

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

До захисту допущено:

«На правах рукопису»

Завідувач кафедри _____ Ігор АНІСІМОВ

« __ » червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

**«СИСТЕМИ РОЗУМНОГО БУДИНКУ. РОЗУМНА РОЗЕТКА НА БАЗІ
МІКРОКОНТРОЛЛЕРУ STM32»**

Виконав:

студент 4-го курсу

денної форми навчання

спеціальності 172 - Телекомунікації та радіотехніка

ОП «Інформаційна безпека телекомунікаційних систем і мереж»

Кривохижа Дмитро Артурович _____

Науковий керівник:

к.т.н., с.н.с. Гахович Сергій Вікторович _____

Рецензент:

к.т.н., доц. Савченко Тетяна Віталіївна _____

Засвідчую, що у цій бакалаврській роботі
немає запозичень з праць інших авторів без
відповідних посилань

Студент _____

Робота допущена до захисту в ЕК рішенням кафедри радіотехніки та радіоелектронних систем від «23» червня 2023 р., протокол № 22.

Завідувач кафедри радіотехніки та радіоелектронних систем,

доктор фіз.-мат. наук, професор

Анісімов Ігор Олексійович _____

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	3
ВСТУП	4
РОЗДІЛ 1. АКТУАЛЬНІ РІШЕННЯ СИСТЕМ «SMART HOME».....	5
1.1. Опис розвитку систем	5
1.2. Можливості та ідейний задум систем «Smart Home».....	6
1.3. Аналіз наявних систем «Smart Home».....	7
1.4. Розетки з функцією дистанційного керування.....	10
1.4.1 Wi-fi розетки з функцією дистанційного керування	12
1.4.2. GSM розетки з функцією дистанційного керування	14
РОЗДІЛ 2. РЕАЛІЗАЦІЯ АПАРАТНОЇ ЧАСТИНИ ПРОЕКТУ	17
2.1. Вибір основних компонентів схеми	17
2.2. Опис архітектури мікроконтролера STM32	18
2.3. Опис архітектури модулю ESP8266-01S	19
2.4. Принципова і структурна схема пристрою керування	21
2.4.1 Опис процесу побудови прототипу схеми пристрою керування	24
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ЧАСТИНИ ПРОЕКТУ	26
3.1 Основні компоненти програмної частини.....	26
3.2. Програмний код для STM32F410	26
3.3. Програмний код для ESP8266-S01	28
3.4. Програмний код для web-застосунку	30
ВИСНОВКИ	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	33
ДОДАТОК 1. КОД КОНФІГУРАЦІЇ USART	34
ДОДАТОК 2. КОД HTML-СТОРІНКИ ДЛЯ ВЕБ-ЗАСТОСУНКУ ..	37

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- ARM (Advanced RISC Machines) - архітектура процесора з простим набором команд, яка забезпечує високу продуктивність.
- UART (Universal Asynchronous Receiver/Transmitter) - універсальний асинхронний приймач/передавач.
- USART (Universal Synchronous/Asynchronous Receiver/Transmitter) - універсальний синхронний/асинхронний приймач/передавач.
- SPI (Serial Peripheral Interface) - послідовний периферійний інтерфейс, протокол зв'язку.
- I2C (Inter-Integrated Circuit) - міжінтегральна шина, протокол зв'язку,
- ADC (Analog-to-Digital Converter) - аналого-цифровий перетворювач.
- DAC (Digital-to-Analog Converter) - цифро-аналоговий перетворювач.
- GPIO (General Purpose Input/Output) - введення/виведення загального призначення.
- TCP (Transmission Control Protocol) - протокол керування передачею.
- МК – скорочення для «мікроконтролер».

ВСТУП

В повсякденному житті сучасної людини невід'ємною складовою є побутова техніка, гаджети, прилади освітлення тощо. Повертаючись додому ми вимикаємо систему охорони, вмикаємо світло, проходимо до ванної кімнати де миємо руки теплою водою що була нагріта бойлером. Наступними вмикаються електро-чайник, кондиціонер/обігрівач а слідом за ними, звичайно, йде телевізор. Кількість приладів що кожного дня оточують користувачів незворотно росте, та подекуди комунікувати з умовним «десятком» пристроїв стає складно. На допомогу в цьому питанні прийшли системи «розумного будинку» що направлені на облегшення комунікації зі складовими повсякдення сучасної людини.

Системи «Smart Home» це найсучасніші засоби автоматизації, що перетворюють наш дім на більш функціональне та комфортне місце для перебування . На ринку технологій представлено велику кількість готових екосистем «Smart Home», які наявні в каталогах більшості технологічних компаній сьогодення.

Насамперед системи «Розумний будинок» забезпечують організацію безпеки житла, створення комфортного мікроклімату, спрощують керування відеоспостереженням та мультимедійними пристроями. Більшість таких систем спрямовані на поліпшення енергоефективності будинку, а деякі мають додаткові «розумні» функції, які полегшують повсякденні обов'язки в домашньому господарстві.

Завдяки можливості віддаленого керування та контролю над приладами, людина має можливість налаштувати систему індивідуально. Крім того, система надає можливість відстежувати статистичні дані про енерговитрати та оптимізувати екологічність будинку, незалежно від місця перебування користувача.

РОЗДІЛ 1. АКТУАЛЬНІ РІШЕННЯ СИСТЕМ «SMART HOME»

1.1. Опис розвитку систем

Початкові кроки у розробці подібних систем можна віднести до кінця XIX століття де поява радіо, електричного освітлення та телефонних технологій назавжди змінила світ. Восени 1898 року винахідник Нікола Тесла вже плідно працював над системою дистанційного керування суднами та іншим транспортом, що можна вважати одним з перших проривів у розробці засобів дистанційного контролю [1]. Електричні побутові прилади набули широкої відомості у 30-х роках XX століття, незважаючи на великі ціни, та згодом перетворилися на невід'ємну частину повсякденного життя, замінивши великий об'єм ручної праці.

Восени 1934 року на Нью-Йоркській виставці були запропоновані ідеї, що наближалися до концепцій автоматизації сьогодення. В 1960-х роках у відомому французькому парку розваг «Діснейленд» було представлено цікавий концепт під назвою "Будинок майбутнього", який було побудовано об'єднаними силами розробників з компанії Disney, Monsanto та технологічним інститутом Масачусетсу. Навіть на той час в будинку вже були наявні функціонуючі автоматичні жалюзі, побутове приладдя з змінним рівнем висоти та велика кількість інших цікавих ідей. Важливим етапом прогресу систем "Smart Home" є виникнення автоматизації побуту в 1976 році, коли шведська компанія Pico Electronics почала використовувати її для керування мультимедійними програвачами. Вперше у історії термін «Smart Home» застосувала американська корпорація «House Builders» яка стверджувала, що такі будинки, на відміну від звичайних, є більш продуктивними і забезпечують оптимальне використання робочого та житлового простору [2, 3].

З появою автоматизованих систем поступово з'явилися можливості застосовувати їх при створенні будівельних споруд. Спочатку вони були поширені в офісних та промислових приміщеннях, а згодом такі системи стали використовуватися й у приватних житлових будинках. Розвиток мікроконтролерів призвів до зниження вартості електронних пристроїв, що сприяло їх популярності та розмаїттю, що в свою чергу вплинуло на швидкість поширення автоматизованих систем у життя звичайних людей з метою поліпшення якості життя та розширення можливостей контролю над своєю домівкою.

