

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки


Пояснювальна записка

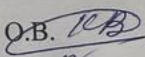
до бакалаврської дипломної роботи


на тему: «Розумний дім з web-інтерфейсом»

Виконав: студент 4 курсу, групи 2КІ-186

123 – «Комп'ютерна інженерія»

Гаврилюк О.С. 

Керівник роботи к.т.н., доц. Кадук О.В. 

Рецензент к.т.н., доц. Карпинець В.В. 

Допущено до захисту
д.т.н., проф. Азаров О.Д.

“ “ 2022 р.

Вінниця 2022

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

Освітньо-кваліфікаційний рівень — бакалавр

Спеціальність — 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри

обчислювальної техніки

проф., д.т.н. О.Д.Азаров

« » 2022 р.

З А В Д А Н Н Я

НА БАКАЛАВСЬКУ ДИПЛОМНУ РОБОТУ

студенту Гаврилюку Олександрю Сергійовичу

Тема бакалаврської дипломної роботи «Розумний дім з web-інтерфейсом», керівник роботи Кадук Олександр Володимирович, к.т.н., доц., затверджені наказом вищого навчального закладу від 24.03.22 року №66

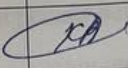
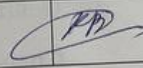
2 Строк подання студентом роботи: 13.06.2022р.

3 Вихідні дані до роботи: Технічні параметри систем «розумних будинків», технічний опис РНР, технічна документація на електронні компоненти.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз застосування сучасних веб-сервісів для надання побутових послуг, вибір засобів для розробки веб-сервісу, опис веб-сервісу, висновки, перелік джерел посилання.

5 Дата видачі завдання: 10.02.2022


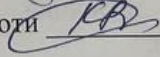
6 Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Кадук О.В., к.т.н., доцент кафедри ОТ		

7 Календарний план наведений в таблиці 1.

Таблиця 1 — Календарний план

з/п	Назва етапів виконання бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задач роботи	14.02.22	
2	Огляд інформаційних джерел	15.02-28.02.22	
3	Аналіз застосування сучасних веб-сервісів для надання побутових послуг	01.03-14.03.22	
4	Вибір засобів для розробки веб-сервісу	15.03-28.03.22	
5	Опис веб-сервісу	29.03-11.04.22	
6	Оформлення пояснювальної записки та ілюстративного матеріалу	12.04-25.04.22	
7	Аналіз виконання роботи. Висновки. Додатки	26.04-09.05.22	
8	Перевірка якості виконання бакалаврської роботи та усунення недоліків	17.05.22	

Студент  О.С. Гаврилюк
 Керівник роботи  О.В. Кадук

АНОТАЦІЯ

В результаті роботи вивчені особливості комплексу апаратно-програмних засобів Arduino, отримані та покращені знання в галузі розробки програмних додатків під управлінням Arduino.

ANNOTATION

As a result, the robots developed the features of the Arduino hardware and software complex, removed and improved knowledge in the gallery of the software add-ons for the Arduino controls.

ЗМІСТ

ВСТУП	7
1 ОГЛЯД І АНАЛІЗ СУЧАСНИХ СИСТЕМ "РОЗУМНОГО БУДИНКУ"	10
1.1 Аналіз плат Arduino.....	10
1.2 Програмно-апаратні комплекси, що реалізуються в системах розумного будинку.....	12
1.3 Огляд традиційних підходів до проектування інтерфейсів користувача.....	22
1.4 Аналіз традиційних способів реалізації управління системами будинку.....	23
1.5 Обґрунтування вибору плати Arduino.....	24
1.6 Висновки.....	25
2 ПРОЕКТУВАННЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»	26
2.1 Постановка задачі.....	26
2.2 Функціональні вимоги до програмного виробу	27
2.3 Архітектура системи	28
2.4 Структура роботи програмно-апаратного комплексу.....	28
2.5 Апаратна реалізація.....	30
2.6 Висновки.....	35
2 РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ «РОЗУМНИЙ БУДИНОК»	36
3.1 Особливості використання системи.....	36
3.1.1 Завантаження кнопок перемикачів.....	37
3.1.2 Реалізація веб-сервера.....	38
3.1.3 Організація індикації сигналізації на web ресурсі.....	39
3.2 Особливості організації віддаленого керування системою.....	39

					08-23.БДР.022.00.000 ПЗ							
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розумний дім з web-інтерфейсом Пояснювальна записка			<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>		
<i>Розроб.</i>		<i>Гаврилюк О.С</i>								6	84	
<i>Перевір.</i>		<i>Кадук О.В.</i>										
<i>Реценз.</i>		<i>Карпінєць В.В.</i>										
<i>Н. Контр.</i>		<i>Швець С. І.</i>										
<i>Затверд.</i>		<i>Азаров О.Д.</i>			ВНТУ, гр. 2КІ – 18б							

3.3 Налагодження та тестування розробленого сервісу, порівняння результатів роботи з наявними аналогами	40
3.4 Можливість розширення функціональних можливостей.....	40
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	46
ДОДАТОК А Технічне завдання.....	48
ДОДАТОК Б Сторінка web-сервісу з індикацією.....	51
ДОДАТОК В Приклад додавання IP-адреси.....	52
ДОДАТОК Г Схема з'єднання елементів.....	53
ДОДАТОК Д Код програми «Розумний дім» з web інтерфейсом	54
ДОДАТОК Е Код для платформи Arduino.....	78
ДОДАТОК Є Протокол перевірки на плагіат.....	86

ВСТУП

Системи типу «Розумний дім» останнім часом стали звичайною справою навіть для жителів невеликих міст, і тим більше для столиць та мегаполісів. Управління енергетичними системами будинку, забезпечення безпеки, контроль за витратою комунальних ресурсів, економія електроенергії — всі ці причини забезпечують актуальність «розумних» технологій. В даний час ні для кого не секрет, що за подібними технологіями майбутнє, і вони розвиватимуться. Актуальним залишається питання про те, якою стане стратегія розвитку систем подібного роду, які технології будуть використовуватися для їх розробки та експлуатації.

«Розумний дім» — це одна з пріоритетних галузей розвитку Інтернету, в якій працює велика кількість людей, компаній та структур. Всі вони в даний час дуже активно включають до своїх рішень різноманітні функції управління споживчими пристроями, а також реалізують обмін даними між цими пристроями. Наприклад, оператори, що надають послуги зв'язку, мають великі шанси завоювати ринок розумного будинку в Україні, оскільки вони здатні забезпечити комплексний підхід до організації роботи в даному напрямку: забезпечують канал зв'язку, наявність спеціального обладнання та гідний рівень обслуговування. Використання сучасних каналів зв'язку, датчиків, включених до складу побутових пристроїв та обладнання, а також наявність програмного забезпечення, що забезпечує узгодження всіх компонентів системи та управління ними, дозволяє створити умови, за яких будинок стає не тільки безпечнішим, а й суттєво економити ресурси. Споживання електроенергії взимку значно збільшує вартість комунальних платежів. День стає коротшим, необхідність включення світла стає більшою. Управління освітленням допоможе знизити енергоспоживання та відключити освітлення у кімнатах, коли там нікого немає. Система управління електроживленням може контролювати всі побутові прилади та відключати їх, якщо ніхто не використовує їх. Термостати, які з'являються зараз, дозволяють не тільки дистанційно керувати кліматом і налаштовувати його на необхідні параметри, але і по різному підходити до

вирішення проблеми. Наприклад, у просторій квартирі чи великому будинку немає причин для нагрівання кухні у нічний час.

Налаштування системи «Розумний дім» та управління сукупністю технічних пристроїв, що забезпечують її функціонування, повинні здійснюватися за допомогою поширених сьогодні стандартних стаціонарних та мобільних пристроїв з дружнім інтерфейсом. Необхідно, щоб користувач у будь-який зручний для себе час мав можливість отримати інформацію про стан тих чи інших систем його будинку незалежно від відстані до нього, і при необхідності міг дати вказівку на включення, вимкнення або налаштування даних систем як окремо, так і в цілому. І все це насамперед має забезпечуватись сучасними інтерфейсними системами.

Таким чином, створення систем «розумного будинку» у сукупності з використанням сучасних засобів забезпечення інтерфейсу між ними та користувачем слід вважати одним із важливих та актуальних напрямків розвитку сучасних інформаційних технологій.

Метою роботи є розробка власного web-інтерфейсу для управління електронними системами будинку на основі комплексу апаратно-програмних засобів Arduino. Потрібно розробити web-інтерфейс, який може бути використаний як ядро системи управління будь-якими електронними системами вдома. Web-інтерфейс, що розробляється, повинен бути розширюваним, і, відповідно, забезпечувати можливість додавання нових і редагування вже наявних модулів. Передбачається реалізувати алгоритм роботи web-сервісу для перегляду статусу електронного приладу (простий лампи та розетки), а також включення та вимкнення будь-яких електронних систем.

Задачі, що вирішуються:

- огляд найрейтинговіших систем «розумного будинку»;
- аналіз особливостей реалізації відомих комплексів управління електронними системами будинку, що використовуються нині споживачами;
- вивчення основних особливостей комплексу апаратно програмних засобів розробки Arduino;

- аналіз підходів, що традиційно використовуються при проектуванні інтерфейсів користувача;
- аналіз традиційних способів реалізації управління системами вдома;
- розробка вимог до проєктованого користувача web-інтерфейсу;
- створення прототипу комплексу керування системами будинку;
- налагодження та тестування розробленого сервісу, порівняння результатів роботи з наявними аналогами.

З іншого боку, під час виконання роботи передбачається вирішення низки допоміжних завдань. Зокрема, вивчити принципи розробки середовища Arduino, а також принципи реалізації систем управління розумного будинку та реалізації сервера Arduino.

Об'єкт дослідження – системи дистанційного керування.

Предмет дослідження – система управління розумним будинком із web-інтерфейсом.

Для реалізації завдання використовується середовище розробки Arduino. Створення інтерфейсу користувача виконується на основі PHP. У процесі виконання роботи буде розроблено відповідний web-сервіс, докладний опис якого буде представлений в основній частині пояснювальної записки. Передбачається, що створений web-сервіс може бути використаний на будь-яких пристроях, що працюють під управлінням будь-якої операційної системи. При цьому цей сервіс буде відкритим для подальшої модифікації, удосконалення та розширення функціональних можливостей.

1 ОГЛЯД І АНАЛІЗ СУЧАСНИХ СИСТЕМ «РОЗУМНОГО БУДИНКУ»

Для вирішення поставленої задачі, насамперед, потрібно проаналізувати ринок подібних продуктів, як виявилось, їх там представлена досить велика кількість подібних систем. Попри це, більшість компаній, що працюють у цій галузі, не розголошує основну технічну інформацію про апаратне та програмне забезпечення, які використовуються для їх реалізації. Система «розумний дім» може бути реалізована на таких платах, як Zigbee, Arduino, ПЛК Овен та інших. Розглянемо особливості створення системи типу «розумний будинок» на платі Arduino.

1.1 Аналіз плат Arduino

Перша плата прототипу була виготовлена Массімо Банці у 2005 році. Плата мала найпростіший дизайн і ще не називалася Arduino. У тому ж році Массімо Банці придумав їй ім'я Arduino [1]. Перший прототип плати показаний рисунку 1.1.

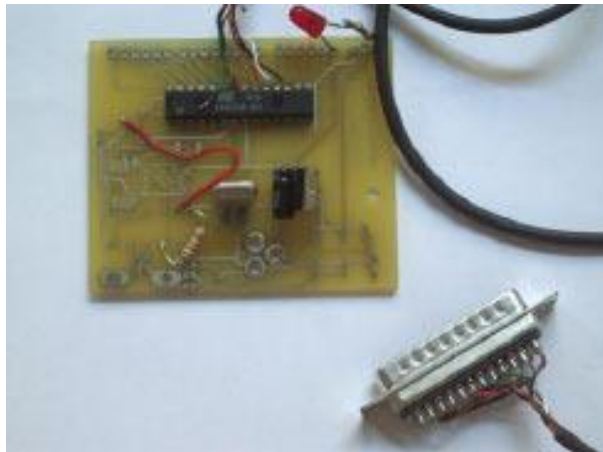


Рисунок 1.1 — Прототип плати Arduino [2]

Випущений у 2005 році як скромний інструмент для студентів Банці в Інституті проектування взаємодій міста Івреа (Interaction Design Institute Ivrea, IDII), Arduino породив міжнародну революцію у сфері міжнародних електронних саморобок. Ви можете купити цю плату за \$30 або зібрати її з нуля. Усі схеми та вихідні коди Arduino доступні безкоштовно за умов відкритих ліцензій. В результаті

Arduino став найвпливовішою апаратною системою свого часу з відкритим вихідним кодом.

Завдяки відкритості системних плат Arduino, допускається їхня вільна модифікація. Тому будь-який виробник плат може випускати аналог плати Arduino, вносити зміни до самої плати, не кажучи вже про вільну комплектацію наборів.

Оскільки Arduino — це бренд виробника, то аналоги зазвичай використовують співзвучні з Arduino назви — Frduino, Freeduino, Xduino, DCcduino, Robotale, Funduino та багато інших плат китайських виробників (Wemos). Бренд може бути і не вказаний, а буде присутній напис типу Arduino. Оригінальні плати Arduino виробляються Італії, а більшу кількість плат-аналогів виробляють у Китаї [3].

Візуально ці плати виглядають однаково (на Arduino є фірмовий логотип у вигляді знака нескінченності).

У виборі плат для розробки розглядатимемо тільки оригінальні плати Arduino.

Плати Arduino відрізнятимуться не лише своїми розмірами, а й кількістю виводів, а також можливістю підключення так званих «шилдів». «Шилд» — це підвид плат розширення, які стикується з Arduino [4]. Розглянемо деякі різновиди Arduino.

Почнемо із найменшої плати Arduino mini. Ця плата не сумісна шилдами, і зазвичай використовується для вбудовування в закінчені аксесуари. На Arduino mini немає USB порту для прошивки. Програмується за допомогою перехідника USB-Serial. Також на платі стоїть дуже малопотужний стабілізатор, і лише один світлодіод (індикація живлення). Arduino mini працюють на мікроконтролері ATmega328 [5].

Arduino Nano, ця плата теж невелика і має вхід mini-usb. Ця плата так само як і Arduino mini не сумісна з "шилдами", але саму плату дуже зручно використовувати як макетну плату. Старі версії цієї плати використовували мікроконтролер ATmega168, у нових версіях використовують ATmega328. Як USB-UART мосту стоять FT232 [6].

Плата Arduino UNO відрізняється тим, що на ній замінили перетворювач USB-UART, на мікроконтролер Atmega8U2. В новіших реалізаціях Atmega16U2 в мікроконтролері закладена прошивка яка робить те, що робить FT32. Завдяки цій зміні підвищилася швидкість прошивки у 2-3 рази [7]. Також варто відзначити, що Arduino UNO підтримує "шилди".

