

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

**Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

До захисту допущено:

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«___» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Системне програмування та спеціалізовані комп'ютерні системи»

спеціальності 123 «Комп'ютерна інженерія»

на тему: «Система освітлення на основі STM32»

Виконала:

студентка IV курсу, групи KB-11

Зубенко Марія Олексіївна _____

Керівник:

доц.каф. СПіСКС, к.т.н., с.н.с

Боярінова Юлія Євгенівна _____

Консультант з нормоконтролю:

доц. каф. СПСКС, к.т.н., доцент,

Клятченко Ярослав Михайлович _____

Рецензент:

ст. викл. кафедри ПЗКС, к. т. н.

Хіцко Яна Володимирівна _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра системного програмування і
спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студентки

Зубенко Марії Олесіївни

1. Тема проєкту «Система освітлення на основі STM32», керівник проєкту доц.каф. СПіСКС, к.т.н., с.н.с. Боярінова Юлія Євгенівна, затверджені наказом по університету від «___» _____ 20__ р. № _____
2. Термін подання студентом проєкту «___» червня 2025 р.
3. Вихідні дані до проєкту див. Технічне завдання
4. Зміст пояснювальної записки:
 - Аналіз існуючих рішень та обґрунтування теми дипломного проєкту;
 - Обрані програмні та апаратні засоби;
 - Опис розробленого пристрою автоматизації керування освітленням;
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):
 - Схема електрична функціональна;

- Схема алгоритму оновлення освітлення на основі поточного режиму роботи;
- Схеми алгоритмів визначення поточної яскравості навколишнього середовища та оновлення освітлення;
- Схеми алгоритмі запуску та обробки запитів WI-FI модулю;

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., к.т.н., доцент кафедри СПСКС		

7. Дата видачі завдання 22 грудня 2024 року.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення літератури за тематикою проекту	15.10.2024	
2.	Розроблення та узгодження технічного завдання	31.01.2025	
3.	Аналіз існуючих рішень	28.02.2025	
4.	Підготовка матеріалів першого розділу дипломного проекту	01.04.2025	
5.	Підготовка матеріалів другого розділу дипломного проекту	15.04.2025	
6.	Підготовка матеріалів третього розділу дипломного проекту	01.05.2025	
7.	Підготовка графічної частини дипломного проекту	20.05.2025	
8.	Оформлення документації дипломного проекту	28.05.2025	
9.	Попередній огляд матеріалів диплому на кафедрі	30.05.2025	

Студент

Марія ЗУБЕНКО

Керівник

Юлія БОЯРІНОВА

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (67 с., 21 рис. 3 табл., 3 додатки).

Об'єкт розробки – створення системи освітлення, яка дозволяє виконувати керування та автоматизацію освітлювальної системи.

Система керування дозволяє: виконувати керування бездротовими каналами зв'язку; налаштовувати автоматизовані сценарії керування; автоматично коригувати кількість світла в залежності від освітлення приміщення; Передбачена можливість зміни режиму роботи у разі потреби. В процесі розробки були використані такі компоненти, як плата STM32F411CEU6, Wi-Fi модуль ESP8266 ESP-01S, а також фоторезистор GL5516.

В ході розробки:

- проведено аналіз методів побудови існуючих систем для керування освітленням приміщення;
- сформульовані вимоги до системи керування освітленням;
- розроблена система керування освітленням;
- розроблено програмне забезпечення мікроконтролера для роботи з датчиком кількості освітлення.
- розроблено програмне забезпечення мікроконтролера для роботи з прийомопередатчиком ESP8266 для керування системою освітлення

Впровадження цієї системи в приміщенні сприятиме зниженню споживання електроенергії завдяки автоматичному регулюванню рівня освітлення, а використання сценаріїв автоматизації усуне потребу в ручному керуванні освітленням.

Ключові слова:

СИСТЕМА КЕРУВАННЯ ОСВІТЛЕННЯМ, STM32, ОСВІТЛЕНІСТЬ, МІКРОКОНТРОЛЕР, ДАТЧИК, АВТОМАТИЗАЦІЯ

ABSTRACT

The qualification work includes an explanatory note (67 p., 21 fig. 3 tables, 3 appendices).

The object of development is the creation of a lighting system that allows you to control and automate the lighting system.

The control system allows you to: control wireless communication channels; configure automated control scenarios; automatically adjust the amount of light depending on the lighting in the room; The possibility of changing the operating mode if necessary is provided. The following components were used in the development process: the STM32F411CEU6 board, the ESP8266 ESP-01S Wi-Fi module, as well as the GL5516 photoresistor.

During the development:

- an analysis of methods for building existing systems for controlling room lighting was carried out;
- requirements for the lighting control system were formulated;
- a lighting control system was developed;
- microcontroller software was developed to work with the lighting sensor.
- microcontroller software has been developed to work with the ESP8266 transceiver to control the lighting system

The implementation of this system in the room will help reduce electricity consumption due to automatic adjustment of the lighting level, and the use of automation scenarios will eliminate the need for manual lighting control.

Keywords:

LIGHTING CONTROL SYSTEM, STM32, LIGHTING, MICROCONTROLLER, SENSOR, AUTOMATION

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.466550.002 ТЗ	Система освітлення на основі STM32 Технічне завдання	5		
	A4	ІАЛЦ.466550.003 ТП	Система освітлення на основі STM32 Відомість технічного проекту	1		
	A4	ІАЛЦ.466550.004 ПЗ	Система освітлення на основі STM32 Пояснювальна записка	67		
	A4	ІАЛЦ.466550.005 Е2	Схема електрична функціональна Схема функціональна	1		
	A4	ІАЛЦ.466550.006 Д1	Схема алгоритму оновлення освітлення на основі поточного режиму роботи Схема алгоритму	1		

					ІАЛЦ.466550.001 ОА					
Змін.	Арк.	№ докум.	Підпис	Дата	Опис альбому <i>Система освітлення на основі STM32</i>					
Розробив	Зубенко М.О.							Літ.	Аркуш	Аркушів
Перевірив	Боярінова Ю.Є.								1	2
Консулт.								КПІ ім. Ігоря Сікорського, ФПМ КВ-11		
Н. контроль	Клятченко Я.М.									
Зав. каф.	Романкевич В.О.									

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до програмного продукту, що розробляється	2
5.2. Вимоги до апаратного забезпечення.....	3
5.3 Вимоги до програмного та апаратного забезпечення користувача	3
6. ЕТАПИ РОЗРОБКИ	5

					ІАЛЦ.466550.002 ТЗ			
Змін.	Арк.	№ докум.	Підпис	Дата				
Розробив		Зубенко М.О.			<i>Система освітлення на основі STM32</i>	Літ.	Аркуш	Аркушів
Перевірив		Боярінова Ю.Є.					1	5
Н. контроль		Клятченко Я.М.			Технічне завдання			
Затвердив		Романкевич В.О.						КПІ ім. Ігоря Сікорського, ФПМ КВ-11

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Система освітлення на основі STM32».

Галузь застосування: інформаційні технології, комп'ютерна інженерія.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проєктування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є покращення контролювання кількості освітлення завдяки системи керування, яка дозволить користувачам збільшити рівень комфорту в приміщенні.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет. Було використано документацію для сімейства мікроконтролерів STM32 та для WI-FI модулю.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- можливість самостійного встановлення системи користувачем;
- можливість зміни режимів роботи системи;
- можливість налаштування сценаріїв автоматизації;
- можливість керування системою;

- можливість підключення до системи регулювання сну (фітнес-браслет) для зміни кількості освітлення в залежності від фази сну користувача, а також для реагування на прокидання.

5.2. Вимоги до апаратного забезпечення

- мікроконтролер STM32F411CEU6, а також інші варіації мікроконтролерів з сімейства STM32, які мають достатню кількість пінів, а також інтерфейсів UART, SPI та I2C для монтажу та підключення сторонніх компонентів, необхідних для функціонування системи.

- пам'ять - вбудована пам'ять для зберігання прошивки системи, яка не є меншою за потрібну для коректного функціонування системи

- наявність доступу до мережі Internet, яке буде забезпечуватись за допомогою WI-FI модулю ESP8266;

- датчик кількості освітлення, який буде здійснювати контроль за навколишнім середовищем і передавати актуальний стан для визначення режиму роботи.

5.3 Вимоги до програмного та апаратного забезпечення користувача

- забезпечення стабільного живлення, яке може виконуватись за допомогою зовнішньої батареї або живлення за допомогою USB;

- надання системі доступу до інтернету завдяки підключенню до мережі WI-FI;

- програмне забезпечення має суміщатись з операційними системами, які можуть бути використані для розробки на мікроконтролерах STM32;

- для створення програмного забезпечення для STM32 має бути підтримка мови програмування, яка буде дозволяти виконати якісну розробку, переважно для таких цілей використовується C або C++. Для покращення та оптимізації роботи коду може бути інтегрована підтримка асемблера;

- для створення прошивки мікроконтролера та завантаження її безпосередньо в систему, потрібні інструменти для виконання компіляції та прошивки мікроконтролера;

- для знаходження та виправлення можливих помилок в роботі системи, повинна бути можливість використання дебагерів та моніторингових систем;

- для роботи з різними системами STM32, програмне забезпечення має включати в себе готові бібліотеки для роботи з інтерфейсами зв'язку(I2S, UART, SPI), системою переривань, регістрами, тощо;

- для підтримки та подальшої розробки програмного забезпечення, необхідно інтегрувати документацію, яка буде описувати логіку роботи системи, містити приклади та пояснення коду, а також інші корисні матеріали.

					ІАЛІЦ.466550.002 ТЗ	Аркуш
Змін.	Арк.	№ докум.	Підпис	Дата		4

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	15.10.2024
2.	Розроблення та узгодження технічного завдання	31.01.2025
3.	Аналіз існуючих рішень	28.02.2025
4.	Підготовка матеріалів першого розділу дипломного проекту	01.04.2025
5.	Підготовка матеріалів другого розділу дипломного проекту	15.04.2025
6.	Підготовка матеріалів третього розділу дипломного проекту	01.05.2025
7.	Підготовка графічної частини дипломного проекту	20.05.2025
8.	Оформлення документації дипломного проекту	28.05.2025
9.	Попередній огляд матеріалів диплому на кафедрі	30.05.2025

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466550.002 ТЗ

Аркуш

5

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.466550.004 ПЗ	Система освітлення на основі STM32	67		
			Пояснювальна записка			
	A4	ІАЛЦ.466550.005 Е2	Схема електрична функціональна	1		
			Схема функціональна			
	A4	ІАЛЦ.466550.006 Д1	Схема алгоритму оновлення освітлення на основі поточного режиму роботи	1		
			Схема алгоритму			
	A4	ІАЛЦ.466550.007 Д2	Схеми алгоритмів визначення поточної яскравості навколишнього середовища та оновлення освітлення	1		
			Схема алгоритму			
	A4	ІАЛЦ.466550.008 Д3	Схеми алгоритмі запуску та обробки запитів WI-FI модулю	1		
			Схема алгоритму			
		Диск CD_ROM	Текст пояснювальної записки	1		
			Графічний матеріал			

					ІАЛЦ.466550.002 ТП		
Змін.	Арк.	№ докум.	Підпис	Дата			
Розробив		Зубенко М.О.			Літ.	Аркуш	Аркушів
Перевірив		Боярінова Ю.Є.				1	1
Консульт.					Відомість технічного проекту КПІ ім. Ігоря Сікорського, ФПМ КВ-11		
Н. контроль		Клятченко Я.М.					
Зав. каф.		Романкевич В.О.					

Пояснювальна записка до дипломного проекту
на тему: «Система Освітлення на основі STM32»

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП.....	4
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ	5
1.1 Існуючі рішення на ринку	5
1.2 Розумні лампи та системи	5
1.2.1 Philips Hue	5
1.2.2 Xiaomi Mi Smart LED.....	6
1.2.3 Доступні альтернативи.....	7
1.2.4 Системи розумного будинку	8
1.3 Аналіз та недоліки існуючих рішень	9
1.4 Обґрунтування теми диплому	10
2. ОБРАНІ ПРОГРАМНІ ТА АПАРАТНІ ЗАСОБИ	12
2.1 Мікроконтролер STM32F411	12
2.1.1 Інтерфейс I2C	14
2.1.2 Інтерфейс SPI.....	16
2.1.3 Інтерфейс USART	18
2.1.4 ADC (Analog-to-Digital Converter).....	20
2.1.5 FreeRTOS	21
2.2 WIFI модуль ESP8266 ESP-01S	23
2.3 Фоторезистор GL5516	25
2.4 RGB LED	26
2.5 Мова програмування C	26
2.6 Програмне забезпечення для мікроконтролера	27
2.6.1 STM32CubeMX	27
2.6.2 STM32CubeIDE	28
2.6.3 STM32CubeProgrammer.....	29
2.7 Бібліотека HAL.....	29

2.8 Git.....	30
3. ОПИС РОЗРОБЛЕНОГО ПРИСТРОЮ АВТОМАТИЗАЦІЇ КЕРУВАННЯ ОСВІТЛЕННЯМ.....	32
3.1 Загальна архітектура та архітектура програми.....	32
3.1.1 Загальна схема апаратної системи.....	32
3.1.2 Архітектура програмного забезпечення: основні блоки, логіка взаємодії.....	35
3.1.3 Організація основної задачі ПЗ через FreeRTOS.....	38
3.2 Налаштування периферії.....	39
3.2.1 Налаштування АЦП.....	39
3.2.2 Налаштування PWM для керування яскравістю RGB LED.....	40
3.2.3 Налаштування та використання USART.....	41
3.3 Розроблена система та її функціональність.....	43
3.3.1 Огляд режимів роботи.....	43
3.3.2 Логіка перемикання між режимами.....	44
3.3.3 Обробка команд від користувача.....	44
3.4 Огляд та аналіз отриманих під час функціонування системи результатів.....	45
3.4.1 Вимірювання реакції системи на зміну освітленості.....	45
3.4.2 Вимірювання часу комунікації.....	46
3.4.3 Аналіз стабільності з'єднання через ESP8266.....	49
3.5 Аналіз ефективності розробленої системи.....	50
3.6 Аналіз можливостей застосування освітлювальної системи.....	52
3.6.1 Застосування у побуті (система розумний дім).....	52
3.6.2 Використання для людей з особливими потребами.....	53
3.6.3 Використання в офісах, готелях, лікарнях та іншій інфраструктурі..	54
3.6.4 Переваги автоматичного освітлення над звичайним.....	57
3.7 Опис можливостей подальшого розвитку та вдосконалення системи.....	58

3.7.1 Додати Bluetooth	58
3.7.2 Підключення до Smart Home-платформ	59
3.7.3 Вдосконалення програмної частини	60
3.7.4 Створення мобільного застосунку	61
3.7.5 Розширення кількості сенсорів.....	62
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65

ДОДАТКИ

Додаток А. Фрагменти програмного коду.

