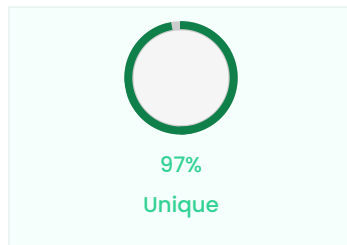
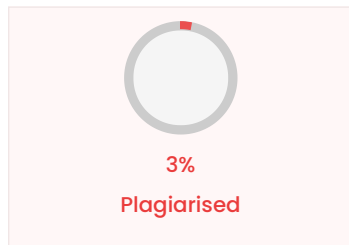


Plagiarism Scan Report



Words Statistics

Words	10040
Characters	78828

Exclude URL: None

Content Checked For Plagiarism

З МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИЙ ТЕХНОЛОГІЙ» Кафедра комп'ютерної інженерії ДО ЗАХИСТУ ДОПУЩЕНА Зав. кафедрою _____ ДИПЛОМНА РОБОТА МІКРОПРОЦЕСОРНА СИСТЕМА УПРАВЛІННЯ ЖИТЛОВИМ КОМПЛЕКСОМ. ПРОГ _____ М.С. Змієвський Керівник професор _____ С.О. Сабанов Запоріжжя 2022 ЗАКЛАД «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ» Кафедра комп'ютерної іі професор А.В. Переверзєв _____ 17.01.2022 р. **З А В Д А Н Н Я** НА БАКАЛАВРСЬКУ ДИПЛОМНУ «Комп'ютерна інженерія» Змієвському Максиму Сергійовичу 1.Тема: Мікропроцесорна система управління комплек по інституту 06.1-50 від 15 січня 2022р. 2. Термін здачі студентом закінченої роботи: 17 червня 2022 року 3. Пере літературу, присвячену темі розробки. 2. Розглянути загальні відомості про загальну концепцію та системи розумної Home. 4. Навести приклади реалізації Smart Home, що присутні на ринку. 5. Ознайомитися з технологіями прог готовими рішеннями, що використовуються на практиці 6. Провести аналіз принципової схеми модуля розумного управління. 7. Виконати вибір інструментальних засобів для створення ПЗ комплексу. 8. Ознайомитись з серед розробити алгоритм функціонування системи та реалізувати його програмно. 9. Відлагодити створену програму і роботи у вигляді звіту. Дата видачі завдання: 18 січня 2022 р. Керівник бакалаврської роботи _____ (_____ М.С. Змієвський РЕФЕРАТ Бакалаврська дипломна робота: 77 сторінок, 22 малюнків, 20 пей системи з мікроконтролерним управлінням. Мета роботи: створення програмного забезпечення для керування під базі мікроконтролера. У бакалаврській роботі розглянуто концепцію розумних житлових комплексів, основні техно основних типів IoT-речей. Наведено принципи роботи основних підсистем розумних будинків та виконано огляд наї на ринку. Запропоновано розробку, що використовує різні апаратні частини, які взаємодіють з сервером з викори даних, Node.js для взаємодії з базою даних, та база даних mongodb. ANGULAR, ARDUINO, API, IOT, MONGOD ФРЕЙМВОРК ЗМІСТ ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, СИМВОЛІВ ТА ТЕРМІНІВ 6 ВСТУП 8 РОЗДІЛ 1 ОПИС Приклади використання 11 1.1.1 Smart-Light рішення 11 1.1.2 Smart Entertainment рішення 13 1.1.3 Розумна побутова і дому 17 1.2 Інтелектуальні IoT платформи 20 1.3 Додатки для роботи з стандартними системами IoT 23 1.3.1 Amazc Home 26 1.3.4 Apple Home 27 РОЗДІЛ 2 ВИБІР ЗАСОБІВ РОЗРОБКИ ТА РІШЕННЯ ЗАДАЧ ПРОЕКТУ 30 2.1 Основи побудови системи 34 2.1.2 Централізовані системи автоматизації Ошибка! Закладка не определена. 2.1.3 Дец Закладка не определена. 2.1.4 Змішана система керування Ошибка! Закладка не определена. 2.1.5 Способи побуд системи автоматизації Ошибка! Закладка не определена. 2.1.7 Бездротові системи автоматизації Ошибка! Закладка для написання системи управління розумним будинком 37 2.2.1 ReactJs 39 2.2.2 Vue.js 40 2.2.3 Angular 41 2.2.4 СИМВОЛІВ ТА ТЕРМІНІВ Слово/словосполучення Скорочення Умова використання М Мегагерц Мгц В тексті Г Гіг тексті А Alternating Current АС В тексті С Customer relationship management CRM В тексті D Direct Current Document Object Model DOM В тексті E Electrically Erasable Programmable Read-Only Memory EEPROM І Things IoT В тексті iPhone Operation System iOS В тексті L Light-emitting diode LED В тексті W Wireless визначення того, що означає термін Smart Home, не існує. Зазвичай, розумний будинок, або в ширшому розумінні, ·

термін для житлового середовища, в якому основні елементи життєзабезпечення, такі як освітлення, водопостачання побутові електроприлади, об'єднані в єдину мережу. Керування такою системою виконується через термінальні пр

На протязі десятиліть окремі операції – від натискання вимикача світла до відкриття дверей гаража за допомогою автоматизовані. Сама ж концепція бере свій початок ще в 1934 році на Всесвітній виставці в Чикаго, де було предст років автоматизований дім перетворився на розумний дім завдяки Інтернету, сучасним датчикам і засобам підключенням до Інтернету вже є розумним будинком. Мобільні пристрої, ноутбуки та телевізори з підключенням частинами середовища, наприклад, транслювати фільм з ноутбука на екран телевізора. Однак справжнє налашту взаємопов'язані, якими користувач може керувати прямо чи опосередковано. Таким чином, з поняттям розумних поняття – Інтернет речей. IoT – це багаторівнева система, що включає датчики і контролери, встановлені на вузлах збираються, та візуалізації цих даних, потужні аналітичні інструменти інтерпретації отриманої інформації та багато етап розвитку Інтернету, що значно розширює можливості збору, аналізу та розподілу даних, які людина може перебу будинку, IoT – це модулі, з яких створюються якісь додаткові внутрішні контури сучасного житла. Зокрема, у склад використовуються автоматичні (розумні) вимикачі, «розумні» розетки, камери з функцією відправлення та арх системи кондиціонування та вентиляції, тощо. Розумні житлові комплекси працюють за принципом введення, обр вихідні параметри можуть впливати на вхідні. Введення інформації може здійснюватися різними способами подач освітлення, жестами, голосом, тощо. Обробка виконується зазвичай централізовано в блоці управління сист з'єднуються за допомогою різноманітних каналів зв'язку, як кабельних, так і бездротових: WLAN, Bluetooth, ZigBee, ; житлових комплексів може бути досить різною. В простіших випадках, деякі розумні будинки можна побудувати комплекси базуються на потужних серверних платформах. Обладнання, що використовується, визначає вибір пр Програмне забезпечення повинно обслуговувати три основні типи пристроїв для Smart Home: * контролер (хаб системи один з одним і пов'язує її із зовнішнім світом; * сенсори (датчики), пристрої, які отримують інформацію і актуатори, пристрої, що відповідають безпосередньо за виконання команд. Дії обладнання, які воно повинно викона команди, визначаються сценаріями. Їх програмування має на увазі голосове, пультове управління та контроль з: розмова з розумним динаміком – це спосіб безпосередньо керувати розумною екосистемою. Прикладом непрямо розумного динаміка виявляє рух і миттєво вмикає світло, коли хтось проходить повз. В роботі наводиться опи житлового комплексу: * визначення етапів взаємодії внутрішніх складових сценарію з користувачем та між собою ситуації; * підбір або створення драйверів модулів системи та відповідного інтерфейсу для кожної групи пристр попередніх етапів. Зважаючи на складність виконання завдання зі створення комплексної програми, було програмування тільки однієї підсистеми. Був використаний фреймворк Angular для візуалізації даних, Node.js MongoDB. Реалізація проекту була виконана з врахуванням можливості поширення та масштабування, додаючи самим контролювати штучне освітлення приміщення, чи зробити автоматичну роботу штор на вікнах. Його можна екосистему оселі.