У платі Arduino Leonardo USB не залежить від UART, а так само не залежить від жодного піна. Ця плата побудована на мікроконтролері ATmega32u4. Збільшилася оперативна пам'ять на 0.5 кб., також з'явилося 12 аналогових входів. Крім цього у Arduino Leonardo підтримується не тільки віртуальний Com-порт, але й клавіатура та миша. Також оснащена Micro-USB роз'ємом для прошивки. Але, крім переваг існують деякі недоліки, наприклад, завантажувач займає 4кб, а також у кожному прошивку програмується підтримка USB-blink. В цілому Leonardo має те ж розведення що і UNO тому так само сумісна зі старими "шилдами" [8].

Плата ArduinoMega підтримує велику кількість "шилдів", але через різні розміри плати та через різне розташування SPI виходів, "шилди" використовують цифрові піни, через що вони будуть не сумісні. Наприклад, старий Ethernet "шилд". Плата має велику кількість цифрових виходів, а саме, 54: (16 аналогових, 15 із ШІМ). Має великий обсяг пам'яті 128 та 256 кб. флеш пам'яті 8кб. оперативної пам'яті та 4 кб. EEPROM. Arduino mega використовує мікроконтролер ATmega1280 і ATmega2560 і тому плати відрізняються обсягом пам'яті. USB інтерфейс виконаний на мікроконтролері ATmega8U2 та ATmega16U2 на пізніших версіях. Так само у Mega є USB-host, завдяки якому він може співпрацювати з android-пристроями [9].

1.2 Програмно-апаратні комплекси, що реалізуються в системах розумного будинку

З метою зробити програмний комплекс більш ефективним для вивчення і максимально зручним для використання, був виконаний аналіз вже наявних комплексів «розумних будинків». У процесі роботи було виявлено їх основні переваги та недоліки.

Система «розумний дім» на базі обладнання ОВЕН

ОВЕН ПЛК – моноблочний контролер з дискретними та аналоговими входами/виходами на борту для автоматизації малих систем. Розроблено компанією «ОВЕН». Компанія заснована в 1991 році, коли група ентузіастів організувала мале підприємство, на якому було зроблено першу розробку. Першою розробкою стало влаштування захисного відключення трифазного двигуна ОВЕН [10].

Контролер ОВЕН ПЛК призначений для створення систем управління малими та середніми об'єктами, побудови систем диспетчеризації. ОВЕН ПЛК випускає контролери як малих систем автоматизації, так систем введення виведення і комунікаційних контролерів. Зовнішній вигляд контролера показаний малюнку 1.2.



Рисунок 1.2 — Зовнішній вигляд контролера ОВЕН ПЛК

Основні переваги контролера ОВЕН:

- наявність послідовних портів (RS-485, RS-232) та Ethernet;
- можливість підключення зовнішніх модулів вводу-виводу за будь-яким із вбудованих інтерфейсів;
- розширений температурний діапазон, за яких може працювати пристрій (від-20 до +70°С);
- наявність вбудованого акумулятора.

Основні недоліки контролера ОВЕН:

- висока вартість;
- щодо складне середовище програмування;
- великий розмір.

Система «розумний дім» VIMAR Bu-me

Компанія Vimar була створена Вальтером Віаро і Франческо Гузі першого травня 1945 [12]. Система Bu-me є одним з основних продуктів компанії, і є «інтелектуальним» ядром «розумного будинку», в якому реалізовано поєднання складних технологій з вишуканим дизайном. Система Bu-me забезпечує координоване управління всім комплексом, що включає «сценарії» включення освітлення, роботу автоматики, налаштування кліматичних параметрів, системи безпеки, енергозбереження, дистанційний контроль. Навіть перебуваючи далеко від дому, користувач підтримує постійний контакт із системою Bu-me, користуючись своїм мобільним телефоном [13].

У системі розумного будинку bu-me застосовуються сенсорні вимикачі всіх систем, це може бути і управління освітленням, управління підігрівом підлоги, управління опаленням. На рисунку 1.3 представлений один із перемикачів. Також bu-me пропонує сенсорну панель для керування всіма компонентами розумного будинку. Сенсорна панель вбудовується в стіну в будь-якому зручному місці та пов'язує всі компоненти компанії vimar.

bu-me, встановлені в будинку для зручності керування. Також системою можна управляти з телефону на базі ОС Android.

Основні переваги VIMAR:

- простота встановлення компонентів «розумного будинку»;
- має зручний додаток для мобільних пристроїв;
- можливість віддаленого керування;
- великий комплекс системи.

Основні недоліки VIMAR:

- висока вартість;

- комплекти для розширення компонентів також мають дуже велику вартість;
- обмежена доступність.



Рисунок 1.3 — Система «розумний дім» VIMAR

Система «розумний будинок» be smart MY HOME Legrand & Vticino

Компанія Legrand була заснована сім'єю Legrand в 1866р. і спочатку займалася виробництвом столового фарфору. З появою електрики фарфор починають використовувати для виготовлення ізоляторів та вимикачів. У 1989 році Legrand була придбана італійською компанією Vticino.

Система з шинною організацією характеризується наявністю «інтелектуальних» блоків, з'єднаних між собою шиною, яка служить як обміну інформацією, так живлення. Фізичним носієм для забезпечення зв'язку та живлення є кабель – неекранована кручена пара, до якого паралельно приєднуються пристрої шинної системи. Активуючі пристрої, призначені для контролю навантажень, підключаються не тільки до шини, але і силової лінії 230 В..

Кожен пристрій, підключений до системи, має інтерфейс і власний логічний блок, за допомогою якого пристрій розпізнає команди, що спрямовуються на його адресу, та обробляє їх у рамках закладеної функції. Для того щоб увімкнути лампу, потрібно натиснути клавішу керуючого пристрою, яка відправить цифровий сигнал на пристрій, що з'єднається з лампою.



Рисунок 1.4 — Система «розумний будинок» be smart MY HOME Legrand & BTicino

Функції системи автоматизації MY HOME, реалізовані в дизайні серій AXOLUTE, LIVING, LIGHT та LIGHT TECH, дозволяють керувати функціями, які досі виконувались окремими системами або групами об'єднаних систем. Ці функції включають:

- управління світлом
- управління жалюзі та/або ролетами, вентиляторів та витяжних систем

На відміну від пристроїв традиційної електричної системи, пристрої системи автоматизації мають електронний модуль з програмованою логікою і повинні підключатися до низьковольтної (27 В.) цифрової 2-провідної шини.

У системі існує 2 типи пристроїв:

- керуючі пристрої, що підключаються лише до шини;
- активуючі пристрої повинні підключатися до шини та електромережі 220 В для управління навантаженнями.

У випадку, коли неможливо реалізувати шинну систему, або потрібно розширити вже існуючу електронну систему, не вдаючись до монтажних робіт,

система автоматизації може бути розширена за допомогою дротових радіо-інтерфейсів, керуючих та радіо пристроїв, що характеризуються високою інсталяційною гнучкістю.

При коректному конфігуруванні пристроїв системи автоматизації керувати навантаженнями можна так:

- управління поодиноким навантаженням (світильник, жалюзі тощо);
- управління однією або декількома групами навантажень (наприклад, тільки жалюзі на 1-му поверсі, або жалюзі північної сторони будинку тощо);
- одночасне керування всіма навантаженнями (наприклад, вимкнути всі світильники в будинку та/або опустити жалюзі).

Існує можливість створення сценаріїв, які складаються з низки одночасних засобів керування, призначених для підвищення комфорту. Наприклад, натиснувши клавішу керуючого пристрою або використовуючи графічне меню сенсорної панелі, можна запустити сценарій, індивідуальний для кожного користувача (синхронне увімкнення кількох світильників, керування жалюзі і т.д.) Якщо система автоматизації об'єднана з 2-провідною звуковою системою та системою термоконтролю, сценарій також може включати звукове оформлення і регулювання необхідної температури у приміщенні.

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:

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

Система «розумний дім» iNels від компанії Elko ep

Компанія ELKO EP — чеська компанія з 18-річною історією. З самого свого заснування компанія спеціалізувалася на розробці та виробництві електронних модульних пристроїв. Завдяки тривалому досвіду у цій сфері, сьогодні компанія пропонує своїм клієнтам досить широкий асортимент продукції, а саме понад 400 типів пристроїв для вирішення різних завдань електропостачання, як у виробничій сфері, так і в побуті.



Рисунок 1.5 — Система «розумний дім» iNels від компанії Elko ep

Система «розумного будинку» iNels, розроблена компанією, використовує сенсорний екран для керування всіма елементами розумного будинку та системами мультимедіа. Меню складено з трьох секцій, за функціями системи. Новинкою є меню управління ІП домофоном. Завдяки сенсорній панелі іТР контролюєте всі ваші електроспоживачі, освітлення та опалення. Можете включити фільм або музику в будь-якій зоні, переглядати камери або інформацію з датчиків безпеки. Сенсорний екран представлено на рисунку 1.6.



Рисунок 1.6 — Сенсорна панель iNels.

Основні переваги iNels:

- великий комплект для самостійного встановлення;
- велика кількість систем і модулів, що підключаються;
- зручне управління на вбудованій системі сенсорної панеллю;
- розширення функціоналу системи без капітальних будівельних робіт.

Основні недоліки iNels:

- немає віддаленого управління;
- не має будь-яких наборів (все окремо);
- важка доступність на ринку;
- найвища вартість периферійних компонентів.

Система «розумний дім» Life Control

Починаючи з 27 грудня 2016 р. у магазинах електроніки, а також на полицях торгових точок у продажу з'являться пристрої, випущені під брендом Life Control — комплексної екосистеми «розумних» пристроїв, доступних кожному та повністю готових до миттєвого підключення та використання.

В ядрі системи «розумного будинку» (модель MCLH-01) використовується стандарт ZigBee Home Automation 1.2. Крім того, вбудований 2G/3G/4G-модуль, в наявності SIM-карта, формат microSIM. У LTE підтримується тільки діапазон 7 і 20, але для домашнього роутера цього вистачає, різноманітним датчикам великі швидкості поки не потрібні, а використовувати роутер для домашнього інтернету,

напевно, багато користувачів не буде, або це робитиме час від часу, коли відсутній звичайний, провідний інтернет.



Рисунок 1.7 — Система «розумний дім» Life Control

У центрі управління «розумним будинком» знаходиться акумуляторна батарея ємністю 2400 мАг, це дозволяє системі функціонувати близько 5 годин після відключення електрики в будинку, що є безперечним плюсом системи. Також варто зазначити, що в даній системі є тарифи щомісячної оплати за використання системи, сума оплати залежить від кількості модулів системи.

Основні переваги Life Control:

- великий комплект для самостійного встановлення;
- велика кількість систем і модулів, що підключаються;
- зручне керування через програму на смартфоні;
- система має доступ до інтернету через SIM карту;
- автономна робота від АКБ.

Основні недоліки Life Control:

- щомісячна оплата за використання системи;
- не має будь-яких наборів (все окремо);
- висока вартість систем управління та модулів управління;
- найвища вартість периферійних компонентів.

1.3 Огляд традиційних підходів до проектування інтерфейсів користувача

Інтерфейс користувача повинен забезпечувати зв'язок між комп'ютером і користувачем, обмін діями та відповідями на них. Дизайн інтерфейсу користувача є фактором, що впливає на три основні показники якості програмного продукту: його функціональність, естетику і продуктивність.

Функціональність є фактором, на який розробники програм часто звертають основну увагу. Вони намагаються створювати програми так, щоб користувачі могли виконувати свої завдання та їм було зручно це робити. Функціональність важлива, проте це не єдиний показник, який повинен враховуватися в ході розробки інтерфейсу користувача.

Естетичний зовнішній вигляд самого додатка та способу його уявлення дозволяє сформувати у споживача позитивну думку про програму. Проте естетичні характеристики дуже суб'єктивні і описати їх кількісно набагато складніше, ніж функціональні вимоги чи показники продуктивності. Вся естетика програми часто зводиться до простого вибору: чи співвідносяться між собою використовувані кольори, чи передають елементи інтерфейсу їх призначення і сенс операцій, що відчувається, при використанні тих чи інших елементів управління і наскільки успішно він їх використовує.

Продуктивність і надійність також впливають на перспективу застосування програми. Якщо програма добре виглядає, має просте та зручне управління, але, наприклад, повільно промальовує екрани, регулярно «підвисає» на десятків-другий секунд або зовсім перестає функціонувати з критичною помилкою при некоректних діях користувача. Напевно, у користувача буде мало шансів на тривалу експлуатацію. У свою чергу, швидка та стабільна робота програми може частково компенсувати її не найкращий дизайн або відсутність якихось додаткових функцій.

1.4 Аналіз традиційних способів реалізації управління системами будинку

Технологія розумного будинку створена, щоб забезпечити автоматичну роботу всіх систем життєзабезпечення та безпеки. Вона розпізнає зміни у навколишньому середовищі та приміщенні, реагуючи на них відповідним чином як за вказівкою користувача, так і самостійно. Основною особливістю такої технології є об'єднання окремих підсистем та пристроїв у єдиний комплекс, керований за допомогою автоматики.

Сучасні квартири є складним інженерним комплексом. Інтелектуальний будинок здатний взяти на себе турботи з управління енергопостачанням, опалювальними системами, водопроводом, вентиляцією та кондиціонуванням.

Щоб власник міг отримати максимальну вигоду з використання таких систем, їхня робота повинна бути узгоджена. Таким чином, батареї не безцільно обігріватимуть приміщення з відкритими вікнами, а бездротові технології дозволять клієнту негайно дізнатися про будь-яку позаштатну ситуацію де б той не знаходився. Одним з варіантів створення розумного будинку є DIY-рішення, коли користувач самостійно збирає та підключає інтелектуальні пристрої. Зазвичай ці рішення пропонуються «з коробки», і їм не потрібне налаштування. Проте більший інтерес представляють професійні системи, їх установкою та налаштуванням займаються професійні компанії, більш того, проект таких рішень часто узгоджується ще на етапі будівництва будинку.

Розумний будинок спочатку не повинен проектуватись як автономна система. Необхідно заздалегідь передбачити способи її взаємодії із системами вищого рівня, наприклад, системами екстреної допомоги, а також системами обліку ресурсів. Виходить, що варто відразу закласти можливість взаємодії із зовнішнім світом, щоб забезпечити високу якість сервісів, що надаються користувачеві.

В ідеальному випадку проектування системи розумного будинку має починатися одночасно з проектуванням самої будівлі (або квартири), оскільки в цій системі задіяна досить велика кількість електричних та сигнальних кабелів, і сховати їх під штукатуркою не вдасться. Також деякі проекти приділяють увагу таким труднощам і намагаються організувати систему без капітальних будівельних робіт.

Другий варіант є більш перспективним і являє собою однорангову мережу, що самоналаштується, на базі протоколів Z-Wave, ZigBee, Bluetooth. Що стосується провідних систем, то вони будуються на основі послідовних протоколів типу 1-wire. Їхнім очевидним недоліком є необхідність прокладання спеціальних кабелів.

