

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

**Факультет комп'ютерних наук і технологій**  
(повне найменування інституту, факультету)

**Кафедра комп'ютерних систем та мереж**  
(повне найменування кафедри)

## Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти)

на тему Розробка системи розумного будинку

Виконав: студент(ка) 4 курсу, групи КНТ-519

Спеціальності 123 «Комп'ютерна інженерія»  
(код і найменування спеціальності)

Освітня програма (спеціалізація)

«Комп'ютерна інженерія»

БОГУНОВ Р.В.

(прізвище та ініціали)

Керівник ГРУШКО С.С.

(прізвище та ініціали)

Рецензент КАЧАН О.І.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
(повне найменування вищого навчального закладу)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій  
Кафедра «Комп'ютерні системи та мережі»  
Ступінь вищої освіти (освітній ступінь) бакалаврський  
Спеціальність 123 Комп'ютерна інженерія  
(код і найменування)  
Освітня програма (спеціалізація) Комп'ютерна інженерія  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри Кудерметов Р.К.

«   »     2023 року

**З А В Д А Н Н Я**  
**НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА**

БОГУНОВУ Радіону Валентиновичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка системи розумного будинку

керівник проекту (роботи) ГРУШКО Світлана Сергіївна, к. т. н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «24» березня 2023 року № 71

2. Строк подання студентом проекту (роботи) 20 травня 2023 року

3. Вихідні дані до проекту (роботи) можливість керування освітленням, можливість керування сигналізацією

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз ринку систем «розумний будинок»

2) Проектування системи розумного будинку

3) Розробка та реалізація програмно-апаратного комплексу "розумний будинок"

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

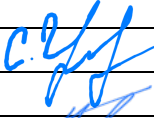
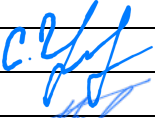


Пл1 - Програмно-апаратні комплекси «розумних будинків»;

Пл2 - Функціональні вимоги до програмного виробу;

Пл3 - Архітектура системи;

Пл4 - Особливості організації віддаленого управління системою.


## 6. Консультанти розділів проекту (роботи)

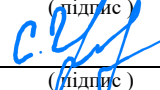
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	ГРУШКО С.С., к. т. н., доцент		
Нормоконтроль	ЩЕРБАК Н.В., ст.викл.		

7. Дата видачі завдання 01.03.2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз технічного завдання. Огляд сучасних рішень	01.03.2023 р.	
2	Розробка структури системи	15.03.2023 р.	
3	Розробка алгоритму роботи системи	20.03.2023 р.	
4	Реалізація проекту	01.04.2023 р.	
5	Дослідження роботи системи	10.04.2023 р.	
6	Оформлення отриманих результатів у ПЗ	15.04.2023 р.	
7	Оформлення графічного матеріалу	01.05.2023 р.	
8	Оформлення допоміжного матеріалу	20.05.2023 р.	

Студент  Радіон БОГУНОВ  
(підпис) (ініціали та прізвище)

Керівник проекту (роботи)  Світлана ГРУШКО  
(підпис) (ініціали та прізвище)

## РЕФЕРАТ

ПЗ: 72 с., 13 рис., 1 додаток, 13 джерел.

### ARDUINO, ОСВІТЛЕННЯ, РОЗУМНИЙ БУДИНОК, СИГНАЛІЗАЦІЯ

Об'єкт розробки – система розумного будинку для керування освітленням і сигналізацією.

Мета роботи – вивчення механізмів, технологій та розробка системи розумного будинку для керування освітленням і сигналізацією.

В дипломній роботі проведено огляд і дослідження різних систем «розумних будинків», виявлені їх переваги і недоліки. Розглянуто традиційні підходи до проектування користувальницьких інтерфейсів, проведено аналіз традиційних способів реалізації управління системами будинку.

Також в роботі описано особливості використання системи та віддаленого управління системою. Проведено відлагодження та тестування розробленого сервісу, порівняння результатів роботи з існуючими аналогами

## ЗМІСТ

Скорочення та умовні позначки .....	7
Вступ.....	8
1 Аналіз ринку систем «розумний будинок» .....	10
1.1 Апаратні платформи.....	10
1.2 Аналіз плат Arduino .....	14
1.3 Програмно-апаратні комплекси «розумних будинків» .....	17
1.3.1 Apple HomeKit .....	17
1.3.2 Google Nest.....	19
1.3.3 Amazon Alexa Smart Home .....	20
1.3.4 Samsung SmartThings.....	22
1.3.5 Xiaomi Smart Home.....	23
1.3.6 VIMAR By-me.....	24
1.3.7 Система «розумний будинок» be smart MY HOME Legrand & BTicino.....	26
1.3.8 Система «розумний будинок» iNels від компанії Elko ep.....	28
1.4 Огляд традиційних підходів до проектування користувацьких інтерфейсів ...	31
1.5 Аналіз традиційних способів реалізації управління системами будинку .....	33
1.6 Обґрунтування вибору плати Arduino.....	36
2 Проектування системи розумного будинку.....	37
2.1 Постановка задачі.....	37
2.2 Функціональні вимоги до програмного виробу .....	38
2.3 Архітектура системи .....	40
2.4 Структура роботи програмно-апаратного комплексу .....	42
3 Розробка та реалізація програмно-апаратного комплексу "розумний будинок" .....	43
3.1 Особливості використання системи .....	43
3.2 Завантаження кнопок перемикачів.....	44
3.3 Реалізація web-сервера.....	45

3.4 Організація індикації сигналізації на веб-ресурсі .....	46
3.5 Особливості організації віддаленого управління системою.....	49
3.6 Відлагодження та тестування розробленого сервісу, порівняння результатів роботи з існуючими аналогами.....	51
Висновки .....	52
Перелік джерел посилання .....	53
Додаток А.....	55

#### Перелік графічного матеріалу

- Пл1 - Програмно-апаратні комплекси «розумних будинків»;
- Пл2 - Функціональні вимоги до програмного виробу;
- Пл3 - Архітектура системи;
- Пл4 - Особливості організації віддаленого управління системою.

**СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ**

CHIP	Connected Home over IP
IDII	Interaction Design Institute Ivrea
PWM	Pulse-width modulation
PHP	Hypertext Preprocessor — PHP

## ВСТУП

В наші дні системи "розумних будинків" стали досить поширеними навіть серед мешканців невеликих міст, не кажучи вже про столиці та мегаполіси. Керування енергетичними системами, забезпечення безпеки, контроль за використанням комунальних ресурсів, економія електроенергії - усі ці фактори підтверджують актуальність "розумних" технологій. Наразі ні для кого не є секретом, що такі технології мають майбутнє та будуть продовжувати розвиватися. Актуальним залишається питання про стратегію розвитку таких систем, технології, які використовуватимуться для їх розробки та експлуатації.

"Розумний будинок" - це одна з пріоритетних галузей розвитку Інтернету, в якій працює велика кількість людей, компаній та структур. Усі вони активно включають різноманітні функції управління побутовими пристроями та забезпечують взаємодію між ними. Наприклад, провайдери зв'язку мають великі шанси зайняти певну нішу на ринку розумних будинків, оскільки вони можуть забезпечити комплексний підхід до організації роботи в цьому напрямку: забезпечують канал зв'язку, наявність спеціального обладнання та достойний рівень його обслуговування.

Використання сучасних засобів зв'язку, датчиків, вбудованих у побутові пристрої та обладнання, а також наявність програмного забезпечення, яке забезпечує взаємодію всіх компонентів системи та їх управління, дозволяє створити умови, при яких будинок стає не тільки більш безпечним, а й значно економить ресурси. В зимовий період споживання електроенергії значно збільшує вартість комунальних послуг. День стає коротшим, і потреба в освітленні зростає.

Крім того, важливо звернути увагу на взаємодію між пристроями у складі системи «Розумний будинок». Для того, щоб усі пристрої працювали синхронізовано і не заважали один одному, потрібно розробити інтелектуальне програмне забезпечення. Це дозволить забезпечити роботу системи без перебоїв і



допоможе уникнути ситуацій, коли пристрої можуть зазнати збою або невірно виконувати свої функції.

Також, важливо звернути увагу на забезпечення безпеки системи «Розумний будинок», оскільки збільшення кількості підключених до Інтернету пристроїв збільшує ризик хакерських атак. Тому розробка надійної системи захисту даних є одним з головних завдань в розробці таких систем.

В цілому, системи розумних будинків мають великий потенціал у покращенні якості життя людей, забезпеченні економії ресурсів та підвищенні безпеки проживання. Розвиток цієї галузі вимагає технічних знань і креативного підходу до розробки і впровадження нових технологій, що дозволить підвищити рівень комфорту та ефективності управління житловим простором.

# 1 АНАЛІЗ РИНКУ СИСТЕМ «РОЗУМНИЙ БУДИНОК»

## 1.1 Апаратні платформи

Аналіз ринку систем "розумний будинок" показує, що цей сегмент ринку зростає зі швидкістю близько 20% щорічно. Це зумовлено збільшенням свідомості споживачів про переваги "розумного будинку" і зниженням вартості технологій для домашньої автоматизації.

На сьогоднішній день на ринку систем "розумний будинок" можна виділити такі основні гравці: Apple HomeKit, Google Nest, Amazon Alexa Smart Home, Samsung SmartThings та Xiaomi Smart Home. Кожен з цих брендів має свої переваги та недоліки, але загалом вони пропонують подібні можливості для домашньої автоматизації.

Протягом останніх кількох років, головними трендами на ринку систем "розумний будинок" стали інтеграція з голосовими помічниками та збільшення кількості підключених пристроїв. Зокрема, все більше брендів починають підтримувати голосові помічники від Google, Amazon та Apple.

Зараз на ринку систем "розумний будинок" є більше 1000 підключених пристроїв, що дозволяє користувачам створювати власні екосистеми для домашньої автоматизації. Однак, багато пристроїв мають власні екосистеми, що може ускладнювати інтеграцію та керування всім домашнім комплексом.

Загалом, ринок систем "розумний будинок" є досить конкурентним та швидкозростаючим. За прогнозами експертів, вартість цього ринку складе більше 100 мільярдів доларів до 2025 року.

Однак, на даний момент, ринок «розумного будинку» ще не досяг свого піку і має потенціал для подальшого росту. Є багато компаній, які активно входять на цей ринок і пропонують свої власні програмно-апаратні комплекси, такі як Facebook Portal, LG SmartThinQ, TP-Link Kasa Smart і т.д.

Також варто зазначити, що ринок «розумний будинок» ще не досить стандартизований. Кожна компанія пропонує свої власні протоколи та стандарти, що ускладнює інтеграцію різних систем між собою. Однак, існують ініціативи, такі як Project Connected Home over IP (CHIP), які мають на меті створення спільного стандарту для систем «розумного будинку».

У цілому, ринок «розумний будинок» є перспективним та має великий потенціал для подальшого росту, але для того, щоб це стало можливим, потрібні стандартизація та зниження вартості обладнання. Також, з урахуванням збільшення кількості пристроїв, які мають доступ до Інтернету речей, необхідно забезпечити надійну та безпечну інфраструктуру для зберігання та обробки даних.

Ще однією причиною зростання ринку є підвищення обізнаності споживачів про переваги технологій "розумних будинків". Дедалі більше людей розуміють, що такі системи можуть допомогти зекономити час, знизити рівень споживання енергії та покращити безпеку в їхньому домі.

Крім того, з'являються нові можливості для розвитку систем "розумний будинок" завдяки прогресу в галузі штучного інтелекту, машинного навчання та інтернету речей. Наприклад, за допомогою штучного інтелекту системи можуть навчитися аналізувати поведінку користувачів та автоматично пристосовуватися до їхніх потреб.

На сучасному ринку присутнє велике число систем "розумних будинків". Більшість компаній, що працюють у цій галузі, не розголошують основну технічну інформацію про використовувані ними плати та модулі. Система "розумний будинок" може бути реалізована на таких платах, як Zigbee, Arduino та інші. Розглянемо особливості реалізації системи типу "розумний будинок" на платі Arduino.

Перша плата прототипу була виготовлена Массімо Банці у 2005 році. Плата мала простий дизайн та ще не називалась Arduino. Того ж року, Массімо Банці дав їй назву Arduino [1]. Перший прототип плати показано на рис. 1.1.

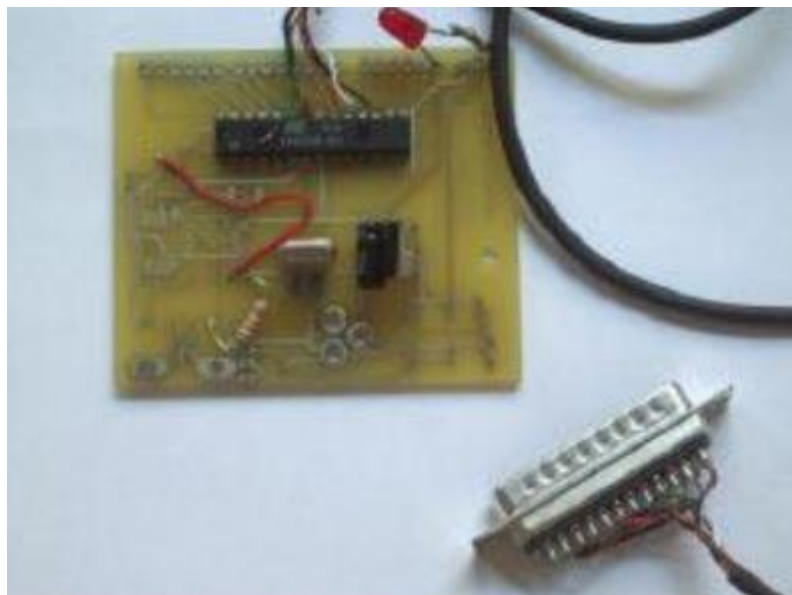


Рисунок 1.1 - Прототип плати Arduino [2]

У 2005 році Банчі, студент Інституту дизайну взаємодії міста Івреа (Interaction Design Institute Ivrea, IDII), створив простий інструмент для студентів - Arduino. Однак, ця скромна плата викликала міжнародну революцію в електронному саморобництві. Зараз її можна придбати за \$30 або зібрати самому. Всі схеми та вихідні коди Arduino доступні безкоштовно за відкритими ліцензіями. Завдяки цьому Arduino стала найвпливовішою апаратною системою свого часу з відкритим вихідним кодом.

Платформа Arduino відкриває безліч можливостей для розробки систем домашньої автоматизації («розумний будинок»). Завдяки маленькому розміру і низькій вартості, платформа Arduino стала популярною серед гоббі-розробників та професіоналів, які шукають доступні рішення для розробки власних проектів домашньої автоматизації.

Однією з переваг використання платформи Arduino є можливість розробки власних проектів з нуля, використовуючи різні сенсори та пристрої зв'язку, такі як Wi-Fi або Bluetooth. За допомогою платформи Arduino можна розробити систему керування освітленням, терморегулюванням, безпекою, розвагами та багато іншого.

Одним з головних компонентів платформи є мікроконтролер, який контролює роботу всіх підключених до нього пристроїв. Крім того, платформа Arduino має ряд сенсорів, наприклад, сенсори температури, вологості, руху та датчики світла, які можуть використовуватися для збору даних та контролю над пристроями, які вони моніторять.

Існує безліч сторонніх бібліотек та додатків, що дозволяють розширити можливості платформи Arduino. Наприклад, існують додатки, які дозволяють підключати платформу до голосових помічників, таких як Amazon Alexa або Google Assistant, щоб керувати системою за допомогою голосу.

Arduino можна використовувати в різних аспектах розумного будинку. Один з таких аспектів - це контроль електричної мережі та освітлення. Наприклад, за допомогою плати Arduino можна створити систему автоматизації освітлення в будинку, яка дозволить вмикати і вимикати світильники за розкладом, контролювати яскравість світла, створювати різноманітні ефекти освітлення, використовуючи світлодіоди та інші елементи.