1.2. Можливості та ідейний задум систем «Smart Home»

Створення сучасних систем "Розумний дім" поєднує інженерні рішення, що використовуються під час будівництва приміщень, з використанням інформаційних засобів керування та контролю в просторі будівель. Основною метою такої системи є контроль та поліпшення умов проживання, підвищення функціональності, комфорту та енергоефективності приміщень. Ці можливості системи "Розумний дім" досягаються за допомогою створення відокремленої екосистеми з автоматизованими датчиками та пристроями, які керуються за допомогою передових бездротових технологій. "Розумний дім" представляє собою систему інтелектуальної автоматики для управління інженерними системами сучасної будівлі, що може задовольнити всі потреби та бажання власника будинку, включаючи регулювання температури, вологості, освітлення, аудіосистеми, забезпечення безпеки та багато іншого. Більшість систем "Розумний дім" включають наступні комплекси та об'єкти автоматизації:

- Контроль клімату – налаштовує температуру та вологість у будівлі, забезпечує комфортний мікроклімат;
- Контроль над мультимедійними комплексами – зручне керування

- бібліотекою музики або фільмів;
- Налаштування освітлення (лампи, LED стрічки, жалюзі, зовнішнє освітлення) – керує освітленням вдома та на дворі, контролює рівень світла та оптимізує енерговитрати;
- Контроль та налаштування енергозбереження та електроприладів (запланований час роботи розеток) – оптимізація роботи електроприладів та комфорт в управлінні;
- Комплекси камер стеження та систем безпеки – налаштовує роботу систем камер та датчиків і сприяє зручному доступу до інформації.
- Протипожежна/охоронна сигналізація;

1.3. Аналіз наявних систем «Smart Home»

Clipsal Smart Home - це інноваційна система розумного будинку, розроблена брендом Clipsal (Рис 1.3.1). Вона складається з невеликих сенсорних модулів, які можуть бути окремо підключені до електроприладів вашого будинку. Ця система забезпечує зручне керування живленням пристроїв завдяки будованому алгоритму.

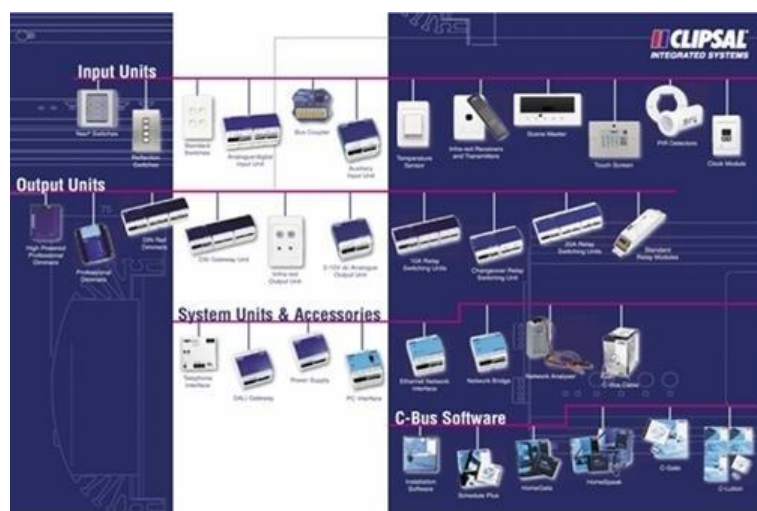


Рис.1.3.1. Рішення від Clipsal

Однією з головних переваг системи Clipsal Smart Home є її низька вартість, що робить її доступною для широкого кола користувачів. Крім того, система є масштабованою, що означає, що ви можете додавати нові модулі і розширювати функціонал системи залежно від ваших потреб. Це рішення забезпечує взаємодію з технологією різних брендів. Це означає, що ви можете інтегрувати компоненти різних виробників в одну цілісну систему управління [4].

Проте, важливо відзначити, що процес налаштування системи може бути дещо складним, і вимагає певного рівня технічних знань.

Allone WiFi Smart Remote Control - це передова система "розумного дому" від виробника Allone, що надає безліч можливостей для контролю та автоматизації будинку (Рис 1.3.2). За допомогою центрального контролера і з'єднання Wi-Fi зручно керувати всіма своїми приладами, незалежно від місця перебування.



Рис.1.3.2. Allone WiFi Smart Remote Control

Основною перевагою цієї системи є широкий спектр функціональних модулів, що дозволяють налаштувати її під ваші унікальні потреби. Ви зможете включати та вимикати світло, регулювати температуру, керувати

розетками та багато іншого, все це зручно керуючи системою зі смартфона або планшета [5].

Крім того, система Allone WiFi Smart Remote Control може бути легко розширена. Є можливість додати додаткові модулі та пристрої для подальшого розширення функціональності "розумного дому". Це дозволяє налаштовувати систему поетапно та відповідно до змінюваних потреб.

Однак, варто враховувати, що ціна модулів може бути високою, що обмежує їх доступність для користувачів. Також слід відзначити, що Wi-Fi-мережа має обмежений радіус дії, тому впевніться, що система буде працювати надійно в межах вашого будинку.

Варіант системи «Smart Home» від Meizu - це ціла збірка з високо-функціональних «Smart» пристроїв, що було об'єднано у єдиний керуючий софт під назвою LifeKit (Рис 1.3.3). Список наявних приладів цієї системи:

- Ваги RyFit
- Лампа X-Light Plus
- Дистанційні розетки,
- Очищувач повітря Air Cube та ін.



Рис.1.3.3. Варіант системи «Smart Home» від Meizu

Система працює та керується на базі центрального керуючого пристрою – хабу.

Серед переваг можна відзначити наявність готових до використання самостійних пристроїв, легкість монтажу і налаштування, а також можливість керування за допомогою смартфона. Однак, є кілька недоліків, таких як обмежений функціонал, обмежений вибір устаткування наразі і прив'язка до пристроїв від Meizu.

1.4. Розетки з функцією дистанційного керування

Використання смарт-розеток додає зручності в повсякденне життя користувача, можливість керувати пристроями віддалено створює велику кількість сценаріїв використання. Крім зручності, смарт-розетки пропонують функції безпеки. Ви можете встановити графік роботи світла або інших пристроїв, щоб створити враження, що ви перебуваєте вдома, навіть коли відсутні. Це може допомогти в запобіганні крадіжкам або просто створити комфортну атмосферу в вашому будинку.

Ринок Wi-Fi розеток пропонує різноманітні моделі і бренди з різними функціями та можливостями. Рейтинг популярних брендів Wi-Fi розеток очолюють компанії TP-Link, Belkin, Amazon, Google і Xiaomi. Вони можуть бути доступні в різних конфігураціях, наприклад з USB-портами або навіть з певними сенсорами. Обираючи між опціями «придбати готовий пристрій» або «зібрати пристрій самостійно» варто спиратися на переваги та недоліки обох. До головних переваг готових пристроїв можна віднести такі пункти як:

- Зручність в установці, налаштуванні та заощадження часу. Придбання готових пристроїв заощаджує ваш час, так як зазвичай фабричні пристрої мають простий процес налаштування та інструменти для

швидкого підключення до мережі.