1.5 Обґрунтування вибору плати Arduino

У першому розділі розглянуто різновиди плат Arduino, а також проаналізовано системи «розумного будинку». Виконано огляд найпопулярніших систем «розумного будинку». Також були виявлені основні переваги та недоліки систем.

При виборі плати Arduino для розробки системи розумного будинку основними критеріями було обрано зручність розробки системи та поширеність на внутрішньому ринку. Також треба враховувати кількість пінів на платі для реалізації як невеликих систем, і великих. Використання плат Arduino mini і Arduino nano переважно підходить для закінчених проєктів, але їх застосування для системи, що розробляється, не є кращим. Arduino UNO і Arduino Leonardo можна використовувати для проєктування систем «розумного будинку», але потрібно врахувати, що можливостей цих плат може вистачити для створення великих систем. Кожна з вище представлених плат має свої переваги і недоліки.

Аналіз представлених даних дозволяє зробити висновок, що найбільш оптимальним варіантом плати для розробки системи розумного будинку є плата Arduino Mega. На користь вибору цієї свідчить велика кількість пінів, а також сумісність з великою кількістю шилдів. Також варто відзначити, що для того, щоб система могла керуватися віддалено, потрібно буде підключити Ethernet шилд і створити web сервер.

1.6 Висновки

У першому розділі розглянуті та проаналізовані плати Arduino, представлені їх характеристики, переваги та недоліки. Узагальнено інформацію про всі найпопулярніші системи «розумного будинку». Також виявлено головні переваги та недоліки розглянутих систем. Проаналізувавши найпопулярніші системи

«розумного будинку», було вирішено розробляти програмно-апаратний комплекс під керуванням плати Arduino. Оскільки вартість плати є достатньо доступною. Цей вибір є найоптимальнішим для досягнення заданої мети. Також для реалізації web сервісу знадобиться Ethernet шилд.

2 ПРОЕКТУВАННЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

2.1 Постановка задачі

Метою роботи є розробка програмно-апаратного комплексу для управління електронними системами будинку за засобами web інтерфейсу. Потрібно розробити web ресурс який буде включати функціонал управління системи розумного будинку, а також індикації сигналізацій і працюючих ламп і розеток. Web-ресурс повинен забезпечувати віддалене керування електронними системами вдома. У системі мають бути реалізовані такі можливості:

- змінювати стани освітлювальних приладів;
- змінювати стан розеток;
- відображати поточний стан освітлювальних приладів, розеток та сигналізацій.

При проектуванні системи мають бути враховані такі рекомендації та положення:

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

Проектований web-ресурс має стати основою з мінімальною функціональністю (вкл/викл лампу або розетку), необхідною та достатньою для простого використання. Передбачається, що можливості системи можуть бути

розширені шляхом поповнення розеток, ламп, електронних приладів або сигналізацій.

2.2 Функціональні вимоги до програмного виробу

У розроблюваної системі має бути реалізований наступний інтерфейсний функціонал та компоненти:

1. Макет головної сторінки web-ресурсу системи управління «розумним будинком» представлено на рисунку 2.1.

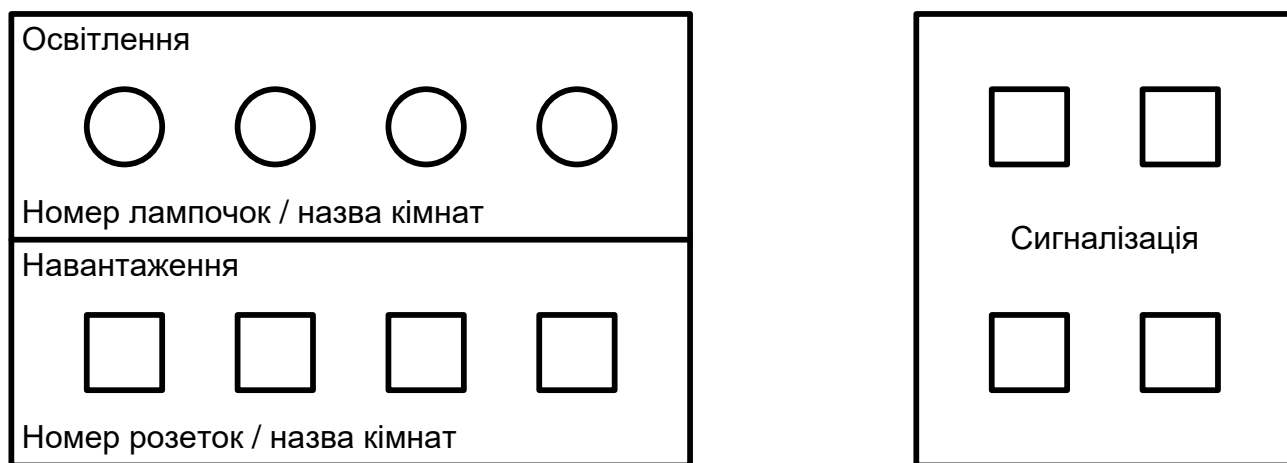


Рисунок 2.1 — Макет головної старіці web-ресурсу

2. Макет сторінки з індикацією web-ресурсу системи управління «розумним будинком» представлено малюнку 2.2.

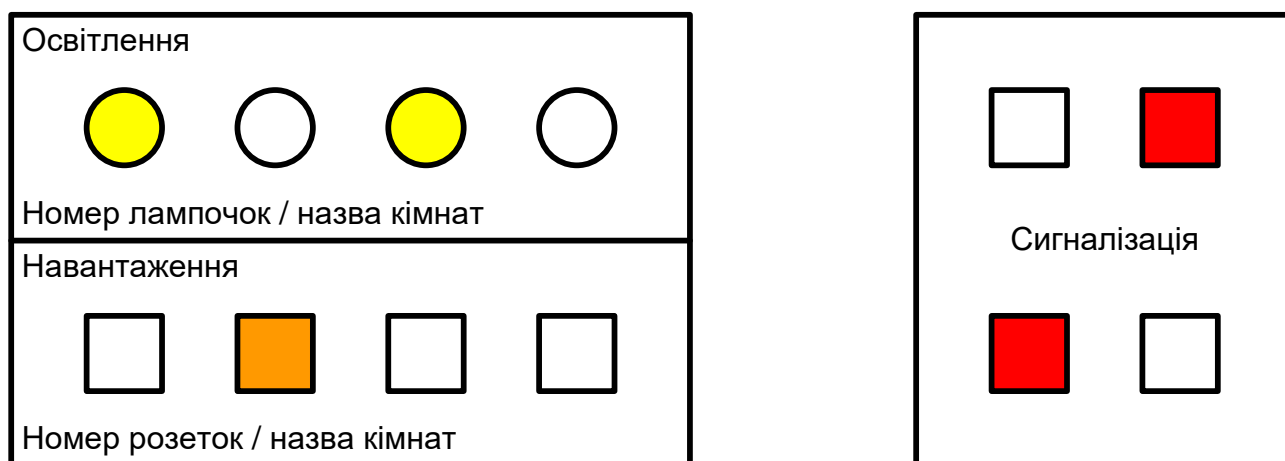


Рисунок 2.2 — Макет сторінки web-ресурсу з індикацією

Програмний комплекс повинен бути працездатним на пристрої, що відповідає наступним вимогам:

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

2.3 Архітектура системи

Як зазначалося вище, система, що розробляється, призначена для експлуатації у складі обчислювальних систем під управлінням будь-якої операційної системи і буде реалізована мовою C++. Для реалізації інтерфейсу використано мову PHP. Архітектура системи досить проста, що дає певні переваги у вигляді безумовної швидкодії, зручності використання. У цілому нині архітектура системи є найпростішу конструкцію для безпосереднього обміну даними між платою контролера і браузером (рис. 2.3).

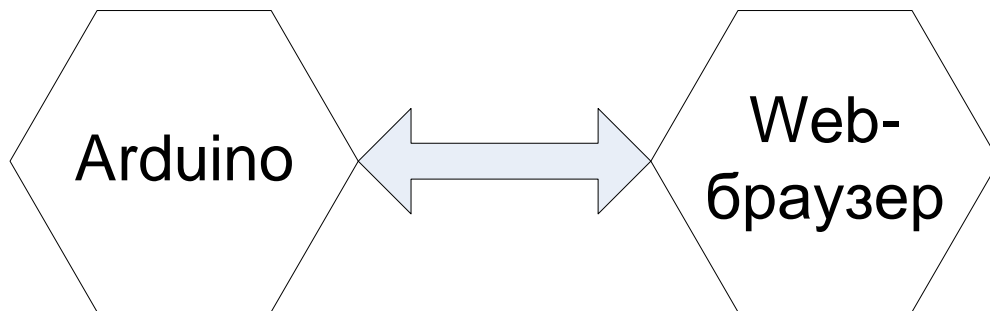


Рисунок 2.3 — Архітектура системи

2.4 Структура роботи програмно-апаратного комплексу

На лампочки і розетки з веб-браузера подається сигнал на вмикання або відключення живлення, ці сигнали передаються на маршрутизатор, маршрутизатор у свою чергу відправляє дані на інтернет-шилд Arduino, який активує піни на Arduino. Сигнали ламп і розеток реалізовані серед розробки Arduino як output, а

сигналізація як input [27]. На рисунку 2.4 показано просту схему роботи системи (обміну даними). Сигналізація проходить таку саму процедуру тільки в зворотному порядку. Приклад оголошення цифрових входів/виходів сигналізації:

```
pinMode(5, INPUT);
```

```
pinMode(6, OUTPUT).
```

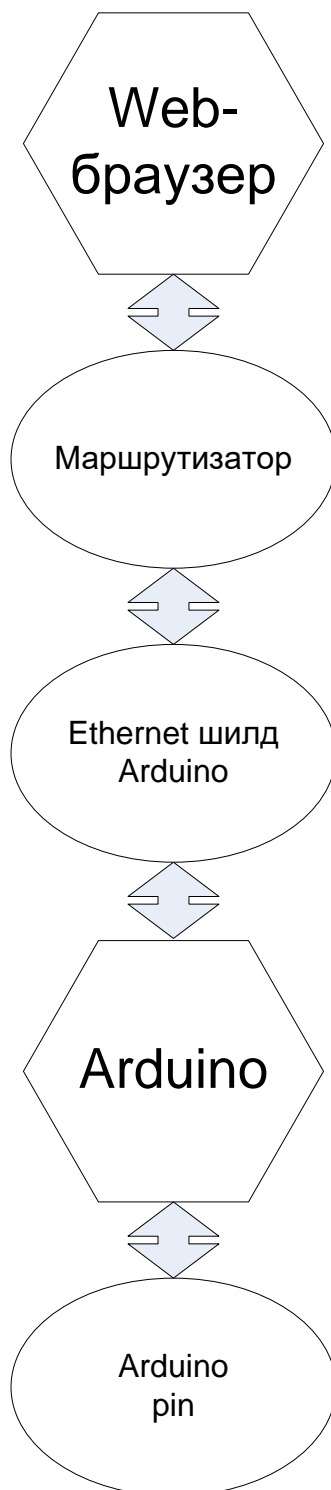
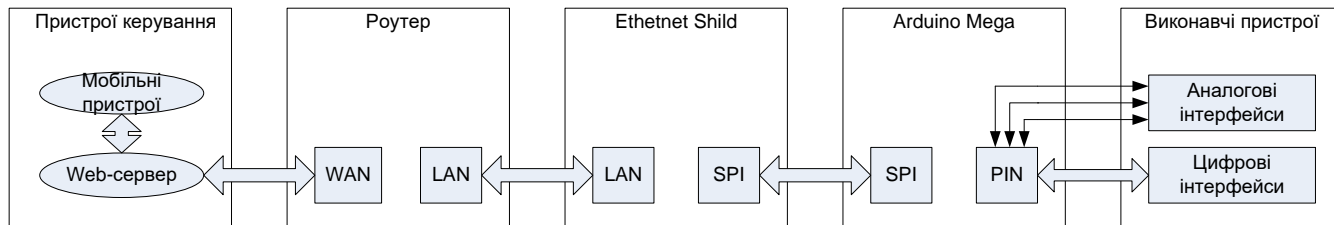


Рисунок 2.4 — Схема роботи системи

2.5 Апаратна реалізація

У представленому проекті для реалізації web-сервера використовується Інтернет з'єднання. В загальному структура апаратної частини має вигляд.



Arduino Mega 2560 є основним керуючим вузлом в системі. Зовнішній вигляд показано на рисунку 2.5.

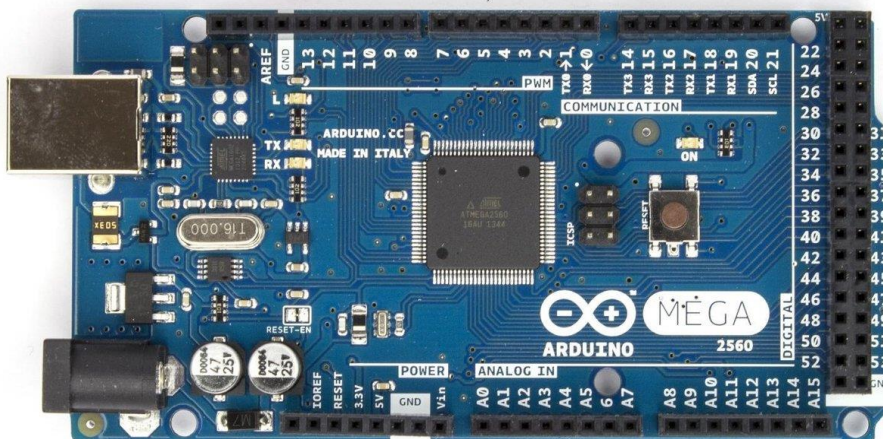


Рисунок 2.5 — Апаратна платформа Arduino Mega 2560

Arduino Mega 2560 — це пристрій на основі мікроконтролера ATmega2560. До його складу входить все необхідне для зручної роботи з мікроконтролером: 54 цифрових входи/виходи (з яких 15 можуть використовуватися як ШІМ-виходи), 16 аналогових входів, 4 UART (апаратних приймачів для реалізації послідовних інтерфейсів), кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP для внутрішньосхемного програмування та кнопка скидання. Для початку роботи з пристроєм досить просто подати живлення від AC/DC-адаптера або батарейки, або підключити його до комп'ютера за допомогою кабелю USB.

Arduino Mega сумісний з більшістю плат розширення, розроблених для Arduino Duemilanove та Diecimila.

Mega 2560 — це оновлена версія Arduino Mega.

Ардуїно Mega 2560 відрізняється від усіх попередніх плат тим, що в ньому для перетворення інтерфейсів USB-UART замість мікросхеми FTDI використовується мікроконтролер ATmega16U2 (ATmega8U2 у версіях плати R1 та R2).