Також, плата Arduino може бути використана для контролю за станом дверей і вікон, наприклад, з допомогою сенсорів магнітного поля. Додатково, плата може використовуватися для контролю температури та вологості повітря в приміщенні, за допомогою датчиків, які підключаються до неї.

Arduino також можна використовувати для створення системи безпеки в будинку. Наприклад, з її допомогою можна підключити датчики руху, які будуть спрацьовувати в разі виявлення підозрілого руху в будинку, або датчики диму, які сповістять про виявлення диму або вогню в будинку. Після спрацювання датчиків можна налаштувати систему, щоб вона надсилала повідомлення на мобільний телефон або інші пристрої.

Arduino може бути використана як базова плата для будь-якої системи розумного будинку, яка потребує автоматизації і контролю різних параметрів. За допомогою різноманітних датчиків, реле та інших пристроїв, що підключаються до плати, можна створити досить потужну інфраструктуру для управління різними аспектами життя в будинку.

## 1.2 Аналіз плат Arduino

Завдяки відкритості системних плат Arduino, допускається їх вільна модифікація. Тому будь-який виробник плат може випускати аналог плати Arduino, вносити зміни у саму плату, не кажучи вже про вільну комплектацію наборів. Оскільки Arduino - це бренд виробника, то аналоги зазвичай використовують подібні назви з Arduino - Frduino, Freeduino, Xdruido, DCcduino, Robotale, Funduino та безліч інших плат китайських виробників (Wemos). Бренд може і не бути вказаним, а буде присутня надпис типу for Arduino. Оригінальні плати Arduino виробляються в Італії, а більшість плат-аналогів - в Китаї [3].

Візуально ці плати виглядають однаково (на Arduino присутній фірмовий логотип у вигляді знака безкінечності).

У виборі плат для розробки ми будемо розглядати тільки оригінальні плати Arduino.

Плати Arduino відрізняються не тільки своїми розмірами, але й кількостями виводів, а також можливістю підключення так званих «шілдів». «Шілд» - це підвид плат розширення, які з'єднується з Arduino [4].

Розглянемо деякі різновидності Arduino.

Arduino Mini - це одна з найменших плат Arduino, яка має компактні розміри всього 30x18 мм. Ця плата призначена для розробки дуже компактних проектів, які потребують мінімального простору.

Arduino Mini має 14 цифрових входів/виходів, 8 аналогових входів та може працювати з напругою від 3,3 до 5 В. На платі є USB-порт для програмування та відлагодження коду.

Однією з переваг Arduino Mini є її малий розмір, що дозволяє вбудовувати її в найрізноманітніші проекти, такі як роботи, пристрої «розумного будинку», системи автоматизації тощо. Однак, через свою малу розмірність, плата має деякі

обмеження в потужності та можливостях порівняно з більшими платами Arduino [5].

Arduino Nano є однією з найбільш компактних плат Arduino. Це мікроконтролер, який може бути використаний для розробки різноманітних електронних проектів. Вона має малий розмір всього 18 мм x 45 мм, що робить її ідеальною для проектів, де місце обмежене.

Arduino Nano заснована на мікроконтролері ATmega328, який працює з тактовою частотою 16 МГц і має 32 Кб флеш-пам'яті та 2 Кб оперативної пам'яті. Плата має 14 цифрових входів/виходів, 8 аналогових входів та може керувати 6 пульсуючими широтно-імпульсними (PWM) виходами. Крім того, є можливість використання преривань, що дозволяє працювати з сенсорами та іншими периферійними пристроями.

Плата має USB-порт для з'єднання з комп'ютером та програмування. Вона може бути програмована за допомогою Arduino IDE, який надає широкі можливості для розробки програмного забезпечення та керування апаратурою.

Arduino Nano має вбудований регулятор напруги, що дозволяє використовувати плату з різними джерелами живлення, включаючи батареї. Плата має також вбудований кварц, що дозволяє їй працювати з точністю.

Завдяки своїм компактним розмірам та широким можливостям, Arduino Nano є популярним вибором для багатьох проектів, що вимагають малого розміру та гнучкості. Вона може бути використана для розробки різноманітних проектів, включаючи автоматизацію домашніх розваг, створення роботів та управління датчиками [6].

Arduino Leonardo - це плата розвитку мікроконтролера, розроблена компанією Arduino LLC. Вона базується на мікроконтролері ATmega32U4 від компанії Microchip Technology, що має вбудований контролер USB. Це дозволяє використовувати Leonardo для створення проектів, що працюють як звичайний пристрій USB, наприклад, клавіатура або миша.

Ця плата має 20 цифрових вхідних/вихідних портів, 7 з яких можуть бути використані для генерації широкоформатних (PWM) сигналів. Крім того, є 12

аналогових вхідних портів, які можуть використовуватися для зчитування аналогових сигналів, таких як показники температури або світла.

Arduino Leonardo має вбудований USB-порт, що дозволяє програмувати плату та передавати дані через USB-інтерфейс. Крім того, плата має вбудований регулятор напруги, який дозволяє живити плату від різних джерел живлення, таких як батареї або адаптер живлення.

Ця платформа також має можливість підключення додаткових плат розширення, відомих як "шілди" (shields), що дає можливість додавати до плати додаткові функції та можливості.

За допомогою Arduino Leonardo можна створювати різноманітні проекти, від простих світлових і звукових ефектів до складних систем автоматизації домашнього освітлення, контролю клімату в приміщенні та інших. Дана плата добре підходить для створення проектів, що вимагають взаємодії з комп'ютером, таких як програмування макросів для клавіатури або створення власних контролерів ігор [8].

Arduino Mega - це більша за розмірами та функціональністю версія платформи Arduino, яка має більше виводів і дозволяє підключення більшої кількості датчиків, виведення пристроїв і інших компонентів.

Плата має мікроконтролер ATmega2560, який є частиною процесора AVR від компанії Microchip. Цей мікроконтролер має 54 цифрових входи / виходи, з яких 14 можуть використовуватися як PWM виходи (модуляція ширини імпульсу), 16 аналогових входів, 4 UART (управління асинхронним передачею даних), 16 МГц кварцовий резонатор, USB-з'єднання, живлення від USB або зовнішнього джерела і т.д.

Ця мікросхема має 256 КБ флеш-пам'яті для зберігання програмного коду і 8 КБ ОЗУ для зберігання даних. Ця плата має додаткові можливості для розширення функціоналу за допомогою роз'ємів, що називаються "шілди", які можна підключати до плати.

Особливості цієї платформи включають такі можливості, як підтримка серійної комунікації з датчиками і пристроями через UART, забезпечення високої



швидкості передачі даних, розширення пам'яті для більш складних проектів, а також підтримка різних програмних бібліотек, які забезпечують більшу функціональність для розробки проектів на Arduino.

Arduino Mega - це ідеальний вибір для більш складних і вимогливих проектів, де необхідно багато виводів та додаткової пам'яті для зберігання програмного коду та даних [9].

### **1.3 Програмно-апаратні комплекси «розумних будинків»**

#### **1.3.1 Apple HomeKit**

Apple HomeKit - це програмна платформа, розроблена компанією Apple, призначена для управління «розумними пристроями» в домі за допомогою мобільних пристроїв з iOS або macOS.

HomeKit включає в себе два компоненти: платформу для розробників та додаток для кінцевих користувачів. Платформа для розробників дозволяє створювати підтримку для різноманітних «розумних» пристроїв, а додаток для кінцевих користувачів надає зручний і простий інтерфейс для управління цими пристроями.

Система підтримує різні типи «розумних» пристроїв, включаючи розетки, лампи, термостати, датчики руху та багато інших. Пристрої, що підтримують HomeKit, можуть бути додані до додатка Home на iPhone або iPad, де їх можна легко налаштувати та керувати. Крім того, HomeKit підтримує голосове керування через Siri, що дозволяє користувачам керувати своїми «розумними» пристроями безпосередньо з голосових команд.

Однією з основних переваг системи є висока рівень безпеки. Всі пристрої, що підтримують HomeKit, мають спеціальний чіп, що забезпечує криптографічний захист даних, що передаються між пристроями та мобільним пристроєм. Крім того,

додаток дозволяє налаштовувати рівні доступу до пристроїв для різних користувачів, що робить його ідеальним вибором для сімей з дітьми.

Основні можливості HomeKit:

- дозволяє керувати різними пристроями, які підтримують цю технологію, включаючи розумні розетки, лампи, термостати, дверні замки, датчики руху та інші;

- дозволяє створювати різні сцени та автоматизації для зручного керування вашим будинком. Наприклад, ви можете налаштувати автоматичне ввімкнення світла, коли ви повертаєтеся додому, або автоматичне вимкнення всіх пристроїв перед сном;

- підтримує роботу з голосовими помічниками, такими як Siri, що дозволяє керувати вашим будинком голосом;

- дозволяє керувати вашим будинком віддалено через додаток на вашому смартфоні або планшеті;

- пропонує різні можливості для забезпечення безпеки вашого будинку, включаючи відеонагляд, дверні дзвінки та системи сигналізації.

Для того, щоб використовувати HomeKit, потрібно мати пристрій Apple з операційною системою iOS або iPadOS, а також розумний пристрій, який підтримує технологію HomeKit. Для налаштування та керування пристроями використовується додаток Home, який поставляється з операційною системою iOS або iPadOS.

Хоча Apple HomeKit є потужним і зручним інструментом для управління розумним будинком, він також має деякі недоліки:

- обмежена кількість пристроїв - працює тільки з певними пристроями, що мають вбудований чіп Apple HomeKit. Це обмежує кількість пристроїв, які можна використовувати з системою.

- обмеження при інтеграції з іншими системами - може бути обмежений у тому, які інші системи можуть бути інтегровані з ним. Наприклад, деякі системи безпеки не працюють з HomeKit, що може обмежити можливості користувача;

- висока вартість пристроїв - пристрої, які сумісні з HomeKit, часто коштують дорожче, ніж аналогічні пристрої, які не підтримують цю систему;
- налаштування HomeKit може бути дещо складним, особливо для користувачів, які не мають досвіду з комп'ютерами або програмним забезпеченням;
- обмежена підтримка сторонніх додатків порівняно зі своїми конкурентами, такими як Google Nest.

### 1.3.2 Google Nest

Google Nest є програмно-апаратним комплексом для «розумного будинку», який був розроблений Google і працює на основі голосового помічника Google Assistant. Він включає в себе різні пристрої для домашньої автоматизації, такі як термостати, камери відеоспостереження, датчики руху та диму, замки для дверей та інші.

Найбільш відомими пристроями Google Nest є термостати, які дозволяють контролювати температуру в будинку за допомогою голосових команд або мобільного додатку. Вони також вміють самостійно встановлювати оптимальний режим опалення і охолодження, зберігаючи енергію.

Крім термостатів, Google Nest має в собі камери відеоспостереження, які забезпечують повний огляд території будинку та можуть відправляти повідомлення на мобільний телефон, якщо вони виявлять підозрілі рухи. Програмно-апаратний комплекс також має датчики руху, які можуть контролювати двері та вікна та сповіщати про їх відкриття або закриття, а також датчики диму, які можуть сповіщати про виявлення диму або вогню.

Google Nest підтримується голосовим помічником Google Assistant, який дозволяє користувачам керувати своїм домом за допомогою голосу. Він також підтримує різні мобільні додатки, що дає користувачам можливість керувати своїм домом з будь-якої точки світу.

Загалом, Google Nest є потужним інструментом для домашньої автоматизації, який може допомогти користувачам ефективніше використовувати енергію та збільшити безпеку в їхньому будинку. За допомогою цього програмно-

апаратного комплексу можна контролювати різні пристрої, які допомагають знизити рівень витрат енергії та забезпечити більш комфортні умови проживання.

Однією з особливостей Google Nest є те, що він має розумний алгоритм управління температурою, який називається "Nest Learning Thermostat". Цей алгоритм вивчає графік користувача та автоматично регулює температуру в будинку, забезпечуючи комфортні умови та економію енергії.

Ще однією цікавою особливістю Google Nest є те, що він може бути інтегрований з різними розумними пристроями, такими як розумні гучномовці, розумні розетки та інші. Це дозволяє користувачам створювати повністю автоматизовані сценарії, які можуть включати в себе різні пристрої та дії.

Однак, Google Nest має свої недоліки, зокрема, його висока вартість та несумісність з деякими іншими розумними пристроями. Також, для використання Google Nest необхідна стабільне і швидке Інтернет-підключення, інакше можуть виникнути проблеми з його функціонуванням.

### **1.3.3 Amazon Alexa Smart Home**

Amazon Alexa Smart Home є одним з провідних програмно-апаратних комплексів для «розумного будинку», розробленим Amazon. Він працює на основі голосового помічника Alexa, який дозволяє користувачам керувати своїм домом за допомогою голосових команд.

Система включає в себе різні пристрої для домашньої автоматизації, такі як розумні динаміки, телевізори, домашній кінотеатр, освітлення, термостати, камери відеоспостереження, датчики руху та диму, замки для дверей та інші.

Особливості Amazon Alexa Smart Home:

- голосовий помічник Alexa - дозволяє користувачам керувати своїм домом за допомогою голосових команд;
- велика кількість пристроїв - програмно-апаратний комплекс підтримує велику кількість пристроїв, які можна інтегрувати в систему;
- розширені можливості - програмно-апаратний комплекс може виконувати різні функції, такі як зміна настрою освітлення, зміна температури, відкривання та закривання дверей та інші;

- широкі можливості інтеграції - програмно-апаратний комплекс може бути інтегрований з іншими сервісами Amazon, такими як Amazon Prime Video, Amazon Music та інші;
- сумісність зі сторонніми пристроями - програмно-апаратний комплекс підтримує сумісність з багатьма сторонніми пристроями, що дозволяє користувачам інтегрувати їх в систему;
- мобільний додаток - Amazon Alexa має мобільний додаток, що дозволяє користувачам керувати своїм домом з будь-якої точки світу.

Окрім голосового керування, Amazon Alexa Smart Home підтримує різноманітні сценарії автоматизації. За допомогою таких сценаріїв можна створювати послідовності з декількох дій, які автоматично виконуються при заданих умовах. Наприклад, можна налаштувати сценарій, за яким ввечері при занесенні ключів у стоянку автомобіля, відкриваються двері в будинок, вмикається підсвітка і включається музика.

Amazon Alexa Smart Home також може працювати з іншими девайсами з різних виробників, наприклад, Philips Hue, Samsung SmartThings, Nest і багатьма іншими. Більшість з них можна підключити до Amazon Alexa Smart Home за допомогою протоколу Zigbee або Wi-Fi.

Крім того, система має можливість інтеграції з Amazon Echo Show - це "розумний" дисплей, на якому можна переглядати відео, аудіо та фотографії, а також читати книги і переглядати інформацію про погоду, новини та календа.

Особливістю Amazon Alexa Smart Home є те, що вона може навчатися. Це означає, що користувач може надавати нові команди і способи взаємодії зі своїм домом, і платформа зможе навчитися їх виконувати.

Загалом, Amazon Alexa Smart Home є однією з найпопулярніших та найбільш розповсюджених програмно-апаратних комплексів для "розумного будинку". Вона має потужні можливості з голосового керування та автоматизації, відкритий інтерфейс для роботи з іншими виробниками, а також дисплей для зручного керування і перегляду інформації.

### 1.3.4 Samsung SmartThings

Samsung SmartThings є програмно-апаратним комплексом для домашньої автоматизації, розробленим компанією Samsung. Він включає в себе різні пристрої, які дозволяють контролювати різні аспекти дому, такі як освітлення, опалення, безпека, електроніка та інші.

Однією з найважливіших особливостей Samsung SmartThings є його відкрита платформа, яка підтримує більше 5 тисяч різних пристроїв від різних виробників. Це означає, що користувачі можуть вибирати з широкого спектру пристроїв, щоб створити свій власний індивідуальний досвід домашньої автоматизації.