- Гарантія та підтримка. Готові пристрої зазвичай поставляються з гарантією від виробника. Це означає, що якщо виникають проблеми з пристроєм, ви можете звернутися до виробника або продавця для отримання підтримки та ремонту. Це надає додаткову впевненість у якості та функціональності продукту.
- Вимоги безпеки. Фабричні пристрої що представлені на ринку часто проходять випробовування безпеки, що забезпечує їх відповідність до певних сталих вимог.

Хоча покупка готового пристрою має серйозні переваги, є деякі потенційні недоліки, які необхідно враховувати:

- Вартість. Готові пристрої, в особливості ті що продаються під лейблом відомого бренду, значно дорожчі в порівнянні з саморобними аналогами. Якщо користувач має обмежений бюджет або необхідна велика кількість пристроїв це може стати вирішальним фактором.
- Обмежена сумісність. На відміну від саморобних аналогів які зазвичай працюють від одного кореневого хабу, готові пристрої можуть мати дуже обмежену сумісність з іншими пристроями навіть одного виробника, не кажучи вже про сумісність с іншими системами на ринку.
- Залежність від компанії-виробника. Дуже часто великі корпорації намагаються створити власну закриту «еко-систему» з пристроїв що ними виробляються. З цього виходить що покупаючи елементи певної смарт системи (наприклад Apple Home) користувач обмежений лише товарами цього виробника, що зачасту не має великої різноманітності.

1.4.1 Wi-fi розетки з функцією дистанційного керування.

Розглянемо модель смарт-розетки від виробника Zeoota, модель PS022 (Рис 1.4.1). Модель смарт-розетки Zeoota PS022 створена для дистанційного

керування побутовими приладами. На пристрій подаються команди за допомогою смартфона, незалежно від місця перебування користувач має можливість вмикати та вимикати електротехніку у власному будинку або на роботі.

Безпосереднє керування здійснюється з додатку для «Smart Life», за умови підключення його до мережі wi-fi [6].



Рис.1.4.1. Рішення від виробника Zeoota, модель - PS022.

Також наявна функція таймеру вимкнення конкретного слоту подовжувача, проте варто враховувати що керування розкладом роботи пристроїв доки реалізовано невдало, та не завжди працює так як повинно. Пристрій можна віднести до середньоцінового сегменту, вартість цієї моделі складає 2550 UAH станом на 2023 рік.

Розглянемо бюджетний варіант подібної Wi-fi розетки:

Модель розумної Wi-fi розетки Smart Plug 16A від виробника Tuuya (рис. 1.4.2)



Рис. 1.4.2. Рішення від виробника Tuuya, модель - Smart Plug 16A

Хоч ця розумна розетка відноситься до бюджетного сегменту, вона надає більшість властивостей та функцій попереднього пристрою PS022. Наприклад: функції дистанційного ввімкнення та вимкнення, таймеру та підрахування енергоспоживання, при вартості лише в 270 UAH станом на 2023 рік. Проте, має лише один сокет для підключення. Tuuya Smart Plug також керується за допомогою додатку для смартфона «Tuuya Smart Life». Єдиним, але вагомим недоліком є той факт що для керування пристроєм смартфон та сам пристрій повинні бути приєднані до однієї wi-fi мережі. Це унеможливує керування розеткою за межами оселі. Проте, з різницею у вартості майже у 10 разів можна вважати такі недоліки незначними.

1.4.2. GSM розетки з функцією дистанційного керування.

Окремо від розеток керованих за допомоги технології WiFi на ринку існують варіанти пристроїв керованих за системою GSM. Такий варіант виконання смарт розетки дозволяє користувачу керувати електроприладами через мобільний зв'язок GSM (Global System for Mobile Communications), зберігаючи усі властивості пристроїв що керуються через wi-fi мережу.

Користувач може підключити свої побутові прилади до розумної розетки, яка має вбудований GSM-модуль, що дозволить віддалено керувати електроприладами з будь-якого місця, використовуючи мобільний телефон або інший зв'язковий пристрій.

GSM розетки мають деякі переваги перед Wi-Fi розетками:

- Незалежність від Wi-Fi. Використовування GSM розетки можливе навіть у випадках, коли немає доступу до Інтернету або коли Wi-Fi мережа нестабільна. Це робить їх більш надійними у складних ситуаціях.
- Широкий спектр покриття. GSM розетки працюють на основі мобільних мереж, таких як 2G, 3G або 4G. Це означає, що вони можуть працювати в широкому радіусі, якщо є покриття мобільного оператора. Керування розумними розетками з будь-якого місця, де є сигнал мобільного зв'язку, є значною перевагою перед аналогічними пристроями керованими за допомоги wi-fi.

Розглянемо наявні приклади реалізації пристрою:

Розглянемо модель розумної GSM розетки від українського бренду Ajax моделі «Socket» (Рис 1.4.3.), що була представлена у лютому 2022 року.



Рис. 1.4.3. Розумна розетка Ajax Socket

Важливо розуміти що більшість функцій у будь яких смарт-розетках збігаються. В цій моделі, як і у розглянутих раніше реалізовано дистанційне керування, налаштування розкладу роботи та підтримка різних сценаріїв роботи. Розглянемо основні відмінності цього пристрою від його аналогів. Ajax Socket під'єднується та керується за допомогою центрального хабу Ajax за радіопротоколом Jeweller. Працює на відстані до 1 км від хабу (незалежно від сигналу Wi-Fi в будівлі) та забезпечує безперебійну роботу в умовах відсутності інтернет з'єднання [7]. Також:

- У разі стрибків напруги та струму подача живлення на пристрій припиняється.
- Надає повідомлення про перебої в роботі за допомогою смс і пуш-повідомлень на пристрої.

Цей варіант має вартість 1730 UAH станом на 2023 рік. Відноситься до преміального сегменту.

Розглянемо бюджетний варіант подібного пристрою:

Бюджетне рішення подібне до попередньої розумної розетки пропонує виробник Tuya, модель – Zenismart GSM 16A (Рис. 1.4.4).



Рис. 1.4.4. Розумна розетка Tuya Zenismart GSM 16A

В Zenismart загалом збережені усі можливості та функції інших варіацій смарт-розеток серед яких: дистанційне ввімкнення/вимкнення, таймер роботи та налаштування декількох сценаріїв використання. Окрім вартості у 500 UAH станом на 2023 рік, особливою перевагою є виконання корпусу з якісного вогнестійкого матеріалу, сумісність з великою кількістю інших систем та керування пристроєм за допомоги голосових помічників Google, Amazon Alexa, Apple Siri та інших.

Підсумовуючи, вибір бажаної смарт-розетки залежить лише від індивідуальних потреб користувача та бюджету на покупку. Відмінності між варіаціями пристроїв незначні, що надає гнучкості у пошуку та звільняє від довгих роздумів між різними пропозиціями.

Раніше були розглянуті переваги та недоліки готових виробів та саморобних аналогів. До наданого списку також можна додати: у випадку коли виробник припиняє підтримку пристрою або не надає достатню підтримку, пристрій не здатен отримувати оновлення та виправлення помилок, що негативно впливає на досвід користування. Тому самостійне виготовлення пристрою є цікавою альтернативою.