На платі Mega 2560 версії R2 доданий резистор, що підтягує до землі лінію HWB мікроконтролера 8U2. Подібна міра дозволяє спростити процес оновлення прошивки та перехід пристрою в режим DFU.

Основні характеристики платформи:

- мікроконтролер – ATmega2560;
- робоча напруга – 5В;
- напруга живлення (рекомендована) – 7-12В;
- напруга живлення (гранична) – 6-20В;
- цифрові входи/виходи – 54;
- аналогові входи – 16;
- flash-пам'ять – 256КБ;
- тактова частота – 16МГц;
- максимальний струм одного виводу 40 мА;
- максимальний вихідний струм одного виводу 50 мА.

Преваги платформи Arduino перед конкурентами:

- Низька ціна в порівнянні з більшістю подібних платформ.
- Кросплатформеність. Arduino працювати можна хоч на Windows, хоч на Mac OS, тоді як більшість платформ розробляється виключно під Windows.
- Проста програмна обстановка. Платформа відмінно підходить для новачків, а докладних уроків в мережі безліч.
- Відкритий ресурс і розширюване програмне забезпечення. Просунуті програмісти можуть без проблем самотійно розширювати під себе ПЗ.

Елементом з'єднання між платформою та роутером може виступати або Ethernet-модуль ENC28J60 (рис. 2.6) або Ethernet-шилд W5100 (2.7).



Рисунок 2.6 — Ethernet-модуль ENC28J60

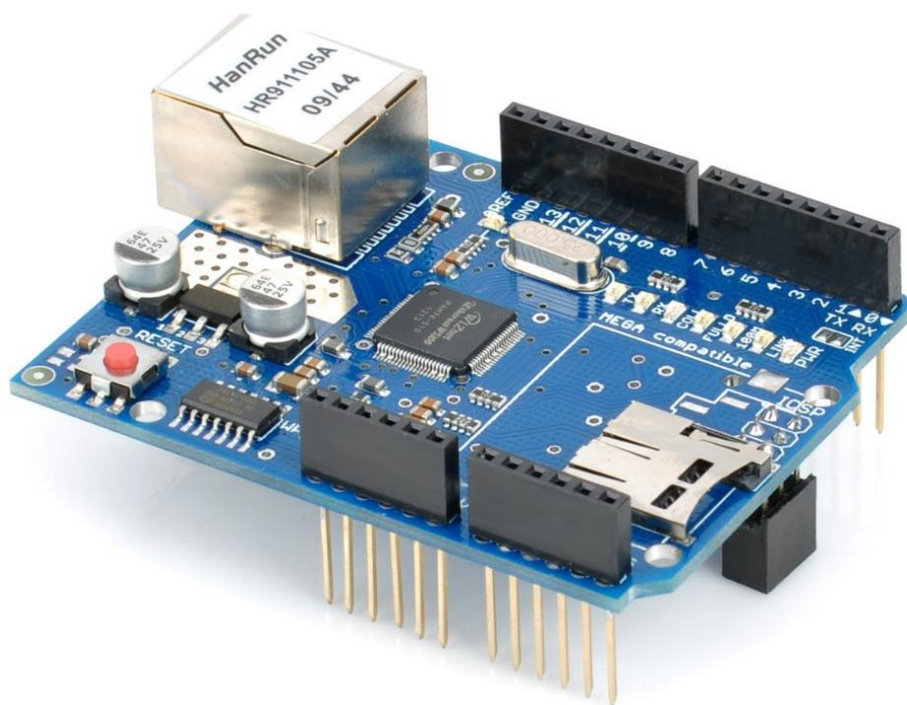


Рисунок 2.7 — Ethernet-шилд W5100

Перевага першого полягає в дешевизні, другого – простота підключення. Щоб підключити Ethernet-модуль ENC28J60 необхідно розглянути SPI інтерфейс, адже цей модуль підключається до Arduino за допомогою інтерфейсу SPI.

Назва інтерфейсу SPI є аббревіатурою від «Serial Peripheral Bus», що можна перекласти як «шина для підключення периферійних пристроїв». Звідси випливає її головне призначення – зв'язати один головний пристрій – Ведучий (Master) – з одним або декількома Ведомими (Slave). Ведучий у цьому інтерфейсі завжди один, тільки він керує всім процесом і він може формувати тактові імпульси. Якщо в нашому випадку Ведучим завжди є мікроконтролер (цю роль може виконувати і комп'ютер, але це інша історія), то В переважній більшості випадків є саме периферичні пристрої. Датчики, дисплеї, мікросхеми ЦАП та АЦП, RFID-рідери, модулі бездротового зв'язку, включаючи приймач-передавачі WiFi та Bluetooth, GPRS-адаптери тощо. Для зв'язку контролерів та/або комп'ютерів зазвичай не використовується, але взагалі це можливо, і такий приклад ми розглянемо нижче. Особливо цей інтерфейс потрібний там, де потрібна висока швидкість передачі даних і не менш висока надійність. SPI саме такий, він найшвидший з усіх наявних у нашому розпорядженні і найлегший з погляду споживаних ресурсів. Розплатою за це є використання більшої кількості дротів, ніж для інших інтерфесів. Тут потрібно аж 3 штуки тільки безпосередньо для передачі даних, не дарма ж його друге ім'я 3-ware. А саме: Розплатою за це є використання більшої кількості дротів, ніж для інших інтерфесів. Тут потрібно аж 3 штуки тільки безпосередньо для передачі даних, не дарма ж його друге ім'я 3-ware. А саме: Розплатою за це є використання більшої кількості дротів, ніж для інших інтерфесів. Тут потрібно аж 3 штуки тільки безпосередньо для передачі даних, не дарма ж його друге ім'я 3-ware. А саме:

MOSI - Master Output Slave Input (Ведучий передає, Ведений приймає), MISO - Master Input Slave Output (Ведучий приймає, Ведений передає), SCLK, інакше SCK - Serial Clock (тактовий сигнал).

Тактова частота інтерфейсу може досягати 20 МГц. Для підключення до мережі TCP/IP використовується роз'єм RJ-45. Модуль який розглядається гальванічно розв'язаний із ним.

Контролер модуля має максимальну тактову частоту 25 МГц. Усі функції роботи з мережними протоколами покладено нього, Arduino залишається лише відправляти пакети даних і приймати отримані.

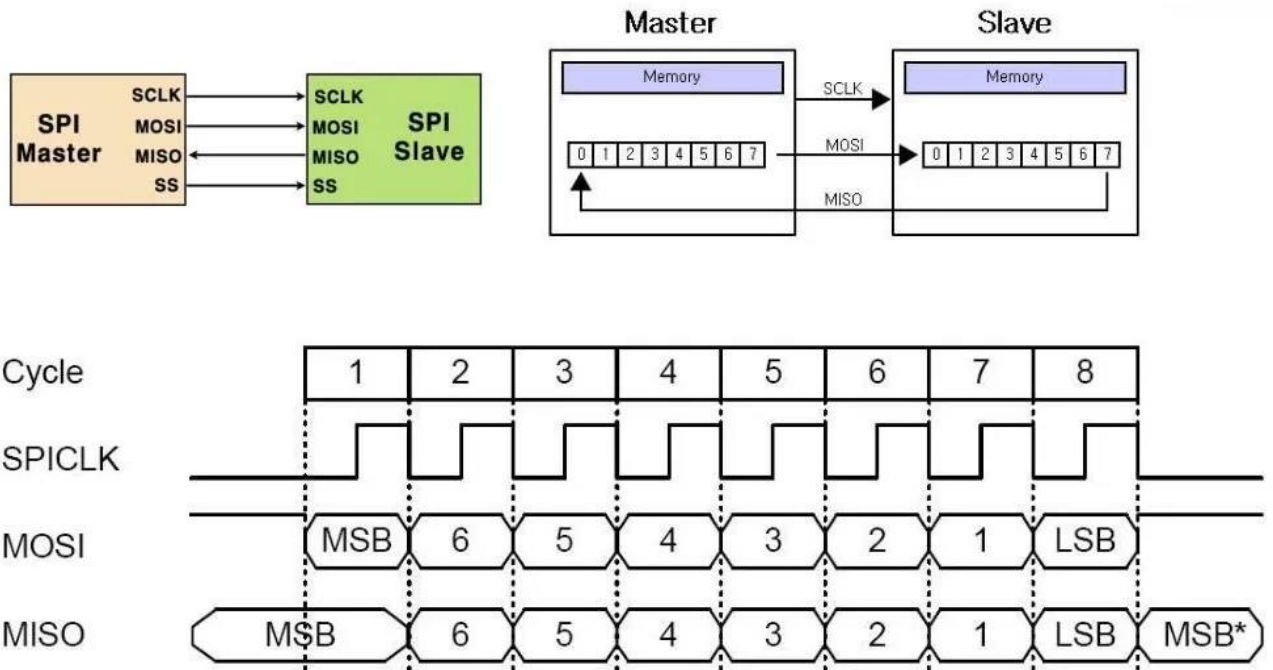


Рисунок 2.8 — SPI інтерфейс

Тепер розберемо підключення ENC28J60 до Arduino.

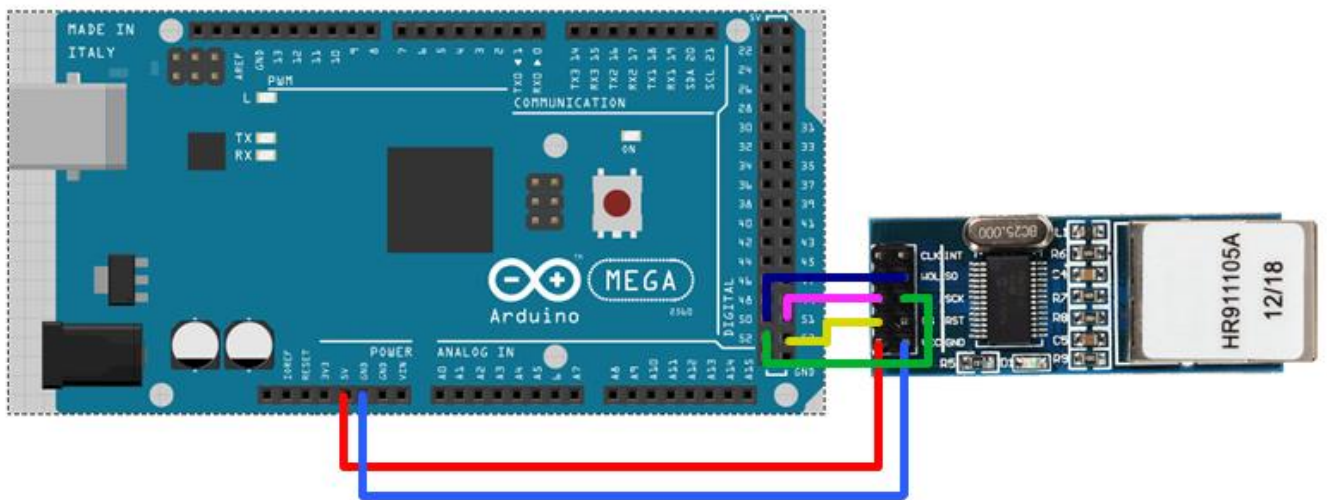


Рисунок 2.9 — Підключення ENC28J60 до Arduino

При використанні Arduino Mega вона виглядатиме так:

VCC — 3v3;

GND — GND;

CS — 53;

SI — 51;

SO — 50;

SCK — 52.

У випадку використання шилда Arduino W5100 схему підключення до Arduino mega 2560 показано на рис. 2.10.

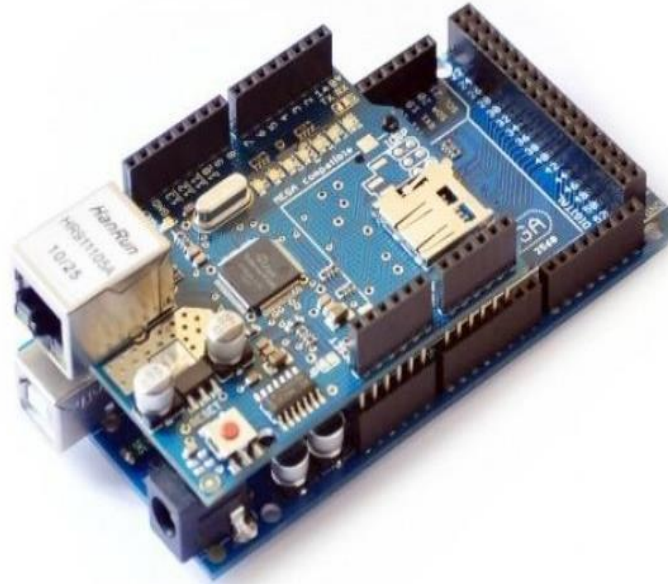


Рисунок 2.10 — Підключення шилда W5100 до Arduino

Не дивлячись на дороговизну, зупинимося на другому варіанті так, як в подальшому планується розширення функціональних можливостей.

2.6 Висновки

У другому розділі було розглянуто такі питання: постановка завдання, функціональні вимоги до програмного виробу, архітектура системи, структура бази даних. Також було розглянуто макет дизайну web сервісу.

Було дано детальну інформацію про мету і функціонал розроблюваної програми, розглянуто важливі аспекти. Також було розглянуто принцип взаємодії веб-браузера з комплексом Arduino. Зроблено аналіз відомих систем розумного будинку. Також були виявлені основні переваги та недоліки систем конкурентів.

3 РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ «РОЗУМНИЙ БУДИНОК»

У рамках роботи було виконано проектування та розробка системи «Розумний дім» для управління електронними системами будинку через web-ресурс. Розроблений комплекс використовується для керування освітленням, навантаженням та індикацією сигналізації. Також у комплексі передбачено розширення систем управління.

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 дистанційного керування системою через web-інтерфейс. Для того, щоб потрапити на web-інтерфейс, потрібно ввести його адресу в рядку браузера, в нашому комплексі це <http://192.168.10.101/>. Під час запуску web-інтерфейсу користувач бачить на екрані свого пристрою сторінку керування освітленням, навантаженням та індикацією сигналізації (рис. 3.1). На даному етапі користувачеві пропонується вибрати розділ керування освітленням або навантаженням, що цікавить його.

