

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
Програмне забезпечення системи оптимізації споживання
енергії для розумного будинку

Виконав здобувач вищої освіти
IV курсу, групи КІ20-ЗСК
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Плотніков В.А.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Босько В.В.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний

Кафедра Кібербезпеки та програмного забезпечення

Рівень вищої освіти бакалавр

Галузь знань . 12 "Інформаційні технології"

Спеціальність 123 "Комп'ютерна інженерія"

Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

" " 2023 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Плотнікову Владиславу Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи оптимізації споживання енергії для розумного будинку

2. Керівник роботи Босько Віктор Васильвич, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від " " 2023 року №

3. Строк подання роботи до захисту 22.05.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи. Метою роботи є розробка програмного забезпечення системи оптимізації енергоспоживання для розумного будинку

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію

6. Висновки.

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму системи «Розумний дім» 1 аркуш

Блок-схема роботи підпрограми керування 1 аркуш

7. Дата видачі завдання « » 20 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем керування	10.03.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2023 р.	
3.	Розробка моделі компонента	20.03.2023 р.	
4.	Розробка структур даних	25.03.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2023 р.	
6.	Програмування алгоритмів	10.04.2023 р.	
7.	Оформлення ПЗ	17.05.2023 р.	
8.	Попередній захист роботи	22.05.2023 р.	

Дата видачі завдання
« » 2023р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« » 2023р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Плотніков В.А. Програмне забезпечення системи оптимізації споживання енергії для розумного будинку. 123 Комп'ютерна Інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній кваліфікаційній бакалаврській роботі розроблено програмне забезпечення, яке призначено для управління розумним будинком з підсистемою оптимізації електроспоживання.

Метою розробки є програмне забезпечення системи оптимізації споживання енергії для розумного будинку.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі C++.

Ключові слова: комп'ютерна інженерія, мікроконтролер, дані, smart-house.

ANNOTATION

Plotnikov V.A. Smart home energy optimization system software. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this qualifying bachelor's thesis, software is developed, which is intended for managing a smart house with a subsystem for optimizing electricity consumption.

The purpose of the development is the software of the energy consumption optimization system for a smart home.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on personal computers of IBM PC architecture with Windows XP/Vista/7/8/10.

The program was developed in the C++ environment.

Keywords: computer engineering, microcontroller, data, smart-house.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	1
ВСТУП.....	2
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	3
1.1 Призначення системи.....	3
1.2 Область застосування.....	4
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	6
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти)	6
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання	15
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	16
3.1 Опис функціонування системи	16
3.2 Розробка структурної схеми.....	22
3.3 Розробка функціональної схеми	35
3.4 Розробка діаграми процесів.....	38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	40
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	40
4.2 Захист розробленого програмного забезпечення.....	56
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	58
6 ОСНОВНІ ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68

ВКРБ-123.23.0023.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Плотніков В.А.			Програмне забезпечення системи оптимізації споживання енергії для розумного будинку	Лім.	Аркуш	Аркушів
Перев.		Босько В.В.				Б	1	76
Н.контр.		Гермак В.С.				<small>Ков</small> ЦНТУ КІ20-3СК		
Затв.		Смірнов О.А.						

ВСТУП

Актуальність теми. Будинок, в якому самі по собі відкриваються двері, включаються побутові прилади, регулюється температура і засуваються жалюзі, схожий на кадр з фільму про далеке майбутнє. Однак майбутнє набагато ближче, ніж нам здається. Система «розумний будинок» дозволяє повною мірою відчувати блага технічного прогресу і позбавляє людину від вирішення безлічі побутових завдань. Вперше поняття «розумний будинок» з'явилося в 50-х роках минулого століття. Прародителькою системи, здатної контролювати обстановку в цілому будинку, є технологія Java. Розробники цієї технології намагалися впровадити її в побутові прилади, тим самим зробивши їх більш «інтелектуальними».

Наприклад, вже в той час почали з'являтися перші вбудовуванні мікрохвильові печі, кондиціонери, здатні регулювати мікроклімат приміщення залежно від погоди за вікном і т. д.

Можливості системи «розумний будинок» багатогранні. Наприклад, щоб запобігти ймовірність пограбування, коли в будинку нікого немає, система імітує присутність господаря шляхом керування жалюзі, включення/вимикання світла і т. д. Якщо ж зловмисники все ж проникають всередину приміщення чи відбувається інша екстраординарна ситуація, система миттєво сповіщає про це господаря. Крім того, технологія «розумний будинок» дозволяє структурувати роботу всього технічного та інженерного обладнання, задавши йому певний сценарій.

Наприклад, перед вашим пробудженням система нагріє підлоги у ванній кімнаті, включить музичний центр, налаштує роботу кондиціонера на задану температуру, відрегулює оптимальну вологість в приміщенні і вирішить безліч інших побутових завдань.

«Розумний будинок» – це високотехнологічна система, яка може об'єднати всі комунікації вашого дому, і керувати ними одним натисканням кнопки.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми і ПЗ дозволяють успішно вирішувати задачі управління «Розумним будинком» і таким чином, виходячи з вищеперерахованого, розробка програмного забезпечення для вирішення завдання оптимізації енергоспоживання у даній кваліфікаційній бакалаврській роботі є актуальною задачею.

Кафедра _ КБПЗ _ 2023 рік

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Зараз зростає популярність використання систем розумного будинку, які забезпечують більш комфортне, безпечне та економічне життя вдома. Одним із головних завдань таких систем є оптимізація споживання енергії, зокрема електроенергії, яка є дуже важливою у сучасному світі. Розумний будинок забезпечує ефективне використання електроенергії та її раціональне споживання за рахунок використання сучасних технологій та алгоритмів.

Розумні будинки, як і більшість досягнень сучасної техніки, початково з'явилися на сторінках фантастичних оповідань. Але матеріалізуватись ідея почала лише у ХХ-му сторіччі після широкого введення електрики у будівлях і розвитку інформаційних технологій. Перше повідомлення про віддалені прилади контролю можна віднести до розробки Ніколою Тесла дистанційного керування судами та транспортними засобами у 1898 році.

Електричні побутові прилади почали з'являтися між 1915 та 1920 рр. І одразу продемонстрували готовність суспільства замінити роботу домашнього персоналу дешевими механічними пристроями. Правда на той час, проблема енергозбереження при використанні нових технологій ще вирішена не була. Тому, певний час, новітні технології були доступні лише дуже заможним людям.

Ідеї більш розвинені до понять сучасних систем автоматизації будинку були продемонстровані на ярмарках у Чикаго (1934) та Нью-Йорку. У «великому яблуці» трохи пізніше (1964-65), представили плани електрофікованих та автоматизованих приміщень. У решті-решт перший серйозний аналог розумного дому з'явився у 1966 році. Це була експериментальна система домашньої автоматизації — «домашній комп'ютер Ехо IV». Його винахідник — Джим Сазерленд, інженер компанії Westinghouse Electric. Його технологія була

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

приватним, некомерційним проектом. Перші «дротові будинки» були зведені американськими винахідниками-любителями у 1960-х, але вони були суттєво обмежені можливостями тогочасних технологій.

Уперше термін «розумний будинок» був вигаданий Американською Асоціацією Housebuilders у 1984 році. Із винаходом мікроконтролерів, вартість на електроприлади швидко падала. Ця ж установа зазначила, що таке помешкання відмінне від звичайного своєю здатністю забезпечувати продуктивне та ефективне використання робочого та житлового середовища.

За цим, віддалені інтелектуальні технології керування були прийняті будівельною промисловістю, яка поступово почала вводити їх не лише у бізнес установах, але і у домашніх помешканнях.

Під час активної домашньої автоматизації 90-х років інформатика та телевізійні системи були поєднані для підтримки інтелектуальних можливостей приміщень. У 1995 році винахідники технологій Java оголосили одним із основних призначень даної технології — «збільшення інтелекту побутових приладів».

Сьогодні технології дозволяють збирати домашню автоматику покомпонентно: обирати лише ті функції розумного будинку, які дійсно потрібні користувачу. Тепер новітні технології керування приміщенням з'являються щодня. Навіть речі, котрі раніше розглядалися лише як красиві предмети інтер'єру тепер можуть виконувати ряд мультимедійних або побутових функцій.

Технології автоматизації «розумний будинок», які є буденними за кордоном вже довший час, з року в рік стрімко набувають популярності в Україні і у Львові в тому числі. Система «розумного будинку» створена для покращення умов проживання та полегшення щоденних клопотів у побуті, економії часу і коштів.

Система «розумний будинок» умовно ділиться на кілька самостійних підсистем: безпека, освітлення, multiroom та клімат-контроль.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Система безпеки розумного будинку включає в себе такі елементи, як сигналізація, протипожежні датчики, а також датчики, що реагують на несправність комунікативних ліній. Під час відсутності господаря в домі автоматично включається сигналізація та протипожежні датчики, а в разі виникнення небезпечної ситуації, будь то небажане вторгнення або загоряння, система сповістить вас про це на віддалене пристрій — мобільний телефон, планшет або комп'ютер. Також програма негайного оповіщення зреагує і в тому випадку, якщо в будинку несподівано протечуть труби чи станеться витік газу.

Система освітлення розумного будинку управляє всіма освітлювальними елементами об'єкта. При необхідності вона відключає непотрібні в даний момент джерела світла, автоматично регулює ступінь освітленості в залежності від часу доби і навіть пори року, а також включає світло, як тільки ви з'являєтеся в будинку. Крім того, щоб відвернути увагу зловмисників, у системі можна встановити програму періодичного включення/вимикання світла у якому-небудь приміщенні під час вашої відсутності в будинку.

Система *multiroom* відповідає за розподіл аудіо — і відеосигналу по всій квартирі. Вона може включати і вимикати техніку, передавати сигнал на всі пристрої, регулювати звук, а також створювати оптимальні умови для перегляду відео або прослуховування музики. Крім того, якщо інтегрувати систему *multiroom* з системою безпеки, то запис з камер відеоспостереження можна переносити на будь-який підключений до програми пристрій. Наприклад, якщо ви знаходитесь далеко від дому, ви завжди можете спостерігати за тим, що відбувається під час вашої відсутності з допомогою допоміжних пристроїв комп'ютера або мобільного телефону, підключеного до системи.

Система клімат-контроль в «розумному будинку» керує пристроями, що відповідають за опалення, кондиціонування, очищення і зволоження повітря. Вибравши необхідний режим, ви можете налаштувати систему так, щоб вона нагрівала приміщення до вашого приходу, періодично очищала повітря, а також зволожувала його до оптимальних показників. Крім своєї зручності, ця система

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

приваблива ще і можливістю заощаджувати енергоресурси. Клімат-контроль самостійно визначає необхідний рівень температури, вимикає прилади в разі перегріву і реагує на погоду за вікном.

Крім вищеназваних підсистем, «розумний будинок» має масу інших допоміжних функцій. Наприклад, ви можете автоматизувати роботу системи так, щоб обмежити доступ дітей до небезпечних елементів, встановити режим поливу і секретів осіннього догляду за газоном: допоможемо йому перезимувати і прокинутися навесні красивим, включити цілодобове внутрішнє і зовнішнє відеоспостереження, налаштувати роботу техніки на необхідні режими і т. д.

В кінці 2016 року Марк Цукерберг продемонстрував свою розробку інтелектуалізації будинку. Протягом минулого року глава Facebook займався створенням «розумного помічника», якого назвав «Джарвіс». У цій системі об'єднані стандартні елементи «розумного будинку» і унікальні напрацювання. «Джарвіс» відрізняє голос Цукерберга і його дружини від чужих голосів, стежить за переміщеннями по дому їхньої дочки. У його функції входить приготування тостів, включення і виключення аудіо-, відеотехніки, світла, управління шторами і подача чистого одягу. Команди помічникові можна віддавати голосом або за допомогою текстових повідомлень в месенджері. У «Джарвіс» запроваджено алгоритм розпізнавання осіб з Facebook.

В Україні про будівництво першого інтелектуального житлового комплексу в 2003 році оголосила компанія «ТММ». Новобудову звали за адресою: вул. Звіринецька, 59. У житловому комплексі «Тріумф» реалізували новітні технології автоматизації будівлі: телекомунікаційну оптику, систему антизаоплення і поліпшену систему пожежної безпеки. Уже тоді інвесторам пропонували за додаткову плату встановити в квартирі повний комплект системи «розумний дім»: дистанційне керування світлом, електро побутовими приладами, клімат-контроль. Будівництво проекту преміум-класу завершили в другому кварталі 2008 року.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

У 2015 році презентували перший серійний енергоефективний будинок OptimaHouse, побудований в Київській області. У новобудові реалізували новітні розробки автоматизації житла з використанням еко-матеріалів. У наступні роки українські забудовники стали частіше інтегрувати елементи систем «розумне будівлю» і «розумний будинок» в проекти житлової багатоквартирної забудови. До 2017 року під впливом споживчого попиту сформувався стандарт базової комплектації систем автоматизації.

У 2017 році в такі системи впроваджують і інші елементи: клімат-контроль, імітацію присутності, систему multiroom.

Автоматизація будівель підвищує енергоефективність і спрощує експлуатацію житла і підвищує комфортність проживання.

У базову комплектацію системи «розумний дім» входить:

- система пожежної безпеки;
- система антизаплення;
- система безпеки;
- ресурсозберігаючі елементи.

Автоматизація будівлі закладена у вартість житла. В окремих випадках забудовник передбачає базові комунікації для установки системи «розумний дім» в кожній квартирі. Інтеграція повного комплекту системи відбувається за індивідуальним замовленням і за додаткову плату.

Системи автоматизації розумного дому поділяються на два типи автоматизація новобудівель та автоматизація будівель які раніше не були обладнані для встановлення таких систем. Вже досить багато Українських житлових комплексів таких як ЖК «Tetris Hall», КД «Art Hall», ЖК «Skyline», ЖК «Zverinetskiy», ЖК «Busov Hill», ЖК «Obolon Residence», ЖК «Триумф», ЖК «Park Stone», ЖК «Edelweiss» встановлені автоматизовані системи опалення, а з початку 2017 року забудовник «Укрбуд», розпочав встановлення української розробки системи розумний дім від компанії «Clap».

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Для додаткової економії опалення будівлі компанія встановить систему центрального вентилявання з рекуператорами. Завдяки цьому приміщення будуть провітрюватися без відкриття вікон, що зменшить ситуації охолодження будинку і скоротить витрати на прогрів будівлі. Скоротить витрату тепла в квартирах установка радіаторів з терморегуляторами і поквартирні лічильники опалення - такі передбачені в цьому комплексі.

У місцях загального користування в новобудові встановлять датчики руху: світло на поверхах та сходової клітки буде запалюватися тільки, коли в цих місцях будуть проходити люди. Це дозволить скоротити витрати на електроенергію.

У проекті ретельно розробляють комплекс заходів з безпеки майбутніх мешканців. Камери відеоспостереження встановлять на поверхах, сходовій клітці, при вході, на даху стилобату, по периметру будівлі і на паркінгу. Так співробітники зможуть цілодобово контролювати ситуацію в комплексі. Доступ до будівлі, ліфти і місця загального користування буде обмежений за допомогою електронної карткової системи. Частина приміщень буде доступна мешканцям і співробітникам комплексу, інша частина - тільки домовласникам.

У квартирах планують встановити системи антизатоплення. Це передбачає розміщення датчиків в місцях, куди може потрапити вода. При попаданні рідини на них - водопостачання квартири автоматично блокується, а власнику приходить повідомлення про потоп. Ця система допоможе уникнути затоплення сусідів і витрати на ремонт в квартирі після потопу.

1.2 Область застосування

Система розумного будинку може застосовуватися у багатьох галузях, включаючи житлову, комерційну та промислову нерухомість. Найпоширеніша область застосування систем розумного будинку - це побутові умови, де система

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

може бути встановлена в будинках, квартирах, оселях для людей похилого віку, дітей та людей з обмеженими можливостями.

Крім того, системи розумного будинку застосовуються в комерційних приміщеннях, таких як офіси, готелі, ресторани та магазини, де вони можуть допомогти в підтримці комфорту та безпеки клієнтів та працівників, а також знизити витрати на електроенергію та інші послуги.

У промисловості системи розумного будинку використовуються для контролю та автоматизації виробничих процесів, управління системами опалення та вентиляції, а також для забезпечення безпеки працівників.

Отже, системи розумного будинку можуть бути застосовані в різних галузях та сферах, де вони можуть покращити якість життя та роботи людей, знизити витрати на енергоспоживання та збільшити безпеку.

Таким чином, основною метою кваліфікаційної роботи є створення автоматизованої системи управління розумним будинком із системою оптимізації енергоспоживання.

Буде розглянуто питання оптимізації споживання електроенергії системою розумного будинку. Ми дослідимо основні принципи роботи таких систем, порівняємо споживання електроенергії в звичайному будинку та в системі розумного будинку, детально розглянемо функції, які допомагають знизити споживання електроенергії, а також розглянемо переваги та недоліки використання систем розумного будинку.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти)

Розумний дім - це технологічно продуманий будинок, у якому використовуються різноманітні системи автоматизації та керування, щоб зробити життя в будинку більш комфортним, безпечним та економічним. Рішення, що використовуються в розумному домі, можуть бути різними, в залежності від потреб та бажань власника будинку. Давайте розглянемо кілька найпоширеніших рішень, що використовуються в розумних будинках.