РОЗДІЛ 2. РЕАЛІЗАЦІЯ АПАРАТНОЇ ЧАСТИНИ ПРОЕКТУ

Метою роботи є побудова прототипу приладу керування розеткою за допомогою Wi-Fi. Задля досягнення мети роботи необхідно підібрати підходящі елементи для побудови схеми. Головним елементом схеми буде мікроконтролер що обробляє дані з пристрою користувача та керує ключем, що в свою чергу керує подачею струму на розетку. Задля забезпечення зв'язку МК з пристроєм користувача необхідно використати модуль Wi-Fi зв'язку.

Також дуже важливо дотримуватись бюджетності при виборі необхідних компонентів схеми. Саме низька вартість пристрою у підсумку, можливість майбутньої модернізації та відносна простота реалізації є основними перевагами саморобного аналогу перед пристроями що вже наявні на ринку технологій.

2.1. Вибір основних компонентів схеми

Вибір необхідних компонентів для розумної розетки є ключовим етапом проектування, який визначає якість, функціональність та надійність системи. Для досягнення успішних результатів у розробці розумної розетки на базі мікроконтролера STM32, необхідно розглянути та оцінити характеристики та вимоги до компонентів, що використовуються. Необхідно обрати підходящу модель мікроконтролеру STM32. Враховуючи вимоги до функціональності, такі як здатність до зв'язку зі зовнішніми пристроями, вбудовані аналого-цифрові та цифро-аналогові перетворювачі, наявність пам'яті та інтерфейсів. Схема пристрою буде керуватися за допомогою Wi-Fi модулю. Працюючи з пристроями стандарту Wi-Fi бажано розуміти принципи його роботи та обирати необхідні компоненти з точки зору можливостей взаємодії з іншими елементами схеми та індивідуальними особливостями проекту.

Для реалізації проекту було обрано мікроконтролер модельного ряду STM32 – F410RB. Він забезпечить достатню кількість вбудованої флеш-пам'яті та відносно велику швидкість тактування процесора. Вартість такого МК складає 508 UAH. Задля забезпечення зв'язку було обрано модуль Wi-Fi ESP8266-01S. Цей модуль має вбудований мікроконтролер та флеш-пам'ять що дозволить нам повністю формувати користувацький веб-застосунок на боці модулю. Вартість модулю становить 63 UAH. Такий вибір компонентів дає нам приблизну підсумкову вартість пристрою у 650 UAH з урахуванням

дрібних елементів таких як реле, транзистори, резистори тощо. Це цілком задовольняє мету з дотримання бюджетності цього проекту.

2.2. Опис архітектури мікроконтролера STM32F410RB

Мікроконтролери модельного ряду STM32 це досить високопродуктивні та гнучкі у використанні прилади які доволі часто застосовуються в саморобних проектах. Абсолютна більшість цих контролерів базується на сімействі ядер ARM Cortex (Cortex-M0, Cortex-M3, Cortex-M4 або Cortex-M7), що в свою чергу надає високий рівень продуктивності, енергоефективності та широкий ряд сценаріїв їх використання. Компоненти архітектури STM32 включають ядро ARM Cortex, периферійні пристрої (UART, SPI, I2C, ADC та DAC), вбудовану флеш-пам'ять для зберігання програмного коду та даних, оперативну пам'ять (RAM) та різні інтерфейси комунікації (USB, Ethernet, CAN, I2S та SDIO). Архітектура мікроконтролера STM32 надає розширені можливості для реалізації різних додатків у системах розумного будинку, забезпечуючи високу продуктивність та надійну роботу з периферійними пристроями.

Головним елементом схеми було обрано модель мікроконтролеру STM32FB410RB від Nucleo (Рис 2.2.1.).

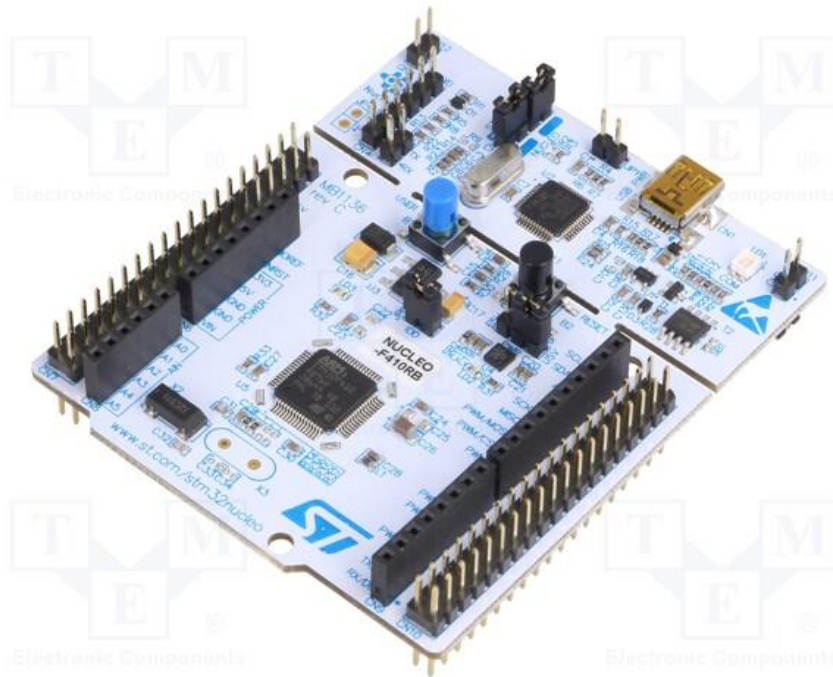


Рис 2.2.1. Зовнішній вигляд мікроконтролера STM32F420RB

Основні характеристики :

- Ядро ARM Cortex-M4F із швидкістю тактування до 100 МГц.
- 128 Кб вбудованої Flash-пам'яті
- 64 Кб оперативної пам'яті SRAM
- Напруга живлення у діапазоні від 2.0 до 3.6 В
- Наявні необхідні роз'єми та інтерфейси (UART, SPI, I2C, USB та CAN)
- 12ти бітний ADC

2.3 Опис архітектури модулю ESP8266-01S.

Елементом що забезпечить зв'язок обрано Wi-Fi модуль на базі ESP8266-01S(Рис 2.3.3).