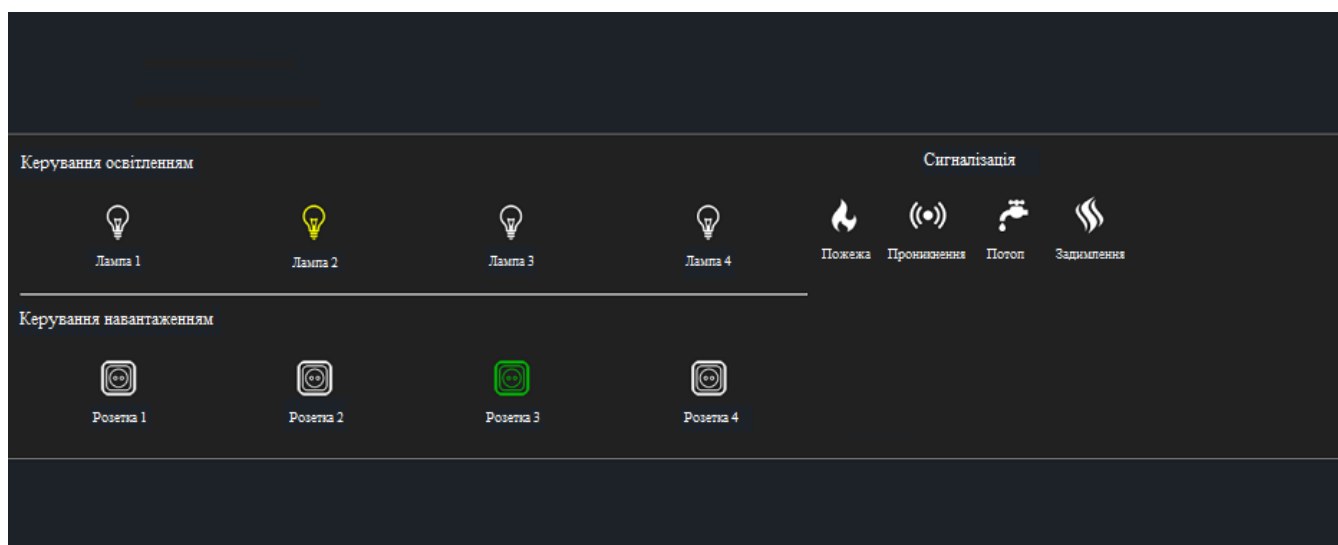


Рисунок 3.1 — Сторінка web-сервісу з індикацією

На сторінці web-ресурсу знаходяться такі розділи:

- управління освітленням;
- управління навантаженням;
- сигналізація.

За управління освітленням та навантаженням на платі Arduino відповідають цифрові виходи з 22 по 29. Виходи пінів реалізовані таким чином:

```
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 до 49. Приклад реалізації:

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

3.1.1 Завантаження кнопок перемикачів

У цьому веб-ресурсі зображення кнопок та індикацій реалізована завантаженням з web-ресурсу. Приклад завантаження картинок:

```
if (_gtv11)
{
_swi8 = String("http://... /arduino/rozetka1.png");
```

```

}
else
{
_sw18 = String("http://... /arduino/rozetka0.png");
}

```

У нашому випадку `rozetka0`-це вимкнений стан, а `rozetka1` — ввімкнений. Всі інші кнопки реалізовані за тим самим принципом. Індикація зображено рисунку 3.2.

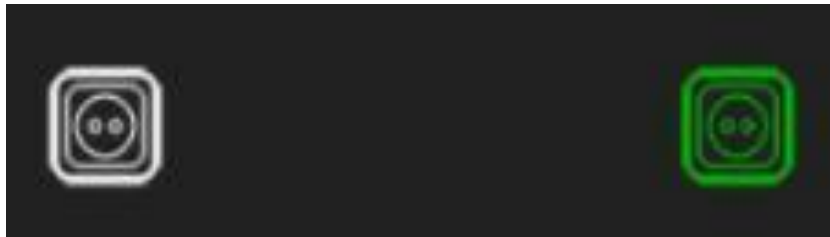


Рисунок 3.2 — Приклад стану модулів ввімкнено/вимкнено

3.1.2 Реалізація веб-сервера

У представленому проекті для реалізації веб-сервера використовується шилд Arduino W5100.

Для того щоб створити веб-сервіс, необхідно вказати бібліотеки, що використовуються для управління Ethernet.

Також нам необхідно вказати параметри нашого мережевого пристрою. Для цього вказуємо MAC-адресу. Вибрана MAC-адреса не повинна збігатися з MAC-адресою ваших мережевих пристроїв. Так само і IP-адреса, вона повинна бути індивідуальною, але перебувати в підмережі. Наприклад, у проєктованій системі роутер має наступну адресу 192.168.0.1, а ПК 192.168.0.4. З цього випливає, що цей пристрій може мати наступна IP адреса: 192.168.0.101.

Далі потрібно вказати порт. За замовчуванням це 80, тому що веб-браузер за замовчанням опитують саме його. Приклад коду:

```

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);
```

3.1.3 Організація індикації сигналізації на web ресурсі

На нашому web-ресурсі частина сторінки, де знаходиться сигналізація (рис. 3.3) оновлюється раз на 10 секунд. Це зроблено для зручності, щоб щоразу не оновлювати сторінку вручну.

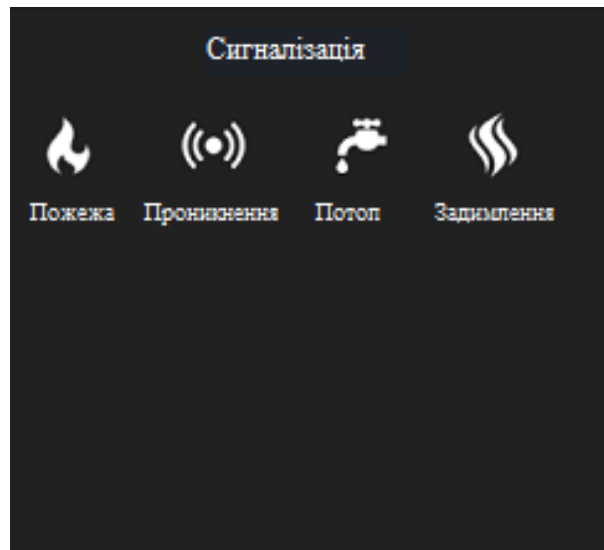


Рисунок 3.3 — Сигналізація

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

3.2 Особливості організації віддаленого керування системою

Розглянемо варіант підключення для керування системою віддалено. Для початку користувачеві необхідно зареєструватись на сайті noip.com. Реєстрація потрібна для того, щоб тільки ви могли редагувати адресу призначення web ресурсу. Після реєстрації на сайті слід перейти до розділу Dynamic DNS. У цьому розділі додається ip-адреса розробленого web-ресурсу (рис. 3.4).

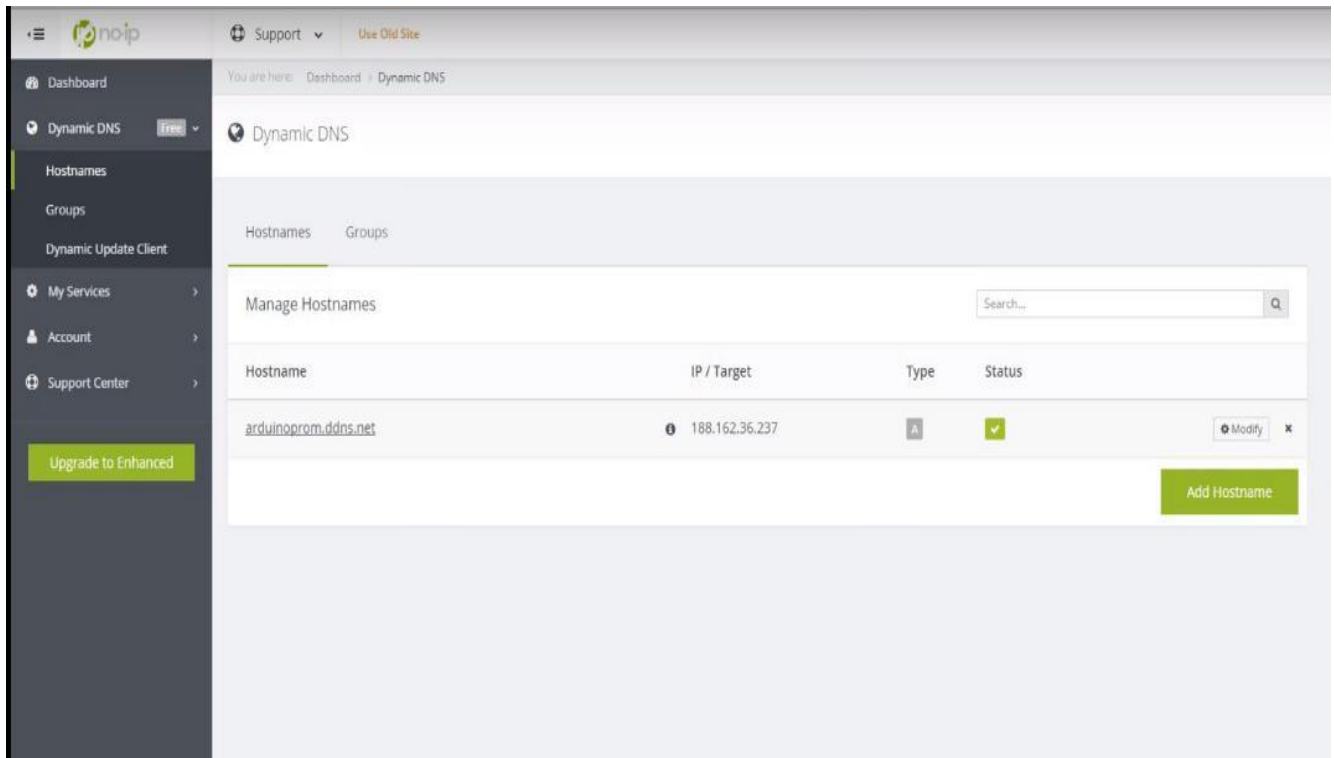


Рисунок 3.4 — Приклад додавання IP-адреси.

Потім потрібно зайти в налаштування нашого маршрутизатора, для цього в браузері вводиться 192.168.0.1. При цьому адреса налаштувань маршрутизатора може бути іншою. Це залежить від провайдера та моделі використовуваного маршрутизатора. У налаштуваннях маршрутизатора вибирається розділ «Налаштування локальної мережі». У налаштуваннях локальної мережі необхідно вказати MAC та IP-адреси пристрою, що підключається. Ці дані беруться із вихідного коду:

```
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 дистанційного управління є відсутність системи безпеки. Це означає, що будь-який користувач, що знає ip-адресу web-ресурсу, зможе керувати створеною системою.

3.3 Налагодження та тестування розробленого сервісу, порівняння результатів роботи з наявними аналогами

Після складання макету системи «розумного будинку» на платі Arduino mega 2560 та Ethernet шилда w5100 було проведено налагодження програмного коду та подальше тестування даного макету. У ході тестування програмно-апаратного комплексу було виявлено такі недоліки: при використанні віддаленого управління системою відсутній такий критерій як безпека. Це обґрунтовано тим, що сторінка управління не має аутентифікації та будь-яка людина знає IP адресу обладнання може втрутитися в роботу системи. Під час налагодження були виявлені недоліки, щоб побачити індикацію сигналізації доводилося оновлювати сторінку вручну. Цей недолік бу виправлений оновленням сигналізації кожні 10 секунд.

Приклад із коду:

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

3.4 Можливість розширення функціональних можливостей

Розглянемо як приклад розширення, реалізацію віддаленого управління кондиціонером через telegram бот. Основні апаратні складові, окрім вищезгаданої Arduino Mega та дещо доступнішого ethernet-shield ENC28J60, це датчик температури і вологості серії DHT, транзистор BD135 і світлодіод TSAL6200.

DHT — цифровий датчик температури і вологості, який показано на рисунку 3.5, містить калібрований цифровий вихідний сигнал з показаннями температури і вологості. Датчик включає в себе резистивний сенсор вологості і компоненти NTC структури для вимірювання температури. Датчик DHT складається з ємнісного датчика вологості і термістора. Також, датчик містить в собі простенький АЦП для перетворення аналогових значень вологості і температури.

Розглянемо 2 види недорогих цифрових датчика температури і вологості DHT11 і DHT22.

По розпіновці дані датчики ідентичні, проте трохи розрізняються характеристиками.

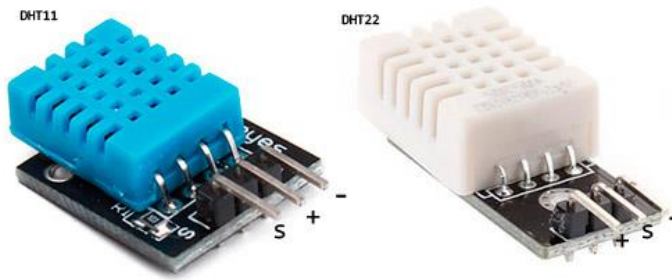


Рисунок 3.5 — Датчик температури і вологості DHT11

Розпіновка:

- Vcc — живлення +3,5 ~ 5,5 В;
- S — передача даних;
- NC — не використовується;
- GND — земля(-).

DHT11:

- низька вартість;
- живлення I/O 3-5 В;
- визначення вологості 20-80% з 5% точністю;
- визначення температури 0-50 град. з 2% точністю;
- частота опитування не більше 1 Гц (не більше одного разу в 1 с);
- розміри 15.5мм x 12мм x 5.5мм;
- 4 виводи з відстанню між ніжками 0.1"

DHT22:

- живлення і I/O 3-5 В;
- визначення вологості 0-100% з 2-5% точністю;
- визначення температури -40 - 125 град. з точністю ± 0.5 град.;
- частота опитування не більше 0.5 Гц (не більше одного разу в 2 с);
- розміри 15.1мм x 25мм x 7.7мм;
- 4 виводи з відстанню між ніжками 0.1 "

Сенсор DHT22 має більш кращі характеристики. Обидва датчика повільні і зчитувати з них інформацію можливо не частіше, ніж раз в 1 або 2 с.

В якості комутуючого елемента використано твердотільні реле зовнішній вигляд якого показано на рисунку 3.6.



Рисунок 3.6 — Твердотільне малогабаритне реле

Характеристики:

- Серія: G3MB;
- Канали: SPST-NO (1 Form A);
- Тип виходу: AC, Zero Cross;
- Струм навантаження: 2A;
- Напруга керуючого входу: 4 ~ 6 VDC;
- Напруга навантаження: 75 ~ 264 V.

Також можна використовувати контактна реле, наприклад SRD-05VDC-SL-C, зовнішній вигляд якого показано на рисунку 3.7.



Рисунок 3.7 — Контактна реле SRD-05VDC-SL-C

Характеристики:

- контакти 1 x DT;
- номінальна напруга котушки 5 В;
- номінальний струм котушки 71,4 мА;
- максимальна напруга комутації (АС) 250 В;
- максимальний комутований струм 10 А.

Транзистор виконує роль ключа та комутує силову частину у вигляді реле. В якості елемента для відображення використовується світлодіод. Транзистор має характеристики, які подано в таблиці 3.1.