РОЗДІЛ 1 ОПИС ТЕХНОЛОГІЙ 1.1 Технології Smart Home. Приклади використання

На ринку є широкий спектр технологій розумного дому, від окремих пристроїв для комунальних послуг, розваг і зв'язку до об'єднати все це разом. Найголовніше, що робить цю технологію настільки важливою – це постійно зростаючий пк Light рішення Розумні технології інтегрували майже всі аспекти домашнього життя, включаючи освітлення. Завдяки точки світу, запрограмувати їх за розкладом і вмикати їх автоматично, коли знадобиться. Одним із прикладів деяки дому є *Hofl SleepCompanion* [1]. Ця спеціалізована розумна лампочка розроблена для того, щоб розбудити природ відтінком, який знижує рівень мелатоніну у тілі. У нього є інший режим, який допомагає швидше заснути. По суті, допоможе відпочити в довгостроковій перспективі. Ще одним інноваційним продуктом освітлення є *Flux Bluetooth* підтримує понад 16 мільйонів кольорних градієнтів і повністю керується за допомогою програми для смартфона. Його різними кольорами в різний час доби, і навіть можна запрограмувати реагування на музику в кімнаті. Рис. 1.1 – Дод має вбудований розширювач сигналу, який розширює діапазон Wi-Fi. Чим більше їх, тим краще буде Wi-Fi по примі освітлювачами (рисунок 1.2). Рис. 1.2 – Лампа *Sengled 1.1.2 Smart Entertainment* рішення Оскільки бездротові техні повсякденного життя, самі наші будинки перетворюються на простір, керований комп'ютером. Це, звісно, вклк доступно з точки зору розумних домашніх розважальних пристроїв, склавши список деяких з найкращих і найпр доходить до розумної автоматизації в домі, розважальні пристрої, мабуть, є першим очевидним кроком, який по готова до підключення, якщо її було придбано нещодавно. Завжди потрібен центр для керування ними, розумний сп домашню автоматизацію. *Logitech Harmony* [2] – використовуючи безкоштовну супутню програму, пристрій є ро на основі смартфона до 8 домашніх розваг та інших пристроїв автоматизації. Це фізична центральна частина ключовий компонент, який пов'язує все разом (рисунок 1.3). Рис. 1.3 – Домашній концентратор *Logitech* На додаток або відтворення з динаміків, він може взаємодіяти з розумними лампами, термостатами та іншими елементами пе будинком. *Amazon Echo* [6] – це центр розумного дому (рисунок 1.4), створений навколо голосу. Він може не

програми або потокове передавання по Bluetooth. Echo може керувати іншими розумними пристроями по всьому і або будь-чого іншого, що можна підключити. Рис. 1.4 – Amazon Echo Пульт NEEO Thinking Remote (рисунок 1.5). Пристроїв, до яких він може підключатися та керувати, NEEO розроблено для спрощеного, централізованого керування дистанційного керування пропонує розпізнавання рук і сенсорний екран з високою роздільною здатністю, щоб перш за все максимально прямим і простим. Також наявна функція SOS. Рис. 1.5 – NEEO Thinking Remote 1.1.3 Розумна побутова техніка може стати більш зручною, безпечною та ефективною. LG Smart ThinQ – це електрична одинична ду. Працюючи зі спеціалізованим додатком для смартфона, цю піч можна програмувати та керувати з будь-якої точки та будь-які інші налаштування можна за допомогою розширеної панелі керування або смартфона[7]. Рис. 1.6 – Індукційна духовка – це очікування попереднього розігріву. Піч Café від GE усуває цю проблему, дозволяючи вмикати ду та таймери та змінювати температуру з будь-якого місця. Рис. 1.7 – Café Плита Wemo Enabled Crock-Pot – за допомогою супутнього додатка можна керувати температурою, таймерами та плануванням за допомогою свого смартфона. Починаючи готувати, розігрівати вже готову їжу та планувати вечерю. Це навіть приносить спокій тим, хто турбується про роботу. Рис. 1.8 – Wemo Enabled Crock-Pot 1.1.4 Модулі та компоненти розумного дому Завдяки технологіям стають більш зручними, ефективними та безпечними. Blossom – розумний контролер поливу керувати вашими , використовуючи розумні технології, щоб поливати ваш ландшафт якомога ефективніше (рисунок 1.9). Рис. 1.9 – Blossom смартфон ви зможете контролювати використання води з будь-якої точки світу, встановлюючи різні поливи для різних зон. Один з перших розумних кондиціонерів (рисунок 1.10). Звичайні віконні блоки змінного струму дають лише два варіанти на день, витрачаючи електроенергію та гроші, або повернутися додому в жаркий і вологий будинок. Замість цього Aros про моделі та звички, адаптуючи його роботу до будинку та способу життя. Рис. 1.10 – Aros Використовуючи в процесі може автоматично підтримувати потрібну температуру в потрібний час, максимізуючи економію в процесі. Крім того, з будь-якої точки світу. Fortrez Z-Wave – бездротовий водяний клапан, відображений на рисунку 1.11. Завдяки цьому клапану з'являється можливість дистанційно керувати водопостачанням за допомогою системи домашньої автоматизації можна бути спокійним, знаючи, що з технікою все гаразд[8]. Рис. 1.11 – Fortrez Z-Wave Наприклад, якщо пристрій це, автоматично перекидаючи подачу води, щоб обмежити пошкодження. Тим часом прийде сповіщення по телебаченню. Це ідеальне рішення для будинків для відпочинку, оренди та навіть комерційних будівель, його можна застосовувати для ванн, холодильників тощо. 1.2 Інтелектуальні IoT платформи Internet of Things – це багаторівнева платформа автоматизації та надання підключених пристроїв у межах Інтернет речей. Працює за допомогою хмарних, охолодження апаратних пристроїв. Для початківців та розробників це готова платформа, доступна для миттєвого використання, IoT відіграє важливу роль у постачальниках та стартапах інтелектуальних пристроїв, які можуть використовувати керуванням, моніторингом у реальному часі, хмарними сервісами та інтеграції з використанням смартфонів використовують штучний інтелект для отримання додаткової інформації зі сховищ. Комбінація хмарних та периферійних моделі, яка розділяє та розподіляє між підключеними пристроями масштабним чином. Доведено, що Інтернет речей Архітектура платформи IoT включає чотири рівні: * речі; * зв'язок; * основні атрибути IoT; * аналітика і програми. На етапі розробки підключених пристроїв та інтелектуальних речей. Рівень зв'язку використовується для обміну повідомленнями, рівень використовується для управління конфігурацією, OTA-сервісів та обміну повідомленнями. Вершина основних речей та системи безпеки, які виявляють аномалії користувальницьких рішень IoT. Деякі переваги платформ IoT: Удосконалена платформа IoT забезпечує гнучку масштабованість відповідно до вимог клієнта. Як правило, розробка інфраструктурою, тому він забезпечує прямий доступ до вихідного коду, систем інтеграції, системи розгортання, миттєво пропонує наскрізне шифрування потоку, включаючи дані у стані спокою, автентифікацію пристрою , управління ідеями приватної хмари Нижче наведено деякі з доступних на ринку платформ для створення програм IoT. * Microsoft Azure і продуктів Microsoft, створена для надання сервісів, сервісів додатків, засобів обміну повідомленнями, сервісів з Azure IoT – забезпечити постійне обслуговування за рахунок ефективного зберігання даних та допомогти компанії керувати даними у сховищі у кожному підрозділі підприємства. * Веб-сервіси Amazon – дозволяє користувачам використовувати кіберпросторі. Деякі організації використовують AWS для створення, розміщення, управління та організації інфраструктури користувачам безліч переваг хмарного зберігання даних, передачі програм та управління сховищем. * Google Cloud надає базу даних із документами, хмарні обчислення та еластичну базу даних. Це третя за величиною платформа для IoT Google тощо. * ThingWorx – платформа IoT, що широко використовується, придбана PTC в 2013 році. Розробник програмного та ефективного виконання, аналізу складних даних та контенту та пропонує рішення для швидкого управління надає послуги з ефективним з'єднанням для передачі даних та голосу, надійним життєвим циклом SIM-карти, ефективною системою, що настроюється. * HP Universal things – забезпечує точні вирішення всіх проблем, пов'язаних із IT. Він надає ефективний збір даних, аналізу даних та правильної платформи для запуску нових мобільних додатків. * Хмарні платформи для віддалених пристроїв, які підключені подвійним чином або до прямих хмарних сервісів. Це обладнання з ком