Додаток Б. Структурні схеми програмних модулів.

Додаток В. Презентація дипломного проєкту

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ПЗ – програмне забезпечення

ADC - Analog-to-digital converter (аналогово-цифровий перетворювач)

C – мова програмування високого рівня

Git - система контролю версій

GPIO - General Purpose Input/Output (універсальний інтерфейс вводу/виводу)

HAL - Hardware Access Level, бібліотека для роботи з периферією

I2C - inter-Integrated Circuit (міжінтегральна схема)

LED - світлодіод, напівпровідниковий пристрій, що випромінює світло

Rtos - операційна система реального часу

SCL - serial clock, послідовна лінія тактування

SDA - serial data, послідовна лінія даних

SPI - Serial Peripheral Interface (синхронний послідовний інтерфейс)

STM32 - мікроконтролер, що належить родині STM32.

VScode, CubeIDE - середовище розробки додатку

UART - Universal Asynchronous Receiver-Transmitter (універсальний асинхронний прийомо-передатчик)

USB-порт - стандарт роз'ємів і кабелів для передачі даних та живлення

ВСТУП

Світлотехнічні системи відіграють важливу роль у забезпеченні комфорту, енергоефективності та автоматизації житлових і робочих просторів. У сучасних умовах розвитку розумних технологій виникає потреба у створенні більш гнучких і адаптивних систем керування освітленням, здатних підлаштовуватися під зміну зовнішніх умов та індивідуальні потреби користувача. Система має бути гнучкою та зручною у використанні, зменшувати навантаження на користувача та дозволяти полегшити та автоматизувати побут.

В дипломному проєкті розроблена система контролю освітленості, яка підтримує кілька режимів роботи. Система може працювати в стандартному режимі постійного освітлення, автоматично регулювати яскравість залежно від рівня природного освітлення, а також взаємодіяти з фітнес-браслетом користувача для вмикання або вимикання світла на основі фаз сну та пробудження. Такий підхід дозволяє забезпечити оптимальні умови для відпочинку та активної діяльності. Окрім того, такий функціонал позбавляє користувача необхідності постійного контролювання освітлення в приміщенні, що зменшує навантаженість, а також сприяє економії електроенергії

Потреба у подібних рішеннях постійно зростає. Значна кількість людей відчуває дискомфорт у повній темряві, особливо діти, літні люди або особи з підвищеною тривожністю. Для них можливість м'якого автоматизованого освітлення під час пробудження чи засинання є важливою складовою психологічного комфорту і безпеки. Крім того, серед цільової аудиторії також можна виділити користувачів розумного дому, які прагнуть оптимізувати витрати електроенергії без зниження рівня особистого комфорту.

					ІАЛЦ.466550.004 ПЗ	Аркуш
Змін.	Арк.	№ докум.	Підпис	Дата		6

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Існуючі рішення на ринку

Системи управління освітлення представлені на ринку в великій кількості, мають різноманітні способи встановлення, кількість функціоналу та вартість продукту, що зумовлено різноманітними потребами та грошовими можливостями великої кількості користувачів. Розглянемо декілька категорії представників таких систем.

1.2 Розумні лампи та системи

Одними з найпопулярніших представників систем керування освітленням є розумні лампи та нічники, які мають багато різноманітних варіантів, які можуть задовольнити потреби великого прошарку користувачів. Серед плюсів, які можна виділити, вони мають: легке встановлення, яке не потребує професійного обладнання чи навичок; простий та інтуїтивно зрозумілий інтерфейс, який не створить складнощів в керуванні та налаштуванні; цінова та функціональна різноманітність, яка дозволяє підібрати рішення майже під будь-який запит.

1.2.1 Philips Hue

Philips Hue (рис.1.1)— це серія девайсів для керування освітленням, яка включає в себе смарт-лампи, світильники, стрічки та багато іншого. Основною ідеєю цієї серії є керування світлом використовуючи смартфон, голосові команди чи автоматичні сценарії. Має широкий функціонал, який складається з плавного регулювання яскравості, зміни кольору освітлення, автоматичне ввімкнення/вимкнення за часом, голосове керування та дистанційне керування за допомогою смартфона. Представники цієї системи відносяться до преміум сегменту, тому їх вартість може бути невідповідною для певної верстви населення[1].



Рисунок 1.1 - Philips Hue

1.2.2 Xiaomi Mi Smart LED

Xiaomi Mi Smart LED (рис.1.2) - це смарт-лампа, керування якою відбувається через додаток завдяки підключенню до Wi-Fi. Вона підтримує як біле світло з регулюванням яскравості, так і кольорове освітлення (у моделей RGB). Крім функціоналу, який є у моделі, яка описана вище, має також налаштування кольорової температури, яке коливається від теплого жовтого до яскравого білого, а також можливість інтеграції в систему розумного будинку. Ця лінійка продуктів відноситься до середнього цінового сегменту, тому буде доступна більшій кількості користувачів і відповідно має більший попит на ринку, оскільки не дивлячись на нижчу якість, функціонал дублює преміум сегмент[2].



Рисунок 1.2 - Xiaomi Mi Smart LED

1.2.3 Доступні альтернативи

Крім всіх пристроїв, про які мова йшла раніше, на ринку також представлений доволі бюджетний прошарок освітлювальних виробів, який доступний майже кожному, хто бажає полегшити собі життя завдяки автоматизації керування освітленням. Їх функціонал менший, а якість нижча, проте встановлення та керування так само легке. Наприклад, дуже бюджетний нічник від українського бренду RZTK(рис.1.3). Цей виріб має значно менший функціонал: датчик руху, який дозволяє йому автоматично вмикатися, і переключення світла між фіксованими значеннями яскравості, що не дозволить користувачу отримати великий рівень персоналізації освітлення, проте якщо це не є принциповим і рівень вимог не такий високий, цей бюджетний варіант стане гарним рішенням. Крім того, на ринку представлені моделі, які автоматично регулюють яскравість, базуючись на зовнішній кількості освітлення.



Рисунок 1.3 – Світильник RZTK

1.2.4 Системи розумного будинку

Різні пристрої можуть бути об'єднані в спільну екосистему, і часткою такої екосистеми можуть бути прилади для керування освітлення. Прикладом такого інтегрування керування автоматизацією в систему може бути LightSwitch(рис.1.4) від компанії Ajax Systems. Девайс представлений в великій кількості варіантів, наприклад, звичайний LightSwitch, який може налаштовуватись як пристрій автоматизації і вмикати/вимикати світло. Також є LightSwitch Dimmer, який окрім автоматизації містить в собі функціонал регулювання яскравості, який також можна додати в сценарії керування освітленням. Девайси інтегруються в охоронну систему, яка є основною діяльністю Ajax Systems, і дозволяють користувачу налаштовувати освітлення дистанційно за допомогою додатку, або власноруч[3].

підключення їх до фітнес-браслетів або смарт-годинників для регуляції освітлення на основі засинання і прокидання користувача, то можна побачити, що такі рішення майже відсутні на ринку. Ті ж пристрої, які зараз доступні для користувачів, знаходяться в преміальному ціновому сегменті і доступні невеликій кількості користувачів. Крім того, налаштування деяких з них потребує додаткових пристроїв, наприклад, блок керування системою розумного будинку, який може не бути необхідним, якщо користувача цікавить тільки освітлювальна система без додаткових девайсів. Тому встановлення повноцінної системи розумного будинку під запит на керування освітленням на основі біоритмів не виглядає як рішення, яке підійде більшості користувачів.

1.4 Обґрунтування теми диплому

Враховуючи все, що було описано та проаналізовано вище, можна зробити висновок, що актуальність теми диплому зумовлена певним списком потреб, а саме:

1. Необхідність у інтелектуальній системі керування освітленням, яка матиме декілька режимів роботи і буде, в залежності від нього, або статично освітлювати, або адаптуватись до освітлення, або синхронізуватись з фізіологічним станом користувача;
2. Розширення функціоналу для існуючих рішень та комбінація функцій з різних систем для створення більш універсального освітлювального пристрою;
3. Зниження споживання електроенергії під час використання освітлювальної системи за рахунок адаптивного режиму роботи, який регулюватиме яскравість базуючись на зовнішньому освітленні;
4. Збільшення доступності системи для рядових користувачів за рахунок її автономності і незалежності від центру керування, необхідність в якому виникає у більшості існуючих освітлювальних систем;

5. Підвищення комфорту завдяки автоматизації освітлення на основі індивідуальних біоритмів користувача через плавне вмикання та вимикання світла при прокиданні та засинанні відповідно;

Таким чином, запропонована система керування освітленням є рішенням, яке має перспективу на ринку, оскільки враховує індивідуальні запити користувачів, поєднує в собі переваги та уникає недоліків вже існуючих пристроїв. Така освітлювальна система дозволить підвищити рівень комфорту в приміщенні, полегшити керування освітленням та зменшити використання електроенергії, а різноманітність режимів роботи дозволяє їй підлаштуватись під вимоги більшої кількості користувачів.

					ІАЛЦ.466550.004 ПЗ	Аркуш
Змін.	Арк.	№ докум.	Підпис	Дата		13

2. ОБРАНІ ПРОГРАМНІ ТА АПАРАТНІ ЗАСОБИ

2.1 Мікроконтролер STM32F411

Для реалізації системи керування освітленням було обрано мікроконтролер STM32F411 від компанії STMicroelectronics (рис. 2.1). Він належить до серії STM32F4, яка базується на ядрі ARM Cortex-M4 з апаратним модулем, який має підтримку обчислень з плаваючою комою. Це ядро ідеально підходить для цифрової обробки сигналів, і тому воно буде корисним для обробки даних, які будуть отримані з датчиків пристрою. Даний мікроконтролер має високий рівень енергоефективності, а також багатий набір периферійних інтерфейсів, і тому ідеально підходить для побудови вбудованих систем керування, зокрема таких, які потребують підключення до датчиків освітленості або бездротових модулів.

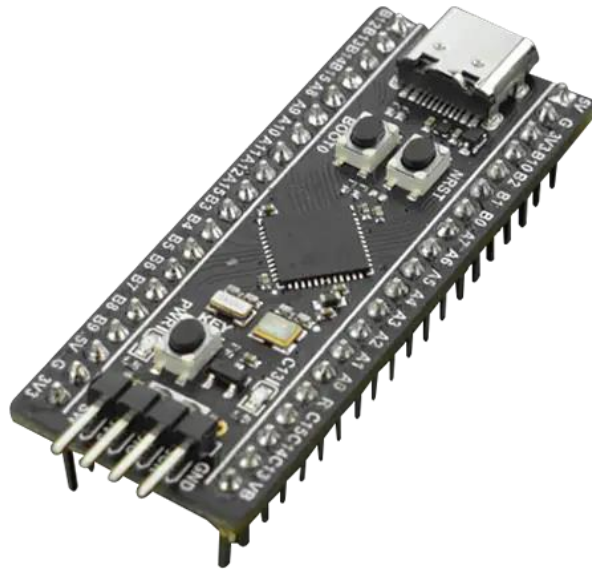


Рисунок 2.1 - Мікроконтролер STM32F411

Якщо роздивлятися його параметри, то окрім вже згаданого ядра та його особливостей, можна підкреслити те, що він має тактову частоту до 100 мегагерц, що дає необхідну кількість виконуваних за одиницю часу операцій, потрібних для коректної роботи системи. Також дана модель мікроконтролера оснащена Flash-пам'яттю об'ємом 512 кілобайтів, яка зберігає в собі інформацію без зовнішнього живлення і має високу швидкість доступу до даних.

Також має енергозалежну статичну оперативну пам'ять з довільним доступом, об'єм якої дорівнює 128 кілобайтам. Якщо ж дивитися на особливості роботи плати, то можна відзначити, що напруга живлення може бути в межах від 1.7 до 3.6 вольт, проте рекомендованою типовою напругою є 3.3 вольт. Живлення плати може здійснюватись за допомогою USB-порта або зовнішнього живлення від батарейки чи адаптера. Робоча температура має доволі широкий діапазон: від -40 градусів до +85 градусів по Цельсію.

Плата має велику кількість інтерфейсів та периферійних пристроїв, частина з яких буде розглянута нижче. Серед них:

1. GPIO, який має до 50 портів;
2. Напівдуплексний синхронний I2C;
3. Синхронні повнодуплексні USART та SPI;
4. USB для живлення та завантаження прошивки на плату;
5. DMA, який надає прямий доступ до пам'яті;
6. ADC для перетворення аналогових сигналів на цифрові.
7. Таймери в кількості 10 штук, зокрема PWM та таймер реального часу;

Дана модель мікроконтролера підтримує різноманітні операцій системи реального часу, зокрема FreeRtos або CMSIS-RTOS, що дозволяє легко створювати з їх допомогою програмне забезпечення з необхідним функціоналом та оперувати всіма наявними інтерфейсами та доданими компонентами.

Для оптимізації споживання енергії та досягнення максимально можливої продуктивності, обраний мікроконтролер має гнучку систему годинників (Clock system) - систему тактування, яка визначає частоту роботи процесора, периферії і інших блоків. Підтримка різних джерел тактового сигналу, а саме HSI, HSE та PLL дозволяє виконувати динамічне переключення між джерелами в залежності від того, який режим роботи потрібен зараз.

Модуль DMA (Direct Memory Access), який відповідає за надання прямого доступу до пам'яті, знижує навантаження на ядро за рахунок того, що завдяки йому дані між периферією та пам'яттю передаються без допомоги CPU.

Оскільки однією з вимог до розробленої системи була можливість налаштування яскравості освітлення, під час обрання мікроконтролера враховувалась наявність PWM, який дасть змогу контролювати потужність LED-індикатора. За допомогою сигналу PWM яскравість буде керуватись плавно, за простим принципом - чим вищий коефіцієнт заповнення, тим сильніше світить світлодіод.

З особливостей використання можна виділити наявність офіційного середовища розробки, яке підтримує генерацію коду, а саме STM32CubeIDE. Для того, щоб графічно налаштувати мікроконтролер та створити зручний шаблон, використовується офіційне середовище STM32CubeMX. Також, мікроконтролер підтримує бібліотеки HAL, які надають високорівневий або низькорівневий доступ до периферії. Важливим моментом є можливість налагодження за допомогою декількох інструментів, а саме OpenOCD, ST-Link та J-Link.

Отже, у висновку отримуємо плату, яка має високу продуктивність і в той же час низьке енергоспоживання і велику кількість вбудованих інтерфейсів та периферії, які корисні при виконанні проектів під велику кількість запитів. За допомогою наявної документації та підтримки від STMicroelectronics, а також широкої спільноти користувачів, цей мікроконтролер може бути легко використаний у промислових та побутових проектах[4].