Таблиця 3.1 — Характеристики транзистора BD135

Структура	nnp
Макс. напр. к-б при заданому зворотному струмі до розімкнутого ланцюга е.($U_{кбо\ макс}$),В	45
Макс. напр. к-е при заданому струмі до розімкнутого ланцюга б.($U_{кез\ макс}$),В	45
Максимально допустимий струм ($I_{к\ макс}$,А)	1.5
Статичний коефіцієнт передачі струму h_{21e}	63...250
Гранична частота коефіцієнта передачі струму $f_{гр}$, МГц	190
Максимальна потужність, що розсіюється ,Вт	8
Корпус	to-126
Вага, г	1.5



Рисунок 3.8 — Транзистор BD135

Схему вмикання елементів показано на рисунку 3.9.

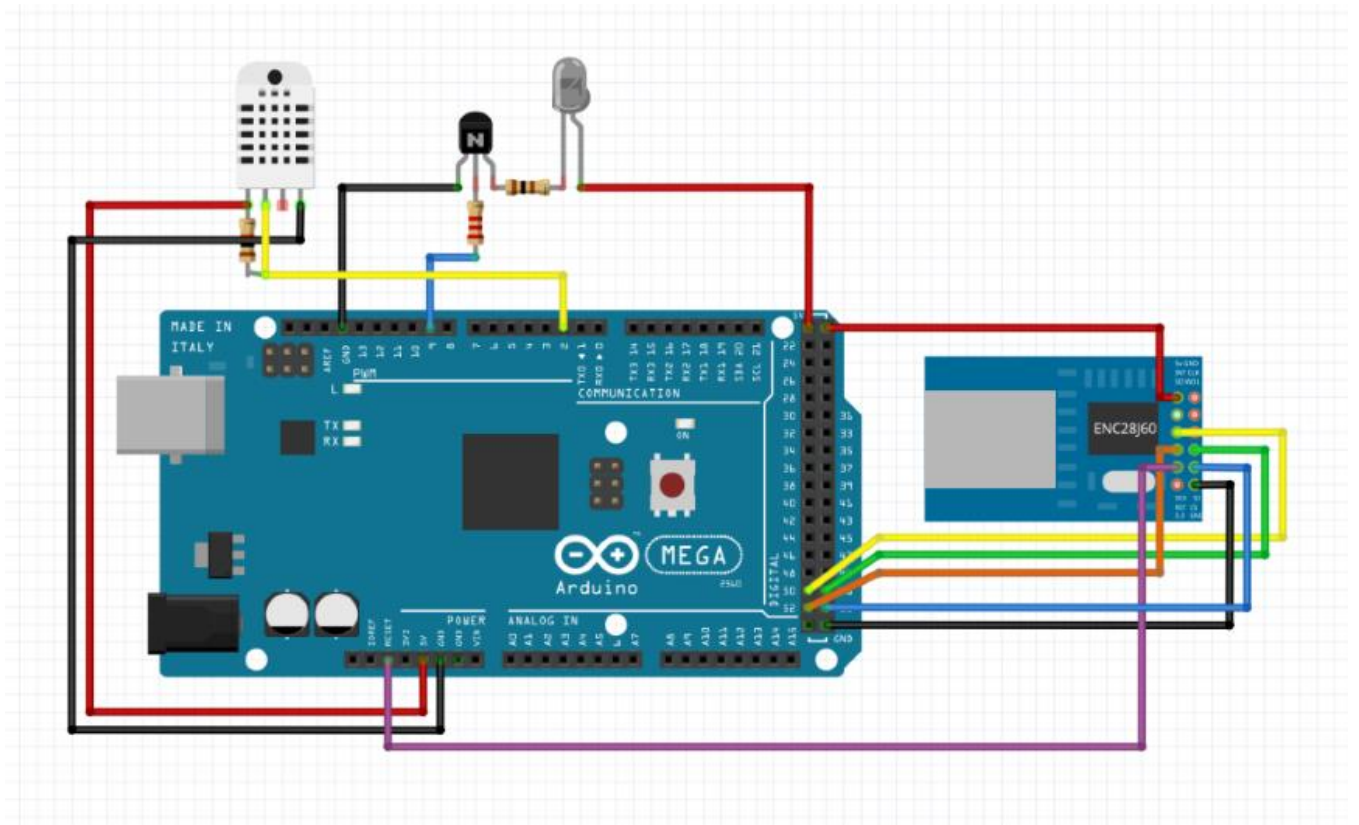


Рисунок 3.9 — Схема з'єднання елементів

Програмні коди показано у відповідних додатках.

ВИСНОВКИ

У процесі виконання роботи було розглянуто основні особливості розробки середовища Arduino, вивчено різні підходи, використовувані при створенні інтерфейсів користувача, а також методи і технології необхідні для їх створення. Проаналізовано найбільш відомі та популярні на сьогоднішній день системи «розумного будинку», мобільні програмні програми, які традиційно використовуються для віддаленого управління системами «розумного будинку». Крім того, виконано аналіз плат Arduino для найточнішого вибору плати для нашої системи. Розроблено вимоги до інтерфейсу проектного програми.

Виконано створення базового варіанта системи «розумного будинку», призначеного для управління електронними системами будинку, проведено його налагодження та тестування.

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

Таким чином, вимоги технічного завдання були виконані, задачі дипломного проектування вирішені, отже мету проекту досягнуто.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Історія створення Arduino. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу https://arduino-ua.com/art2-arduino_istoriya_sozdaniya - Загл. З екрану
2. Arduino. [Електронний ресурс]. - Електрон. зображення. - Режим доступу https://arduino-ua.com/images/letters/arduino_about1.jpg - Загл. З екрану
3. Як вибрати та купити Arduino. [Електронний ресурс]. Електрон. Текстові дані. - Режим доступу <http://edurobots.ru/2014/07/kak-vybrat-i-kupit-arduino-nachinayushhemu/> - Загл. З екрану
4. Що таке Arduino "шилд". [Електронний ресурс]. Електрон. Текстові дані – Режим доступу http://zelectro.cc/what_is_%D0%B0rduino_shield - Загл. З екрану
5. Arduino mini. [Електронний ресурс]. Електрон. Текстові дані. - Режим доступу <http://arduino.ru/Hardware/ArduinoBoardProMini> - Загл. З екрану
6. Arduino nano. [Електронний ресурс]. Електрон. Текстові дані. - Режим доступу <http://arduino.ua/ru/hardware/Nano> - Загл. З екрану
7. Arduino uno. [Електронний ресурс]. Електрон. Текстові дані. - Режим доступу <http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:arduino-uno> - Загл. З екрану
8. Arduino Leonardo. [Електронний ресурс]. Електрон. Текстові дані. - Режим доступу <http://arduino.ru/Hardware/ArduinoBoardLeonardo> - Загл. З екрану
9. Arduino Mega. [Електронний ресурс]. Електрон. Текстові дані. - Режим доступу <http://arduino.ru/Hardware/ArduinoBoardMega> - Загл. З екрану
10. Історія створення компанії ОВЕН. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <http://www.owen.ru/istoriya> - Загл. З екрану
11. ОВЕН ПЛК 154. [Електронний ресурс]. - Електрон. Зображення. - Режим доступу http://www.owen.ru/uploads/_thumbs/plk154_1xx_x200x167xxx.jpg - Загл. З екрану
12. Історія створення компанії Vimar Ву-me. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <http://vimar.ru/about/history/sozdanie/> - Загл. З екрану

13. Система Ву-ме. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <http://www.orens.com.ua/sistema-Vy-me.html> - Загл. З екрану
14. Управління пристроями автоматики. [Електронний ресурс]. - Електрон. Зображення. - Режим доступу <http://vimar.spb.ru/media/eikonserii/4655.png>
- Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <http://www.legrand.ru/> - Загл. З екрану
16. Система шинної організації. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <http://www.buderus-salon.zp.ua/stati/avtomatizatsiya/obzor-sistemyi-umnyiu-dom-my-home-bticino> - Загл. З екрану
17. Огляд системи «Розумний дім» Ву-ме [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу http://www.besmart.su/article/statya_4
18. Модульні прилади Elko ep. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <http://www.elkoep.ru/> - Загл. З екрану
19. Elko ep. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <http://www.beng.ru/elko> - Загл. З екрану
20. Сенсорна панель iTP iNels. [Електронний ресурс]. - Електрон. Зображення. - Режим доступу <http://www.elkoep.ru/produkcija/inels-bus-system/sensornye-paneli/shirokoformatnye/sensornaja-panel-itp-1902/> - Загл. З екрану
21. Сенсорна панель iNels. [Електронний ресурс]. - Електрон. Зображення. - Режим доступу <http://www.elkoep.ru/typo3temp/pics/4052aceea5.jpg> - Загл. З екрану
22. Тарифи використання «розумного будинку» від Мегафону [Електронний ресурс]. - Електрон. Зображення. - Режим доступу <http://www.mobile-review.com/articles/2017/image/life-control/1.jpg> - Загл. З екрану
23. Проектування розумних будинків. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <https://geektimes.ru/company/gsgroup/blog/277416/> - Загл. З екрану
24. Розумний дім. Все починається з дротів. [Електронний ресурс]. - Електрон. Текстові дані. - Режим доступу <https://geektimes.ru/post/268850/> - Загл. З екрану

ДОДАТОК А

Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

_____ О. Д. Азаров

“__” _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи

«Розумний дім з web-інтерфейсом»

08-23.БДР.022.00.000 ТЗ

Науковий керівник к.т.н., доц. каф. ОТ

_____ Кадук О.В.

Студент групи 1КІ-186

_____ Гаврилюк О.С.

Вінниця 2022

1. Найменування та область застосування

Розумний дім з web-інтерфейсом. Системи для обслуговування побутових та промислових приміщень.

2. Основи для розробки

Необхідність реалізації системи віддаленого управління через безпроводні мережі та Інтернет-з'єднання.

3. Мета та призначення розробки

Створення автономної системи управління з віддаленим доступом через безпроводні мережі та Інтернет-з'єднання, яка керується через web-інтерфейс.

4. Етапи БДР та очікувані результати

Робота виконується в п'ять етапів, що наведені в таблиці 4.1.

Таблиця 4.1 – Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	14.02.20	14.02.20	Вступ
2	Огляд і аналіз сучасних систем «розумного будинку»	15.02.20	28.02.20	Розділ 1
3	Проектування системи «розумний будинок»	01.03.20	14.03.20	Розділ 2
4	Реалізація програмно-апаратного комплексу «розумний будинок»	15.03.20	28.4.20	Розділ 3
5	Оформлення пояснювальної записки	29.04.20	17.05.20	ПЗ, презентація

5. Матеріали, що подаються до захисту БДР

Пояснювальна записка БДР, графічні і ілюстративні матеріали, протокол попереднього захисту БДР на кафедрі, відзив наукового керівника, рецензія опонента, протоколи складання державних екзаменів, анотації до БДР

українською та іноземною мовами, довідка про відповідність оформлення БДР діючим вимогам.

6. Порядок контролю виконання та захисту БДР

Виконання етапів графічної та розрахункової документації БДР контролюється науковим керівником згідно зі встановленими термінами. Захист БДР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

7. Вимоги до оформлення БДР

Вимоги викладені в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008-95, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

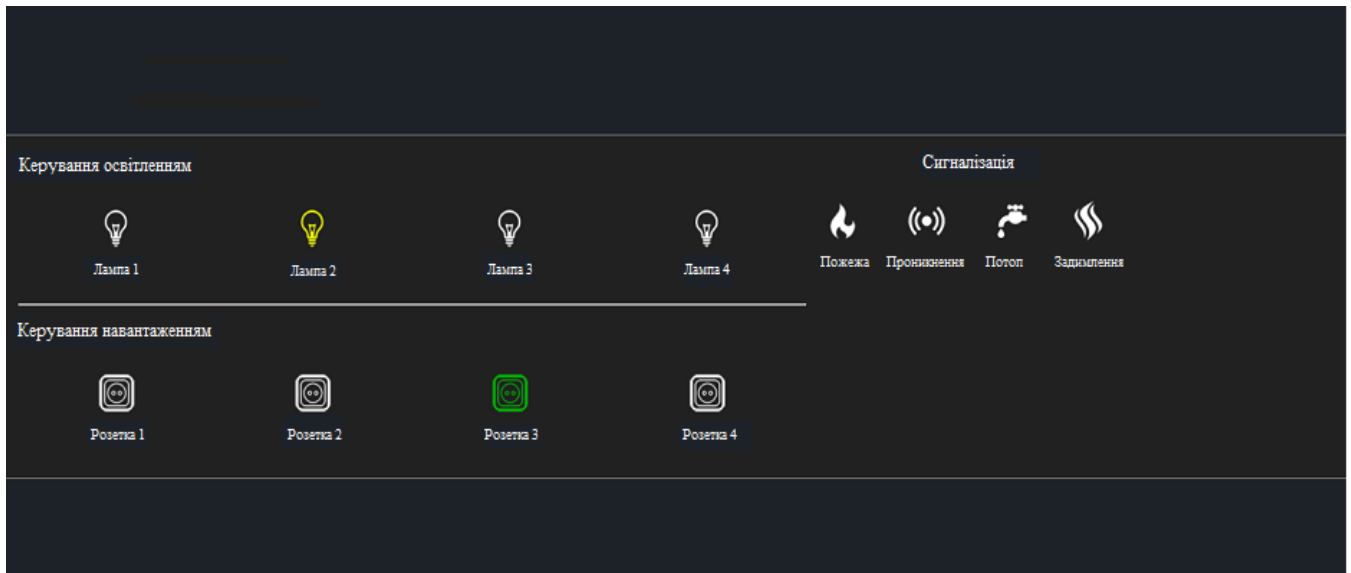
8. Вимоги щодо технічного захисту інформації в БДР з обмеженим доступом

Відсутні.

Технічне завдання до виконання отримав _____ Гаврилюк О.С.

ДОДАТОК Б

Сторінка web-сервісу з індикацією



Приклад додавання IP-адреси.