додатком. * Оракул Інтернет речей – встановлює зв'язок між програмним забезпеченням компанії та її політикою додатка та використання його в комерційних цілях. * Bosch IoT Suite – розроблений у Німеччині та відрізняється тим, що він доступний у середовищі з відкритим вихідним кодом. * IBM Watson Інтернет речей – пропонує такі послуги, як дані без будь-яких перешкод та хмарне сховище. Він пропонує кілька варіантів використання платформи IoT для додатків для роботи з стандартними системами IoT. Додаток для домашньої автоматизації може бути двох типів - програми створені лише для одного конкретного пристрою, припустимо, для певної системи безпеки будинку, ці іншою системою Інтернету речей, яка може бути вдома, ці програми працюватимуть лише з системами, які сумісні з - це програми можуть допомогти керувати всією системою IoT, яку встановили у домі. Кожним розумним домом допомогло багатозадачною програмою. Тепер, яким додатком для керування розумним будинком буде користуватися, віддають перевагу однозадачним додаткам, думаючи, що вони забезпечать кращу безпеку та безпеку, тоді як багатозадачною програмою. 1.3.1 Amazon Alexa Amazon Alexa вважається одним з найкращих додатків для керування налаштувати пристрої з підтримкою Amazon, створювати списки покупок, слухати музику. Додаток Alexa призначений існує багато інших пристроїв, сумісних із цим додатком, наприклад Philips Hue Smart Bulbs, Wemo smart plug, Lu Roomba 690 та багато інших[14]. Рис. 1.12 – додаток Alexa Налаштування програми Alexa на смартфоні є дієвим інтерфейсом розумного дому. Щоб розбудити Алексу, все, що потрібно зробити, це назвати її ім'я. Функція голосовою всіма розумними пристроями, підключеними до Alexa, за допомогою усних команд. Однією з найкращих альтернативна система керування. Ті, хто боїться використовувати систему голосового керування через безпечно скористатися альтернативним інтерфейсом Alexa. Це служить бонусним фактором успіху Alexa. 1.3.2 SmartThings SmartThings Додаток SmartThings може допомогти користувачам отримати доступ до кількох пристроїв швидше, домашніми пристроями IoT, такими як холодильник, пральна та сушильна машина, кондиціонер для посудомийної функції цього додатка є можливість дистанційно керувати та перевіряти стан пристроїв, групувати кілька пристроїв налаштувати параметри пристрою, отримувати сповіщення про різні пристрої. Багато пристроїв розумного дому SmartThings. Ці пристрої включають розумний термостат Ecobee, камеру безпеки Netgear Arlo без проводів Pro після 2020 року також пристрої Google Nest . Це означає, що якщо вдома є концентратор Samsung SmartThings Home, не доведеться використовувати кілька програм для різних пристроїв. Успіх програми SmartThings залежить від дизайну нової та покращеної програми. Однак, будучи повністю підключеним до хмари, він не надсилає важливі повідомлення Google Home – додаток від технологічної компанії Google (рисунок 1.14). Рис. 1.14 – додаток Google Home По Android, так і на платформі iOS, програма Home дозволяє налаштувати пристрої Google Home і Chromecast пристроїв, як-от світильники, камери та багато іншого. Пристроями сторонніх розробників, які сумісні з програмами Samsung SmartThings, Honeywell Home, WeMo тощо. Програма Google Home не лише для Android для розумного керування, організувати й керувати сумісними лампами й колонками, а також керувати ними лише одним або двома діями, таких як увімкнення освітлення, перевірка погоди, отримання найцікавіших новин та інші дії, і все це ефективною програмою з часом персоналізуються. Завдяки своїй універсальності програма Google Home є однією з найкращих розумного дому на ринку зараз[13]. 1.3.4 Apple Home Домашній комплект Apple, ймовірно, є однією з найповніших на ринку, а додаток для домашньої автоматизації iOS – ідеальний компаньйон для нього. Він призначений для роботи з іншими пристроями розумного дому. Додаток Apple Home Kit можна використовувати будь-де, будь то інформаційною панеллю розумного дому, яка полегшує роботу з кожним аспектом розумного будинку, на рисунку додатком[15]. Рис. 1.15 – додаток Apple Home Kit Додаток дозволяє користувачам створювати «сцени», що діють натисканням на своєму смартфоні. Так само, як і вихід з дому: виходячи з дому, все, що потрібно зробити, це натиснути переконавшись, що штори по всьому будинку закриті, світло вимкнено, а термостат налаштований на енергоефективні бренди з пристроями, які підтримуються HomeKit. Додатковими факторами успіху цієї програми є безпека та ефективність прямо з головного екрана, що позбавляє від необхідності переміщатися по програмі. Крім того, додаток Apple HomeKit розумного дому, що забезпечує повну безпеку для більшості користувачів. 1.4 Висновки за першим розділом На сьогоднішній день досить широкий спектр технологій розумного дому, що обумовлено постійно зростаючим попитом на рішення для житлових комплексів можна вважати Internet of Things – багаторівневу платформу, яка здійснює безпосередньо підключених пристроїв у межах Інтернету речей. Серед найбільш популярних рішень можна відмітити Hoff SleepCo Boost Logitech, Harmony, Amazon Echo, LG Smart ThinQ, Fortrez Z-Wave та ін. В якості інтерфейсу користувача програми однозадачний і багатозадачний. Яким саме додатком для керування розумним будинком буде користуватися, віддають перевагу однозадачним додаткам, думаючи, що вони забезпечать кращу безпеку та безпеку, тоді як багатозадачною програмою. Типові представники подібного ПЗ є, наприклад, Amazon Alexa, SmartThings, Google