Samsung SmartThings має мобільний додаток, який дозволяє користувачам контролювати свій дім з будь-якої точки світу, де є доступ до Інтернету. Додаток містить різні функції, такі як контроль освітлення, термостатів, замків для дверей, датчиків руху та інших пристроїв.

Крім того, Samsung SmartThings має так звані «сцени», що дозволяє користувачам програмувати декілька пристроїв, щоб вони працювали разом. Наприклад, ви можете створити сцену, що включатиме освітлення та музику, коли ви повертаєтесь додому.

Ще одна важлива функція Samsung SmartThings - це можливість створення різних «умовних дій», які запускаються під час виникнення певних подій. Наприклад, якщо датчик руху виявляє рух в певній кімнаті, то може включатися освітлення в цій кімнаті та відправлятися повідомлення на мобільний телефон.

Крім того, система має велику кількість інтегрованих пристроїв, що дозволяє йому легко взаємодіяти з багатьма різними марками та типами пристроїв. Зокрема, він підтримує пристрої з підтримкою протоколів Zigbee та Z-Wave, що робить його сумісним з багатьма різними пристроями різних виробників.

Один з найбільш цікавих аспектів Samsung SmartThings полягає в його здатності використовувати різні датчики та інші пристрої для створення складних автоматизацій. Наприклад, можна налаштувати сценарій, в якому коли ви

відкриваєте двері, термостат змінює температуру, а розетки включають світло та кавоварку.

Система працює з голосовим помічником Bixby від Samsung, а також з голосовими помічниками Amazon Alexa та Google Assistant. Він також підтримує мобільні додатки для iOS та Android, що дозволяє керувати пристроями з будь-якої точки світу.

Узагалі, Samsung SmartThings є потужним та гнучким рішенням для домашньої автоматизації, яке може бути використано для керування різними пристроями та створення складних автоматизацій, що дозволяє забезпечити більш зручне та ефективне використання дому.

### **1.3.5 Xiaomi Smart Home**

Xiaomi Smart Home - це програмно-апаратний комплекс для розумного будинку, розроблений компанією Xiaomi, який складається з різних пристроїв, таких як датчики, камери, пульт дистанційного керування та інші.

Один з ключових пристроїв в системі Xiaomi Smart Home - це мініатюрні датчики руху, вологості, температури та інших параметрів. Вони можуть бути розташовані у різних місцях в будинку і надсилають інформацію до головної контрольної панелі. Ці датчики можуть бути використані для автоматичного керування світлом, опаленням та кондиціонуванням повітря.

Окрім датчиків, система включає в себе камери відеоспостереження, які можуть бути використані для спостереження за будинком в режимі реального часу, а також датчики диму та вуглекислого газу для попередження про пожежу та інші небезпеки.

Контроль над Xiaomi Smart Home може бути здійснений через мобільний додаток Mi Home, який доступний для смартфонів на базі Android та iOS. Цей додаток дозволяє користувачам контролювати стан пристроїв, налаштувати розклади та автоматичні правила для їх роботи.

Загалом, Xiaomi Smart Home є потужним інструментом для домашньої автоматизації з великою кількістю різноманітних пристроїв, що дозволяє користувачам збільшувати ефективність та безпеку в їхньому домі.

### 1.3.6 VIMAR By-me

VIMAR By-me - це інтелектуальна система "розумного будинку", яка дозволяє управляти всіма електричними пристроями в будинку з допомогою мобільного телефону, комп'ютера або планшета. Ця система розроблена компанією Vimar, однією з провідних компаній в галузі електротехніки та автоматизації будівель.

VIMAR By-me включає в себе центральний блок керування, який підключається до мережі електроживлення будинку, та різні модулі для керування окремими пристроями. Всі модулі підключаються до центрального блоку керування за допомогою мережевого кабелю.

За допомогою системи VIMAR By-me користувач може керувати різними пристроями в будинку, такими як освітлення, опалення, кондиціонування повітря, жалюзі, відеодомофони та багато інших. Крім того, ця система може інтегруватися з системами безпеки, такими як відеоспостереження та сигналізація, що забезпечує додатковий рівень безпеки для будинку.

Основні переваги системи VIMAR By-me полягають у зручності керування, економії енергії та забезпеченні комфорту для користувача. Користувач може керувати всіма пристроями з допомогою одного додатку на мобільному телефоні або планшеті, що дозволяє ефективно використовувати час і енергію. Крім того, система VIMAR By-me дозволяє налаштовувати різні сценарії роботи, що дозволяє автоматично включати і вимикати пристрої залежно від умов.

Основні компоненти системи VIMAR By-me включають в себе:

- центральний блок керування (By-me Server) - головна частина системи, яка відповідає за керування всіма підключеними до неї пристроями та датчиками. Цей блок може бути підключений до мережі Інтернет, що дозволяє дистанційно керувати будинком;
- дисплей (By-me Touch Screen) - сенсорний екран, який встановлюється на стіні та відображає різні параметри системи, такі як температура, освітлення та інші;



- модулі керування електроприладами (By-me Actuators). Ці модулі призначені для керування електричними пристроями в будинку, такими як освітлення, розетки та інші;
- датчики (By-me Sensors). Система підтримує підключення різноманітних датчиків, таких як датчики руху, температури, вологості та інші. Ці датчики дозволяють системі автоматично реагувати на зміни у середовищі;
- різноманітні пульти керування (By-me Remotes), що дозволяють керувати системою з будь-якої точки будинку;
- модулі розширення (By-me Expansion Modules), що дозволяють розширити функціональні можливості системи, додавши до неї нові функції, такі як контроль за витратою енергії або автоматичне керування занавісками та жалюзі.



Рисунок 1.2 – Керування пристроями автоматики

#### Переваги системи VIMAR By-me:

- широкі можливості інтеграції. Система може бути легко інтегрована з іншими системами домашньої автоматизації і взаємодіяти з різноманітними пристроями, такими як телефони, комп'ютери, телевізори та інші;
- простота управління. Система має інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко керувати всіма функціями домашньої автоматизації;
- розширюваність. Систему можна розширювати, додавати нові компоненти та функції, що робить її більш гнучкою та зручною у використанні;

- надійність та безпека. Забезпечує високий рівень безпеки та захисту від злому.

Недоліками системи є:

- висока вартість, що може стати перешкодою для багатьох користувачів;
- обмежена сумісність. Система може не підтримувати певні типи пристроїв та компонентів;
- потребує професійної установки. Для встановлення та налаштування системи необхідно звернутися до кваліфікованих фахівців, що також може збільшити вартість.

### **1.3.7 Система «розумний будинок» be smart MY HOME Legrand & BTicino**

Компанія Legrand була заснована сім'єю Legrand у 1866 році та спочатку займалася виробництвом посуду [15]. З появою електрики, фарфор почали використовувати для виготовлення ізоляторів та вимикачів. У 1989 році Legrand була придбана італійською компанією Bticino.

Система з шинною організацією характеризується наявністю "інтелектуальних" блоків, з'єднаних між собою шиною, яка слугує як для обміну інформацією, так і для живлення [16]. Фізичним носієм для забезпечення зв'язку та живлення є кабель - незахищена вита пара, до якої паралельно підключаються пристрої шинної системи. Активуючі пристрої, призначені для контролю навантажень, підключаються не тільки до шини, але й до силової лінії 230 В.

Кожен пристрій, підключений до системи MY HOME Legrand & Bticino, має інтерфейс та власний логічний блок, завдяки якому він розпізнає команди, що направляються до нього та обробляє їх згідно зі своєю функцією. Щоб включити лампу, достатньо натиснути на кнопку управляючого пристрою, який надсилає цифровий сигнал на активуючий пристрій, що підключений до лампи.

Функції системи автоматизації MY HOME, реалізовані в дизайнах серій AXOLUTE, LIVING, LIGHT та LIGHT TECH, дозволяють керувати функціями, які

раніше виконувалися окремими системами або групами об'єднаних систем, включаючи:

- керування освітленням;
- керування жалюзі та/або ролетами, вентиляторами та витяжними системами.

У відмінність від пристроїв традиційної електричної системи, пристрої системи автоматизації мають електронний модуль з програмованою логікою та повинні бути підключені до низьковольтної (27 В) цифрової 2-провідної шини.

Кожен пристрій, підключений до системи, має свій власний інтерфейс та логічний блок, за допомогою яких пристрій розпізнає команди, що надсилаються до нього та обробляє їх в межах заданої функції. Щоб увімкнути лампу, необхідно натиснути на клавішу управління пристроєм, який надішле цифровий сигнал на активуючий пристрій, пов'язаний з лампою.

У системі існують 2 типи пристроїв:

- управляючі пристрої, які підключаються тільки до шини;
- активуючі пристрої, які повинні підключатися до шини та до електромережі 220 В для управління навантаженням.

Якщо встановлення шинної системи неможливе, або ви бажаєте розширити вже існуючу електричну систему, не залучаючи будівельних робіт, систему автоматизації можна розширити за допомогою провідних радіоінтерфейсів, управляючих і радіо-пристроїв, що характеризуються високою гнучкістю встановлення.

При правильній конфігурації пристроїв системи автоматизації можна керувати навантаженнями наступними способами:

- керування окремим навантаженням (лампюю, жалюзі тощо);
- керування однією або декількома групами навантажень (наприклад, тільки жалюзі на першому поверсі, або жалюзі з північної сторони будинку і т.д.);
- одночасне керування всіма навантаженнями (наприклад, вимкнути всі лампи в будинку та / або опустити жалюзі).

Є можливість створення сценаріїв, які складаються з низки одночасних засобів управління, призначених для підвищення комфорту. Наприклад, натиснувши на кнопку управляючого пристрою або використовуючи графічне меню сенсорної панелі, можна запустити сценарій, індивідуальний для кожного користувача (синхронне включення кількох світильників, управління жалюзі тощо). Якщо система автоматизації об'єднана з 2-провідною звуковою системою та системою термоконтролю, сценарій також може включати звукове оформлення та регулювання необхідної температури в приміщенні [17].

Основні переваги My home:

- простота встановлення освітлювальних та інших приладів;
- доступна ціна;
- велика кількість компонентів системи;
- розширення функціоналу системи без капітальних будівельних робіт.

Основні недоліки My home:

- не має віддаленого управління;
- не має яких-небудь наборів (все окремо);
- складна доступність на нашому ринку.

### **1.3.8 Система «розумний будинок» iNels від компанії Elko ep**

Компанія ELKO EP - чеська компанія з 18-річним досвідом. З самого початку свого існування компанія спеціалізується на розробці та виробництві електронних модульних пристроїв [18]. Завдяки тривалому досвіду в цій галузі, сьогодні компанія пропонує своїм клієнтам досить широкий асортимент продукції, а саме, більше 400 типів пристроїв для вирішення різних завдань електроживлення, як в виробничій сфері, так і в побуті [19].

Система «розумного будинку» iNels, розроблена компанією, використовує сенсорний екран для управління всіма елементами розумного будинку та системами мультимедіа [20]. Меню складається з 3 секцій за функціями системи. Новинкою є меню для управління IP домофоном. За допомогою сенсорної панелі iTP можна контролювати всі електроприлади, освітлення та опалення. Можна

включити фільм або музику в будь-якій зоні, переглядати камери або інформацію з датчиків безпеки. Сенсорний екран показано на рисунку 1.3.



Рисунок 1.3 – Сенсорна панель iNels [21]

Вона базується на передових технологіях і забезпечує зручний та ефективний спосіб контролювати освітлення, опалення, безпеку, мультимедіа та інші системи в будинку.

Система iNels може інтегруватися з різними технологіями та пристроями, такими як освітлення, опалення, вентиляція, охорона, аудіо-відео системи, розетки та багато іншого. Це робить її універсальним рішенням для будь-якого типу будинку або об'єкту.

iNels надає зручний інтерфейс керування, що дозволяє власникам будинку з легкістю управляти всіма системами з будь-якого місця. Інтерфейс може бути доступним через смартфон, планшет або комп'ютер, що забезпечує зручність і гнучкість в управлінні будинком.

Система дозволяє забезпечити ефективне використання енергії в будинку. Вона надає можливість програмувати графіки роботи систем опалення, кондиціонування та освітлення з урахуванням режимів присутності, температурних умов, освітлення та інших факторів. Це дозволяє знижувати споживання енергії та економити кошти на комунальних послугах.

iNels має вбудовану систему безпеки, яка дозволяє контролювати та моніторити безпеку будинку. Вона підтримує функції, такі як виявлення вторгнень, датчики проти пожежі, відеоспостереження та систему контролю доступу. Власники будинку можуть отримувати повідомлення про події безпеки через мобільний додаток або електронну пошту.

Система iNels дозволяє гнучко налаштовувати та розширювати функціональність в залежності від потреб користувача. Вона підтримує різні протоколи зв'язку, такі як KNX, Modbus, Z-Wave, і може інтегруватися з іншими системами управління будинком. Це дозволяє створювати індивідуальні рішення для конкретного будинку та задовольняти потреби користувача.

iNels дозволяє створювати різні сценарії та автоматичні дії для спрощення життя власників будинку. Наприклад, ви можете налаштувати сценарій "Вітаюча домівка", коли система автоматично вмикає світло та музику, регулює температуру і відкриває двері, коли ви повертаєтесь додому. Це забезпечує зручність та комфорт у повсякденному житті.

Система дозволяє власникам будинку отримувати віддалений доступ та моніторити роботу систем з будь-якого місця через Інтернет. Ви можете включати та вимикати пристрої, контролювати стан систем та отримувати повідомлення про стан будинку.

#### Основні переваги iNels:

- великий комплект для самостійної установки;
- велика кількість підключуваних систем та модулів;
- зручне керування на вбудованій в систему сенсорній панелі;
- розширення функціоналу системи без капітальних будівельних робіт.

#### Основні недоліки iNels:

- не має віддаленого керування;
- не має жодних комплектів (все окремо);
- складна доступність на нашому ринку;
- висока вартість периферійних компонентів.

## 1.4 Огляд традиційних підходів до проектування користувацьких інтерфейсів

Користувацький інтерфейс повинен забезпечувати зв'язок між комп'ютером та користувачем, обмін діями та відповідями на них. Дизайн користувацького інтерфейсу є фактором, що впливає на три основні показники якості програмного продукту: його функціональність, естетику та продуктивність. Функціональність є фактором, на який розробники додатків зазвичай звертають основну увагу. Вони намагаються створювати програми так, щоб користувачі могли виконувати свої завдання та їм було зручно це робити. Функціональність важлива, але це не єдиний показник, який повинен враховуватися під час розробки користувацького інтерфейсу.

Естетичний зовнішній вигляд самого додатка та способу його представлення дозволяє сформувати у споживача позитивне відношення до програми. Однак естетичні характеристики досить суб'єктивні та описати їх кількісно набагато складніше, ніж функціональні вимоги або показники продуктивності. Вся естетика додатка зазвичай сводиться до простого вибору: чи відповідають між собою використовувані кольори, передають елементи інтерфейсу їх призначення та зміст представлених операцій, що відчуває людина при використанні тих чи інших елементів управління та наскільки успішно вона їх використовує.

Продуктивність та надійність також впливають на перспективу застосування програми. Якщо додаток має хороший вигляд, просте та зручне керування, але, наприклад, повільно відображає екрани, регулярно "зависає" на десятках-других секунд або зовсім перестає функціонувати з критичною помилкою при некоректних діях користувача, то користувач мало має шансів на тривалу експлуатацію. У свою чергу, швидка та стабільна робота додатка можуть частково компенсувати його не дуже стильний дизайн або відсутність якихось додаткових функцій.

Традиційні підходи до проектування користувацьких інтерфейсів (UI) визначаються кількома основними принципами та методологіями. Розглянемо декілька таких підходів.

