Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Система керування шторами / жалюзі через ESP32 та мобільний застосунок
Назва теми

КвРКІ 210250.21.02.10 ПЗ
Шифр

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Виконав: студент IV курсу, група КІ2-21-2 _____ Владислав ТКАЧУК
Підпис Ініціали, прізвище

Керівник _____ Олег САВЕНКО
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ Тетяна КИСІЛЬ
Підпис, дата Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Ольга ПАВЛОВА
Підпис Ініціали, прізвище

«___» _червня_ 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 10 " 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

_____
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система керування шторами / жалюзі через ESP32 та мобільний застосунок

Керівник проекту (роботи) Олег САВЕНКО, д.т.н., проф.
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі

Проєктування системи керування шторами/жалюзі через ESP32 та мобільний застосунок

Програмно-апаратна реалізація та тестування системи керування шторами на базі ESP32

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

Схема роботи пристроїв в системі

Електрична принципова схема

Схема підключення пристроїв

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання «_10_» 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 –	01.03.2025	виконано
4	Робота над розділом 2 –	01.04.2025	виконано
5	Робота над розділом 3 –	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент _____ Владислав ТКАЧУК
Підпис Ініціали, прізвище

Керівник роботи _____ Олег САВЕНКО
Підпис Ініціали, прізвище

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 210250.21.02.10 ПЗ	Пояснювальна записка	65		
			Графічні матеріали			
2		КвРКІ 210250.21.02.10 Е8	Схема роботи пристроїв в системі	1		
3		КвРКІ 210250.21.02.10 Е8	Електрична принципова схема	1		
4		КвРКІ 210250.21.02.10 Е8	Схема підключення пристроїв	1		

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 210250.21.02.10 ВП		
Розробив		Ткачук		01.06.15	Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Савенко				У	1	1
Н. контр.		Кисіль		02.06.15		ХНУ, КІ2-21-2		
Затв.		Павлова		02.06.15				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система керування шторами / жалюзі через ESP32 та мобільний застосунок».

Автор роботи: Владислав ТКАЧУК.

Керівник роботи: Савенко Олег Станіславович.

Пояснювальна записка: 64 с., 21 рис., 4 табл., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

ESP32, IoT, Розумний будинок, Архітектура, EasyEDA, Blynk, MQTT,Wi-Fi, Автоматизація.

У роботі розглядається розробка та дослідження кіберфізичної системи керування шторами та жалюзі з використанням мікроконтролера ESP32 та мобільного застосунку. Метою дипломної роботи є визначення умов та особливостей застосування апаратного і програмного забезпечення для реалізації автоматизованого керування елементами інтер'єру на базі IoT-рішень, а також оцінка ефективності обробки інформації в межах системи для забезпечення стабільного функціонування в умовах побутового середовища.

Об'єктом дослідження є функціонування моніторингових та керувальних елементів системи автоматизації побутових процесів. Предметом дослідження є оцінка режимів застосування елементів керування шторами та жалюзі в рамках побудови кіберфізичної системи.

Під час виконання дослідження застосовано метод систематичного огляду літератури для аналізу сучасних технологій у сфері IoT та розробки мобільних застосунків, а також проведено практичне моделювання архітектури системи з використанням ESP32, бази даних та мобільного інтерфейсу для дистанційного керування.

Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП 4

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ 6

1.1 Аналіз предметної області та її особливості 6

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень 10

1.3 Методологічні підходи до вирішення задачі за темою дослідження.... 14

1.4. Постановка завдання 16

1.5. Висновок до першого розділу 17

2 ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ ШТОРАМИ/ЖАЛЮЗІ ЧЕРЕЗ ESP32 ТА МОБІЛЬНИЙ ЗАСТОСУНОК 18

2.1 Апаратна частина 18

2.2 Архітектура та принцип функціонування системи керування жалюзі .. 27

2.3. Застосування протоколу MQTT та інтеграція з Blynk 29

2.4. Електричні характеристики системи 34

2.5 Висновки до другого розділу 38

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ КЕРУВАННЯ ШТОРАМИ НА БАЗІ ESP32 40

3.1 Налаштування Arduino IDE та Blynk 40

3.2 Монтажна схема 46

3.3 Алгоритм та код програми 51

3.4 Застосунок Blynk та інтерфейс програмно-технічного засобу системи керування шторами 59

3.5 Вартість матеріалів 61

3.6 Висновки до третього розілу 62

ВИСНОВКИ..... 64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 66
ДОДАТОК А 73
ДОДАТОК Б 74
ДОДАТОК В 75
ДОДАТОК Г..... 76

ВСТУП

У сучасному світі автоматизація домашніх процесів стала невід'ємною частиною комфорту, безпеки та енергоефективності. Розвиток технологій Інтернету речей (IoT) суттєво вплинув на концепцію розумного дому, перетворивши традиційні будинки на інтерактивні простори з можливістю дистанційного або автономного керування більшістю побутових пристроїв. Одним із важливих елементів цієї екосистеми є автоматизовані системи керування шторами та жалюзі. Такі системи дають можливість регулювати рівень освітлення в приміщенні, покращувати мікроклімат, зменшувати навантаження на системи кондиціонування, підвищувати рівень приватності для мешканців.

Інтеграція механічних елементів керування (двигунів, сервоприводів) з електронними мікроконтролерами та мобільними додатками створює гнучкі рішення, які можна адаптувати до конкретних умов використання. Зокрема, використання сучасних мікроконтролерів, таких як ESP32, дозволяє здійснювати як локальне, так і дистанційне керування, завдяки вбудованим модулям Wi-Fi і Bluetooth. Це відкриває можливості для створення енергоефективних, масштабованих і зручних для користувача систем.

Завдяки мобільним додаткам, розробленим на таких платформах, як React Native, Flutter або нативних фреймворках Android/iOS, користувач отримує інтуїтивно зрозумілий інтерфейс управління. Це не тільки спрощує взаємодію з пристроєм, але й дозволяє інтегрувати такі функції, як таймери, автоматичне відкриття/закриття залежно від часу доби або рівня освітленості, синхронізація з голосовими помічниками тощо.

В даній дипломній роботі було поставлено за мету розробити систему керування шторами або жалюзі на базі мікроконтролера ESP32, з можливістю дистанційного керування за допомогою мобільного додатку. Розробка власної системи управління шторами з керуванням із мобільного додатку - це перспективний напрямок, адже дозволяє задовольнити потреби автоматизації та

віддаленого керування, оптимізувати витрати і отримати цінний досвід в сфері інформаційних технологій та розумних будинків. В якості основної технології обміну даними було обрано Wi-Fi, оскільки вона забезпечує стабільне бездротове з'єднання в межах домашньої мережі. Для розробки клієнтської частини (мобільного додатку) були використані готові фреймворки для скорочення часу розробки та забезпечення крос-платформної сумісності.

В ході дослідження було проаналізовано існуючі на ринку рішення, визначено системні вимоги, підібрано оптимальну апаратну конфігурацію та розроблено архітектуру взаємодії мікроконтролера та мобільного додатку. Отримані результати можуть бути використані для подальшої розробки та розгортання подібних систем в домашніх та комерційних приміщеннях, включаючи офіси, готелі або навчальні заклади. Розробка також може слугувати основою для більш складних рішень у сфері автоматизації та систем екологічного менеджменту.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області та її особливості

Автоматизовані системи керування пройшли довгий шлях розвитку від простих ручних пристроїв до складних автоматизованих рішень. Спочатку для керування освітленням і приватністю використовували дроти і прути, що вимагало безпосередньої фізичної взаємодії. З розвитком технологій поява електричних систем дозволила керувати шторами та жалюзі за допомогою вимикачів. Однак справжня революція відбулася з появою розумних систем, які використовують передові технології для забезпечення зручності, енергоефективності та інтеграції з екосистемами «розумного будинку» [22].

Все більшого значення в сучасних житлових приміщеннях набуває автоматизація штор та жалюзі. Це пов'язано з кількома ключовими факторами, такими як зростаючі вимоги до комфорту і зручності, зростання цін на енергоносії і тенденція до інтегрованих систем «розумного будинку». Розумні системи керування шторами та жалюзі пропонують різні переваги, такі як дистанційне керування, автоматичне регулювання залежно від часу доби або зовнішніх умов, голосове керування та інтеграція з іншими розумними пристроями [1].

Історія жалюзі як функціонального елементу архітектурного простору сягає глибокої давнини. Перші згадки про ці конструкції можна знайти в джерелах, що описують життя в Стародавньому Єгипті та Греції, де для захисту від сонячних променів використовували дерев'яні решітки або тканинні завіси, натягнуті на віконні прорізи. Ці пристрої виконували як утилітарну, так і естетичну функцію, обмежуючи проникнення світла і тепла в приміщення.

Значний прорив у розвитку штор стався в епоху Відродження (рис 1.1), коли зріс інтерес до комфорту внутрішнього середовища в приміщенні. У цей період були розроблені перші сучасні горизонтальні жалюзі з дерев'яних планок. Вони кріпилися до рами і дозволяли змінювати кут нахилу ламелей вручну, регулюючи

таким чином інтенсивність світла. Згодом штори стали частиною інтер'єру знатних будинків, а їхня форма та матеріали були оптимізовані відповідно до архітектурних та дизайнерських тенденцій [29].

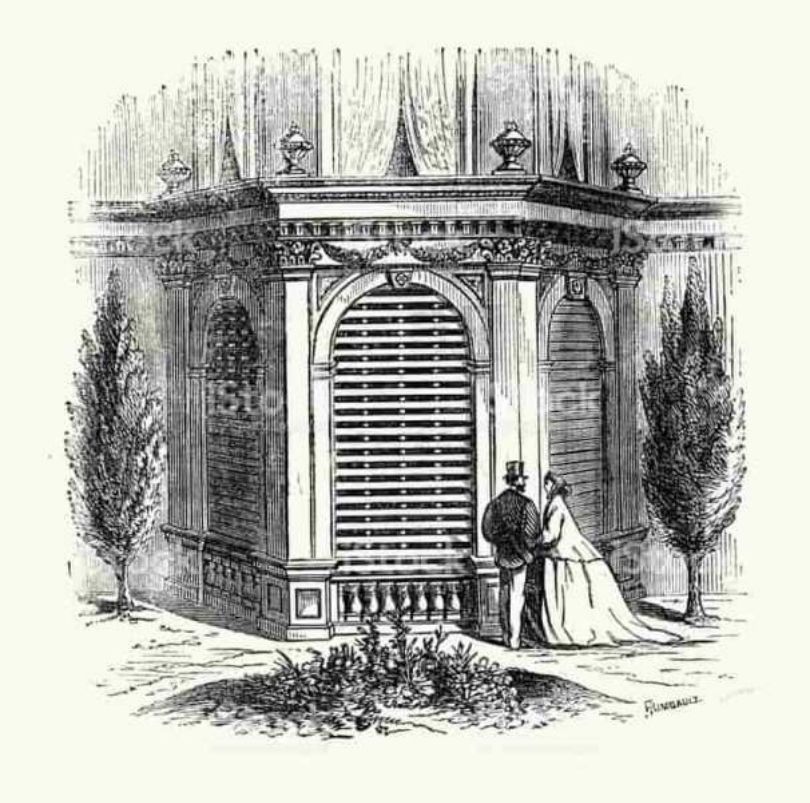


Рисунок 1.1 – Жалюзі в епоху Відродження [3]

У 18 столітті французькі майстри значно вдосконалили дизайн горизонтальних жалюзі, що стало важливим кроком до їх широкої популяризації. У 20 столітті жалюзі отримали подальше технічне вдосконалення завдяки впровадженню нових матеріалів, таких як алюміній і пластик. Це зробило конструкції легшими, дешевшими і доступнішими для ширшого кола споживачів. У повоєнні роки, особливо в 1950-х і 1960-х, жалюзі стали невід'ємною частиною модерністської архітектури, в якій домінували функціоналізм і прагнення до простоти. Пізніше з'явилися вертикальні жалюзі, рулонні та касетні системи, а також автоматизовані моделі з дистанційним управлінням, які були інтегровані в системи «розумного будинку».

Сьогодні вертикальні жалюзі - це не тільки спосіб організації природного освітлення, а й частина енергозбереження та естетичного оформлення внутрішніх і зовнішніх просторів. Їх еволюція продовжує рухатися в напрямку екологічності, інноваційних матеріалів та технологічної інтеграції, ілюструючи динамічну еволюцію цього елементу архітектурного дизайну протягом століть [9].

З розвитком електроніки та концепції «розумного будинку» поява моторизованих шторних систем стала логічною частиною дизайну інтер'єру. Перші спроби автоматизації з'явилися ще в середині 20 століття - у 1960-х роках у США розробили електродвигуни, які відкривали та закривали штори за допомогою настінних вимикачів. Функціональність цих систем була обмеженою, але були закладені основи для подальшого розвитку з метою підвищення комфорту користувачів.

З 1980-х років, з широким розповсюдженням мікропроцесорних технологій, почали з'являтися перші інтегровані системи освітлення та кондиціонування, а жалюзі стали частиною загальної інтегрованої автоматизованої системи. Важливим кроком у цьому напрямку стало формування на початку 1990-х років концепції «розумного будинку», яка передбачає взаємодію побутової техніки з центральною або дистанційною системою управління.

На початку 21 століття моторизовані жалюзі почали масово з'являтися в приватних і комерційних будівлях, особливо в енергоефективних будівлях. Сучасні системи використовують датчики освітленості та температури, а також таймери, що дозволяють жалюзі самостійно регулювати положення ламелей відповідно до умов навколишнього середовища або заздалегідь визначених сценаріїв. Провідні бренди, такі як Somfy (Франція) і Lutron (США), стали піонерами в розробці цих рішень, пропонуючи системи, якими можна керувати за допомогою Wi-Fi, Bluetooth або голосових помічників[1].

З розвитком Інтернету речей (IoT) моторизовані жалюзі стали важливою частиною стратегій енергоефективності будівель, особливо в проектах «зеленого» будівництва. Вони допомагають зменшити навантаження на системи

кондиціонування та опалення, динамічно керуючи потоком світла і тепла. Таким чином, сучасні моторизовані жалюзі поєднують в собі довгу практичну історію з новітніми технологіями будівництва та комфорту. Візуальну репрезентацію можна спостерігати на рисунку 1.2.

Рисунок 1.2 – Жалюзі в системі “розумного дому” [8]

Системи керування шторами та жалюзі, які можна класифікувати як «розумні», характеризуються здатністю виконувати функції автоматично або на основі команд користувача через інтелектуальний інтерфейс. Ключові особливості цих систем включають дистанційне керування, програмування, інтеграцію зі смартфонами і голосовими помічниками, а також оптимізацію енергоспоживання і підвищення безпеки будинку. Вони автоматизують відкривання/закривання віконних завіс на підставі датчиків освітленості, температури, розкладів тощо, забезпечуючи зручність та енергозбереження. Ринок таких рішень активно зростає: за оцінками, обсяг світового ринку автоматичних штор сягне близько 6,13 млрд дол. до 2034 р. при середньорічному зростанні ≈5.5 % . Це пояснюється попитом на підвищений комфорт і економію енергії – наприклад, системи з вбудованими

датчиками світла та руху можуть зменшувати витрати на освітлення до 20 % [9]. Автоматичне керування шторами дозволяє підтримувати оптимальну температуру та освітленість приміщення (закривати штори влітку для зменшення нагрівання і відкривати взимку для використання сонячного тепла), а також підвищує безпеку та комфорт (напр., керування одним натисканням з мобільного додатку).

Розробка спеціалізованих систем управління, в тому числі на основі мікроконтролерів ESP32 і додатків для мобільних телефонів, є актуальною тенденцією, яка може бути адаптована до конкретних потреб і надавати гнучкі, економічно ефективні рішення [6].

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Сучасні системи управління шторами та жалюзі пропонують цілий ряд методів взаємодії, від простих механічних до високотехнологічних автоматизованих рішень. До них відносяться традиційні (механічні) методи, електромеханічні системи, дистанційно керовані системи, а також інтегровані рішення в рамках концепції «Розумний дім» [4].

Традиційні методи управління (рис. 1.3) вимагають безпосередньої фізичної взаємодії з механізмом відкривання або закривання вікна. Це можуть бути ручні шнури для підняття або опускання штор або жалюзі, а також штанги або ланцюжки для повороту ламелей горизонтальних і вертикальних жалюзі. Такі системи не потребують електроживлення, що робить їх дешевими в установці та обслуговуванні. Завдяки своїй простій конструкції вони надійні та довговічні.

Крім того, відсутність центрального управління унеможливлює синхронізацію декількох віконних систем, що може бути важливим для досягнення комфортного рівня освітлення, теплоізоляції або безпеки. З цієї причини традиційні методи все частіше замінюються більш сучасними рішеннями, що забезпечують інтелектуальне управління за заздалегідь визначеними сценаріями, графіками або зовнішніми умовами (освітлення, температура, присутність людей тощо).

Рисунок 1.3 – Традиційний метод взаємодії [1]

Звичайні електричні системи керування дозволяють керувати жалюзі або шторами за допомогою перемикачів, безпосередньо підключених до двигуна. Ці системи зазвичай забезпечують лише базові функції відкривання та закривання. Наприклад, у випадку двигуна жалюзі, підключеного до чотирипровідного шнура, напрямок руху контролюється перемиканням між двома електричними дротами, призначеними для руху вгору і вниз. Важливо, щоб ці два дроти не підключалися одночасно, щоб уникнути пошкодження двигуна [36, 37]. Існують також варіанти, які використовують двигуни низької напруги (наприклад, 24 В), де для керування напрямком руху використовується трипровідний кабель від контролера. У випадку двигунів змінного струму 240 В, керування може здійснюватися через «сухі контакти», які не перебувають під напругою, за допомогою кабелю CAT 5/6.5 Хоча ці системи зручніші, ніж ручне керування, їм, як правило, бракує розширених функцій, таких як дистанційне керування, програмування за розкладом або інтеграція з «розумним» домом. Їх використання часто обмежується ситуаціями, коли потрібен простий контроль одного або декількох віконних покриттів з фіксованого місця. Також є системи дистанційного керування жалюзі та шторами використовують бездротові технології, такі як радіочастотні (РЧ) або інфрачервоні (ІЧ), для роботи на фіксованій відстані (рис 1.4). Радіочастотне дистанційне керування має перевагу над ІЧ, оскільки не потребує прямої видимості між пультом дистанційного керування та приймачем і може працювати на більших відстанях.

Деякі системи використовують власні радіочастотні протоколи, що може ускладнити інтеграцію зі сторонніми системами керування [25].

Рисунок 1.4 – Електрична система керування [8]

Сучасні системи керування шторами та жалюзі все більше інтегруються в екосистеми «розумного будинку» за допомогою бездротових протоколів, таких як Wi-Fi [40], Bluetooth [39], Zigbee та Z-Wave.4 Ці протоколи дозволяють керувати віконними конструкціями за допомогою смартфонів, голосових помічників (таких як Amazon Alexa, Google Assistant, Apple Siri) та інших розумних пристроїв. Зростаюча популярність «розумних будинків» збільшує попит на такі інтегровані системи управління.8 Споживачі прагнуть до безперешкодної інтеграції всіх домашніх пристроїв, включаючи віконні ролети, щоб підвищити зручність і можливості автоматизації.

Використання цифрових мереж керування двигунами дозволяє індивідуально керувати кожним двигуном, навіть якщо вони з'єднані послідовно.4 Це забезпечує більшу гнучкість і контроль, а також полегшує майбутні зміни в конфігурації

простору. Наприклад, зміна положення перемикачів для цих жалюзі може бути виконана програмно без необхідності фізичного перепідключення [16].