2.1.1 Інтерфейс I2C

Інтерфейс I2C (Inter-Integrated Circuit) - це синхронний напівдуплексний послідовний інтерфейс, який було розроблено фірмою Philips. Його призначенням є обмін інформацією між периферійними пристроями та мікроконтролером, що знаходяться на малій відстані один від одного.

На обраній для розробки платі наявна підтримка трьох таких інтерфейсів, які мають можливість працювати на швидкості до 400 кілобіт на секунду, якщо мають стандартний режим роботи, або ж у швидкісному режимі перейти на швидкість до одного мегабіту на секунду.

Для передачі даних цей інтерфейс використовує шину, що складається з двох проводів, один з яких має назву провід даних - SDA, а інший - провід тактування - SCL. Третім провідником між пристроями виступає земля - GND. Принцип роботи шини заключається в тому що на ній присутній головний пристрій, назва якого відповідає терміну master, та один або декілька підлеглих пристроїв, які мають назву slave. Також, одна шина може містити декілька головних пристроїв[6].

Передача даних починається лише за ініціацією майстра, підлеглі девайси не можуть зробити запит на передачу. Комунікація відбувається за допомогою того, що головний пристрій на шині генерує тактовий сигнал, Через унікальну адресу підлеглих пристрої відбувається звернення до них і відповідно запис або вчитка даних з їх регістрів. Апаратно інтерфейс реалізовано таким чином, що формування сигналів відбувається через логічне І (рис. 2.2)

Вихід ведучого	Вихід веденого	Результат на шині
0	0	0
0	1	0
1	0	0
1	1	1

Рисунок 2.2 - Таблиця формування сигналів I2C

Так як шина може працювати як в режимі запису, так і читання, відповідно вона має два типи повідомлень, які відповідають за цей функціонал. Повна структура повідомлення складається з таких бітів (рис. 2.3):

1. Старт, який позначає початок передачі даних;
2. 7 біт адреси підлеглого пристрою;
3. Біт R/W, який визначає режим роботи для підлеглого пристрою, чи має він приймати (значення 0) чи передавати (значення 1);
4. Біти даних;
5. Біт, який вказує підтвердження або непідтвердження;
6. Стоп, який позначає кінець передачі даних;



Рисунок 2.3 - Графічне зображення структури повідомлення I2C

2.1.2 Інтерфейс SPI

Інтерфейс SPI (Serial Peripheral Interface) (рис 2.4) - це синхронний послідовний інтерфейс, є чотирьох-провідним і його призначення заключається в повнодуплексному обміні інформацією між керуючим та підлеглими пристроями. Периферійними пристроями, з якими по SPI зв'язується мікроконтролер, можуть бути датчики, оперативна та flash-пам'ять, аналогово-цифрові та цифро-аналогові перетворювачі та багато іншого.

Принцип роботи інтерфейсу заключається в тому, що керуючий пристрій генерує SCLK - тактовий сигнал синхронізації - для того щоб синхронізувати вхідні та вихідні дані. В цьому інтерфейсі вхідний та вихідний сигнали мають назви MOSI та MISO.

MOSI (Master Out Slave In) - для ведучого пристрою цей сигнал вихідний, для підлеглого - вхідний. Відповідно дані передаються від майстра до слейва. Протилежний йому сигнал MISO (Master In Slave Out), який відповідно є вхідний для керуючого та вихідний для керованого, і дані передаються від слейва до майстра.

Якщо на шині є можливість підключення більше ніж одного підлеглого пристрою до керуючого, на ній буде також сигнал SS або CS, які розшифровуються як Slave Select та Chip Select відповідно і призначені для вибору підлеглого пристрою, з яким керуючий хоче почати взаємодію. Для кожного ведомого пристрою керуючий має мати окремий сигнал, тоді як наприклад сигнал SCLK один для всіх.

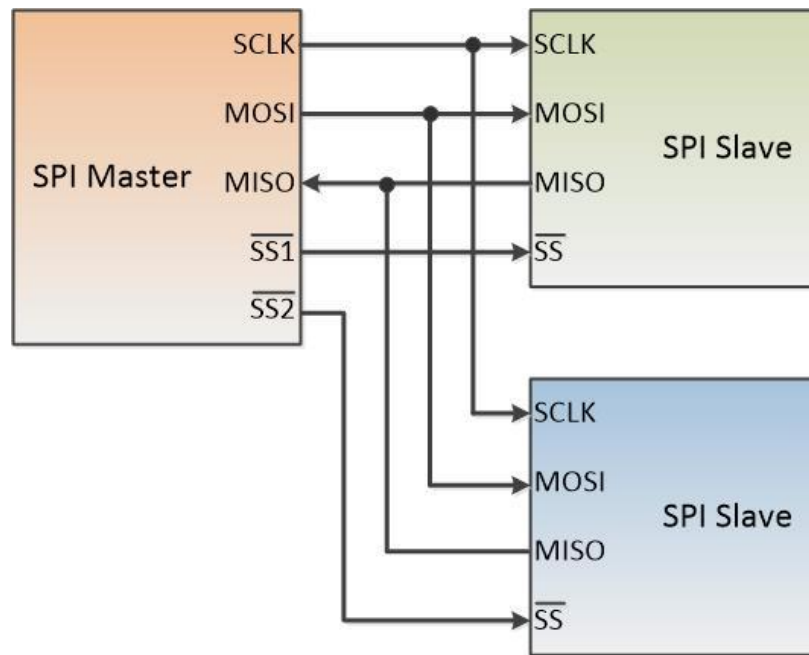


Рисунок 2.4 - Графічне зображення поєднання пристроїв через SPI

Передача даних відбувається пакетами, за один сеанс передачі в обох напрямках, від майстра до слейва та назад, передається один байт, одночасна передача робить SPI повнодуплексним інтерфейсом. Дані, яку будуть передаватись, перед початком процесу обміну інформацією поміщають в зсувні регістри. Наступним кроком керуючий пристрій починає генерацію синхронізаційних імпульсів на лінії SCLK, після чого починається передача даних. Обмін інформацією здійснюється поступово, біт за бітом, і у випадку якщо інформації більше ніж один байт, регістри перезавантажується і процес циклічно повторюється. Принцип роботи зображено на рисунку 2.5.

Проте цей процес передачі даних, не дивлячись на переваги у вигляді синхронної повнодуплексної роботи та достатньо високу швидкість передачі даних, яка може досягати 50 мегабіт на секунду, має і свої недоліки. Наприклад, потребує більшу кількість виводів ніж описаний раніше I2C і вимагає окремий сигнал для кожного підлеглого пристрою Також відсутні підтвердження прийому даних та протокол визначення помилок, що може призвести до ймовірної втрати даних[7].

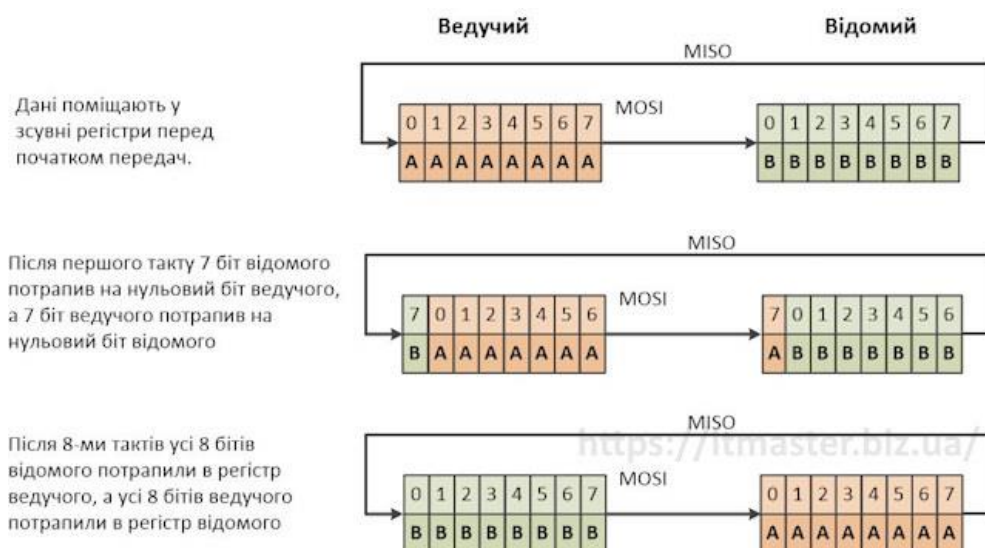


Рисунок 2.5 - Графічне зображення принципу передачі даних в SPI

2.1.3 Інтерфейс USART

USART (Universal Synchronous/Asynchronous Receiver/Transmitter) - це протокол синхронного універсального приймача/передавача, який полягає в наборі правил для послідовного обміну даними між двома пристроями. На відміну від UART, асинхронного протоколу в якому інформація передається за допомогою лише двох дротів, цей вид протоколу має додатковий вивід синхронізації (рис 2.6).

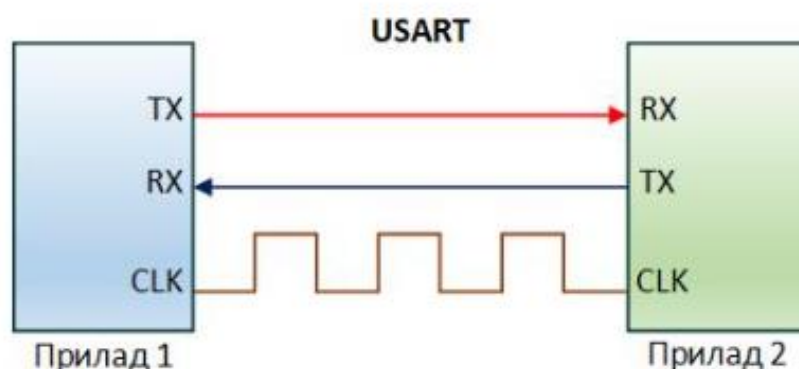


Рисунок 2.6 - Графічне зображення поєднання пристроїв через USART

USART в системах STM32 має певний набір основних характеристик, серед яких можна виділити такі:

1. При частоті тактування 82 мегагерци, передачі може відбуватись зі швидкістю до 10,5 мегабіт на секунду;

2. Дані мають розмірність 8 або 9 біт;
3. Наявна підтримка фізичного рівню інфрачервонного порту;
4. Обмін даних може відбуватись завдяки використанню контролера прямого доступу до пам'яті;
5. Режим роботи може бути як синхронний, так і асинхронний;

Сигнали кожного пристрою USART мають назви Передавач (Tx) та Приймач (Rx), які перехрестно поєднуються для обміну даними між двома мікроконтролера. Також, для підтримки на обох пристроях однакової напруги на логічних рівнях, використовується провід землі, він же GND.

Передача інформації за допомогою USART відбувається в кадрах, які в свою чергу складаються з бітів. Кадр складається з початкового біту, бітів інформації, додаткового біту парності, а також одного або двох бітів зупинки. Передачі кожного з бітів відбувається за однаковий проміжок часу, який ще має назву тайм-слот[8].

Початок передачі оголошується через спад сигналу на лінії, оскільки в період так званого простою сигнал тримається в високому положенні. Ця зміна сигналу виступає командою для приймача, що зараз почнеться передача даних у вигляді ряду нулів та одиниць. Закінчення передачі відбувається після передачі восьмого біта за допомогою стоп-сигналу високого рівню.

Швидкість передачі залежить від того, в якому режимі працює інтерфейс. В мікроконтролері, який був обраний для розробки, є підтримка обидвох цих видів роботи. В синхронному режимі USART працює по принципу, який схожий на роботу інтерфейсу SPI. Завдяки цьому, швидкість може досягати декількох мегабіт в секунду, в той час як в асинхронному режимі швидкість передачі зазвичай встановлюється в 115200 бітів за секунду, хоча при гарному та точному налаштуванні можна досягти швидкості в один мегабіт на секунду[9].

2.1.4 ADC (Analog-to-Digital Converter)

ADC (аналого-цифровий перетворювач) використовується для того, щоб отримувати аналогові сигнали від датчиків, розташованих на платі, і перетворювати їх в цифрові. На мікроконтролері STM32F411 наявний дванадцятибітний перетворювач, який має шістнадцять каналів, і це дозволяє йому забезпечувати високу точність. Основними його характеристиками є:

1. Роздільна здатність в дванадцять біт, яка забезпечує дискретні рівні в кількості 4096 штук;
2. Шістнадцять мультиплексних каналів, серед яких є внутрішні джерела;
3. Частота дискретизації, яка досягає 2.4 мега семплів за секунду;
4. Підтримка різних режимів роботи;
5. Можливість керувати тригерами запуску, використовуючи програмне забезпечення та зовнішні події, наприклад, таймери;
6. Передача результатів замірів відбувається автоматично, і записується в пам'ять напряму, в обхід центрального процесора;
7. Контроль перевищення заданих порогів, який підтримує можливість генерувати переривання;

Режими роботи наявні в кількості чотирьох штук і мають такі особливості:

1. Одиночний, який виконує перетворення для обраного одного каналу;
2. Безперервний, який виконує постійне перетворення конкретного каналу;
3. Сканування, яке відбувається шляхом перетворення декількох каналів послідовно;
4. Інжектований, який використовується для того щоб генерувати високопріоритетне перетворення, яке перериває регулярно;

На платі STM32F411 використовується АЦП з принципом роботи, який має назву SAR (Successive Approximation Register), послідовне наближення. Для перетворення аналогового сигналу в цифровий, АЦП по черзі “вмикає” один за одним біти в порядку від старшого до молодшого, і таким чином збирає найбільш точне представлення сигналу. Приклад перетворення графічно зображено на рисунку 2.7.

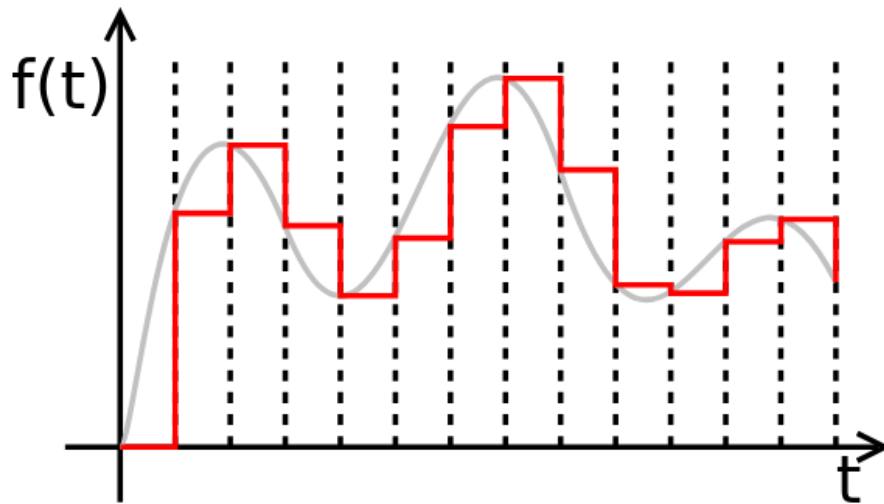


Рисунок 2.7 - Графік перетворення аналогового сигналу в цифровий

Цей вид АЦП має велику кількість переваг, а саме високу швидкість та точність перетворень, що дозволяє обробляти навіть швидкозмінні аналогові сигнали. Налаштування доволі гнучке, відповідно можлива адаптація до великої кількості різних типів сигналів і джерел. Інтеграція із сторонніми модулями відбувається доволі легко, що дає можливість створити ефективну систему для того, щоб збирати дані та обробляти їх. А можливість автономного режиму роботи з низьким енергоспоживанням дозволяє виконувати автоматичний запуск перетворень[5].