Системи автоматизації освітлення та опалення

Розумний дім може мати систему, яка автоматично регулює температуру та освітлення в будинку, залежно від часу доби, дня тижня, погодних умов та потреб власника. Це може допомогти знизити енергоспоживання та зберегти кошти на комунальні послуги.

Системи безпеки

Розумний дім може мати систему безпеки, яка складається з камер відеоспостереження, датчиків диму та вуглекислого газу, датчиків води, дверей та вікон, що допомагають власнику контролювати стан будинку в режимі реального часу та уникнути небезпечних ситуацій.

Голосові асистенти

У розумному домі можна встановити голосового асистента, який допоможе виконувати різноманітні завдання, такі як вмикання світла, зміна температури, відтворення музики та багато іншого.

Системи управління розвагами

Розумний дім може мати систему, яка дозволяє власнику керувати звуком

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

та відео на різних пристроях, таких як телевізори, аудіосистеми та інші.

Основними окремими блоками керування розумним домом є:

Освітлення

При проектуванні будинку, величезні зусилля витрачаються на те, щоб створити в будинку правильне освітлення. Світло повинен бути спроектований так, щоб під кожне настрій і під кожную ситуацію в будинку можна було створити індивідуальне освітлення. Але таке рішення - досить складне: в приміщенні доводиться розміщувати кілька груп світла, іноді - до десяти. І кожна група керується своїм вимикачем. У підсумку, створення потрібного освітлення перетворюється в гру на вимикачах. З підключенням освітлення до системи Розумний Дім - завдання спрощується. Одним натисканням кнопки на панелі управління, Ви включаєте певний сценарій освітлення. Сценаріїв для управління світлом може бути дуже багато, їх можна вибирати за настроєм («Романтична вечера», «Домашній кінотеатр», «Вечірка», ін.).

Налаштувавши ландшафтне освітлення, Ви зможете створити по-справжньому казкову атмосферу вечора у себе у дворі або в саду. Підсвічування клумб, окремих дерев, садових доріжок, басейну і архітектурних елементів будинку, зробить незабутніми вечірні прогулянки по заміському ділянці. Можливості розумного будинку дозволяють втілити, практично, всі Ваші ідеї.

Безпека

Розумний Дім забезпечує безпеку Вашої оселі, як від зовнішнього проникнення, так і від аварій всередині будинку.

До всіх вікнах і дверях підключаються датчики відкриття. У кімнатах встановлюються датчики руху та відеокамери. При бажанні, Ви з будь-якої точки світу, завжди будете бачити, що відбувається у Вашому домі. Це важливо, як при захисті будинку від сторонніх, так і в разі, якщо Ви залишаєте дітей вдома з нянею.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Окремо, в будинку, встановлюються датчики витоку газу і протікання води. При спрацьовуванні вони перекривають захисні клапани, і Ваш будинок залишається цілим і неушкодженим.

Відеоспостереження

Встановивши камери спостереження, Ви завжди можете контролювати ситуацію в Вашому домі. При цьому, камери можуть виконувати не тільки охоронну функцію. Ви можете простежити за тим, як доглядає за близькою людиною доглядальниця або займається з дитиною няня. Приклад прекрасного взаємодії системи відеоспостереження та домофона: Ви, перебуваючи поза домом, можете побачити гостя, який зателефонував в домофон, і, при необхідності, навіть, впустити його в будинок.

Мікроклімат

У вашому домі завжди підтримується комфортна температура і вологість. В потрібний час включається провітрювання. У нічний час, Розумний Дім охолоджує спальню і забезпечує вам міцний і здоровий сон. З настанням ранку він включає підігрів підлоги в спальні і у ванній кімнаті. Де б Ви не знаходилися і що б ви не робили, Розумний Дім стежить, щоб вам було комфортно.

Коли всі їдуть, і будинок залишається порожнім, опалення переходить в економний режим і Розумний Дім чекає вашого приїзду, економлячи ресурси.

Мультирум

Система мультирум - дозволить вам слухати музику в будь-якій кімнаті, без установки в кожній з них аудіосистем. При цьому, в кожній кімнаті може звучати своя музика. Управління всією системою мультирум відбувається за допомогою планшета або телефону.

На динаміки системи може бути виведений звук з домофона. Вона ж може озвучувати будь-які попередження або події. Наприклад, при відкритті гаражних воріт, Ви можете отримати оповіщення про те, що хтось приїхав.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Догляд за садом

Система розумного будинку стежить за вологістю ґрунту в Вашому саду і поливає його, не даючи рослинам засохнути. Вона вибирає потрібний режим поливу для кожної зони з рослинами. Якщо Ваші рослини вимагають поливу строго за розкладом - Розумний Дім зробить і це. Догляд за садом може бути повністю автоматичним і не вимагати Вашої участі.

Всі ці «електронні чудеса», на превеликий жаль, не можуть працювати без електрики, а точніше без електроенергії належної якості. Приведемо простий приклад, з якого стане ясно, що саме розуміють під «електроенергією належної якості». Дуже часто ми, як споживачі, навіть не замислюємося над тим, чому саме трапилася та чи інша поломка і у всьому прагнемо звинуватити недбайливого виробника. Перш ніж когось звинувачувати, варто уважно прочитати інструкцію з експлуатації, наприклад рідкокристалічного телевізори Full HD 1080 фірми SONY BRAVIA. Там можна побачити такі цікаві цифри, як напруга 230 В ($\pm 10\%$). Це означає, що виробник гарантує працездатність свого виробу тільки в тому випадку, якщо напруга у Вашій домашній мережі знаходиться в межах від 207 до 253. В інших випадках Ваші претензії щодо якості виробу будуть відхилені як безпідставні. Зробіть вимір напруги у Вашій домашній мережі — багато хто з Вас будуть неприємно здивовані отриманими результатами.

Особливо це стосується будівель, які знаходяться в сільській місцевості, де напруга в 190 В вважається межею мрій. Природно, що якщо Ви плануєте встановити в будинку систему управління «Розумний будинок», так і просто використовувати складну побутову техніку, то без нормалізатор (стабілізатора напруги) Вам просто не обійтися. Без цього приладу вся «розумна» електроніка просто не буде нормально функціонувати. Такий прилад буде цілком доречний як у міській квартирі, так і в заміському будинку.

Крім цього можна подбати про аварійному електропостачанні будинку на випадок відключення електроенергії. Для цих цілей можна встановити

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

використання датчиків, а також за аналогією - на об'єкти іншої природи, наприклад, біологічні.

Датчики за своїм призначенням і технічної реалізації близькі до поняття «вимірювальний інструмент» («вимірювальний прилад»). Однак показання приладів сприймаються людиною, як правило, безпосередньо (за допомогою дисплеїв, табло, панелей, світлових і звукових сигналів і ін.), В той час як показники датчиків вимагають перетворення в форму, в якій вимірювальна інформація може бути сприйнята людиною. Датчики можуть входити до складу вимірювальних приладів, забезпечуючи вимір фізичної величини, результати якого потім перетворюються для сприйняття оператором вимірювального приладу.

В автоматизованих системах управління датчики можуть виступати в ролі ініціюючих пристроїв, приводячи в дію обладнання, арматуру і програмне забезпечення. Показання датчиків в таких системах, як правило, записуються на накопичувач для контролю, обробки, аналізу і виводу на дисплей або принтер. Величезне значення датчики мають в робототехніці, де вони виступають в ролі рецепторів, за допомогою яких роботи і інші автоматичні пристрої отримують інформацію з навколишнього світу і своїх внутрішніх органів.

У побуті датчики використовуються в термостатах, вимикачі, термометрах, барометрах, смартфонах, посудомийних машинах, кухонних плитах, тостери, праски та іншої побутової техніки.

Класифікація датчиків

За методом вимірювання:

- активні (генераторні);
- пасивні (параметричні).

За вимірюваним параметром

- датчики тиску;
- розрідження;
- датчики витрати;

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- рівня;
- ємнісні;
- ультразвукові;
- температури;
- термопара;
- датчик теплового потоку;
- датчик концентрації;
- переміщення;
- положення;
- безконтактні;
- фотодатчики;
- фотодіод;
- датчик кутового положення;
- датчик вібрації;
- датчик віброприскорення (акселерометр);
- датчик вологості;

За принципом дії

- волоконно-оптичні;
- оптичні датчики (фотодатчики);
- магнітоелектричний датчик (на основі ефекту Холла);
- п'єзоелектричний датчик;
- ємнісний датчик;
- індуктивний датчик.

За середовищі передачі сигналів

- провідні;
- бездротові.

За кількістю вхідних величин

- одномірні;
- багатовимірні;

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

За технологією виготовлення

- елементні;
- інтегральні.

Інтерфейси з'єднання та зчитування даних з датчиків

Для різних типів датчиків різні інтерфейси з'єднання, аналогові датчики в більшості випадків використовуються для систем де не потрібна цифрові дані тож і з'єднання аналогових датчиків є стандартним, а для підключення аналогових датчиків до мікроконтролера використовуються аналогові порти контролера. Для підключення цифрових та бінарних датчиків є декілька інтерфейсів, основними з яких є:

I²C інтерфейс

I²C — послідовна шина даних для зв'язку інтегральних схем, розроблена фірмою Philips на початку 1980-х як проста шина внутрішнього зв'язку для створення керуючої електроніки. Використовується для з'єднання низькошвидкісних периферійних компонентів з материнською платою, вбудовуваними системами та мобільними телефонами. Назва є аббревіатурою слів Inter-Integrated Circuit. 1 жовтня 2006 року скасовані ліцензійні відрахування за використання протоколу I²C. Однак, відрахування зберігаються для виділення ексклюзивної підлеглої адреси на шині I²C.

I²C використовує дві двонапрямлені лінії, підтягнуті до напруги живлення та керовані через відкритий колектор або відкритий стік — послідовна лінія даних (SDA, англ. Serial Data) і послідовна лінія тактування (SCL, англ. Serial CLock). Стандартні напруги +5 В або +3,3 В, проте допускаються й інші.

Класична адресація включає 7-бітовий адресний простір з 16 зарезервованими адресами. Це означає до 112 вільних адрес для підключення периферії на одну шину.

Основний режим роботи — 100 кбіт/с; 10 кбіт/с в режимі роботи із зниженою швидкістю. Зауважимо, що стандарт допускає припинення тактування для роботи з повільними пристроями.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Після перегляду стандарту 1992 року стає можливим підключення ще більшої кількості пристроїв на одну шину (за рахунок можливості 10-бітної адресації), а також велику швидкість до 400 кбіт/с у швидкісному режимі.

Відповідно, доступна кількість вільних вузлів зросла до 1008. Максимальна допустима кількість мікросхем, приєднаних до однієї шини, обмежується максимальною ємністю шини в 400 пФ.

Версія стандарту 2.0, випущена 1998 року представила високошвидкісний режим роботи зі швидкістю до 3,4 Мбіт/с зі зниженим енергоспоживанням. Остання версія 2.1 2001 року включила лише незначні доопрацювання.

Послідовний периферійний інтерфейс SPI

Серійний периферійний інтерфейс або SPI - послідовний периферійний інтерфейс, служить для зв'язку мікроконтролерів. Наприклад, в якості периферії може бути: дисплей, різні датчики, FLASH пам'ять, SD карта (так, да, SD карта або «флешка», яку ви використовуєте в телефонах і фотоапаратах, обмінюється зовнішнім світом за допомогою інтерфейсу SPI) тощо. В SPI завжди є один ведучий і один / кілька ведених. Передачу даних завжди розпочинає ведучий.

В SPI використовуються чотири лінії зв'язку:

- MOSI або SI - вихід ведучого, вхід веденого (англ. MasterOutSlaveIn).

Служить для передачі даних від провідного пристрою відомому.

- MISO або SO - вхід ведучого, вихід веденого (англ. MasterInSlaveOut).

Служить для передачі даних від відомого пристрою ведучому.

- SCLK або SCK - послідовний тактовий сигнал (англ. SerialClock).

Служить для передачі тактового сигналу для ведених пристроїв.

- CS або SS - вибір мікросхеми, вибір веденого (англ. Chip Select, Slave Select).

SPI може бути реалізований в мікроконтролері апаратно, тоді завдання з управління інтерфейсом вирішується для кожного мікроконтролера окремо, тому що реалізації можуть бути різними. Наприклад, для ATmega328P (мікроконтролер фірми Atmel) при роботі з SPI нам потрібно самим програмним шляхом

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

рази перевищувало кількість транзисторів в 16-розрядному мікропроцесорі i8086.

На сьогоднішній день існує більше 200 модифікацій мікроконтролерів, сумісних з i8051, що випускаються двома десятками компаній. А також велика кількість мікроконтролерів інших типів. Популярністю у розробників користуються 8-бітові мікроконтролери PIC від фірми «Microchip Technology» і «AVR» від фірми «Atmel».

При проектуванні мікроконтролерів доводиться дотримувати баланс між розмірами і вартістю з одного боку і гнучкістю і продуктивністю з іншого. Для різних застосувань оптимальне співвідношення цих і інших параметрів може розрізнятися дуже сильно. Тому існує величезна кількість типів мікроконтролерів, що відрізняються архітектурою процесорного модуля, розміром і типом вбудованої пам'яті, набором периферійних пристроїв, типом корпусу.

В той час, як 8-розрядні процесори загального призначення повністю витіснені продуктивнішими моделями, 8-розрядні мікроконтролери продовжують широко використовуватися. Це пояснюється тим, що існує велика кількість застосувань, в яких не потрібна висока продуктивність, але важлива низька вартість. В той же час, є мікроконтролери, з більшими обчислювальними можливостями, наприклад цифрові сигнальні процесори.

Обмеження за ціною і енергоспоживанням стримують також зростання тактової частоти контролерів. Хоча виробники прагнуть забезпечити роботу своїх виробів на високих частотах, вони, в той же час, надають замовникам вибір, випускаючи модифікації, розраховані на різні частоти і напругу живлення. У багатьох моделях мікроконтролерів використовується статична пам'ять для ОЗП і внутрішніх регістрів. Це дає контролеру можливість працювати на менших частотах і навіть не втрачати дані при повній зупинці тактового генератора. Часто передбачені різні режими енергозбереження, в яких відключається частина периферійних пристроїв і обчислювальний модуль.

Окрім ОЗП, мікроконтролер може мати вбудовану незалежну пам'ять для зберігання програми і даних. У багатьох контролерах взагалі немає шин для

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Мікроконтролери виготовляють багато фірм основними з них є AVR виготовляє фірма Atmel та мікроконтролери PIC які виготовляє фірма Microchip Technology Inc. Фірми конкуренти Atmel та Microchip Technology Inc мають Гарвардську архітектуру також є не менш відома фірма STMicroelectronics яка виготовляє мікроконтролери серій STM.

Мікроконтролери AVR

AVR — родина восьмибітових мікроконтролерів фірми Atmel.

Мікроконтролери AVR мають гарвардську архітектуру[1] (програма і дані розташовані в різних адресних просторах) і систему команд, близьку до ідеології RISC. Процесори AVR мають 32 8-бітових реєстри загального призначення. Максимальна тактова частота — 20 мегагерц (XMEGA AVR — до 32 МГц), короткі команди виконуються за один такт.

На відміну від «ідеального» RISC, реєстри не абсолютно ортогональні:

- деякі команди працюють тільки з реєстрами r16—r31;
- 16-бітний результат множення (у тих моделях, в яких є модуль множення) завжди розміщується в реєстровій парі r0:r1;
- для непрямой адресації пам'яті даних використовуються три «здвоєні» 16-бітові реєстри-вказівники X (r26:r27), Y (r28:r29) та Z (r30:r31);
- в командах ADIW, SBIW додавання та віднімання короткої константи від 16-бітного аргумента можуть використовуватися лише вказані вище пари X, Y, Z та пара (r24:r25);
- В командах роботи з пам'яттю програм LPM, SPM використовується лише реєстрова пара Z (r30:r31);
- Команди прямого переходу та виклику підпрограм ICALL, IJMP також можуть використовувати лише реєстрову пару Z.

Робота з периферійними пристроями здійснюється через адресний простір даних. Для зручності з першими 64-ма адресами периферійних пристроїв можна працювати за допомогою «скорочених» команд IN/OUT доступу до простору вводу-виводу (I/O — Input/Output). Роботу з окремими бітами периферійних

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

позначається весь комплекс пристроїв, правил та технічних засобів, що регламентують та забезпечують обмін інформацією між мікропроцесором (включаючи пам'ять) та зовнішніми пристроями. Головними в інтерфейсі є шини, або, як їх ще часто називають, магістралі. Магістраль — це сукупність провідників, для яких строго нормовані логічні рівні «0» та «1». Потужність сигналів на шинах має бути достатньою для живлення необхідної кількості приєднаних до них пристроїв. Для забезпечення цієї потужності використовуються спеціальні мікросхеми — шинні підсилювачі (ШП).