яка використовується для обміну даними між усіма системними модулями комплексу. Ці різні технології відрізняють передачі даних тощо. Деякі з цих технологій наведено нижче. Interbus – це шина даних датчика / приводу. Він має інтерфейс RS485, що приносить деякі переваги, а саме: можливість передачі даних всередині системи на довжину критичних ситуаціях. Недоліком цієї системи є те, що при виході з ладу одного пристрою всі інші також виходять з ладу через топологію кільця, положення всередині кільця достатньо для ідентифікації. Швидкість передачі даних по цій шині master-slave. Він використовує інтерфейс RS485, а також можна використовувати інтерфейс TP або RS232. Максимальна довжина такої шини – 1200 м, а швидкість передачі даних – 76,8 кбіт/с.; Profibus використовує середовище передачі даних. Існує три варіанти цієї технології: Profibus-FMS (специфікація повідомлень fieldbus пристрої) і Profibus-PA (автоматизація процесів), які охоплюють широкий спектр можливих застосувань і можуть бути використані в мережах (Area Network): спеціалізується на автомобільних додатках завдяки високій швидкості передачі даних 1 Мбіт/с і в час передачі даних він використовує метод CSMA / CA (множинний доступ із визначенням несучого зв'язку з у мережеве рішення для автоматизації та управління мережами, розроблене американською компанією Echelon. Її було використано в централізованих контролерах автоматизації будівель, а також у децентралізованих комп'ютеризованих системах шин (ANSI/CEA-709.1-B і EN ISO/IEC 14908), яка дозволяє інтелектуальним пристроям керувати мережею керування. LON означає локальну операційну мережу; BACnet (Мережа автоматизації та управління будівельними даними, розроблений Американським товариством інженерів з опалення, охолодження та кондиціонування повітря будівель, щоб дозволити пристроям і системам обмінюватися інформацією[18]. BACnet використовується в багатьох світу і отримав міжнародний стандарт ISO 16484-5 у 2003 році. BACnet розвинувся через потребу в стандартизованій різноманітній автоматизації та контролі компонентів у будівлі для зв'язку один з одним, забезпечення сумісності бездротова система нового покоління, спочатку розроблена компанією Zensys в Данії, яка дозволяє всім електронним пристроям так і з користувачем бездротовим способом. Після перших установок у 2005 році було створено Z-Wave Alliance, підключені пристрої, керовані через програми на смартфонах, планшетах або комп'ютерах, використовуючи техніка. Воно використовує радіосигнал з частотою 868.4МГц і максимальна випромінювана потужність близько 20 – 30 мВт в радіохвилі малої потужності, які легко проходять крізь стіни. Z-Wave використовує топологію мережі Mesh, тому радіохвилі окремого пристрою. Система дуже проста і легка як в установці, так і в експлуатації, і практично не вимагає на початку експлуатації. Ця система використовується особливо для будинків з рішеннями «зроби сам». Епосеа використовує найменших змін у навколишньому середовищі для отримання необхідної енергії, яка потім перетворює енергію для передачі радіочастотних сигналів від датчиків до приводів. Усі компоненти EPOSea працюють без батареї. Найпростішим прикладом продукту, заснованого на РЧ-протоколі, є вимикач освітлення без батареї такий перемикач, в нього вкладається стільки енергії, що перемикач виробляє стільки електроенергії, скільки радіочастотного сигналу на актуатор, щоб увімкнути світло. Або, повертаючи ручку на вікні, виробляється стільки енергії до клапана на радіаторі, який повідомляє йому, що вікно відкрите і що клапан повинен закритися. Система вимагає проводку між вимикачем і світлом. Gould Modicon, підтримується більшістю програмованих логічних пристроїв Master/Slave. Він дуже простий і зручний у використанні, тому його використовують як виробники контролерів обладнання. Він став дуже популярним, оскільки є безкоштовним. Він обмежений через простий обмін даними і велику кількість потреб. X10 – це протокол для зв'язку між електронними пристроями, що використовуються для домашньої автоматизації ліній електропередачі для сигналізації та керування. Також визначено транспортний протокол на основі Bluetooth в 2002 році для дистанційного керування побутовою технікою. Це була перша технологія домашньої автоматизації, яка досі є в застарілих, оскільки дуже повільний (близько 20 біт/с). Через невеликий набір команд і погану надійність він сьогодні використовує ZigBee описує інфраструктуру та послуги, доступні для програм, що працюють на платформі ZigBee. Він використовує малопотужні цифрові радіосигнали на основі стандарту IEEE 802.15.4 для потреб бездротових мереж (сигналізації, автоматизація будівель). Сама технологія проста і дешева, ніж з іншими бездротовими мережами ZigBee містить лише 10% програмного забезпечення, що міститься в типовій Bluetooth або звичайній бездротовій мережі. Момент між ZigBee та комісією BACnet тривають домовленості щодо встановлення зв'язку між дротовим та бездротовим (DALIa, DALIb) – це протокол цифрового зв'язку, розроблений спеціально для схем освітлення. DALI дуже підходить для несправних ламп. Тому він дуже підходить в поєднанні з автоматизацією будівель, де потрібен дистанційний контроль освітлення представлений у 1999 році виробниками електронних баластів, які хотіли стандартизований цифровий протокол зроблений так, що його дуже легко встановити і налаштувати. Всі приводи, контролери та датчики з'єднані один з одним складається з електронних баластів, перемикачів, датчиків, інтерфейсів керування та контролерів 1-10 В. DALI – це система управління освітленням. DMX 512/1990 є стандартом цифрової передачі даних для диммерів і контролерів освітлення може керувати до 512 каналами. Дані передаються пакетами. Кожен пакет оновлює всі вбудовані пристрої. Кожний пакет позначають початок і кінець кожного пакета. Ми не отримуємо доступ до пристроїв безпосередньо, але інформація, :

конкретній структурі кожного пакета. Стандарт DMX широко використовується в театрах, дискотеках і клубах, ос протитуманних апаратів та інших пристроїв реалізують його. Коппех – технологія була стандартизована та виз (EN50090), китайськими (GB/T 20965), американськими правилами (ANSI/ASHRAE 135) та світовими норматив одного виробника, який виробляє елементи KNX. Натомість сьогодні в асоціації Коппех зареєстровано понад 41 понад 7000 різних елементів KNX. Це кількість виробників забезпечує 100-відсоткове забезпечення елементів будинку.

2.1.1 Принципи побудови системи Існує 3 підходи до організації управління системами автоматизації: системи[20]. Суть централізованого розумного будинку полягає в тому, що йде програмування лише одного центру програмований контролер, в який записується заздалегідь спеціально створена під об'єкт програма, основі яко інженерними системами. Це дозволяє використовувати широкий вибір обладнання та складні сценарії. Переваги складних сценаріїв, прив'язаних до часу доби; * станом мешканця, температури, місячному циклі; * Простот виконуються лише на центральному контролері. Недоліки: * потрібне обов'язкове програмування системи; * зале потребує наявність резервного обладнання. У децентралізованих системах автоматизації кожен виконавчи енергонезалежно пам'яттю. Цим пояснюється надійність таких систем. При виході з експлуатації одного пристро підключених до цього пристрою. Переваги: висока надійність. Недоліки: * висока вартість обладнання; * скл складність інтеграції коїться з іншими апаратними рішеннями; * часто має закритий протокол керування. Система, день – це система змішаного управління. У ролі центрального управління стоїть контролер, але управляючі моду чином, при виході з ладу центрального контролера всі життєво важливі системи переводять на ручне управління висока вартість обладнання; * легкість у початковому налаштуванні; * не потребує резервного обладнання.

2.1.2 Ст того, щоб контролер міг передавати сигнали управління датчикам і виконавчим пристроям, потрібно вибрати яким ч собою. Є два основних типи передачі сигналу – провідний та бездротовий. Дротові системи автоматизації – це пристрої керування кліматом, керуючі панелі зв'язуються єдиною провідною інформаційною шиною, якою йдуть сиг електричному щиті. Як провідної інформаційної шини використовуються спеціальні кабелі, а в окремих випадках зв переваги і недоліки. Переваги: * висока швидкість відгуку; * великий вибір дизайну елементів, що управляють; * від * передача керуючих сигналів великі відстані. Недоліки: * проектування на стадії ремонту; * потрібне прокладання та обслуговування системи потрібна висока кваліфікація інсталюатора. У бездротових системах автоматизації сигнал радіоканалу. Це дозволяє скоротити кількість дротів, а також час на інсталяцію системи. Ці системи можна монтува проводкою. Бездротовий вимикач може бути радіопередавачем, який зв'язується з усіма іншими вимикачами. Пе будинки з уже готовим ремонтом із класичною проводкою; * невелика кількість дротів. Недоліки: * погана перешкод вигляді батарейок; * складна інтеграція з іншими системами; * низька безпека передачі інформаційних повідомлені складність передачі радіосигналу через товсті перегородки та великі відстані. Усі системи можуть мати відкрит протокол дозволяє користувачеві самому розробляти програмне забезпечення та нове обладнання. Є можливіс протокол не дозволяє змінювати або вбудоване програмне забезпечення модифікацію, але за це користувач може вартість.

2.2 Фреймворки та платформи для написання системи управління розумним будинком Фреймворк – це і Він забезпечує основу, на якій розробники програмного забезпечення можуть створювати програми для певної плат попередньо визначені класи та функції, які можна використовувати для обробки введених даних , керування апаї програмним забезпеченням. Це спрощує процес розробки, оскільки програмістам не потрібно винаходити вел програму. Фреймворк схожий на інтерфейс прикладного програмування (API), хоча технічно фреймворк включає компілятор та інші програми, що використовуються в процесі розробки програмного забезпечення. Існує кілька рі: приклади включають ActiveX і .NET для розробки Windows, Cocoa для Mac OS X, Cocoa Touch для iOS, Angular для веб-додатків. Комплекти для розробки програмного забезпечення (SDK) доступні для кожної програмування, розроблені спеціально для відповідної платформи. Наприклад, програмне забезпечення для розр Mac OS X, призначений для написання та компіляції програм для фреймворку Cocoa. У багатьох випадках прог системою. Наприклад, програма, написана для Android Application Framework, працюватиме на пристрої додаткових файлів. Однак для запуску деяких програм потрібна спеціальна структура. Наприклад, для програми Framework 4.0, який інстальовано не на всіх машинах Windows (особливо на комп'ютерах зі старими версіє програми необхідно встановити пакет інсталюатора Microsoft .NET Framework 4.