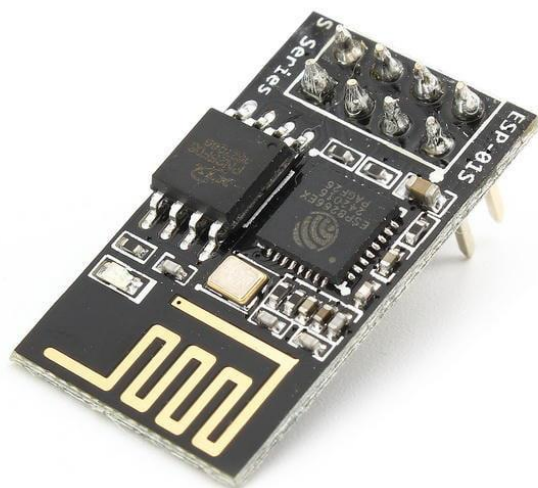


Рис 2.3.3. Wi-fi модуль ESP8266-01S

Основні характеристики:

- Вбудований мікроконтролер Xtensa LX106
- Швидкість процесору 80-160 МГц (Tensilica L106, 32bit)
- Флеш-пам'ять 512 Кб (до 16 Мб)
- ОЗУ 32кб
- Максимальна кількість з'єднань TCP – 5
- Електроспоживання 10мкА-170мА

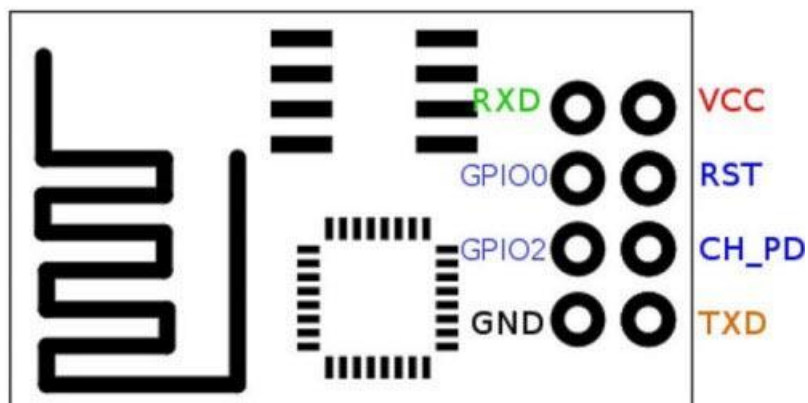


Рис. 2.3.4. Схематичне зображення компонентів на платі ESP8266-01S

ESP8266-01S пропонує широкий функціонал за низьку вартість, що робить його найкращим вибором для цієї схеми. Модуль може підключатися до бездротових мереж Wi-Fi, що дозволяє пристрою отримувати доступ до Інтернету, обмінюватися даними з хмарними сервісами, а також зв'язуватися з іншими пристроями у мережі. ESP8266 має загальнопризначені вхідно-вихідні порти (GPIO), які можуть бути використані для підключення різних пристроїв та сенсорів (Рис 2.3.4.). Це дозволяє керувати зовнішніми пристроями, включаючи світлодіоди, реле, датчики температури та інші пристрої. ESP8266 може взаємодіяти з мікроконтролером STM32F410RB за допомогою протоколу UART. Це дозволяє передавати дані та команди між модулем ESP8266 та мікроконтролером. ESP8266 підтримує набір команд AT, які можна використовувати для керування модулем і виконання різних дій, таких як підключення до мережі, передача даних, налаштування параметрів тощо [9, 10].

2.4. Принципова і структурна схема пристрою керування.

На рисунку 2.4 зображено принципову схему пристрою на базі мікроконтролера STM32F410RB та модулю Wi-Fi ESP8266.

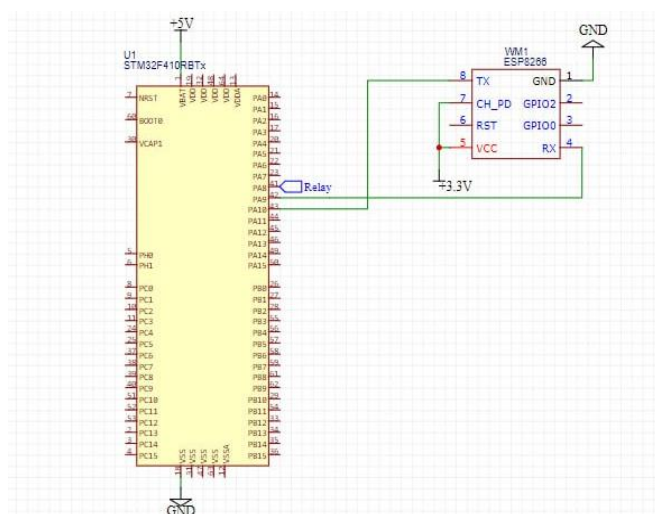


Рис 2.4. Структурна схема модулю пристрою.

На принциповій і структурній схемі модулю пристрою (Рис 2.4) встановлено мікроконтролер STM32F410RB. До піну VBAT підключено джерело живлення +5V. Джерело може виступати сама розетка (за умови використання перетворювача що понижує напругу), або будь яке інше джерело живлення. Також, встановлено Wi-Fi модуль ESP8266 що з'єднаний з МК через UART інтерфейси обох елементів: піни PA9 та PA10 на МК, піни TX та RX на ESP8266. Це забезпечує з'єднання двох елементів, та можливість надійного обміну даними між ними. Модуль ESP8266, виконує функцію забезпечення зв'язку між мікроконтролером STM32F410RB та користувачем.

До піну PA8 мікроконтролеру під'єднано ключ на базі реле HRS1H (Рис 2.5). Саме він відповідає за ввімкнення та вимкнення пристрою.

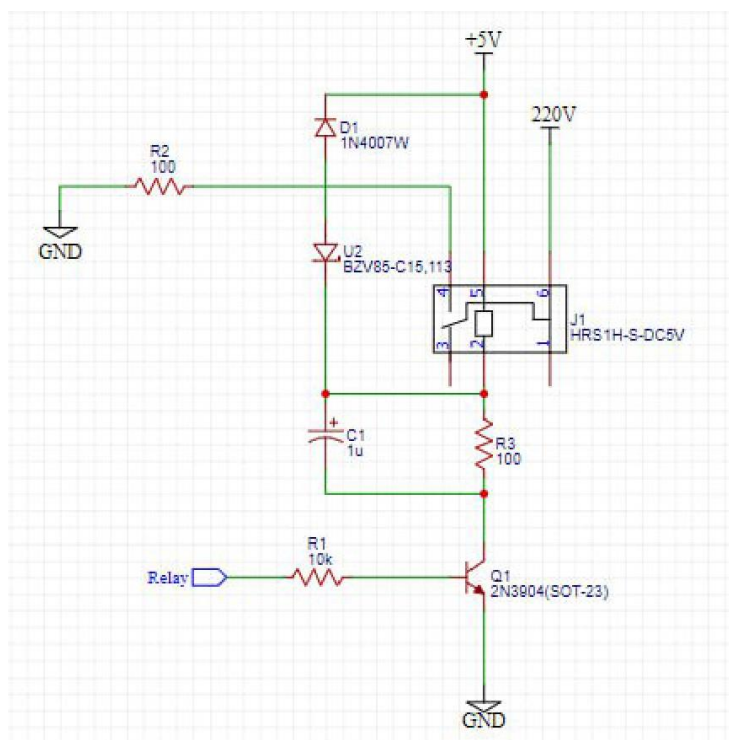


Рис. 2.5. Принципова схема ключу на базі реле.

На зображеній схемі використано транзистор 2N3904, який функціонує в якості ключа, оскільки для живлення реле потрібний значний струм. Резистор R1 виконує функцію обмеження струму, що протікає через базу

транзистора. У зв'язку з тим, що навантаження транзистора має індуктивний характер, для ефективного перемикання енергії котушки необхідно використовувати діод 1N4007, який забезпечує шлях для випрямлення навколоіндуктивного струму при закритті транзистора. Включення діода 1N4007 у схему є необхідним для запобігання появи зворотної електромагнітної енергії під час вимикання реле. Цей діод дозволяє струму, що протікає через індуктивне навантаження, замкнутися на себе, забезпечуючи безпечне вимикання транзистора і захищаючи його від перенапруги. Таким чином, використання діода у схемі забезпечує надійну та стабільну роботу системи керування реле.

