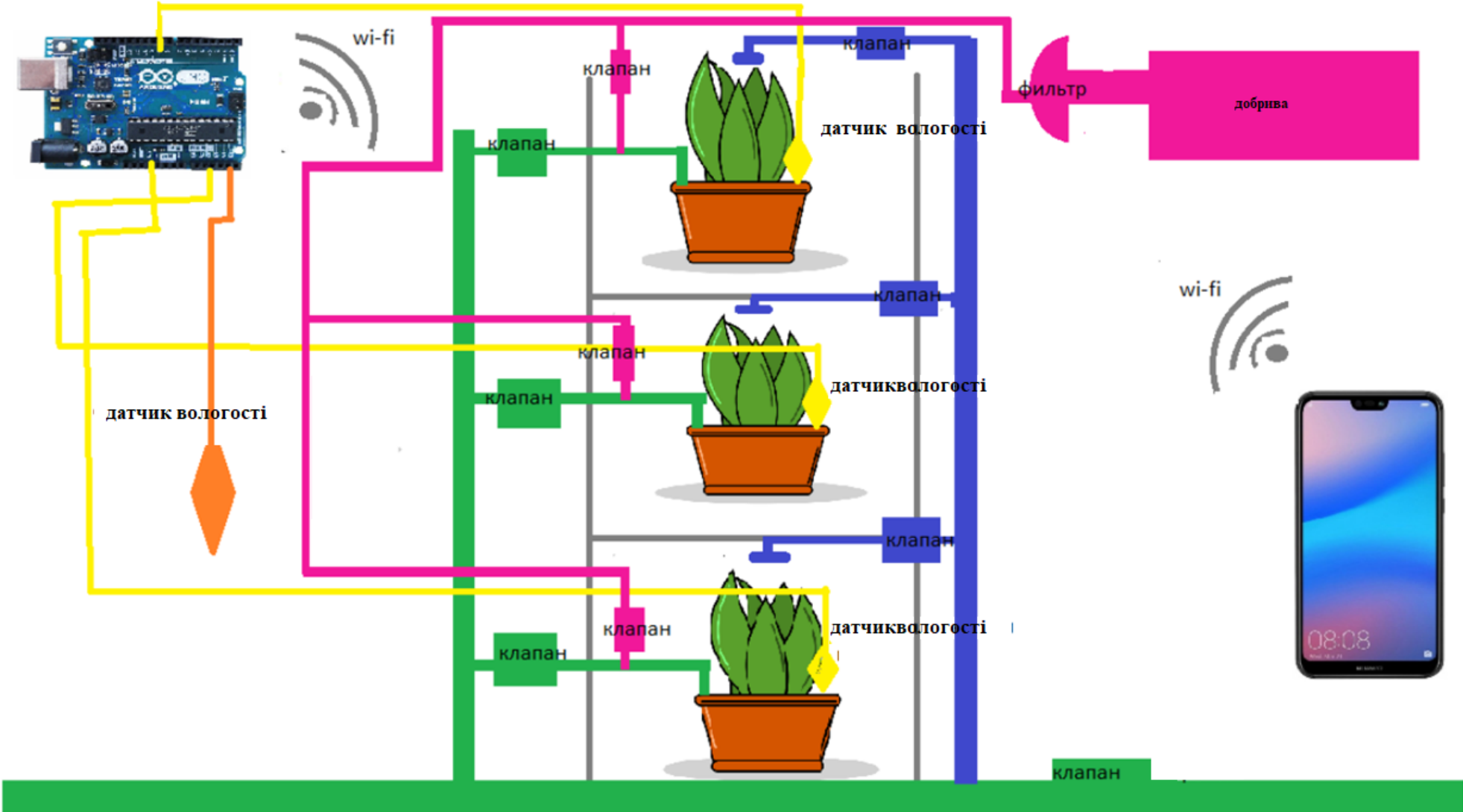
```
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a
```

```
href=\"\"http://192.168.0.115/5\"><button>3 рослина:
```

```
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td>");
_tspWebServer_client.println("");
```