Системи управління розумним будинком (BMS) також можуть включати в себе управління шторами і жалюзі, використовуючи дані з датчиків освітленості і температури для автоматичного регулювання положення віконних покриттів з метою підвищення енергоефективності та створення комфортного середовища.4

На ринку існує кілька платформ для керування IoT-пристроями, зокрема розумними шторами. Blynk — це зручна мобільна платформа з простим графічним інтерфейсом і швидким підключенням ESP32 через Wi-Fi. Вона ідеально підходить для швидкого прототипування завдяки готовим віджетам і базовій безкоштовній версії. Приклад додатку на смартфоні зображений на рисунку 1.5. Проте новіша хмарна версія Blynk IoT може вимагати підписки для доступу до розширеного функціоналу, а частково закритий код обмежує кастомізацію [11].

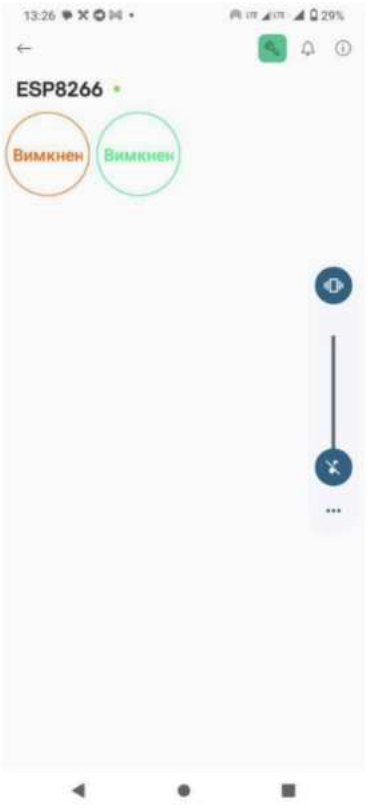


Рисунок 1.5 – Застосунок Blynk

Home Assistant - це потужне рішення з відкритим вихідним кодом і великою спільнотою розробників. Він працює локально, підтримує багато протоколів (MQTT, REST) і має гнучкий інтерфейс, але вимагає серверного обладнання та часу на навчання (налаштування YAML, інтеграції). Він ідеально підходить для тих, хто планує централізовано керувати кількома пристроями, включаючи ESP32 [27].

Домашні додатки є найбільш гнучким рішенням з повним контролем над функціональністю, інтерфейсом і способом підключення ESP32. Він працює без хмари, може бути локальним, але вимагає значних знань і ресурсів для розгортання та обслуговування.

На основі порівняння доступних технологічних рішень можна припустити, що найбільш ефективним варіантом для прототипування системи є поєднання мікроконтролера ESP32 з мобільним інтерфейсом управління через платформу Blynk. Для створення надійної передачі даних рекомендується використання MQTT або REST API [34].

1.3 Методологічні підходи до вирішення задачі за темою дослідження

При розробці інтелектуальних систем керування шторами одним з найпопулярніших рішень є використання мікроконтролерів ESP32. Це пов'язано з поєднанням низької вартості, широких можливостей і простоти інтеграції з іншими пристроями. ESP32 оснащений двоядерним процесором, інтегрованими модулями Wi-Fi і Bluetooth (BLE), що дозволяє реалізувати як прості, так і більш складні сценарії автоматизації. Завдяки підтримці Wi-Fi ESP32 легко підключається до домашньої мережі без використання додаткового обладнання. Це забезпечує стабільну і швидку передачу даних на великі відстані всередині приміщення. Наприклад, пристрій, встановлений біля вікна, може легко зв'язатися з сервером або мобільним додатком, навіть якщо роутер знаходиться в іншій кімнаті.

Bluetooth Low Energy (BLE) - ще одна перевага ESP32 [6]. Його часто використовують як альтернативний або резервний канал зв'язку, особливо за

відсутності Wi-Fi або для прямого локального управління через смартфон на невеликих відстанях. BLE споживає набагато менше енергії, тому ця функція особливо важлива у випадку автономного живлення або коли важлива економія заряду батареї. Деякі проекти також використовують інші бездротові технології, такі як Zigbee, Z-Wave або навіть LoRa, але вони потребують додаткових модулів або шлюзів. Такі протоколи обирають, коли важливо покрити більшу площу, уникнути перевантаження Wi-Fi або забезпечити стабільність у мережі з великою кількістю пристроїв. Однак для більшості домашніх застосувань Wi-Fi залишається найзручнішим і найпоширенішим вибором.

На рівні обміну даними між пристроями найчастіше використовується протокол MQTT [3]. Це легкий та ефективний мережевий протокол, розроблений спеціально для систем з обмеженими ресурсами та ненадійними каналами зв'язку - підходить для потреб IoT. Протокол базується на моделі «опублікувати/записати», яка дозволяє асинхронну доставку повідомлень між пристроями. Наприклад, ESP32 надсилає дані про положення штор або поточну температуру відповідному MQTT-посереднику, а у відповідь отримує команди відкрити або закрити штори від іншого посередника. Такий підхід дозволяє легко масштабувати систему, додаючи нові датчики, сервіси або логіку без зміни структури взаємодії пристроїв.

Посередником MQTT в цьому проекті може бути, наприклад, Home Assistant [27], розгорнутий на локальному сервері або мікрокомп'ютері типу Raspberry Pi. Це дозволяє забезпечити повністю автономну систему, яка не залежить від зовнішніх хмарних сервісів. MQTT підходить на роль «нервової системи» розумного будинку, завдяки низькій затримці, здатності повторно відправляти повідомлення в разі втрати зв'язку і невеликому обсягу переданих даних.

Хоча альтернативами MQTT можуть бути такі підходи, як HTTP/REST API або WebSocket, вони, як правило, менш енергоефективні і не завжди забезпечують гнучкість, необхідну для розширення системи. Наприклад, REST-запити зазвичай передбачають прямі запити до конкретних пристроїв, тоді як MQTT дозволяє легко

керувати кількома пристроями одночасно через одного брокера без прив'язки до конкретної IP-адреси [30].

Таким чином, поєднання ESP32 як потужного та доступного мікроконтролера з підтримкою Wi-Fi/BLE [4] та використання MQTT як основного протоколу обміну повідомленнями створює ефективну основу для розробки інтелектуальної системи керування шторами. Це рішення поєднує в собі простоту реалізації, функціональність та масштабованість, що робить його особливо привабливим для домашніх проектів та ентузіастів «зроби сам».

1.4. Постановка завдання

Метою цього дослідження є розробка системи управління шторами або жалюзі на базі мікроконтролера ESP32 та мобільного додатку. Ця система покликана створити сучасне, зручне рішення для розумних будинків. Основною метою цього проекту є розробка функціонального прототипу, який дозволить користувачам без зусиль керувати положенням штор за допомогою інтуїтивно зрозумілого інтерфейсу на своєму смартфоні. Це включає як ручне дистанційне керування, таке як відкривання та закривання, так і можливість налаштування автоматичних сценаріїв. Наприклад, систему можна запрограмувати на відкривання штор у визначений час доби або при досягненні певного освітлення в приміщенні. Це буде досягнуто за допомогою датчиків освітлення, які будуть відправляти дані до процесора, а той, в свою чергу, давати команди для сервопривода, такі як відкривання/закривання жалюзі. Це створює новий сучасний підхід до взаємодії із власним оточенням.

Проект охоплює кілька ключових областей, включаючи апаратне забезпечення (вибір та інтеграція ESP32, двигунів і датчиків), логіку програмного забезпечення контролера (керування двигунами, обробка команд, зворотний зв'язок) та розробку мобільного додатка, який функціонує як клієнт системи.

Загалом, цей проект демонструє можливості ESP32 в області IoT, одночасно слугуючи практичним кроком на шляху до розробки індивідуалізованої системи розумного будинку, з акцентом на комфорті, доступності та спрощенні управління побутовими приладами. Подальше розширення проєкту може бути глибшою інтеграцією із «розумним будинком», адже система легко зможе інтегруватись до інших пристроїв та працювати в зв’язці з ними.

1.5. Висновок до першого розділу

Аналіз предметної області систем керування шторами та жалюзі показує, що ця галузь активно розвивається, пропонуючи користувачам все більше зручних, ефективних та інтелектуальних рішень. Існує широкий спектр технологій та підходів, від простих ручних механізмів до складних систем, інтегрованих у розумні будинки. Розробка власної системи керування на базі ESP32 та мобільного додатку – це перспективний напрямок, який дозволяє задовольнити конкретні потреби, оптимізувати витрати та отримати цінний досвід у сфері Інтернету речей. Наступні розділи кваліфікаційної роботи будуть присвячені детальному розгляду архітектури, розробки та тестування такої системи. В результаті аналізу було виявлено, що системи керування розумними шторами є актуальним напрямком розвитку IoT-технологій, що поєднують автоматизацію побутових процесів, енергоефективність та комфорт користувача. Вони здатні значно зменшити витрати на освітлення та опалення, підвищити безпеку та створити персоналізовані умови для приміщення. На основі порівняння доступних технічних рішень було визначено, що найефективнішим варіантом створення прототипу системи є поєднання мікроконтролера ESP32 з мобільним інтерфейсом керування через платформу Blynk. Для створення надійної передачі даних доцільно використовувати MQTT або REST API.

2 ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ ШТОРАМИ/ЖАЛЮЗІ ЧЕРЕЗ ESP32 ТА МОБІЛЬНИЙ ЗАСТОСУНОК

2.1 Апаратна частина

Оскільки ми продовжуємо дослідження з розробки інтелектуальної системи керування шторами та жалюзі на базі мікроконтролера ESP32, вибір компонентної бази є важливим кроком у реалізації проекту. Нижче ви знайдете детальний опис найважливіших апаратних компонентів, що використовуються для створення системи автоматизації:

- Мікроконтролер ESP32;
- фоторезистори LDR (2 шт.)
- сервопривід SG90;
- резистори 10 кОм (2 шт.);
- Макетна плата;
- З'єднувальний кабель;
- Блок живлення 5В;
- Смартфон з мобільним додатком Blynk.

ESP32 - це потужний мікроконтролер з бездротовим Wi-Fi з'єднанням, який дозволяє розробляти автономні IoT-пристрої (рис. 2.1) [28]. Я обрав ESP32 через його низьку ціну, високу продуктивність та сумісність з Arduino IDE, що значно полегшує розробку прошивки. Завдяки аналоговим входам мікроконтролер може зчитувати аналогові значення з світлочутливих резисторів (LDR), які використовуються для визначення рівня освітленості. ESP32 було обрано завдяки багатим функціональним можливостям, зокрема підтримці Wi-Fi і Bluetooth, великій кількості GPIO-контактів, здатності обробляти аналогові сигнали. Це значно розширює можливості розробників, дозволяючи підключати як цифрові, так і аналогові датчики. Крім того, він забезпечує сервоуправління для відкриття або закриття штор і бездротовий зв'язок з мобільним додатком через платформу Blynk для дистанційного керування.

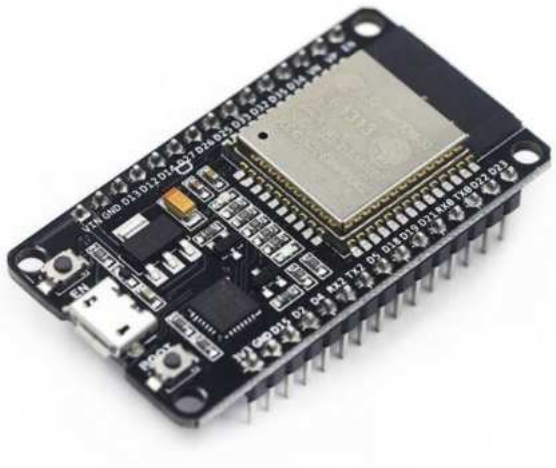


Рисунок 2.1 – Мікроконтролер ESP32 [28]

Історія ESP32 тісно пов'язана з успіхом його попередника, ESP8266.ESP8266 був випущений в 2014 році. Він завоював популярність завдяки низькій вартості та вбудованим можливостям Wi-Fi. Спочатку використовувався як Wi-Fi-модем для мікроконтролерів, таких як Arduino, але незабаром спільнота розробників усвідомила його потенціал як самостійного мікроконтролера. Його поширенню значно сприяла наявність відкритого прошивання, такого як NodeMCU, та сумісність з Arduino [33].

ESP32 (2016): Espressif випустила ESP32 як більш потужний і багатофункціональний наступник. Він включав двоядерний процесор, більше GPIO-контактів, збільшену пам'ять і вбудований Bluetooth (Classic і Low Energy). Espressif також надала вичерпну документацію та інструменти для розробки, включаючи менеджер плат Arduino IDE. Ці внески допомогли ESP32 завоювати популярність серед виробників і експериментаторів.

З моменту першого випуску сімейство ESP32 розширилося і тепер включає різні лінійки продуктів і моделі, включаючи ESP32-S3 і ESP32-C3. Кожна модель має унікальні функції і можливості, що відповідають різним потребам застосування.

ESP32 став одним із основниз інструментів Інтернету речей (IoT) завдяки своїй універсальності, низькій вартості і вбудованій бездротовій зв'язності. Його ключові особливості роблять його дуже придатним для широкого спектру застосувань IoT. Наприклад, вбудований Wi-Fi та Bluetooth: Вбудований Wi-Fi (802.11 b/g/n) дозволяє пристроям підключатися до існуючих мереж або створювати власні точки доступу. Дворежимний Bluetooth (v4.2 BR/EDR та BLE) забезпечує зв'язок з широким спектром Bluetooth-пристроїв та додатків з низьким енергоспоживанням [42].

Більшість моделей ESP32 оснащені двоядерним 32-бітним мікропроцесором LX6, що забезпечує значну обчислювальну потужність для виконання складних завдань і багатозадачності, що є критично важливим для обробки різних протоколів IoT і даних. ESP32 пропонує широкий набір GPIO-контактів та інтерфейсів, включаючи ADC, DAC, SPI, I2C, UART, PWM та сенсори дотику, що забезпечує безперебійну взаємодію з різними сенсорами, виконавчими механізмами та іншими периферійними пристроями, які зазвичай використовуються в пристроях IoT.

Розроблений з урахуванням ефективності, ESP32 має різні режими енергозбереження, що робить його придатним для пристроїв IoT, що працюють від батарей, та додатків, що вимагають тривалого терміну експлуатації. ESP32 має вбудовані апаратні функції безпеки, такі як безпечний запуск, шифрування флеш-пам'яті та криптографічні прискорювачі (AES, SHA, RSA, ECC), необхідні для захисту конфіденційних даних у підключених пристроях. Завдяки вбудованій оперативній та постійній пам'яті, а також підтримці зовнішньої флеш-пам'яті, ESP32 може обробляти код додатків, зберігати дані та оновлювати прошивку бездротовим способом (OTA), що є життєво важливим для управління та обслуговування розгорнутих пристроїв IoT. Espressif надає ESP-IDF (IoT Development Framework), комплексний SDK з бібліотеками та інструментами для розробки додатків IoT. ESP32 також широко підтримується Arduino IDE та іншими середовищами розробки, що робить його доступним для широкого кола розробників.

В Україні, як і в багатьох інших частинах світу, ESP32 є популярним вибором як для любителів, так і для професіоналів, які розробляють інноваційні рішення для Інтернету речей у різних галузях. Доступна ціна та набір функцій роблять його чудовою платформою для створення прототипів та розгортання підключених пристроїв.

Датчики LDR (рис. 2.1) реагують на зміну освітленості в приміщенні, тому штори відкриваються або закриваються автоматично залежно від умов. Вони використовуються в якості сенсорних елементів, що дозволяють системі автоматично реагувати на зміни зовнішнього освітлення. У них проста конструкція і невисока вартість, що робить їх ідеальними для проектів автоматизації будинку. Використання двох світлорезисторів дозволяє підвищити точність аналізу освітлення і виявляти зміни освітлення, що відбуваються в різних напрямках (наприклад, зі сходу або заходу кімнати). Для зчитування значень фоторезисторів використовуються дільники напруги з резисторами 10 кОм, щоб забезпечити стабільну і точну передачу даних на аналогові входи ESP32 [28].

Модуль складається з фоторезистора, який змінює свій електричний опір залежно від інтенсивності падаючого світла. Чим яскравіше світло, тим нижчий опір фоторезистора, і навпаки. Має два виходи: аналоговий (AO) і цифровий (DO). Аналоговий вихід видає напругу, пропорційну інтенсивності світла. Цифровий вихід порівнює аналоговий сигнал з пороговим значенням, встановленим потенціометром, і видає логічний рівень (HIGH або LOW). Цей модуль використовується в різних проектах, пов'язаних з управлінням освітленням, таких як автоматичне вмикання/вимикання освітлення, системи контролю рівня освітлення, робототехніка, а також проекти Arduino і Raspberry Pi. Він простий в підключенні і використанні, що робить його відмінним вибором для всіх, від початківців до досвідчених розробників.

Рисунок 2.2 – Датчики LDR [44]

Сервопривід SG90 (рис 2.2) був обраний за його компактність, точність позиціонування і простоту підключення до цифрового виводу ESP32. Це електромеханічний пристрій, що використовується для точного контролю положення, швидкості та сили механізму. Цей механізм приводу може повертатися під певним кутом відповідно до вхідного сигналу, що підходить для нашої системи, адже нам потрібно привести в дію механізм руху штор. Таким чином можна точно регулювати положення жалюзі або ролет від повністю відкритого до повністю закритого положення. Оскільки в більшості випадків достатньо просто повернути вісь або важіль на 180°, сервопривід є ідеальним виконавчим механізмом для таких завдань. Він складається з двигуна, датчиків та системи управління, що дозволяє підтримувати задані значення цих параметрів. Всі ці датчики працюють в зв’язці, щоб забезпечити виконання задач, що поставлена на сервопривід. Він керує положенням штори механічно за допомогою ШІМ-сигналу, що генерується мікроконтролером. Нижче можна побачити зображення цього пристрою, а конкретно популярної моделі SG90, яку часто використовують в подібних проєктах та системах.

Рисунок 2.3 – Сервопривід SG90 [44]

Система живиться від стабілізованого джерела напруги 5 В, яке відповідає вимогам всіх задіяних компонентів. Всі з'єднання виконуються за допомогою кабелів-перемичок на макетній платі, що спрощує монтаж і дозволяє вносити зміни в конструкцію під час тестування. Для ESP32, сервоприводів і фоторезисторної системи управління шторами дуже важливо вибрати оптимальне джерело живлення, яке забезпечить стабільну роботу пристрою, мінімальні перешкоди і зручність використання. Для живлення таких систем існує кілька основних типів, кожен з яких має свої переваги та недоліки.

Перший варіант – живлення змінним струмом від джерела живлення (адаптера) 5 В або 12 В, що є найпоширенішим і найстабільнішим рішенням для стаціонарних установок. Це забезпечує безперебійну роботу без необхідності заміни або заряджання батарей, дозволяє використовувати більш потужні компоненти та зменшує ризик перебоїв у роботі. Основним недоліком є залежність від доступності джерела живлення та кабельного з'єднання, що може ускладнити установку та зменшити мобільність пристрою. Змінний струм – дешевий і

практичний варіант, проте зовсім незручний через наявність дротів та потсійної потреби в доступі до електромережі.

Другий варіант – автономне джерело живлення за допомогою акумуляторних батарей (літій-іонних або літій-полімерних). Цей метод дозволяє встановлювати систему в місцях, де немає доступу до електромережі, що збільшує мобільність і спрощує установку. Однак батареї вимагають регулярного контролю рівня заряду, періодичного заряджання або заміни, а також використання енергозберігаючої системи в програмному забезпеченні (наприклад, режим сну ESP32) для продовження часу роботи. Це відносно оптимізований метод серед усіх інших, проте він не буде обраний для даного проєкту.

Третій варіант – сонячна енергія з додатковою батареєю. Це рішення характеризується екологічністю та енергоефективністю, що робить його ідеальним для віддалених місць і дозволяє значно збільшити автономність системи. Однак для ефективної роботи системи необхідно встановити високоякісний контролер заряду та виділити достатньо місця для встановлення сонячних панелей. Тому це буде максимально дорогий варіант, який також потребує створення окремої міні-системи для заряджання та підтримки контролю енергії. В сьогоденні маємо тренд на екологічність, тому, скоро багато систем будуть мати подібне живлення, незалежно від інших характеристик. Цей варіант також не підійде для системи, адже це найдорожчий метод серед усіх.