Для зниження споживання катушки в ввімкненому режимі (режим утримання якоря реле) додаємо в схему конденсатор С1 та резистор R3, тому що реле має лише великий струм при ввімкненні. При відкритті транзистора конденсатор С1 ще не заряджений, тобто в момент його заряджання він представляє собою майже коротке замикання і струм через катушку іде без обмежень. Всього цього вистачає для ввімкнення реле. Далі конденсатор перетвориться на обрив ланцюга, а реле буде житись через резистор що обмежує струм.

Недоліком цієї схеми є можливе виникнення проблем при спробі ввімкнути реле коли конденсатор ще не повністю не розряджено. У такому випадку, може виникнути нестача струму для успішного ввімкнення реле. Тому, для покращення функціональності цієї системи, необхідно внести зміни до логіки роботи мікроконтролера шляхом введення затримки при ввімкненні та вимкненні реле. Ця затримка дозволить забезпечити необхідний час для розрядження конденсатора перед ввімкненням реле та для стабілізації сигналу після його вимкнення.

Додатково, для скорочення часу вимкнення реле і підвищення ефективності системи, додано стабілітрон BZV85-C15. Цей компонент забезпечує стабільну напругу на своєму виході та допомагає уникнути

коливань та непередбачуваних змін в електричній схемі. Шляхом включення стабілітрона BZV85-C15 у схему, можна покращити точність та надійність процесу вимкнення реле, сприяючи стабілізації електричного сигналу та запобігаючи можливим збуренням чи нестабільностям у роботі системи.

Безпосереднє обмеження струму що виходить до 220V пристрою здійснюється перемикачем у реле HRS1H, між пінами 3 та 4 (off та on відповідно).

Загальний вигляд схеми зображено на рисунку 2.6..

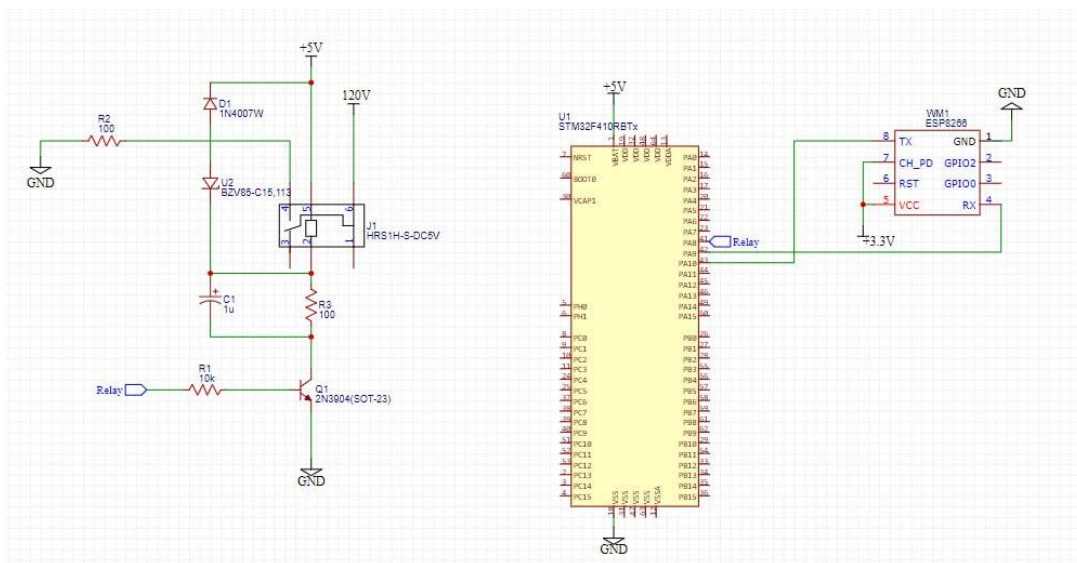


Рис. 2.6. Загальний вигляд принципової та структурної схеми пристрою керування.

2.4.1. Опис процесу побудови прототипу схеми пристрою керування.

При побудові прототипу пристрою керування використано обрані елементи що зібрані на макетній платі (Рис 2.4.1). Було з'єднано вихідні контакти ESP8266 та вхідні контакти мікроконтролера за допомогою UART інтерфейсу. Підключено реле HRS1H до STM32F410RB за допомогою транзистора 2N3904. З'єднано базу транзистора з вихідним контактом мікроконтролера та катод діода 1N4007 з колектором транзистора. Входи/виходи мікроконтролера було підключено до відповідних контактів

реле HRS1H. Забезпечено обмеження струму, що протікає через базу транзистора, за допомогою резистора R1. Один кінець резистора підключено до бази транзистора, а інший кінець до вихідного контакту мікроконтролера. Підключено котушку реле до живлення через резистор R3 та конденсатор C1. Забезпечена правильна полярність підключення діода 1N4007, для запобігання зворотної електромагнітної енергії під час вимикання реле. Додано стабілітрон BZV85-C15 у схему для стабілізації напруги і забезпечення точності та надійності вимкнення реле.

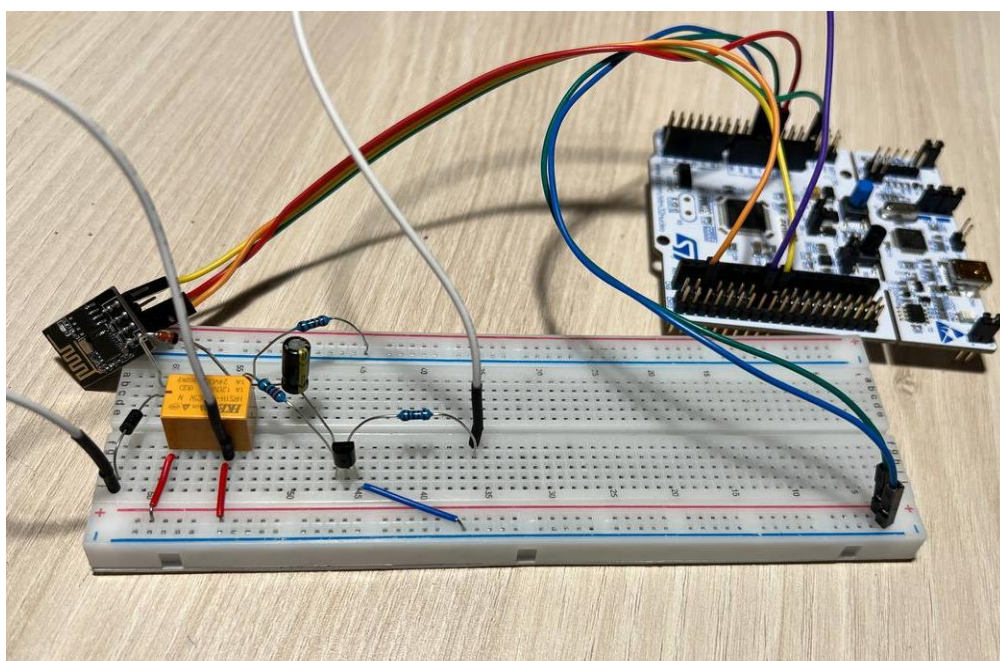


Рис 2.4.1 Прототип схеми керування пристроєм.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ЧАСТИНИ ПРОЕКТУ

3.1 Основні компоненти програмної частини.