2.1.5 FreeRTOS

FreeRTOS (Free Real-Time Operating System) - це операційна система реального часу, яка має відкриту кодову базу, підтримує багатозадачність, і до того ж є абсолютно безкоштовною. Розроблена вона була спеціально для роботи з мікроконтролерами та вбудованими системами. Її гнучкість та ефективність дозволяє широко використовувати її як в промислових, так і в особистих проектах.

Дана операційна система майже повністю написана з використанням мови програмування C, присутні асемблерні вставки але їх кількість мінімальна. Присутня підтримка трьох типів багатозадачності, а саме витісняючої,

кооперативної та гібридної. Малий розмір ядра займає невелику кількість пам'яті, приблизно від восьми до дванадцяти кілобайт[10].

Окрім підтримки завдань, які також мають назву таски, для мікроконтролерів з малим обсягом оперативної пам'яті спеціально було створені співпрограми (co-routines). Окрім цього, присутня модульність, яка дозволяє включати лише необхідні компоненти, що суттєво зменшує розмір прошивки. В цій операційній системі відсутні програмні обмеження на кількість виконуваних задач та їх пріоритетів, а використання пріоритетів також не має обмежень - декілька різних завдань можуть мати абсолютно однаковий пріоритет.

Засоби синхронізації добре розвинені, для реалізації синхронізації “задача - задача” та “задача - переривання” наявні такі засоби як черга, двійковий семафор, лічильниковий семафор рекурсивний семафор та м'ютекс. М'ютекси в свою чергу мають здатність наслідувати пріоритет.

Одним з основних компонентів FreeRTOS є таски - незалежні один від одного потоки виконання, які працюють паралельно в межах однієї програми. Кожна таска має свій власний стек, має пріоритет - чим вищий, тим швидше вона планується, і може знаходитись у різних станах, а саме:

1. Running - задача знаходиться в процесі виконання;
2. Ready - задача повністю готова до виконання;
3. Blocked - очікує на певну подію, наприклад, затримку чи сигнал;
4. Suspended - задача тимчасово призупинена;
5. Deleted - задача видалена;

Система зміни станів графічно зображена на рисунку 2.8.

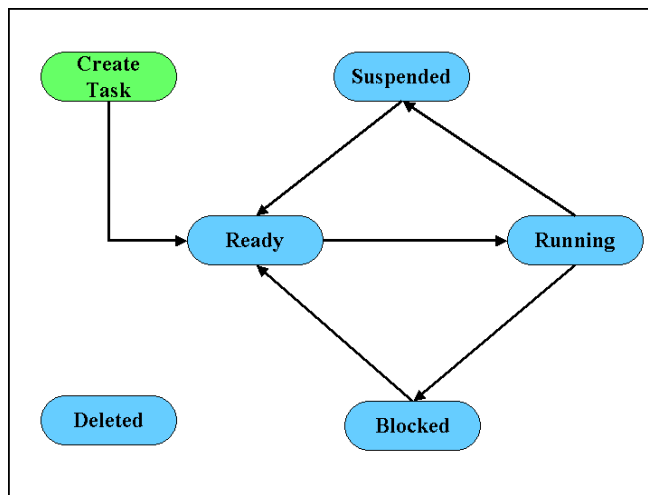


Рисунок 2.8 - Графічне зображення системи зміни станів тасок у FreeRTOS

Виконання завдань в операційній системі відбувається завдяки планувальнику (Scheduler), який приймає рішення, яка саме таска буде виконуватись в даний момент часу. Для реалізування багатозадачності присутні декілька типів планування, одне з яких витісняюче - при ньому задача, яка має вищий пріоритет, витісняє поточну, якщо вона нижча за пріоритетом, і починає своє негайне виконання. Інший тип має назву кооперативний, і в ньому поточне завдання самостійно передає контекст наступному, використовуючи програмне забезпечення. Для оновлення часу очікування та активації завдань планувальник використовує тік-таймер, де найменша одиниця часу відповідно має назву тік[11].

Таким чином, операційна система FreeRTOS є ідеальним рішенням для створення програмного забезпечення для мікроконтролерів, оскільки містить в собі велику кількість функціоналу, необхідного для керування пристроями та датчиками, має відкриту кодову базу, документацію та велику спільноту користувачів, і дозволяє вирішити задачі будь-якої складності за рахунок можливості легкого інтегрування стороннього програмного забезпечення.

2.2 WIFI модуль ESP8266 ESP-01S

Модуль ESP8266 ESP-01S (рис. 2.9) - це компактний Wi-Fi-модуль, який було створено на базі мікроконтролера ESP8266EX. Цей мікроконтролер має в наявності велику кількість модулів на своїй основі, і на відміну від його попередніх версій в наявності власна flash-пам'ять, що не призводить до необхідності в користувацьких програмах використовувати зовнішні мікросхеми, які мають в своїй комплектації flash-пам'ять і полегшує роботу з модулем. Модуль має всю необхідну для підключення та комунікації периферію, а саме GPIO, SPI, I2C, UART, PWM та АЦП на десять бітів.

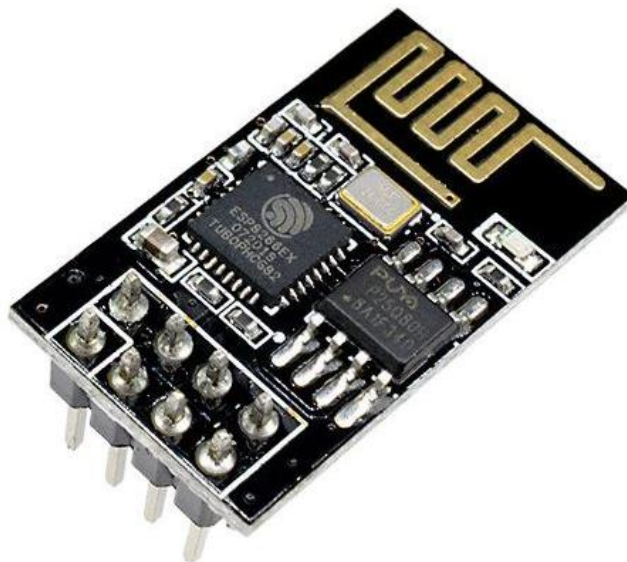


Рисунок 2.9 - WIFI модуль ESP8266

Модуль має тридцяти двох бітний процесор, частота роботи якого має межі від вісімдесяти до ста шістидесяти мегагерц. Середній споживаний струм рівний приблизно вісімдесяти міліамперах, проте це значення може змінюватись в залежності від режиму роботи. Наприклад, споживання може піднятись до двохсот п'ятнадцяти міліампер в режимі передачі та ста міліампер в режимі приймання. Напруга живлення може коливатись від 2.5 до 3.6 вольт, проте найчастіше на нього подають 3.3, які є так званим “золотим стандартом”.

Окрім того, щоб використовувати цей модуль як WIFI адаптер, який додає можливість бездротового з'єднання до мікроконтролеру з використанням UART, він може бути використаний також як самостійний контролер, який може використовувати найпростіші команди і без додаткового під'єднаних пристроїв. Використання його як модулю IoT часто застосовується в розумних пристроях автоматизації, так само, як він використаний в розробленій системі освітлення[12].

2.3 Фоторезистор GL5516

Фотоелектричний аналоговий сенсор GL5516 (рис. 2.10) - це світлочутливий резистор, у якого в залежності від рівня освітленості змінюється опір.

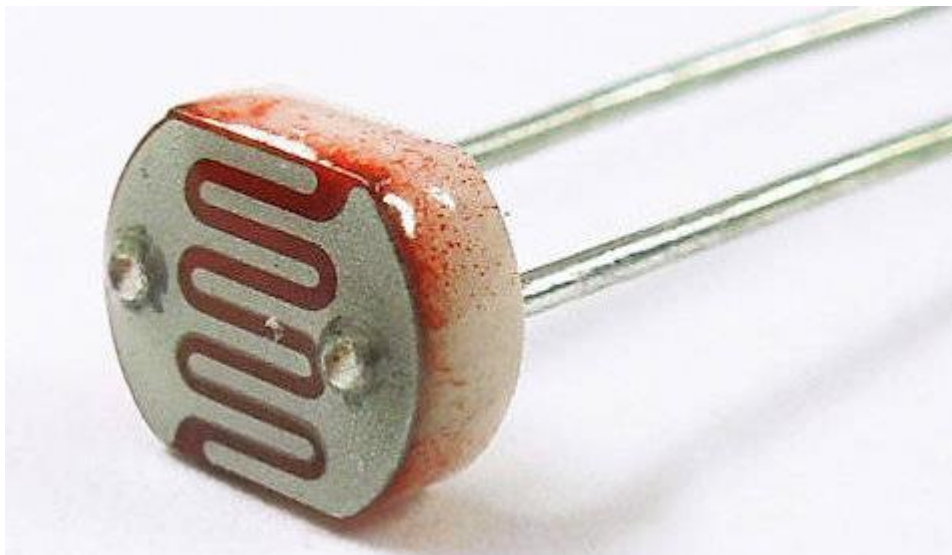


Рисунок 2.10 - Фоторезистор GL5516

Принцип його роботи заключається в тому, що він виготовлений з напівпровідникового матеріалу, а саме сульфїду кадмію, провідність якого зростає під впливом світла, що відповідно призводить до того, що його опір зменшується. Відповідно, на противагу до великої кількості освітлення, в темряві опір фоторезистору сильно зростає. Завдяки цій властивості напівпровіднику, даний датчик було використано для вимірювання кількості освітлення навколишнього середовища[13].

2.4 RGB LED

Для реалізації системи освітлення було обрано світлодіод, який підтримує в собі три окремі світлодіоди: червоний, синій та зелений, що розміщено в спільному корпусі. Таке рішення було прийнято для надання можливості регулювати колір освітлення, оскільки змінюючи яскравість кожного окремого внутрішнього світлодіода, можна ітогово отримати майже будь який відтінок. Світлодіоди мають два типи будови, з загальним катодом та загальним анодом. Під час розробки системи керування освітленням було використано світлодіод із загальним анодом, принцип роботи та особливості якого заключаються в тому, що всі аноди в ньому об'єднані, і при подачі живлення на анод, окремі катоди підключаються до землі використовуючи керування.

Яскравість сигналу керується за допомогою PWM-сигналу - способу регулювання подачі середньої напруги на діод без зміни фактичної напруги живлення. Для контролю використовується таймер мікроконтролера, який генерує сигнал із заданим коефіцієнтом заповнення.

2.5 Мова програмування C

Однією з найпотужніших мов програмування для вбудованих систем та при роботі з мікроконтролерами є мова C. Вона має низькорівневий доступ до апаратного забезпечення, що дозволяє їй виконувати керування пам'яттю, регістрами та портами вводу і виводу. Швидкість виконання коду настільки висока, що наближена до швидкості машинного коду.

Для пристроїв із обмеженою пам'яттю важливим буде те, що споживання ресурсів цією мовою програмування мінімальне і досягає всього кількох кілобайтів. А для роботи з мікроконтролерами важливим є велика кількість готових бібліотек, що дозволяє працювати з периферією[14].

Для таких платформ, як STM32, мова програмування C по суті є стандартом, оскільки не тільки має доступ до апаратних регістрів, но і має повну сумісність з операційними системами реального часу, наприклад вже описаним

вище FreeRTOS, який також майже повністю реалізований на C. Також ця мова програмування підтримується офіційними середовищами розробки для STM32, а саме STM32CubeMX та STM32CubeIDE, що дозволяє швидко приступити до розробки без налаштування додаткового доступу для мови C.

Інші альтернативні мови програмування не так добре підходять для взаємодії та розробки на мікроконтролерах, наприклад, C++ перевантажує пам'ять за рахунок своєї більшої складності, а сильно вищий рівень абстракції буває просто не потрібний в вбудованих системах. А MicroPython має доволі низьку швидкість виконання, не дає повного контролю над периферією, і вимагає великого об'єму пам'яті, що не всі вбудовані системи можуть собі дозволити. Тому можна зробити висновок, що найкращим вибором для розробки на платі родини STM32 була мова програмування C.

2.6 Програмне забезпечення для мікроконтролера

Екосистема STM32 від STMicroelectronics включає в себе три основні інструменти, які використовуються при розробці вбудованих систем, а саме STM32CubeMX, STM32CubeProgrammer та STM32CubeIDE.

2.6.1 STM32CubeMX

Це графічний конфігуратор для мікроконтролерів STM32. Завдяки ньому можливо зручно виконати налаштування пінів мікроконтролера, наприклад, GPIO, SPI, UART та іншої, необхідної для функціонування системи, периферії. Також завдяки ньому налаштовується тактова частота, периферійні модулі по типу ADC та DMA, драйвери та інтегрувати FreeRTOS.

Серед великої кількості можливостей цього програмного забезпечення, основними та найважливішими для розробки під мікроконтролери, є автоматичне створення шаблону для STM32CubeIDE та генерація коду з одночасною ініціалізацією периферії. Це дозволяє швидко і без великої кількості марних зусиль на дублювання вже існуючого для мікроконтролеру коду

6. Генерація PWM та таймерів затримок через ТІМ;
7. Зовнішні переривання EXTI;
8. Програмування пам'яті да передачі даних без участі CPU з використанням DMA;

Бібліотека HAL має велику кількість переваг, таких як простота використання на підключення, уніфікованість функцій, що дозволяє легко в них орієнтуватись, легка підтримка та оновлення, а також сумісність з FreeRTOS, що робить її кращим вибором для роботи з мікроконтролерами родини STM32 і дозволяє полегшити процес розробки програмного забезпечення[18].

2.8 Git

Git - це система контролю версій, яка дозволяє слідкувати за змінами в коді, зберігати історію розробки, організовувати командну роботу над проектом, поєднувати зміни між собою та легко перемикає між версіями програмного забезпечення і повернутись до попередніх змін у випадку помилок.