2.2.1 ReactJs Створена використовуваною бібліотекою відповідно до опитування State of JavaScript 2021. React технічно не є фреймвор користувача, але в розмовній мові вона розглядається як фреймворк[9]. React.js забезпечує швидкий інтерфе використовують дані, миттєво застосовуючи зміни до елементів, а не оновлюючи всі випадки одночасно, як це поетапне використання віртуальної DOM (об'єктна модель документа) для швидкого оновлення вмісту веб-сторі домену, створений тією ж командою. Особливості: * Компоненти React: React ділить сторінку на кілька компоне користувача і має власну логіку та дизайн, що робить їх доступнішими та швидшими для повторного викори

віртуального документа представляє дані в деревоподібній структурі. Він також поділяє дані на модулі, що міст інтерфейсу користувача, присутній в документі. Це гарантує, що коли перезавантажується сторінка, вона оновлює JavaScript XML – це синтаксис, подібний до HTML, який обробляється у виклики JavaScript, що дозволяє р елементи HTML. * Декларативний інтерфейс користувача: синтаксис у стилі HTML JSX дозволяє контролювати потік компонент. 2.2.2 Vue.js Vue був створений у 2014 році Еваном Ю., колишнім співробітником Google. Він має на м можливості Angular і React. Vue зберігає синтаксис шаблону Angular і прив'язування даних React, реквізити та розробникам із знаннями HTML, CSS і JavaScript створювати SPA і кросплатформні програми корпоративного проекти за допомогою HTML або JSX. Особливості: * Анімовані переходи: Vue надає різні способи застосування додавання або оновлення їх із DOM. Є можливість інтегрувати сторонні бібліотеки анімації для ще більшої інтеракт які пов'язують DOM з даними екземпляра Vue. Він компілює шаблони у віртуальний DOM у вигляді HTML, які браузером. * Маршрутизація Vue: маршрутизація дозволяє користувачам перемикатися між сторінками, не оновл та швидко. * Директиви: директиви – це інструкції, які прив'язують себе до властивості екземпляра Vue. Вони частиною програми, щоб змінити те, що користувач бачить перед нею. 2.2.3 Angular Angular – це платформа клієнтських програм за допомогою HTML і TypeScript. Angular написаний на TypeScript. Він реалізує основні т TypeScript, які імпортує у свої програми[10]. Архітектура Angular спирається на певні фундаментальні фреймворка Angular є компоненти Angular, які організовані в NgModules. NgModules збирає пов'язаний к визначається набором NgModules. Додаток завжди має принаймні кореневий модуль, який дозволяє завантажу функцій. Компоненти визначають перегляди, які є наборами елементів екрану, які Angular може вибирати та змін Компоненти використовують служби, які надають певну функціональність, не пов'язану безпосередньо з предста введені в компоненти як залежності, що робить код модульним, багаторазовим та ефективним. Модулі, компоне декоратори. Ці декоратори позначають свій тип і надають метадані, які повідомляють Angular, як їх використовуват його з шаблоном, який визначає подання. Шаблон поєднує звичайний HTML з директивами Angular і розміткою HTML перед відображенням його для відображення. Метадані для класу служби надають інформацію, необхі компонентів за допомогою ін'єкції залежностей (DI). Компоненти програми зазвичай визначають багато представл Router послугу, яка допоможе визначити шляхи навігації між представленнями. Маршрутизатор надає складні нає найкращим варіантом для створення та розгортання CRM-систем. Angular поєднує в собі декларативні шаблони інструменти та інтегрований набір найкращих практик для вирішення проблем розробки. Особливості: * Двосторонн моделі, тому якщо змінити модель, користувачі зможуть автоматично бачити зміни в моделі перегляду, що скорочує парадигма програмування дозволяє класам, компонентам і модулям працювати взаємозалежно, зберігаючи узгод Архітектура MVC. Цей тип архітектури ізолює логіку програми від рівня UI, підтримуючи розділення проблем і з Node.js – програмна платформа, заснована на двигуні V8, що перетворює JavaScript з вузькоспеціалізованої м представляє цікавий перетин фронтенду та бекенд-технологій. Побудований на JavaScript, який є мовою веб-с можливості JavaScript для роботи на серверній частині архітектури веб-додатків, а також для архітектури без серв середовищі виконання на сервері. Гарною аналогією для проведення є порівняння між Node.js і Java, яка також заг виконання. Середовище виконання Node.js розроблено, щоб бути легким і ефективним, з неблокуючим введенням створення програм в екосистемі Node.js ще простіше. Рис. 2.1 – Схема роботи Node.js Менеджер пакетів для Node.js ідексом бібліотек, створених спільноту розробників Node.js, які легко надаються та імпортуються іншими проєк поширених функцій і коду, які спрощують створення нових проєктів і покращення старих. Node.js є застосовним використання. Як фреймворк на стороні сервера, Node.js підходить для додатків у внутрішній частині технол важливим для вибору правильної технології для його створення. Гнучка й ефективна природа Node.js дозволяє програми. Додатки в режимі реального часу, такі як миттєві повідомлення та інструменти для співпраці, є с можливостями швидкої синхронізації Node.js, також робить його корисним для програм на основі подій. Прикладам і WebRTC. Завдяки своїй гнучкості Node.js дуже добре підходить для створення як безсерверних, так і мікросервісн для економії ресурсів, так і для ефективного керування життєвими циклами додатків. Безсерверна архітектура пог обчислювальні витрати, вимагаючи лише ресурсів, необхідних для роботи програми без додаткових витрат. Ці проєк Node.js із безсерверною архітектурою чудовою комбінацією. Існують пакети npm для безсерверного проектуванн добре працює із загальною архітектурою мікросервісів у випадках використання C2C та B2C, коли навантаженн обробляє одночасні з'єднання. Оскільки IoT побудований на багатьох пристроях, які надсилають невеликі повідомл хорошим бекендом для таких видів додатків, забезпечуючи безсерверну архітектуру та підтримку зв'язку в реаль Додатки для розумних будинків можуть бути побудовані на базі різноманітних рішень. Додатки для систем розумни досить популярні у світі, через свою доступність, та те що обидва ці фреймворки побудовані на базі JavaScript. Е цих платформах. Такий вибір виникає через велику кількість документації та навчального матеріалу. Протягом трив:

розробниками як інтерфейсний інструмент через його універсальність і здатність поєднувати бізнес-логіку та ел найпопулярніших технологій розробки програмного забезпечення, що пропонує різноманітні функції, які д прискорювати платформи в Інтернеті. Поряд з окремими технологіями мобільної розробки, він також використе кросплатформених додатків. Завдяки такому зв'язку як Angular + Node.js, з'явився стек під назвою MEAN – це JavaScript з відкритим вихідним кодом для створення динамічних веб-сайтів і веб-додатків [3]. Оскільки всі кс написані на JavaScript, програми MEAN можуть бути написані однією мовою як для серверного, так і для клієнтськ що їх часто порівнюють безпосередньо з іншими популярними стеками веб-розробки, такими як стек LAMP , компон рівень презентації веб-додатків і не включають рівень операційної системи.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОСТУПУ

3.1 Вибір мікроконтролеру

Для практичної реалізації проекту було обрано платформу Arduino. Серед ц рішення у користь Arduino UNO. Для створення більш бюджетного проекту було обрано Atmega328 Maker UNO аналогом Arduino UNO. Рис. 3.1 – Atmega328 Maker UNO Він сумісний з Arduino UNO R3, тому Maker UNO і сумісний з усіма прикладами коду та бібліотеками для Arduino UNO, що не доставить труднощів у розробці. ESP8266 виробника Espressif з інтерфейсом Wi-Fi. Рис. 3.2 – ESP8266 Крім Wi-Fi мікроконтролер відрізняється можливіс пам'яті з інтерфейсом SPI. Дана плата використовується для програмування, контролю, управління та підключення на комп'ютері.