Для даного проєкту було обрано перший варіант через простоту імплементації. Також вирішальну роль грає простота демонстрації та роботи системи. Якщо аналізувати оптимальний варіант для такої системи – це акумуляторні батареї, адже немає ніяких дротів, а також присутнє оптимальне живлення для роботи. З мінусів – потреби в зарядці та контроль живлення, проте це все можна інтегрувати в додаток (наприклад контроль заряду). Загалом, на ринку інснують різні віріанти таких систем, проте кожна має свої недоліки й переваги. Детальніше порівняння наведено в таблиці 2.1.

Таблиця 2.1 – Методи живлення системи керування шторами/жалюзі

Метод живлення	Переваги	Недоліки	Застосування
Мережевий адаптер 5В/12В	Стабільна та безперервна робота, не вимагає обслуговування	Залежність від наявності електромережі, потребує прокладки кабелів	Стаціонарні системи в будинку чи офісі
Акумуляторні батареї	Мобільність, незалежність від мережі	Обмежений час роботи, потреба в зарядці, контроль заряду	Віддалені або тимчасові об’єкти
Сонячна батарея + акумулятор	Екологічність, автономність	Висока вартість, залежність від сонячного освітлення, потребує контролера заряду	Віддалені об’єкти без доступу до мережі

Мобільний додаток Blynk використовується для ручного керування жалюзі через користувацький інтерфейс смартфона (рис 2.4) [11]. З'єднання між ESP32 і додатком здійснюється через Wi-Fi, що забезпечує стабільний канал зв'язку в локальній мережі. Головною перевагою Blynk є те, що з його допомогою можна легко створити користувальницький інтерфейс без необхідності розробки складних мобільних додатків з нуля. За допомогою візуального інструменту проектування можна швидко додавати різні віджети: кнопки для відкриття/закриття жалюзі, повзунки для точного налаштування сервомотора, а також індикатори стану або графіки для відображення даних датчиків. Інтерфейс користувача доступний на смартфонах Android та iOS.

У проекті ESP32 він підключається до локальної мережі Wi-Fi і через інтернет-з'єднання спілкується з серверами Blynk. Мікроконтролер через додаток отримує команди від користувача і відповідно до них керує сервомоторами, змінюючи положення штор. Крім того, ESP32 може надсилати до додатка інформацію про поточний стан системи, наприклад, про яскравість світла, виміряну датчиками LDR. Це дозволяє здійснювати автоматичне та ручне керування.

Використання Blynk значно спрощує реалізацію проекту, оскільки не вимагає розробки спеціального мобільного додатка, що скорочує час і ресурси, необхідні для створення користувальницького інтерфейсу. Додаток підтримує зручну аутентифікацію, що гарантує безпечне управління пристроями, а також надає можливість налаштування сповіщень та інших корисних функцій.

Таким чином, інтеграція Blynk в систему управління шторами/жалюзі робить проект більш доступним для кінцевих користувачів і забезпечує гнучке, зручне та надійне дистанційне керування за допомогою мобільного телефону.

Рисунок 2.4 – Інтерфейс додатку Blynk [11]

2.2 Архітектура та принцип функціонування системи керування жалюзі

Система управління розумними шторами/жалюзі на базі мікроконтролера ESP32 з пультом дистанційного керування за допомогою мобільного додатку Blynk призначена для автоматизації освітлення та забезпечення комфорту в житлових і комерційних приміщеннях. Система інтегрує апаратні компоненти з програмним забезпеченням, дозволяючи автоматично та дистанційно керувати механізмами штор.

Апаратне забезпечення складається з таких основних компонентів та пристроїв:

− мікроконтролер ESP32, який працює як центральний процесор, контролює зв'язок з датчиками та сервоприводом і забезпечує зв'язок з мобільним додатком за допомогою Wi-Fi;

− LDR (світлочутливий резистор) світлочутливі датчики, які вимірюють інтенсивність світла і передають аналогові сигнали на ESP32 для аналізу умов навколишнього середовища;

− сервопривід типу SG90, який відповідає за фізичний рух штор або жалюзі та точно позиціонує їх відповідно до команд контролера;

− блок живлення, який забезпечує стабільну подачу струму до всіх компонентів системи.

Компоненти програмного забезпечення, яке у висновку буде відображати користувацький інтерфейс та управління системою включає: програмне забезпечення, розроблене для мікроконтролера ESP32, яке обробляє сигнали датчиків LDR, виконує логіку управління сервоприводом і зв'язується з сервером Blynk для отримання команд користувача і передачі стану системи; мобільний додаток Blynk, що забезпечує інтуїтивно зрозумілий інтерфейс для дистанційного керування шторами та перегляду поточного стану та налаштувань системи. Дані компоненти допоможуть керувати системою.

Основні етапи роботи системи керування жалюзі через мобільний додаток Blynk виглядатиме наступним чином:

− ініціалізація: при включенні живлення мікроконтролер ESP32 ініціалізує всі периферійні пристрої, зокрема датчики LDR і сервоприводи, встановлює Wi-Fi-з'єднання і підключається до серверів Blynk для обміну командами;

− автоматичне керування: ESP32 регулярно зчитує показання світлових датчиків і, на основі встановлених граничних значень, визначає бажане положення жалюзі, а потім генерує сигнали, необхідні для керування сервоприводами, щоб регулювати положення жалюзі;

− дистанційне керування: за допомогою мобільного додатку Blynk користувач може в будь-який момент дати команду відкрити, закрити або встановити жалюзі в проміжне положення. Отримана команда надсилається на ESP32, який керує відповідним сервоприводом;

− моніторинг: система надсилає інформацію про поточний стан (положення жалюзі, рівень освітлення) на додаток, щоб користувач міг у реальному часі стежити за роботою системи;

− безпека та стабільність: програмне забезпечення містить механізми повідомлення про помилки, які запобігають порушенням роботи сервоприводу та гарантують безперебійну роботу системи навіть у разі тимчасового переривання з'єднання.

Після аналізу архітектури та приципу функціонування системи, отримуємо просту схему роботи (рис 2.5), який представляє основні пристрої та взаємодію між ними. В основному, «мозок» системи – ESP32 обробляє інформацію із датчиків виявлення світла та передає вказівки сервоприводу, який в свою чергу відкриває/закриває жалюзі або штори. З допомогою додатку Blynk можна буде детальніше налаштовувати специфіку роботи, розклади відкривання чи закривання штор, чутливість сенсорів та інші параметри. Це проста система яка спрощує щоденне використання домашнього текстилю, адже не потрібно навіть фізично взаємодіяти із жалюзі/шторами по кілька разів на день.

Рисунок 2.5 – Процес взаємодії пристроїв в системі

2.3. Застосування протоколу MQTT та інтеграція з Blynk

MQTT (Message Queue Telemetry Transport) — це легкий протокол публікації та підписки в мережі, який завдяки своїй простоті, ефективності та низькому навантаженню на мережу широко використовується в системах Інтернету речей (IoT) [31]. Цей протокол розроблений спеціально для нестабільних мережевих середовищ, де ресурси клієнтів і каналів передачі даних обмежені. MQTT використовує модель «агент-клієнт», в якій всі клієнтські пристрої підключені до центрального агента і обмінюються повідомленнями через нього. У цьому процесі клієнт може виступати в ролі видавця (publisher), передплатника (subscriber) або виконувати обидві функції одночасно. Повідомлення передаються у вигляді текстових або бінарних пакетів з зазначеною темою (topics), що забезпечує ефективну маршрутизацію інформації між вузлами системи.

Інтеграція протоколу MQTT в проект інтелектуальної системи управління шторами/жалюзі на базі ESP32 дозволяє реалізувати централізований, розширюваний і асинхронний обмін даними між пристроями та користувачами або іншими службами. У цьому контексті мікроконтролер ESP32 виступає в ролі клієнта MQTT, встановлюючи з'єднання з зовнішнім проксі-сервером MQTT (наприклад, Mosquitto або HiveMQ) через Wi-Fi. Система реалізує кілька каналів обміну повідомленнями: передача поточного стану жалюзі (наприклад, «відкрито» або «закрито»), прийом команд зміни положення та обробка аварійних або сервісних сигналів. За допомогою протоколу MQTT можна побудувати більш складну архітектуру IoT, в якій ESP32 взаємодіє з іншими пристроями розумного будинку (наприклад, системою освітлення, системою регулювання температури тощо) для повної автоматизації середовища. На рисунку 2.6 нижче показана структура системи управління шторами, яка використовує протокол обміну повідомленнями MQTT для встановлення зв'язку між клієнтським додатком на смартфоні та виконавчим пристроєм на базі мікроконтролера ESP32.

У системі смартфон користувача працює як клієнт MQTT і видає команди керування шторами. Команди, такі як «відкрити», «закрити» або «встановити певне положення», надсилаються до певної теми MQTT. Основним елементом, що забезпечує маршрутизацію повідомлень між клієнтами, є MQTT-проксі, розташований в Інтернеті. ESP32 також працює як MQTT-клієнт, підписується на ту саму тему і чекає на надходження команд управління. ESP32 встановлює бездротове з'єднання з Інтернетом через Wi-Fi-роутер.

Мікроконтролер може аналізувати рівень освітлення і самостійно визначати положення штор. Пряме управління шторами здійснюється за допомогою сервоприводу, де сигнали управління генеруються ESP32 відповідно до отриманих команд MQTT або даних датчика освітлення. Таким чином, ця система використовує надійний і ефективний протокол MQTT для обміну даними, що забезпечує гнучке і інтегроване управління ручним і автоматичним режимом штор.

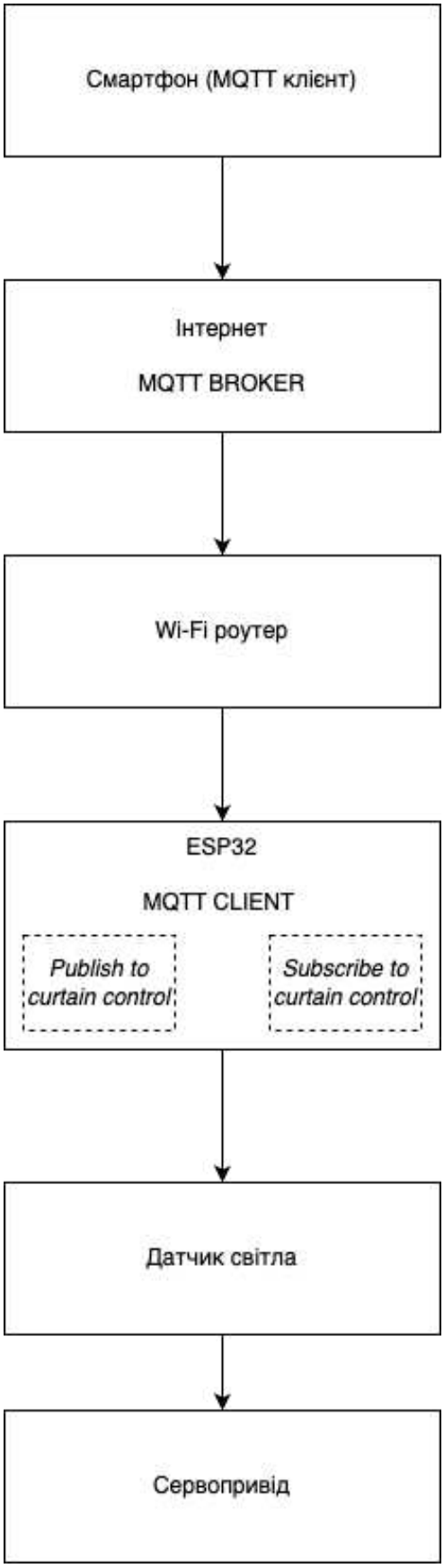


Рисунок 2.6 – Проста блокова діаграма протоколу MQTT

У цьому проєкті особливо важливою є інтеграція MQTT з мобільним додатком Blynk, який дозволяє реалізувати інтуїтивно зрозумілий інтерфейс користувача для дистанційного моніторингу та керування пристроями. Blynk підтримує власний хмарний сервер, а також може інтегруватися з MQTT-проксі, що забезпечує гнучкість побудови архітектури системи. У типовому варіанті реалізації ESP32 підключається як MQTT-клієнт і публікує повідомлення на певну тему, які зчитуються мобільним додатком Blynk. Зворотний зв'язок відбувається таким самим чином: користувач взаємодіє з віджетами додатка, які формують повідомлення MQTT і надсилають їх до теми, підписаної мікроконтролером.

Використання протоколу MQTT (Message Queue Telemetry Transport) для розробки систем розумного будинку з мобільним додатком Blynk, особливо для управління пристроями на базі ESP32, має ряд значних переваг, що робить його надзвичайно ефективним і зручним рішенням. MQTT — це легкий протокол обміну повідомленнями, розроблений спеціально для пристроїв з обмеженими ресурсами та нестабільним підключенням. Його базова архітектура базується на принципі «публікація-підписка», що дозволяє пристроям надсилати та отримувати повідомлення через центральний проксі-сервер без необхідності встановлення постійного прямого з'єднання між усіма учасниками системи [32].

Однією з головних переваг MQTT є надзвичайно низьке споживання трафіку, що є дуже важливим для пристроїв, що працюють від батарей або в мережах з низькою пропускною здатністю. Завдяки компактному заголовку повідомлень та ефективній структурі, передача даних відбувається швидко, а навантаження на мікроконтролер є мінімальним. У середовищі Blynk MQTT забезпечує майже миттєву реакцію на дії користувачів у мобільних додатках. Коли стан перемикача або кнопки змінюється, відповідна команда негайно надсилається на пристрій. Це має вирішальне значення для доступності та ефективності управління.

Ще однією важливою перевагою є надійність. MQTT підтримує три рівні якості обслуговування (QoS), і розробники можуть вибрати необхідний рівень гарантії передачі повідомлень відповідно до вимог надійності системи. Це

запобігає втраті важливої інформації, наприклад, команд відкриття або закриття жалюзі чи дверей, або повідомлень про зміну стану датчиків.

У поєднанні з Blynk, MQTT легко інтегрується в мобільний інтерфейс без складної логіки клієнт-сервер. Розробники можуть створювати інтерфейси управління за допомогою зручних віджетів Blynk, зосередившись на функціональності проекту, а MQTT-проксі надійно обробляє обмін даними. В таблиці 2.1 подане порівняння з іншими методами обміну даними [30]:

Таблиця 2.2 – Порівняння методів обміну даними в IoT системах

Критерій	MQTT в Blynk	HTTP-запити / REST API	WebSockets / TCP-з'єднання
Швидкість передавання даних	Висока (майже реального часу)	Нижча (особливо при періодичних запитах)	Висока
Енергоефективність	Висока (підходить для пристроїв з живленням від батарей)	Низька (часті запити = більше споживання)	Середня
Підтримка зворотного зв'язку	Має вбудовану підтримку підписки/публікації	Односторонній (зазвичай від пристрою до сервера)	Потрібна реалізація вручну
Масштабованість	Висока (працює через брокер, легко додавати пристрої)	Обмежена через зростання запитів	Середня (вимагає керування підключеннями)

Кінець таблиці 2.2

Простота інтеграції в Blynk	Повна інтеграція (автоматичне підключення до серверів Blynk)	Часткова (через Webhooks)	Потребує ручної реалізації

Інтеграція Blynk і MQTT забезпечує високу надійність, масштабованість і простоту використання системи. Завдяки такому підходу система може працювати не тільки в локальній мережі, але й бути доступною через Інтернет без складної мережевої конфігурації. Крім того, поєднання Blynk і MQTT дозволяє зберігати журнали подій, встановлювати автоматичні сценарії (наприклад, відкривати жалюзі в певний час або при досягненні певної яскравості) і інтегруватися з іншими сервісами Інтернету речей через хмарний API.

2.4. Електричні характеристики системи

Інтелектуальна система автоматичного керування базується на мікроконтролері ESP32 і складається з низки апаратних компонентів, які збирають інформацію з навколишнього середовища, логічно обробляють сигнали та виконують фактичні заходи керування. Ця система базується на мікроконтролері ESP32, який є потужним мікропроцесором із вбудованими функціями Wi-Fi та Bluetooth. Цей пристрій може зчитувати аналогові та цифрові вхідні сигнали, а також здійснювати бездротовий зв'язок зі смартфоном через мережу. Пристрій вимагає робочої напруги 3,3 В, тому його необхідно підключити до стабільного джерела живлення.

Два фоторезистори (LDR) вимірюють інтенсивність світла. Кожен фоторезистор поєднаний з резистором 10 кОм, утворюючи розподільник напруги. Один вихід LDR підключається до напруги 3,3 В, інший вихід підключається через резистор до загального заземлення (GND), а точка між двома виходами

підключається до аналогового входу ESP32. Така конфігурація дозволяє мікроконтролеру визначати зміну опору LDR залежно від освітлення, що полегшує прийняття рішення про відкриття або закриття штор. Фактичний рух штор здійснюється за допомогою сервомотора SG90. Цей мотор є невеликим електромеханічним пристроєм, який керує механізмом відповідно до імпульсного сигналу (PWM) мікроконтролера. Сервомотор потребує живлення 4,8-6,0 В, тому підключається безпосередньо до зовнішнього джерела живлення 5 В.

Компоненти системи розташовані на експериментальній платі, що дозволяє легко зібрати прототип без зварювання. З'єднання між компонентами здійснюється за допомогою з'єднувальних кабелів. Джерелом живлення служить блок живлення, зафіксований або працюючий на 5 В (вольт). Цей блок живлення підключається до відповідних вхідних клем ESP32 і вхідних клем сервомотора.

Підсумовуючи вищеописане, отримуємо електричну схему для розробленої системи (рис 2.7):

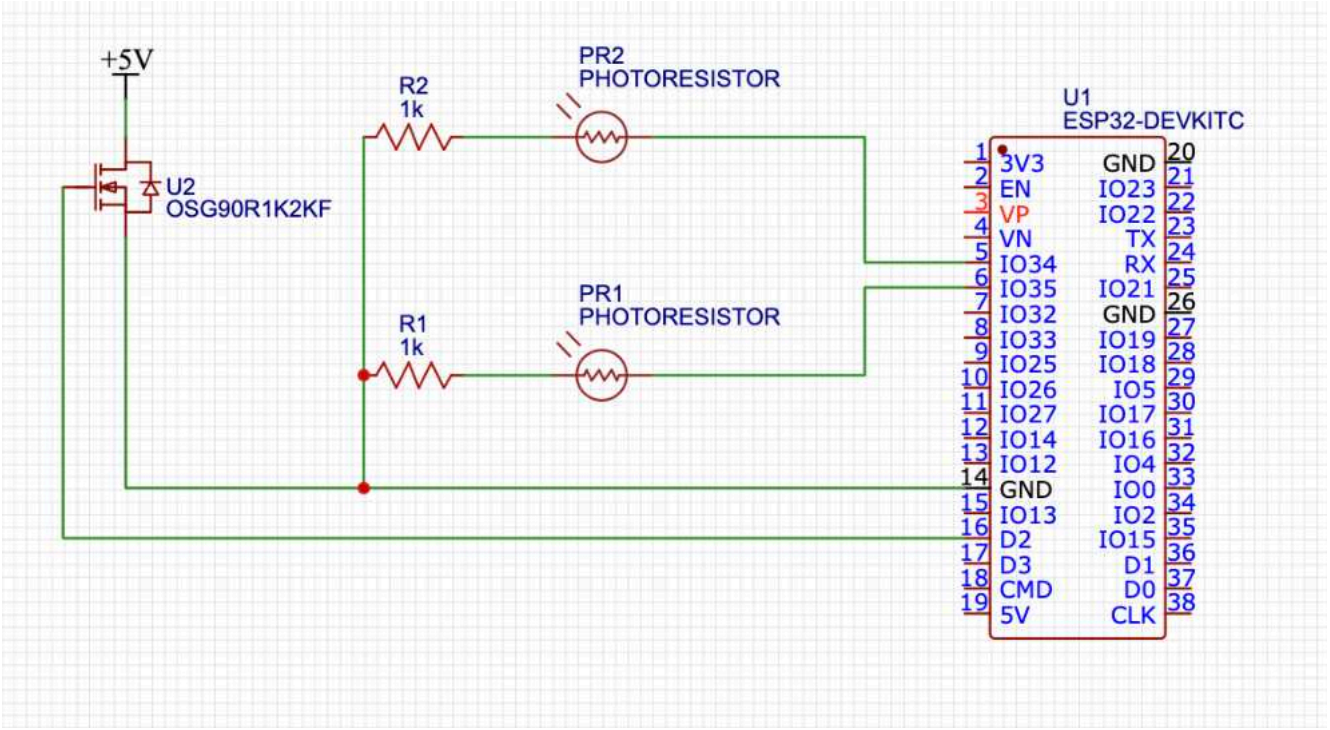


Рисунок 2.7 – Електрична схема

Система керується за допомогою мобільного додатка Blynk, встановленого на смартфоні. З'єднання Wi-Fi полегшує комунікацію між ESP32 і хмарним сервером Blynk, що дозволяє передавати команди з інтерфейсу користувача на реальний пристрій. Це дозволяє користувачеві дистанційно керувати відкриттям або закриттям штор і налаштовувати параметри автоматизації залежно від рівня освітлення. Всі ці компоненти інтегровані в єдину систему, що працює на основі логіки, вбудованої в прошивку мікроконтролера, що забезпечує надійність, зручність і масштабованість системи управління розумними шторами в будинку.