Кожна зроблена та збережена зміна в коді залишається там у вигляді коміту, який завжди можна змінити, перенести або видалити у разі необхідності. що дозволяє не тільки контролювати, але і керувати історією змін. Завдяки можливості створювати гілки - по суті локальні копії коду -можлива робота в команді, оскільки кожен розробник буде працювати незалежно від інших, не заважаючи процесу розробки решті команди. Потім усі гілки можна злити в одну і отримати готовий результат роботи, який весь цей час розроблявся паралельно, що сильно дозволяє економити час.

Для зручної роботи з кодом також присутні веб-сервіси, які дозволяють зберігати онлайн репозиторії з кодом, по суті створюючи резервну копію. Одним з таких сервісів є GitHub. Він надає можливість створювати Pull requests - патчі, які роблять запит на зміни коду. Завдяки їх використанню розробники можуть перевіряти код один одного перед поєднанням змін в єдине ціле, залишати зауваження, робити виправлення, і таким чином усі зміни в загальному коді

будуть контрольованими і з меншою ймовірністю призведуть до проблем з програмним забезпеченням.

Також GitHub містить в собі можливість відслідковування задач і багів, для того що бачити, які саме вимоги призвели до створення саме такого патчу з кодом. Також всередині нього є підтримка CI/CD, який відповідає за автоматичну збірку та тестування, що дозволяє перед злиттям коду з основним програмним забезпеченням перевірити, що проблеми не виникають на етапі компіляції.

Важливою є можливість вести локальну документацію, яка буде допомагати новим розробникам та користувачам програмного забезпечення легко розібратись в призначенні того чи іншого функціоналу та коду, і зменшить навантаження на розробників завдяки тому, що певна кількість питань буде вирішуватись через прочитання прикладеної до проекту документації.

З цього можна зробити висновки, що Git має велику низку переваг, а саме локальність, яка дозволяє зберігати зміни на комп'ютері розробника без необхідності постійно оновлювати віддалений репозиторій. Надійність, яка дозволяє повернутися до попередньої стабільної версії у випадку виникнення проблем. Масштабованість, яка дозволяє користуватись Git та GitHub на проєкті будь-якого розміру з будь-якою кількістю розробників, які працюватимуть над ним. І з цього впливає важливість можливості спільної роботи, яка дозволяє різним людям одночасно працювати над спільним проєктом без конфліктів в коді. Тому це все підтверджує, що використання Git є максимально зручним, і навіть необхідність вивчити певну кількість команд для керування ним є незначною на фоні усіх перерахованих переваг, які дозволяють комфортно організувати роботу в будь-яких умовах[19].

3. ОПИС РОЗРОБЛЕНОГО ПРИСТРОЮ АВТОМАТИЗАЦІЇ КЕРУВАННЯ ОСВІТЛЕННЯМ

3.1 Загальна архітектура та архітектура програми

3.1.1 Загальна схема апаратної системи

Апаратна система розроблена на основі мікроконтролера STM32F411, додаткового модуля ESP, а саме ESP8266 ESP-01S, фоторезистора GL5516 який використовується в якості сенсора освітленості та RGB-світлодіода для візуальної індикації. Загальна мета системи полягає у зчитуванні рівня освітленості, подальшій обробці цих даних та їхній візуалізації, а також отриманні команд та передачі через мережу Wi-Fi.

У системі мікроконтролер STM32F411 виконує роль основного обчислювального блоку, що здійснює зчитування аналогових сигналів з фоторезистора, обробляє їх та керує RGB-світлодіодом за допомогою широтно-імпульсної модуляції (PWM). Також передбачено обмін даними між STM32 та WI-FI модулем ESP8266 за допомогою інтерфейсу UART. ESP-модуль відповідає за передачу інформації у зовнішні мережі, зокрема в хмарне середовище або на віддалені сервери, використовуючи бездротове з'єднання Wi-Fi. Також завдяки ньому відбувається отримання команд для керування системою, а також інформація з зовнішніх джерел про стан користувача, якщо система працює в режимі адаптації під біоритми користувача на основі його стану.

Фоторезистор підключено до аналогового входу мікроконтролера через подільник напруги, верхнє плече подільника - це сам фоторезистор, нижнє - резистор з фіксованим номіналом, підключений до землі. Середня точка подільника, тобто аналоговий сигнал, подається на PA1 – вхід ADC1_IN1 мікроконтролера STM32. Цей вхід дозволяє зчитувати рівень освітленості за допомогою вбудованого аналого-цифрового перетворювача.

RGB-світлодіод підключається до трьох цифрових виходів STM32, що підтримують PWM, кожен з його каналів має свій колір, відповідно до назви

червоний, зелений та синій (Red, Green, Blue), і з'єднаний із відповідним виводом через обмежувальні резистори (приблизно 220 Ом), що запобігає перевищенню допустимого струму через світлодіод. Тип RGB-світлодіода (анодний) визначає логіку його підключення - відповідно або до живлення, або до землі, і відповідно нашому випадку підключення відбувається до живлення 3.3В.

RGB-світлодіод підключено до трьох виводів мікроконтролера STM32411 наступним чином:

1. PB0 – червоний канал (R);
2. PA6 – синій канал (B);
3. PA7 – зелений канал (G);

Ці пін-коди забезпечують керування яскравістю кожного кольору світлодіода шляхом зміни скважності ШІМ-сигналу.

Модуль ESP8266 ESP-01S підключається до мікроконтролера STM32F411 за допомогою UART-інтерфейсу. Для забезпечення сумісності рівнів логіки (STM32F411 може працювати як на 3.3 В, так і на 5 В, тоді як Wi-Fi модуль підтримує лише 3.3 В) було передбачено використання понижуючого перетворювача логічного рівня на лініях UART, особливо на лінії передачі (TX) зі сторони STM32F411. Передача даних між STM32 та ESP8266 здійснюється за такими лініями:

1. TX STM32 (передає до ESP) – вивід PA9;
2. RX STM32 (приймає від ESP) – вивід PA10;

Окрему увагу необхідно приділити живленню системи, тому що модуль ESP8266 ESP-01S потребує стабільного джерела живлення на 3.3 В з достатнім струмовим запасом (не менше 500 мА), оскільки під час активної передачі даних по Wi-Fi можливі імпульси споживання високого струму. У випадку нестабільного живлення можуть виникати збої в роботі модуля або його перезавантаження.

Такиим чином, можна зробити висновок, що загальна апаратна архітектура передавання отриманих даних на схемі забезпечує збір даних з фоторезистора, обробку інформації мікроконтролером STM32, виведення результатів на RGB-

світлодіод та керування через модуль ESP8266 завдяки зовнішній мережі. Завдяки такій структурі створилась можливість реалізувати систему моніторингу освітленості з функцією віддаленого доступу до керування системою.

Також необхідно звернути увагу на те, що окрім компонентів, необхідних для прямого функціонування системи, розробка містить в собі додаткові допоміжні частини. Наприклад, для прошивки мікроконтролера використовується відлагоджувальний інтерфейс SWD (Serial Wire Debug) - програматор ST-Link, що підключається до STM32 за допомогою чотирьох сигнальних ліній:

1. SWDIO (дані) – підключено до виводу SWD на STM32;
2. SWCLK (тактовий сигнал) – підключено до виводу SWC;
3. GND (земля) – загальний провід;
4. VCC (живлення) – підключено до лінії 3.3 В;

Це стандартне з'єднання для ST-Link, яке дозволяє не лише прошивати мікроконтролер, але й відлагоджувати програму у реальному часі.

Для виведення логів і налагодження роботи мікроконтролера використовується інтерфейс UART1, який підключено до зовнішнього USB-UART перетворювача. Детальний опис підключення кожного виводу:

1. TX (передача даних зі сторони STM32) – вивід PA2;
2. RX (прийом даних до STM32) – вивід PA3;
3. GND – загальний провід з USB-UART адаптером;

Цей інтерфейс дозволяє виводити діагностичну інформацію на комп'ютер під час роботи системи, що є особливо корисним при розробці і тестуванні.

Розпіновка кожного компоненту на мікроконтролері STM32F411 зображена на рисунку 3.1. Функціональна схема міститься в Додатку А, ІАЛЦ.466550.005 Е2.

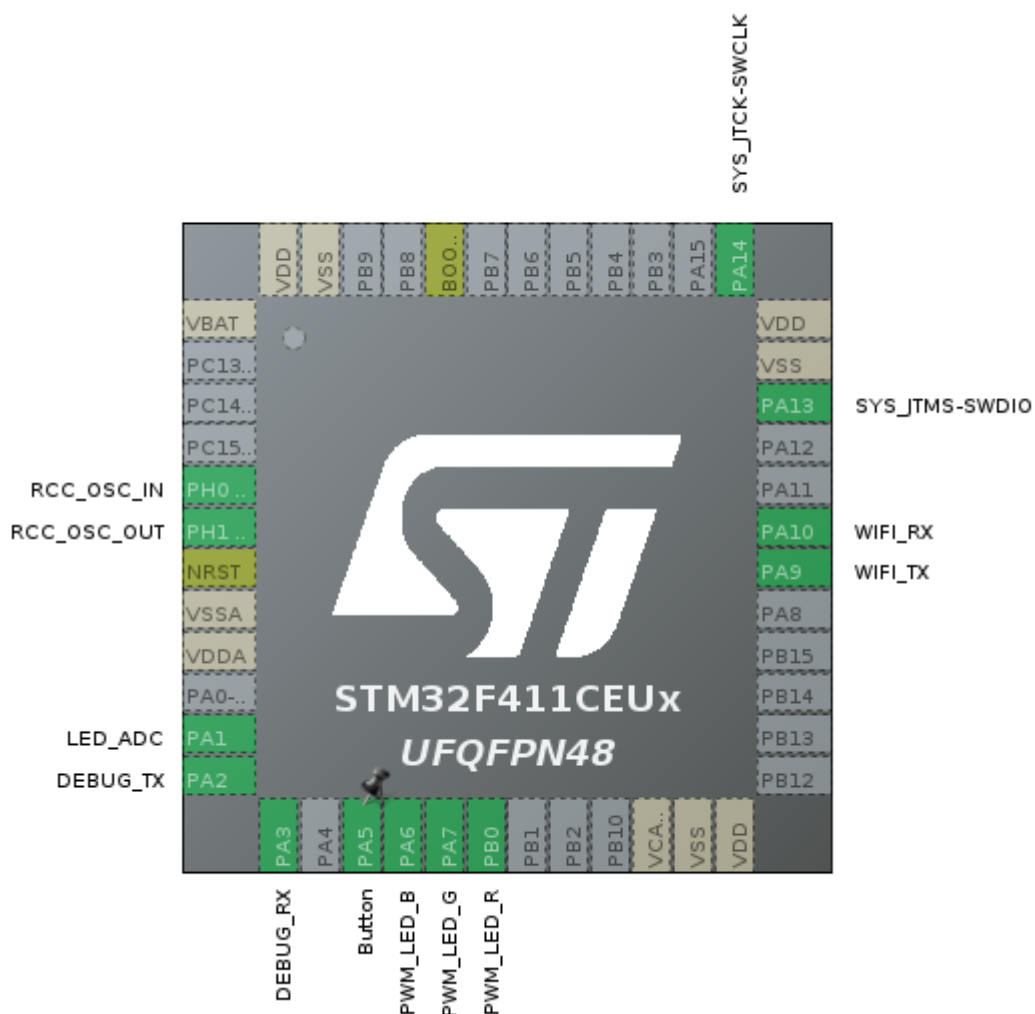


Рисунок 3.1 - Розпіновку компонентів на мікроконтролері STM32

3.1.2 Архітектура програмного забезпечення: основні блоки, логіка взаємодії

Програмне забезпечення, розроблене для функціонування апаратної системи, реалізоване з використанням мікроконтролера STM32, модуля ESP8266 та фоторезистора GL5516. Архітектура програмної частини базується на модульному принципі з використанням операційної системи реального часу FreeRTOS, а основна логіка програми полягає в циклічній перевірці стану зовнішнього середовища, реагуванні на зміну умов та взаємодії з користувачем через веб-інтерфейс. Розглянемо детальніше основні програмні блоки:

1. Блок керування RGB-світлодіодом

Реалізує керування яскравістю трьох каналів світлодіода (R, G, B) за допомогою широтно-імпульсної модуляції, а бажана яскравість світлодіода досягається завдяки налаштуванню значення ШІМ обернено-пропорційно до значення зовнішньої яскравості - чим яскравіше середовище, тим менше горить світлодіод, і навпаки (Див. Додаток А, ІАЛЦ.466550.007 Д2).

2. Блок взаємодії з ESP8266

Здійснює обмін даними між STM32 і ESP8266 через UART, а також передбачає передачу поточних значень сенсорів, прийом команд користувача, і займається ініціалізацією та обслуговуванням веб-сервера.

3. Блок обробки режимів роботи

Визначає поточний режим роботи системи, який задається користувачем через веб-інтерфейс. Наразі система підтримує такі три режими:

1. «Стабільне освітлення»
2. «Освітлення на основі рівня освітленості»
3. «Освітлення на основі стану користувача»

4. Блок зв'язку з користувачем (веб-сервер)

Побудований на базі ESP8266, який виступає у ролі Wi-Fi модуля та ініціалізується мікроконтролером STM32F411 через AT-команди, після чого ESP створює точку доступу або підключається до наявної Wi-Fi мережі та піднімає простий веб-сервер. Через цей веб-інтерфейс користувач може обрати бажаний режим роботи, передати команду або змінити параметри.

5. Блок зчитування даних з фоторезистора

Відповідає за зчитування аналогового сигналу з виводу PA1, обробку та нормалізацію значення яскравості для подальшого використання в логіці керування світлодіодом (Див. Додаток А, ІАЛЦ.466550.007 Д2).

У структурі програмного забезпечення основним є регулярне виконання головної задачі, яка з певною періодичністю, а саме кожну секунду, здійснює перевірку стану системи та виконує необхідні дії відповідно до поточного

режиму (Див. Додаток А, ІАЛЦ.466550.006 Д1). Логіку функціонування задачі можна описати наступним чином:

1. Режим «Стабільне освітлення»

У цьому режимі RGB-світлодіод залишається ввімкненим з фіксованим значенням яскравості, яке було задане користувачем раніше, і жодна дія з боку системи не буде здійснена, доки не станеться зміна режиму роботи.

2. Режим «Освітлення на основі рівня освітленості»

У даному режимі задачею щосекунди зчитується поточне значення фоторезистора з ADC, після чого це значення порівнюється з попереднім (збереженим) значенням яскравості. Якщо зафіксовано істотну зміну освітлення, система оновлює значення ШІМ на каналах RGB-світлодіода таким чином, щоб відповідати новому рівню освітлення, і завдяки цьому адаптує яскравість під поточні умови зовнішнього середовища.

3. Режим «Освітлення на основі стану користувача»

У цьому режимі система орієнтується на стан користувача, отриманий через інтерфейс з ESP8266. Якщо отримано сигнал, що користувач спить - світлодіод вимикається, і навпаки - якщо користувач прокинувся, то вмикається підсвічування на мінімальній яскравості для комфортного освітлення.