Для того щоб створити програмну частину яка забезпечить комунікацію між мікроконтролером, модулем Wi-Fi та кінцевим пристроєм, потрібно прибїгти до використання стороннього програмного забезпечення для роботи з мовами C, C++, HTML та CSS. В даному випадку було використано:

- Visual Studio
- STM32CubeIDE

Та інтернет браузер Opera для відображення веб-застосунку та комунікації з приладом.

3.2 Програмний код для STM32F410RB.

Задля написання програмного коду для мікроконтролеру використано мову програмування C. Мікроконтролер STM32 використовується у даному проекті з метою керування реле на основі отриманого сигналу від модуля ESP8266-S01.

Для початку роботи з мікроконтролером STM32 необхідно акцентувати увагу на точці входу програми - функції main. Саме в цій функції відбувається налаштування інтерфейсу USART та конфігурація портів входу-виходу. У головному циклі програми виконується наступна логіка: за допомогою умовного оператора перевіряється наявність даних у буфері прийому USART1. Функція USART_GetFlagStatus використовується для перевірки стану прапорця USART (конкретно, USART_FLAG_RXNE), що дозволяє визначити наявність отриманих даних. Якщо дані отримані, вони зберігаються у змінній receivedData за допомогою функції USART_ReceiveData.

Наступний блок коду перевіряє, чи отримані дані дорівнюють символу '0'. Якщо ця умова виконується, це означає, що мікроконтролер STM32F410RB отримав низький рівень сигналу від модуля ESP8266-S01. Відповідно до цього, викликається функція GPIO_ResetBits для встановлення заданого піна GPIO (RELAY_PIN) у низький рівень логіки, що призводить до вимкнення реле.

У наступному умовному блоку коду перевіряється, чи отримані дані дорівнюють символу '1'. Якщо ця умова виконується, це означає, що мікроконтролер STM32F410RB отримав високий рівень сигналу від ESP8266-S01, що в свою чергу призводить до виклику функції GPIO_SetBits для встановлення заданого піна RELAY_PIN у високий рівень логіки, що фактично вмикає реле.

```
✓ int main(void) {  
    USART_Configuration();  
    GPIO_Configuration();  
  
    while (1) {  
  
        if (USART_GetFlagStatus(USART1, USART_FLAG_RXNE)) {  
            uint16_t receivedData = USART_ReceiveData(USART1);  
  
            if (receivedData == '0') {  
                GPIO_ResetBits(GPIOA, RELAY_PIN);  
            } else if (receivedData == '1') {  
                GPIO_SetBits(GPIOA, RELAY_PIN);  
            }  
        }  
    }  
}
```

Все що залишається зробити – це описати конфігурацію та налаштування у функціях USART_Configuration і GPIO_Configuration. Повна їх реалізація знаходиться у додатку 1.

3.3 Програмний код для ESP8266-S01.

Для написання програмного коду для модулю ESP8266-01S використано мову програмування C++. Після налаштування електричної схеми з реле та написання початкового коду для STM32, потрібно написати код для ESP8266-S01, який буде виступати у ролі інтерактивного керування самим реле. Для цього потрібно написати простий веб-інтерфейс та конфігураційний код для Wi-Fi мережі.

Для початку в функції setup прописуємо ініціалізацію Wi-Fi модуля та інших допоміжних модулів та в нескінченному циклі викликаємо обробку зміни сигналу.

```
void setup() {
  Serial.begin(115200);
  stmSerial.begin(9600);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  server.on("/", handleRoot);
  server.on("/signal", handleSignal);

  server.begin();
  Serial.println("Server started");
}

void loop() {
  server.handleClient();
}
```

Далі потрібно зосередитись на обробці сигналу, який буде оброблюватись в функції handleClient. Для цього знадобиться обробити

серверні запити та аргументи , які будуть приходити на сервер. В цих аргументах буде зберігатись значення сигналу або низький або високий.

```
void handleSignal() {
    if (server.method() == HTTP_POST) {
        if (server.hasArg("signal")) {
            String signalValue = server.arg("signal");

            if (signalValue == "low") {
                stmSerial.write('0');
            } else if (signalValue == "high") {
                stmSerial.write('1');
            }

            server.send(200, "text/plain", "Signal value set: " + signalValue);
        } else {
            server.send(400, "text/plain", "Bad Request");
        }
    } else {
        server.send(405, "text/plain", "Method Not Allowed");
    }
}
```

Останнє що потрібно буде реалізувати це функція яка буде обробляти саму веб-сторінку. За допомогою змінної html типу String буде сформовано повністю вся сторінка, яка буде відображатись в браузері. Також для того щоб відправити сторінку до браузера , використано метод send об'єкту server.

```
void handleRoot() {
    String html = "<html><body>";
    html += "<h1>Signal Control</h1>";
    html += "<form action='/signal' method='POST'>";
    html += "<label>Signal:</label>";
    html += "<input type='radio' name='signal' value='low'> Low";
    html += "<input type='radio' name='signal' value='high'> High";
    html += "<br>";
    html += "<input type='submit' value='Submit'>";
    html += "</form>";
    html += "</body></html>";

    server.send(200, "text/html", html);
}
```

3.4. Програмний код для веб-застосунку.

Для написання програмного коду для веб-застосунку використано мови програмування HTML та CSS. Так як сторінка повністю формується на боці ESP8266-S01 її можна зробити більшою та стилізувати. Так як принцип роботи формування сторінки залишається незмінним і формується в змінній html , можна написати весь код сторінки в окремому файлі та вставити його в змінну html.

Повноцінний код для html-сторінки знаходиться в додатку 2. Так як основною задачею веб-інтерфейсу буде керування сигналом , знадобиться лише форма з можливістю вибору низького (off) або високого (on) сигналу. Остаточний варіант сторінки зображено на рис 3.4.

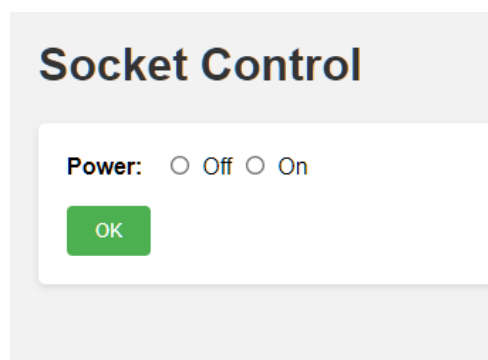


Рис. 3.4. Остаточний варіант сторінки

ВИСНОВКИ

Метою цієї роботи було побудувати прототип приладу що надаватиме можливість дистанційного керування звичайною розеткою. Дослідження та аналіз існуючих систем "розумного будинку" дозволили виявити актуальні рішення та ідейні задуми в цій галузі. В розділі, присвяченому апаратній частині проекту, було проведено вибір основних компонентів схеми, таких як мікроконтролер STM32F410RB та Wi-Fi модуль ESP8266-01S.