Розробка на основі водопаду: Цей підхід передбачає послідовну структуру проектування, в якій кожен етап виконується послідовно перед переходом до наступного. Зазвичай це включає аналіз вимог, проектування, реалізацію, тестування та впровадження. Цей метод підходить для проектів з попередньо визначеними та стабільними вимогами.

Ітеративна розробка: Цей підхід передбачає поділ проекту на серію ітерацій, де кожна ітерація включає фази проектування, реалізації та тестування. Після кожної ітерації отримані результати використовуються для вдосконалення та визначення наступних етапів проекту. Цей підхід дозволяє гнучко вносити зміни та пристосовуватися до змінних вимог користувачів.

Проектування на основі виконання завдань: Цей підхід зосереджується на створенні інтерфейсу, який дозволяє користувачам виконувати свої завдання швидко та ефективно. Значна увага приділяється аналізу завдань користувача, їх послідовності та способу виконання. Це допомагає створити інтуїтивно зрозумілий та ефективний інтерфейс.

Реактивне проектування: Цей підхід покладається на активну взаємодію між користувачем та системою. Він використовує відгуки користувачів, аналіз поведінки та контексту використання для адаптації інтерфейсу та надання персоналізованого досвіду. Реактивне проектування дозволяє інтерфейсу швидко реагувати на зміни та потреби користувача, покращуючи його задоволення та ефективність.

Дослідницьке проектування: Цей підхід базується на проведенні досліджень з метою розуміння потреб та поведінки користувачів. Це включає збір даних, проведення спостережень, інтерв'ю, тестування та інші методи дослідження, щоб отримати інсайти щодо користувацького контексту та вимог. Результати досліджень використовуються для налаштування та оптимізації інтерфейсу.



Універсальний дизайн: Цей підхід зосереджений на створенні інтерфейсу, який може бути доступним та використовуваним різними категоріями користувачів, включаючи людей з обмеженими можливостями. Він враховує різні потреби та здібності користувачів і прагне створити інклюзивне та доступне середовище.

### **1.5 Аналіз традиційних способів реалізації управління системами будинку**

Технологія розумного будинку розроблена для автоматизації різних систем, які забезпечують життєзабезпечення та безпеку. Вона розпізнає зміни в оточуючому середовищі та приміщенні і реагує на них шляхом виконання вказівок користувача або автоматично. Головною особливістю цієї технології є об'єднання окремих підсистем та пристроїв в один комплекс, який керується автоматично.

Сучасні квартири складаються зі складного інженерного комплексу. Розумний будинок може взяти на себе керування системами енергозабезпечення, опалення, водопостачання, вентиляції та кондиціонування. Щоб максимально скористатися такими системами, вони повинні працювати взаємозв'язано. Наприклад, батареї не будуть нагрівати приміщення з відкритими вікнами, а бездротові технології дозволять користувачеві отримувати негайну інформацію про будь-яку надзвичайну ситуацію, незалежно від його місця перебування.

Існує два підходи до створення розумного будинку. Перший - це DIY-рішення, коли користувач самостійно збирає та підключає інтелектуальні пристрої. Такі рішення зазвичай поставляються готовими до використання і не вимагають налаштування. Але більш цікаві є професійні системи, які встановлюються спеціалізованими компаніями. Часто проект таких систем узгоджується ще на етапі будівництва будинку.

Розумний будинок не повинен бути проєктований як самостійна система. Важливо заздалегідь передбачити способи взаємодії з вищими системами, такими як системи надання екстреної допомоги або системи обліку ресурсів. Вже на етапі проєктування необхідно враховувати можливості взаємодії з зовнішнім світом, щоб забезпечити високу якість надання послуг користувачеві.

Ідеальною ситуацією є початок проєктування системи розумного будинку одночасно з проєктуванням самої будівлі або квартири, оскільки ця система вимагає великої кількості електричних і сигнальних кабелів, які необхідно розташувати відповідно.

Існує ще інший підхід, який є більш перспективним - це самоналаштовувальна мережа, яка базується на протоколах, таких як Z-Wave, ZigBee, Bluetooth та інші. Щодо провідних систем, вони будуються на основі послідовних протоколів, наприклад, 1-wire. Однак, недоліком провідних систем є необхідність прокладання спеціальних кабелів. Традиційні способи управління системами будинку, як правило, включають в себе прості перемикачі та датчики, що можуть бути керовані окремо або через центральну систему управління. Основною перевагою традиційних систем є їх простота та доступність, але вони часто не забезпечують таку гнучкість і контроль, які можуть бути досягнуті за допомогою систем розумного будинку.

Одним з найпоширеніших традиційних методів управління системами будинку є використання перемикачів і датчиків. Наприклад, переключення вимикача для включення світла, що дозволяє управляти освітленням в окремих кімнатах будинку. Датчики можуть бути встановлені для вимірювання температури, вологості, руху та інших параметрів, які можуть використовуватися для управління системами будинку.

Однак, такі традиційні методи мають певні недоліки. Наприклад, вони можуть бути складні для встановлення та налагодження, тому що вони часто потребують проводів і фізичної інсталяції. Крім того, вони не можуть забезпечувати таку гнучкість, яку можуть забезпечити системи розумного будинку,

такі як зміна налаштувань системи за допомогою мобільного додатку або голосових команд.

У той же час, традиційні системи управління також можуть бути менш безпечними, оскільки вони не мають такої ж рівня захисту від злону, як системи розумного будинку. Наприклад, якщо відключити світло в будинку за допомогою традиційного вимикача, то злодій може легко відслідкувати, коли будинок порожній.

Традиційні способи управління системами будинку в основному базуються на фізичному контакті з пристроями. Для керування освітленням, термостатами, вентиляцією та іншими пристроями використовуються перемикачі, термостати та регулятори, які встановлюються в певних місцях у будинку.

Наприклад, для керування освітленням в будинку можуть бути встановлені перемикачі на стінах кімнат. Крім того, деякі будинки мають системи автоматичного вимикання світла, які відслідковують присутність людей в кімнаті та вимикають світло, якщо немає руху протягом певного періоду часу.

Для контролю температури можуть використовуватися термостати, які встановлюються в певних місцях у будинку. Ці термостати можуть бути програмовані для автоматичного встановлення температури в будинку відповідно до розкладу та приватних налаштувань користувача.

Для керування вентиляцією, системами безпеки та іншими пристроями можуть використовуватися спеціальні панелі керування або перемикачі. Ці пристрої можуть бути встановлені в центральній частині будинку або в окремих кімнатах.

Однак, традиційні способи управління системами будинку мають свої недоліки. Вони не дозволяють користувачам керувати пристроями з дистанції та не забезпечують автоматизованого керування. Крім того, вони можуть бути незручними в використанні та не завжди ефективними в економії енергії.

Ще одним традиційним способом управління системами будинку є використання пультів дистанційного керування. Ці пульти дозволяють користувачам керувати різними пристроями в будинку з відстані, наприклад, з

крісла або ліжка. Зазвичай ці пульти мають кнопки, які включають або вимикають певні пристрої, а також регулюють їхні налаштування, такі як гучність, температуру тощо.

Інший традиційний спосіб управління системами будинку - це використання провідних або бездротових вимикачів. Провідні вимикачі дозволяють включати та вимикати різні пристрої у будинку, такі як лампи, вентилятори, електроприлади тощо, за допомогою перемикачів на стіні. Бездротові вимикачі можуть бути керовані з мобільного додатку або пульта дистанційного керування, а також можуть мати можливість програмувати певні налаштування, такі як час включення або вимкнення.

Також, традиційні системи управління системами будинку можуть включати в себе програмовані термостати, які дозволяють налаштовувати температуру в будинку заздалегідь, щоб зменшити витрати на енергію. Ці термостати можуть мати певні налаштування, які дозволяють регулювати температуру в залежності від часу доби або залежно від того, чи присутні люди у будинку.

Узагалі, традиційні способи управління системами будинку досі залишаються доволі популярними, оскільки вони зазвичай менш складні та більш доступні для користувачів.

## **1.6 Обґрунтування вибору плати Arduino**

У першій главі було розглянуто різновиди плат Arduino та проаналізовано системи "розумного будинку". Було проведено огляд найбільш популярних систем "розумного будинку" та визначені їх переваги та недоліки.

При виборі платформи Arduino для розробки системи розумного будинку, було враховано кілька важливих критеріїв. Перш за все, ми шукали платформу, яка була б зручною для розробки системи і мала широке поширення на ринку. Також

було важливо мати достатню кількість пінів на платі для підключення як малих, так і великих компонентів системи.

Хоча плати Arduino Mini і Arduino Nano підходять для готових проєктів, вони не є найкращим варіантом для розробки системи розумного будинку. Arduino UNO і Arduino Leonardo можна використовувати для таких систем, але їх можливостей може не вистачити для створення великих систем. Кожна з цих плат має свої переваги і недоліки.

Після аналізу даних можна зробити висновок, що оптимальним варіантом для розробки системи розумного будинку є плата Arduino Mega. Ця плата має значну кількість пінів і сумісна з великою кількістю шілдів, що робить її досить гнучкою для розширення системи. Крім того, для забезпечення дистанційного керування системою можна підключити Ethernet шілд і створити веб-сервер.

Вибір Arduino Mega як основної платформи дозволить розробити потужну і розширювану систему розумного будинку з багатьма функціями та можливістю дистанційного керування.

## **2 ПРОЄКТУВАННЯ СИСТЕМИ РОЗУМНОГО БУДИНКУ**

### **2.1 Постановка задачі**

Метою дипломної роботи є розробка програмно-апаратного комплексу для керування електронними системами в домі за допомогою web-інтерфейсу. Необхідно створити web-ресурс, що включатиме функціонал керування системою розумного будинку, індикації сигналізацій, а також робочих ламп та розеток. Web-ресурс має забезпечувати віддалене керування електронними системами в домі. В система повинна реалізовані наступні можливості:

- змінювати стани освітлювальних приладів;
- змінювати стан розеток;

- відображати поточний стан освітлювальних приладів, розеток та сигналізацій.

При проектуванні системи необхідно врахувати наступні рекомендації та положення:

- інтерфейс системи повинен бути максимально простим та зрозумілим для користувача;
- дизайн web-ресурсу повинен бути достатньо інформативним;
- реєстраційне вікно не створюється та не використовується;
- для роботи web-ресурсу потрібний доступ до Інтернету, весь процес роботи та керування системою повинен здійснюватися в онлайн-режимі;
- повинна бути передбачена можливість додавання нових елементів та пристроїв до програмно-апаратного комплексу;
- додавання нових елементів повинно виконуватися без порушення загальної структури web-ресурсу.

## **2.2 Функціональні вимоги до програмного виробу**

У системі, що розробляється, повинен бути реалізований наступний інтерфейсний функціонал та компоненти.

Макет головної сторінки веб-ресурсу системи управління «розумним будинком» представлено на рис. 2.1.

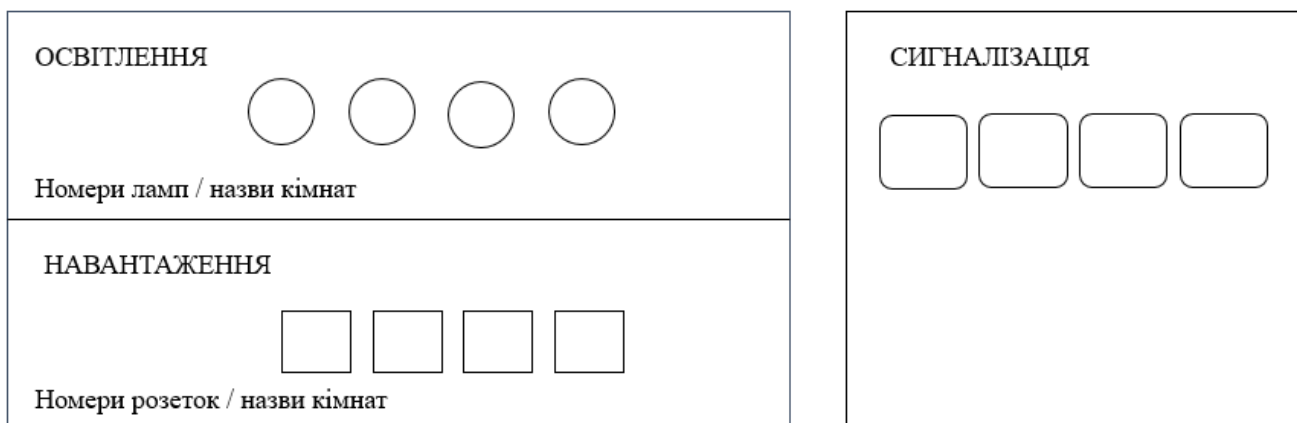


Рисунок 2.1 – Макет головної сторінки web-ресурса

Макет сторінки з індикацією веб-ресурсу системи управління «розумним будинком» представлений на рис. 2.2.

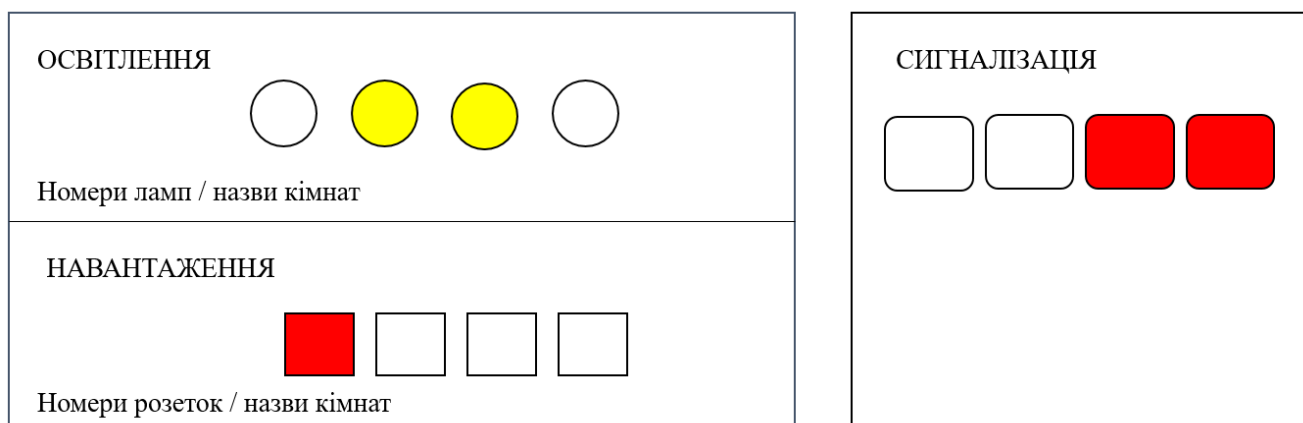


Рисунок 2.2 – Макет сторінки веб-ресурсу з індикацією

Програмний комплекс також повинен бути працездатним на пристрої, що відповідають вимогам:

- будь-яка операційна система в тому числі і мобільна;
- не менше 256 Мб оперативної пам'яті;
- доступ в інтернет зі швидкістю не менше 256 кб/с;
- будь-який браузер.

## 2.3 Архітектура системи

Як відзначалося вище, розроблювана система призначена для експлуатації в складі обчислювальних систем під керуванням будь-якої операційної системи та буде реалізована на мові C++. Для реалізації інтерфейсу використано мову PHP.