Треба приділити увагу тому, як відбувається ініціалізація та активація WI-FI модуль ESP8266, а саме які AT-команди там використовуються і в якій послідовності (Див. Додаток А, ІАЛЦ.466550.008 Д3):

1. Підключення до мережі Wi-Fi або створення власної точки доступу.
2. Запуск веб-сервера на ESP8266.
3. Реакція на запити користувача з інтерфейсу браузера.
4. Передача команд до STM32.

Команди, отримані ESP8266, пересилаються STM32 у вигляді послідовних повідомлень, а зі свого боку STM32 періодично надсилає до ESP оновлену інформацію про рівень освітленості та статус системи.

Розроблена архітектура програмного забезпечення базується на модульному підході з чітким розділенням відповідальності між окремими блоками: сенсорики, керування виконавчими пристроями, взаємодія з мережею та логіка адаптивної реакції, що дозволяє легко масштабувати систему, додавати нові режими роботи або змінювати логіку взаємодії без істотного впливу на загальну структуру.

3.1.3 Організація основної задачі ПЗ через FreeRTOS

У розробленій системі керування освітленням було використано операційну систему реального часу FreeRTOS, що дозволяє реалізовувати багатозадачність, яка забезпечує одночасну паралельну роботу декількох логічних блоків програми. FreeRTOS дає можливість ефективно керувати ресурсами мікроконтролера та робить можливою реалізацію задач, які виконуються з різною періодичністю та пріоритетністю. Основною задачею системи є `defaultTask`, конфігурація якої зображена на рисунку 3.2. Для повноцінного функціонування системи, ми зберігаємо дескриптор задачі, що має назву `defaultTaskHandle`, а також виділяємо статичну область пам'яті, яка буде використовуватись в якості стека задачі (`defaultTaskBuffer`). Детальне налаштування задачі здійснюється через ініціалізацію та заповнення структури `osThreadAttr_t`, яка зберігає в собі всю необхідну інформацію, а саме:

1. Ім'я задачі, яке може бути використане для налагодження.
2. Вказівник на масив, який ми виділили для використання його в якості стеку задачі.
3. Розмір стеку задачі, який вказано в байтах.
4. Вказівник на структуру, яка виконує керування задачею та її розмір.
5. Пріоритет цієї задачі.

3. Режим перетворення налаштований таким чином, що вимірювання відбувається тільки після явного запуску, а не безперервно, а запуск перетворення виконується програмно, без використання зовнішніх тригерів.
4. Дані зчитуються вручну, без використання DMA, а після завершення вичитки генерується подія завершення, що дає змогу вичитати результат.

```
hadcl.Instance = ADC1;  
hadcl.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV2;  
hadcl.Init.Resolution = ADC_RESOLUTION_12B;  
hadcl.Init.ScanConvMode = DISABLE;  
hadcl.Init.ContinuousConvMode = DISABLE;  
hadcl.Init.DiscontinuousConvMode = DISABLE;  
hadcl.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;  
hadcl.Init.ExternalTrigConv = ADC_SOFTWARE_START;  
hadcl.Init.DataAlign = ADC_DATAALIGN_RIGHT;  
hadcl.Init.NbrOfConversion = 1;  
hadcl.Init.DMAContinuousRequests = DISABLE;  
hadcl.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
```

Рисунок 3.3 - Детальна конфігурація АЦП з використанням HAL

3.2.2 Налаштування PWM для керування яскравістю RGB LED

У системі керування освітленням було використано RGB-світлодіод, для якого реалізоване плавне регулювання яскравості кожного з трьох кольорових каналів: червоного (Red), зеленого (Green) та синього (Blue). Це регулювання здійснюється за допомогою широтно-імпульсної модуляції (ШІМ, або PWM), що реалізується на базі таймера TIM3 мікроконтролера STM32F411. Налаштування та ініціалізація, аналогічно до АЦП, генеруються бібліотекою HAL і містять в собі такі параметри для коректної роботи:

1. Для забезпечення плавної зміни яскравості, кількість тактів на один повний цикл ШІМ було встановлено в значення 1000, таким чином отримуючи відповідно тисячу рівнів яскравості.
2. Для гарантування стабільності роботи таймер використовує внутрішнє тактування, а також має увімкнене попереднє завантаження регістру ARR (Auto Reload Register), що дозволяє уникати збоїв при зміні частоти або робочого циклу в реальному часі.

3. Для забезпечення єдиного методу керування яскравістю, було використано три канали ШІМ, кожен з яких відповідає за свій колір світлодіоду, вони конфігуровані однаково і мають назви TIM_CHANNEL_1, TIM_CHANNEL_2 та TIM_CHANNEL_3 (див. рис. 3.4).

```
if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_2) != HAL_OK)
{
    Error_Handler();
}
if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_3) != HAL_OK)
{
    Error_Handler();
}
```

Рисунок 3.4 - Конфігурація трьох каналів ШІМ

3.2.3 Налаштування та використання USART

Для реалізації обміну даними між мікроконтролером та зовнішніми пристроями, у даній системі було використано два апаратних інтерфейси для послідовної передачі даних, які мають назви USART1 та USART2. Кожен з цих інтерфейсів налаштовано на роботу у режимі прийом-передача, і вони мають однакові параметри конфігурації.

Для зв'язку з WI-FI модулем ESP8266 використовується USART1, і це дозволяє реалізувати передачу та прийом команд, його налаштування відбувається завдяки згенерованій функції MX_USART1_UART_Init.. Розглянемо основні параметри конфігурації (див. рис. 3.5):

1. Швидкість передачі є стандартною і дорівнює 115200 біт на секунду.
2. Довжина слова(патчу передаваних даних) дорівнює 8 біт.
3. Наявний один стоп-біт.
4. Перевірка парності вимкнена.
5. Для роботи з перериваннями також присутнє додаткове налаштування пріоритету, а також включення обробки відповідного IRQ.

```
huart1.Instance = USART1;
huart1.Init.BaudRate = 115200;
huart1.Init.WordLength = UART_WORDLENGTH_8B;
huart1.Init.StopBits = UART_STOPBITS_1;
huart1.Init.Parity = UART_PARITY_NONE;
huart1.Init.Mode = UART_MODE_TX_RX;
huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart1.Init.OverSampling = UART_OVERSAMPLING_16;
```

Рисунок 3.5 - Конфігурація USART1

Для логування системи використовується другий інтерфейс із назвою USART2, який має абсолютно аналогічні до попередньо описаного інтерфейсу налаштування і ініціалізується згенеровано функцією MX_USART2_UART_Init. Логування можна здійснювати як використовуючи HAL_UART_Transmit, так і через стандартне printf, якщо потік stdout перенаправлено коректно. Для забезпечення повноцінної роботи підсистем інтерфейсу UART без додаткового втручання розробника, бібліотека HAL використовує функцію HAL_UART_MspInit, яка виконує налаштування низькорівневих ресурсів, а саме:

1. Вмикання тактовання для USART1 та USART2.
2. Ініціалізація відповідних ліній GPIO.
3. Увімкнення відповідних переривань.

Завдяки використанню двох апаратних UART-інтерфейсів, система здатна паралельно обробляти як бездротову взаємодію з ESP8266, так і вести журналювання або налагодження використовуючи UART. Завдяки такому підходу досягається висока надійність та гнучкість у передачі даних в реальному часі. Налаштування інтерфейсів для взаємодії з конкретними виводами мікроконтролера зображено на рисунку 3.6.

```

    /**USART1 GPIO Configuration
    PA9-----> USART1_TX
    PA10-----> USART1_RX
    */
    GPIO_InitStruct.Pin = WIFI_TX_Pin|WIFI_RX_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF7_USART1;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
    /**USART2 GPIO Configuration
    PA2-----> USART2_TX
    PA3-----> USART2_RX
    */
    GPIO_InitStruct.Pin = DEBUG_TX_Pin|DEBUG_RX_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF7_USART2;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

```

Рисунок 3.6 - Конфігурація виводів мікроконтролера для взаємодії з USART

3.3 Розроблена система та її функціональність

Розроблена система є інтелектуальним контролером освітлення з підтримкою кількох режимів роботи, адаптацією до умов навколишнього середовища та можливістю дистанційного керування через бездротові інтерфейси. Розглянемо детальніше реалізований функціонал та його особливості:

3.3.1 Огляд режимів роботи

Система підтримує три основні режими функціонування:

- ручний режим, в якому освітлення регулюється безпосередньо користувачем за допомогою веб-інтерфейсу, який передає команди через модуль ESP8266. У цьому режимі не відбувається зміни освітлення ніяким чином, крім прямого увімкнення чи вимкнення системи.
- автоматичний режим, в якому система самостійно регулює яскравість RGB-світлодіоду відповідно до показників фоторезистора, а саме

адаптує яскравість освітлення у разі зміни рівня зовнішньої освітленості для підтримання комфортного рівня світла в приміщенні. Цей режим дозволяє автоматизувати переключення освітлення відповідно до зовнішньої обстановки і надає можливість більш економно розходити електроенергію.

- режим взаємодії з фітнес-браслетом, в якому пристрій отримує інформацію про стан користувача (сон або активність) з фітнес-браслета за допомогою Wi-Fi-з'єднання. Якщо користувач спить, освітлення переводиться у нічний режим, а саме вимикається заради економії електроенергії, а при пробудженні - відновлюється м'яке світло.

3.3.2 Логіка перемикавання між режимами

Перемикавання між режимами реалізується через веб-інтерфейс, який надає користувачеві змогу встановити поточний режим у реальному часі. Зміна режиму фіксується у змінній стану системи, яку щосекунди опитує основна задача у RTOS-середовищі. Якщо детектовано зміну режиму, то система негайно адаптує алгоритм роботи відповідно до нового стану.

3.3.3 Обробка команд від користувача

Користувацькі команди приймаються через веб-інтерфейс ESP8266, який працює як локальний веб-сервер. Кожна команда передається до мікроконтролера STM32 по UART, розбір команд реалізований у вигляді окремої задачі, яка обробляє вхідні повідомлення з буфера UART..

Підтримуються два типи команд:

1. Увімкнути/вимкнути світло
2. Змінити режим роботи

Також веб-інтерфейс містить в собі функціонал відображення поточного стану системи освітлення - включена вона чи вимкнена.

3.4 Огляд та аналіз отриманих під час функціонування системи результатів

3.4.1 Вимірювання реакції системи на зміну освітленості

В розробленій системі керування освітленням реалізовано механізм автоматичного регулювання яскравості RGB-світлодіода залежно від зовнішньої освітленості. Роль основного сенсорного елементу виконує фоторезистор, який підключено до аналогового входу мікроконтролера STM32F411. Обробка значень з фоторезистора відбувається за допомогою вбудованого аналогово-цифрового перетворювача (АЦП), після чого отриманий сигнал використовується для динамічного керування яскравістю світлодіода через широтно-імпульсну модуляцію (ШІМ). Значення напруги на фоторезисторі вимірюється 12-бітним АЦП, який генерує цифрове значення в діапазоні 0...4095, де 0 відповідає повній темряві (мінімальна напруга на вході), а 4095 показує яскраве освітлення (максимальна напруга).

Оскільки логіка роботи системи передбачає зменшення яскравості світлодіода при збільшенні зовнішнього освітлення, тобто обернену залежність, було проведено ряд експериментів для визначення, як фоторезистор реагує на різні рівні освітлення. Розглянемо таблицю 3.1 усереднених результатів реакції системи на різні рівні освітлення

Таблиця 3.1 - Відповідність яскравості світлодіоду до умов освітлення

Умови освітлення	Значення АЦП	Значення ШІМ	Поведінка світлодіода
Повна темрява	~100	~975	Максимальна яскравість
Напівтемрява (сутінки)	~800	~804	Яскраве світіння
Слабке денне	~1600	~607	Середня яскравість

Розрахуємо час одного такту, використовуючи формулу $1/\text{частота АЦП}$, що в нашому випадку після підстановки перетворюється на $1/36 \text{ МГц}$ і в результаті дає значення 27.78 нс . Для отримання повного часу, помножимо кількість циклів на отримане значення часу одного такту, і можемо зробити висновок що час конверсії $\sim 0.4 \text{ мкс}$ (400 наносекунд). В підсумку, якщо враховувати при розрахунках запуск через `HAL_ADC_Start()`, очікування завершення `HAL_ADC_PollForConversion()` та зчитування значення `HAL_ADC_GetValue()`, повний цикл вимірювання у типовому випадку займає приблизно $5\text{--}10 \text{ мкс}$.

3.4.2.2. Затримка при передачі даних через UART

Передача даних до модуля ESP8266 відбувається через використання інтерфейсу USART1, який налаштовано зі швидкістю $115200 \text{ біт на секунду}$. На основі цього можемо провести розрахунок часу передачі повідомлення:

1. Сумарна кількість інформації = $1 \text{ старт-біт} + 8 \text{ біт даних} + 1 \text{ стоп-біт} = 10 \text{ біт}$
2. $115200 \text{ біт в секунду} = 11.52 \text{ тис. байт/сек}$, відповідно на один байт витрачаємо приблизно 86.8 мкс

Розглянемо приклад передачі команд: якщо надсилається рядок `AT+RST\r\n` (розмір - 8 байт), його передача займе приблизно 700 мкс . Якщо приймається відповідь з модуля у вигляді `OK\r\n` (розмір - 4 байти), це відповідно займе вдвічі менше часу, а саме 350 мкс . У разі передавання/отримання типового HTTP-запиту або JSON-повідомлення, довжина якого приблизно рівна 100 байтам , можемо розрахувати що повна затримка UART TX+RX буде становити близько $10\text{--}20 \text{ мс}$.

3.4.2.3. Затримка ініціалізації та обробки модуля ESP8266

При старті або перезапуску модуля ESP8266 надсилається команда `AT+RST`, яка запускає перезавантаження модулю. Приблизний час очікування відповіді на ініціалізацію займає приблизно $1\text{--}2 \text{ секунди}$, а підключення модулю до мережі WI-FI з використанням команди `AT+CWJAP=` займає від однієї до

трьох секунд, час операції залежить від якості сигналу, а також він проходження автентифікації. Після встановлення з'єднання по Wi-Fi, ESP8266 починає працювати як TCP-сервер. За наявності запиту з клієнта, який в системі відбувається через веб-інтерфейс, час відповіді залежить від обробки запиту і передачі відповіді і його значення залежить від типу запиту та його розміру.

Час обробки HTTP GET-запиту (веб-сервер на AT-командах або AT+CIPSEND) дорівнює приблизно 20–100 мс для простих відповідей і 150–300 мс для великих відповідей (розміром приблизно 500 байт). В той час ping (перевірка наявності з'єднання) до ESP8266 у локальній мережі займає всього 1–3 мс.