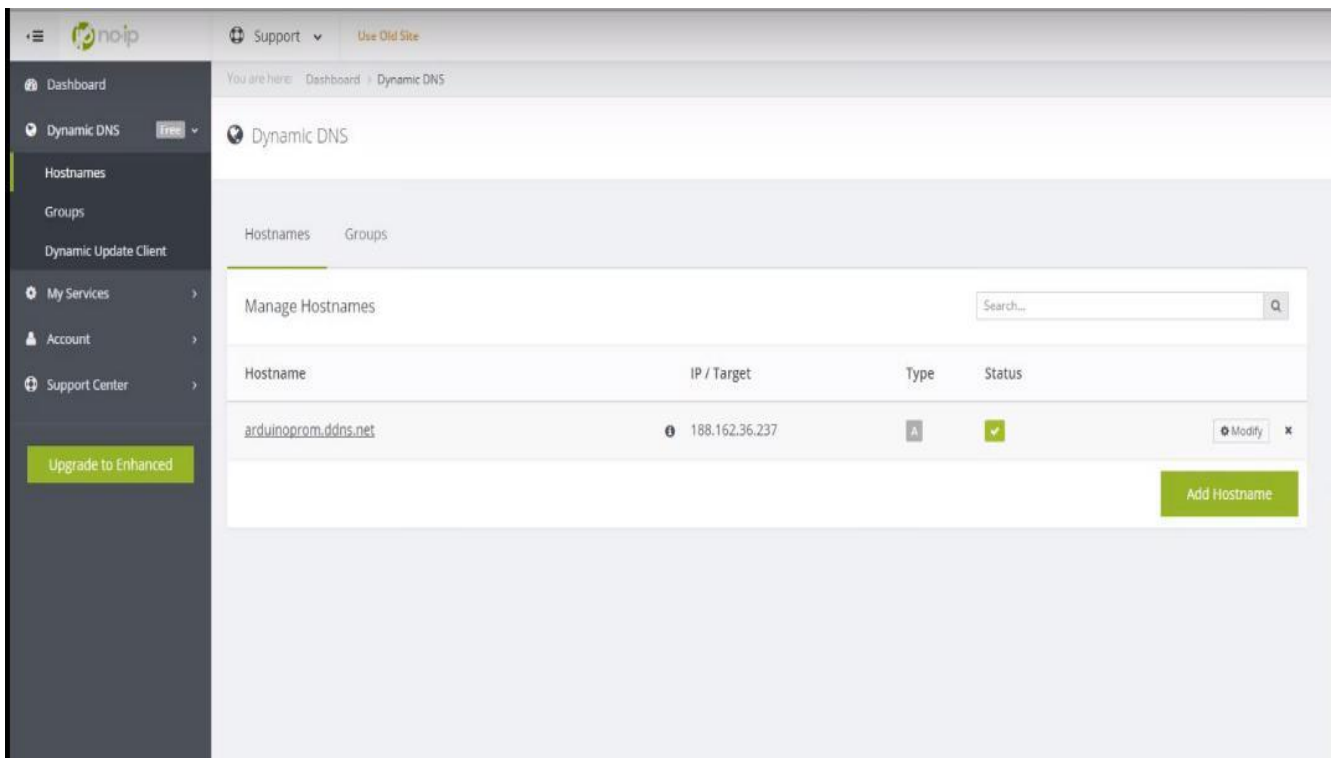
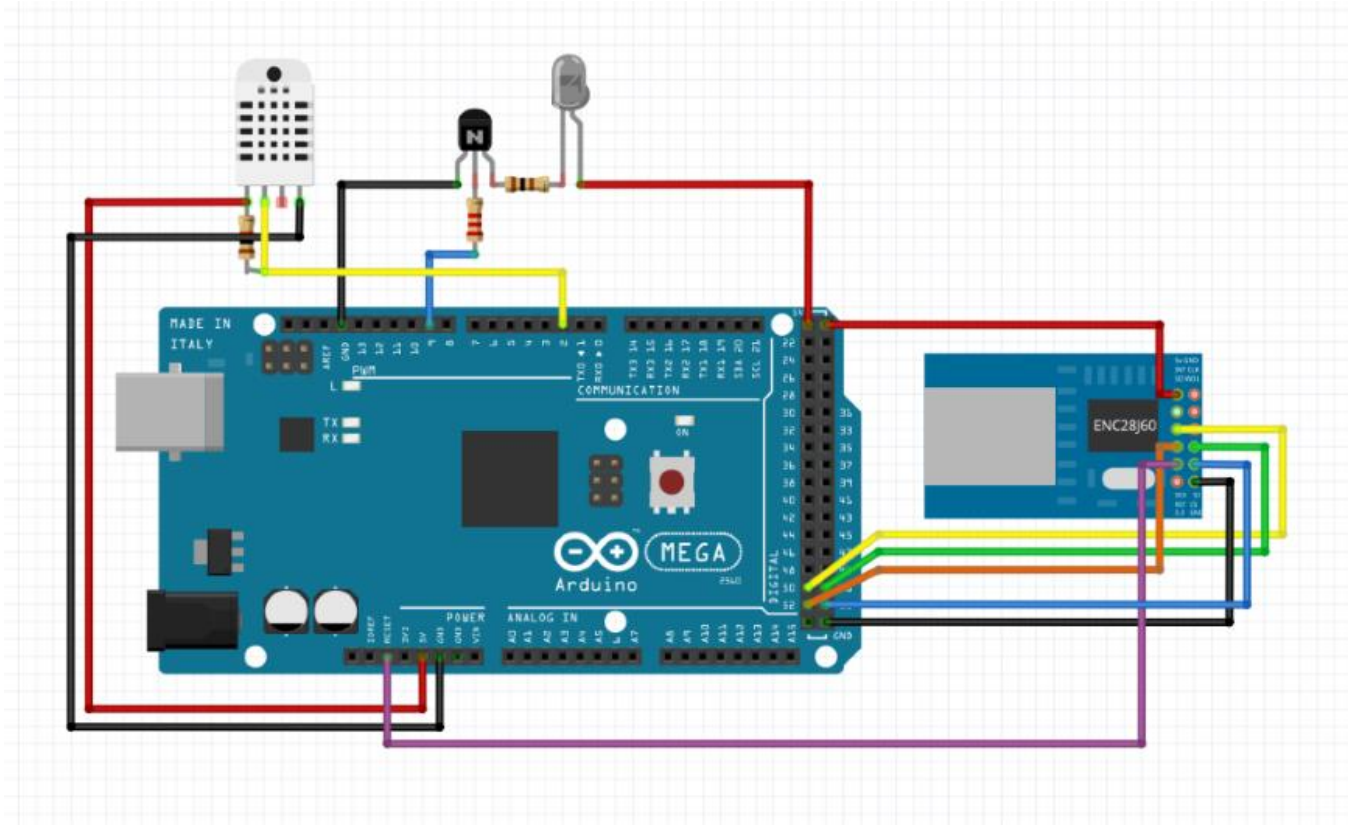


Схема з'єднання елементів



Код програми «Розумний дім» з 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 _trgrt1l = 0;  
bool _trgt1 = 0;  
bool _trgt1l = 0;  
bool _trgrt2 = 0;  
bool _trgrt2l = 0;  
bool _trgt2 = 0;  
bool _trgt2l = 0;  
bool _trgrt3 = 0;  
bool _trgrt3l = 0;  
bool _trgt3 = 0;  
bool _trgt3l = 0;  
bool _trgrt4 = 0;  
bool _trgrt4l = 0;  
bool _trgt4 = 0;  
bool _trgt4l = 0;  
bool _trgrt5 = 0;  
bool _trgrt5l = 0;  
bool _trgt5 = 0;  
bool _trgt5l = 0;  
bool _trgrt6 = 0;  
bool _trgrt6l = 0;  
bool _trgt6 = 0;  
bool _trgt6l = 0;  
bool _trgrt7 = 0;  
bool _trgrt7l = 0;  
bool _trgt7 = 0;  
bool _trgt7l = 0;  
bool _trgrt8 = 0;  
bool _trgrt8l = 0;  
bool _trgt8 = 0;  
bool _trgt8l = 0;  
bool _trgrt9 = 0;  
bool _trgrt9l = 0;  
bool _trgt9 = 0;  
bool _trgt9l = 0;  
bool _trgrt10 = 0;  
bool _trgrt10l = 0;  
bool _trgt10 = 0;  
bool _trgt10l = 0;  
bool _trgrt11 = 0;  
bool _trgrt11l = 0;  
bool _trgt11 = 0;
```

```
bool _trgt11l = 0;
bool _trgt12 = 0;
bool _trgt12l = 0;
bool _trgt12 = 0;
bool _trgt12l = 0;
bool _trgt13 = 0;

bool _trgt13l = 0;
bool _trgt13 = 0;
bool _trgt13l = 0;
bool _trgt14 = 0;
bool _trgt14l = 0;
bool _trgt14 = 0;
bool _trgt14l = 0;
bool _WSP6_A1 = 0;
bool _WSP6_A2 = 0;
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 _bounselInputD49S = 0;
bool _bounselInputD49O = 0;
unsigned long _bounselInputD49P = 0UL;
bool _bounselInputD46S = 0;
bool _bounselInputD46O = 0;
unsigned long _bounselInputD46P = 0UL;
bool _bounselInputD48S = 0;
bool _bounselInputD48O = 0;
unsigned long _bounselInputD48P = 0UL;
bool _bounselInputD47S = 0;
bool _bounselInputD47O = 0;
unsigned long _bounselInputD47P = 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);
    _bounseInputD49O = digitalRead(49);
    _bounseInputD46O = digitalRead(46);
    _bounseInputD47O = digitalRead(47);
    _bounseInputD48O = digitalRead(48);

    Ethernet.begin(ethernet_mac, ethernet_ip, ethernet_dns, ethernet_gateway,
ethernet_subnet); delay(1000);
    _tspWebServer.begin();
}
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) {

            _WSCPgeNumber
            _parseWebServerReqest(_WSCReqest); _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));

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;

```

```

    _bounselInputD47P = millis();
  }

}
if (!(_bounselInputD46O))
{
  _swi4 = String("http://provuz22.ru/arduino/pozhar1.png");
}
else
{
  _swi4 = String("http://provuz22.ru/arduino/pozhar0.png");
}
if (!(_bounselInputD47O))
{
  _swi5 = String("http://provuz22.ru/arduino/signalizaciya1.png");
}
else

{
  _swi5 = String("http://provuz22.ru/arduino/signalizaciya0.png");
}
if (!(_bounselInputD48O))
{
  _swi1 = String("http://provuz22.ru/arduino/voda1.png");
}
else
{
  _swi1 = String("http://provuz22.ru/arduino/voda0.png");
}
if (!(_bounselInputD49O))
{
  _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/termostat1.png");
}
else

```

```
{
  _swi6 = String("http://provuz22.ru/arduino/thermostat0.png");
}
if (_gtv12)
{

  _swi7 = String("http://provuz22.ru/arduino/rozetka1.png");
}
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)
{

  _swi10 = String("http://provuz22.ru/arduino/rozetka1.png");
}
else
{
  _swi10 = String("http://provuz22.ru/arduino/rozetka0.png");
}
if (_gtv9)
{
  _swi11 = String("http://provuz22.ru/arduino/rozetka1.png");
}
else
{
  _swi11 = String("http://provuz22.ru/arduino/rozetka0.png");
}
if (_gtv4)
{
  _swi17 = String("http://provuz22.ru/arduino/lampochka1.png");
}
else
{
  _swi17 = String("http://provuz22.ru/arduino/lampochka0.png");
}
if (_gtv3)
```

```

{
  _swi18 = String("http://provuz22.ru/arduino/lampochka1.png");
}
else

{
  _swi18 = String("http://provuz22.ru/arduino/lampochka0.png");
}
if (_gtv2)
{
  _swi19 = String("http://provuz22.ru/arduino/lampochka1.png");
}
else

{
  _swi19 = String("http://provuz22.ru/arduino/lampochka0.png");
}
if (_gtv1)
{
  _swi20 = String("http://provuz22.ru/arduino/lampochka1.png");
}
else
{
  _swi20 = String("http://provuz22.ru/arduino/lampochka0.png");
}
if (_WSP6_A13) {
  if (_trgrt1l) {
    _trgrt1 = 0;
  } else {
    _trgrt1 = 1;
    _trgrt1l = 1;
  }
} else {
  _trgrt1 = 0;
  _trgrt1l = 0;
};
bool _tmp1 = _trgrt1;
if (_tmp1) {
  if(!_trgrt1l) _trgrt1 = !_trgrt1;
}

_trgrt1l = _tmp1;
if (_WSP6_A1) {
  if (_trgrt2l) {
    _trgrt2 = 0;
  } else {

```

```

        _trgrt2 = 1;
        _trgrt2l = 1;
    }
} else {
    _trgrt2 = 0;
    _trgrt2l = 0;
};
bool _tmp2 = _trgrt2;
if (_tmp2) {
    if (!_trgrt2l) _trgrt2 = !_trgrt2;
}

_trgrt2l = _tmp2;
if (_WSP6_A2) {

    if (_trgrt3l) {
        _trgrt3 = 0;
    } else {
        _trgrt3 = 1;
        _trgrt3l = 1;
    }
} else {
    _trgrt3 = 0;

    _trgrt3l = 0;
};
bool _tmp3 = _trgrt3;
if (_tmp3) {
    if (!_trgrt3l) _trgrt3 = !_trgrt3;
}

_trgrt3l = _tmp3;
if (_WSP6_A3) {
    if (_trgrt4l) {
        _trgrt4 = 0;
    } else {
        _trgrt4 = 1;
        _trgrt4l = 1;
    }
} else {
    _trgrt4 = 0;
    _trgrt4l = 0;

};
bool _tmp4 = _trgrt4;
if (_tmp4) {
    if (!_trgrt4l) _trgrt4 = !_trgrt4;
}

```

```

    }

    _trgt4l = _tmp4;
    if (_WSP6_A4) {
        if (_trgrt5l) {
            _trgt5 = 0;
        } else {
            _trgt5 = 1;
            _trgrt5l = 1;
        }
    } else {
        _trgt5 = 0;
        _trgrt5l = 0;
    };
    bool _tmp5 = _trgt5;
    if (_tmp5) {
        if(!_trgrt5l) _trgt5 = ! _trgt5;
    }

    _trgt5l = _tmp5;
    if (_WSP6_A5) {
        if (_trgrt6l) {
            _trgt6 = 0;
        } else {
            _trgt6 = 1;

            _trgrt6l = 1;
        }
    } else {
        _trgt6 = 0;
        _trgrt6l = 0;
    };
    bool _tmp6 = _trgt6;
    if (_tmp6) {

        if (!_trgrt6l) _trgt6 = ! _trgt6;
    }

    _trgt6l = _tmp6;
    if (_WSP6_A6) {
        if (_trgrt7l) {
            _trgt7 = 0;
        } else {
            _trgt7 = 1;
            _trgrt7l = 1;
        }
    } else {

```



```

    _trgrt7 = 0;
    _trgrt7l = 0;
};
bool _tmp7 = _trgrt7;
if (_tmp7) {
    if (!_trgrt7l) _trgrt7 = !_trgrt7;
}

_trgrt7l = _tmp7;
if (_WSP6_A7) {
    if (_trgrt8l) {
        _trgrt8 = 0;
    } else {
        _trgrt8 = 1;
        _trgrt8l = 1;
    }
} else {
    _trgrt8 = 0;
    _trgrt8l = 0;
};
bool _tmp8 = _trgrt8;
if (_tmp8) {
    if (!_trgrt8l) _trgrt8 = !_trgrt8;
}

_trgrt8l = _tmp8;
if (_WSP6_A8) {
    if (_trgrt9l) {
        _trgrt9 = 0;
    } else {
        _trgrt9 = 1;
        _trgrt9l = 1;
    }
} else {
    _trgrt9 = 0;

    _trgrt9l = 0;
};
bool _tmp9 = _trgrt9;
if (_tmp9) {
    if (!_trgrt9l) _trgrt9 = !_trgrt9;
}