За призначенням, шини поділяються на три типи: адресні, даних та керування.

Але реально як в мікропроцесорній техніці, так і в комп'ютерній часто дві шини суміщують шляхом мультиплексування, що дещо знижує їх швидкодію, але набагато зменшує кількість виводів мікросхем.

AVR мікроконтролери використовують безліч розробників, одними із яких є апаратно обчислювана система під назвою Arduino, розроблений компанією Arduino Software.

Платформи побудовані на основі мікроконтролерів

Arduino (Ардуіно) — апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері (наприклад: Processing, Adobe Flash, Max/MSP, Pure Data, SuperCollider). Інформація про плату (рисунок друкованої плати, специфікації елементів, програмне забезпечення) знаходяться у відкритому доступі і можуть бути використані тими, хто воліє створювати плати власноруч.

Плата Arduino складається з мікроконтролера Atmel AVR, а також елементів обв'язки для програмування та інтеграції з іншими пристроями. На

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

площадки або штирові роз'єми. Також існує багато різних зовнішніх плат розширення, які називаються «shields» («щити»), які приєднуються до плати Arduino через штирові роз'єми.

ESP8266 (Node MCU)

ESP8266 - мікроконтролер китайського виробника Espressif з інтерфейсом Wi-Fi. Крім Wi-Fi мікроконтролер відрізняється можливістю виконувати програми з зовнішньої флеш-пам'яті з інтерфейсом SPI. Мікроконтролер привернув увагу в 2014 році в зв'язку з виходом перших продуктів на його базі за незвичайно низькою ціною. Навесні 2016 року розпочалося виробництво ESP8285, що поєднує ESP8266 і флеш-пам'ять на 1 МБайт. Восени 2015 року Espressif представила розвиток лінійки - мікросхему ESP32.

Характеристики ESP8266:

- 80 MHz 32-bit процесор Xtensa L106. Можливий негарантований розгін до 160 МГц.
- IEEE 802.11 b / g / n Wi-Fi. Підтримується WEP і WPA / WPA2.
- 14 портів введення-виведення (з них можливо використовувати 11), SPI, I²C, I²S, UART, 10-bit АЦП.
- Живлення 2,2 ... 3,6 В. Споживання до 215 мА в режимі передачі, 100 мА в режимі прийому, 70 мА в режимі очікування. Підтримуються три режими зберігання, все без збереження з'єднання з точкою доступу: Modem sleep (15 мА), Light sleep (0.4 мА), Deep sleep (15 мкА).

Мікроконтролер не має на кристалі користувальницької незалежній пам'яті. Виконання програми ведеться з зовнішньої SPI ПЗУ шляхом динамічного навантаження необхідних ділянок програми в кеш інструкцій. Вантаження йде апаратно, прозора для програміста. Підтримується до 16 МБ зовнішньої пам'яті програм. Можливий Standard, Dual або Quad SPI інтерфейс.

Виробник не надає документації на внутрішню периферію мікроконтролера. Замість цього він дає набір бібліотек, через API яких програміст отримує доступ до периферії. Оскільки ці бібліотеки інтенсивно використовують

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

приводи для штор і жалюзі, якими не керує інтелектуальна система розумний будинок: керовані з пульта інфрачервоного передавача, і не мають ніякого зв'язку з контролером розумного будинку. Проте, такі пристрої теж вважаються окремими елементами розумного будинку.

Клапани низького, середнього і високого тиску - це більш прості механізми для управління водопровідними мережами (в т. Ч. І пожежогасіння), газовими трубами і вентиляцією. Клапани застосовуються в системах контролю: газові - в системах контролю за газом, водні - контроль водозабезпечення, запобігання затопленню і пожежогасіння.



Рисунок 2.4 - Електричний управляємий водяний клапан

Дуже корисно для здоров'я, і варто відзначити систему вентиляційних клапанів, яка спрацьовує від датчиків вуглекислого газу (CO₂). Система вивітрює шкідливий для людини газ і створює сприятливий мікроклімат. До подібних елементарним механізмам можна віднести електромагнітні замки та інші елементи контролю доступу в приміщення, що входять в систему безпеки розумного будинку. Такі пристрої в основному не мають мікропроцесорів, мають тільки два положення: відкрито і закрито.

9) Підсистема забезпечення безпеки передачі даних(система шифрування)
– Base64/AES/RSA.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну дипломну роботу, реалізації підлягає програмне забезпечення для системи управління енергозбереженням розумного будинку. В процесі розробки кваліфікаційної бакалаврської дипломної роботи необхідно виконати наступний обсяг проекту:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методика побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

Виходячи з кваліфікаційної роботи потрібно розробити програмне забезпечення системи управління для проекту «Розумний дім» та розробити саму автоматизовану систему управління з встановленням необхідних датчиків та систем які б дали змогу оптимізувати енергозбереження в розумному будинку

Система «розумний» дім може бути як і багато функціональною так і з обмеженням по функціоналу, для побудови можна використовувати як і мікроконтролери так і мікрокомп'ютери. В першому випадку функціонал системи може бути обмежений так як сучасні системи потребують великої обчислювальної потужності, яку не мають звичайні мікроконтролери, що не можна сказати про мікрокомп'ютери. Мікрокомп'ютери включають в себе окремий процесор, ОЗУ, відео процесор, що дасть можливість завантажити на такому пристрої навіть графічний інтерфейс або операційну систему, також мікрокомп'ютер має безліч портів та інтерфейсів вводу виводу даних.

Але в побудові системи на мікрокомп'ютері є й мінуси одним із яких є велика вартість та обслуговування такої системи. Що не можна сказати про використання мікроконтролерів для побудови тієї ж самої системи, ціна яких майже в половину менша.

Для побудови системи можна використовувати мікрокомп'ютери різних розробників таких як Raspberry PI, Orange PI, Banana PI, Latte Panda, навіть така відома фірма як Asus має свої однопалатні комп'ютери під назвою Asus Tinker Board. Якщо ж побудову робити на мікроконтролерах то в цьому випадку є також не мала кількість різних платформ для реалізації різних системи, прикладом є платформи STM, Arduino, Espruino, модулі ESP і безліч інших. Можлива навіть побудова системи з використанням поєднання мікрокомп'ютерів з мікроконтролерами.

В випадку побудови системи «розумний» дім потрібно також враховувати

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

від погодних умов та наявності людей в будинку.

3. Система безпеки: камери відеоспостереження, датчики диму та вуглекислого газу, які забезпечують контроль за безпекою в будинку.
4. Система звукового та візуального сповіщення: система, яка дозволяє відправляти повідомлення про стан будинку через звукові та візуальні сигнали.
5. Система керування електроживленням: система, яка забезпечує ефективне використання енергії, регулює напругу та струм у мережі, контролює стан електроприладів в будинку.
6. Система домашнього кінотеатру: система, яка забезпечує якісний звук та зображення для перегляду відео та фільмів.
7. Система управління водопостачанням: система, яка забезпечує оптимальне використання води в будинку.

3.2 Розробка структурної схеми

Структурна схема системи – це сукупність об'єктів та частин та взаємозв'язки між ними. Вона дозволяє зрозуміти як система працює, які компоненти входять у її склад, як вони пов'язані між собою та як вони взаємодіють.

Структурна схема може бути використана для опису будь-якого виду систем, від механічних та електричних до програмних та біологічних. Зазвичай вона складається з блоків, які представляють компоненти системи, та ліній, які показують зв'язки між ними.

Призначенням структурної схеми є наглядне відображення складових частин розробляємої системи, її основних блоків, вузлів та взаємозв'язок між ними.

Структурна схема розробленої системи зображена на рисунку 3.1.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

3.3 Розробка функціональної схеми

Функціональна схема - це графічне зображення функцій та зв'язків між ними у системі, що відображає спосіб реалізації певної функції в системі.

Схема зазвичай містить блоки, які представляють окремі функції, та стрілки, які показують зв'язки між ними. Кожен блок може мати вхідні та вихідні порти, які представляють вхідні та вихідні параметри блоку.

Функціональна схема може бути використана для моделювання та проектування будь-яких систем, від електричних та механічних до програмних та біологічних. Вона дозволяє розглядати систему на рівні функцій та взаємодії між ними, що сприяє більш ефективному проектуванню та аналізу системи.

Існує кілька переваг функціональної схеми порівняно зі структурною:

1. Фокус на функціональній діаграмі: Функціональна схема зосереджена на функціональних аспектах системи, що дозволяє розуміти, які функції виконує система, і як вони пов'язані між собою.
2. Спрощення складної системи: Функціональна схема дозволяє розглядати систему як набір функцій та їх зв'язків, що спрощує розуміння та аналіз складної системи.
3. Розуміння меж компонентів: Функціональна схема дозволяє краще зрозуміти, як кожен компонент системи взаємодіє з іншими, і як вони разом працюють для досягнення спільної мети.
4. Підвищення ефективності: Функціональна схема дозволяє розуміти, як кожен компонент системи впливає на інші, що може допомогти знайти способи підвищення ефективності та оптимізації системи в цілому.
5. Універсальність: Функціональна схема може бути використана для моделювання та проектування будь-яких систем, від електричних та механічних до програмних та біологічних.

Функціональна схема розробленої системи зображена на рисунку 3.3. З рисунку видно, що розроблена система складається з наступних блоків:

- мобільного додатку для зв'язку з системою;
- блоку управління з інженерними системами;
- блок управління кліматом;
- системи безпеки;
- системи управління мультимедією;
- блоку управління ландшафтом.

До кожного блоку входять відповідні датчики.

Датчики які використовуються в системах розумного будинку:

- сутінковий датчик - призначений для управління штучним світлом в залежності від рівня освітленості навколо;
- датчик витоку чадного газу - визначає наявність чадного газу і різкі зміни температури в приміщенні;
- датчик руху - призначений для виявлення присутності та пересування людей в приміщенні;
- датчик освітленості – використовується для вимірювання рівня освітленості як всередині так і зовні приміщення;
- датчик температури – використовується для вимірювання навколишньої температури повітря як всередині приміщення, так і зовні;
- датчик затоплення - призначений для виявлення затоплення і протікання в приміщенні, що охороняється;
- датчик відкриття дверей / вікна - призначений для відстеження відкриття вікон, дверей, гаражних воріт, ролетів тощо;
- датчик пожежі (диму) - використовується для відстеження загорянь і наявності диму в приміщеннях.

Для різних типів датчиків різні інтерфейси з'єднання, аналогові датчики в більшості випадків використовуються для систем де не потрібна цифрові дані тож і з'єднання аналогових датчиків є стандартним, а для підключення аналогових

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

датчиків до мікроконтролера використовуються аналогові порти контролера.

Центральним блоком керування буде виступати платформа ESP8266 яка включає в себе процесор Tensilica L106 32-bit, який працює на частоті 80 MHz, також платформа включає в себе 80 кб ОЗУ та 35 кб ІРАМ, А також плата має вбудований Wi Fi який працює на частоті 2.4 Ghz, що дає можливість приймати та передавати дані бездротовими технологіями. Платформа має 14 портів введення-виведення (з них можливо використовувати 11), SPI, I²C, I²S, UART, 10-bit АЦП.

Для управління системою з мобільного телефону розроблено відповідне АРІ.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

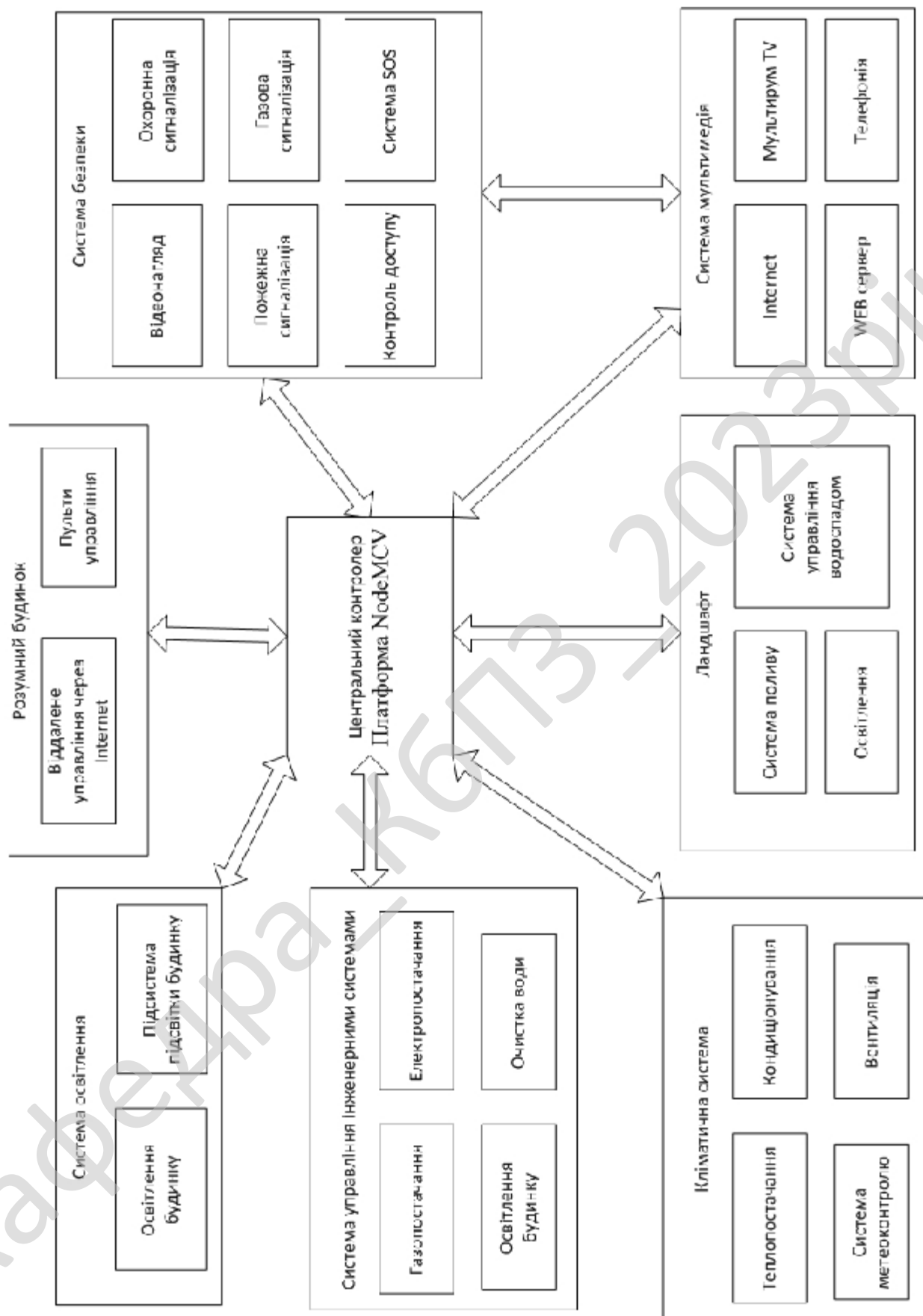


Рисунок 3.2 – Функціональна схема системи «Розумний дім»

3.4 Розробка діаграми процесів

Діаграма процесів - це графічне зображення процесу, що ілюструє потік даних між різними компонентами системи. Кожен блок представляє оброблюваний процес, а стрілки показують потік даних між процесами та об'єктами.

Діаграма процесів в управлінні розумним будинком містить такі блоки процесу, як збір даних з датчиків, аналіз отриманих даних, виконання дій та зворотній зв'язок з користувачем.

На діаграмі знаходяться такі різні процеси, які відбуваються в розумному будинку.

1. Керування освітленням:

- датчик руху виявляє людину у кімнаті;
- система керування освітленням включає світильники у кімнаті;
- якщо людина покидає кімнату, то освітлення автоматично вимикається.

2. Керування температурою:

- датчик температури вимірює температуру у кімнаті;
- система керування температурою увімкнути або вимкнути систему опалення або кондиціонування повітря для досягнення заданої температури;
- система може навчитися розпізнавати уподобання власника будинку і автоматично налаштувати температуру відповідно до них.

3. Керування безпекою:

- датчик руху та дверей виявляє незвичайну активність у будинку;
- система безпеки може автоматично сповістити власника будинку про незвичайну активність через мобільний додаток;
- власник будинку може перевірити відеозапис з камери безпеки для отримання додаткової інформації про подію.

4. Керування електроживленням:

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

- система може виявляти витрату енергії та керувати споживанням електроенергії;
- система може автоматично вимкнути підключені до розеток прилади, які не використовуються;
- система може зберігати історію споживання електроенергії та давати рекомендації щодо оптимального споживання.

Діаграма процесів розробленої системи зображена на рисунку 3.3 .

ВКРБ-123.23.0023.00.00.ПЗ

Арк.