Характеристики основних компонентів, такі як з’єднання, вимоги до напруги та загальні коментарі щодо їх роботи та ролі в системі керування шторами або жалюзі наведені в таблиці 2.1 нижче:

Таблиця 2.3 – Характеристики компонентів

Компонент	З’єднання	Вимоги до напруги	Коментарі
ESP32	VIN, GND, GPIO (для LDR, сервоприводу)	3.0–3.6 В (типово 3.3 В)	Основний контролер, забезпечує обчислення, підключення до Wi-Fi, обробку сигналів LDR та керування сервоприводом.
Фоторезистор LDR	Один контакт до 3.3 В, інший до GPIO через резистор 10 кОм і GND	Не потребує живлення напряму	Формує подільник напруги; змінює опір залежно від рівня освітлення.
Резистор 10 кОм	Послідовно з LDR (між GPIO та GND)		Використовується в парі з LDR для формування сигналу

Кінець таблиці 2.3

Сервопривід SG90	PWM-сигнал до GPIO ESP32, живлення до 5 В і GND	4.8–6.0 В	Забезпечує механічне відкривання/закривання штор; потребує окремого живлення через стабілізоване джерело.
Джерело живлення 5 В	VIN ESP32, сервопривід	5.0 В	Забезпечує стабільне живлення мікроконтролера та сервоприводу.

Інтеграція апаратних компонентів вимагає глибокого розуміння функціональних характеристик кожного компонента та дотримання стандартів електротехніки.

Першим етапом цього процесу є встановлення електричного з'єднання між оптичним датчиком і мікроконтролером. У представленій конфігурації кожен модуль використовує фоторезисторний модуль з аналоговими та цифровими виходами. Для збору даних про інтенсивність освітлення в широкому діапазоні необхідно підключити аналоговий вихід (AO) фоторезистора до аналогового входу (ADC) ESP32. Розташування аналогових входів залежить від моделі ESP32. Тому необхідно уважно прочитати технічну документацію (схему роз'ємів), щоб визначити відповідні контакти. Цифровий вихід (DO) фоторезистора надає бінарну інформацію про рівень освітлення відносно заданого порогу. Він підключений до цифрових входів/виходів (GPIO) ESP32, що дозволяє легко визначити, чи достатнє освітлення.

Другим важливим кроком є інтеграція сервомотора, який є виконавчим механізмом, відповідальним за фізичний рух. Сервомотори зазвичай живляться від джерела живлення 5 В постійного струму, і споживання струму сервомотора,

особливо під навантаженням, може перевищувати вихідну потужність виводів живлення мікроконтролера. Це може призвести до нестабільності системи або пошкодження компонентів. Якщо ви використовуєте дискретне джерело живлення, обов'язково встановіть спільну точку заземлення (GND) між джерелом живлення та платою ESP32. Це необхідно для забезпечення сумісності з рівнем керуючих сигналів. Сигнал управління сервоприводом (зазвичай PWM — імпульсно-широтно-модульований) підключається до одного з цифрових виходів ESP32, який підтримує генерацію PWM-сигналів. На останньому етапі сама плата ESP32 підключається до джерела живлення. Це можна зробити за допомогою USB-роз'єму або через виходи живлення.

2.5 Висновки до другого розділу

У розділі 2 було проаналізовано та детально визначено технічні засоби та програмне забезпечення апаратних і програмних підсистем, що складають основу автоматизованої системи управління шторами. Були досліджені можливі способи взаємодії цих підсистем з урахуванням забезпечення надійної та енергоефективної роботи в житлових приміщеннях. В аналіз включено функціональні можливості основних апаратних модулів, їх призначення, переваги та обмеження, а також історичний розвиток відповідних технологій.

На основі технічних характеристик обраних компонентів було обґрунтовано їх придатність для використання в проєктованій системі. Були враховані вимоги щодо точності, енергоефективності, сумісності з іншими елементами системи, надійності та стабільності в реальних умовах експлуатації. Особлива увага була приділена електричним характеристикам компонентів, які безпосередньо впливають на безпеку та стабільність роботи системи при використанні в житлових приміщеннях.

Центральним апаратним елементом системи є мікроконтролер ESP32, який працює як обчислювальний модуль. Він відповідає за зчитування та обробку

сигналів від датчиків – в даному випадку фоторезисторів типу LDR, які реагують на зміни рівня освітлення. На основі отриманих даних ESP32 формує логічні команди і надсилає їх на виконавчий пристрій – сервомотор SG90, який безпосередньо відкриває або закриває жалюзі. Додаткові компоненти, що забезпечують стабільну роботу системи, – стабілізований джерело живлення 5 В, резистори для формування розподільника напруги та обмеження струму, платформа для монтажу схеми та з'єднувальні дроти. Всі ці елементи разом утворюють повну апаратну платформу для реалізації системи автоматичного управління.

Архітектура системи має модульну структуру, що базується на взаємодії трьох основних блоків: сенсорної частини (LDR-сенсори), комп'ютерного модуля (ESP32) та користувацького інтерфейсу (мобільний додаток). Принцип полягає в тому, що світло вимірюється безперервно, виміряні значення аналізуються, приймаються рішення про положення штор і надсилається команда механізму, який її виконує. Крім автоматичного режиму, штори можна керувати вручну за допомогою мобільного додатка, що працює на платформі Blynk.

З огляду на особливості використання платформи Blynk та її можливості, при розробці програмного забезпечення особлива увага була приділена інтеграції з протоколом MQTT, який використовується для організації обміну даними між мікроконтролером та мобільним додатком. MQTT забезпечує надійну, просту і швидку передачу повідомлень навіть при низькій швидкості або нестабільному інтернет-з'єднанні.

В результаті другого етапу роботи було створено стабільну технічну та функціональну базу для запланованої системи. Було чітко описано архітектуру, структуру взаємодії програмних і апаратних модулів, перевірено сумісність компонентів і можливість ефективної роботи з використанням сучасних рішень IoT. Досягнуті результати стали основою для переходу до наступного етапу – розробки алгоритмів управління, мікропрограмування та створення фізичного прототипу системи.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ КЕРУВАННЯ ШТОРАМИ НА БАЗІ ESP32

3.1 Налаштування Arduino IDE та Blynk

Щоб імплементувати нашу розробку, будемо використовувати Arduino IDE, адже це важливий інструмент для реалізації вбудованих та інтернет-проектів, що дозволяє швидко створювати, тестувати та налаштовувати програмне забезпечення мікроконтролерів. Активна спільнота, багаті бібліотеки та гнучкість у використанні роблять його найкращим вибором для навчальних закладів, любителів та професійних програмістів.

Arduino IDE (Integrated Development Environment) — це офіційне середовище розробки для платформ Arduino та безлічі сумісних мікроконтролерів, включаючи ESP32. Це програмне забезпечення полегшує написання, компіляцію та завантаження коду на мікроконтролер, що використовує мову програмування на основі C/C++. Програмне забезпечення має простий інтерфейс, доступність для новачків та широкі можливості для професійних розробників. Arduino IDE сумісний з основними операційними системами, включаючи: Розглядаються операційні системи Windows, macOS та Linux.

Для встановлення Arduino IDE необхідно спочатку завантажити інсталяційний файл з офіційного веб-сайту Arduino (www.arduino.cc) (рис 3.1). Важливо вибрати версію, що відповідає вашій операційній системі. Після завантаження файл запускається для інсталяції у звичайному режимі. В операційній системі Windows процес інсталяції супроводжується майстром інсталяції. Натомість у macOS додаток перетягується до папки «Програми». У Linux інсталяцію можна виконати з архіву або за допомогою менеджера пакетів. Після завершення процесу інсталяції користувач отримує доступ до інтерфейсу розробки, де може розпочати створення, редагування та компіляцію програм.

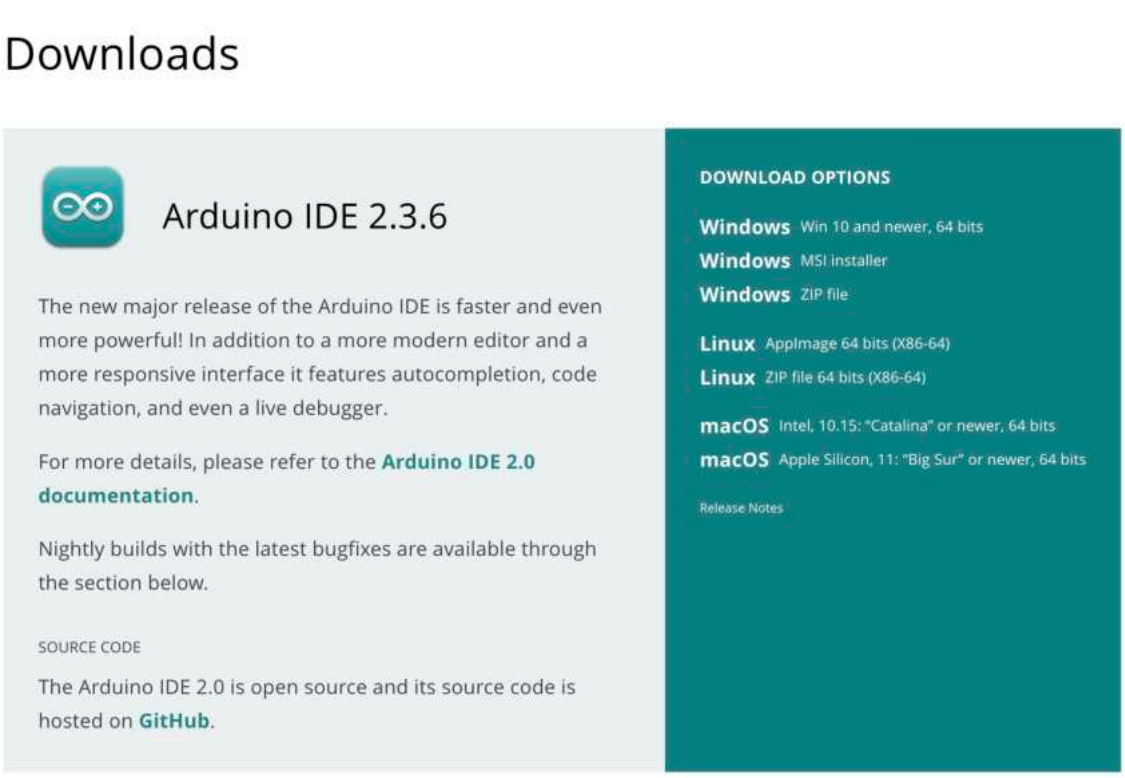


Рисунок 3.1 – Сторінка завантаження Arduino IDE

Для роботи з мікроконтролерами ESP32, окрім стандартних процедур, необхідно налаштувати Arduino IDE. У розділі налаштувань необхідно додати посилання на диспетчер плат Espressif. Після встановлення підтримки ESP32 за допомогою вбудованого диспетчера процес завершено. Після встановлення нових плат користувач може вибрати їх модель, наприклад ESP32 DevKitC, підключити їх через USB і розпочати програмування. Процес завантаження програм на мікроконтролер запускається натисканням відповідної кнопки «Завантажити». Після цього інтегроване середовище розробки (IDE) автоматично компілює код і зберігає його в пам'яті пристрою. У середовищі також доступна послідовна консоль для моніторингу результатів роботи програми.

Інструмент програмування Arduino IDE має простий, але потужний інтерфейс, який підходить як для початківців, так і для досвідчених розробників. Він підтримує базове підсвічування синтаксису в мові програмування C/C++ і пропонує автоматичне доповнення коду (у нових версіях IDE), миттєве виявлення помилок, форматування коду та навігацію по функціях. Це значно спрощує процес

написання програм та функцій і дозволяє швидко орієнтуватися в структурі проєкту (рис 3.2).

У верхній частині редактора знаходиться панель інструментів, з якої можна виконувати основні дії: компілювати ескізи (перевіряти на наявність помилок), завантажувати на плату розробки, відкривати нові проекти, зберігати файли, відкривати послідовний монітор для перегляду повідомлень плати розробки тощо. Крім того, є консоль, яка відображає результати компіляції, помилки та стан завантаження прошивки. Редактор дозволяє працювати з декількома вкладками або файлами одночасно в одному проекті. Це дуже корисно при створенні більш складних додатків, оскільки код у цих додатках організований у модулі.

Також важливі бібліотеки, що підтримують середовище – бібліотеки є незалежними модулями коду, які можна підключити до проектів для розширення їх функціональності. Наприклад, у випадку ESP32 для інтеграції з Blynk використовуються бібліотеки BlynkSimpleEsp32.h та ESP32Servo.h. Усі ці бібліотеки можна встановити за допомогою диспетчера бібліотек або вручну.

Загалом, редактор коду Arduino IDE є практичним інструментом для розробки прошивки, оскільки він простий, гнучкий і має активну підтримку спільноти. З його допомогою ви можете швидко розпочати створення систем IoT, наприклад, автоматизованої системи жалюзі з мікроконтролером ESP32 і хмарним сервісом Blynk.

Рисунок 3.2 – Інтерфейс Arduino IDE

Далі переходимо до Blynk. Для коректної роботи потрібно зареєструватися і налаштувати роботу в середовищі Arduino IDE.

Щоб розпочати роботу з платформою Blynk, перейдіть на офіційний веб-сайт: https://blynk.io/. Потім натисніть кнопку «Sign Up», щоб створити новий обліковий запис. Під час реєстрації потрібно ввести дійсну адресу електронної пошти, на яку буде надіслано підтвердження. Щоб активувати обліковий запис, потрібно натиснути посилання в цьому листі та встановити пароль для доступу до платформи. Після завершення реєстрації система надасть короткий посібник з основних функцій Blynk IoT та навігації по інтерфейсу.

Після створення облікового запису можна приступити до налаштування проєкту. Для реалізації інтелектуальної системи управління жалюзі на базі мікроконтролера ESP32 необхідно виконати наступні дії. У веб-інтерфейсі Blynk створіть новий шаблон (New Template) і виберіть тип обладнання ESP32. Потім введіть назву шаблону і короткий опис (рис 3.3). Після збереження шаблону буде створено унікальний Template ID, Device Name та Auth Token. Ці параметри потрібно ввести в програмний код мікроконтролера в середовищі Arduino IDE.

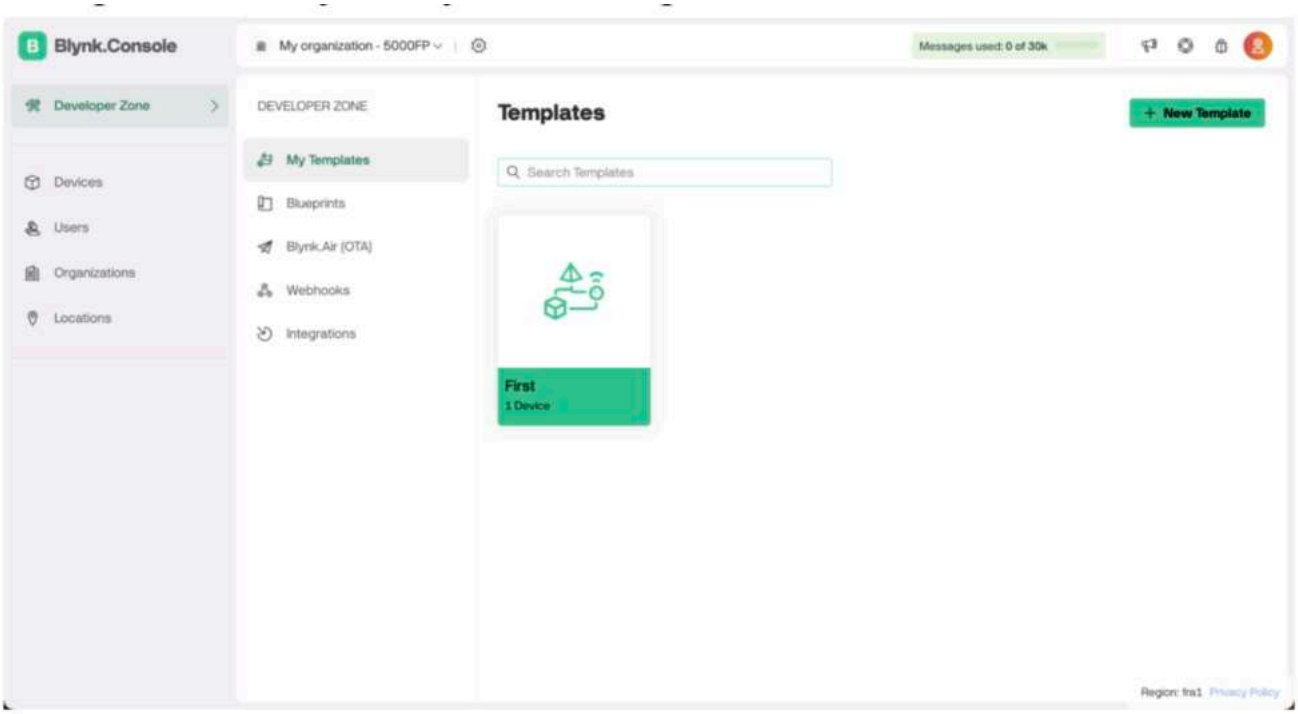


Рисунок 3.3 – Створення шаблону на платформі Blynk

У розділі «Datastreams» шаблону створіть необхідні віртуальні контакти. Наприклад, V0 може використовуватися для керування положенням жалюзі (передача значення від 0 до 100 за допомогою повзунка), а V1 і V2 — для зчитування даних з фотоелементів освітлення в приміщенні (indoor) і на вулиці (outdoor) відповідно. Тип контакту слід вибрати Virtual, а тип даних — Integer або Analog Input, залежно від потреби. Оновлення даних можна налаштувати на періодичне, наприклад, кожну секунду.

Наступний крок – інтеграція платформи Blynk в Arduino IDE для подальшого прототипування, тестування й реалізації системи. Щоб інтегрувати Blynk в середовище розробки, необхідно виконати кілька послідовних кроків, щоб забезпечити належну взаємодію між мікроконтролером (наприклад, ESP32) і хмарною платформою Blynk. Цей процес включає встановлення необхідних бібліотек, підключення до мережі Wi-Fi та налаштування автентифікації для взаємодії з обліковим записом Blynk IoT.

В Arduino IDE переходимо до диспетчера бібліотек (Sketch → Include Library → Manage Libraries…), знаходимо «Blynk» і встановлюємо останню версію бібліотеки Blynk, яка офіційно підтримує Blynk IoT (версія, позначена Blynk Inc.) (рис 3.4). Після встановлення бібліотеки середовище Arduino IDE надасть доступ до прикладів коду, включаючи базовий шаблон для ESP32. Ці приклади можна використовувати як основу для власного коду.