PHP є однією з найпоширеніших мов програмування для веб-розробки, і вона має численні переваги для реалізації інтерфейсу:

- PHP є досить простою мовою програмування, особливо для початківців. Синтаксис PHP схожий на синтаксис C і має низький поріг входження. Це означає, що програмісти швидко засвоять основи мови і зможуть швидко почати створювати функціональні інтерфейси;

- наявність великої активної спільноти розробників. Це означає, що завжди є велика кількість документації, підручників, форумів та інших ресурсів, які можна використовувати для вирішення проблем або отримання порад щодо реалізації інтерфейсу;

- багато розширень та фреймворків, які допомагають розробникам прискорити процес розробки. Фреймворки, такі як Laravel, Symfony, CodeIgniter та Yii, надають готові компоненти для швидкої розробки інтерфейсу, такі як маршрутизація, автентифікація, валідація форм і багато іншого;

- PHP має велику кількість вбудованих функцій та розширень, які дозволяють розробникам створювати різноманітні функціональні інтерфейси. Вбудовані функції PHP дозволяють працювати з базами даних, обробляти форми, створювати зображення, виконувати розрахунки і багато іншого;

- має високу швидкодію, що дозволяє ефективно обробляти запити на сервері. Це особливо важливо для інтерфейсів, які мають велику кількість користувачів або потребують великого обсягу обробки даних. PHP часто використовується у поєднанні з веб-сервером Apache або Nginx, що дозволяє досягти ще кращої швидкодії та масштабованості;



– є кросплатформеною мовою програмування, що означає, що ви можете розробляти інтерфейси на PHP і запускати їх на різних операційних системах, таких як Windows, Linux, macOS тощо. Це робить PHP універсальним вибором для розробки інтерфейсу, оскільки він працює на більшості серверів та хостингових платформ;

– PHP широко використовується для розробки інтерфейсів різного типу, від невеликих веб-сторінок до великих веб-додатків. Він також підтримує інтеграцію з багатьма базами даних, такими як MySQL, PostgreSQL, Oracle та іншими, що дозволяє створювати потужні інтерфейси зі збереженням та обробкою даних;

– дозволяє використовувати сторонні розширення та бібліотеки для розширення його функціональності. Це дозволяє розробникам використовувати готові рішення для реалізації певних функцій, що прискорює процес розробки та полегшує роботу з певними завданнями.

Архітектура системи досить проста, що дає певні переваги у вигляді безумовної швидкодії та зручності використання. У цілому архітектура системи представляє собою просту конструкцію для безпосереднього обміну даними між платою контролера та браузером (рис. 2.3).

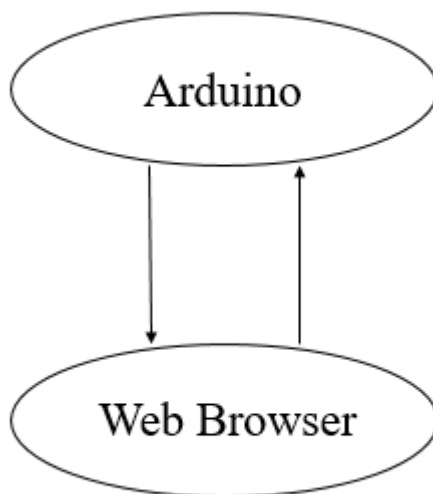


Рисунок 2.2 – Архітектура системи

## 2.4 Структура роботи програмно-апаратного комплексу

На лампочки і розетки з веб-браузера подається сигнал на включення або відключення живлення, ці сигнали передаються на маршрутизатор, маршрутизатор в свою чергу відправляє дані на інтернет шильд arduino, який активує піни на Arduino.

Сигнали ламп і розеток реалізовані в середовищі розробки Arduino як вихід, а сигналізація як input [27]. На рисунку 2.4 показана проста схема роботи системи (обміну даних).

Сигналізація проходить точно таку ж процедуру тільки в зворотному порядку.

Приклад оголошення цифрових входів/виходів сигналізації:

```
pinMode(46, INPUT);
```

```
pinMode(22, OUTPUT);
```

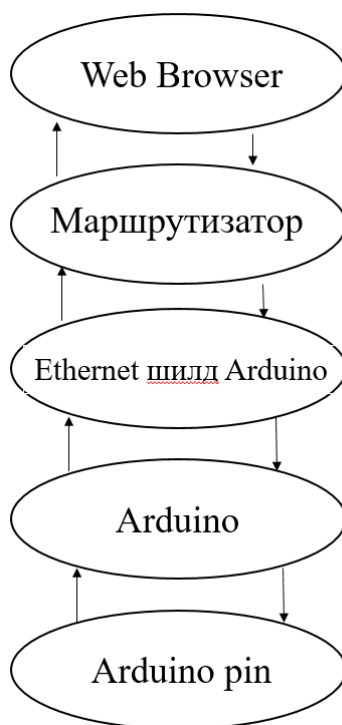


Рисунок 2.4 – Схема роботи системи

## 3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ "РОЗУМНИЙ БУДИНОК"

### 3.1 Особливості використання системи

Для початку роботи з програмно-апаратним комплексом, будь-якому користувачеві необхідно підключити плату Arduino до свого маршрутизатора через Ethernet-шилд. Ця процедура необхідна для віддаленого управління системою через веб-інтерфейс. Для того, щоб потрапити на веб-інтерфейс, необхідно ввести його адресу в рядку браузера, в нашому комплексі це `http://192.168.10.101/`. При запуску веб-інтерфейсу користувач бачить на екрані свого пристрою сторінку управління освітленням, навантаженням та індикацією сигналізації (рис. 3.1). На даному етапі користувачеві пропонується вибрати цікавлячий його розділ управління освітленням або навантаженням.

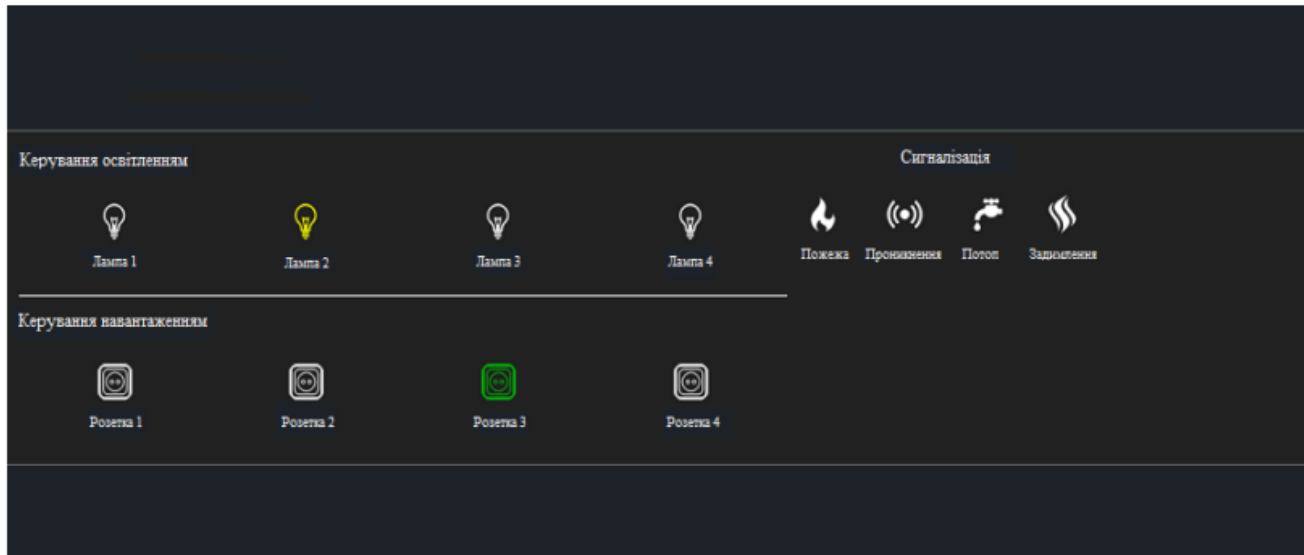


Рисунок 3.1 - Сторінка веб-сервісу з індикацією

На сторінці веб-ресурсу знаходяться наступні розділи:

- управління освітленням;
- управління навантаженням;
- сигналізація.

На платі Arduino за керування освітленням та навантаженням відповідають цифрові виходи від 22 до 29. Вихідні піни реалізовані наступним чином (ліст. 3.1).

### Лістинг 3.1 – Програмування вхідних пінів

```
pinMode(22, OUTPUT);  
pinMode(23, OUTPUT);  
pinMode(24, OUTPUT);  
pinMode(25, OUTPUT);  
pinMode(26, OUTPUT);  
pinMode(27, OUTPUT);  
pinMode(28, OUTPUT);  
pinMode(29, OUTPUT);
```

За сигналізацію відповідають цифрові входи з 46 по 49. Приклад реалізації.

### Лістинг 3.2 – Програмування пінів сигналізації

```
pinMode(46, INPUT);  
digitalWrite(46, HIGH);  
pinMode(47, INPUT);  
digitalWrite(47, HIGH);  
pinMode(48, INPUT);  
digitalWrite(48, HIGH);  
pinMode(49, INPUT);  
digitalWrite(49, HIGH);
```

## 3.2 Завантаження кнопок перемикачів

На цьому веб-ресурсі зображення кнопок та індикацій реалізовані завантаженням з веб-ресурсу. Приклад завантаження зображень показаний у ліст. 3.3.

### Лістинг 3.3 – Програмування пінів сигналізації

```

if (_gtv11)
{
_sw18 = String("http://provuz22.com/arduino/rozetka1.png");
}
else
{
_sw18 = String("http:// provuz22.com/arduino /rozetka0.png");
}

```

У нашому випадку `rozetka0`- це виключене стан, а `rozetka1` – виключений.

Всі інші кнопки реалізовані за таким же принципом. Індикація зображена на рисунку 3.2.



Рисунок 3.2 – Приклад станів модулів вкл/викл

### 3.3 Реалізація web-сервера

У представленому проекті для реалізації web-сервера використовується шілд для Arduino W5100.

Для створення web-сервісу необхідно вказати бібліотеки, які використовуються для управління Ethernet.

Також потрібно вказати параметри мережевого пристрою. Для цього вказується MAC-адреса. Обрана MAC-адреса не повинна співпадати з MAC-адресою інших мережевих пристроїв. Також IP-адреса повинна бути індивідуальною, але знаходитися в вашій підмережі. Наприклад, у проектованій

системі маршрутизатор має наступну адресу 192.168.0.1, а ПК - 192.168.0.4. З цього випливає, що дане пристрій може мати наступну IP-адресу: 192.168.0.101.

Далі необхідно вказати порт. За замовчуванням це 80, так як web-браузери за замовчуванням опитують саме його. Приклад коду показаний у лістингу 3.3.

### Лістинг 3.3 – Зазначення параметрів мережевого пристрою

```
byte ethernet_mac [] = {0x78, 0xAC, 0xC0, 0x90, 0x04, 0x30};  
IPAddress ethernet_ip(192, 168, 10, 101);  
byte ethernet_dns [] = {192, 168, 10, 1};  
byte ethernet_gateway [] = {192, 168, 10, 1};  
byte ethernet_subnet [] = {255, 255, 255, 0};  
EthernetServer _tspWebServer(80);
```

## 3.4 Організація індикації сигналізації на веб-ресурсі

На використаному веб-ресурсі частина сторінки, де знаходиться сигналізація (рис. 3.3), оновлюється кожні 10 секунд. Це зроблено для зручності, щоб не оновлювати сторінку вручну кожен раз.

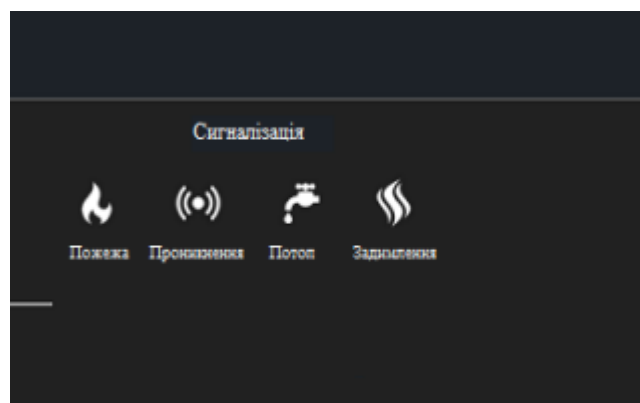


Рисунок 3.3 – Сигналізація

Схема підключення Arduino для управління електронними системами будинку показана на рис. 3.4.

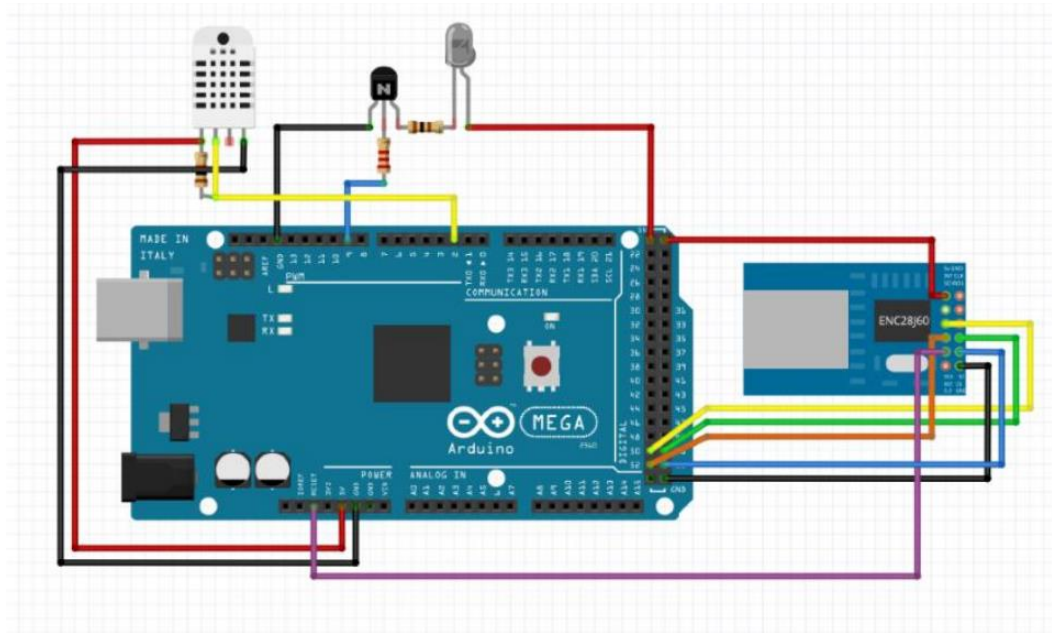


Рисунок 3.4 – Модулі системи

Підключення Ethernet шилда до Arduino mega 2560 показано на рис. 3.5



Рисунок 3.5 - Ethernet шилд + Arduino Mega 2560

Ethernet Shield - це додатковий модуль, який можна підключити до мікроконтролера або платформи Arduino для додавання підтримки мережевого з'єднання по протоколу Ethernet. Він включає в себе контролер Ethernet і фізичний інтерфейс (часто з роз'ємом RJ45) для підключення до локальної мережі.

Він надає можливість здійснювати зв'язок з Інтернетом або локальною мережею з використанням стандартних протоколів передачі даних, таких як TCP/IP. Це відкриває широкі можливості для взаємодії з серверами, обміну даними та доступу до веб-ресурсів.

За допомогою Ethernet Shield можна реалізувати різні проекти, такі як мережеві контролери, веб-сервери, IoT-пристрої та багато іншого. Він дозволяє взаємодіяти з Інтернетом, передавати та отримувати дані, керувати пристроями через мережу і виконувати інші мережеві завдання.

Ethernet Shield може бути використаний в поєднанні з різними бібліотеками програмного забезпечення, які спрощують роботу з мережевим з'єднанням і забезпечують простоту програмування. Цей модуль є популярним серед розробників, оскільки він дозволяє швидко додати мережеві можливості до їхніх проектів, простим і зрозумілим способом.

Arduino Mega 2560 є однією з платформ мікроконтролерів Arduino і є розширеною версією оригінальної Arduino Uno. Вона має більше ресурсів та можливостей, що робить її ідеальним вибором для проектів, які вимагають більшої обчислювальної потужності, більшої кількості пінів вводу-виводу або розширення пам'яті.