Вим.	Арк.	№ докум.	Підпис	Дата		
						46

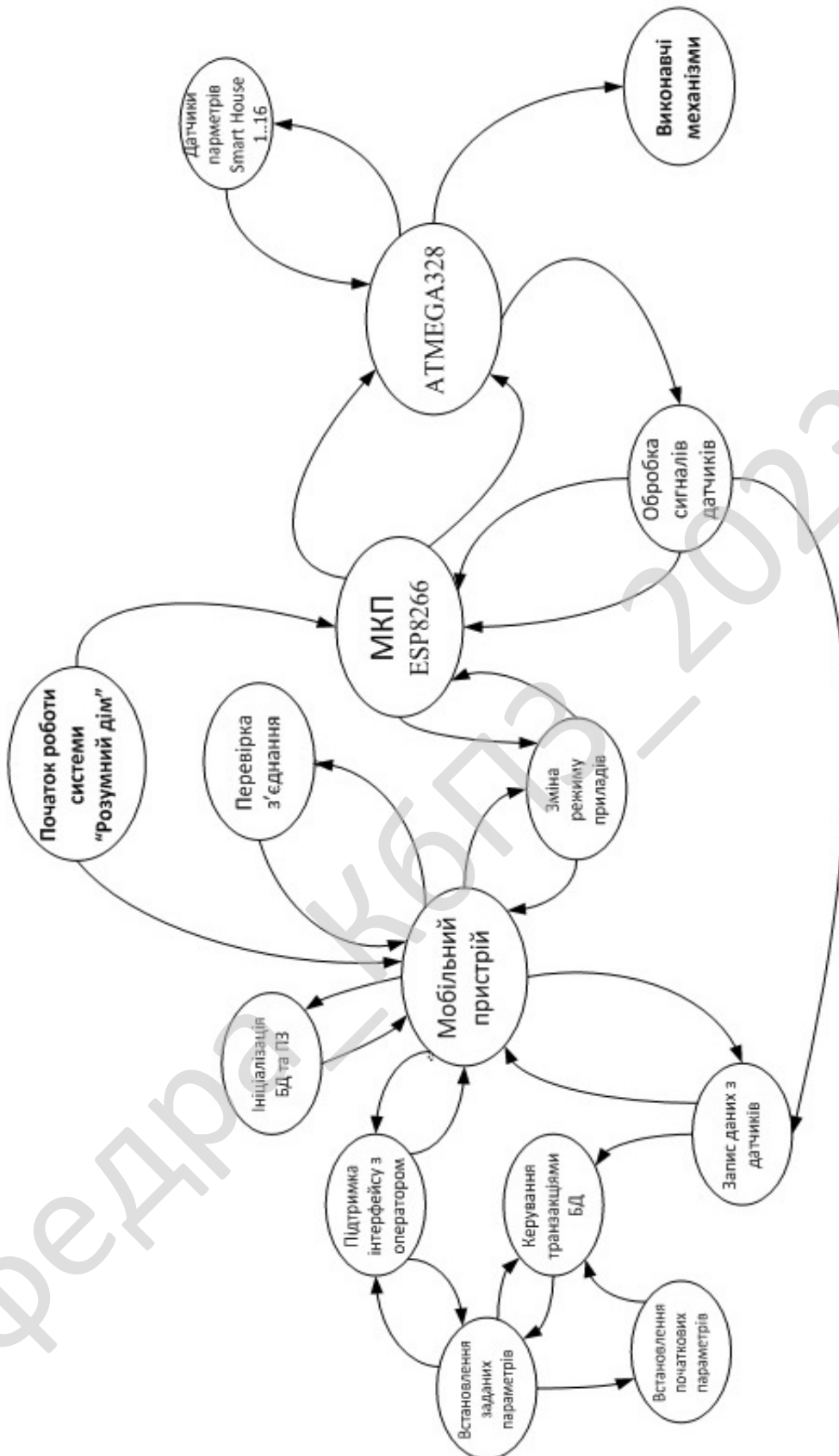


Рисунок 3.3 – Діаграма процесів системи «Розумний дім»

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

Кафедра _ КБПЗ _ 2023 рік

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

Під час написання даної кваліфікаційної роботи було розроблено програмний продукт, а саме віконний застосунок, призначений для керування процесом управління розумним будинком та енергоспоживанням.

Для системи розумного будинку був використаний ADSL роутер Huawei HG532e.

Технічні характеристики:

- тип антени все напрямлена
- підтримка WiFi 802.11 b, g, n
- бездротова передача даних (Мбіт / сек) 300
- передача даних LAN (Мбіт / сек) 10/100
- роз'єм LAN 4
- вбудований Брандмаузер
- підтримка шифрування WEP, WPA, WPA2

Центральним блоком керування буде виступати платформа ESP8266 яка включає в себе процесор Tensilica L106 32-bit, який працює на частоті 80 MHz, також платформа включає в себе 80 кб ОЗУ та 35 кб ІРАМ, А також плата має вбудований Wi Fi який працює на частоті 2.4 Ghz, що дає можливість приймати та передавати дані бездротовими технологіями. Платформа має 14 портів введення-виведення (з них можливо використовувати 11), SPI, I²C, I²S, UART, 10-bit АЦП.



Рисунок 4.1 – Платформа NodeMCV на основі ESP8266

Температура:

- 4овторюваність: $\pm 0,2$;
- діапазон: На 25 ± 2 ;
- час відгуку: $1 / e$ (63%) 10S;

Електричні характеристики:

- живлення: DC 3,5 ~ 5,5 В;
- струм живлення: вимір 0.3mA очікування 60μ;
- період вибірки: більше 2 секунд.

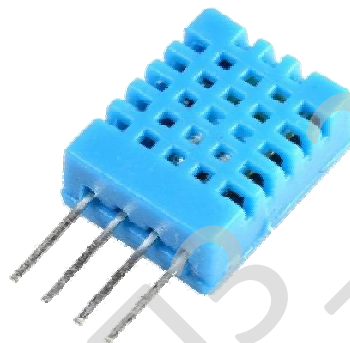


Рисунок 4.3 – Датчик температури та вологості DHT11

Також для системи можна використовувати й інші датчики наприклад BMP280. Барометричний датчик тиску BMP280 використовується для вимірювання барометричного, абсолютного, диференціального, надлишкового тиску, а також значення температури і вологості навколишнього середовища.

Для використання датчика тиску спочатку потрібно його підключити до Arduino контролера або іншому мікропроцесорній керуючому пристрою. Потім потрібно завантажити і встановити бібліотеку для роботи з датчиком. За замовчуванням, вимірювання тиску відбувається в паскалях. Чутливим елементом датчика є мембрана в корпусі, яка працює на п'єзорезистивного принципі. П'єзорезистивного ефект - залежність опору матеріалу від величини його деформації.

ВМР280 має 6-ти піновий роз'єм для підключення даних і живлення.

Характеристики:

- модель: ВМР280;
- діапазон вимірюваного тиску: 300 - 1100 гектопаскалях (+9000 - -500 м над рівнем моря);
- напруга живлення: 1,62 - 3,6;
- споживання струму в режимі «низького споживання»: 0,5 мА;
- інтерфейси: I2C, SPI;
- час спрацьовування: 2 мс;
- точність вимірювання тиску: 0,1 гектопаскалях;
- точність вимірювання температури: 0,1 °С;
- робоча температура: -40 - 85 °С.

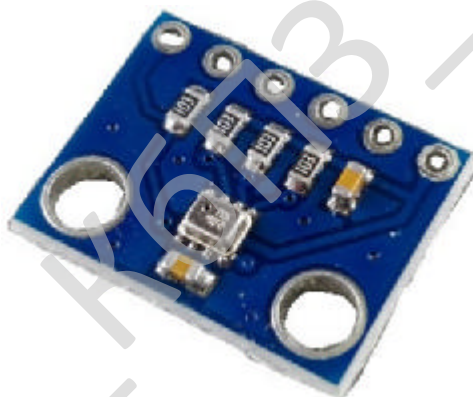


Рисунок 4.4 – Датчик температури та атмосферного тиску ВМР280

Що стосується датчиків які використовуються в системі протипожежної безпеки можна використати наступні типи датчиків: MQ-2, MQ-7 датчик даму та датчик огню.

Датчик газу MQ-2 використовується для детектування горючого газу і диму. Широкий діапазон, швидкий час реакції, висока чутливість, стабільна робота і довгий термін служби, проста схема підключення.

Основні характеристики датчика MQ-2:

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

- газ - Горючий газ, дим;
- діапазон чутливості 300-10000 ppm;
- s (опір чутливого елемента) 1 ... 20 кОм 50ppm;
- газ, для якого нормується датчик Ізобутан, 1000ppm;
- час відгуку ≤ 10 с;
- чутливість $(R \text{ в повітрі}) / (R \text{ в присутності характерного газу}) \geq 5$ с;
- R_h (опір нагрівача) $31\Omega \pm 3\Omega$;
- I_h (струм нагрівача) ≤ 180 мА;
- V_h (напруга нагрівача) $5\text{В} \pm 0,2 \text{В}$;
- P_h (потужність нагрівача) ≤ 900 мВт;
- V_c (напруга схеми) ≤ 24 В;
- Стандартні робочі умови Температура: $-10 \sim + 50 \text{ }^\circ \text{C}$, вологість: $\leq 95\%$ RH, концентрація кисню: 21% (стандартні умови);
- Умови зберігання Температура: $-20 \sim + 70 \text{ }^\circ \text{C}$, вологість: $\leq 70\%$ RH
- Конфігурація А або В (металевий або пластиковий корпус)



Рисунок 4.5 – Датчик диму MQ-2

Датчик MQ-7 служить для виявлення угарного газу.

Основні характеристики датчика MQ-7:

- детектуючий газ: чадний газ
- Нагрівач напруга: 50,2 В (acdc)

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

- робоча напруга: 140mA
- напруга петлі: 0 В (Максимальна постійного 15 В)
- навантажувальний опір: 10 К (регульована)
- виявлення концентрації діапазон: 10-1000ppm
- напруга в очищення повітря: 1.5 В
- чутливість: 3%
- час відгуку: 1 S (підігріву 3-5 хвилин)
- час відповіді: 30 s
- компонент харчування Потужність: 0.7 Вт
- робоча температура: -10 ~ 50 градусів (номінальна температура: 20 градусів)
- робоча вологість: 95% RH (номінальний вологість: 65% rh)
- термін служби: 5 років
- розмір: 35 мм × 20 мм × 11 мм
- вага: 4,8 г



Рисунок 4.6 – Датчик угарного газу MQ-7

Для системи безпеки в будинку можна використовувати наступні датчики, інфрачервоний датчик руху HC-SR501, датчик відкриття дверей MC-38, датчик

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

розбиття скла та інші.

Основні технічні характеристики HC-SR501

- дальність виявлення: 0 - 7 м
- кут спрацьовування: 110 ° на дистанції до 7 м
- харчування: 4.5 - 6 В
- вихідна напруга логічного рівня: 0 - 3.3 В
- час затримки: 0.3 - 18 секунд (регулюється)
- метод спрацьовування: L є повторюваною перемикання; Н повторюване перемикання
- споживаний струм: 65 мА
- робочі температури: -20 - +50 град/ц

МС-38 використовується для сигналізування відкриття дверей, вікон і т.д. У середині датчика знаходиться геркон з нормально-розімкнутими контактами. При закритих дверях, коли магніт перебуває близько датчика, контакти датчика замкнені і розмикаються при відкритті дверей.

Характеристики:

- розмір: 27 x 9 x 7 мм (датчик і магніт однакові)
- максимальний струм: 100 мА
- максимальна напруга: 200 V
- відстань спрацьовування: від 15 мм до 25 мм

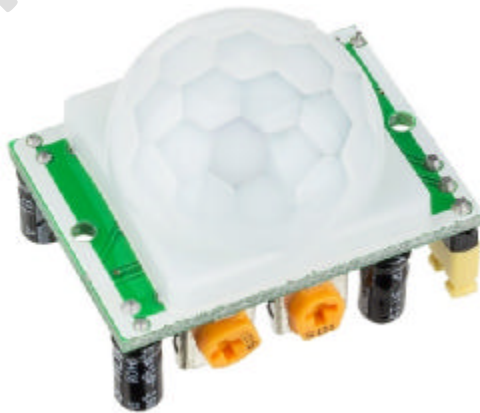


Рисунок 4.7 – Датчик руху HC-SR501

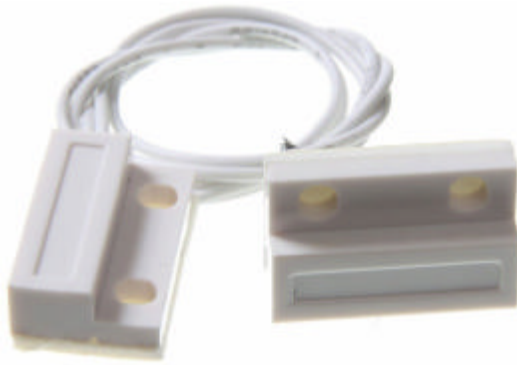


Рисунок 4.8 – Датчик відкриття дверей МС-38

Також для контролю освітленості в приміщенні можна використати датчик освітленості (фоторезистор).

Характеристики фоторезистора:

- Максимальна напруга: 150 В
- Максимальна потужність: 90 мВт
- Світлове опір (10 люкс): 5-10 кому
- Темнове опір (0 люкс): 0.5 МОм
- Спектральний пік: 540 нм
- Час відгуку: 30 мс
- Робоча температура: від -30 до +70 ° С

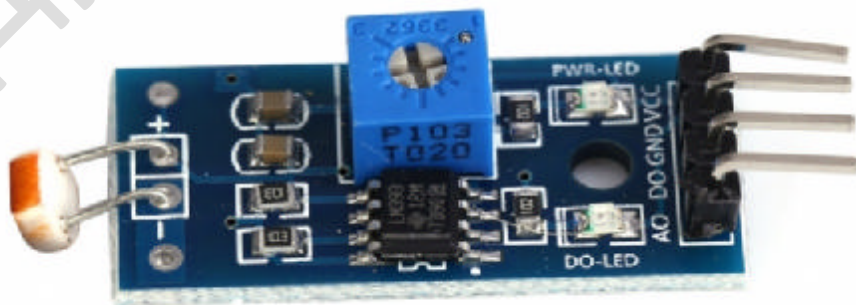


Рисунок 4.9 – Датчик освітленості

Для керування навантаженням, в системі можна використовувати як транзисторне керування виконавчими пристроями з невеликою напругою так і керування за допомогою реле які дають можливість керувати виконавчими пристроями з напругою в 220В та більше. Можна використати модулі реле 4-х каналні, 8-ми каналні та 16-ти каналні в залежності від потреби керувати великою кількістю виконавчих пристроїв.

Характеристики модулів реле:

- Струм спрацьовування: 15-20мА при напрузі 12 В
- Управління: 5В TTL, який може бути поданий безпосередньо з виходу мікроконтролера
- Комутоване навантаження: 10А при 250В.

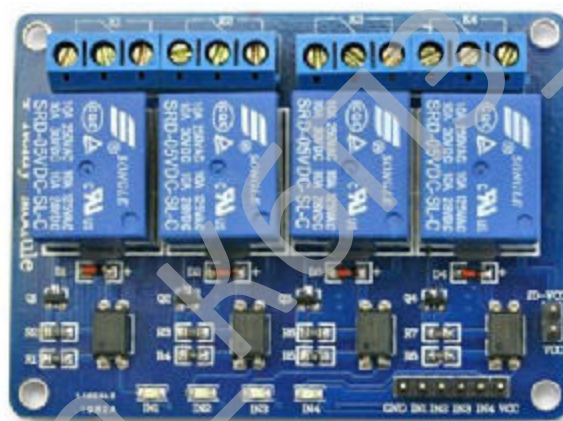


Рисунок 4.10 – Чотирьох каналний модуль-реле

З'єднання компонентів сигнальними лініями зв'язку системи між собою буде відбуватися за допомогою дротів, але є і можливість модифікації системи для використання бездротового з'єднання між сигналізацією, блоком реле, датчиками та освітленням. Для цього в системі можна використати модулі бездротового зв'язку nRF24L01+, є дві версії модуля з дальністю прийому та передачі до 100м та версія з дальністю до 1км, також модуль може працювати з 5

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

такими ж модулями цим самим створювати мережу.

Основні технічні характеристики:

- Напруга 1,9-3,6 В
- Частота 2.4 ГГц
- Кількість каналів 126
- Швидкості обміну: 256 Кбод, 1 Мбод, 2 Мбод
- модуляція GFSK
- Чутливість приймача -85 dBm при швидкості 1 Мбод
- Гранична температура повітря
- робота -40 ... 85 ° C
- зберігання -40 ... 125 ° C

Для прийому та передачі даними між роутером та модулем ESP8266 на модулі створюється Web сервер який працює за протоколом HTTP.

Візьмемо для прикладу керування освітленням, підключившись пристроєм до роутера та відкривши програмне забезпечення на своєму пристрої ми задаємо команду увімкнути світло в кімнаті. Програма в свою чергу відсилає запит до роутера з IP адресом (192.168.1.7) після чого роутер шукає ip адрес приладів що підключені. Знайшовши потрібний адрес відправляє модулю команду (192.168.1.7/relay1_on), в нашому випадку увімкнути реле, що відповідає за освітлення. Після чого в ESP8266 запрограмовано передавати високий рівень сигналу на порт до якого підключене реле відповідно команді яку ми надіслали та прийняли тобто увімкнути реле, аналогічним чином відбувається керування іншими приладами в системі.