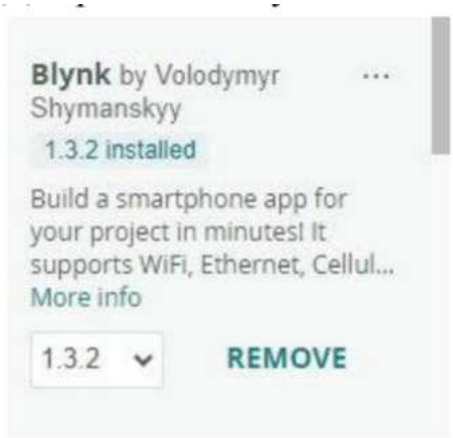


Рисунок 3.4 – Бібліотека Blynk в Arduino IDE

Наступним кроком є налаштування плати ESP32 в меню Tools (рис 3.5). Виберіть відповідну модель (наприклад, ESP32 Dev Module або ESP32 DevkitC) в полі Board, вкажіть налаштування порту (COM) та встановіть швидкість передачі коду (115200 бод).

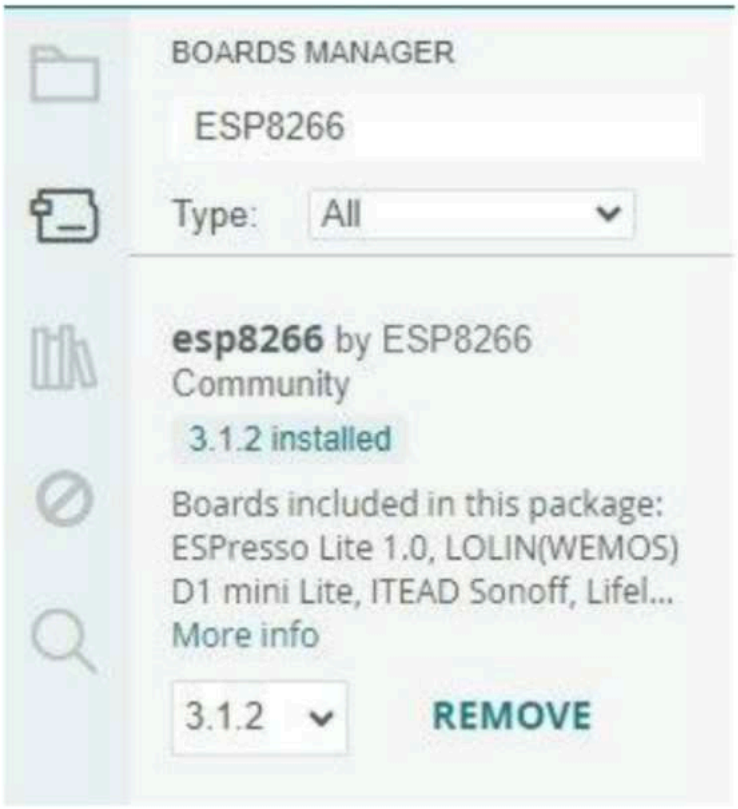


Рисунок 3.5 – Встановлення плати

Далі потрібно вставити в код програми унікальні параметри, надані платформою Blynk під час створення шаблону: BLYNK_TEMPLATE_ID, BLYNK_DEVICE_NAME та BLYNK_AUTH_TOKEN. Копіюємо ці значення з консолі Blynk і замінюємо ними відповідні рядки в шаблоні. Крім того, потрбіно ввести ім'я мережі, Wi-Fi та пароль. Ця мережа дозволить пристрою підключитися до хмарного сервісу Blynk і підтримувати з'єднання.

Останній крок — скомпілювати ескіз і завантажити його на ESP32. Після успішного оновлення прошивки пристрій автоматично підключиться до мережі Wi-Fi і почне обмін даними з хмарним сервісом. Ви можете стежити за станом у режимі

реального часу через панель управління Blynk або мобільний додаток, а також керувати пристроєм за допомогою віртуальних елементів управління (повзунків, графіків, кнопок тощо).

3.2 Монтажна схема

Для створення монтажної схеми будемо використовувати онлайн-ресурс EasyEDA (рис 3.6) та онлайн-симулятор Wokwi. Це веб-середовище для проектування електронних схем і друкованих плат (PCB), яке широко використовується як аматорами, так і професіоналами в галузі електроніки. У цьому проєкті даний інструмент використовувався для створення схеми апаратної частини автоматизованої системи управління жалюзі. Завдяки інтуїтивно зрозумілій структурі та широкому функціоналу EasyEDA дозволяє виконувати весь цикл проектування — від створення схеми до формування файлів Gerber для виробництва друкованих плат.

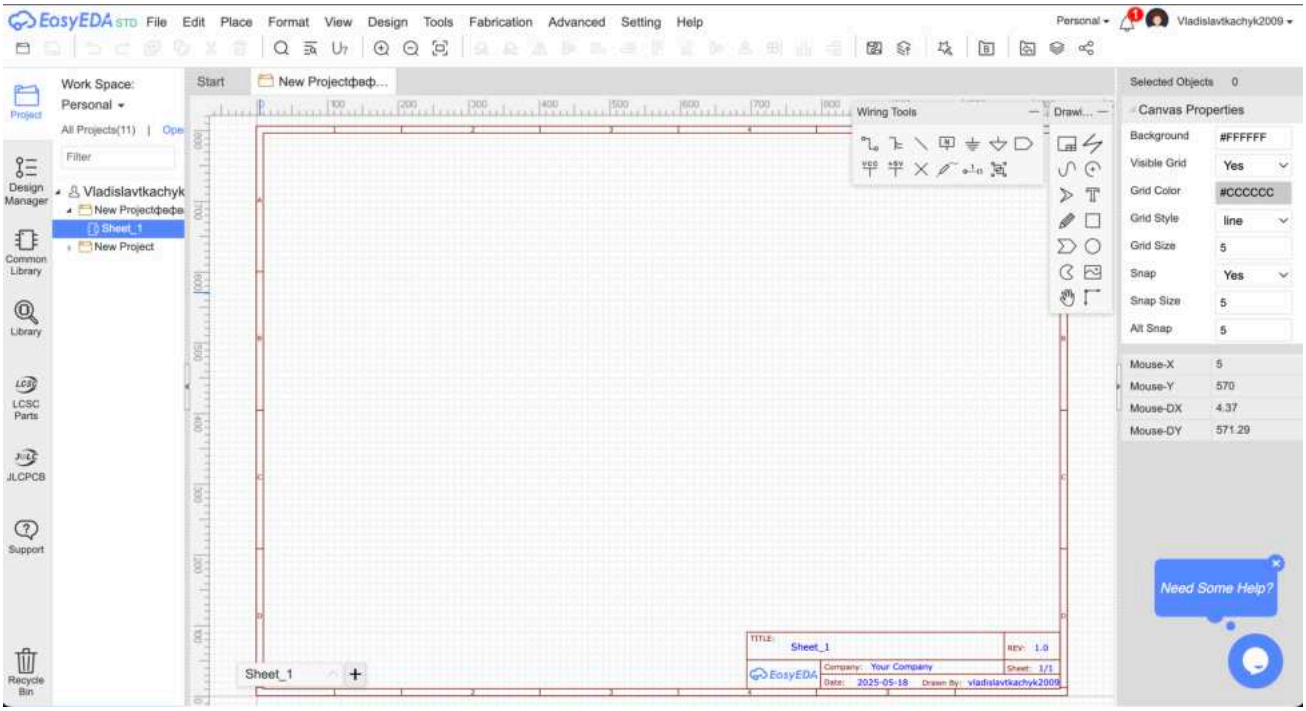


Рисунок 3.6 – Веб-інтерфейс EasyEDA

Для кращого розуміння принципу роботи апаратних компонентів системи рекомендується ознайомитися зі схемою підключення всіх пристроїв, наведеною на рис. 3.7. У даному випадку мова йде про базове підключення основних компонентів, таких як мікроконтролер ESP32, сервомотор і два світлочутливі датчики (фоторезистори). Комбінація цих компонентів забезпечує автоматичне регулювання положення штор або жалюзі в залежності від рівня зовнішнього освітлення.

У даному проекті два модулі фоторезисторів живляться безпосередньо від плати ESP32. Зокрема, два датчики підключені до лінії живлення (VCC) і заземлення (GND). Для передачі інформації про рівень освітлення використовуються цифрові виходи (DO) модуля, підключені відповідно до виводів 32 і 33. Таким чином, мікроконтролер може отримувати бінарні сигнали про рівень яскравості (наприклад, «є світло» або «темно»), що дозволяє не тільки визначати наявність освітлення, але і порівнювати інтенсивність освітлення в двох напрямках. На даному етапі аналоговий вихід (AO) фоторезисторів не використовується, але якщо потрібно більш точне вимірювання рівня освітлення, його можна підключити до відповідного аналогового входу ESP32, що легко реалізується.

Особливу увагу слід звернути на підключення сервомотора, який відповідає за прямий механічний рух штор або жалюзі. На відміну від датчиків, сервомотор не живиться від плати ESP32, а від зовнішнього джерела живлення. Це пов'язано з тим, що сервомотори під час роботи споживають велику кількість струму, щоб запобігти перевантаженню мікроконтролера. У цьому випадку заземлення сервомотора повинно бути з'єднане із заземленням ESP32, що є необхідною умовою для стабільної та точної роботи системи. Вихідний сигнал сервомотора (прийом сигналу PWM) підключається до цифрового виводу 15 ESP32, що дозволяє точно регулювати кут повороту відповідно до даних, оброблених фоторезистором. Загалом, ця апаратна структура є простою, але дуже гнучкою. Вона ефективно реагує на зміни освітлення і відповідно регулює положення штор у приміщенні.

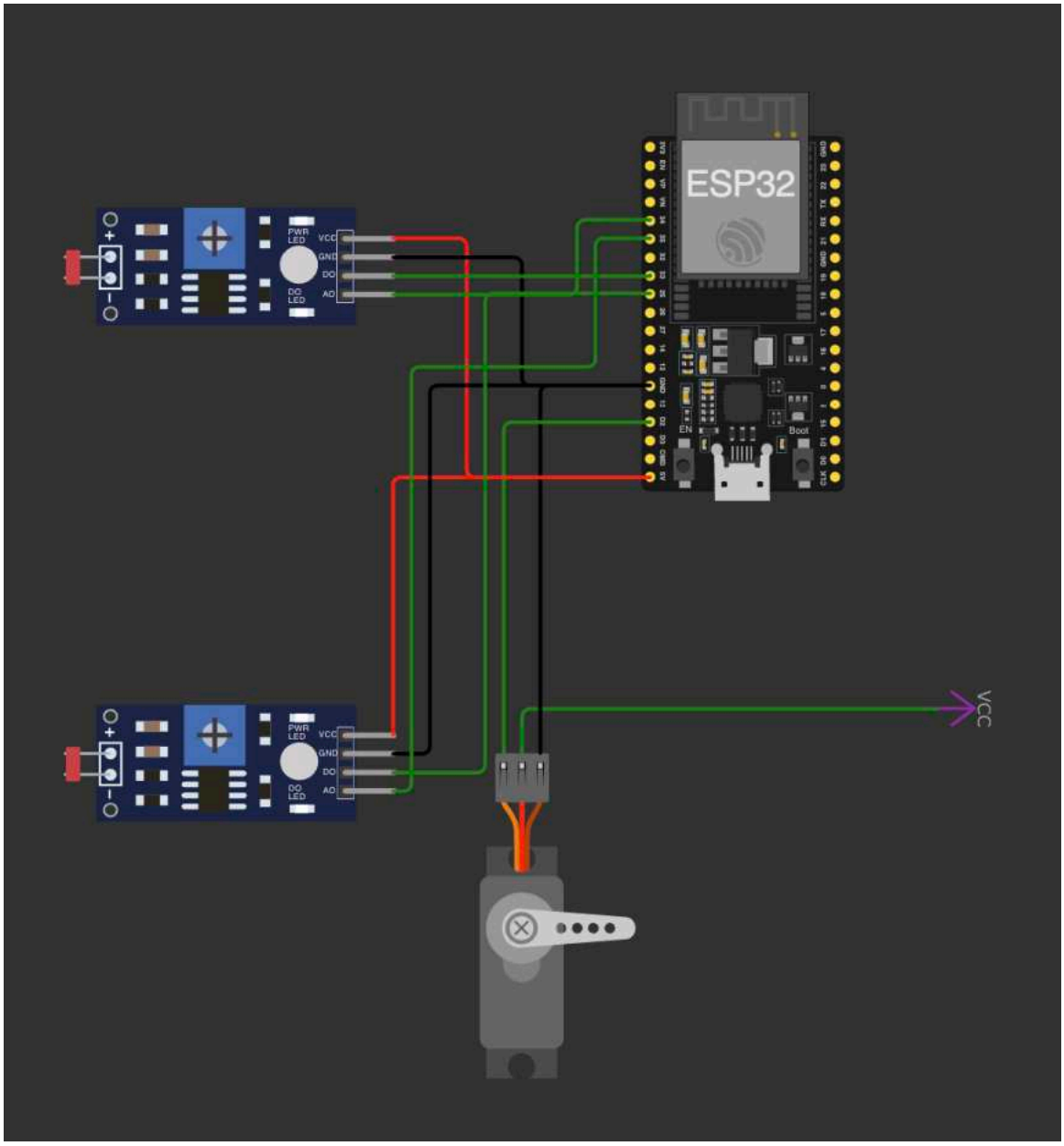


Рисунок 3.7 – Підключення компонентів

Далі перейдемо до створення плати в EasyEDA. Однією з головних переваг EasyEDA є доступність. Програма працює безпосередньо в браузері без встановлення, що особливо зручно для користувачів з обмеженими ресурсами або різних операційних систем. Інтерфейс платформи підтримує бібліотеку з тисячами компонентів, таких як мікроконтролери, резистори, датчики, роз'єми, а також популярні модулі, наприклад ESP32 DevKitC, що значно спрощує створення складових схем складних систем. Іншою перевагою є тісна інтеграція з платформою JLCPCB. Це дозволяє відразу після завершення розробки передавати

плати у виробництво. Крім того, EasyEDA підтримує автоматичне відстеження, включаючи моделювання електронних схем, експорт у формати PDF і PNG, перевірку правил проектування (DRC).

Однак, як і всі інструменти, EasyEDA має деякі недоліки. Зокрема, оскільки це онлайн-інструмент, він безпосередньо залежить від стабільності інтернет-з'єднання. Крім того, для повного використання деяких функцій (наприклад, 3D-візуалізації або експорту в формати третіх сторін) може знадобитися створити обліковий запис і підключитися до преміум-функцій. Деякі складні бібліотеки або нестандартні компоненти можуть бути відсутніми, тому їх доведеться імпортувати вручну або створювати самостійно.

При розробці системи управління функціональними фоторезистивними елементами розробники стикаються з проблемою необхідності ретельного проектування електричних з'єднань між різними компонентами. На початковому етапі необхідно інтегрувати чутливі елементи, в даному випадку фоторезистори (фоторезистори). Фоторезистори, як початкові датчики, реагують на зміни рівня освітлення навколишнього середовища. Фоторезистори (позначені PR1 і PR2) не можуть працювати як самостійні вимірювальні пристрої, їм необхідні допоміжні елементи, які перетворюють зміни опору в сигнали, зрозумілі мікроконтролеру. Для цього кожен фоторезистор з'єднаний послідовно з фіксованими резисторами R1 і R2, утворюючи просту схему розподілу напруги. Зміни яскравості навколишнього середовища впливають на опір фоторезисторів, що безпосередньо перетворюється на зміну напруги на точці з'єднання. Ця зміна напруги як інформаційний сигнал повинна передаватися до «мозку» системи — мікроконтролера ESP32-DEVKITC. Для обробки такого аналогового сигналу точки з'єднання фоторезисторів з резисторами підключаються до спеціальних аналогових вхідних виводів мікроконтролера (VP і VN). Ці клеми мають вбудований аналого-цифровий перетворювач, який перетворює аналоговий рівень напруги, що надходить на мікроконтролер, в цифровий сигнал для подальшого аналізу та прийняття рішень щодо керування.

Одночасно, для нормальної роботи розподільника напруги, вільні кінці R1 і R2 підключені до заземлення мікроконтролера (GND), утворюючи замкнутий ланцюг. Далі розглянемо елемент керування системою — сервомотор U2, який відповідає за фізичне переміщення світлозахисного елемента. Розробники спроектували цей двигун так, щоб він живився від зовнішнього джерела живлення, а саме від виходу +5 В мікроконтролера. Такий спосіб живлення зручний для спрощення схеми, але потрібно враховувати обмеження струму мікроконтролера та вплив на стабільність роботи при великому навантаженні сервомотора. Керування кутом повороту сервомотора здійснюється за допомогою спеціального імпульсного сигналу, що генерується одним з цифрових виводів D2 мікроконтролера і подається на вихід сигналу сервомотора. Цей цифровий вихід налаштований на режим широтно-імпульсної модуляції (ШІМ), що кодує необхідний кут повороту сервомотора в тривалість імпульсу. Нарешті, для забезпечення злагодженої роботи всіх елементів системи необхідно забезпечити спільну базову точку потенціалу (GND). Відповідні виходи всіх елементів, що потребують заземлення, підключені до виводу GND мікроконтролера, утворюючи єдину електричну основу для передачі сигналів і живлення.