Основні характеристики Arduino Mega 2560:

- використовує мікроконтролер ATmega2560, який працює на частоті 16 МГц та має 256 Кбайт вбудованої флеш-пам'яті для зберігання програмного коду. Це надає достатньо ресурсів для великих і складних проектів;
- має 54 цифрових піна вводу-виводу (з них 14 можуть бути використані для ШІМ-сигналів) та 16 аналогових вхідних пінів. Це дозволяє підключати багато різних сенсорів, пристроїв та модулів до платформи Arduino;
- підтримує різні інтерфейси комунікації, такі як UART, I2C та SPI. Це дозволяє зв'язуватися з іншими пристроями, наприклад, з датчиками, дисплеями, модулями зв'язку та багатьма іншими;
- має вбудований USB-інтерфейс, що дозволяє підключати його безпосередньо до комп'ютера для програмування та зв'язку. Це робить процес



розробки та завантаження програмного коду на платформу Arduino зручним та простим;

- має роз'єми для підключення розширювальних плат (шитів) Arduino, таких як Ethernet Shield, Wi-Fi Shield, LCD-дисплеї, сенсорні екрани та інші. Це дозволяє розширити функціональність платформи Arduino Mega 2560 та виконувати більш складні завдання;

- може живитися від USB-порту комп'ютера або від зовнішнього джерела живлення. При потужності 7-12 вольт, плата може автоматично вибирати оптимальну напругу живлення;

- Arduino Mega 2560 підтримує широкий спектр бібліотек та компонентів, розроблених для платформ Arduino. Це дозволяє використовувати існуючі ресурси та код з інших проектів та спільноти Arduino;

- є потужною та розширеною платформою, яка надає багато можливостей для розробки різноманітних проектів. Вона особливо корисна для проектів, які вимагають більшої кількості пінів вводу-виводу, більшої обчислювальної потужності або розширення функціональності.

Також варто відзначити, що в програмі передбачено розширення системи розумного будинку, для цього в коді програми оголошені змінні, які при потребі можна використовувати.

### **3.5 Особливості організації віддаленого управління системою**

Розглянемо варіант підключення для управління системою віддалено. Спочатку користувачеві необхідно зареєструватися на сайті [noip.com](http://noip.com). Реєстрація потрібна для того, щоб лише ви могли редагувати адресу призначення веб-ресурсу. Після реєстрації на сайті слід перейти в розділ Dynamic DNS. У цьому розділі додається ір-адреса розробленого веб-ресурсу (рис. 3.6).

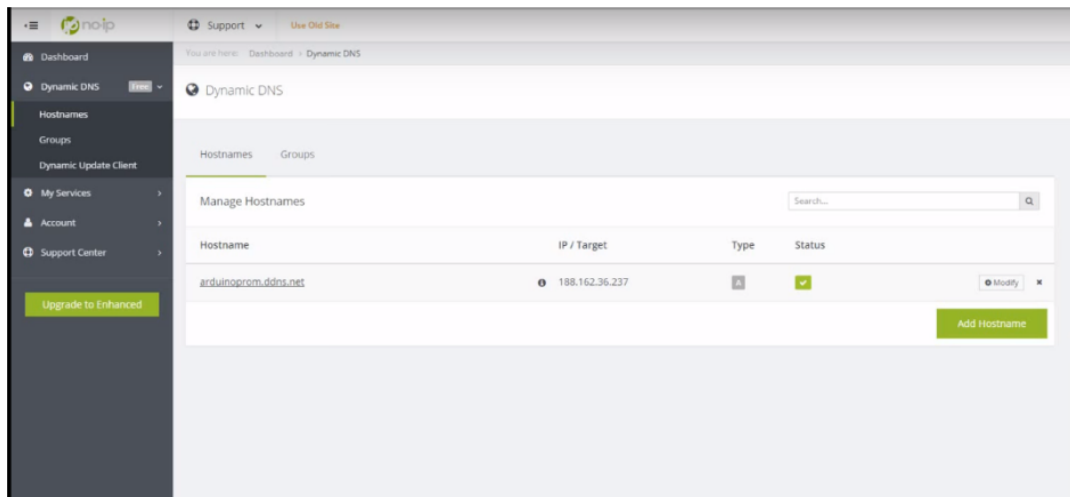


Рисунок 3.6 - Приклад додавання ір-адреси

Потім потрібно зайти в налаштування нашого маршрутизатора, для цього в веб-браузері вводиться 192.168.0.1. При цьому адреса налаштувань маршрутизатора може бути і іншою.

Це залежить від провайдера і моделі використовуваного маршрутизатора. У налаштуваннях маршрутизатора вибирається розділ "налаштування локальної мережі". У налаштуваннях локальної мережі необхідно вказати MAC і IP-адреси. Ці дані беруться з вихідного коду (ліст. 3.4).

#### Лістинг 3.4 – Налаштування адрес

```
byte ethernet_mac [] = {0x78, 0xAC, 0xC0, 0x90, 0x04, 0x30};
IPAddress ethernet_ip(192, 168, 10, 101);
byte ethernet_dns [] = {192, 168, 10, 1};
byte ethernet_gateway [] = {192, 168, 10, 1};
byte ethernet_subnet [] = {255, 255, 255, 0};
```

Після налаштування маршрутизатора системою можна керувати віддалено. Великим мінусом віддаленого керування є відсутність системи безпеки. Це означає, що будь-який користувач, який знає ір-адресу веб-ресурсу, зможе керувати створеною системою.

### 3.6 Відлагодження та тестування розробленого сервісу, порівняння результатів роботи з існуючими аналогами

Після збірки макету системи "розумного будинку" на платі Arduino mega 2560 та Ethernet шілді w5100 було проведено відлагодження програмного коду та наступне тестування даного макету. Під час тестування програмно-апаратного комплексу були виявлені наступні недоліки: при використанні віддаленого керування системою відсутній такий критерій як безпека. Це обґрунтовано тим, що сторінка керування не має аутентифікації і будь-хто, хто знає ір адресу обладнання, може втрутитися в роботу системи. Під час відлагодження був виявлений недолік, що для того, щоб побачити індикацію сигналізації, доводилося оновлювати сторінку вручну. Даний недолік був виправлений оновленням блоку сигналізації кожні 10 секунд:

Лістинг 3.5 – Оновлення блоку сигналізації кожні 10 секунд

```
_tspWebServer_client.println("<meta http-equiv=""Refresh""  
content=""10"" />");
```

## ВИСНОВКИ

Під час виконання дипломної роботи було проаналізовано найбільш відомі і популярні на сьогодні системи "розумного будинку", мобільні програмні застосунки, які традиційно використовуються для віддаленого керування системами "розумного будинку". Було розглянуто основні переваги використання Arduino, особливості розробки в середовище Arduino, вивчені різні підходи, що використовуються при створенні користувацьких інтерфейсів, а також методи і технології, необхідні для їх створення. Крім того, проведений аналіз плат Arduino для найточнішого вибору плати для проектуємої системи. Розроблені вимоги до користувацького інтерфейсу проектованого застосунку.

В роботі висунуто функціональні вимоги до програмного виробу, розроблено архітектуру системи, виконано створення базового варіанту (макету) системи "розумного будинку", призначеного для керування електронними системами освітлення і сигналізації будинку, проведена його налагодження та тестування.

Розроблений комплекс дозволяє користувачам віддалено керувати електронними системами будинку, а також відстежувати стан сигналізації.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Patrascu M. Integrating Services and Agents for Control and Monitoring: Managing Emergencies in Smart Buildings. Service Orientation in Holonic and MultiAgent Manufacturing and Robotics. / Patrascu., 2014. – 544 с.

2 Dickson B. How to prevent your IoT devices from being forced into botnet bondage / Dickson. – 2015. URL: <https://techcrunch.com/2016/08/16/how-to-prevent-your-iot-devices-from-being-forced-into-botnet-slavery> (дата звернення: 25.04.2023).

3 Power Load Event Detection and Classification Based on Edge Symbol Analysis and Support Vector Machine, 2014. URL: <https://www.hindawi.com/journals/acisc/2012/742461> (Last accessed: 25.04.2023).

4 An Overview of Home Automation Systems, 2017. URL: <https://ieeexplore.ieee.org/document/7791223> (Last accessed: 25.04.2023).

5 Granzer W. P. Security in Building Automation Systems / Wolfgang Praus Granzer. Munich: Apress, 2018. – 578 с.

6 Що таке розумний будинок? Все що потрібно знати про систему Розумний Будинок. URL: <https://bron.ua/article/schotake-rozumnij-budinok-vse-scho-potrбно-znati-pro-sistemu-rozumnij-dm/5> (дата звернення: 25.04.2023).

7 Розумне освітлення. URL: <https://milight.com.ua/ua/umное-osveshchenie> (дата звернення: 25.04.2023).

8 Технологія розумного будинку: як AI створює простір, комфортний для життя. URL: <https://www.everest.ua/tehnologiya-rozumnogo-budynku-yak-ai-stvoryuye-prostirkomfortnyj-dlya-zhyttya> (дата звернення: 25.04.2023).

9 Лучшие системы «Умный дом» в 2021 году. URL: <https://yanashla.com/luchshie-sistemy-umnyj-dom/#i-3>

10 Из чего собрать умный дом в 2020 году: от хаба и до лампочки. URL: <https://habr.com/ru/company/mvideo/blog/499706> (дата звернення: 25.04.2023).

11 Обзор контроллеров (аппаратной платформы) Arduino URL: [https://supereyes.com/articles/other/obzor\\_kontrollerov\\_apparatnoy\\_platfornoy\\_arduino](https://supereyes.com/articles/other/obzor_kontrollerov_apparatnoy_platfornoy_arduino) (дата звернення: 25.04.2023).

12 Котунова, Д. Г. Огляд DIY елементів для систем «Smart Home» / Д. Г. Котунова, О. М. Павловський // XIII Науково-практична конференція студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування», 13-14 травня 2020 р., м. Київ, Україна : збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2020. – С. 35–38.

13 Моніт Я.В. Система «Розумний будинок» з відкритим програмним забезпеченням/ Я.В.Моніт // XIX науково-технічна конференція студентів та молодих учених «Гіротехнології, навігація, керування рухом та конструювання авіаційно-космічної техніки», 15-16 лютого 2016 р. – К.: «Політехніка», 2016. – С. 43-44.

## ДОДАТОК А

## Лістинг А.1 - Код програми «Розумний будинок» с web інтерфейсом

```

#include <OneWire.h>
#include <SPI.h>
#include <Ethernet.h>
byte _d18x2x1Addr[8] = {0x28, 0xFF, 0x3B, 0x8B, 0x72, 0x15, 0x3, 0xA0};
OneWire _ow3(3);
byte ethernet_mac [] = {0x78, 0xAC, 0xC0, 0x90, 0x04, 0x30};
IPAddress ethernet_ip(192, 168, 10, 101);
byte ethernet_dns [] = {192, 168, 10, 1};
byte ethernet_gateway [] = {192, 168, 10, 1};
byte ethernet_subnet [] = {255, 255, 255, 0};
EthernetServer _tspWebServer(80);
EthernetClient _tspWebServer_client;
bool _gtv1; //Лампа 1
bool _gtv2; //Лампа 2
bool _gtv3; //Лампа 3
bool _gtv4; //Лампа 4
bool _gtv9; //Розетка 1
bool _gtv10; //Розетка 2
bool _gtv11; //Розетка 3
bool _gtv12; //Розетка 4
String _swi4;
String _swi5;
String _swi1;
String _swi2;
unsigned long _d18x2x1Tti = 0UL;
float _d18x2x1O = 0.00;
String _swi6;
String _swi7;
String _swi8;
int _swi9;
String _swi10;
String _swi11;
String _swi12;
int _swi13;
String _swi14;
String _swi15;
String _swi16;
String _swi17;
String _swi18;
String _swi19;
String _swi20;
bool _trgrt1 = 0;
bool _trgrt1I = 0;
bool _trgt1 = 0;
bool _trgt1I = 0;
bool _trgrt2 = 0;
bool _trgrt2I = 0;33
bool _trgt2 = 0;
bool _trgt2I = 0;
bool _trgrt3 = 0;
bool _trgrt3I = 0;
bool _trgt3 = 0;
bool _trgt3I = 0;
bool _trgrt4 = 0;
bool _trgrt4I = 0;

```

```
bool _trgt4 = 0;
bool _trgt4I = 0;
bool _trgrt5 = 0;
bool _trgrt5I = 0;
bool _trgt5 = 0;
bool _trgt5I = 0;
bool _trgrt6 = 0;
bool _trgrt6I = 0;
bool _trgt6 = 0;
bool _trgt6I = 0;
bool _trgrt7 = 0;
bool _trgrt7I = 0;
bool _trgt7 = 0;
bool _trgt7I = 0;
bool _trgrt8 = 0;
bool _trgrt8I = 0;
bool _trgt8 = 0;
bool _trgt8I = 0;
bool _trgrt9 = 0;
bool _trgrt9I = 0;
bool _trgt9 = 0;
bool _trgt9I = 0;
bool _trgrt10 = 0;
bool _trgrt10I = 0;
bool _trgt10 = 0;
bool _trgt10I = 0;
bool _trgrt11 = 0;
bool _trgrt11I = 0;
bool _trgt11 = 0;
bool _trgt11I = 0;
bool _trgrt12 = 0;
bool _trgrt12I = 0;
bool _trgt12 = 0;
bool _trgt12I = 0;
bool _trgrt13 = 0;
bool _trgrt13I = 0;
bool _trgt13 = 0;
bool _trgt13I = 0;
bool _trgrt14 = 0;
bool _trgrt14I = 0;
bool _trgt14 = 0;
bool _trgt14I = 0;
bool _WSP6_A1 = 0;
bool _WSP6_A2 = 0; 34
bool _WSP6_A3 = 0;
bool _WSP6_A4 = 0;
bool _WSP6_A5 = 0;
bool _WSP6_A6 = 0;
bool _WSP6_A7 = 0;
bool _WSP6_A8 = 0;
bool _WSP6_A9 = 0;
bool _WSP6_A10 = 0;
bool _WSP6_A11 = 0;
bool _WSP6_A12 = 0;
bool _WSP6_A13 = 0;
bool _WSP6_A14 = 0;
bool _bounseInputD49S = 0;
bool _bounseInputD49O = 0;
unsigned long _bounseInputD49P = 0UL;
bool _bounseInputD46S = 0;
bool _bounseInputD46O = 0;
unsigned long _bounseInputD46P = 0UL;
bool _bounseInputD48S = 0;
bool _bounseInputD48O = 0;
```



```

unsigned long _bounceInputD48P = 0UL;
bool _bounceInputD47S = 0;
bool _bounceInputD47O = 0;
unsigned long _bounceInputD47P = 0UL;
void setup()
{
  pinMode(46, INPUT);
  digitalWrite(46, HIGH);
  pinMode(47, INPUT);
  digitalWrite(47, HIGH);
  pinMode(48, INPUT);
  digitalWrite(48, HIGH);
  pinMode(49, INPUT);
  digitalWrite(49, HIGH);
  pinMode(22, OUTPUT);
  pinMode(23, OUTPUT);
  pinMode(24, OUTPUT);
  pinMode(25, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(27, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(29, OUTPUT);
  _bounceInputD49O = digitalRead(49);
  _bounceInputD46O = digitalRead(46);
  _bounceInputD47O = digitalRead(47);
  _bounceInputD48O = digitalRead(48);
  Ethernet.begin(ethernet_mac, ethernet_ip, ethernet_dns, ethernet_gateway,
ethernet_subnet);
  delay(1000);
  _tspWebServer.begin();
}35
void loop()
{
  _tspWebServer_client = _tspWebServer.available();
  if (_tspWebServer_client) {
    boolean _WSCLineIsBlank = true;
    String _WSCRequest = "";
    bool _WSCIsFirsLine = 1;
    bool _WSCIsParse = 0;
    int _WSCPPageNumber = 0;
    while (_tspWebServer_client.connected())
    { while (_tspWebServer_client.available())
    { char _tempWebServerChar = _tspWebServer_client.read();
    if (_WSCIsFirsLine) {
      _WSCRequest += _tempWebServerChar;
    }
    if (_tempWebServerChar == '\n' && _WSCLineIsBlank) {
      _sendWebServerPage(_WSCPPageNumber);
      break;
    }
    if (_tempWebServerChar == '\n') {
      _WSCLineIsBlank = true;
      _WSCIsFirsLine = 0;
      if (!_WSCIsParse) {
        _WSCPPageNumber = _parseWebServerReqest(_WSCRequest);
        _WSCIsParse = 1;
      }
    } else if (_tempWebServerChar != '\r') {
      _WSCLineIsBlank = false;
    }
  }
}
}
}
}
}
bool _bounceInputTmpD49 = (digitalRead (49));
if (_bounceInputD49S)