Програмна реалізація системи

Для реалізації системи, що об'єднує описані раніше пристрої в єдину мережу, було спроектовано серверну архітектуру. Вона містить наступні процеси:

- Мобільний додаток Smart House – API для керування системою за допомогою мобільного телефону
- Система управління – побудована на мікроконтролерах

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58


```

// готовим GPI12 для реле
pinMode(relay_1, OUTPUT);
digitalWrite(relay_1, HIGH);
pinMode(relay_2, OUTPUT);
digitalWrite(relay_2, HIGH);
WiFiInit();
server.on("/1on", rel_on_1);
server.on("/1off", rel_off_1);
server.on("/2on", rel_on_2);
server.on("/2off", rel_off_2);
server.on("/signal_on", signals_on);
server.on("/signal_off", signals_off);
server.on("/svet_on", svet_on);
server.on("/svet_off", svet_off);
server.on("/a_svet_on", a_svet_on);
server.on("/a_svet_off", a_svet_off);
server.on("/data", data_on);
server.on("/t", data);
}
void loop() {
  datchiki();
  server.handleClient();
}
void datchiki() {
  temp = dht.readTemperature();
  hum = dht.readHumidity();
}
void rel_on_1() {
  digitalWrite(relay_1, LOW);
  server.send(200, "text/html", "1_off");
}
void rel_off_1() {
  digitalWrite(relay_1, HIGH);
  server.send(200, "text/html", "1_off");
}
void rel_on_2() {
  digitalWrite(relay_2, LOW);
  server.send(200, "text/html", "1_off");
}
void rel_off_2() {
  digitalWrite(relay_2, HIGH);
  server.send(200, "text/html", "1_off");
}

```

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

```

}
void signals_on() {
    Signal.print("1");
    server.send(200, "text/html", "/on");
}
void signals_off() {
    Signal.print("2");
    server.send(200, "text/html", "/off");
}
void svet_on() {
    Svet.print("1");
    server.send(200, "text/html", "Світло ввімкнена");
}
void svet_off() {
    Svet.print("2");
    server.send(200, "text/html", "Світло вимкнена");
}
void a_svet_on(){
    Svet.print("3");
    server.send(200, "text/html", "Світло в автоматичном режимі");
}
void a_svet_off(){
    Svet.print("4");
    server.send(200, "text/html", "Автоматичний режим освітлення
вимкнено");
}
void data() {
    webString = (String((float)temp) + "," + String((float)hum));
    server.send(201, "text/plain", webString);
}
void data_on(){
    data();
}

```

Допоміжний файл WIFI.ino

```

void WIFIinit() {
    // Попытка подключения к точке доступа
    WiFi.mode(WIFI_STA);
    byte tries = 11;
    WiFi.begin(ssid.c_str(), _password.c_str());
    WiFi.config(ip, gateway, subnet);
    while (--tries && WiFi.status() != WL_CONNECTED)

```

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61


```

// Signals serial : TX = digital pin 8, RX = digital pin 9
SoftwareSerial Signal(9,8);

void setup() {
  Serial.begin(9600);
  Signal.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(ton, OUTPUT);
  pinMode(data, INPUT);
}

void loop() {
  if(Signal.available()){
    esp = Signal.read();
  }
  if(esp == '1'){
    esp_data = 1;
    digitalWrite(led, HIGH);
  }
  else if(esp == '2'){
    digitalWrite(led, LOW);
    esp_data = 0;
    flag = 0;
    val = 0;
  }
  val = digitalRead(data);
  if((val == HIGH) && esp_data == 1 && flag == 0){
    Serial.println("Определенно движение");
    tone(ton, 2880, 100);
    delay(20);
    digitalWrite(led, HIGH);
    delay(100);
    digitalWrite(led, LOW);
    delay(100);
    digitalWrite(led, HIGH);
    delay(100);
    digitalWrite(led, LOW);
    flag++;
    delay(2800);
  }
  else if((val == HIGH) && esp_data == 1 && flag == 1){
    Serial.println("Определенно повторное движение, включаю серену!!!");
    flag++;
  }
}

```



```

    }
}

void days(){
    digitalWrite(led, HIGH);
    data = digitalRead(dvijenie);
    if(data == 0){
        digitalWrite(led, HIGH);
    }
}
}

```

4.1 Блок-схеми та опис алгоритмів функціонування системи

Як тільки користувач потрапив на головну сторінку розробленого додатку(для керування параметрами он-лайн) він повинен пройти авторизацію або реєстрацію у випадку відсутності аккаунта. Перший запис містить введену користувачем інформацію про його аккаунт, а другий містить інформацію потрібну для активації даного аккаунта, що зберігається у таблиці з інформацією про аккаунти, що очікують активації. Далі система надсилає листа на введену користувачем поштову адресу з посиланням, яке містить код активації аккаунта, і повідомляє користувача, що на вказану ним адресу було відправлено листа з кодом активації.

Після того як користувач перейшов по отриманому посиланню система перевіряє інформацію у цьому посиланні, а саме код активації аккаунта і якщо даний код буде збігатися з кодом у рядку в таблиці з неактивованими аккаунтами – система видалить інформацію про неактивований аккаунт користувача та встановить, що даний аккаунт користувача активовано.

Потім система перенаправляє користувача на сторінку авторизації на якій він має можливість увійти на свій аккаунт та розпочати роботу.

На рисунку 4.11 зображено блок-схему алгоритму роботи сторінки реєстрації користувача.

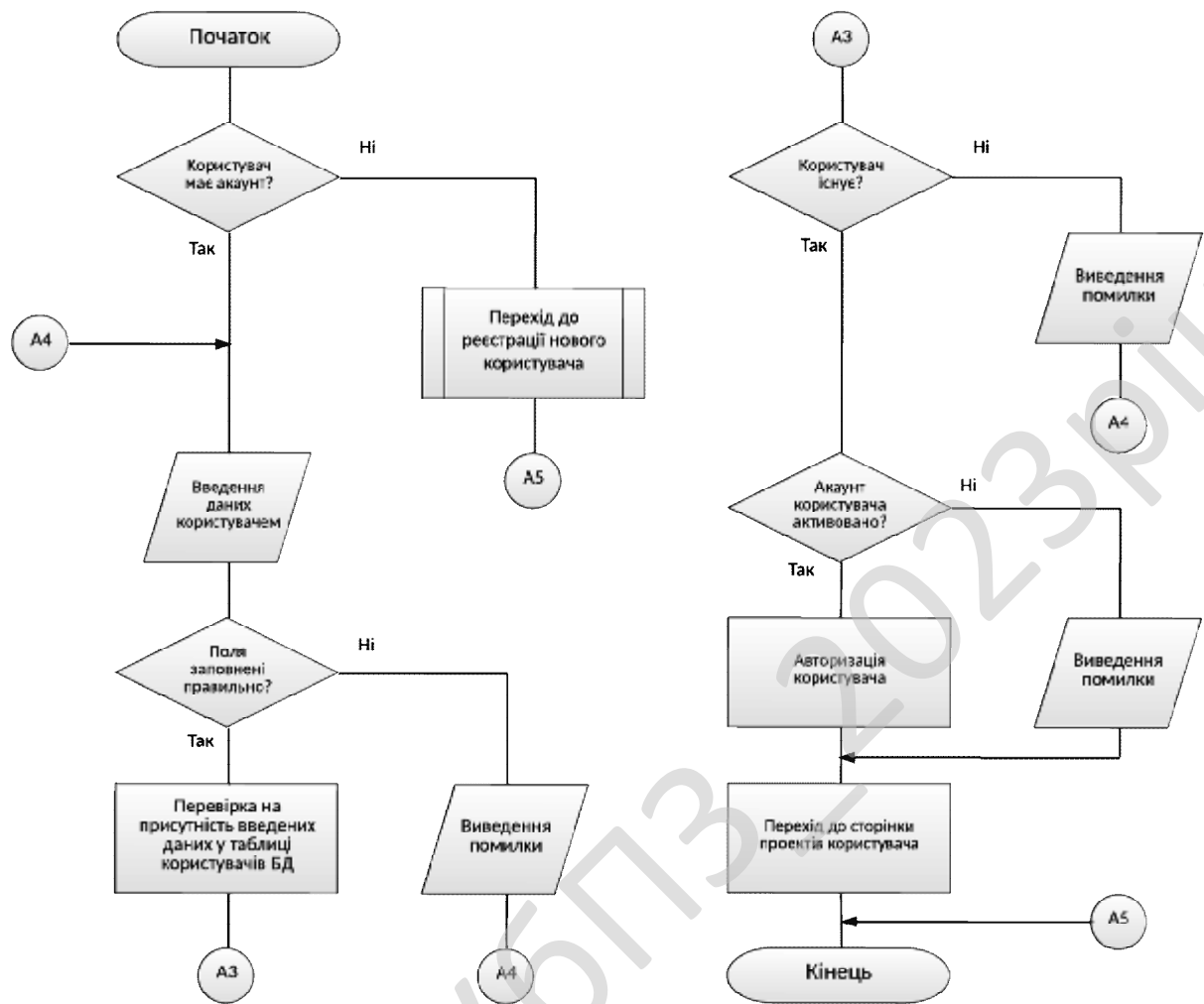


Рисунок 4.11 - блок-схема алгоритму роботи сторінки реєстрації (авторизації) користувача.

Розглянемо алгоритм роботи основної програми в режимі керування температурою в приміщенні. Його блок-схема зображена на рисунку 4.12.

Для управління необхідно зробити вибір натиснувши відповідне вікно.

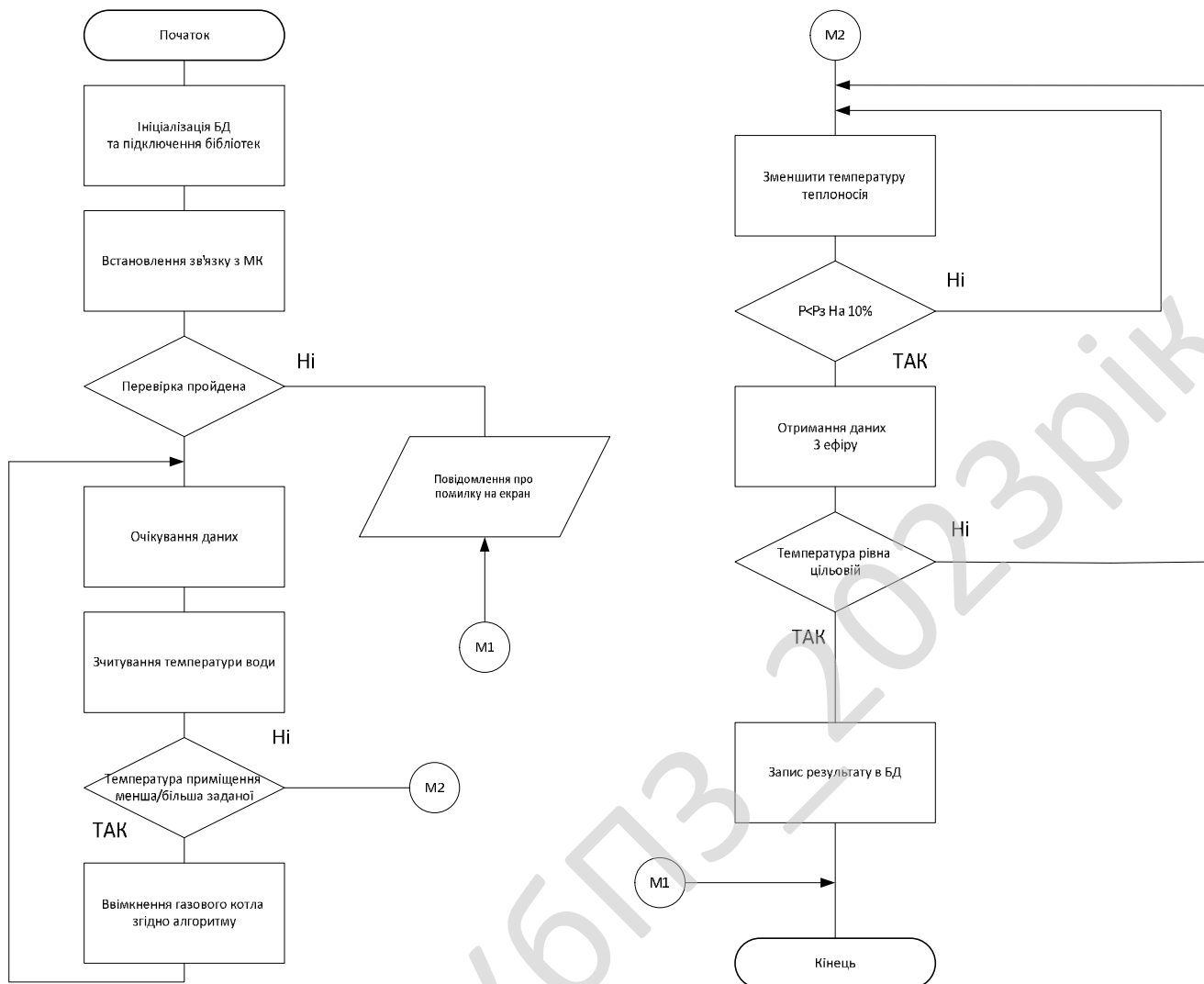


Рисунок 4.12 - Блок-схема основного алгоритму роботи системи.

4.2 Захист розробленого програмного забезпечення

Для захисту даних розробленого програмного забезпечення веб-системи для командного керування проектами було використано стандартні функції мови PHP: `password_hash()` і `password_verify()`.

`Password_hash()` використовується для хешування паролю користувача, а `password_verify()` – для перевірки вказаного паролю на відповідність до хешу.

Першим параметром функції `password_hash()` являється пароль, який необхідно захистити, а другий параметр визначає алгоритм за яким буде

RewriteEngine

On

RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]

RewriteCond %{QUERY_STRING} GLOBALS(=|\\|%[0-9A-Z]{0,2}) [OR]

RewriteCond %{QUERY_STRING} _REQUEST(=|\\|%[0-9A-Z]{0,2})

RewriteRule ^(.*)\$ index.php [F,L]

Для захисту сайту від SQL-ін'єкцій було використано php-функцію `mysql_real_escape_string`. Ця функція екранує спеціальні символи в рядку, що використовуються в SQL-запиті, беручи до уваги кодування з'єднання.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення даної магістерської роботи додатком для смартфона командного керування системами розумний дім. Для того щоб почати користуватись програмою потрібно перейти за посиланням <http://> та завантажити розроблений додаток для смартфона рисунок 5.1.

З рисунку видно, що після запуску програми спочатку відбувається вивід основного вікна програми та встановлення зв'язку з мікроконтролером. Потім здійснюється отримання поточних даних з датчиків.

Пройти авторизацію можна натиснувши на кнопку “Налаштування”. Після натиснення на цю кнопку буде відображено сторінку авторизації, яку зображено на рисунку 5.1.

Авторизація

Ваше Ім'я

Пароль

Увійти

Забули пароль?

Реєстрація

Рисунок 5.1– Сторінка авторизації користувача

Після пройденної верифікації ми отримаємо доступ до управління процесами.

На рисунку 5.2 представлено розроблену панель управління.



Рисунок 5.2 – Смарт додаток до смартфона

При натисненні наприклад віконця кондиціонер ми перейдемо до панелі зображеної на рисунку 5.3.



Рисунок 5.3 – Управління кондиціонування повітря

Точно по такому ж принципу проходить регулювання і усіх інших параметрів.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

6 ОСНОВНІ ВИСНОВКИ

В даній кваліфікаційній роботі мною було розроблена автоматизована система управління розумний дім з використанням мікроконтролерних засобів та розробкою програмного забезпечення для візуального управління процесом управління розумним будинком.