Після необхідних підключень отримуємо монтажну плату (рис 3.8):

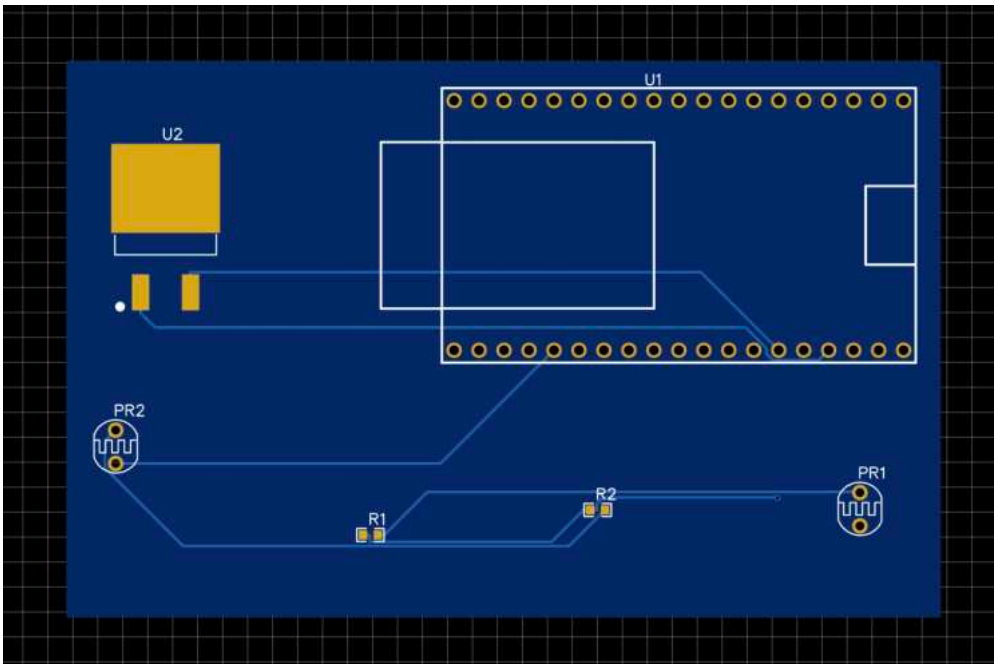


Рисунок 3.8 – Плата в EasyEDA

3.3 Алгоритм та код програми

Для реалізації інтелектуальної системи керування жалюзі були задіяні програмні засоби, які забезпечують як дистанційне керування через мобільний застосунок, так і автономну роботу пристрою на основі показників зовнішнього та внутрішнього освітлення. Основна логіка проєкту реалізована за допомогою мікроконтролера ESP32, що взаємодіє з сервоприводом, датчиками освітленості та платформою Blynk. Наведені нижче фрагменти коду ілюструють ключові функціональні елементи проєкту та розкривають логіку їхньої реалізації.

```
void transmitLightSensorData()
{
int indoorLight = analogRead(LDR_INDOOR_PIN);
int outdoorLight = analogRead(LDR_OUTDOOR_PIN);

Blynk.virtualWrite(V1, indoorLight);
Blynk.virtualWrite(V2, outdoorLight);

Serial.print("Indoor LDR: ");
Serial.print(indoorLight);
Serial.print(" | Outdoor LDR: ");
Serial.println(outdoorLight);

if (isAutoMode) {
int lightDifference = outdoorLight - indoorLight;
int targetAngle;
if (lightDifference > 300) {
targetAngle = 0;
} else if (lightDifference < -300) {
targetAngle = 180;
} else {
targetAngle = 90;
}
```

```
blindsServo.write(targetAngle);
Serial.print("Auto mode angle set to: ");
Serial.println(targetAngle);
}
}
```

Ця функція відповідає за періодичне зчитування аналогових значень з двох фоторезисторів, які встановлені відповідно всередині та зовні приміщення. Значення внутрішньої та зовнішньої освітленості передаються у мобільний застосунок Blynk через віртуальні піни V1 і V2 для зручного моніторингу користувачем. У разі активного автоматичного режиму система порівнює освітлення за межами та в середині приміщення. Якщо зовні значно яскравіше (різниця понад 300 одиниць), жалюзі відкриваються повністю (0°), щоб впустити більше світла. Якщо ситуація протилежна — жалюзі закриваються (180°), інакше встановлюється проміжне положення (90°). Таке рішення дозволяє пристрою динамічно адаптуватися до змін умов освітлення без потреби у втручанні користувача. Це створює певну автоматизацію, що так чи інакше, спрощує користування пристроєм, тобто не потрібно кожного разу змінювати положення жалюзі через додаток.

```
BLYNK_WRITE(V0)
{
if (!isAutoMode) {
int userValue = param.asInt();
int servoAngle = map(userValue, 0, 100, 0, 180);
blindsServo.write(servoAngle);
Serial.print("Manual control angle: ");
Serial.println(servoAngle);
}
}
```

Даний блок забезпечує функціональність ручного керування жалюзі через мобільний інтерфейс користувача. Він активується тільки у випадку, якщо система переведена в ручний режим (isAutoMode == false). Функція BLYNK_WRITE(V0) викликається при зміні значення повзунка на віртуальному піні V0. Отримане значення масштабується з діапазону 0–100 у діапазон 0–180 градусів, що відповідає куту повороту сервопривода. Таким чином користувач може вручну регулювати положення жалюзі, незалежно від освітленості.

Наступний фрагмент реалізує логіку перемикання між автоматичним та ручним режимами роботи. Після зміни стану перемикача у мобільному застосунку, значення параметра, отриманого з віртуального піну V3, зберігається у змінній isAutoMode. В залежності від обраного режиму, на віртуальний пін V4 виводиться відповідне текстове повідомлення, яке інформує користувача про поточний режим роботи системи.

У наступному фрагменті коду виконується початкова конфігурація апаратної частини системи. Здійснюється ініціалізація серійного порту для моніторингу, налаштування таймерів ШІМ для сервоприводу, встановлення відповідних частот та меж тривалості імпульсів. Також визначаються порти, до яких підключено сенсори освітленості, як вхідні. Через виклик Blynk.begin() система підключається до Wi-Fi-мережі та встановлює з’єднання з платформою Blynk. Завершується конфігурація налаштуванням таймера, який кожну секунду викликає функцію зчитування сенсорів та обробки логіки керування. Це дозволяє забезпечити безперервну адаптацію системи до умов навколишнього середовища.

Алгоритми, що беруть участь у роботі інтелектуальної системи управління жалюзі на базі мікроконтролера ESP32 це алгоритм зчитування даних з фоторезисторів, алгоритм обробки інформації про освітлення, алгоритм управління сервоприводом, алгоритм прийняття рішення про обертання, алгоритм калібрування та адаптації.

Алгоритм зчитування даних з фоторезистора базується на послідовній обробці цифрових сигналів у системі сліпого керування, які приймаються модулем,

що має вбудований фоторезистор (LDR) та компаратор. У цьому проекті використовуються два модулі, кожен з яких має цифровий вихід (DO), що передає логічний сигнал на мікропроцесор відповідно до рівня освітлення, виявленого фоторезистором.

Перший крок алгоритму — ініціалізація контактів мікропроцесора ESP32. Під час цього процесу призначаються вхідні порти, підключені до виходу модуля, до якого підключено фоторезистор. Ці порти налаштовані в режимі INPUT, що дозволяє мікроконтролеру отримувати дані з цифрового виходу датчика. Цей режим гарантує готовність до зчитування логічного стану пристрою (0 або 1), що відповідає фіксованому рівню освітлення.

Після цього мікроконтролер переходить у цикл нескінченного циклу сканування, під час якого через певні проміжки часу (наприклад, кожні 100 мілісекунд) зчитує стан двох фоторезисторів. Така періодичність дозволяє системі працювати майже в режимі реального часу і швидко реагувати на зміни зовнішнього освітлення.

При зчитуванні сигналу фоторезистор передає логічне значення залежно від того, чи перевищує рівень освітлення встановлене для модуля порогове значення. Цей пороговий рівень встановлюється вручну за допомогою регулятора (потенціометра), вбудованого в кожен модуль. Якщо рівень освітлення перевищує встановлений пороговий рівень, модуль передає логічний сигнал 1 (1), в іншому випадку — логічний сигнал 0 (0). Таким чином, система отримує цифрову інформацію про зміни освітлення в зоні дії кожного фоторезистора.

Наступним кроком є перевірка стану зчитаних даних. Алгоритм записує отримані значення у відповідні програмні змінні, які використовуються для порівняння та визначення різниці освітлення між двома сторонами. Така логіка порівняння використовується для визначення напрямку основного джерела світла (наприклад, руху Сонця протягом дня) та оцінки необхідності часткового або повного відкриття або закриття жалюзі.

Алгоритм обробки світлової інформації аналізує сигнали, отримані світлочутливим резистором, і визначає, чи потрібно змінювати положення жалюзі. Перший крок цього процесу полягає в порівнянні значень двох цифрових сигналів, що надходять на модуль світлочутливого резистора. Наприклад, якщо сигнал лівого датчика дорівнює 1 (тобто виявлено високий рівень освітлення), а правого — 0 (темрява або низький рівень освітлення), мікроконтролер інтерпретує це як сигнал про те, що джерело світла (наприклад, сонце) знаходиться зліва.На основі цього система може почати переміщати жалюзі у відповідному напрямку для досягнення оптимального ефекту затемнення або пропускання світла.

Другий крок — виявлення змін. Тобто система не тільки оцінює поточний сигнал, але й порівнює його з попереднім значенням. Це допомагає визначити, чи дійсно джерело світла перемістилося на значну відстань, чи зміни були незначними. Такий підхід запобігає виконанню механізмом непотрібних дій через тимчасові або незначні зміни освітлення, спричинені хмарами або рухом людей.

Третій крок — фільтрація помилкових сигналів, що може бути реалізовано за допомогою програмного забезпечення. Для підвищення стабільності системи найкраще використовувати прості методи фільтрації. Наприклад, усереднення значень за кілька циклів або реалізація «логіки більшості» (наприклад, реакція після трьох послідовних однакових результатів). Це дозволяє зменшити реакцію на випадкові або тимчасові перешкоди, підвищити стабільність і надійність системи управління освітленням.

Алгоритм управління сервоприводом постійно виконує кілька важливих кроків, щоб забезпечити точне позиціонування приводу відповідно до умов освітлення. Спочатку виконується запуск PWM: мікроконтролер ESP32 встановлює один із цифрових виходів (наприклад, вивід GPIO15) у режим модуляції ширини імпульсів. Цей режим дозволяє регулювати точний кут повороту сервоприводу шляхом зміни ширини імпульсу в сигналі.

Наступним кроком є формування сигналу. Після аналізу даних світлочутливого резистора система вибирає відповідне значення кута повороту, яке

необхідно передати сервомотору. Наприклад, якщо визначено, що жалюзі повинні бути повністю закриті, мікроконтролер генерує сигнал з шириною імпульсу приблизно 2 мілісекунди, що відповідає повороту сервоприводу на 180 градусів. Для напіввідкритого положення можна створити сигнал тривалістю 1,5 мілісекунди, а для повністю відкритого положення — сигнал тривалістю 1 мілісекунду.

Потім сигнал передається: згенерований сигнал PWM надсилається на контакт управління сервоприводу SG90. Сервомотор реагує на зміни тривалості імпульсу і змінює своє положення відповідно до отриманого значення.

Після передачі сигналу управління виконується перевірка стану. Система може містити певну затримку або логіку підтвердження, щоб забезпечити досягнення сервомотором необхідного положення до надсилання наступної команди. Це особливо важливо для точного позиціонування в системах автоматизації, що вимагають високої надійності.

Алгоритм прийняття рішення про поворот базується на аналізі рівнів освітлення, що надходять від двох фоторезисторів, і дозволяє системі динамічно реагувати на зміни напрямку світла. Перший етап — це логічна операція: мікроконтролер порівнює сигнали, що надходять від обох датчиків, і визначає, де світло є більш інтенсивним. Наприклад, якщо лівий фоторезистор виявляє високу яскравість, а правий фоторезистор виявляє низьку яскравість, система інтерпретує це як світло, що надходить зліва, і вирішує обертати сервопривід у протилежному напрямку, щоб досягти бажаного рівня затемнення.

Потім перевіряються граничні умови. Система постійно контролює поточне положення сервомотора, щоб запобігти перевищенню його фізичних обмежень (наприклад, обертання за межі діапазону від 0° до 180°). Це запобігає механічному зносу, несправностям або втраті точності позиціонування.

Потім оновлюється стан. Коли положення сервомотора змінюється, поточне значення кута, що вказує ступінь відкриття затвора, зберігається в системі. Це

значення використовується в наступних розрахунках для прийняття нових рішень або обмеження кількості рухів.

Останній крок передбачає використання таймера. Рішення щодо обертання сервоприводу приймаються з певною затримкою між циклами (наприклад, кожні 500 мілісекунд або 1 секунда). Такий підхід запобігає надмірній частоті рухів, зменшує навантаження на механізм і підвищує загальну стабільність роботи автоматичної системи.

Алгоритми корекції та адаптації відіграють важливу роль у забезпеченні точності та надійності автоматизованої системи керування жалюзі. На початковому етапі, коли система вмикається або перезапускається, вона виконує початкову корекцію. Це означає, що мікроконтролер встановлює жалюзі у початкове положення, яке вважається нульовим або базовим. Ця ініціалізація запобігає накопиченню відхилень, спричинених попередніми рухами, та забезпечує нормальну роботу в майбутньому.

Далі, для збереження стабільності та відновлюваності налаштувань, система записує інформацію про конфігурацію. Прошивка може зберігати останнє відоме положення жалюзі у вбудованій пам'яті ESP32 або зовнішньому EEPROM. При перезавантаженні або тимчасовій втраті живлення система зчитує ці дані та відновлює попередній стан, забезпечуючи безперервність роботи.

Крім того, алгоритм може бути доповнений функцією адаптації до зовнішніх умов. У цьому випадку програмне забезпечення динамічно коригує порогові значення, які використовуються для зчитування сигналів з фоторезисторів. Таким чином, пристрій може автоматично адаптуватися до змін природного освітлення (наприклад, перехід від яскравого сонячного світла до хмарного або сутінкового) без необхідності ручного втручання. Такий підхід підвищує точність реакції системи на зміни навколишнього середовища та посилює гнучкість і незалежність її роботи.

Проста блокова діаграма системи керування шторами/жалюзі представлена на рисунку 3.9

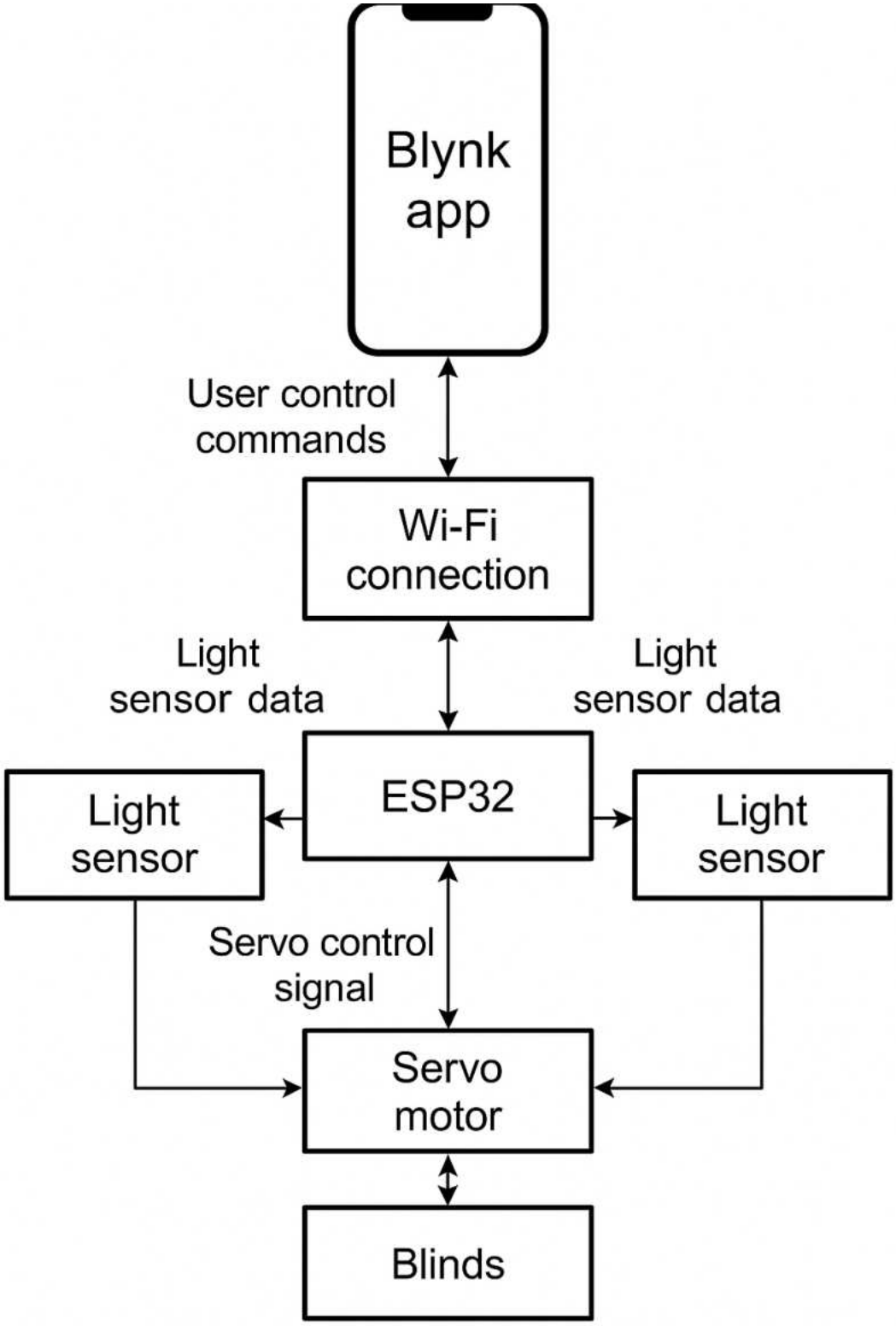


Рисунок 3.9 – Структурна схема системи

3.4 Застосунок Blynk та інтерфейс програмно-технічного засобу системи керування шторами

Для взаємодії з системою управління жалюзі користувач використовує мобільний додаток Blynk. Цей додаток надає зручний графічний інтерфейс для управління пристроями на базі мікроконтролера ESP32.

Додаток можна завантажити та встановити з офіційного магазину додатків (App Store або Google Play) під назвою Blynk IoT. Після встановлення користувач реєструється або входить у свій обліковий запис, створює новий проект і вибирає тип пристрою ESP32 та тип Wi-Fi-з'єднання. Після створення проекту система автоматично генерує унікальний токен автентифікації (Auth Token), який потрібно ввести у відповідне поле програмного коду, щоб забезпечити зв'язок між мобільним інтерфейсом і мікроконтролером.

Інтерфейс Blynk має мінімальний набір функцій для управління жалюзі, але простий у використанні (рис 3.9). Основні елементи інтерфейсу: кнопка типу перемикача, яка підключена до віртуального виводу і дозволяє вручну відкривати або закривати жалюзі; курсор або покроковий регулятор, за допомогою якого користувач може вибрати кут нахилу жалюзі або бажане положення; віджет відображення значення (Value Display або Label), який показує поточний стан системи (відкрито або закрито); можна також встановити блоки автоматизації на основі часу (наприклад, Time Input або Eventor), за допомогою яких можна встановити розклад відкриття або закриття жалюзі в певний час доби.

Завдяки такій конфігурації мобільного додатка користувач може зручно керувати станом системи в ручному або автоматичному режимі, не маючи фізичного доступу до пристрою.

При застосуванні платформи Blynk до системи автоматичного управління шторами на базі мікроконтролера ESP32, це забезпечує гнучкість налаштувань, а також значно спрощує взаємодію між кінцевим користувачем і пристроєм. Blynk – це багатоплатформова мобільна програма, призначена для розробки інтерфейсу

управління пристроями Інтернету речей. Використовуючи Blynk у цьому проєкті, ви можете реалізувати візуальну, зручну та інтуїтивно зрозумілу систему управління шторами або жалюзі без необхідності розробки окремого мобільного додатка.

Під час розгортання клієнта Blynk користувач завантажує офіційний додаток з App Store або Google Play і проходить швидку реєстрацію. Після створення нового проекту в додатку Blynk, користувач вибирає тип контролера (в даному випадку ESP32) і отримує унікальний код авторизації, який використовується для запису прошивки пристрою. Цей код використовується для створення безпечного з'єднання з хмарним сервером Blynk.

Однією з головних переваг Blynk є підтримка локального з'єднання (в межах однієї мережі Wi-Fi) і взаємодії з хмарою, що забезпечує повне дистанційне керування. Таким чином, користувачі можуть керувати станом штор в режимі реального часу з будь-якого місця, підключеного до Інтернету. Це значно підвищує комфорт і відповідає вимогам сучасного «розумного будинку».

Крім того, Blynk підтримує функцію миттєвих сповіщень, що дозволяє надсилати сповіщення користувачам при зміні стану штор, в екстрених випадках або при запуску тригерів. Важливою функцією є також реєстрація подій, що дозволяє відстежувати історію змін положення штор або час запуску автоматичних сценаріїв. Всі ці функції роблять Blynk універсальним рішенням для подібних проєктів та сценаріїв, адже є розширений інструментарій та оптимізована робота системи з мобільним додатком.

За допомогою функції управління через віртуальні контакти (Virtual Pins) розробники можуть реалізувати гнучку логіку взаємодії між мобільними додатками та мікроконтролером ESP32. Ця функція не обмежується стандартними виходами GPIO, а дозволяє обробляти дані, отримані з інтерфейсу користувача Blynk, безпосередньо в прошивці. За допомогою віртуальних контактів (Virtual Pins) можна, наприклад, налаштувати параметри роботи пристрою з мобільного додатка.