_trgrt9l = _tmp9;
if (_WSP6_A9) {

```

```

if (_trgrt10l) {
    _trgrt10 = 0;
} else {
    _trgrt10 = 1;
    _trgrt10l = 1;
}
} else {
    _trgrt10 = 0;
    _trgrt10l = 0;
};
bool _tmp10 = _trgrt10;
if (_tmp10) {
    if (!_trgrt10l) _trgrt10 = !_trgrt10;
}
_trgrt10l = _tmp10;
if (_WSP6_A10) {
if (_trgrt11l) {

_trgrt11 = 0;
} else {
    _trgrt11 = 1;
    _trgrt11l = 1;
}
} else {
    _trgrt11 = 0;
    _trgrt11l = 0;
};
bool _tmp11 = _trgrt11;
if (_tmp11) {
    if (!_trgrt11l) _trgrt11 = !_trgrt11;
}
_trgrt11l = _tmp11;
if (_WSP6_A11) {
if (_trgrt12l) {
_trgrt12 = 0;
} else {
    _trgrt12 = 1;
    _trgrt12l = 1;
}
} else {
    _trgrt12 = 0;
    _trgrt12l = 0;
};
bool _tmp12 = _trgrt12;
if (_tmp12) {

    if (!_trgrt12l) _trgrt12 = !_trgrt12;

```

```

    }
    _trgt12l = _tmp12;
    if (_WSP6_A12) {
if (_trgt13l) {
    _trgt13 = 0;
    } else {
        _trgt13 = 1;

        _trgt13l = 1;
    }
    } else {
        _trgt13 = 0;
        _trgt13l = 0;
    };
    bool _tmp13 = _trgt13;
    if (_tmp13) {
        if(!_trgt13l) _trgt13 = !_trgt13;
    }
    _trgt13l = _tmp13;
    if (_WSP6_A14) {
if (_trgt14l) {
    _trgt14 = 0;
    } else {
        _trgt14 = 1;
        _trgt14l = 1;

    }
    } else {
        _trgt14 = 0;
        _trgt14l = 0;
    };
    bool _tmp14 = _trgt14;
    if (_tmp14) {
        if (!_trgt14l) _trgt14 = !_trgt14;
    }
    _trgt14l = _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);
}
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\"\">");
    _tspWebServer_client.println("<html><head>");

    _tspWebServer_client.println("<META content=\"\"text/html; charset=utf-8\"\"
http-equiv=\"\"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;
    } 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") {
    _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 _sendWebServerPage1(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"">");
    _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"">Пожежа</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"">Проникнення</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"">Задимлення</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("</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.pri
ntln("");
    _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>");
        _tspWebServer_client.println("<td align=""top"">");
        bgcolor=""212121"" width=""12%""

        _tspWebServer_client.println("</td>");
        _tspWebServer_client.println("<td align=""top"">");
        bgcolor=""212121"" width=""48%""

        _tspWebServer_client.println("</td>");
        _tspWebServer_client.println("");
        _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"">Управління освітленням</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(_swi19);
_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>");
_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(_swi18);
_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(_swi17);
_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"">Управління навантаженням</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(_swi11);
_tspWebServer_client.println("""></a></p>");

_tspWebServer_client.println("<p><font      color=""FFFFFF""      size=""3""
face=""Open Sans"">Позетка 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=""/10""><img src="">");

_tspWebServer_client.println(_swi10);
_tspWebServer_client.println("""></a></p>");

_tspWebServer_client.println("<p><font      color=""FFFFFF""      size=""3""
face=""Open Sans"">Позетка 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(");
_tspWebServer_client.println(");
_tspWebServer_client.println(");
_tspWebServer_client.println(");
_tspWebServer_client.println(");

```

```
_tspWebServer_client.println("<p><font      color=""FFFFFF""      size=""4""  
face=""Open Sans"">Сигналізації</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=""<br>    } void _sendWebServerPage6(void)  
    {  
  
_tspWebServer_client.println("<meta      http-  
equiv=""refresh""  
content=""0;URL=http://192.168.10.101"">");  
  
    }
```


ДОДАТОК В

Код для платформи Arduino

```

#include <SPI.h> //Бібліотека для роботи з SPI
#include <UIPEthernet.h> //Бібліотека для роботи з Ethernet
#include <dht.h> //Бібліотека для роботи з DHT22
#include <IRremote.h> //Бібліотека для роботи з IR

#define DHT22_PIN 2

//ініціалізація бібліотеки dht
dht DHT;

int dht_status = 0;

float MAX_TEMP = 25.0;
float MIN_TEMP = 24.0;
int last_condey_status = 0;

boolean newInfo = 0; //Змінна для нової інформації
//MAC адресу вашого Ethernet-модуля, якщо його у вас немає, введіть будь-яку
//або залиште той, що на прикладі
byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDA, 0x02};

//вказуємо IP адресу
//будьте уважні, якщо адреса зайнята іншим пристроєм, то нічого не буде працювати
IPAddress ip(x, x, x, x);

//ініціалізація бібліотеки Ethernet server library
EthernetServer server(80);

560, 540, 560, 540, 540, 540, 580, 540, 560, 520, 600, 520, 1660, 540, 1680, 540, 560,
520, 170, 5, 5 520, 1700, 540, 540, 540, 580, 540, 1660, 520, 600, 520, 1660, 540, 1680,
540, 1680, 540}; // DefaultOn (21C + Auto)
524, 572, 520, 1624, 516, 57, 5, 5, 5, 5, 5, 5 580, 516, 552, 520, 576, 520, 1636, 504,
1652, 512, 1624, 520, 1652, 512, 1652, 512}; // Off
540, 580, 500, 580, 520, 580, 540, 560, 540, 560, 560, 560, 1640, 540, 1680, 540, 1680,
540, 5 5 560, 1640, 540, 580, 540, 560, 540, 540, 540, 1700, 520, 560, 540, 1680, 540,
1680, 540}; // 22C + Dry
560, 560, 540, 560, 520, 560, 580, 540, 540, 560, 560, 540, 560, 540, 1680, 540, 1660,
520, 580, 5 5 520, 1700, 520, 1660, 540, 580, 540, 540, 540, 1680, 540, 1660, 540, 1680,
540, 1680, 540}; // 19C + Cool + Fan High
1604, 532, 564, 532, 1608, 528, 568, 532, 540, 524, 1644, 520, 1644, 524, 1616, 520, 576,
520, 5 5 520, 576, 520, 552, 516, 580, 516, 1640, 500, 1672, 496, 1644, 496, 1660, 504,
1668, 500}; // Swing

```

```

int freq = 38;

//Ініціалізація бібліотеки irsend
IRsend irsend;

unsigned long currentMillis;
unsigned long previousMillis = 0;
const long interval = 4000;

int set_auto_mark = 0;

void SetAutoMode(float temp, int trigger)
{
int status = 0;
if (trigger == 1)
{
Serial.println("Trigger is on");

if (temp > MAX_TEMP) {
status = 1;
Serial.print("Condition status STAT1 is:");
Serial.println(status);
}
if (temp <MIN_TEMP) {
status = 2;
Serial.print("Condition status STAT2 is:");
Serial.println(status);
}

if (status!= last_condey_status) {
last_condey_status = status;
if (status == 2) {
irsend.sendRaw(OffMode, sizeof(OffMode) / sizeof(OffMode[0]), freq);
}
if (status == 1) {
irsend.sendRaw(ComfortMode, sizeof(ComfortMode) / sizeof(ComfortMode[0]), freq);
delay(50);
irsend.sendRaw(SwingMode, sizeof(SwingMode) / sizeof(SwingMode[0]), freq);
}
}
Serial.print("Condition status is:");
Serial.println(status);
}
else {
Serial.println("Trigger is off");
}
}
}

```

```

void setup()
{
  Serial.begin(9600);
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Server is at");
  Serial.println(Ethernet.localIP());
}

void loop()
{
  currentMillis = millis();
  if (currentMillis - previousMillis >= interval)
  {
    Serial.println("status,\tHumidity (%),\tTemperature (C)");
    // зчитуємо значення датчика DHT22
    int chk = DHT.read22(DHT22_PIN);
    switch (chk)
    {
      case DHTLIB_OK:
        dht_status = 200;
        Serial.print("OK,\t");
        break;
      case DHTLIB_ERROR_CHECKSUM:
        dht_status = 501;
        Serial.print("Checksum error,\t");
        break;
      case DHTLIB_ERROR_TIMEOUT:
        dht_status = 504;
        Serial.print("Time out error,\t");
        break;
      default:
        dht_status = 500;
        Serial.print("Unknown error,\t");
        break;
    }
    // Відображаємо значення з датчика DHT22
    Serial.print(DHT.humidity, 1);
    Serial.print(",\t");
    Serial.println(DHT.temperature, 1);
    previousMillis = currentMillis;
  }

  SetAutoMode(DHT.temperature, set_auto_mark);

  //Приймаємо дані, що посилаються клієнтом
  EthernetClient client = server.available();

```

```

if (client) { //якщо запит закінчується порожнім рядком
boolean currentLineIsBlank = true; //ставимо мітку про закінчення запиту (дослівно:
поточна лінія чиста)
while (client.connected()) { // поки є з'єднання з клієнтом
if (client.available()) { //якщо клієнт активний
char c = client.read(); //зчитуємо інформацію, що посилається, в змінну "с"
if (newInfo && c == ' ') { //якщо змінна нової інформації = 1 і "с", в якій записаний запит,
дорівнює порожньому рядку
NewInfo = 0; //то обнулюємо змінну надходження нової інформації
}
if (c == '$') { //якщо змінна "з", що несе відправлений нам запит, містить символ $
//(всі нові запити) - "$" має на увазі поділ одержуваної інформації (символів)
NewInfo = 1; //то прийшла нова інформація, ставимо мітку нової інформації в 1
}

//Перевіряємо вміст URL - є $1 або $2
if (newInfo == 1) { //якщо є нова інформація
Serial.println(c);
if (c == '1') { //і "с" містить 1
Serial.println("ON");
irsend.sendRaw(ComfortMode, sizeof(ComfortMode) / sizeof(ComfortMode[0]), freq);
set_auto_mark = 0;
}

if (c == '2') {
Serial.println("OFF");
irsend.sendRaw(OffMode, sizeof(OffMode) / sizeof(OffMode[0]), freq);
set_auto_mark = 0;
}

if (c == '3') {
Serial.println("AUTO");
set_auto_mark = 1;
}

if (c == '4') {
Serial.println("VITALY");
irsend.sendRaw(VitalyMode, sizeof(VitalyMode) / sizeof(VitalyMode[0]), freq);
set_auto_mark = 0;
}

if (c == '5') {
Serial.println("COMFORT");
irsend.sendRaw(ComfortMode, sizeof(ComfortMode) / sizeof(ComfortMode[0]), freq);
set_auto_mark = 0;
}
}
}

```

```

if (c == '\n') { //якщо "с" дорівнює символу нового рядка
currentLinsBlank = true; //то починаємо новий рядок
}
else if (c != '\r') { // інакше, якщо "с" не дорівнює символу повернення курсору на
початок рядка
currentLinsBlank = false; //то отримуємо символ на поточному рядку
}

if (c == '\n' && currentLinsBlank) { //виводимо HTML сторінку
client.println("HTTP/1.1 200 OK"); //заголовна інформація
client.println("Content-Type: text/html");
client.println("Connection: close");
//client.println("Refresh: 30"); //автоматичне оновлення кожні 30 с
client.println();
client.print("<!DOCTYPE html>");
client.print("<html>");
client.print("<head>");
client.print("<link href='\"https://fonts.googleapis.com/css?family=Jura\"'
rel='\"stylesheet\"'>");
client.print("<meta http-equiv='\"Content-Type\"' content='\"text/html; charset=utf-8\">");
client.print("<meta http-equiv='\"refresh\"' content='\"5; url=http://xxxx^\">");
client.print("<meta name='\"viewport\"' content='\"width=device-width, initial-scale=1.0\">");
client.print("<style type='\"text/css\"'");
client.print("* { margin: 0; padding: 0; font-family: 'Jura', sans-serif;}");
client.print("h1 {margin-left: 15px; font-size: 26px;}");
client.print("div {border-radius: 3px; background: #51A8FF; color: white; text-align: center;
padding-top: 5px; padding-bottom: 5px; width: 180px; margin-left: 15px; font-size: 22px;}");
client.print(".button {border-radius: 3px; border: none; color: white; text-align: center;
margin-bottom: 5px; width: 88px; height: 32px; margin-left: 15px; text-decoration: none;
display: inline-block; font-size: 16px; -webkit-transition-duration: 0.4s; transition-duration:
0.4s; cursor: pointer;}");
client.print(".button:hover {background-color: white; color: black;}");
client.print(".button1 {background-color: #4CAF50; border: 2px solid #4CAF50;}");
client.print(".button2 {background-color: #f44336; border: 2px solid #f44336; margin-left:
1px;}");
client.print(".button3 {background-color: #808080; border: 2px solid #808080; width:
177px;}");
client.print(".button4 {background-color: #004CFF; border: 2px solid #004CFF;}");
client.print(".button5 {background-color: #FF9000; border: 2px solid #FF9000; margin-left:
1px;}");
client.print("@media only screen and (max-width: 425px) {");
client.print("h1{text-align: center;}");
client.print("div {width: 92%; padding-top: 25px; padding-bottom: 25px;}");
client.print(".button {width: 45%; height: 45px;}");
client.print(".button3 {width: 91%; height: 45px;}");
client.print("}");
client.print("</style>");

```

```

client.print("</head>");
client.print("<body>");
client.print("<title>Air-Conditioning Dashboard</title>");
client.print("<h1>Кабінет №000</h1>");
client.print("<br />");
client.print("<div>");
client.print("<p>Температура");
client.print("<br />");
client.print(DHT.temperature, 1);
client.print("C<sup>0</sup>");
client.print("</p>");
client.print("</div>");
client.print("<br />");
client.print("<div>");
client.print("Вологість");
client.print("<br />");
client.print(DHT.humidity, 1);
client.print("%");
client.print("<br />");
client.print("</p>");
client.print("</div>");
client.print("<br /><br />");
client.print("<a href='\"/$1\"'><button class='button button1'>ВКЛ</button></a>");
client.print("<a href='\"/$2\"'><button class='button button2'>ВИКЛ</button></a>");
client.print("<br />");
client.print("<a href='\"/$3\"'><button class='button button3'>АВТО</button></a>");
client.print("<br />");
client.print("<a href='\"/$4\"'><button class='button button4'>ХОЛОД</button></a>");
client.print("<a href='\"/$5\"'><button class='button button5'>КОМФОРТ</button></a>");
client.print("</body>");
client.print("</html>");
break; //Вихід
}

}

}

delay(1); //час отримання нових даних
client.stop(); //закриваємо з'єднання
}

}

```

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розумний дім з web-інтерфейсом

Тип роботи: бакалаврська дипломна робота
(Б.ДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 81,6% Схожість 18,4%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Гаврилюк О.С.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Кадук О.В.
(підпис) (прізвище, ініціали)