Також розроблено і спроектовано схеми: функціональна, схема зовнішніх з'єднань, принципова електрична схема, схема монтажу електрокомпонентів, та розроблено необхідне програмне забезпечення

У процесі рішення завдань, отримані наступні результати:

- проведено огляд технологій зв'язку в системах «Розумний дім»;
- проведено класифікацію існуючих архітектурних рішень та обрано оптимальну з точки зору керування архітектуру;
- Розроблено вітчизняний продукт керування системою «Розумний дім», який має більш широкі можливості, на відміну від існуючих аналогів в даному ціновому діапазоні.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. -СПб.: Питер, 2011. -496 с.
2. Лебеденко М. С. Вебометричний ранг як показник ефективності електронного ресурсу підприємства / М. С. Лебеденко // Економічний вісник Національного технічного університету України "Київський політехнічний інститут". - 2014. - № 11. - С. 401-408.
3. Лучшие публикации за сутки / Хабрахабр [Электронный ресурс]: Что такое SEO / Блог компании Trinion / Хабрахабр – Режим доступа до ресурсу: <https://habrahabr.ru/company/trinion/blog/302408/> - Дата доступу: 05.05.2017.
4. Котеров Д.В., Симдянов И. В. PHP 7 -СПб.: БХВ-Петербург, 2016. -1088 с.
5. Люк Веллинг, Лора Томсон. Разработка Web -дополнителей за допомогою PHP і MySQL : Вільямс, 2012р., - 880 с.
6. Пасічник О. Г., Пасічник О. В., Стеценко І. В. Основи веб-дизайну: Навч. посіб. -К.: Вид. група ВНУ. 2011р. -336 с:
7. Мейер Э. А. CSS. Каскадные таблицы стилей. Подробное руководство. – М.: Символ-Плюс, 2008. – 235 с.
8. Основы программирования на PHP Интернет-университет информационных технологий - ИНТУИТ.ру, 2005
9. Хоган Б. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б. Хоган, К. Уоррен, М. Уэбер, К. Джонсон, А. Годин. – СПб.: Питер, 2013. – 288 с.
10. Приемы объектно-ориентированного проектирования, из-во “ДМК”, 376 стр., 2011р.
11. Регулярные выражения. 3-е издание, из-во “Символ-Плюс”, 608 стр., 2008р.
12. JavaScript. Подробное руководство, 5-е издание, из-во “Символ-Плюс”, 992 стр., 2009р.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

13. jQuery. Подробное руководство по продвинутому JavaScript, 2-е издание, из-во “Символ-Плюс”, 624 стр., 2011р.
14. HTML и XHTML. Подробное руководство, из-во “Символ-Плюс”, 752 стр., 2012р.
15. CSS – каскадные таблицы стилей. Подробное руководство, 2-е издание, из-во “Символ-Плюс”, 576 стр., 2005р.
16. Смірнов О.А. Програмування комп'ютерних мереж. Основи HTML, CSS, JAVA-SCRIPT. Методичні вказівки. Кіровоград 2007–107 с.
17. Савельева Н.В. Основы программирования на PHP Интернет-университет информационных технологий - ИНТУИТ.ру, 2005.
18. Вязовик Н.А. Программирование на Java Интернет-университет информационных технологий - ИНТУИТ.ру, 2003
19. Алекс Феррара, Мэтью Мак-Дональд Программирование web-сервисов для .NET.
20. Сироткин С.А., Чалышев И.В., Воробьев С.Е Самоучитель WML и WMLScript СПб.: БХВ-Петербург, 2001.-240 с.
21. Непейвода Н.Н. Стили и методы программирования Интернет-университет информационных технологий - ИНТУИТ.ру, 2005.
22. Селезнев К. От SQL к NoSQL и обратно [Электронный ресурс] / Константин Селезнев. – Режим доступа: <http://www.osp.ru/os/2012/02/13014127>.
23. <https://dou.ua/> - Співтовариство програмістів.
24. <https://habr.com> - веб-сайт, який поєднує ознаки соціальної мережі і колективного блогу, створений для публікації новин, аналітичних статей, думок, пов'язаних із інформаційними технологіями, бізнесом та Інтернетом.
25. <https://getbootstrap.com/> - клієнтський фреймворк.
26. <https://idg.net.ua/blog/uchebnik-css> - Підручник CSS.
27. <http://www.elibukr.org/> - Електронна бібліотека України.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

28. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» [Електронний ресурс]. – Режим доступу к ресурсу: <http://zakon4.rada.gov.ua/laws/show/2594-15>
29. Бен Фрейн «HTML5 и CSS3 разработка сайтов для любых браузеров и устройств», 2014 г.
30. Селезнев К. От SQL к NoSQL и обратно [Електронний ресурс] / Константин Селезнев. – Режим доступу: <http://www.osp.ru/os/2012/02/13014127>.
31. Алекс Феррара, Мэтью Мак-Дональд Программирование web-сервисов для .NET.
32. jQuery. Подробное руководство по продвинутому JavaScript, 2-е издание, из-во “Символ-Плюс”, 624 стр., 2011р.
33. PHP 5, из-во “ВНУ-СПб”, 1104 стр., 2008р.
34. Гмурман В.Е. Теория вероятностей и математическая статистика / В.Е. Гмурман. – М.: Высшая школа, 2003. – 479 с.
35. Хоган Б. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б. Хоган, К. Уоррен, М. Уэбер, К. Джонсон, А. Годин. – СПб.: Питер, 2013. – 288 с.
36. Касимов Р.Р. Вдосконалення алгоритмів QoS маршрутизації в мережах з технологією IP/MPLS на основі прогнозу трафіка: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.12.02 –телекомунікаційні системи та мережі / Р.Р. Касимов. – Київ: Київський державний університет інформаційно-комунікаційних технологій, 2011. – 24 с.
37. Игнатенко Е.Г., Бессараб В.И. Алгоритм адаптивного мониторинга загрузки кластерных web-серверов / Е.Г. Игнатенко, В.И. Бессараб // Збірник тез. Нові технології в телекомунікаціях. VI міжнародний науково-технічний симпозиум. – Карпати, Вишків: ДУІКТ. – 2011. – С.34.
38. Закон України «Про Концепцію Національної програми інформатизації» (Відомості Верховної Ради (ВВР), 1998, N 27-28, ст.181) (Із змінами,

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

внесеними згідно із Законом N 5463-VI (5463-17) від 16.10.2012, ВВР, 2014, N 4, ст.61). – <http://zakon1.rada.gov.ua/laws/show/74/98-%D0%B2%D1%80>.

39. Зайченко Ю.П. Компьютерные сети / Ю.П. Зайченко. – К.: Слово, 2003. – 256 с.
40. Зубко Р.А. Алгоритми стиснення зображень в системах цифрової обробки даних / Р.А. Зубко // Восточно-Европейский журнал передовых технологий. – 2013. – №1/2(61). – С. 40-44
41. Хоган Б. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б. Хоган, К. Уоррен, М. Уэбер, К. Джонсон, А. Годин. – СПб.: Питер, 2013. – 288 с.
42. Rao, K., Yip P. (eds.), The Transform and Data Compression Handbook, CRC Press, Baton Rouge, 2001.
43. Scholz M., Fraunholz M., Selbig J., Nonlinear Principal Component Analysis: Neural Network Models and Applications, In: Gorban A. N. et al (Eds.), LNCSE 58, Springer, 2007 ISBN 978-3-540-73749-0
44. Селезнев К. От SQL к NoSQL и обратно [Электронный ресурс] / Константин Селезнев. – Режим доступа: <http://www.osp.ru/os/2012/02/13014127>.
45. Knol A., Dependency Injection with AngularJS // Knol A. - Packt Publishing, 2015 – с. 78
46. Toigo, J.W. The holy Grail of enterprise data storage. / Toigo, J.W. // Prentice-Hall, 1999 – pp. 64 – 69.
47. Офіційний сайт AngularJS. – Режим доступу <https://angularjs.org/> – Дата доступу : 27.04.2016.
48. Williamson K., Learning AngularJS. // Williamson K. - O'Reilly Media, 2015 – 212 с.
49. А. Філд, П. Харрісон Функціональне програмування. / А. Філд, П. Харрісон. – К: Москва "Мир", 1993. – 55 с.

Б. Лоусон Изучаем HTML5. Библиотека специалиста / Б. Лоусон, Р. Шарп. – Питер : Питер, 2011. – 235 с.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

50. Абрамов Н.Н. Водоснабжение. Учебник для вузов. Изд. 2-е, перераб. и доп. М., Стройиздат, 1974. 480 с.
51. Лезнов Б.С. Экономия электроэнергии в насосных установках. – М.: Энергоатомиздат, 1991. – 144 с.: ил. – (Экономия топлива и электроэнергии)
52. Карелин В.Я., Минаев А.В. Насосы и насосные станции: Учеб. для вузов.– 2-е изд., перераб. и доп.– М.: Стройиздат, 1986.– 320 с.: ил.
53. Волотковский С.А., Емец В.И., Козло В.К. и др. Типовой электропривод и автоматизация общепромышленных механизмов. - Киев: Вища школа, 1983. - 312 с.
54. Справочник по электрическим машинам : В 2 т./ Под общ. ред. И.П. Копылова и Б.К. Клокова. т. 1. – М.: Энергоатомиздат, 1988. – 456 с.: ил.
55. Справочник по гидравлике / Под ред. В.А. Большакова, – 2-е изд., перераб. и доп. – К.: Вища шк. Головное изд-во, 1984.– 343 с.
56. Шевелёв Ф.А. Исследование основных гидравлических закономерностей турбулентного движения в трубах. – М.; Л. : Госстройиздат, 1953. – 208 с.
57. Сыромятников И.А. Режимы работы асинхронных и синхронных двигателей/ Под ред. Л.Г. Мамикоянца. – 4-е изд., переработ. и доп. – М.: Энергоатомиздат, 1984. – 240 с., ил.
58. Справочник по автоматизированному электроприводу. – М.: Энергоатомиздат, 1979. – 640 с., ил.
59. Сандлер А.С. и Сарбатов Р.С. Автоматическое частотное управление асинхронными двигателями. М., "Энергия", 1974, 328 с., с ил.
60. В.П. Андреев, Ю.А. Сабинин Основы электропривода. – 2-е изд., переработ. – М.: ГЭИ, 1963. – 767 с.
61. Черный А.П., Луговой А.В., Д.И.Родькин, Сисюк Г.Ю., Садовой А.В. Моделирование электромеханических систем / Учебн. пособие. - Кременчуг : 1999. - 202с.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

62. Файнштейн В.Г., Файнштейн Э.Г. микропроцессорные системы управления тиристорными электроприводами / Под ред. О.В. Слежановского. – М.: Энергоатомиздат, 1986. – 240 с.: ил.
63. Методичні вказівки щодо виконання розділу “Електропостачання промислової установки” при дипломному проектуванні для студентів зі спеціальності 7.092203 “Електромеханічні системи автоматизації та електропривод” – денної та заочної форм навчання / П. Г. Василенко. – Кременчук: КДПУ, 2001. – 73 с.
64. Князевский Б.А., Липкин Б.Ю. Электроснабжение промышленных предприятий: Учебник. – 2-е изд., перераб. и доп. – М.: Высш. школа, 1979. – 431 с., ил.
65. Т.А. Пьявченко. Проектирование АСУТП в SCADA-системе. Технологический институт ЮФУ, Таганрог – 84 – 2007.15. Князевский Б.А., Липкин Б.Ю. Электроснабжение промышленных предприятий: Учебник. – 2-е изд., перераб. и доп. – М.: Высш. школа, 1979. – 431 с., ил.
66. Андреев Е.Б., Куцевич Н.А., Синенко О.В. SCADA-системы: взгляд изнутри. Москва: РТСофт – 176 – 2004 – ISBN: 5-9900271-1-7.
67. MasterSCADA. Основы проектирования – ИнСАТ. – 186 с.
68. Учебный курс по SCADA Системе Simatic WinCC v7.0.

					ВКРБ-123.23.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Перелік документів, що розробляються.....	5
8	Етапи розробки.....	6
9	Порядок контролю та приймання.....	6

					ВКРБ-123.23.0023.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Плотніков В				Літ.	Аркуш	Аркушів
Перевірів	Босько В.В				Б	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ20-3СК		
Затв.	Смірнов О.А.						
<i>Програмне забезпечення системи оптимізації споживання енергії для розумного будинку</i>							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку програмного забезпечення системи оптимізації споживання енергії для розумного будинку

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу, видане на кафедрі програмування та захисту інформації (нак. №8-02 від 05.01.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної бакалаврської роботи є розробка програмне забезпечення системи оптимізації споживання енергії для розумного будинку.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської дипломної роботи є відносна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРБ-123.23.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- роботу системи керування процесом водопостачання;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРБ-123.23.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

При розробці ПЗ потрібно використовувати наступні технології та мови програмування: HTML, CSS, JavaScript, бібліотека jQuery, PHP + PDO, AJAX. Для програмування мікроконтролера мова низького рівня C++. Середовище програмування – PHP-Shtorm.

					ВКРБ-123.23.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи керування – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму – 1 аркуш.
- Блок-схема алгоритму системи «Розумний дім» – 1 аркуш.
- Пояснювальна записка – 76 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи.

Постановка задачі на виконання кваліфікаційної роботи (складання ТЗ).

					ВКРБ-123.23.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної роботи.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання кваліфікаційної роботи на попередній захист 21.05.2023 р.

9.2 Подання кваліфікаційної роботи на захист 16.05.2023 р.

					ВКРБ-123.23.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник кваліфікаційної бакалаврської роботи
_____ Босько В.В.

*Програмне забезпечення системи оптимізації споживання енергії для
розумного будинку*

Лістинг програми

Код документу 12
Носій: CD/DVD-диск

Загальна кількість аркушів: 24

Літера: РП

```

// Copyright (C) Плотников Владислав
// 2023

ESP8266
Головний файл ESP.ino
#include <SoftwareSerial.h>
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
ESP8266WebServer server(80);
#define DHTPIN 0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Signals serial : TX = digital pin 13, RX = digital pin 15
SoftwareSerial Signal(13,15);
// software serial : TX = digital pin 4, RX = digital pin 5
SoftwareSerial Svet(4,5);

String webString = ""; // текст для отправки
String _ssid = "Huawei_HG532e_20663"; // Для хранения SSID
String _password = "ivanchsg"; // Для хранения пароля сети
String _ssidAP = "SmartHome"; // SSID AP точки доступа
String _passwordAP = ""; // пароль точки доступа
int relay_1 = 14;
int relay_2 = 12;
float temp, hum;

IPAddress apIP(192, 168, 1, 7);
// Определяем переменные
IPAddress ip(192,168,1,7); //статический IP
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);
void setup() {
  dht.begin();
  Serial.begin(115200);
  Signal.begin(115200);
  Svet.begin(115200);
  // готовим GPI12 для реле
  pinMode(relay_1, OUTPUT);
  digitalWrite(relay_1, HIGH);
  pinMode(relay_2, OUTPUT);
  digitalWrite(relay_2, HIGH);
  WiFiInit();
  server.on("/1on", rel_on_1);
  server.on("/1off", rel_off_1);
  server.on("/2on", rel_on_2);
  server.on("/2off", rel_off_2);
  server.on("/signal_on", signals_on);
  server.on("/signal_off", signals_off);
  server.on("/svet_on", svet_on);
  server.on("/svet_off", svet_off);
  server.on("/a_svet_on", a_svet_on);
  server.on("/a_svet_off", a_svet_off);
  server.on("/data", data_on);
  server.on("/t", data);
}
void loop() {
  datchiki();
  server.handleClient();
}

void datchiki() {
  temp = dht.readTemperature();

```

```

    hum = dht.readHumidity();

}
void rel_on_1() {
    digitalWrite(relay_1, LOW);
    server.send(200, "text/html", "1_off");
}
void rel_off_1() {
    digitalWrite(relay_1, HIGH);
    server.send(200, "text/html", "1_off");
}
void rel_on_2() {
    digitalWrite(relay_2, LOW);
    server.send(200, "text/html", "1_off");
}
void rel_off_2() {
    digitalWrite(relay_2, HIGH);
    server.send(200, "text/html", "1_off");
}
void signals_on() {
    Signal.print("1");
    server.send(200, "text/html", "/on");
}
void signals_off() {
    Signal.print("2");
    server.send(200, "text/html", "/off");
}
void svet_on() {
    Svet.print("1");
    server.send(200, "text/html", "Світло ввімкнена");
}
void svet_off() {
    Svet.print("2");
    server.send(200, "text/html", "Світло вимкнена");
}
void a_svet_on(){
    Svet.print("3");
    server.send(200, "text/html", "Світло в автоматичном режимі");
}
void a_svet_off(){
    Svet.print("4");
    server.send(200, "text/html", "Автоматичний режим освітлення вимкнено");
}
void data() {
    webString = (String((float)temp) + "," + String((float)hum));
    server.send(201, "text/plain", webString);
}
void data_on(){
    data();
}
Допоміжний файл WIFI.ino
void WIFIinit() {
    // Попытка подключения к точке доступа
    WiFi.mode(WIFI_STA);
    byte tries = 11;
    WiFi.begin(_ssid.c_str(), _password.c_str());
    WiFi.config(ip, gateway, subnet);
    while (--tries && WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(3000);
    }
    if (WiFi.status() != WL_CONNECTED)
    {
        // Если не удалось подключиться запускаем в режиме AP
        Serial.println("");
        Serial.println("WiFi up AP");
        StartAPMode();
        server.begin();
    }
}