```

```

{
if (millis() >= (_bounceInputD49P + 40))
{
_bounceInputD49O = _bounceInputTmpD49;
_bounceInputD49S = 0;
}
}
else
{
if (_bounceInputTmpD49 != _bounceInputD49O )
{
_bounceInputD49S = 1;
_bounceInputD49P = millis();
}
}
bool _bounceInputTmpD46 = (digitalRead (46));36
if (_bounceInputD46S)
{
if (millis() >= (_bounceInputD46P + 40))
{
_bounceInputD46O = _bounceInputTmpD46;
_bounceInputD46S = 0;
}
}
else
{
if (_bounceInputTmpD46 != _bounceInputD46O )
{
_bounceInputD46S = 1;
_bounceInputD46P = millis();
}
}
bool _bounceInputTmpD48 = (digitalRead (48));
if (_bounceInputD48S)
{
if (millis() >= (_bounceInputD48P + 40))
{
_bounceInputD48O = _bounceInputTmpD48;
_bounceInputD48S = 0;
}
}
else
{
if (_bounceInputTmpD48 != _bounceInputD48O )
{
_bounceInputD48S = 1;
_bounceInputD48P = millis();
}
}
bool _bounceInputTmpD47 = (digitalRead (47));
if (_bounceInputD47S)
{
if (millis() >= (_bounceInputD47P + 40))
{
_bounceInputD47O = _bounceInputTmpD47;
_bounceInputD47S = 0;
}
}
else
{
if (_bounceInputTmpD47 != _bounceInputD47O )
{
_bounceInputD47S = 1;
_bounceInputD47P = millis();
}
}
}

```

```

}
}37
if (!(_bounseInputD460))
{
_swi4 = String("http:// provuz22.ru /arduino/pozhar1.png");
}
else
{
_swi4 = String("http:// provuz22.ru /arduino/pozhar0.png");
}
if (!(_bounseInputD470))
{
_swi5 = String("http:// provuz22.ru /arduino/signalizaciy1.png");
}
else
{
_swi5 = String("http:// provuz22.ru /arduino/signalizaciya0.png");
}
if (!(_bounseInputD480))
{
_swi1 = String("http:// provuz22.ru /arduino/voda1.png");
}
else
{
_swi1 = String("http:// provuz22.ru /arduino/voda0.png");
}
if (!(_bounseInputD490))
{
_swi2 = String("http:// provuz22.ru /arduino/zadimlenie1.png");
}
else
{
_swi2 = String("http:// provuz22.ru /arduino/zadimlenie0.png");
}
if (_isTimer(_d18x2x1Tti, 3000)) {
_d18x2x1Tti = millis();
_d18x2x1O = _readDS18_ow3(_d18x2x1Addr, 0);
}
if (_gtv13)
{
_swi6 = String("http:// provuz22.ru /arduino/thermostat1.png");
}
else
{
_swi6 = String("http:// provuz22.ru /arduino/thermostat0.png");
}
if (_gtv12)
{
_swi7 = String("http:// provuz22.ru /arduino/rozetka1.png");38
}
else
{
_swi7 = String("http:// provuz22.ru /arduino/rozetka0.png");
}
if (_gtv11)
{
_swi8 = String("http:// provuz22.ru /arduino/rozetka1.png");
}
else
{
_swi8 = String("http:// provuz22.ru /arduino/rozetka0.png");
}
if (_gtv10)
{

```

```

_swil0 = String("http:// provuz22.ru /arduino/rozetkal.png");
}
else
{
_swil0 = String("http:// provuz22.ru /arduino/rozetka0.png");
}
if (_gtv9)
{
_swil1 = String("http:// provuz22.ru /arduino/rozetkal.png");
}
else
{
_swil1 = String("http:// provuz22.ru /arduino/rozetka0.png");
}
if (_gtv4)
{
_swil7 = String("http:// provuz22.ru /arduino/lampochkal.png");
}
else
{
_swil7 = String("http:// provuz22.ru /arduino/lampochka0.png");
}
if (_gtv3)
{
_swil8 = String("http:// provuz22.ru /arduino/lampochkal.png");
}
else
{
_swil8 = String("http:// provuz22.ru /arduino/lampochka0.png");
}
if (_gtv2)
{
_swil9 = String("http:// provuz22.ru /arduino/lampochkal.png");
}
else
{39
_swil9 = String("http:// provuz22.ru /arduino/lampochka0.png");
}
if (_gtv1)
{
_swil20 = String("http:// provuz22.ru /arduino/lampochkal.png");
}
else
{
_swil20 = String("http://provuz22.ru/arduino/lampochka0.png");
}
if (_WSP6_A13) {
if (_trgrt1I) {
_trgrt1 = 0;
} else {
_trgrt1 = 1;
_trgrt1I = 1;
}
} else {
_trgrt1 = 0;
_trgrt1I = 0;
};
bool _tmp1 = _trgrt1;
if (_tmp1) {
if (! _trgrt1I) _trgrt1 = ! _trgrt1;
}
_trgrt1I = _tmp1;
if (_WSP6_A1) {
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;
if (_WSP6_A2) {
if (_trgrt3I) {
_trgrt3 = 0;
} else {
_trgrt3 = 1;
_trgrt3I = 1;
}
} else {
_trgrt3 = 0;
_trgrt3I = 0;40
};
bool _tmp3 = _trgrt3;
if (_tmp3) {
if (! _trgt3I) _trgt3 = ! _trgt3;
}
_trgt3I = _tmp3;
if (_WSP6_A3) {
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;
if (_WSP6_A4) {
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;
if (_WSP6_A5) {
if (_trgrt6I) {

```

```

_trgrt6 = 0;
} else {
_trgrt6 = 1;
_trgrt6I = 1;
}
} else {
_trgrt6 = 0;
_trgrt6I = 0;
};
bool _tmp6 = _trgrt6;
if (_tmp6) {
if (! _trgrt6I) _trgt6 = ! _trgt6;41
}
_trgt6I = _tmp6;
if (_WSP6_A6) {
if (_trgrt7I) {
_trgrt7 = 0;
} else {
_trgrt7 = 1;
_trgrt7I = 1;
}
} else {
_trgrt7 = 0;
_trgrt7I = 0;
};
bool _tmp7 = _trgrt7;
if (_tmp7) {
if (! _trgrt7I) _trgt7 = ! _trgt7;
}
_trgt7I = _tmp7;
if (_WSP6_A7) {
if (_trgrt8I) {
_trgrt8 = 0;
} else {
_trgrt8 = 1;
_trgrt8I = 1;
}
} else {
_trgrt8 = 0;
_trgrt8I = 0;
};
bool _tmp8 = _trgrt8;
if (_tmp8) {
if (! _trgrt8I) _trgt8 = ! _trgt8;
}
_trgt8I = _tmp8;
if (_WSP6_A8) {
if (_trgrt9I) {
_trgrt9 = 0;
} else {
_trgrt9 = 1;
_trgrt9I = 1;
}
} else {
_trgrt9 = 0;
_trgrt9I = 0;
};
bool _tmp9 = _trgrt9;
if (_tmp9) {
if (! _trgrt9I) _trgt9 = ! _trgt9;
}
_trgt9I = _tmp9;
if (_WSP6_A9) {
if (_trgrt10I) {42

```

```

_trgrt10 = 0;
} else {
_trgrt10 = 1;
_trgrt10I = 1;
}
} else {
_trgrt10 = 0;
_trgrt10I = 0;
};
bool _tmp10 = _trgrt10;
if (_tmp10) {
if (! _trgt10I) _trgt10 = ! _trgt10;
}
_trgt10I = _tmp10;
if (_WSP6_A10) {
if (_trgrt11I) {
_trgrt11 = 0;
} else {
_trgrt11 = 1;
_trgrt11I = 1;
}
} else {
_trgrt11 = 0;
_trgrt11I = 0;
};
bool _tmp11 = _trgrt11;
if (_tmp11) {
if (! _trgt11I) _trgt11 = ! _trgt11;
}
_trgt11I = _tmp11;
if (_WSP6_A11) {
if (_trgrt12I) {
_trgrt12 = 0;
} else {
_trgrt12 = 1;
_trgrt12I = 1;
}
} else {
_trgrt12 = 0;
_trgrt12I = 0;
};
bool _tmp12 = _trgrt12;
if (_tmp12) {
if (! _trgt12I) _trgt12 = ! _trgt12;
}
_trgt12I = _tmp12;
if (_WSP6_A12) {
if (_trgrt13I) {
_trgrt13 = 0;
} else {
_trgrt13 = 1;
_trgrt13I = 1;43
}
} else {
_trgrt13 = 0;
_trgrt13I = 0;
};
bool _tmp13 = _trgrt13;
if (_tmp13) {
if (! _trgt13I) _trgt13 = ! _trgt13;
}
_trgt13I = _tmp13;
if (_WSP6_A14) {
if (_trgrt14I) {

```

```

    _trgrt14 = 0;
} else {
    _trgrt14 = 1;
    _trgrt14I = 1;
}
} else {
    _trgrt14 = 0;
    _trgrt14I = 0;
};
bool _tmp14 = _trgrt14;
if (_tmp14) {
if (! _trgrt14I) _trgrt14 = ! _trgrt14;
}
    _trgrt14I = _tmp14;
    _gtv14 = _trgt2;
    _gtv13 = _trgt3;
    _gtv12 = _trgt4;
    _gtv11 = _trgt5;
    _gtv10 = _trgt6;
    _gtv9 = _trgt7;
    _gtv4 = _trgt12;
    _gtv3 = _trgt13;
    _gtv2 = _trgt1;
    _gtv1 = _trgt14;
    digitalWrite(34, _gtv13);
    digitalWrite(33, _gtv12);
    digitalWrite(35, _gtv14);
    digitalWrite(32, _gtv11);
    digitalWrite(31, _gtv10);
    digitalWrite(30, _gtv9);
    digitalWrite(25, _gtv4);
    digitalWrite(24, _gtv3);
    digitalWrite(23, _gtv2);
    digitalWrite(22, _gtv1);44
}
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));
    }
}
float _convertDS18x2xData(byte type_s, byte data[12])
{
    int16_t raw = (data[1] << 8) | data[0];
    if (type_s)
    {
        raw = raw << 3;
        if (data[7] == 0x10) {
            raw = (raw & 0xFFF0) + 12 - data[6];
        }
    }
    else
    {
        byte cfg = (data[4] & 0x60);
        if (cfg == 0x00) raw = raw & ~7; else if (cfg == 0x20) raw = raw & ~3; else
if (cfg == 0x40)
        raw = raw & ~1;
    }
    return (float)raw / 16.0;
}

```



```

float _readDS18_ow3(byte addr[8], byte type_s)
{ byte data[12];
byte i;
_ow3.reset();
_ow3.select(addr);
_ow3.write(0xBE);
for ( i = 0; i < 9; i++) {
data[i] = _ow3.read();
}
_ow3.reset();
_ow3.select(addr);
_ow3.write(0x44, 1);
return _convertDS18x2xData(type_s, data);
} void _sendWebServerPage(int sendPageNumber)
{ _tspWebServer_client.println("HTTP/1.1 200 OK");
_tspWebServer_client.println("Connection: close");
_tspWebServer_client.println();
_tspWebServer_client.println("<!DOCTYPE HTML PUBLIC \"\"-//W3C//DTD HTML 4.01
Transitional//EN\"\">");45
_tspWebServer_client.println("<html><head>");
_tspWebServer_client.println("<META content=\"\"text/html; charset=utf-8\"\"
httpequiv=\"\"Content-Type\"\">");
if (sendPageNumber == 1) {
_sendWebServerPage1();
}
if (sendPageNumber == 2) {
_sendWebServerPage2();
}
if (sendPageNumber == 3) {
_sendWebServerPage3();
}
if (sendPageNumber == 4) {
_sendWebServerPage4();
}
if (sendPageNumber == 5) {
_sendWebServerPage5();
}
if (sendPageNumber == 6) {
_sendWebServerPage6();
}
_tspWebServer_client.println("</body></html>");
delay(1); _tspWebServer_client.stop();
}
int _parseWebServerRequest(String requestAddress)
{
int index;
int result = 0;
index = requestAddress.indexOf("/");
requestAddress = _stringWithoutCharWithIndex(requestAddress, 0, (index));
index = requestAddress.indexOf(" ");
requestAddress = _stringWithoutCharWithIndex(requestAddress, index,
(requestAddress.length() -
index));
if (requestAddress == "signalizacii") {
result = 1;
}
if (requestAddress == "acp") {
result = 2;
}
if (requestAddress == "termostat") {
result = 3;
}
if (requestAddress == "temperatura") {
result = 4;
}
}

```

```

}
if (requestAddress == "") {
result = 5;
}
if (requestAddress == "14") {
_WSP6_A1 = 1;
result = 6;46
} else {
_WSP6_A1 = 0;
}
if (requestAddress == "13") {
_WSP6_A2 = 1;
result = 6;
} else {
_WSP6_A2 = 0;
}
if (requestAddress == "12") {
_WSP6_A3 = 1;
result = 6;
} else {
_WSP6_A3 = 0;
}
if (requestAddress == "11") {
_WSP6_A4 = 1;
result = 6;
} else {
_WSP6_A4 = 0;
}
if (requestAddress == "10") {
_WSP6_A5 = 1;
result = 6;
} else {
_WSP6_A5 = 0;
}
if (requestAddress == "9") {
_WSP6_A6 = 1;
result = 6;
} else {
_WSP6_A6 = 0;
}
if (requestAddress == "8") {
_WSP6_A7 = 1;
result = 6;
} else {
_WSP6_A7 = 0;
}
if (requestAddress == "7") {
_WSP6_A8 = 1;
result = 6;
} else {
_WSP6_A8 = 0;
}
if (requestAddress == "6") {
_WSP6_A9 = 1;
result = 6;
} else {
_WSP6_A9 = 0;
}
if (requestAddress == "5") {47
_WSP6_A10 = 1;
result = 6;
} else {
_WSP6_A10 = 0;
}
}

```

```

if (requestAddress == "4") {
    _WSP6_A11 = 1;
    result = 6;
} else {
    _WSP6_A11 = 0;
}
if (requestAddress == "3") {
    _WSP6_A12 = 1;
    result = 6;
} else {
    _WSP6_A12 = 0;
}
if (requestAddress == "2") {
    _WSP6_A13 = 1;
    result = 6;
} else {
    _WSP6_A13 = 0;
}
if (requestAddress == "1") {
    _WSP6_A14 = 1;
    result = 6;
} else {
    _WSP6_A14 = 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 _sendWebServerPagel(void)
{
    _tspWebServer_client.println("<meta http-equiv=""Refresh"" content=""10""
/>");
    _tspWebServer_client.println("<table width=""100%"" cellspacing=""0""
cellpadding=""10"">");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("<tr>");
    _tspWebServer_client.println("<td align=""center"">");48
    _tspWebServer_client.println("");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("<p><a href=""><img src="">");
    _tspWebServer_client.println(_swi4);
    _tspWebServer_client.println("</a></p>");
    _tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Пожар</font></p>");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("</td>");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("<td align=""center"">");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("");
    _tspWebServer_client.println("<p><a href=""><img src="">");
    _tspWebServer_client.println(_swi5);
    _tspWebServer_client.println("</a></p>");