На основі всього дослідженого вище, можна помістити результати у таблицю 3.2 та зробити висновок, що система демонструє задовільні показники часу відповіді для задач реального часу середньої точності, характерних для побутових автоматизованих систем освітлення.

Таблиця 3.2 - Час реагування компонентів та процесів системи

Компонент або процес	Орієнтовний час роботи
АЦП - повний цикл вимірювання	~5–10 мкс
UART - передача 1 байта	~87 мкс
UART повідомлення розміром 100 байт	~10–20 мс
Ініціалізація ESP8266 (AT+RST)	~1.5–2.0 с
Підключення до Wi-Fi (CWJAP)	~1–3 с
Відповідь ESP8266 на HTTP запит	~20–300 мс
Ping до ESP8266 у локальній мережі	~1–3 мс
Час реакції на освітлення (по тасці)	~1000 мс (1 секунда)

3.4.3 Аналіз стабільності з'єднання через ESP8266

Для безперебійної роботи системи та керування нею важливо мати стабільне з'єднання з мережею WI-FI. Для цього необхідно розглянути фактори, які впливають на стабільність зв'язку та проаналізувати методи усунення можливих проблем.

3.4.3.1 Фізичні умови стабільності

Живлення є необхідною умовою для функціонування модулю. ESP8266 споживає до 250 мА під час передачі, проте кількість необхідного струму може зростати під час передачі великої кількості даних. Нестабільне або недостатнє живлення, наприклад, при використанні стабілізатора без запасу по струму, може призводити до самовільного перезавантаження модуля. Окрім цього, якість сигналу може падати через екранування сигналу, наприклад, металевими корпусами.

3.4.3.2 Програмні умови стабільності

Передача AT-команд через UART має бути ретельно синхронізована, оскільки для неї типовими причинами нестабільності є відсутність затримок після команд AT та обрив з'єднання при перевантаженні UART-буфера. Для уникнення виникнення таких помилок, було додано затримки після кожної AT-команди довжиною в 300 мілісекунд - середнє рекомендоване значення, якого достатньо для виконання обробки запиту.

Також для того, щоб не продовжувати роботу модуля при виникненні помилок, виконується обробка відповідей модуля, і у випадку якщо модуль повертає помилку, виконується повторна відправка команд для гарантії виконання команди.

3.5 Аналіз ефективності розробленої системи

Важливим показником для системи керування освітленням є ефективність її роботи, продуктивність та використання ресурсів мікроконтролера та користувача. Дана система була створена для полегшення та автоматизації

життя, а також економії електроенергії, тому в цьому пункті розглянемо та проаналізуємо використання ресурсів системою.

Енергоефективність системи забезпечується завдяки використанню апаратних і програмних механізмів, які автоматично зменшують використання енергії за рахунок режиму роботи, а саме:

1. Керування яскравістю світлодіодів через PWM

Для керування яскравістю RGB-світлодіодів використовується широтно-імпульсна модуляція (PWM), що дозволяє ефективно зменшити енергоспоживання без втрати візуального сприйняття, оскільки яскравість світлодіодів контролюється коефіцієнтом заповнення PWM. При низькому рівні освітлення у приміщенні ШІМ підвищується, а при яскравому – знижується до майже нульових значень, що знижує середній струм споживання світлодіодами, а також під час роботи системи у режимі «режим сну користувача» яскравість знижується до мінімуму, що забезпечує мінімальне споживання енергії без пустого і зайвого горіння світлодіоду.

2. Обмежене використання активних компонентів

Фоторезистор підключений до АЦП, який активується лише для періодичних вимірювань в режимі «освітлення на основі рівня освітленості», а не працює постійно в режимі безперервної конверсії, що сильно зменшує споживання енергії та ресурсів мікроконтролеру. Модуль ESP8266 активується лише в певних режимах або на запит, уникаючи постійної передачі даних, що так само призводить до того, що ресурси мікроконтролеру STM32 споживаються менше, так само як і енергія, необхідна для функціонування WI-FI модулю.

Для нормального функціонування системи протягом тривалого періоду, необхідно було також провести аналіз навантаження мікроконтролера для визначення слабких місць системи та пошуку способів покращення та прискорення роботи функціоналу, а також визначення можливості майбутнього розвитку системи.

Типовими джерелами навантаження на центральний процесор мікроконтролера є виконання періодичної таски, вимірювання рівня освітленості, обчислення нового рівня ШІМ та оновлення каналів освітлення, обробка комунікації з модулем ESP8266 через UART та обробка користувачьких команд, що надійшли через веб-інтерфейс. Усі ці задачі, виконувані процесором, наразі займають доволі мало часу і пам'яті, що по ітогу підрахунків дає можливість визначити, що середнє навантаження знаходиться десь в межах 5-10 відсотків, що залишає великий простір для масштабування системи в майбутньому.

Не менш важливим чинником є час реагування системи на події, які відбуваються. Цей показник визначається через затримки в обробці кожного типу подій, і для розробленої системи сумарний час реакції на зміну освітлення, який включає в себе зчитування нового значення з АЦП, його обробку та оновлення яскравості світлодіода, має значення приблизно в одну мілісекунду. Ця швидкість досягається завдяки прямій роботі з регістрами, які використовуються для зчитування даних з фоторезистору та запису значень для оновлення яскравості світлодіоду.

Наразі розроблена система керування освітленням має не таку велику кількість функціоналу та шляхів підключення, які призвели б до значного завантаження центрального процесору мікроконтролеру. З цього можна зробити висновок, що наразі система є стабільною і не приймає на себе великі навантаження, а також має потенціал для розвитку в майбутньому. Оскільки із можливим додаванням датчиків, прийомо-передатчиків та виводів збільшиться навантаження на периферію, відповідно потрібно буде провести нові дослідження та обрахунки для визначення слабких місць системи. Крім цього, зайвим не буде стрес-тест, який дозволить виявити недоліки роботи під час великого навантаження на систему.

3.6 Аналіз можливостей застосування освітлювальної системи

3.6.1 Застосування у побуті (система розумний дім)

Якщо роздивлятися вже існуючі інтелектуальні освітлювальні системи, то можна зробити висновок, що вони часто є частиною концепції “розумний дім”, яка дозволяє автоматизувати середовище користувача, і робити його більш зручним, адаптивним та енергоефективним. На основі цього розроблена система керування освітленням може мати такі сценарії використання:

1. Автоматичне регулювання яскравості освітлення залежно від природного освітлення приміщення. Вдень, коли в кімнаті достатньо світла - система автоматично зменшує яскравість світлодіодів або повністю вимикає їх, що зменшує енергоспоживання, і навпаки, у сутінках чи вночі плавно підвищує яскравість.
2. Нічний режим освітлення. Система може автоматично вимикати світло, коли користувач засинає, і навпаки, вмикати тьмяне світло коли користувач прокидається, що робить навігацію вночі безпечною та комфортною.
3. Сценарії автоматизації, завдяки яким можна налаштувати різноманітні автоматичні «сцени», які мають різний рівень підсвічування, що вмикаються за розкладом.
4. Контроль через веб-інтерфейс, організація якого відбувається через модуль ESP8266, що забезпечує доступ до системи з будь-якого пристрою у домашній мережі без потреби у хмарних сервісах. Це зручно, безпечно та не залежить від сторонніх сервісів.

На основі цього можна виділити низку переваг розробленої системи керування освітленням, серед яких є зниження споживання електроенергії, комфорт та автоматизація повсякденних дій, а також можливість адаптації відносно особистих уподобань користувача.

3.6.2 Використання для людей з особливими потребами

Особливу важливість подібні освітлювальні системи мають для людей з обмеженими можливостями, таких як порушення зору, опорно-рухового апарату чи когнітивні порушення. Завдяки подібним системам їх життям можна суттєво полегшити та покращити, і ось які можливі сценарії допомоги можуть бути:

1. Автоматичне освітлення без потреби вмикати вручну: людина з обмеженою рухливістю не змушена шукати вимикач - освітлення вмикається на основі умов навколишнього середовища або її фізіологічного стану, наприклад, браслет сигналізує про прокидання.
2. Плавна зміна яскравості для людей з підвищеною чутливістю до світла, наприклад, після травм або при епілепсії, яким важливо уникати різких спалахів. Управління з використанням широтно-імпульсної модуляції забезпечує м'яку зміну яскравості, що знижує стрес та навантаження на очі.
3. У темному приміщенні система сама вмикає світло мінімальної яскравості, реагуючи на зовнішнє освітлення. Це зручно для людей, які мають порушення зорового апарату або для літніх людей, оскільки освітлення адаптується без необхідності вручну змінювати налаштування і дає можливість спокійно та комфортно орієнтуватись в темному приміщенні.
4. Можливість віддаленого керування, яке надає можливість людям, які не можуть дотягнутися до звичних перемикачів, використовувати веб-інтерфейс для управління та налаштування системи.

Виходячи з перерахованого вище, можна підбити підсумок про те, що для людей з обмеженими можливостями та людей похилого віку розроблена інтелектуальна система освітлення має переваги у вигляді зменшення залежності освітлення від фізичних дій, можливість безпечно пересуватися в темний період дома без ризику травмування, адаптації освітлення під індивідуальні потреби на основі того, які саме особливі потреби має людина, а також простота у використанні навіть без спеціальних навичок.

3.6.3 Використання в офісах, готелях, лікарнях та іншій інфраструктурі

Окрім вже описаного вище, що відноситься до персонального використання, розроблена освітлювальна система має потенціал для використання у комерційних та громадських установах, зокрема в офісах, готелях та медичних закладах, які статистично є найбільшими споживачами електроенергії. Використання автоматизації дозволить зменшити навантаження на персонал і підвищити комфорт клієнтів.

3.6.3.1 Використання в офісах

У сучасних офісах важливо забезпечити ергономічне освітлення, яке сприяє продуктивній роботі, не перевантажує зір і динамічно адаптується до умов навколишнього середовища, тому встановлення системи автоматизації освітлення дозволить користуватись такою низкою функціоналу:

1. Автоматичне регулювання яскравості протягом дня. Зранку і ввечері – інтенсивніше освітлення, вдень – зменшене за рахунок природного світла, що знижує енергоспоживання та створює приємні умови для працівників.
2. Налаштування освітлення за зонами, наприклад, робочі місця освітлюються яскравіше, ніж зони відпочинку або коридори. Кожна зона потребує власної системи, проте кожна з них може бути індивідуально налаштована під певні потреби.
3. Синхронізація з календарем завдяки використанню автоматичних сценаріїв - світло автоматично змінюється відповідно до розкладу зустрічей або подій.

Таке налаштування призведе до зменшення витрат на електроенергію, покращить самопочуття працівників, зменшить кількість ручних дій, відповідно полегшивши роботу офіс-менеджерів, які більше не будуть зобов'язані перемикати системи освітлення власноруч.

3.6.3.2 Застосування в готелях

У готельному секторі освітлення відіграє ключову роль у формуванні атмосфери комфорту, безпеки та естетики, а розумне освітлення дозволить

зробити перебування гостей приємнішим, і водночас з цим знизити експлуатаційні витрати. Роздивимось список функціоналу, який дозволить гостям готелю почуватись комфортніше:

1. Плавне нічне підсвічування з мінімальною яскравістю для безпечного пересування гостей вночі, що дозволить зменшити травматизм і позбавить керівництво готелю переживань за здоров'я та безпеку клієнтів, а також зменшить кількість конфліктних ситуацій через травми в темряві.
2. Користувач матиме можливість персоналізовано налаштувати освітлення в номері під власні потреби, що покращить його досвід відпочинку і дозволить менше думати про необхідність керування освітленням. Окрім цього, адаптація під зовнішнє освітлення, яке дозволить у номерах із великою кількістю природного світла освітлення автоматично приглушати освітлення.
3. Централізоване керування, яке дозволить адміністрації готелю дистанційно перевіряти, чи вимкнено світло в незаселених номерах..

Це призведе не тільки до економії електроенергії, як в усіх попередньо розглянутих випадках використання розробленої системи керування освітленням, но і підвищить рівень задоволеності гостей, які відповідно можуть збільшити потік нових клієнтів в готель за рахунок гарних рекомендацій. Також використання розумної системи освітлення покращить імідж закладу як технологічного та сучасного, таким чином підвищуючи свій рейтинг серед альтернативних готелів.

3.6.3.3 Використання в медичних закладах

В медичних закладах освітлення має функціональне, безпечне та адаптивне значення, а правильно організоване освітлення полегшує роботу персоналу, покращує стан пацієнтів та зменшує ризики. Через це використання автоматичної системи керування освітленням в лікарнях матиме такий список переваг:

1. Автоматичне зменшення яскравості вночі у палатах для збереження спокою пацієнтів, яке не виключає можливості втручання персоналу або пацієнтів.
2. У процедурних та операційних медичні заклади мають необхідність у яскравому білому освітленні, що важливо для точності маніпуляцій, і оскільки система підтримує високий рівень освітленості, завдяки ній буде закрыта потреба у освітленні маніпуляційних приміщень.
3. Інтеграція з браслетами пацієнтів, світло може автоматично включатися при пробудженні пацієнта, що дозволить медичному персоналу зекономити сили та час, який пішов би на ввімкнення світла по запиті пацієнта вручну.
4. Безконтактне керування через веб-інтерфейс призведе до зниження ризику інфікування.

Таким образом, система допоможе створити комфортні умови для пацієнтів, підвищить безпеку персоналу під час нічних чергувань, зменшить ручне втручання персоналу, який зазвичай сильно загрузений, і звісно оптимізує витрату ресурсів.

3.6.4 Переваги автоматичного освітлення над звичайним

Автоматичне освітлення, побудоване на основі мікроконтролерних систем, датчиків та інтелектуальних алгоритмів, суттєво відрізняється від традиційних систем керування освітленням, і переваги від його використання охоплюють енергоефективність, зручність, безпеку, довговічність обладнання та адаптивність до умов середовища.

Розроблена система здатна динамічно регулювати яскравість джерел світла відповідно до рівня освітлення навколишнього середовища на основі показників фоторезистору, а також на основі заданого сценарію автоматизації. А у звичайному освітленні, навпаки, світло часто залишається ввімкненим без потреби або має занадто сильну яскравість, яка може бути не потрібна в

конкретний час доби, і це призводить до значного перевитрачання ерегії. Окрім цього, автоматичне регулювання яскравості сильно підвищує рівень комфорту для зору, підлаштовується під біоритми людини на основі освітленості в певний час доби, а також зменшує стомлення за рахунок більш м'якого регулювання освітлення в приміщенні.

Автоматична система керування освітленням не потребує постійного втручання користувача, оскільки вмикання та вимикання, а також регуляція яскравості відбуваються автоматично, без прямого втручання користувача. На відміну від цього, стандартні системи освітлення потребують постійного контролю та уваги до себе, що може бути доволі незручно, особливо в нічний час або коли активність користувача потребує зайнятості рук. Також автоматичне вмикання світла вночі зменшує ризик травмування, оскільки доволі частою ситуацією є падіння та удари в темряві через відсутність хоча б мінімального освітлення в приміщенні.