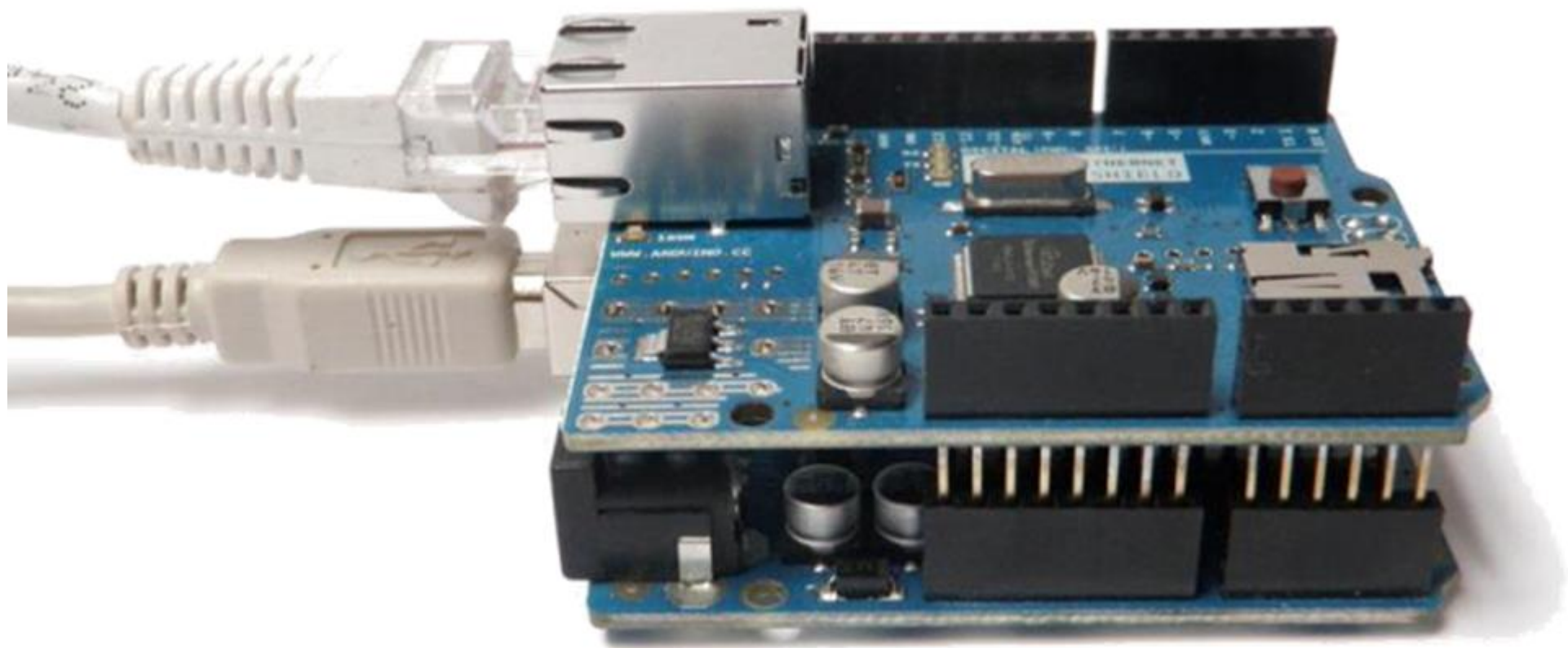
```
_tspWebServer_client.println("");  
_tspWebServer_client.println("<p><a  
href=\\\"http://192.168.0.115/6\\\"><button>3  
добрива</button></a></p>");  
_tspWebServer_client.println("");  
_tspWebServer_client.println("");  
_tspWebServer_client.println("</td>");  
_tspWebServer_client.println("");  
_tspWebServer_client.println("");  
_tspWebServer_client.println("</tr>");  
_tspWebServer_client.println("</table>");
```