3.2 Вибір фреймворку та бібліотек для написання веб-додатку

Для написання веб-додатку було (найкращим рішенням для створення закритих додатків. Бібліотеки які використовувалися: * Chart.js * Bootstrap JavaScript з відкритим вихідним кодом для візуалізації даних , яка підтримує вісім типів діаграм: * стовпчаста; * л радар; * полярна; * розсіяна. Створена лондонським веб-розробником Ніком Дауні в 2013 році, тепер бібліо популярністю бібліотекою діаграм JavaScript на GitHub за кількістю зірочок після D3.js, яка вважається значно легі ніж D3.js. Chart.js відтворюється на полотні HTML5 і широко розглядається як одна з найкращих бібліотек віз платформа веб-розробки з відкритим вихідним кодом. Він призначений для полегшення процесу веб-розробки ад пристрої, надаючи набір синтаксису для дизайну шаблонів. Іншими словами, Bootstrap допомагає веб-розробник не потрібно турбуватися про основні команди та функції. Він складається зі скриптів на основі HTML, CSS та JS для і дизайном. Основною метою Bootstrap є створення адаптивних веб-сайтів, орієнтованих на мобільні пристрої. Це інтерфейсу веб-сайту на всіх розмірах екрана. Bootstrap доступний у двох варіантах: попередньо скомпільо Bootstrap із вихідним кодом дозволяє отримати доступ до порту Sass . Це означає, що він створює власну таблиці змінювати та розширювати інструмент за потреби. Також можна встановити Bootstrap за допомогою менеджера фреймворку, бібліотеки та активи. Деякі з найпопулярніших менеджерів пакетів включають npm, Composer і сервера, тоді як Composer зосереджується на інтерфейсі. Деякі з компонентів інтерфейсу Bootstrap включа: зображень і кнопки. Щоб зменшити час завантаження сторінки сайту, Bootstrap мінімізує файли CSS і JavaScript синтаксису між веб-сайтами та розробниками, що ідеально підходить для командних проектів. Bootstrap поста сіток, що позбавляє від створення системи з нуля. Система сіток складається з рядків і стовпців, що дозволяє ство вводити медіа-запити у файлі CSS. Крім того, система сіток Bootstrap робить процес введення даних більш г дозволяє визначити індивідуальні точки зупинки кожного стовпця на основі потреб веб-проекту. RxJS (Reactive E реактивного програмування з використанням Observable, що полегшує створення асинхронного коду або кс програмування – це парадигма асинхронного програмування, що стосується потоків даних і поширення змін. RxJ потрібна до тих пір, поки тип не стане частиною мови і поки браузер не підтримають його. Бібліотека та спостережуваних і роботи з ними. Ці допоміжні функції можна використовувати для: * перетворення існуючого коду ітерація значень у потоці; * відображення значень до різних типів; * фільтрація потоків; * створення кількох потокі підключений до ESP8266 у відношенні провідний-підпорядкований, надсилаючи дані до API. API зберігає дані в баз протокол HTTP, щоб отримати дані та відобразити їх у вигляді діаграми або таблиці. API було розроблено на Node.J

```

mongoose = require("mongoose");
mongoose.connect("mongodb+srv://localhost:8080:test@sm
useNewUrlParser: true }) .then(console.log("Connected to mongoDB...")) .catch(() => { console.log(
mongoose.set("useCreateIndex", true); module.exports = mongoose;
Для сигналізації було розроблено 5 запит
const _ = require('lodash'); exports.controller = {
Отримання статусу сигналізації: async getStatus(req, res) {
res.status(200).send({ alarm: status }); },
Створення тривоги: async createAlarm(req, res) {
const alarm = c
const newAlarm = new Alarm({ status: 0 })
await newAlarm.save().then(res.status(200).send("Alarm create
} },
Включення сигналізації: async enableAlarm(req, res, next) {
const alarm = await Alarm.findOne();
if (!a
alarm.save();
return res.status(200).send("Alarm enabled");
},
Виключення сигналізації: async disableAla
Alarm.findOne();
if (!alarm) return next();
alarm.status = 0;
await alarm.save();
return res.status(200).se
статусу для esp:
async getShortStatus(req, res, next) {
const alarm = await Alarm.findOne();
if (!alal
(alarm.status) shortStatus = "1";
return res.send(shortStatus);
} }

```

Для даних про температуру бу


```

require('../models/data'); const _ = require('lodash'); exports.controller = { // :id -> yyyy/mm/dd async f
  _lowerCase(req.params.currentDay); const data = []; await Data.find((err, records) => { if (!records) return
  date = _lowerCase(record.date); if (date === currentDay) { data.push(record); } }); }); res.send({ data: da
  async findAll(req, res) { const data = await Data.find().sort({ createdAt: 'desc' }); return res.status(200).ser
  даних: async removeAll(req, res, next) { Data.deleteMany((err) => { if (err) next(); else return res.status(200
  новий запис в базі даних: async create(req, res) { const date = new Date().toISOString().slice(0, 10); const
  req.body.temperature, humidity: req.body.humidity, date: date }).save(); return res.status(201).send({ data:
  removeByDay(req, res, next) { let currentDay = _lowerCase(req.params.currentDay); await Data.find((er
  records.forEach((record) => { const date = _lowerCase(record.date); if (date === currentDay) { Data.de
  (err) return next(); }); }); }); res.send("Records posted on " + req.params.currentDay + " deleted"); }); } } Для
  виглядає таким чином: const express = require("express"); const router = expres
  require('../controllers/alarmController'); const catchAsync = require('../middlewares/errors').catchAsync; coi
  router.get('/', catchAsync(alarmController.controller.getStatus)); router.get('/short', catchAsync(alarmCor
  alarm create one router.post('/create', jwtAuth.jwtAuth, catchAsync(alarmController.controller.createAla
  catchAsync(alarmController.controller.enableAlarm)); router.post('/disable', jwtAuth.jwtAuth, catchAsyn
  module.exports = router; const express = require("express"); const router = expres
  require('../controllers/dataController'); const catchAsync = require('../middlewares/errors').catchAsync; coi
  router.get('/', catchAsync(dataController.controller.findAll)); router.post('/', jwtAuth.jwtAuth, ca
  router.delete('/:id', jwtAuth.jwtAuth, catchAsync(dataController.controller.removeAll)); //yyyy
  catchAsync(dataController.controller.findByDay)); router.delete('/:currentDay', jwtAuth.jwtAuth, catchAsyn
  module.exports = router; Веб-додаток був написан на Angular 13. Він використовує протокол HTTP для зв'язку з API
  та зберегти їх. У проєкті постійно використовується Observable – це послідовність подій у часі. Сервіс який г
  Observable { return this._http.get(this.apiUrl); } getAlarm(): Observable { return this._http.get(this.apiUrl)
  Observable, дані про температуру: weather: WeatherModule[] = []; constructor(private _weather: WetherSer
  void { this._weather.getWether().subscribe((res: WeatherModule[]) => { this.weather = res }); } Виконується :
  компонента ngOnInit(Ініціалізація компонента) зберігається температура. Потім, вибравши дату, компонент фір
  таблицю за допомогою інтерполяції рядка. {{weather.date}} {{weather.time}} {{weather.humidity}} {{weathe
  про температуру та вологість повітря За допомогою бібліотки ChartJs є діаграма яка показує почасово зміни волог
  ті ж самі дані, що й для таблиці. Для того щоб використовувати ChartJs необхідно ініціалізувати canvas у шабл
  інтерполяції. canvas id="canvas">{{ chart }} chart = new Chart('canvas', { type: 'line', data: { labels: this.times
  "#3cba9f", fill: false, label: "Temperature" }, { data: this.humis, borderColor: "#ffcc00", fill: false, label: "Humic
  scales: { xAxes: [{ display: true }], yAxes: [{ display: true }], } }); Рис. 3.4 – Діаграма з даними про температ
  керування сигналізацією. Щоб поставити сигнал тривоги, користувач повинен увійти. Рис. 3.5 – Вхід до додатка При
  на Observable який повертає сервіс: login(email: string, password: string): Observable<{token: string}>
  (this.apiUrlLogin, { email, password }); } Підпис на Observable: login(): void { this._wetherService.lc
  localStorage.setItem("token", res.token); }); if (localStorage.getItem("token")) { this.isLogged = true;
  використовується для запиту PUT та оновлення статусу тривоги, токен зберігається до LocalStorage. LocalS
  отриманий токен поміщається в заголовок запиту. setAlarmOn() { const httpOptions = {headers: new Http
  'Authorization': 'Bearer ' + localStorage.getItem("token") }); }; return this._http.post(this.apiUrlAlarmEnable,
  httpOptions = {headers: new HttpHeaders({ 'Content-type': 'application/json', 'Authorization': 'Bearer ' +
  this._http.post(this.apiUrlAlarmDisable, null, httpOptions) } Після входу у шапці додатка з'являється додаткова к
  виконується підпис на Observable та при успіху статус зберігається до змінної isAlarmSet. В залежності від поточн
  на який Observable підписуватись. Також використовується оператор RxJs switchMap – це оператор в основном
  map. Частина дозволяє зіставляти значення з джерела вищого порядку, яке можна спостерігати, у внутрішній Ob
  так switchAll: вона підписується на останній наданий внутрішній Observable, випромінювану Observable вищого
  підписаного внутрішнього Observable. switchMap має лише одну активну підписку за раз, з якої значення
  спостережуваний вищого порядку видає нове значення, switchMap виконує функцію, щоб отримати новий внутріш
  Він скасовує підписку на поточний потік і підписується на новий внутрішній спостережуваний. Який повертає Obse
  void { const setAlarm = this.isAlarmSet ? 'setAlarmOff()' : 'setAlarmOn()'; this._wetherService[setA
  this._wetherService.getAlarm()])) .subscribe(res => { this.isAlarmSet = res['alarm'].stc
  this.isAlarmSet.toString()); }); } В залежності від змінної відображається статус в шапці додатка. Або червоного к
  зеленого кольору з текстом «Увімкнути сигналізацію». Увімкнути сигналізацію Вимкнути сигналізацію Рис. 3.6 –