Також не можна не відмітити довговічність системи керування освітленням, оскільки вона використовує периферію лише за необхідністю, а не постійно, і таким чином продовжує час служіння, на відміну від звичайних статичних освітлювальних приладів, які часто, незалежно від потреби, використовуються на повну потужність і це пришвидшує зношення обладнання.

3.7 Опис можливостей подальшого розвитку та вдосконалення системи

3.7.1 Додати Bluetooth

Для прямого зв'язку з фітнес-браслетом користувача в майбутньому є можливість додати з'єднання використовуючи протокол Bluetooth. На даний момент в системі отримання команд про стан користувача відбувається за допомогою WI-FI, який потребує постійно підключення до мережі, в той час як протокол Bluetooth потребує тільки прямого зв'язку між системою та браслетом користувача. Це дозволить прямо зчитувати дані з браслета і за рахунок цього

зменшити затримки у взаємодії системи та користувача і швидше оновлювати освітлення приміщення на основі індивідуальних біоритмів.

Серед переваг такого підходу можна виділити низьке енергоспоживання, оскільки BLE потребує меншої кількості потужності для функціонування, швидке з'єднання між пристроями в межах приміщення, незалежність від доступу до мережі інтернет, а також зменшення навантаження на модуль ESP8266 завдяки тому, що Bluetooth буде працювати окремо.

Проте такий підхід має і певну низку недоліків, а саме обмежений радіус дії, необхідність додаткового апаратного забезпечення та місця для розміщення периферії на платі, а також можливі перешкоди та конфлікти між WI-FI та Bluetooth якщо буде використовуватись один спільний UART порт.

Також важливим є те, що браслети часто закриті для прямої роботи із стороннім програмним забезпеченням, і для них потрібна підтримка окремих персональних протоколів. Така проблема виникла і під час розробки даної системи керування освітленням - всі популярні фітнес-браслети мають закриті протоколи передачі даних на хмарні сервіси і для їх отримання потрібна дороговартісна ліцензія. Тому на даний момент переключення стану користувачу відбувається завдяки симуляції через веб-інтерфейс, оскільки можливості отримати доступ до хмарних сервісів не було.

Тому для реалізації комунікування системи з фітнес-браслетом через Bluetooth потрібно не тільки додати модуль BLE до мікроконтролера та написати програмне забезпечення, но і знайти можливість зробити інтеграцію з API фітнес-браслету, що підтримує відкриту комунікацію.

3.7.2 Підключення до Smart Home-платформ

Наразі існує багато систем автоматизації будинку, які полегшують життя користувачів, проте не завжди лінійка пристроїв має усе необхідне для задоволення потреб конкретних користувачів, тому одним з етапів розвитку розробленої освітлювальної системи може бути додавання протоколів для її

інтеграції із сторонніми платформами, наприклад, такими як Google Home або Home Assistant, які є одними із найпопулярніших на даний момент.

Завдяки підтримці інтеграції в сторонні системи, розробка отримує можливість бути видимою у загальному інтерфейсі розумного дому, отримувати команди з голосових асистентів, а також синхронізуватись з іншими пристроями для повноцінної організації простору.

Така взаємодія призведе до можливості централізовано керувати усіма пристроями системи розумного будинку, дозволить інтегруватись в популярні екосистеми, які часто бувають вже встановлені у користувачів які знаходяться в пошуках системи керування освітленням, а також завдяки взаємодії з іншими пристроями надасть можливість розширити сценарії використання, організувавши, наприклад, автоматизацію в поєднанні з датчиком руху “увімкнути світло при наявності людини в приміщенні”.

Окрім датчиків руху, можуть бути використані такі пристрої як: датчики відкриття дверей, камери, сигналізації, інші освітлювальні пристрої, телевізори, автоматичні штори тощо.

Окрім вже розглянутого сценарію із датчиком руху, можуть бути ще такі варіанти інтеграції:

1. Вмикання світла при відкритті дверей.
2. При зниженні температури в вечірній час переходити у нічний режим пониженої яскравості освітлення.
3. За відсутності будь-якої активності гасити світло.

Це дозволить реалізувати комплексну логіку взаємодії різних пристроїв, і покращить рівень комфорту, безпеки та енергоефективності.

Проте реалізація цього, звісно, має певні недоліки, а саме потенційні ризики безпеки і необхідність сплачувати ліцензію для розробки і взаємодії з певними лінійками продуктів, а також необхідність синхронізації подій та логіки кількох пристроїв для створення кросплатформених сценаріїв.

3.7.3 Вдосконалення програмної частини

На основі даних, які будуть отримані від користувача, наприклад, рівень освітлення в певний час, активність фітнес-браслета, перемикання режимів та реакція на зовнішні події, є можливість застосувати елементи машинного навчання, і це дозволить отримати такий новий функціонал:

1. Автоматична адаптація поведінки системи до звичок користувача,
2. Передбачення бажаних сценаріїв освітлення в залежності від часу доби, днів тижня, рівня активності тощо,
3. Розпізнавання шаблонів, наприклад, коли користувач найчастіше вмикає нічне освітлення, приглушене світло тощо.

Потенційна технічна реалізація може містити в собі такі етапи:

1. Збір даних, які були описані вище, для формування індивідуальної картини персональних звичок конкретного користувача.
2. Обробка на зовнішньому сервісі або в хмарному середовищі, оскільки реалізація моделей машинного навчання на самому чіпі STM32 обмежене через брак ресурсів, а залучення віддаленого сервісу дозволить зробити організацію та аналіз даних набагато швидше та простіше.
3. Адаптація поведінки системи керування освітленням у вигляді автоматичного підлаштування під поведінку користувача на основі зібраних та оброблених системою даних, прогнозування необхідності нічного освітлення на основі середньостатистичного часу сну, а також перемикання режимів роботи на основі активності без прямого втручання користувача.

Інтеграція машинного навчання дозволить ще більше зменшити навантаження на користувача і відповідно сильніше автоматизувати процес керування освітленням, додатково підвищить енергоефективність та індивідуалізує освітлення під персональні особливості користувача.

Проте реалізація цього підходу має низку складностей та недоліків, а саме необхідність довготривалого зберігання та аналізу даних, що вимагає підтримки віддаленого хмарного середовища або сплати за існуюче, розробку протоколу

безпеки для захисту персональних даних користувача та великий об'єм часу для ефективної обробки інформації для персоналізації, оскільки моделі машинного навчання потребують доволі великий об'єм історичних даних.

3.7.4 Створення мобільного застосунку

Мобільний застосунок дозволить користувачеві керувати освітлювальною системою дистанційно, контролювати її статус у реальному часі, перемикаючи режими, налаштовувати яскравість, кольори, а також переглядати історію використання, що значно розширює зручність і функціональні можливості системи. Окрім цього, поточний веб-інтерфейс потребує підключення системи керування освітленням та пристрою для керування через єдину WI-FI мережу для можливості передачі команд, а мобільний застосунок прибере цю незручність.

Також це дозволить збільшити безпеку користування завдяки захисту з'єднання через пароль або токен, шифруванню повідомлень при мережевій передачі та перевірці автентичності пристрою.

Тому розробка мобільного застосунку суттєво підвищить ефективність та зручність користування, додасть гнучкість, адаптивність та інформативність в досвід користування системою освітлення, і дозволить перетворити існуючу систему на повноцінний елемент Smart Home екосистеми.

3.7.5 Розширення кількості сенсорів

Додавання додаткових сенсорів, наприклад, температури, вологості, якості повітря або руху, дозволить значно розширити функціональні можливості освітлювальної системи, зробити її більш адаптивною до умов навколишнього середовища та інтелектуальною у прийнятті рішень щодо регулювання освітлення. У таблиці 3.3 наведено приклади сенсорів, які можуть бути інтегровані в систему керування освітленням, існуючі найпопулярніші моделі сенсорів та приклади використання.

Таблиця 3.3 - Типи сенсорів для розширення функціональності системи

Тип сенсора	Популярні моделі	Призначення та вплив на освітлення
Температури	DS18B20, DHT11, DHT22	Зниження яскравості при підвищеній температурі для уникнення перегріву приміщення.
Вологості	DHT11, DHT22	У вологих умовах, наприклад, у ванній - автоматичне вмикання м'якого освітлення.
Руху	PIR HC-SR501, RCWL-0516	Автоматичне вмикання світла при виявленні руху.
Звуку	KY-038, MAX9814	Реакція на гучні звуки, наприклад, ввімкнення світла у разі клацання.

Програмно датчики можуть бути інтегровані через окремі таски, створені в межах FreeRTOS, які будуть відповідати за періодичне опитування сенсорів, та передавати дані з них у центральну логіку контролю освітлення, яка буде приймати рішення про керування освітленням на основі отриманої інформації. Також ці дані можуть передаватись до веб-інтерфейсу з використанням модуля ESP8266 для моніторингу та аналізу історії змін параметрів навколишнього середовища, щоб користувач, навіть не приймаючи пряму участь у керуванні освітленням, мав можливість контролювати все що відбувається у приміщенні та був в курсі змін освітлення на основі показників датчиків.

Інтеграція нових типів датчиків, крім вже підключеного фоторезистору, покращить адаптивність системи завдяки реакції не тільки на рівень навколишньої освітленості, але й на інші умови, додатково підвищить енергоефективність системи за рахунок додаткових сценаріїв регулювання яскравості, і значно збільшить комфорт користувача.

ВИСНОВКИ

У рамках даної дипломної роботи розроблено та реалізовано систему автоматичного керування освітленням на основі мікроконтролера STM32F411, модуля бездротового зв'язку ESP8266 та фоторезистора GL5516. Основною метою проєкту було створення енергоефективної, адаптивної та інтелектуальної системи, здатної реагувати на зміну освітленості в навколишньому середовищі, а також забезпечити зручне керування режимами роботи через бездротовий інтерфейс.

У процесі реалізації виконано створення апаратної частини системи з налаштуванням ШІМ для керування яскравістю RGB-світлодіодів, впровадженню три режими роботи: автоматичний, ручного та інтеграція з фітнес-браслетом для реагування на біоритми користувача, та організовано зв'язок через ESP8266 за допомогою інтерфейсу USART з підтримкою обробки користувацьких команд.

Провівши аналіз ефективності, продуктивності та стабільності роботи в умовах реального навантаження, було зроблено висновки що система може бути використана у побуті (розумний дім), в офісах, готелях, навчальних та медичних установах, а також у спеціалізованих приміщеннях для людей з особливими потребами, де важлива автоматична або сенсорна адаптація освітлення до зовнішніх умов.

Запропоноване рішення є гнучким, модульним і масштабованим. Воно поєднує апаратні засоби та програмні алгоритми, здатні адаптуватися до конкретних сценаріїв застосування, тим самим підвищуючи комфорт, безпеку та енергоефективність у житлових і робочих просторах.

Таким чином, проєкт робить практичний внесок у розвиток сфери автоматизації будинку, сприяючи популяризації інтелектуальних систем керування освітленням, а також слугує прикладом інтеграції вбудованих систем, бездротового зв'язку та сенсорних технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Philips Hue // [Електронний ресурс]. — Режим доступу:
https://en.wikipedia.org/wiki/Philips_Hue — Дата доступу: травень 2025
2. Mi Smart LED Bulb Essential // [Електронний ресурс]. — Режим доступу:
<https://homehub.com.ua/uk/smart-lampy/mi-smart-led-bulb-essential-white-and-color-uk.html> — Дата доступу: травень 2025
3. LightSwitch Dimmer Jeweller // [Електронний ресурс]. — Режим доступу:
<https://ajax.systems.ua/products/lightswitch-dimmer-jeweller/> — Дата доступу: травень 2025
4. STM32F411CE Datasheet // [Електронний ресурс]. — Режим доступу:
<https://www.st.com/resource/en/datasheet/stm32f411ce.pdf> — Дата доступу: травень 2025
5. How to optimize the ADC accuracy in the STM32 MCUs // [Електронний ресурс]. — Режим доступу:
https://www.st.com/resource/en/application_note/cd00211314-how-to-optimize-the-adc-accuracy-in-the-stm32-mcus-stmicroelectronics.pdf — Дата доступу: травень 2025
6. I2C – Протокол обміну даними // [Електронний ресурс]. — Режим доступу: <https://www.itmaster.biz.ua/directory/standarts/i2c.html> — Дата доступу: травень 2025
7. SPI – Послідовний периферійний інтерфейс // [Електронний ресурс]. — Режим доступу: <https://www.itmaster.biz.ua/directory/standarts/spi.html> — Дата доступу: травень 2025
8. UART – Асинхронний послідовний інтерфейс // [Електронний ресурс]. — Режим доступу: <https://www.itmaster.biz.ua/directory/standarts/uart.html> — Дата доступу: травень 2025
9. UART в STM32 // [Електронний ресурс]. — Режим доступу:
<https://www.itmaster.biz.ua/electronics/stm32/stm32-uart.html> — Дата доступу: травень 2025

10. FreeRTOS для STM32 (частина 1) // [Електронний ресурс]. — Режим доступу: <https://itmaster.biz.ua/programming/linux/freertos1.html> — Дата доступу: травень 2025
11. FreeRTOS Official Site // [Електронний ресурс]. — Режим доступу: <https://www.freertos.org/> — Дата доступу: травень 2025
12. ESP8266 Datasheet // [Електронний ресурс]. — Режим доступу: https://components101.com/sites/default/files/component_datasheet/ESP8266%20Datasheet.pdf — Дата доступу: травень 2025
13. Фотоелемент 5516 Datasheet // [Електронний ресурс]. — Режим доступу: <https://yourduino.com/docs/Photoresistor-5516-datasheet.pdf> — Дата доступу: травень 2025
14. Мова програмування C // [Електронний ресурс]. — Режим доступу: [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language)) — Дата доступу: травень 2025
15. STM32CubeMX // [Електронний ресурс]. — Режим доступу: <https://www.st.com/en/development-tools/stm32cubemx.html> — Дата доступу: травень 2025
16. STM32CubeProg // [Електронний ресурс]. — Режим доступу: <https://www.st.com/en/development-tools/stm32cubeprog.html> — Дата доступу: травень 2025
17. STM32CubeIDE // [Електронний ресурс]. — Режим доступу: <https://www.st.com/en/development-tools/stm32cubeide.html> — Дата доступу: травень 2025
18. STM32Cube MCU/MPU Packages // [Електронний ресурс]. — Режим доступу: <https://www.st.com/en/embedded-software/stm32cube-mcu-mpu-packages.html> — Дата доступу: травень 2025
19. Git Documentation // [Електронний ресурс]. — Режим доступу: <https://git-scm.com/doc> — Дата доступу: травень 2025