Графік ринку IoT



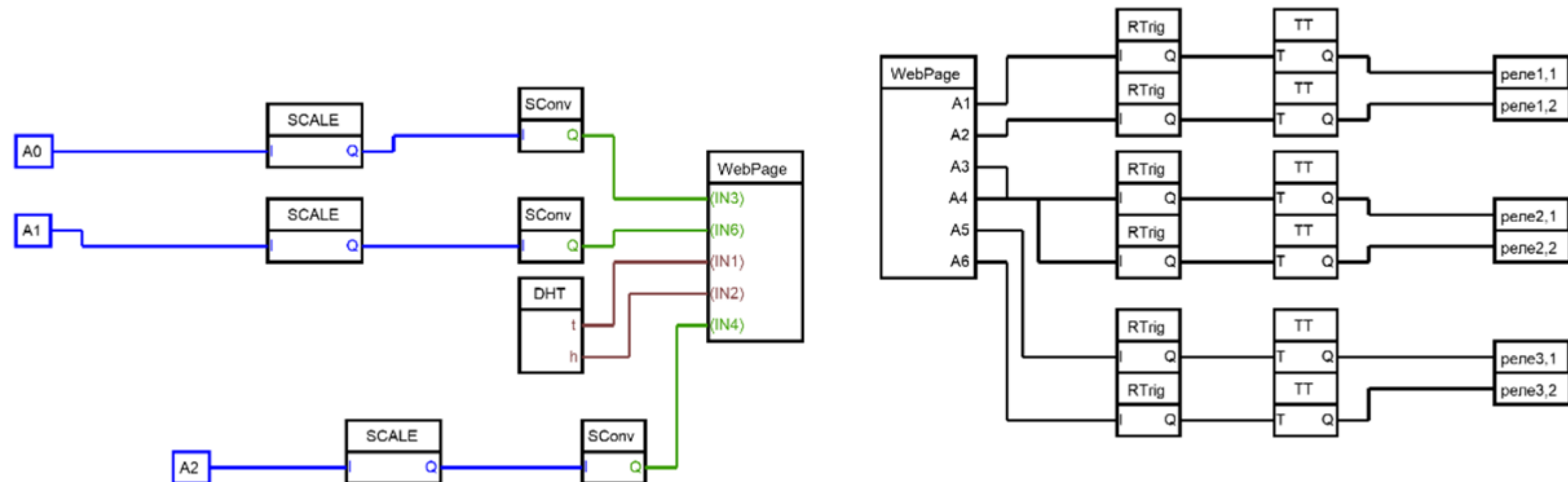
					13.02070849.51922 ПЛ1									
					Схематичне зображення системи Розробка системи дистанційного керування поливом з контролем стану рослин					Лит.		Маса	Масит.	
Зам.	Лист.	№ докум.	Підп.	Дата										
Розроб.		Чорний Д.О.												
Перев.		Скрупський С.Ю.												
Т.контр.										Лист 1		Листів 1		
Н.контр.		Щербак Н.В.								НУ «Запорізька політехніка» КНТ-519				
Затв.		Кудерметов Р.К.												

Підключення shield w5100 до плати



					13.02070849.51922 ПЛ2									
					Підключення shield w5100 до плати Розробка системи дистанційного керування поливом з контролем стану рослин					Лит.		Маса	Масит.	
Зам.	Лист.	№ докум.	Підп.	Дата										
Розроб.		Чорний Д.О.												
Перев.		Скрупський С.Ю.												
Т.контр.										Лист 1		Листів 1		
Н.контр.		Щербак Н.В.								НУ «Запорізька політехніка» КНТ-519				
Затв.		Кудерметов Р.К.												

Реалізація схеми системи



					13.02070849.51922 ПЛЗ									
					Реалізація схеми системи Розробка системи дистанційного керування поливом з контролем стану рослин					Лист.		Маса	Масшт.	
Зам.	Лист.	№ докум.	Підп.	Дата										
Розроб.		Чорний Д.О.												
Перев.		Скрупський С.Ю.												
Т.контр.										Лист 1		Листів 1		
Н.контр.		Щербак Н.В.								НУ «Запорізька політехніка»				
Затв.		Кудерметов Р.К.								КНТ-519				

Реалізація системи



Зібраний дослідний зразок

Керування клапанами

1
рослина:
полив

1
рослина:
добриво

2
рослина:
полив

2
рослина:
добриво

3
рослина:
полив

3
рослина:
добриво

Вид веб-сторінки

					13.02070849.51922 ПЛ4									
					Реалізація системи Розробка системи дистанційного керування поливом з контролем стану рослин					Лист.		Маса	Масшт.	
Зам.	Лист.	№ докум.	Підп.	Дата										
Розроб.		Чорний Д.О.												
Перев.		Скрупський С.Ю.												
Т.контр.										Лист 1		Листів 1		
Н.контр.		Щербак Н.В.								НУ «Запорізька політехніка» КНТ-519				
Затв.		Кудерметов Р.К.												