3.5 Вартість матеріалів

Вартість матеріалів, запропонованих для системи керування шторами/жалюзі. Вартость матеріалів наведена в таблиці 3.1

Таблиця 3.1 – Вартість матеріалів

№	Компонент	К-сть	Ціна за 1шт (грн)	Загальна ціна (грн)
1	Мікроконтролер ESP32 DevKit	1	200	200
2	Модуль фоторезистора (з аналоговим та цифровим виходами)	2	30	60
3	Сервопривід SG90	1	60	60
4	Блок живлення 5V (адаптер або USB-адаптер)	1	70	70
5	Макетна плата (breadboard)	1	40	40
6	Провідники (Dupont проводи)	1 (набір)	30	30

Для створення автоматичної системи управління шторами були використані доступні компоненти за прийнятною ціною, загальна вартість яких становить приблизно 460 гривень. В цілому проект доводить, що створення функціональної системи IoT для автоматизації домашніх процесів можливе з невеликими фінансовими витратами. Такий підхід дозволяє ефективно реалізувати подібні пристрої в освітньому, експериментальному та домашньому середовищі без використання дорогого обладнання.

3.6 Висновки до третього розілу

Даний розділ присвячений представленню результатів комплексного дослідження. Дослідження стосувалося реалізації основних програмних та апаратних аспектів автоматичної системи управління шторами на базі мікроконтролера ESP32. Розробка системи була полегшена завдяки використанню мобільного додатку Blynk. Важливим етапом на стадії розробки було створення середовища Arduino IDE, яке спростило процеси компіляції, завантаження та налагодження коду для ESP32. Це середовище також забезпечило необхідні з'єднання з бібліотеками для сервомоторів та модуля Blynk.

Була розроблена схема з'єднання основних компонентів, що включає два фоторезистори, які виконують функцію датчиків освітлення, та сервомотори, які фізично регулюють положення штор. Під час складання сервомотори були підключені до незалежного джерела живлення з метою запобігання перевантаженню ESP32 та забезпечення стабільної роботи пристрою. Була розроблена блок-схема, що ілюструє взаємозв'язки між апаратними та програмними компонентами системи. Розглянута схема охоплює мобільний інтерфейс, мікроконтролер, виконавчий модуль та модуль датчика.

Алгоритм програмного забезпечення був реалізований на C++ з використанням бібліотек Blynk, Servo і Wi-Fi. Ці бібліотеки забезпечують автоматичну реакцію системи на зміни рівня освітлення та дозволяють вручну керувати жалюзі через інтерфейс мобільного додатка. Алгоритм включає зчитування стану датчика освітлення, обробку отриманих даних, логіку прийняття рішень та генерацію відповідних команд керування сервомотором. Інтерфейс додатка Blynk дозволяє зручно керувати системою за допомогою простих візуальних елементів (наприклад, кнопок, повзунків або дисплея) на смартфоні без необхідності знання програмування або електроніки. Крім того, вони дозволяють користувачам вручну взаємодіяти з пристроєм через інтуїтивно зрозумілий інтерфейс мобільного додатку Blynk.

Програмне забезпечення містить кілька основних функцій: зчитування сигналів з датчиків, обробка даних і фільтрація шуму, прийняття рішень на основі аналізу освітлення, а також генерація керуючих сигналів для приводу сервомотора. Всі ці операції виконуються в організованому робочому циклі, що забезпечує послідовність і надійність виконання команд.

Інтерфейс користувача Blynk спрощує взаємодію з системою за допомогою візуальних елементів, таких як кнопки, повзунки, текстовий дисплей тощо. Це дозволяє користувачам без знань у програмуванні або електроніці налаштовувати параметри системи, змінювати режим роботи або переглядати поточний стан, що особливо важливо для широкого кола користувачів.

Особлива увага приділена аналізу вартості компонентів. Розрахунок фізичних витрат показує, що навіть при обмеженому бюджеті реалізація цієї системи є економічно вигідною. Завдяки використанню доступних компонентів, таких як плата ESP32, двигун SG90, модуль фоторезистора, стабільне джерело живлення та основні елементи з'єднання, цей проект може бути використаний в освіті, лабораторних роботах або системах домашньої автоматизації.

Таким чином, у розділі 3 наведено повний процес розробки системи автоматичного керування шторами – від вибору апаратних компонентів, проектування схеми до реалізації програмних алгоритмів та розробки користувацького інтерфейсу. Запропоноване рішення є простим у реалізації, економічно вигідним, легко адаптується до різних сценаріїв використання та може бути ефективно інтегровано в системи розумного будинку або більш широкі платформи Інтернету речей.

ВИСНОВКИ

Під час дослідження було розроблено комплексний набір програмних і апаратних засобів для управління системою штор і жалюзі на базі мікроконтролера ESP32 з дистанційним керуванням за допомогою мобільного додатку. Цей проект відповідає сучасним тенденціям розвитку «розумного будинку» і забезпечує більш високий рівень комфорту, безпеки та енергоефективності в житлових приміщеннях.

Апаратна частина системи використовує мікроконтролер ESP32, який поєднує високу обчислювальну потужність та вбудовані модулі Wi-Fi і Bluetooth, що забезпечує надійний бездротовий зв'язок з мобільним додатком. Сервомотор для механічного керування жалюзями має функцію точного позиціонування, що забезпечує плавне переміщення жалюзей у визначене положення та їх точне позиціонування. Сервомотор підключений до ESP32 за допомогою PWM (широтно-імпульсна модуляція), що забезпечує високу точність контролю кута повороту.

Програмне забезпечення складається з двох основних частин: прошивка ESP32 та мобільний додаток, розроблений на платформі React Native. Прошивка відповідає за обробку команд користувача, що надходять через Wi-Fi, управління сервоприводом та моніторинг стану системи. Мобільний додаток і мікроконтролер обмінюються даними за протоколом MQTT, що забезпечує мінімальну затримку і надійність передачі команд в режимі реального часу.

При розробці алгоритму управління були враховані особливості механічної конструкції жалюзі, такі як швидкість, плавний пуск і зупинка, а також безпечне положення при відключенні живлення. Також реалізовано механізм обробки помилок і аварійних ситуацій для запобігання пошкодженню обладнання.

У процесі розробки виникли технічні складнощі, зокрема інтеграція ESP32 з сервоприводами різних моделей, а також забезпечення стабільності зв'язку в умовах можливих перешкод Wi-Fi. Для вирішення цих проблем було проведено

серію тестів, оптимізовано налаштування бездротової мережі та скориговано програмне забезпечення для підвищення стабільності системи.

За допомогою мобільного додатку користувачі тепер можуть керувати жалюзі через інтуїтивно зрозумілий інтерфейс, не тільки вручну регулюючи їх положення, але й встановлюючи автоматичні сценарії, наприклад, автоматичне відкривання або закривання жалюзі за розкладом або залежно від інтенсивності освітлення.

Перспективи подальшого розвитку системи включають реалізацію голосового керування за допомогою інтеграції з популярними голосовими помічниками, розробку адаптивного керування на основі даних датчиків освітлення та температури, а також розширення системи за рахунок додавання підтримки декількох зон керування.

На завершення, система управління шторами є важливим кроком у напрямку автоматизації домашніх процесів і підвищення якості життя, що підтверджує ефективність технологій Інтернету речей в системах розумного будинку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Adhikary A., Halder S., Bose R., Panja S., Halder S., Pratihar J., Dey A. Design and Implementation of an IoT-Based Smart Home Automation System in Real World Scenario. *EAI Endorsed Transactions on Internet of Things*. 2024. Vol. 10, no. 1. Art. e1. DOI: doi.org/10.4108/eai.13-7-2018.163932.

2. Taiwo O., Ezugwu A. E. Internet of Things-Based Intelligent Smart Home Control System. *Security and Communication Networks*. 2021. Vol. 2021. Art. ID 9928254. DOI: doi.org/10.1155/2021/9928254.

3. Oluwafemi I. B., Bello O. O., Obasanya T. D. Design and Implementation of a Smart Home Automation System. *Innovare Journal of Engineering and Technology*. 2023. Vol. 11, no. 1. P. 3–7. DOI: doi.org/10.22159/ijet.2023.v11i1.46883.

4. Imam A. S., Garba A., Isah A. S., Chukwu I. F., Baballe M. A., Shehu S. Design of an ESP32-Based IoT Smart Home Automation Management System. *Global Journal of Research in Engineering & Computer Sciences*. 2024. Vol. 4, no. 4. P. 121–130. DOI: doi.org/10.5281/zenodo.13334218.

5. Oner V. O. *Developing IoT Projects with ESP32: Automate Your Home or Business with Inexpensive Wi-Fi Devices*. Birmingham : Packt Publishing, 2021. 474 p. ISBN 978-1-83864-116-0. URL: https://www.amazon.com/Developing-IoT-Projects-ESP32-inexpensive/dp/1838641165 (дата звернення: 27.05.2025).

6. Hercog D., Lerher T., Truntič M., Težak O. Design and Implementation of ESP32-Based IoT Devices. *Sensors*. 2023. Vol. 23, no. 15. Art. 6739. DOI: doi.org/10.3390/s23156739.

7. Paul A., Tiwari R. Smart Home Automation System Based on IoT Using Chip Microcontroller. *Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, March 23-25, 2022. New Delhi : IEEE, 2022. P. 564–568. DOI: doi.org/10.23919/INDIACom54597.2022.9763287.

8. Yao G., Gu M., Chen Y., Chen Y., Chen H., Tian S. Design of Intelligent Curtain. *Academic Journal of Engineering and Technology Science*. 2023. Vol. 6, no. 11. P. 43–50. DOI: doi.org/10.25236/AJETS.2023.061107.

9. Wolniak R., Grebski W. The Usage of Smart Window Blinds in Smart Home. *Zeszyty Naukowe Politechniki Śląskiej. Organizacja i Zarządzanie*. 2023. Nr 190. P. 249–258. DOI: doi.org/10.29119/1641-3466.2023.190.17.

10. Sharif Z., Jung L. T., Ayaz M., Yahya M., Khan D. Smart Home Automation by Internet-of-Things Edge Computing Platform. *International Journal of Advanced Computer Science and Applications*. 2022. Vol. 13, no. 4. P. 400–407. DOI: doi.org/10.14569/IJACSA.2022.0130455.

11. Omran M. A., Hamza B. J., Saad W. K. The Design and Fulfillment of a Smart Home (SH) Material Powered by the IoT Using the Blynk App. *Materials Today: Proceedings*. 2022. Vol. 60, Part 3. P. 1199–1212. DOI: doi.org/10.1016/j.matpr.2021.09.346.

12. Kindangen J. I., Rompas L. M., Mandey J. C. An Automatic Curtain System for Indoor Daylight Control. *The International Journal of Engineering and Science (IJES)*. 2024. Vol. 13, no. 1. P. 76–83. DOI: https://doi.org/10.9790/1813-13017683.

13. Adepegba A., Forrest J., Meinke R., Tran T. *Electric Blind*. 2021. URL: https://www.ece.ucf.edu/seniordesign/sp2021su2021/g13/ (дата звернення: 27.05.2025).

14. Biswas S. *Development of Closed Loop Illuminance Control Using Motorized Vertical Blinds and Dimmable LED Sources for Energy Saving*. Master's thesis. University of Saskatchewan, Saskatoon, 2022. 86 p. URL: https://harvest.usask.ca/handle/10388/14270 (дата звернення: 27.05.2025).

15. Roy S., Kabir M. H., Ahmed M. T. IoT Based Low-Cost Smart Home Automation and Security System Using Wireless Technology. *Australian Journal of Engineering and Innovative Technology*. 2023. Vol. 5, no. 3. P. 101–112. DOI: https://doi.org/10.34104/ajeit.023.01010112.

16. Chen Z., Xie M., Zu Q., Abdufattokhov S. Electrical Automation Intelligent Control System Based on Internet of Things Technology. *Electrica*. 2023. Vol. 23, no. 2. P. 329–337. DOI: https://doi.org/10.5152/electrica.2023.22117.

17. Mursidah, Setiawan H., Harahap M. K., Ilham D. N., Yunan A. Arduino Based Light Intensity Auto Curtain. *Brilliance: Research of Artificial Intelligence*. 2022. Vol. 2, no. 1. P. 1–6. DOI: https://doi.org/10.47709/brilliance.v2i1.1456.

18. Meirita C., Syahril, Azizahwati. Automatic Curtain Prototype Design Based on Arduino Uno. *Journal of Frontier Research in Science and Engineering*. 2024. Vol. 2, no. 1. P. 12–15. URL: https://journal.riau-edutech.com/index.php/jofrise/article/view/27 (дата звернення: 27.05.2025).

19. Babayev A. K. Development and Implementation of Android Application Based Curtain Control System. *International Journal of Multidisciplinary Research Transactions*. 2023. Vol. 5, no. 4. P. 10–15. DOI: https://doi.org/10.5281/zenodo.7775756.

20. Zhang Q., Zhao J., Wen H., Hao H. Design of Intelligent Curtain Control Circuit Based on Single Chip Microcomputer. *Artificial Life and Robotics* (Proceedings of the 2021 International Conference on Artificial Life and Robotics). 2021. Vol. 26, no. 3. P. 676–679. URL: https://alife-robotics.co.jp/members2021/icarob/data/html/data/OS/OS12/OS12-7.pdf (дата звернення: 27.05.2025).

21. Inkarojrit V. *Balancing Comfort: Occupants' Control of Window Blinds in Private Offices* : Doctoral Dissertation. Berkeley : University of California, 2005. 522 p. URL: https://escholarship.org/uc/item/3rd2f2bg (дата звернення: 27.05.2025).

22. *Inside the Smart Home* / ed. by R. Harper. London : Springer-Verlag, 2003. 264 p. ISBN 978-1-85233-688-2. DOI: https://doi.org/10.1007/b97527.

23. Leitner G., Harper R. *The Future Home is Wise, Not Smart: A Human-Centric Perspective on Next Generation Domestic Technologies*. Cham : Springer, 2015. 128 p. ISBN 978-3-319-23092-4. DOI: https://doi.org/10.1007/978-3-319-23093-1.

24. Harrold C. A Brief History of Smart Things. *Practical Smart Device Design and Construction: Understanding Smart Technologies and How to Build Them Yourself*.

Berkeley, CA : Apress, 2020. P. 3–14. ISBN 978-1-4842-5613-8. DOI: https://doi.org/10.1007/978-1-4842-5614-5.

25. Ali S., Yusuf Z. *Mapping the Smart-Home Market*. Boston : The Boston Consulting Group, 2018. URL: https://www.bcg.com/publications/2018/mapping-smart-home-market (дата звернення: 27.05.2025).

26. Kurniawan A. *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Birmingham : Packt Publishing Ltd, 2019. 386 p. ISBN 978-1-78953-850-3.

27. Foltýnek P., Babiuch M., Šuránek P. Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board. *Measurement and Control*. 2019. Vol. 52, No. 7–8. P. 970–984. DOI: https://doi.org/10.1177/0020294019857748.

28. Akbar A., Zaenudin Z., Mutaqin Z., Samsumar L. D. IoT-Based Smart Room Using Web Server-Based ESP32 Microcontroller. *Formosa Journal of Computer and Information Science*. 2022. Vol. 1, No. 2. P. 91–98. DOI: https://doi.org/10.55927/fjcis.v1i2.1241.

29. Özkendirci B. *Aesthetic and Functional Interaction of Curtain with Architectural Elements*. [б.р.]. URL: https://www.academia.edu/35679237/AESTHETIC_AND_FUNCTIONAL_INTERACTION_OF_CURTAIN_WITH_ARCHITECTURAL_ELEMENTS (дата звернення: 27.05.2025).

30. Shanmugapriya D., Patel A., Srivastava G., Lin J. C.-W. MQTT Protocol Use Cases in the Internet of Things. *Big Data Analytics: 9th International Conference, BDA 2021, Virtual Event, December 15–18, 2021, Proceedings*. Cham : Springer, 2021. P. 146–162. DOI: https://doi.org/10.1007/978-3-030-93620-4_12.

31. Paris I. L. B. M., Habaebi M. H., Zyoud A. M. Implementation of SSL/TLS security with MQTT protocol in IoT environment. *Wireless Personal Communications*. 2023. Vol. 132, No. 1. P. 163–182. DOI: https://doi.org/10.1007/s11277-023-10323-z.

32. Al Hanif A., Ilyas M. Effective feature engineering framework for securing MQTT protocol in IoT environments. *Sensors*. 2024. Vol. 24, No. 6. Art. 1782. DOI: https://doi.org/10.3390/s24061782.

33. Sahmi I., Abdellaoui A., Mazri T., Hmina N. MQTT-PRESENT: Approach to secure internet of things applications using MQTT protocol. *International Journal of Electrical and Computer Engineering*. 2021. Vol. 11, No. 5. P. 4342–4349. DOI: https://doi.org/10.11591/ijece.v11i5.pp4342-4349.

34. Manowska A., Wycisk A., Nowrot A., Pielot J. The use of the MQTT protocol in measurement, monitoring and control systems as part of the implementation of energy management systems. *Electronics*. 2023. Vol. 12, No. 1. Art. 17. DOI: https://doi.org/10.3390/electronics12010017.

35. Rafiquzzaman M. *Microcontroller Theory and Applications with the PIC18F*. Hoboken : John Wiley & Sons, 2025. 592 p. ISBN 978-1-119-60970-9.

36. Asadi F. *Electric Circuit Analysis with EasyEDA*. Cham : Springer, 2022. 257 p. DOI: https://doi.org/10.1007/978-3-031-12935-0.

37. Fan Hua, Yang Tianchi, Liu Jianming, Li Zonglin, Guo Xinkai, Sun Yan. EXPLORATION OF THE TEACHING METHOD OF THE COURSE "CIRCUIT ANALYSIS AND ANALOG CIRCUITS".

38. Грабіщенко Я. О. Огляд сучасного програмного забезпечення для дизайну друкованих плат. [Тип публікації]. Харків : ХНУРЕ, 2025. [Інші необхідні дані: Назва видання, сторінки, URL/DOI].

39. Pandit P. *Study of Bluetooth protocol and applications*. Preprint. 2021. URL: https://www.researchgate.net/publication/354263007_Study_of_Bluetooth_protocol_and_applications (дата звернення: 27.05.2025).

40. Pahlavan K., Krishnamurthy P. Evolution and impact of Wi-Fi technology and applications: A historical perspective. *International Journal of Wireless Information Networks*. 2021. Vol. 28. P. 3–19. DOI: https://doi.org/10.1007/s10776-021-00500-2.

41. *Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy* / ed. by A. G. Khang, V. Abdullayev, V. Hahanov, H. Shah. London :

Routledge, 2024. 324 p. URL: https://www.routledge.com/Advanced-IoT-Technologies-and-Applications-in-the-Industry-40-Digital-Economy/Khang-Abdullayev-Hahanov-Shah/p/book/9781032552040 (дата звернення: 27.05.2025).

42. *AI-Aided IoT Technologies and Applications for Smart Business and Production* / ed. by A. G. Khang, A. Misra, S. K. Gupta, H. Shah. Boca Raton : CRC Press, 2023. 342 p. URL: https://www.routledge.com/AI-Aided-IoT-Technologies-and-Applications-for-Smart-Business-and-Production/Khang-Misra-Gupta-Shah/p/book/9781032490076 (дата звернення: 27.05.2025).

43. *Big Data, Cloud Computing and IoT: Tools and Applications* / ed. by S. Rani, P. Bhambri, A. Kataria, A. G. Khang, A. K. Sivaraman. Boca Raton : CRC Press, 2023. 452 p. URL: https://www.routledge.com/Big-Data-Cloud-Computing-and-IoT-Tools-and-Applications/Rani-Bhambri-Kataria-Khang-Sivaraman/p/book/9781032287430 (дата звернення: 27.05.2025).

44. Chen C., Chen X., Das D., Akhavan-Tabatabaei R. Overview and performance evaluation of Wi-Fi 7. *IEEE Communications Standards Magazine*. 2022. Vol. 6, no. 2. P. 12–18. DOI: https://doi.org/10.1109/MCOMSTD.2022.9752732.

45. Kim H., Choi H., Kang H., An S. A systematic review of the smart energy conservation system using IoT-based home automation. *Renewable and Sustainable Energy Reviews*. 2021. Vol. 140. Art. 110755. DOI: https://doi.org/10.1016/j.rser.2021.110755.