```

```

    Serial.println("Server started");
}
else {
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    server.begin();
    Serial.println("Server started");
}
}
}
bool StartAPMode()
{
    WiFi.disconnect();
    WiFi.mode(WIFI_AP);
    WiFi.softAPConfig(apIP, apIP, IPAddress(255, 255, 255, 0));
    WiFi.softAP(_ssidAP.c_str(), _passwordAP.c_str());
    return true;}
Сигналізація Arduino 1
Файл Signalization.ino
#include <SoftwareSerial.h>
int led = 13;
int ton = 12;
int data = 4;
int val = 0;
int flag = 0;
int esp_data;
char esp;
char for_data;
// Signals serial : TX = digital pin 8, RX = digital pin 9
SoftwareSerial Signal(9,8);

void setup() {
    Serial.begin(9600);
    Signal.begin(115200);
    pinMode(led, OUTPUT);
    pinMode(ton, OUTPUT);
    pinMode(data, INPUT);
}
void loop() {
    if(Signal.available()){
        esp = Signal.read();
    }
    if(esp == '1'){
        esp_data = 1;
        digitalWrite(led, HIGH);
    }
    else if(esp == '2'){
        digitalWrite(led, LOW);
        esp_data = 0;
        flag = 0;
        val = 0;
    }
    val = digitalRead(data);
    if((val == HIGH) && esp_data == 1 && flag == 0){
        Serial.println("Определенно движение");
        tone(ton, 2880, 100);
        delay(20);
        digitalWrite(led, HIGH);
        delay(100);
        digitalWrite(led, LOW);
        delay(100);
        digitalWrite(led, HIGH);
        delay(100);
        digitalWrite(led, LOW);
        flag++;
        delay(2800);
    }
    else if((val == HIGH) && esp_data == 1 && flag == 1){

```

```

        Serial.println("Определенно повторное движение, включаю серену!!!");
        flag++;
    }
    else if(flag >= 2){
        for(;;){
            esp_data = for_data;
            tone(ton, 2880, 100);
            delay(5);
            tone(ton, 2800, 100);
            leds();
            if(for_data == 0)break;
        }
    }
    Serial.println(esp);
    Serial.println(esp_data);
    Serial.println(flag);
}
void leds(){
    digitalWrite(led, HIGH);
    delay(100);
    digitalWrite(led, LOW);
    delay(100);
}
Зовнішнє освітлення Arduino 2
Файл Svet.ino
#include <SoftwareSerial.h>
int flag = 0;
int light = A3;
int raw = 0;
int data = 0;
int dvijenie = 2;
char esp = "";
int led = 12;

// software serial : TX = digital pin 4, RX = digital pin 5
SoftwareSerial Svet(9,8);

void setup() {
    Serial.begin(9600);
    Svet.begin(115200);
    pinMode(light, INPUT);
    pinMode(led, OUTPUT);
    digitalWrite(led, HIGH);
    pinMode(dvijenie, INPUT);
}

void loop() {
    if (Svet.available() > 0) {
        esp = Svet.read();
    }
    Serial.println(esp);

    if((esp == '1') && flag == 0){
        digitalWrite(led, LOW);
    }
    else if((esp == '2') && flag == 0){
        digitalWrite(led, HIGH);
    }
    Serial.print(esp);
    Serial.print(flag);
    if(esp == '3'){
        flag = 1;
    }
    raw = analogRead(light);
    //Serial.println(raw);
    if((esp == '3') && flag == 1){
        if(raw < 200){
            night();
        }
    }
}

```

```

    else if(raw > 300){
        days();
    }
}
else if(esp == '4'){
    flag = 0;
}
}

void night(){
    data = digitalRead(dvijenie);
    if(data == 1){
        digitalWrite(led, LOW);
    }
    else if(data == 0){
        digitalWrite(led, HIGH);
    }
}

void days(){
    digitalWrite(led, HIGH);
    data = digitalRead(dvijenie);
    if(data == 0){
        digitalWrite(led, HIGH);
    }
}

```

Файл light.cpp - керування освітленням

```

//-----
#include <vcl.h>
#pragma hdrstop

#include "light.h"
#include "CPU_LPWANSocket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_light *Form_light;
//-----
__fastcall TForm_light::TForm_light(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

//ввімкнення ліхтаря біля входу в будинок з вказаною яскравістю
void __fastcall TForm_light::torch_onClick(TObject *Sender)
{
    torch->Caption="Ввімкнено";
    Image_torch_on->Visible=true;
    Image_torch_off->Visible=false;
    TrackBar_torch->Enabled=true;

    ClientSocket->Send_LPWANCommand(0,1,18,1); //0-код будинку; 1-код пристрою, що
керує ліхтарем; 18-код команди "on"; 1-кількість повторних надсилянь команди
ClientSocket->SendLeviton_LPWAN(0,1,TrackBar_torch->Position); //встановлення
яскравості, вказаної користувачем за допомогою TrackBar-у

}
//-----
//вимкнення ліхтаря біля входу в будинок
void __fastcall TForm_light::torch_offClick(TObject *Sender)
{
    torch->Caption="Вимкнено";
    Image_torch_off->Visible=true;
    Image_torch_on->Visible=false;
    TrackBar_torch->Enabled=false;
}

```

```
ClientSocket->Send_LPWANCommand(0,1,19,1); //0-код будинку; 1-код пристрою, що керує ліхтарем; 19-код команди "off"; 1-кількість повторних надсилань команди
```

```

}
//-----
//ввімкнення світла в коридорі з вказаною яскравістю
void __fastcall TForm_light::corridor_onClick(TObject *Sender)
{
corridor->Caption="Ввімкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
TrackBar_corridor->Enabled=true;

ClientSocket->Send_LPWANCommand(0,2,18,1);
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_corridor->Position);

}
//-----

//ввімкнення світла у вітальні з вказаною яскравістю
void __fastcall TForm_light::drawing_room_onClick(TObject *Sender)
{
drawing_room->Caption="Ввімкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
TrackBar_drawing_room->Enabled=true;

ClientSocket->Send_LPWANCommand(0,3,18,1);
ClientSocket->SendLeviton_LPWAN(0,3,TrackBar_drawing_room->Position);

}
//-----

//ввімкнення світла на кухні з вказаною яскравістю
void __fastcall TForm_light::kitchen_onClick(TObject *Sender)
{
kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
TrackBar_kitchen->Enabled=true;

ClientSocket->Send_LPWANCommand(0,4,18,1);
ClientSocket->SendLeviton_LPWAN(0,4,TrackBar_kitchen->Position);

}
//-----

//ввімкнення світла в кабінеті з вказаною яскравістю
void __fastcall TForm_light::cabinet_onClick(TObject *Sender)
{
cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
TrackBar_cabinet->Enabled=true;

ClientSocket->Send_LPWANCommand(0,5,18,1);
ClientSocket->SendLeviton_LPWAN(0,5,TrackBar_cabinet->Position);

}
//-----

//ввімкнення світла у спальні з вказаною яскравістю
void __fastcall TForm_light::bedroom_onClick(TObject *Sender)
{
bedroom->Caption="Ввімкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
TrackBar_bedroom->Enabled=true;

ClientSocket->Send_LPWANCommand(0,6,18,1);
ClientSocket->SendLeviton_LPWAN(0,6,TrackBar_bedroom->Position);

}

```

```

//-----
//ввімкнення світла в дитячій кімнаті з вказаною яскравістю
void __fastcall TForm_light::baby_room_onClick(TObject *Sender)
{
baby_room->Caption="Ввімкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
TrackBar_baby_room->Enabled=true;

ClientSocket->Send_LPWANCommand(0,7,18,1);
ClientSocket->SendLeviton_LPWAN(0,7,TrackBar_baby_room->Position);
}
//-----

//ввімкнення світла у ванній кімнаті з вказаною яскравістю
void __fastcall TForm_light::bathroom_onClick(TObject *Sender)
{
bathroom->Caption="Ввімкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
TrackBar_bathroom->Enabled=true;
ClientSocket->Send_LPWANCommand(0,8,18,1);
ClientSocket->SendLeviton_LPWAN(0,8,TrackBar_bathroom->Position);
}

//-----

//вимкнення світла в коридорі
void __fastcall TForm_light::corridor_offClick(TObject *Sender)
{
corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
TrackBar_corridor->Enabled=false;
ClientSocket->Send_LPWANCommand(0,2,19,1);
}
//-----

//вимкнення світла у вітальні
void __fastcall TForm_light::drawing_room_offClick(TObject *Sender)
{
drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
TrackBar_drawing_room->Enabled=false;
ClientSocket->Send_LPWANCommand(0,3,19,1);
}
//-----

//вимкнення світла на кухні
void __fastcall TForm_light::kitchen_offClick(TObject *Sender)
{
kitchen->Caption="Вимкнено";
Image_kitchen_off->Visible=true;
Image_kitchen_on->Visible=false;
TrackBar_kitchen->Enabled=false;
ClientSocket->Send_LPWANCommand(0,4,19,1);
}
//-----

//вимкнення світла в кабінеті
void __fastcall TForm_light::cabinet_offClick(TObject *Sender)
{
cabinet->Caption="Вимкнено";
Image_cabinet_off->Visible=true;
Image_cabinet_on->Visible=false;
TrackBar_cabinet->Enabled=false;
}

```

```

ClientSocket->Send_LPWANCommand(0,5,19,1);
}
//-----

//Вимкнення світла у спальні
void __fastcall TForm_light::bedroom_offClick(TObject *Sender)
{
bedroom->Caption="Вимкнено";
Image_bedroom_off->Visible=true;
Image_bedroom_on->Visible=false;
TrackBar_bedroom->Enabled=false;
ClientSocket->Send_LPWANCommand(0,6,19,1);
}
//-----

//Вимкнення світла у дитячій кімнаті
void __fastcall TForm_light::baby_room_offClick(TObject *Sender)
{
baby_room->Caption="Вимкнено";
Image_baby_room_off->Visible=true;
Image_baby_room_on->Visible=false;
TrackBar_baby_room->Enabled=false;
ClientSocket->Send_LPWANCommand(0,7,19,1);
}
//-----

//Вимкнення світла у ванній кімнаті
void __fastcall TForm_light::bathroom_offClick(TObject *Sender)
{
bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
TrackBar_bathroom->Enabled=false;
ClientSocket->Send_LPWANCommand(0,8,19,1);
}
//-----

//Ввімкнення світла скрізь
void __fastcall TForm_light::Button18Click(TObject *Sender)
{

torch->Caption="Ввімкнено";
Image_torch_on->Visible=true;
Image_torch_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,1,18,1);
ClientSocket->SendLeviton_LPWAN(0,1,TrackBar_torch->Position);

corridor->Caption="Ввімкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,2,18,1);
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_corridor->Position);

drawing_room->Caption="Ввімкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,3,18,1);
ClientSocket->SendLeviton_LPWAN(0,3,TrackBar_drawing_room->Position);

kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,4,18,1);
ClientSocket->SendLeviton_LPWAN(0,4,TrackBar_kitchen->Position);

cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,5,18,1);
}

```

```

ClientSocket->SendLeviton_LPWAN(0,5,TrackBar_cabinet->Position);

bedroom->Caption="Вимкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,6,18,1);
ClientSocket->SendLeviton_LPWAN(0,6,TrackBar_bedroom->Position);

baby_room->Caption="Вимкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,7,18,1);
ClientSocket->SendLeviton_LPWAN(0,7,TrackBar_baby_room->Position);

bathroom->Caption="Вимкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,8,18,1);
ClientSocket->SendLeviton_LPWAN(0,8,TrackBar_bathroom->Position);

TrackBar_torch->Enabled=true;
TrackBar_bedroom->Enabled=true;
TrackBar_corridor->Enabled=true;
TrackBar_drawing_room->Enabled=true;
TrackBar_kitchen->Enabled=true;
TrackBar_cabinet->Enabled=true;
TrackBar_baby_room->Enabled=true;
TrackBar_bathroom->Enabled=true;
}
//-----

//вимкнення світла скрізь
void __fastcall TForm_light::Button17Click(TObject *Sender)
{
torch->Caption="Вимкнено";
Image_torch_off->Visible=true;
Image_torch_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,1,19,1);

corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,2,19,1);

drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,3,19,1);

kitchen->Caption="Вимкнено";
Image_kitchen_off->Visible=true;
Image_kitchen_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,4,19,1);

cabinet->Caption="Вимкнено";
Image_cabinet_off->Visible=true;
Image_cabinet_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,5,19,1);

bedroom->Caption="Вимкнено";
Image_bedroom_off->Visible=true;
Image_bedroom_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,6,19,1);

baby_room->Caption="Вимкнено";
Image_baby_room_off->Visible=true;
Image_baby_room_on->Visible=false;

```

```
ClientSocket->Send_LPWANCommand(0,7,19,1);
```

```
bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,8,19,1);
```

```
TrackBar_torch->Enabled=false;
TrackBar_bedroom->Enabled=false;
TrackBar_corridor->Enabled=false;
TrackBar_drawing_room->Enabled=false;
TrackBar_kitchen->Enabled=false;
TrackBar_cabinet->Enabled=false;
TrackBar_baby_room->Enabled=false;
TrackBar_bathroom->Enabled=false;
}
```

```
//-----
```

```
//імітація присутності господарів
//ввімкнення та вимкнення світла випадковим чином
void __fastcall TForm_light::Button1Click(TObject *Sender)
{
```

```
if(Button1->Caption=="Імітація присутності")
{
Timer1->Enabled=true; //затуск таймеру, що запрограмований на імітацію
присутності
Button1->Caption=="Вимкнути імітацію"
}
else
{
Timer1->Enabled=false; //зупинтка таймеру
Button1->Caption=="Імітація присутності"
}
}
```

```
//-----
```

```
//таймер, запрограмований на імітацію присутності
void __fastcall TForm_light::Timer1Timer(TObject *Sender)
{
int x, y;
randomize();
x=random(6)+2; //генерація випадкового номера лампи в діапазоні 2-8
ClientSocket->Send_LPWANCommand(0,x,18,1);
y=random(6)+2;
ClientSocket->Send_LPWANCommand(0,y,19,1);
}
```

```
//-----
```

```
void __fastcall TForm_light::brightnessTimer(TObject *Sender)
{
//зміна яскравості
}
//-----
```

```
//зміна яскравості ліхтаря
void __fastcall TForm_light::TrackBar_torchChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,1,TrackBar_torch->Position);
}
//-----
```

```
//зміна яскравості освітлення коридору
void __fastcall TForm_light::TrackBar_corridorChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_corridor->Position);
}
```

```

//-----
//зміна яскравості освітлення вітальні
void __fastcall TForm_light::TrackBar_drawing_roomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_drawing_room->Position);
}
//-----

//зміна яскравості освітлення кухні
void __fastcall TForm_light::TrackBar_kitchenChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_kitchen->Position);
}
//-----

//зміна яскравості освітлення кабінету
void __fastcall TForm_light::TrackBar_cabinetChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_cabinet->Position);
}
//-----

//зміна яскравості освітлення спальні
void __fastcall TForm_light::TrackBar_bedroomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_bedroom->Position);
}
//-----

//зміна яскравості освітлення дитячої
void __fastcall TForm_light::TrackBar_baby_roomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_baby_room->Position);
}
//-----

//зміна яскравості освітлення ванної
void __fastcall TForm_light::TrackBar_bathroomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_bathroom->Position);
}
//-----

//визначення та виведення на екран поточного стану освітлення
void __fastcall TForm_light::StatusTimer(TObject *Sender)
{
ShortString st;

ClientSocket->Send_LPWANStatusQuery();

st=_LPWANState->GetStateOnOff(0, 1);
if(st=="On") {
torch->Caption="Ввимкнено";
Image_torch_on->Visible=true;
Image_torch_off->Visible=false;
TrackBar_torch->Enabled=true;
}
else {
torch->Caption="Вимкнено";
Image_torch_off->Visible=true;
Image_torch_on->Visible=false;
TrackBar_torch->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 2);
if(st=="On") {
corridor->Caption="Ввимкнено";
Image_corridor_on->Visible=true;
}
}

```

```
Image_corridor_off->Visible=false;
TrackBar_corridor->Enabled=true;
}
else {
corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
TrackBar_corridor->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 3);
if(st=="On") {
drawing_room->Caption="Ввимкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
TrackBar_drawing_room->Enabled=true;
}
else {
drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
TrackBar_drawing_room->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 4);
if(st=="On") {
kitchen->Caption="Ввимкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
TrackBar_kitchen->Enabled=true;
}
else {
kitchen->Caption="Вимкнено";
Image_kitchen_off->Visible=true;
Image_kitchen_on->Visible=false;
TrackBar_kitchen->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 5);
if(st=="On") {
cabinet->Caption="Ввимкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
TrackBar_cabinet->Enabled=true;
}
else {
cabinet->Caption="Вимкнено";
Image_cabinet_off->Visible=true;
Image_cabinet_on->Visible=false;
TrackBar_cabinet->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 6);
if(st=="On") {
bedroom->Caption="Ввимкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
TrackBar_bedroom->Enabled=true;
}
else {
bedroom->Caption="Вимкнено";
Image_bedroom_off->Visible=true;
Image_bedroom_on->Visible=false;
TrackBar_bedroom->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 7);
if(st=="On") {
baby_room->Caption="Ввимкнено";
```

```

Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
TrackBar_baby_room->Enabled=true;
}
else {
baby_room->Caption="Вимкнено";
Image_baby_room_off->Visible=true;
Image_baby_room_on->Visible=false;
TrackBar_baby_room->Enabled=false;
}
st=_LPWANState->GetStateOnOff(0, 8);
if(st=="On") {
bathroom->Caption="Ввимкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
TrackBar_bathroom->Enabled=true;
}
else {
bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
TrackBar_bathroom->Enabled=false;
}
}
}