В ході виконання роботи було розроблено прототип пристрою керування для системи "розумного будинку", що працює на базі обраного мікроконтролеру STM32F410RB. Реалізовано систему зв'язку між елементами схеми, використовуючи інтерфейс UART що наявний як у обраному модулі Wi-fi зв'язку ESP8266-01S так і у мікроконтролері STM32. Розроблена принципова схема пристрою повинна забезпечити необхідну функціональність розумної розетки, що включає керування ввімкненням та вимкненням за допомоги користувацького веб-застосунку. Також, на макетній платі було зібрано фізичний прототип пристрою з обраних компонентів схеми.

У розділі, присвяченому програмній частині проекту, мовами C та C++ були цілком реалізовані програмні коди що налагоджують комунікацію між модулем ESP8266-S01, мікроконтролером STM32F410 та пристроєм користувача. Реалізований код для STM32 дозволяє керувати реле на основі отриманого сигналу від ESP8266. Конфігураційний код для ESP8266 забезпечив зв'язок між модулем та мікроконтролером, та дозволив модулю виступати у ролі інтерактивного керування реле. Також, мовами програмування HTML та CSS було розроблено функціональний веб-застосунок що представляє собою користувацький інтерфейс для дистанційного керування подачею електроенергії на прилад.

Наступними етапами розробки пристрою для дистанційного керування розеткою можуть бути:

- Реалізація функціонуючої схеми шляхом безпосереднього програмування обох мікроконтролерів.
- Вбудовування схеми до власного корпусу з «гніздами» розеток для підключення електроприладів.
- Інтегрування інших корисних функцій та сценаріїв використання приладу. Наприклад: функція відображення напруги та споживаної електроенергії, таймер вимкнення та ввімкнення приладу.
- Реалізація оновленого web-застосунку задля поліпшення досвіду користувача.

Загалом, ця робота допомогла дослідити ефективність та можливості систем "розумного будинку" на прикладі системи керування розеткою. Результати дослідження та виконання проекту будуть використані задля подальшого розвитку системи та реалізації нових розумних пристроїв, спрямованих на автоматизацію домашнього середовища та забезпечення комфорту користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bluebird Marine Systems, Donald Wale: Tesla Radio Control. URL: [https://www.bluebird-electric.net/Bluebird Boats Ships Systems/Nikola Tesla.htm](https://www.bluebird-electric.net/Bluebird_Boats_Ships_Systems/Nikola_Tesla.htm)
2. MiMi Smart UA. Історія розумного будинку. URL: <https://www.smarthouse.ua/ua/istoriya-umnogo-doma.html>
3. Eugene Zhilina, Projector magazine: Monsanto's House of the Future. URL: [https://www.academia.edu/4690915/Monsanto House of the Future](https://www.academia.edu/4690915/Monsanto_House_of_the_Future)
4. Clipsal by Schneider, Creating the smart home you always dreamed of. URL: <https://www.clipsal.com/smarthome>
5. Orvibo New Allone WiFi Smart Controller. URL: <https://www.amazon.co.uk/Orvibo-Intelligent-Controller-Automation-Compatible-black/dp/B07FKL3BJV>
6. Tkachenko Roman, Center of Technical Safety: WiFi розетка Zeoota PS022. URL: <https://tehbezpeka.ua/ua/p1110924909-wifi-rozetka-umnaya.html>
7. Alexandr Konotopski. Ajax Systems. Socket: A smart plug for a smart home. URL: <https://ajax.systems/blog/meet-socket/>
8. STMicroelectronics. Datasheet for Arm-Cortex-M4 STM32F410xB. 10 серпня 2016 року. PDF's URL: <https://www.st.com/resource/en/datasheet/stm32f410cb.pdf> с. 16 - 24
9. AI-Thinker team. 0C-ESP8266 WROOM Wifi Module Datasheet. 2015 рік. PDF's URL: <https://www.microchip.ua/wireless/esp01.pdf> с. 7 – 14
10. Aero Arduino. ESP8266-01S Serial Wireless WLAN WiFi Transceiver Module. URL: <https://aeroarduino.com/store/azdelivery-esp8266-esp-01-serial-wireless-wlan-wifi-transceiver-module-with-esp-01-breakout-breadboard-adapter-for-arduino-including-ebook/>

КОД КОНФІГУРАЦІЇ USART

```
#include "stm32f4xx.h"
#include "stdio.h"

#define UART_TX_PIN GPIO_Pin_9
#define UART_RX_PIN GPIO_Pin_10
#define RELAY_PIN GPIO_Pin_12

void USART_Configuration(void);
void GPIO_Configuration(void);

int main(void) {
    USART_Configuration();
    GPIO_Configuration();

    while (1) {

        if (USART_GetFlagStatus(USART1, USART_FLAG_RXNE)) {
            uint16_t receivedData = USART_ReceiveData(USART1);

            if (receivedData == '0') {

                GPIO_ResetBits(GPIOA, RELAY_PIN);
            } else if (receivedData == '1') {

                GPIO_SetBits(GPIOA, RELAY_PIN);
            }
        }
    }
}
```

```

    }
}
}
}

```

```

void USART_Configuration(void) {
    USART_InitTypeDef USART_InitStructure;
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource9, GPIO_AF_USART1);
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource10, GPIO_AF_USART1);
    GPIO_InitStructure.GPIO_Pin = UART_TX_PIN;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
    GPIO_Init(GPIOA, &GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin = UART_RX_PIN;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOA, &GPIO_InitStructure);
    USART_InitStructure.USART_BaudRate = 115200;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;
}

```

```
        USART_InitStructure.USART_Mode = USART_Mode_Rx |  
USART_Mode_Tx;  
        USART_Init(USART1, &USART_InitStructure);  
        USART_Cmd(USART1, ENABLE);  
    }
```

```
void GPIO_Configuration(void) {  
    GPIO_InitTypeDef GPIO_InitStructure;  
  
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);  
    GPIO_InitStructure.GPIO_Pin = RELAY_PIN;  
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;  
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;  
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;  
    GPIO_Init(GPIOA, &GPIO_InitStructure);  
}
```

КОД HTML-СТОРИНКИ ДЛЯ ВЕБ-ЗАСТОСУНКУ

```
<!DOCTYPE html>
<html>
<head>
<title>Signal Control</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    padding: 20px;
  }

  h1 {
    color: #333333;
  }

  form {
    background-color: #ffffff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  }

  label {
    font-weight: bold;
    margin-right: 10px;
  }

  input[type='radio'] {
    margin-right: 5px;
  }

  input[type='submit'] {
    background-color: #4CAF50;
    color: #ffffff;
    padding: 10px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
  }
</style>
</head>
<body>
  <h1>Signal Control</h1>
  <form>
    <label>Signal Control</label>
    <input type='radio' />
    <input type='submit' value='Signal Control' />
  </form>
</body>
</html>
```

```
    input[type='submit']:hover {
      background-color: #45a049;
    }
  </style>
</head>
<body>
  <h1>Socket Control</h1>
  <form action='/signal' method='POST'>
    <label>Signal:</label>
    <input type='radio' name='signal' value='low'> Off
    <input type='radio' name='signal' value='high'> On
    <br>
    <br>
    <input type='submit' value='Submit'>
  </form>
</body>
</html>
```