```

```

приладів прикладних до Arduino(Slave): Створюємо змінні які відносяться до відповідних контактів мікроконтролера
#define ECHO 4 #define PIEZO 8 #define BUTTON 3 #define YELLOW_LED 2 bool alarmArmed; bool object
роботи змінних. pinMode(PIEZO, OUTPUT); pinMode(TRIG, OUTPUT); pinMode(ECHO,
attachInterrupt(digitalPinToInterrupt(3), NULL, RISING); alarmArmed = false; objectDetected =
dht.setup(DHT11_PIN); Wire.onReceive(receiveHandler); Wire.onRequest(requestHandler); } double temper
unsigned long nowTime = 0; unsigned long measuresTime = 0; unsigned long lookForObjectTime = 0; void
objectDetected = false; } Перевіряємо статус сигналізації на увімкнено чи вимкнено, в залежності від змінної
if(alarmArmed) { digitalWrite(YELLOW_LED, HIGH); } else { digitalWrite(YELLOW_LED, LOW); } nowTime = mill
вологість повітря, якщо вони більше 0 тоді записуємо до масиву if(nowTime - measuresTime >= 2100UL) {
dht.getTemperature(); humidity = dht.getHumidity(); if(temperature > 0 && humidity > 0) { Strin
(String)temperature; responseTmp += "I"; responseTmp += (String)humidity; for(int i = 0; i < 11; i++) { respo
перевіряємо статус сигналізації, а саме дистанцію до об'єкту в 7 метрів. if(nowTime - lookForObjectTime :
if(alarmArmed && !objectDetected) { objectDetected = lookForObject(); } } if(objectDetected) { tone(8
delay(100); tone(8, 1000, 300); delay(100); tone(8, 500, 300); delay(100); } } bool lookForObject() { long
delayMicroseconds(2); digitalWrite(TRIG, HIGH); delayMicroseconds(10); digitalWrite(TRIG, LOW); _time = p
Serial.println(distance); if(distance <= 7) { return true; } else { return false; } } Повертаємо знач
Wire.write(response); } void receiveHandler() { int value = Wire.read(); if(value == 1) { alarmArmed = true; }
Arduino(Master): bool alarmArmed; const char* ssid = ""; const char* password = ""; char dataTempHum
thermometer[8] = { B00100, B01010, B01010, B01110, B01110, B11111, B11111, B01110 }; byte droplet[8] = { B00100, B00100
}; Задаємо параметри роботи змінних. void setup () { pinMode(D7,INPUT); Wire.begin(D2,D1);
lcd.createChar(2,droplet); lcd.backlight(); lcd.setCursor(0,0); lcd.print("INITIALIZATION"); delay(500);
lcd.print("Network:"); lcd.setCursor(0, 1); lcd.print(ssid); Serial.begin(115200); WiFi.begin(ssid, password);
delay(1000); lcd.clear(); lcd.print("Connecting.."); } lcd.clear(); } unsigned long nowTime = 0; unsigned
checkMeasurementsTime = 0; unsigned long sendDataTime = 0; String response = ""; void loop() Вхідимо д
String token = loginPOSTrequestApi(); alarmArmed = setAlarmInAPI(token, alarmArmed); } nowTime = mill
виводу: if(nowTime - checkMeasurementsTime >= 3000UL) { response=""; checkMeasurementsTime =
while(Wire.available()) { char c = Wire.read(); response += c; } printTempHumOnLCD(response); print
sendDataTime >= 1200000UL) { sendDataTime = nowTime; String token = loginPOSTrequestApi(); send
if(nowTime - checkAlarmTime >= 300UL) { checkAlarmTime = nowTime; alarmArmed = setAlarm(); } }
parseValuesToJSON(String rawData) { String JSON = ""; JSON += "{"; JSON += "\"temperature\""; JSON += rc
JSON += "\"humidity\""; JSON += rawData[6]; JSON += rawData[7]; JSON += "}"; return JSON; }
sendValuesPOSTrequestApi(String token, String dataToPost) { HTTPClient http; String postData
http.begin("http://localhost:8080/api/data"); http.addHeader("Content-Type", "application/json"); http.addH
httpCode = http.POST(postData); Serial.println(httpCode); http.end(); } Метод для входу до свого користувач
http; String postData; String login = "test@test"; String password = "test"; postData = "email="
http.begin("http://localhost:8080/api/auth/login"); http.addHeader("Content-Type", "application/x-www
http.POST(postData); String payload = http.getString(); http.end(); //Close connection int payloadLe
payload.substring(10,181); return token; } void printTempHumOnLCD(String response) { String temperature
temperature += response[i]; } for(int i = 6; i < 11; i++) { humidity += response[i]; } lcd.setCursor(0, 0); lcd
lcd.print((char)223); lcd.print("C"); lcd.setCursor(0, 1); lcd.write(2); lcd.print(humidity); lcd.print("% I"); lcd.noB
сигналізації. bool setAlarmInAPI(String token, bool alarmStatus) { if(alarmStatus == 1) {
http.begin("http://localhost:8080/api/alarm/disable"); http.addHeader("Content-Type", "application/json");
token); int httpCode = http.POST(""); Serial.println(httpCode); http.end(); } else { HTTPClient http; http.begin
http.addHeader("Content-Type", "application/json"); http.addHeader("Authorization", "Bearer " +
Serial.println(httpCode); http.end(); } return !alarmStatus; } int requestGETapiAlarm() { String payload; if (Wi
http; http.begin("http://localhost:8080/api/alarm/short"); int httpCode = http.GET(); if (httpCode == 200) {
Serial.println(payload); } http.end(); //Close connection return payload.toInt(); } } int setAlarm() {
Wire.beginTransmission(3); Wire.write(alarmStatus); Wire.endTransmission(); return alarmStatus; } void
lcd.setCursor(9, 0); lcd.print("ALARM:"); lcd.setCursor(9,1); if(alarmStatus == 1) { lcd.print("Сигналізація працює
} } ВИСНОВКИ На даний момент існує багато різноманітних додатків для систем «Розумний дім, але переважна
апаратним забезпеченням, або велику ціну. Огляд предметної області показав, що розробити додаток для управління
допомогою поширеного стеку MEAN, але цей стек не виключає використання інших фреймворків для нап