```

Файл light.h - бібліотека для файлу light.cpp

```

#ifndef lightH
#define lightH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//-----
class TForm_light : public TForm
{
__published:      // IDE-managed Components
    TTrackBar *TrackBar_corridor;
    TLabel *Label11;
    TImage *Image_torch_on;
    TImage *Image_torch_off;
    TTrackBar *TrackBar_torch;
    TLabel *Label8;
    TImage *Image_corridor_on;
    TImage *Image_corridor_off;
    TTrackBar *TrackBar_drawing_room;
    TLabel *Label9;
    TImage *Image_drawing_room_on;
    TImage *Image_drawing_room_off;
    TTrackBar *TrackBar_cabinet;
    TLabel *Label12;
    TImage *Image_cabinet_on;
    TImage *Image_cabinet_off;
    TTrackBar *TrackBar_bedroom;
    TLabel *Label13;
    TImage *Image_bedroom_on;
    TImage *Image_bedroom_off;
    TTrackBar *TrackBar_baby_room;
    TLabel *Label14;
    TImage *Image_baby_room_on;
    TImage *Image_baby_room_off;
    TTrackBar *TrackBar_kitchen;
    TLabel *Label15;
    TImage *Image_kitchen_on;

```

```
TImage *Image_kitchen_off;
TTrackBar *TrackBar_bathroom;
TLabel *Label16;
TImage *Image_bathroom_on;
TImage *Image_bathroom_off;
TButton *Button17;
TButton *Button18;
TLabel *Label17;
TLabel *torch;
TLabel *Label19;
TLabel *corridor;
TLabel *Label21;
TLabel *drawing_room;
TLabel *Label23;
TLabel *kitchen;
TLabel *Label25;
TLabel *cabinet;
TLabel *Label27;
TLabel *bedroom;
TLabel *Label29;
TLabel *baby_room;
TLabel *Label31;
TLabel *bathroom;
TBevel *Bevel1;
TLabel *Label10;
TBevel *Bevel2;
TLabel *Label11;
TBevel *Bevel3;
TBevel *Bevel4;
TBevel *Bevel5;
TBevel *Bevel6;
TBevel *Bevel7;
TBevel *Bevel8;
TLabel *Label3;
TLabel *Label4;
TLabel *Label7;
TLabel *Label2;
TLabel *Label6;
TLabel *Label5;
TButton *torch_on;
TButton *torch_off;
TButton *corridor_on;
TButton *corridor_off;
TButton *drawing_room_on;
TButton *drawing_room_off;
TButton *kitchen_on;
TButton *kitchen_off;
TButton *cabinet_on;
TButton *cabinet_off;
TButton *bedroom_on;
TButton *bedroom_off;
TButton *baby_room_on;
TButton *baby_room_off;
TButton *bathroom_on;
TButton *bathroom_off;
TButton *Button1;
TTimer *Timer1;
TTimer *Status;
void __fastcall torch_onClick(TObject *Sender);
void __fastcall torch_offClick(TObject *Sender);
void __fastcall corridor_onClick(TObject *Sender);
void __fastcall drawing_room_onClick(TObject *Sender);
void __fastcall kitchen_onClick(TObject *Sender);
void __fastcall cabinet_onClick(TObject *Sender);
void __fastcall bedroom_onClick(TObject *Sender);
void __fastcall baby_room_onClick(TObject *Sender);
void __fastcall bathroom_onClick(TObject *Sender);
void __fastcall corridor_offClick(TObject *Sender);
void __fastcall drawing_room_offClick(TObject *Sender);
```

```

void __fastcall kitchen_offClick(TObject *Sender);
void __fastcall cabinet_offClick(TObject *Sender);
void __fastcall bedroom_offClick(TObject *Sender);
void __fastcall baby_room_offClick(TObject *Sender);
void __fastcall bathroom_offClick(TObject *Sender);
void __fastcall Button18Click(TObject *Sender);
void __fastcall Button17Click(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Timer1Timer(TObject *Sender);
void __fastcall brightnessTimer(TObject *Sender);
void __fastcall TrackBar_torchChange(TObject *Sender);
void __fastcall TrackBar_corridorChange(TObject *Sender);
void __fastcall TrackBar_drawing_roomChange(TObject *Sender);
void __fastcall TrackBar_kitchenChange(TObject *Sender);
void __fastcall TrackBar_cabinetChange(TObject *Sender);
void __fastcall TrackBar_bedroomChange(TObject *Sender);
void __fastcall TrackBar_baby_roomChange(TObject *Sender);
void __fastcall TrackBar_bathroomChange(TObject *Sender);
void __fastcall StatusTimer(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm_light(TComponent* Owner);
};
//-----
extern PACKAGE TForm_light *Form_light;
//-----
#endif

```

Файл climate.cpp - керування кліматом

```

#include <vcl.h>
#pragma hdrstop

#include "climate.h"
#include "CPU_LPWANSocket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_climate *Form_climate;
//-----
__fastcall TForm_climate::TForm_climate(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm_climate::Button3Click(TObject *Sender)
{
    ClientSocket->Send_LPWANCommand(0,21,18,1);
}
//-----
void __fastcall TForm_climate::Button4Click(TObject *Sender)
{
    ClientSocket->Send_LPWANCommand(0,21,19,1);
}
//-----

void __fastcall TForm_climate::Button1Click(TObject *Sender)
{
    ClientSocket->Send_LPWANCommand(0,22,18,1);
}
//-----

void __fastcall TForm_climate::Button2Click(TObject *Sender)
{
    ClientSocket->Send_LPWANCommand(0,22,19,1);
}

```

```
}
//-----

void __fastcall TForm_climate::Button5Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,23,18,1);
}
//-----

void __fastcall TForm_climate::Button6Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,23,19,1);
}
//-----

void __fastcall TForm_climate::Button7Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,24,18,1);
}
//-----

void __fastcall TForm_climate::Button8Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,24,19,1);
}
//-----

void __fastcall TForm_climate::Button9Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,26,18,1);
}
//-----

void __fastcall TForm_climate::Button10Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,26,19,1);
}
//-----

void __fastcall TForm_climate::Button11Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,27,18,1);
}
//-----

void __fastcall TForm_climate::Button12Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,27,19,1);
}
//-----

void __fastcall TForm_climate::Button13Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,28,18,1);
}
//-----

void __fastcall TForm_climate::Button14Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,28,19,1);
}
//-----

void __fastcall TForm_climate::Button15Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,29,18,1);
}
//-----
```

```

void __fastcall TForm_climate::Button16Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,29,19,1);
}
//-----

void __fastcall TForm_climate::TrackBar1Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,21,TrackBar1->Position);
}
//-----

void __fastcall TForm_climate::TrackBar2Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,22,TrackBar2->Position);
}
//-----

void __fastcall TForm_climate::TrackBar3Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,23,TrackBar3->Position);
}
//-----

void __fastcall TForm_climate::TrackBar4Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,24,TrackBar4->Position);
}
//-----

void __fastcall TForm_climate::TrackBar5Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,24,TrackBar5->Position);
}
//-----

//визначення та виведення на екран поточного стану клімат-контролю

void __fastcall TForm_climate::StatusTimer(TObject *Sender)
{
ShortString st;
int n;

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 21);
if(st=="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 22);
if(st=="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 23);
if(st=="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 24);
if(st=="On") Label_4->Caption="Ввімкнено"; else Label_4->Caption="Вимкнено";

n=GetVariable(25);
Label_t->Caption=n;
n=GetVariable(34);
Label_v->Caption=n;

n=GetVariable(22);
Label_tp1->Caption=n;
n=GetVariable(23);
Label_tp2->Caption=n;

st=_LPWANState->GetStateOnOff(0, 26);
if(st=="On") Label_5->Caption="Відкрито"; else Label_5->Caption="Закрито";
st=_LPWANState->GetStateOnOff(0, 27);
if(st=="On") Label_6->Caption="Відкрито"; else Label_6->Caption="Закрито";

```

```

st=_LPWANState->GetStateOnOff(0, 28);
if(st=="On") Label_7->Caption="Відкрито"; else Label_7->Caption="Закрито";
st=_LPWANState->GetStateOnOff(0, 29);
if(st=="On") Label_8->Caption="Відкрито"; else Label_8->Caption="Закрито";
}

```

Файл climate.h - бібліотека для файлу climate.cpp

```

#ifndef climateH
#define climateH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//-----
class TForm_climate : public TForm
{
__published:      // IDE-managed Components
    TGroupBox *GroupBox2;
    TLabel *Label3;
    TLabel *Label_1;
    TButton *Button3;
    TButton *Button4;
    TTrackBar *TrackBar1;
    TLabel *Label5;
    TLabel *Label6;
    TLabel *Label8;
    TLabel *Label9;
    TLabel *Label10;
    TLabel *Label11;
    TLabel *Label12;
    TLabel *Label13;
    TLabel *Label7;
    TGroupBox *GroupBox1;
    TLabel *Label11;
    TLabel *Label_2;
    TLabel *Label14;
    TLabel *Label15;
    TLabel *Label16;
    TLabel *Label17;
    TLabel *Label18;
    TLabel *Label19;
    TLabel *Label20;
    TLabel *Label21;
    TLabel *Label22;
    TLabel *Label27;
    TLabel *Label_tpl1;
    TLabel *Label29;
    TLabel *Label30;
    TBevel *Bevel2;
    TButton *Button1;
    TButton *Button2;
    TTrackBar *TrackBar2;
    TGroupBox *GroupBox3;
    TLabel *Label31;
    TLabel *Label_3;
    TLabel *Label33;
    TLabel *Label34;
    TLabel *Label35;
    TLabel *Label36;
    TLabel *Label37;
    TLabel *Label38;
    TLabel *Label39;

```

```
TLabel *Label140;
TLabel *Label141;
TLabel *Label142;
TLabel *Label_tp2;
TLabel *Label144;
TLabel *Label145;
TBevel *Bevel3;
TButton *Button5;
TButton *Button6;
TTrackBar *TrackBar3;
TGroupBox *GroupBox4;
TLabel *Label146;
TLabel *Label_4;
TLabel *Label148;
TLabel *Label149;
TButton *Button7;
TButton *Button8;
TGroupBox *GroupBox5;
TLabel *Label157;
TLabel *Label_5;
TButton *Button9;
TButton *Button10;
TGroupBox *GroupBox6;
TLabel *Label159;
TLabel *Label_6;
TButton *Button11;
TButton *Button12;
TGroupBox *GroupBox7;
TLabel *Label161;
TLabel *Label_7;
TButton *Button13;
TButton *Button14;
TGroupBox *GroupBox8;
TLabel *Label163;
TLabel *Label_8;
TButton *Button15;
TButton *Button16;
TGroupBox *GroupBox9;
TLabel *Label167;
TLabel *Label168;
TLabel *Label_t;
TLabel *Label170;
TLabel *Label171;
TLabel *Label_v;
TImage *Image8;
TImage *Image7;
TTrackBar *TrackBar4;
TLabel *Label166;
TLabel *Label173;
TLabel *Label174;
TLabel *Label175;
TLabel *Label176;
TLabel *Label177;
TLabel *Label178;
TLabel *Label179;
TLabel *Label165;
TLabel *Label23;
TTrackBar *TrackBar5;
TLabel *Label124;
TLabel *Label125;
TLabel *Label126;
TLabel *Label150;
TLabel *Label151;
TLabel *Label152;
TLabel *Label153;
TLabel *Label154;
TLabel *Label155;
TLabel *Label156;
TLabel *Label180;
```

```

TLabel *Label181;
TButton *Button17;
TLabel *Label182;
TLabel *Label183;
TEdit *Edit1;
TLabel *Label184;
TEdit *Edit2;
TLabel *Label185;
TTimer *Status;
void __fastcall Button3Click(TObject *Sender);
void __fastcall Button4Click(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button2Click(TObject *Sender);
void __fastcall Button5Click(TObject *Sender);
void __fastcall Button6Click(TObject *Sender);
void __fastcall Button7Click(TObject *Sender);
void __fastcall Button8Click(TObject *Sender);
void __fastcall Button9Click(TObject *Sender);
void __fastcall Button10Click(TObject *Sender);
void __fastcall Button11Click(TObject *Sender);
void __fastcall Button12Click(TObject *Sender);
void __fastcall Button13Click(TObject *Sender);
void __fastcall Button14Click(TObject *Sender);
void __fastcall Button15Click(TObject *Sender);
void __fastcall Button16Click(TObject *Sender);
void __fastcall TrackBar1Change(TObject *Sender);
void __fastcall TrackBar2Change(TObject *Sender);
void __fastcall TrackBar3Change(TObject *Sender);
void __fastcall TrackBar4Change(TObject *Sender);
void __fastcall TrackBar5Change(TObject *Sender);
void __fastcall StatusTimer(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm_climate(TComponent* Owner);
};
//-----
extern PACKAGE TForm_climate *Form_climate;
//-----
#endif

```

Файл water.cpp - керування системою водопостачання

```

#include <vcl.h>
#pragma hdrstop

#include "water.h"
#include "CPU_LPWANSocket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_water *Form_water;
//-----
__fastcall TForm_water::TForm_water(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
//ввімкнення системи контролю протікання води у ванній
void __fastcall TForm_water::Button2Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,9,18,1);
}
//-----

//відкриття клапану холодної води у ванні
void __fastcall TForm_water::Button6Click(TObject *Sender)
{

```

```

ClientSocket->Send_LPWANCommand(0,10,18,1);
}
//-----

//відкриття клапану гарячої води у ванні
void __fastcall TForm_water::Button8Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,11,18,1);
}
//-----
//ввімкнення системи контролю протікання води на кухні
void __fastcall TForm_water::Button4Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,12,18,1);
}
//-----
//відкриття клапану холодної води на кухні
void __fastcall TForm_water::Button10Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,13,18,1);
}
//-----

//відкриття клапану гарячої води на кухні
void __fastcall TForm_water::Button12Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,14,18,1);
}
//-----

//вимкнення системи контролю протікання води у ванній
void __fastcall TForm_water::Button1Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,9,19,1);
}
//-----
//закриття клапану холодної води у ванні
void __fastcall TForm_water::Button5Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,10,19,1);
}
//-----
//закриття клапану гарячої води у ванні
void __fastcall TForm_water::Button7Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,11,19,1);
}
//-----

//вимкнення системи контролю протікання води на кухні
void __fastcall TForm_water::Button3Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,12,19,1);
}
//-----
//закриття клапану холодної води на кухні
void __fastcall TForm_water::Button9Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,13,19,1);
}
//-----

//закриття клапану гарячої води на кухні
void __fastcall TForm_water::Button11Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,14,19,1);
}
//визначення та виведення на екран поточного стану системи водопостачання

```

```

void __fastcall TForm_water::StatusTimer(TObject *Sender)
{
ShortString st;
int n;

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 9);
if(st="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 12);
if(st="On") Label_4->Caption="Ввімкнено"; else Label_4->Caption="Вимкнено";

n=GetVariable(9);
if(n="0") Label_11->Caption="В нормі"; else Label_11->Caption="Протікання води";
n=GetVariable(12);
if(n="0") Label_44->Caption="В нормі"; else Label_44->Caption="Протікання води";

st=_LPWANState->GetStateOnOff(0, 10);
if(st="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 11);
if(st="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";

st=_LPWANState->GetStateOnOff(0, 13);
if(st="On") Label_5->Caption="Ввімкнено"; else Label_5->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 14);
if(st="On") Label_6->Caption="Ввімкнено"; else Label_6->Caption="Вимкнено";
}

```

Файл water.h - бібліотека для файлу water.cpp

```

#ifndef waterH
#define waterH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
//-----
class TForm_water : public TForm
{
__published: // IDE-managed Components
    TGroupBox *GroupBox1;
    TButton *Button1;
    TButton *Button2;
    TLabel *Label1;
    TLabel *Label_1;
    TGroupBox *GroupBox2;
    TLabel *Label3;
    TLabel *Label_4;
    TButton *Button3;
    TButton *Button4;
    TGroupBox *GroupBox3;
    TLabel *Label5;
    TLabel *Label_2;
    TButton *Button5;
    TButton *Button6;
    TGroupBox *GroupBox4;
    TLabel *Label7;
    TLabel *Label_3;
    TButton *Button7;
    TButton *Button8;
    TGroupBox *GroupBox5;
    TLabel *Label9;
    TLabel *Label_5;
    TButton *Button9;
    TButton *Button10;

```

```
TGroupBox *GroupBox6;
TLabel *Label11;
TLabel *Label_6;
TButton *Button11;
TButton *Button12;
TLabel *Label13;
TLabel *Label_11;
TLabel *Label15;
TLabel *Label_44;
TTimer *Status;
void __fastcall Button2Click(TObject *Sender);
void __fastcall Button6Click(TObject *Sender);
void __fastcall Button8Click(TObject *Sender);
void __fastcall Button4Click(TObject *Sender);
void __fastcall Button10Click(TObject *Sender);
void __fastcall Button12Click(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button5Click(TObject *Sender);
void __fastcall Button7Click(TObject *Sender);
void __fastcall Button3Click(TObject *Sender);
void __fastcall Button9Click(TObject *Sender);
void __fastcall Button11Click(TObject *Sender);
void __fastcall StatusTimer(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm_water(TComponent* Owner);
};
//-----
extern PACKAGE TForm_water *Form_water;
//-----
#endif
```