```

```

_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Проникновение</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""><img src="">");
_tspWebServer_client.println(_swi1);
_tspWebServer_client.println("</a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Потоп</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""><img src="">");
_tspWebServer_client.println(_swi2);
_tspWebServer_client.println("</a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Задымление</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</tr>");
};
_tspWebServer_client.println("</table>");
} void _sendWebServerPage2(void)
{
_tspWebServer_client.println("<meta http-equiv=""Refresh"" content=""3""
/>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""5""
face=""Open
Sans"">");
_tspWebServer_client.println((( analogRead (5))) * (0.0049));
_tspWebServer_client.println("</font></p>");
} void _sendWebServerPage3(void)
{
_tspWebServer_client.println("<meta http-equiv=""Refresh"" content=""3""
/>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""5""
face=""Open
Sans"">");
_tspWebServer_client.println((_d18x2x10));
_tspWebServer_client.println("</font></p>");
} void _sendWebServerPage4(void)
{
_tspWebServer_client.println("<meta http-equiv=""Refresh"" content=""3""
/>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""5""
face=""Open
Sans"">");
_tspWebServer_client.println((_d18x2x10));
_tspWebServer_client.println("</font></p>");
} void _sendWebServerPage5(void)

```

```

    {
        _tspWebServer_client.println("<table width=""100%"" height=""100%""
cellspacing=""0""
        cellpadding=""0"" border=""0"" bgcolor=""1d2228"">");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("<td>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<table width=""100%"" align=""center""
border=""1""
        cellspacing=""0"" cellpadding=""20"">");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<td bgcolor=""212121"" width=""100%""
valign=""top""
        colspan=""2"">");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<table width=""100%"" bgcolor=""212121""
cellspacing=""0""
        cellpadding=""0"">");
        _tspWebServer_client.println("<tr>");50
        _tspWebServer_client.println("<td bgcolor=""212121"" width=""12%""
valign=""top"">");
        _tspWebServer_client.println("</td>");
        _tspWebServer_client.println("<td bgcolor=""212121"" width=""48%""
valign=""top"">");
        _tspWebServer_client.println("</td>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _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("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<td width=""60%"" valign=""top""
colspan=""2"">");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<p><font color=""FFFFFF"" size=""5""
face=""Open
        Sans"">Управление освещением</font></p>");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<table width=""100%"" cellspacing=""0""
cellpadding=""20"">");
        _tspWebServer_client.println("<tr>");
        _tspWebServer_client.println("<td align=""center"">");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("");
        _tspWebServer_client.println("<p><a href=""/1""><img src="">");
        _tspWebServer_client.println(_swi20);
        _tspWebServer_client.println("</a></p>");
        _tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
        Sans"">Лампа 1</font></p>");
    }

```

```

_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""/2""><img src=""");
_tspWebServer_client.println(_swil9);
_tspWebServer_client.println("""></a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Лампа 2</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");51
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""/3""><img src=""");
_tspWebServer_client.println(_swil8);
_tspWebServer_client.println("""></a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Лампа 3</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""/4""><img src=""");
_tspWebServer_client.println(_swil7);
_tspWebServer_client.println("""></a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Лампа 4</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</tr>");
_tspWebServer_client.println("</table>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<hr>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""5""
face=""Open
Sans"">Управление нагрузкой</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<table width=""100%"" cellspacing=""0""
cellpadding=""20"">");
_tspWebServer_client.println("<tr>");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""/9""><img src=""");
_tspWebServer_client.println(_swil1);
_tspWebServer_client.println("""></a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open

```

```

Sans"">Розетка 1</font></p>");
_tspWebServer_client.println("");52
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""/10""><img src=""");
_tspWebServer_client.println(_swi10);
_tspWebServer_client.println("</a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Розетка 2</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""/11""><img src=""");
_tspWebServer_client.println(_swi8);
_tspWebServer_client.println("</a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Розетка 3</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td align=""center"">");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><a href=""/12""><img src=""");
_tspWebServer_client.println(_swi7);
_tspWebServer_client.println("</a></p>");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""3""
face=""Open
Sans"">Розетка 4</font></p>");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</tr>");
_tspWebServer_client.println("</table>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("</td>");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<td width=""40%"" valign=""top"">");
_tspWebServer_client.println("");53
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("");
_tspWebServer_client.println("<p><font color=""FFFFFF"" size=""4""
face=""Open
Sans"">Сигнализации</font></p>");
_tspWebServer_client.println("<iframe src=""signalizacii"" width=""420""
height=""120""
scrolling=""no"" frameborder=""no""></iframe>");


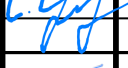


```

```
_tspWebServer_client.println("");
_tspWebServer_client.println("");
// _tspWebServer_client.println("<p align=""center""><a href=""/14""><img
src=""");
} void _sendWebServerPage6(void)
{
_tspWebServer_client.println("<meta http-equiv=""refresh""
content=""0;URL=http://192.168.10.101"">");
}
```



# Програмно-апаратні комплекси «розумних будинків»

- Apple HomeKit;
- Amazon Alexa Smart Home;
- Samsung SmartThings;
- Xiaomi Smart Home;
- VIMAR By-me;
- система «розумний будинок» be smart MY HOME Legrand & BTicino ;
- система «розумний будинок» iNels від компанії Elko ep.

					13.02070849.51901 ПЛ1			
					Розробка системи розумного будинку Програмно-апаратні комплекси «розумних будинків»	Лист.	Маса	Масшт.
Зам.	Лист.	№ док.м.	Підп.	Дата				
Розроб.		Богунів Р.В.						
Перев.		Грушко С.С.						
Т.контр.								
Н.контр.		Щербак Н.В.						
Затв.		Кудерметов Р.К.						
						Лист 1	Листів 1	
						НУ «Запорізька політехніка» КНТ-519		

# Функціональні вимоги до програмного виробу

<p>ОСВІТЛЕННЯ</p> <p>○ ○ ○ ○</p> <p>Номери ламп / назви кімнат</p>	<p>СИГНАЛІЗАЦІЯ</p> <p>□ □ □ □</p>
<p>НАВАНТАЖЕННЯ</p> <p>□ □ □ □</p> <p>Номери розеток / назви кімнат</p>	

Рисунок - Макет головної сторінки web-ресурса

<p>ОСВІТЛЕННЯ</p> <p>○ ● ● ○</p> <p>Номери ламп / назви кімнат</p>	<p>СИГНАЛІЗАЦІЯ</p> <p>□ □ ■ ■</p>
<p>НАВАНТАЖЕННЯ</p> <p>■ □ □ □</p> <p>Номери розеток / назви кімнат</p>	

Рисунок – Макет сторінки веб-ресурсу з індикацією

					<b>13.02070849.51901 ПЛ2</b>		
					Розробка системи розумного будинку Функціональні вимоги до програмного виробу		
<i>Зам.</i>	<i>Лист.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>			
<i>Розроб.</i>		Богунів Р.В.	<i>[Підпис]</i>				
<i>Перев.</i>		Грушко С.С.	<i>[Підпис]</i>				
<i>Т.контр.</i>							
<i>Н.контр.</i>		Щербак Н.В.	<i>[Підпис]</i>				
<i>Затв.</i>		Кудерметов Р.К.	<i>[Підпис]</i>				
					Лист 1      Листів 1		
					НУ «Запорізька політехніка» КНТ-519		

# Архітектура системи

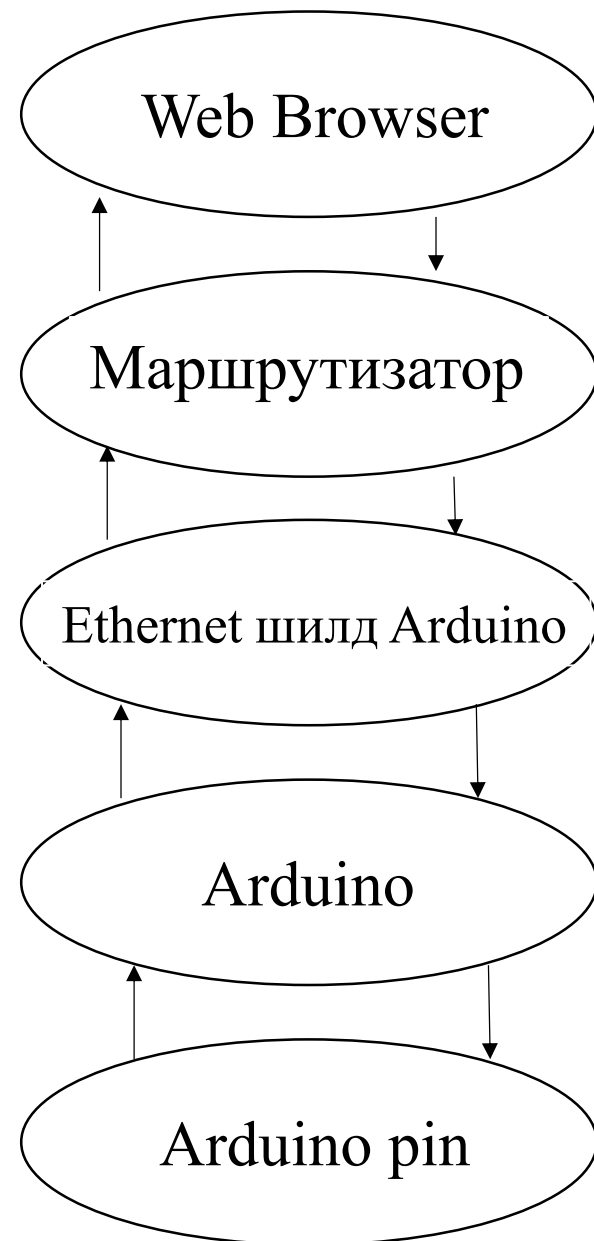


Рисунок – Схема роботи системи

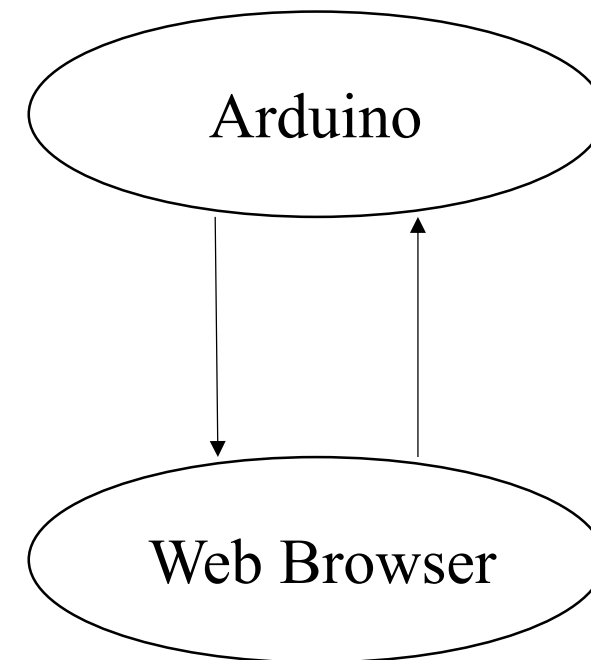


Рисунок – Архітектура системи

					<b>13.02070849.51901 ПЛЗ</b>		
					Розробка системи розумного будинку Архітектура системи		
Зам.	Лист.	№ докум.	Підп.	Дата			
Розроб.		Богунюв Р.В.	<i>[Signature]</i>				
Перев.		Грушко С.С.	<i>[Signature]</i>				
Т.контр.							
Н.контр.		Щербак Н.В.	<i>[Signature]</i>				
Затв.		Кудерметов Р.К.	<i>[Signature]</i>				
					Лист 1      Листів 1 НУ «Запорізька політехніка» КНТ-519		

# Особливості організації віддаленого управління системою

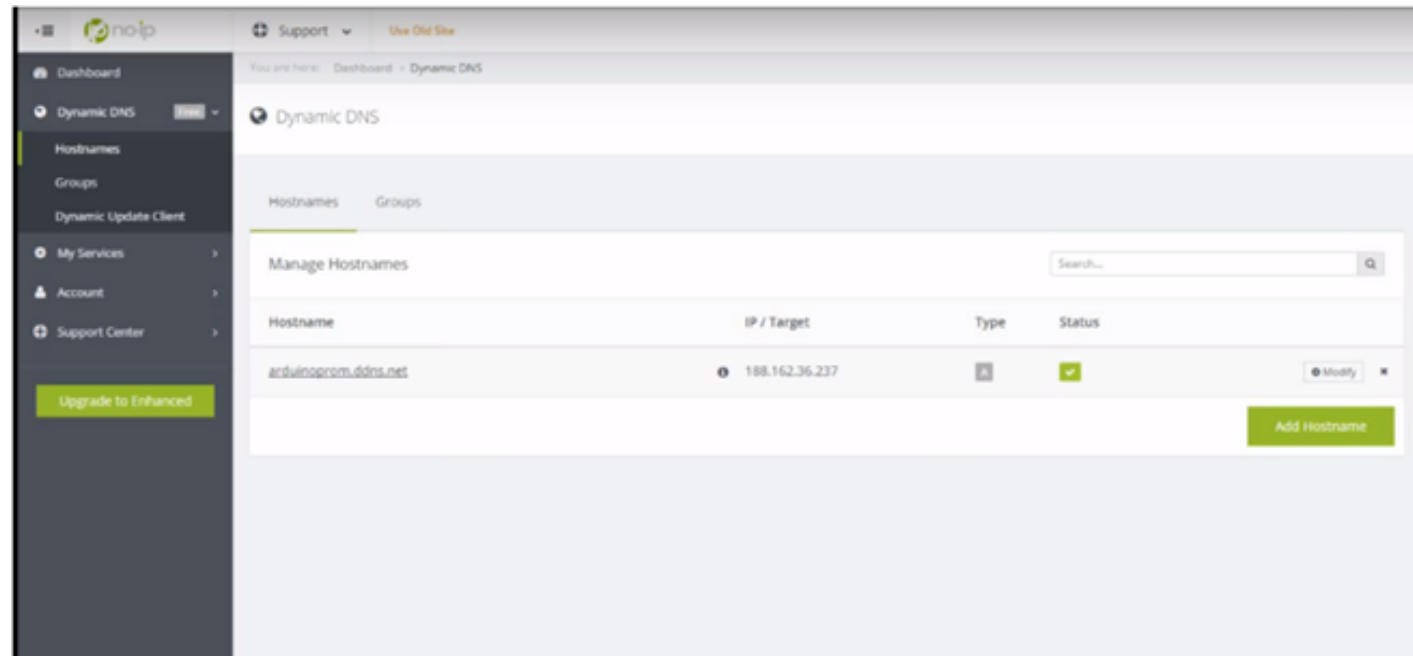


Рисунок - Приклад додавання ір-адреси

## Лістинг - Налаштування адрес

```
byte ethernet_mac [] = {0x78, 0xAC, 0xC0, 0x90, 0x04, 0x30};
IPAddress ethernet_ip(192, 168, 10, 101);
byte ethernet_dns [] = {192, 168, 10, 1};
byte ethernet_gateway [] = {192, 168, 10, 1};
byte ethernet_subnet [] = {255, 255, 255, 0};
```

					<b>13.02070849.51901 ПЛ4</b>		
					Розробка системи розумного будинку Особливості організації віддаленого управління системою		
<i>Зам.</i>	<i>Лист.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>			
<i>Розроб.</i>		Богунов Р.В.	<i>[Signature]</i>				
<i>Перев.</i>		Грушко С.С.	<i>[Signature]</i>				
<i>Т.контр.</i>					<i>Листів 1</i>		<i>Листів 1</i>
<i>Н.контр.</i>		Щербак Н.В.	<i>[Signature]</i>		НУ «Запорізька політехніка» КНТ-519		
<i>Затв.</i>		Кудерметов Р.К.	<i>[Signature]</i>				