46. Le-Vu T., Truong D. C., Nguyen T. K. Edge intelligence: A survey on enabling technologies and applications. *Future Generation Computer Systems*. 2021. Vol. 115. P. 278–298. DOI: https://doi.org/10.1016/j.future.2020.09.013.

47. Ghazal M., Harkous H., Harb H. Towards a secure and privacy-preserving smart home. *Procedia Computer Science*. 2022. Vol. 207. P. 2856–2861. DOI: https://doi.org/10.1016/j.procs.2022.09.343.

48. Silva V., Gomes F. Intelligent automation system for residential lighting using ESP32. *Journal of Physics: Conference Series*. 2023. Vol. 2569, no. 1. Art. 012038. DOI: https://doi.org/10.1088/1742-6596/2569/1/012038.

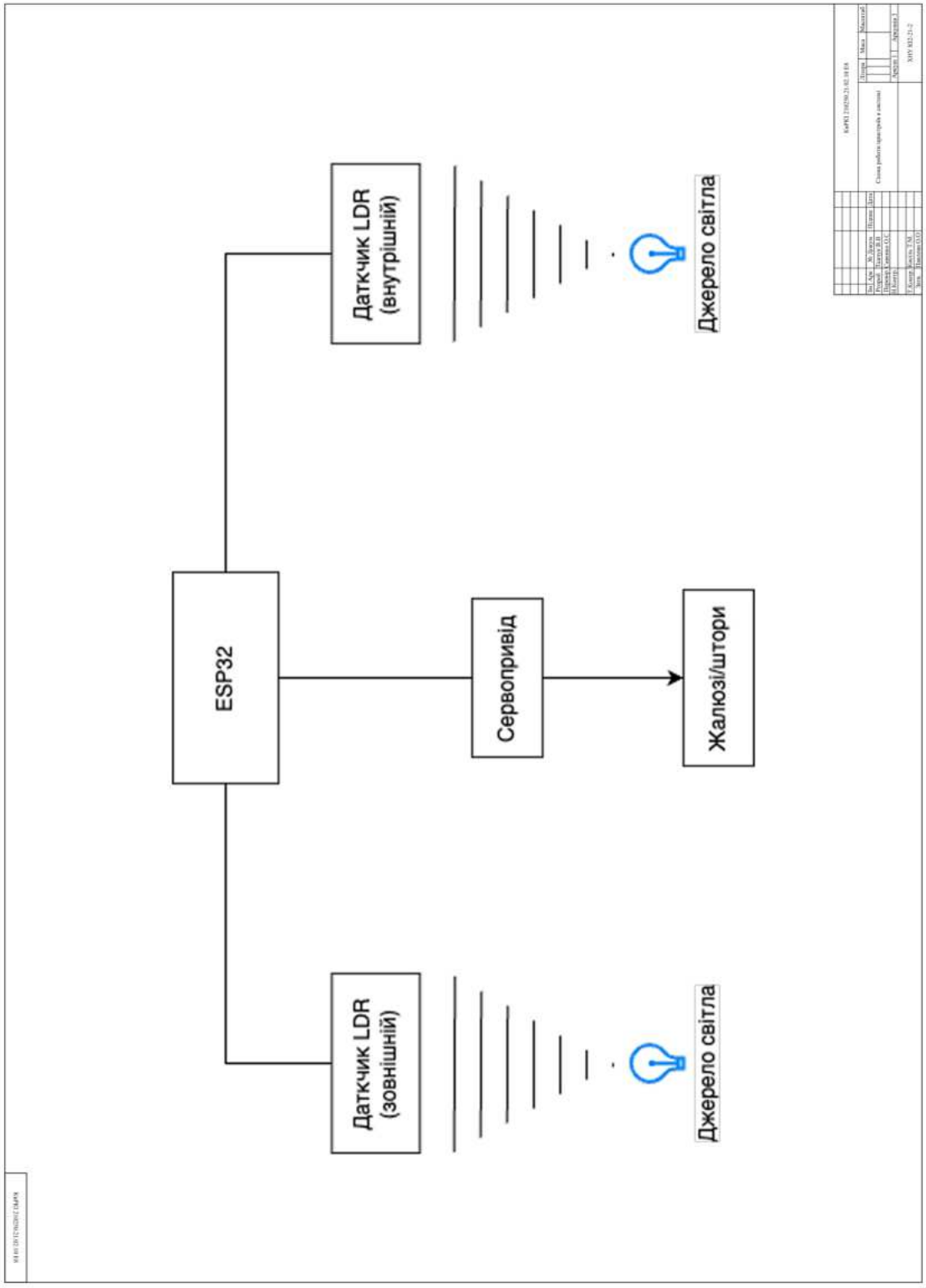
49. Ashraf M. U., Malik S. A., Mahmood H. WiFi-enabled IoT architecture for controlling home appliances using ESP32. *Advances in Internet of Things*. 2023. Vol. 13, no. 3. P. 97–110. DOI: https://doi.org/10.4236/ait.2023.133007.

50. Rahman M. S., Hossain M. T., Jahan N. Home automation using ESP32 and Google Firebase. *International Journal of Scientific Research and Engineering Development*. 2022. Vol. 5, no. 4. P. 122–128. URL: https://ijsred.com/home-automation-using-esp32-and-google-firebase/ (дата звернення: 27.05.2025).

Додаток А

(обов’язковий)

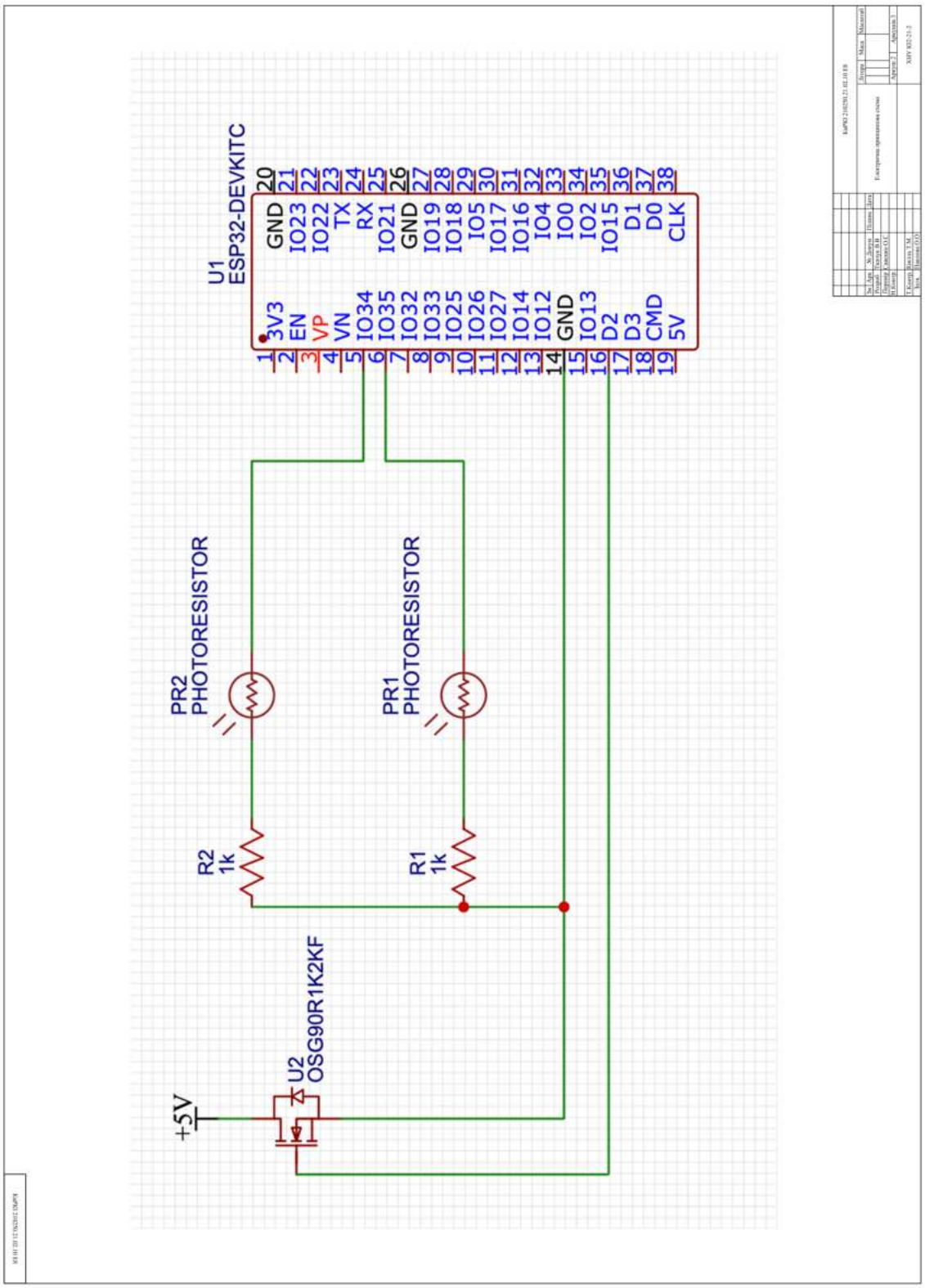
КОПІЯ КРЕСЛЕННЯ «СХЕМА РОБОТИ ПРИСТРОЇВ У СХЕМІ»



Додаток Б

(обов'язковий)

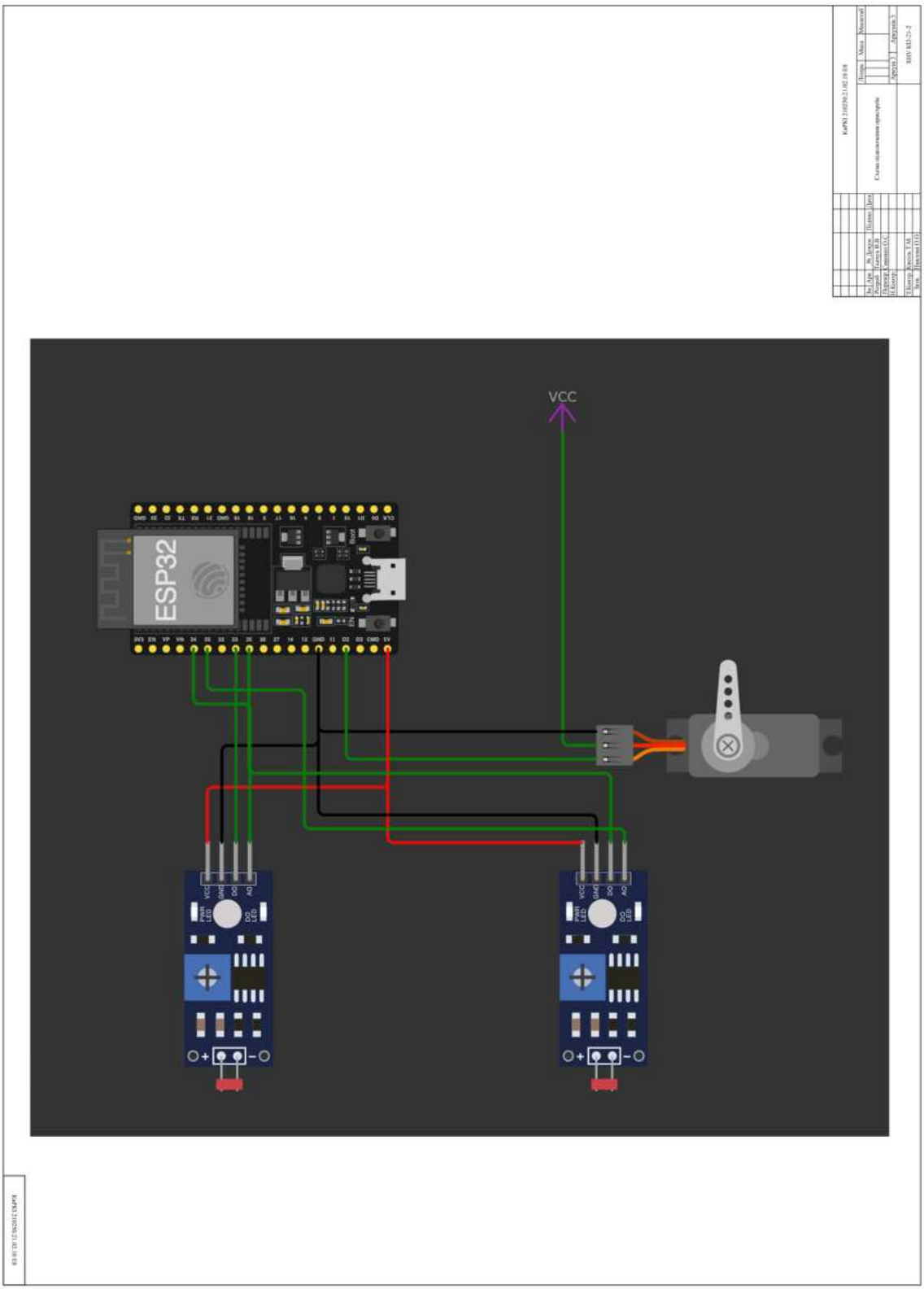
КОПІЯ КРЕСЛЕННЯ «ЕЛЕКТРИЧНА ПРИНЦИПОВА СХЕМА»



Додаток В

(обов’язковий)

КОПІЯ КРЕСЛЕННЯ «СХЕМА ПІДКЛЮЧЕННЯ ПРИСТРОЇВ»



Додаток Г

(обов'язковий)

КОД ПРОГРАМИ

```
#include <BlynkSimpleEsp32.h>
#include <ESP32Servo.h>

#define BLYNK_TEMPLATE_ID           "xxxxxxxxxxxxxxxxx"
#define BLYNK_DEVICE_NAME           "SmartBlinds"
#define BLYNK_AUTH_TOKEN            "xxxxxxxxxxxxxxxxx"

#define SERVO_PIN 13
#define LDR_INDOOR_PIN 34
#define LDR_OUTDOOR_PIN 35

Servo blindsServo;

char blynkAuth[] = BLYNK_AUTH_TOKEN;
char wifiSSID[] = "xxxxxxxxx";
char wifiPass[] = "xxxxxxxxx";

BlynkTimer dataTimer;

bool isAutoMode = true; // true = авто, false = ручне керування

BLYNK_WRITE(V0) // Повзунок ручного керування
{
if (!isAutoMode) {
int userValue = param.asInt();
int servoAngle = map(userValue, 0, 100, 0, 180);
blindsServo.write(servoAngle);
Serial.print("Manual control angle: ");
Serial.println(servoAngle);
}
```

```
}

BLYNK_WRITE(V3) // Перемикач режиму (0 – ручний, 1 – авто)
{
isAutoMode = param.asInt();
if (isAutoMode) {
Blynk.virtualWrite(V4, “Автоматичний режим”);
} else {
Blynk.virtualWrite(V4, “Ручний режим”);
}
}

void transmitLightSensorData()
{
int indoorLight = analogRead(LDR_INDOOR_PIN);
int outdoorLight = analogRead(LDR_OUTDOOR_PIN);

Blynk.virtualWrite(V1, indoorLight);
Blynk.virtualWrite(V2, outdoorLight);

Serial.print(“Indoor LDR: “);
Serial.print(indoorLight);
Serial.print(” | Outdoor LDR: “);
Serial.println(outdoorLight);

if (isAutoMode) {
int lightDifference = outdoorLight - indoorLight;
int targetAngle;
if (lightDifference > 300) {
  targetAngle = 0; // відкрити повністю
} else if (lightDifference < -300) {
  targetAngle = 180; // закрити повністю
} else {
  targetAngle = 90; // напіввідкрито
}
```

```
blindsServo.write(targetAngle);
Serial.print("Auto mode angle set to: ");
Serial.println(targetAngle);
}
}

void setup()
{
Serial.begin(115200);

ESP32PWM::allocateTimer(0);
ESP32PWM::allocateTimer(1);
ESP32PWM::allocateTimer(2);
ESP32PWM::allocateTimer(3);

blindsServo.setPeriodHertz(50);
blindsServo.attach(SERVO_PIN, 1000, 2000);

pinMode(LDR_INDOOR_PIN, INPUT);
pinMode(LDR_OUTDOOR_PIN, INPUT);

Blynk.begin(blynkAuth, wifiSSID, wifiPass);
Blynk.virtualWrite(V4, “Автоматичний режим”);

dataTimer.setInterval(1000L, transmitLightSensorData);
}

void loop()
{
Blynk.run();
dataTimer.run();
}
```

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Владислав ТКАЧУК

Співавтор:

Назва: Ткачук_Система керування шторами / жалюзі через ESP32 та мобільний застосунок

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:3.8%

Коефіцієнт подібності 2:1%

Мікропробіли: 10

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-05-29 07:55:56.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-05-29 Доцент Андрій Нічепорук

Дата експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 11%**

	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
ID: 242343 Title: БКР Система керування шторами / жалюзі через ESP32 та мобільний застосунок Added in a DB: 2025-05-29 Authors: Владислав ТКАЧУК Heads: Олег САВЕНКО Consultants: Opponents:	97634	759	1748 (2%)	24 (3%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Ткачук Владислав Віталійович

Тема: Система керування шторами / жалюзі через ESP32 та мобільний застосунок

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3___ Кількість сторінок записки ___65___

1. Короткий зміст роботи та прийнятих рішень:

Метою кваліфікаційної роботи є розробка та моделювання системи керування шторами/жалюзі за допомогою мікроконтролера ESP32 та мобільного застосунку. У роботі реалізовано зв'язок між ESP32 та мобільним застосунком, що дозволяє дистанційно керувати положенням штор/жалюзі. У якості засобу реалізації програмної частини використано середовище розробки для мобільних додатків (Expo/React Native) та прошивку для ESP32 з підтримкою Wi-Fi з'єднання.

2. Висновок про відповідність роботи дипломному завданню:

Робота повністю відповідає поставленому завданню, охоплюючи як апаратну, так і програмну складову системи.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи:

У першому розділі проведено дослідження предметної області, проаналізовано сучасні технології автоматизованого керування шторами/жалюзі, обрано технічні засоби реалізації системи (ESP32, сервоприводи, сенсори, мобільний застосунок). У другому розділі виконано розробку системи: створено схему підключення елементів до ESP32, розроблено програмне забезпечення для контролера, налагоджено передачу даних між контролером та мобільним застосунком через Wi-Fi. Третій розділ присвячено розробці мобільного застосунку, який забезпечує інтерфейс користувача для керування шторами, а також реалізовано обмін даними з мікроконтролером у реальному часі. У роботі застосовано сучасні підходи до

розробки вбудованих систем та мобільних додатків, використано бібліотеки та фреймворки, що активно використовуються у промисловій розробці.

4. Позитивні сторони роботи:

Робота має високу практичну цінність, демонструє інтеграцію апаратної та програмної складових, є прикладом реалізації IoT-проєкту з актуальним застосуванням у побуті.

5. Негативні сторони роботи:

У роботі недостатньо глибоко розглянуто аспекти енергозбереження пристрою та безпеки передачі даних у бездротовому середовищі.

6. Оцінка графічного оформлення та пояснювальної записки роботи:

Пояснювальна записка оформлена відповідно до чинних стандартів, графічні матеріали подані чітко, з поясненням функціонального призначення кожного компонента.

7. Відгук про роботу в цілому:

Кваліфікаційна робота виконана на високому науково-технічному рівні, містить всі необхідні складові для повноцінного функціонування системи керування шторами через ESP32 та мобільний застосунок. Результати можуть бути впроваджені у побутові або офісні умови для автоматизації.8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Федратюк А. Т. д. ф.-м. н., професор, завідувач кафедри ІПЗ

"28" 05 2025 р. _____ (підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система керування шторами / жалюзі через ESP32 та мобільний застосунок

Автор: Владислав ТКАЧУК

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Олег САВЕНКО, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 3.8% і адресується до 35 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Олег САВЕНКО

Гарант ОП _____ Андрій НІЧЕПОРУК

Завідувач кафедри КІС _____ Ольга ПАВЛОВА

Завідувачу кафедри КІІС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Владислава ТКАЧУКА
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

28.05 2025 року