```

запропонованої розробки від стандартних рішень полягає в тому, що для налаштування такої системи можна виконувати взаємодію з сервером та надавати данні одного й того ж виду (інтерфейсу). Також ця система надає змогу виконувати приміщеннях, де єдиним затратним джерелом буде лише апаратна частина. Був використаний фреймворк Angular базуючи на базі даних, та база даних mongoDb. Реалізація проекту була виконана з врахуванням можливості поширення контролю освітлення, щоб тим самим контролювати штучне освітлення приміщення, чи зробити автоматичну роботу іншими модулями і створити цілу екосистему оселі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is home automation [Електронний ресурс] / Режим доступу: [www. URL: https://www.networkworld.com/article/2874914/what-is-home-automation.html](http://www.networkworld.com/article/2874914/what-is-home-automation.html) – 23.05.2022 р.
2. Design and Realization of Smart Home Terminal Applications Based on IOT Technology [Електронний ресурс] / Режим доступу: https://www.researchgate.net/publication/283109456_Design_and_Realization_of_Smart_Home_Terminal_Applications_Based_on_IOT_Technology – 23.05.2022 р.
3. What is MEAN [Електронний ресурс] / Режим доступу: [www. URL: https://en.wikipedia.org/wiki/MEAN_\(programming_framework\)](http://en.wikipedia.org/wiki/MEAN_(programming_framework)) – 23.05.2022 р.
4. Arduino [Електронний ресурс] / Режим доступу: [www. URL: https://doc.arduino.cc/ru/about/](http://doc.arduino.cc/ru/about/) – 23.05.2022 р.
5. What is Smart Home [Електронний ресурс] / Режим доступу: [www. URL: https://www.enginess.io/insights/what-is-smart-home-technology](http://www.enginess.io/insights/what-is-smart-home-technology) – 23.05.2022 р.
6. What is Smart Home [Електронний ресурс] / Режим доступу: [www. URL: https://www.pcmag.com/picks/the-best-smart-home-devices](http://www.pcmag.com/picks/the-best-smart-home-devices) – 23.05.2022 р.
7. Smart Home [Електронний ресурс] / Режим доступу: [www. URL: https://www.techtarget.com/iot/definition/smart-home](http://www.techtarget.com/iot/definition/smart-home) – 23.05.2022 р.
8. React A JavaScript library for building user interfaces [Електронний ресурс] / Режим доступу: [www. URL: https://reactjs.org/](http://reactjs.org/) – 23.05.2022 р.
9. Angular [Електронний ресурс] / Режим доступу: [www. URL: https://angular.io/guide/template-refactoring](http://angular.io/guide/template-refactoring) – 23.05.2022 р.
10. Vue.js A progressive JS Framework [Електронний ресурс] / Режим доступу: [www. URL: https://vuejs.org/](http://vuejs.org/) – 23.05.2022 р.
11. Node.js [Електронний ресурс] / Режим доступу: [www. URL: https://nodejs.org/en/about/](http://nodejs.org/en/about/) – 23.05.2022 р.
12. Google Home Application [Електронний ресурс] / Режим доступу: [www. URL: https://support.google.com/chromecast/answer/7071794?hl=ru&co=GENIE.Platform%3DAndroid](http://support.google.com/chromecast/answer/7071794?hl=ru&co=GENIE.Platform%3DAndroid) – 23.05.2022 р.
13. Amazon Alexa [Електронний ресурс] / Режим доступу: [www. URL: https://ru.wikipedia.org/wiki/Amazon_Alexa](http://ru.wikipedia.org/wiki/Amazon_Alexa) – 23.05.2022 р.
14. iOS Home [Електронний ресурс] / Режим доступу: [www. URL: https://www.apple.com/ua/ios/home/](http://www.apple.com/ua/ios/home/) – 23.05.2022 р.
15. Z-Wave [Електронний ресурс] / Режим доступу: [www. URL: https://en.wikipedia.org/wiki/Z-Wave](http://en.wikipedia.org/wiki/Z-Wave) – 23.05.2022 р.
16. Zigbee [Електронний ресурс] / Режим доступу: [www. URL: https://en.wikipedia.org/wiki/Zigbee](http://en.wikipedia.org/wiki/Zigbee) – 23.05.2022 р.
17. BACnet [Електронний ресурс] / Режим доступу: [www. URL: https://en.wikipedia.org/wiki/BACnet](http://en.wikipedia.org/wiki/BACnet) – 23.05.2022 р.
18. X-10 Industry standard [Електронний ресурс] / Режим доступу: [www. URL: https://en.wikipedia.org/wiki/X10_\(industry_standard\)](http://en.wikipedia.org/wiki/X10_(industry_standard)) – 23.05.2022 р.
19. Smart home plc automation control [Електронний ресурс] / Режим доступу: [www. URL: https://dl.acm.org/doi/10.1145/334111](https://dl.acm.org/doi/10.1145/334111) – 23.05.2022 р.

застосування

Source

Запорізький інститут економіки та інформаційних ...Підрозділи - Запорізь...

0.43%

Запорізький інститут економіки та інформаційних ...Підрозділи - Запорізький інститут економіки та ...

<https://www.zieit.edu.ua/>

З А В Д А Н Н Я Всеукраїнської українознавчої гри «Соняшник» 5–6 класи ...

0.22%

З А В Д А Н Н Я Всеукраїнської українознавчої гри «Соняшник» 5–6 класи в) буковинці; г) поліщуки. 6. У якому з рядків усі слова можуть належати і до дієслів, і до іменників? а) три, край, звір;

<https://ivanna-site.in.ua/images/sonyasnik/2010/5-6-kl.pdf>

Платформа IoT | Архітектура та топ-10 платформ IoT

0.22%

Платформа IoT | Архітектура та топ-10 платформ IoT

<https://uk.education-wiki.com/9690551-iot-platform>

by ВВ Шевченко · 2020 — виходу з дому все, що вам потрібно зробити, – ...

0.22%

by ВВ Шевченко · 2020 — виходу з дому все, що вам потрібно зробити, – це натиснути один раз на смартфон, щоб переконатися, що штори закриті по всьому будинку, ...

https://er.nau.edu.ua/bitstream/NAU/50632/1/%D0%A4%D0%9A%D0%9A%D0%9F%D0%86_2020_123_%D0%A8%D0%B5%D0%B2%D1%87%D0%B5%D0%BD%D0%BA%D0%BE%D0%92%D0%92.pdf

Нові технології «Розумний дім. "Розумний будинок" своїми ...

0.43%

Нові технології «Розумний дім. "Розумний будинок" своїми ...

<https://allcosmoshop.ru/uk/otoplenie/the-latest-technology-smart-home-smart-home-with-their-own-hands-a-scheme-and-equipment/>

by ЮО Гунда · 2020 — Фреймворк схожий на інтерфейс прикладного прог...

0.22%

by ЮО Гунда · 2020 — Фреймворк схожий на інтерфейс прикладного програмування (API), хоча технічно фреймворк включає API. Як впливає з назви, фреймворк.

<http://lib.pnu.edu.ua:8080/bitstream/123456789/10768/1/%D0%9C%D0%B0%D0%B3%D1%96%D1%81%D1%82%D0%B5%D1%80%D1%81%D1%8C%D0%BA%D0%B0%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B0.%20%D0%93%D1%83%D0%BD%D0%B4%D0%B0.pdf>

by ОО Полуянов · 2021 — TypeScript. Він реалізує основну та додаткову ф...

0.22%

by ОО Полуянов · 2021 — TypeScript. Він реалізує основну та додаткову функціональність як набір бібліотек TypeScript, які ви імпортуєте у свої програми.

https://ela.kpi.ua/bitstream/123456789/44024/1/Poluianov_bakalavr.pdf

by КА Литвиненко · 2020 — Постачальники послуг можуть бути введені в ...

0.22%

by КА Литвиненко · 2020 — Постачальники послуг можуть бути введені в компоненти як залежності, що робить ваш код модульним, багаторазовим та ефективним.

https://ela.kpi.ua/bitstream/123456789/36449/1/Litvinenko_bakalavr.pdf

by ВА Тимошенко · 2019 — Як компоненти, так і служби – це класи, де деко...

0.22%

by ВА Тимошенко · 2019 — Як компоненти, так і служби – це класи, де декоратори позначають їх тип і надають метадані, які повідомляють Angular, як їх використовувати.

https://ela.kpi.ua/bitstream/123456789/29297/1/Тymoshenko_bakalavr.pdf

виробника Espressif з інтерфейсом Wi-Fi. Крім Wi-Fi мікроконтролер відр...

0.22%

виробника Espressif з інтерфейсом Wi-Fi. Крім Wi-Fi мікроконтролер відрізняється можливістю виконувати програми з зовнішньої флеш-пам'яті з інтерфейсом SPI. Модуль йде з базовою прошивкою AT v0.19 SDK

https://er.knutd.edu.ua/bitstream/123456789/9703/1/MSIE2018_P085-086.pdf/

Aug 10, 2019 — char exampleArray[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd' }; ...

0.22%

Aug 10, 2019 — char exampleArray[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd' }; for(int i = 0; i <= 11; i++) { print(exampleArray[i]) }.

<https://stackoverflow.com/questions/2939869/what-is-an-off-by-one-error-and-how-do-i-fix-it>

i<5;++i) and for (int i=0;i<5;i++) in C? - Quora

0.22%

i<5;++i) and for (int i=0;i<5;i++) in C? - Quora

<https://www.quora.com/Whats-the-difference-between-for-int-i-0-i-5-++i-and-for-int-i-0-i-5-i++-in-C>



[Home](#)

[Blog](#)

[Testimonials](#)

[About Us](#)

[Privacy](#)

Copyright © 2022 Plagiarism Detector. All right reserved