

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Завідувач кафедри

(підпис)

“ ” червня 2025 р.

на здобуття ступеня бакалавра

за освітньо-професійною програмою

**«Системне програмування та спеціалізовані комп'ютерні системи»
спеціальності 123 «Комп'ютерна інженерія»**

на тему: «Портативна система моніторингу розумного освітлення на основі ESP32»

студент IV курсу, групи КВ-13

Гармаш Дмитро Олегович

(підпис)

ас.каф. СПіСКС, д.ф.

Сергієнко Павло Анатолійович

(підпис)

Консультант з нормоконтролю

доц. каф. СПСКС, к.т.н., доцент, Клятченко Я.М.

(підпис)

Рецензент професор каф. ОТ, д.т.н., проф.

Гордієнко Юрій Григорович

(підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань
Студент

(підпис)

Київ – 2025 року

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

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування та спеціалізованих комп'ютерних систем
Рівень вищої освіти – перший (бакалаврський)
Освітньо-професійною програма
«Системне програмування та спеціалізовані
комп'ютерні системи»
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віталій РОМАНКЕВИЧ

(підпис)

(ініціали, прізвище)

“ ____ ” червня 2025 р.

ЗАВДАННЯ

на дипломний проект студента

Гармаша Дмитра Олеговича

1. Тема проекту «Портативна система моніторингу розумного освітлення на основі ESP32», керівник проекту ас.каф. СПіСКС, д.ф., Сергієнко Павло Анатолійович, затверджені наказом по університету від «29» травня 2025 р. №1808-С
2. Термін подання студентом проекту _____
3. Вихідні дані до проекту див. Технічне завдання
4. Зміст пояснювальної записки
 - Аналіз наявних рішень та огляд літератури
 - Огляд обраних технологій та апаратних пристроїв
 - Реалізація програмного забезпечення моніторингу розумного освітлення на основі ESP32
 - Оцінка роботи розробленої системи
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):
 - Схема функціональна;
 - Блок-схема алгоритму функціонування системи;
 - Структурна схема взаємодії модулів підпрограм;
 - Структурна схема зв'язків програмних та апаратних модулів.

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 «3» січня 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення літератури за тематикою проекту	03.01.2025	
2.	Розроблення та узгодження технічного завдання	17.01.2025	
3.	Аналіз існуючих рішень	19.02.2025	
4.	Підготовка матеріалів першого розділу проекту	05.05.2025	
5.	Підготовка матеріалів другого розділу проекту	12.05.2025	
6.	Розробка програмного забезпечення проекту	18.05.2025	
7.	Підготовка матеріалів графічної частини	24.05.2025	
8.	Оформлення документації дипломного проекту	26.05.2025	
9.	Попередній огляд матеріалів на кафедрі	27.05.2025	

Студент

(підпис)

Дмитро ГАРМАШ

(ініціали, прізвище)

Керівник проекту

(підпис)

Павло СЕРГІЄНКО

(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проекту

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (63 с., 15 рис., 6 табл., список використаної літератури з 18 найменувань, 6 додатків)

Об'єкт розробки – процес автоматичного керування освітленням.

Предмет дослідження — апаратні та програмні засоби для регулювання освітлення на основі сенсорних даних датчиків GY-30 BH1750FVI та TCS34725.

Мета роботи – створення портативної автономної системи освітлення з локальною нейромережею для адаптації яскравості та кольорової температури світла.

У процесі реалізації було проаналізовано існуючі рішення, розроблено технічні вимоги, створено апаратну частину та програмну логіку. Результати тестування підтвердили ефективність адаптації освітлення та потенціал впровадження системи в побутових і комерційних просторах.

Основу системи становить мікроконтролер ESP32, що забезпечує періодичний збір даних з датчиків освітленості та кольорової температури, обробку через вбудовану нейромережу та керування світлодіодним освітленням. Система працює локально, без хмарних сервісів, у режимі зниженого енергоспоживання.

У проєкті реалізовано локальну нейромережу для прогнозу кольорової температури без підключення до хмари, досягнуто точності прогнозування з відхиленням менше 2% та забезпечено стабільну автономну роботу системи без потреби у втручанні користувача протягом усього циклу функціонування.

Ключові слова: ESP32, Освітленість, Автоматизація, Нейронна мережа, Енергозбереження.

ANNOTATION

The qualification work includes an explanatory note (63 p., 15 fig., 6 tables, a list of used literature from 18 items, 6 appendices)

The object of development is the process of automatic lighting control.

The subject of research is hardware and software for regulating lighting based on sensor data from GY-30 BH1750FVI and TCS34725 sensors.

The purpose of the work is to create a portable autonomous lighting system with a local neural network for adapting the brightness and color temperature of light.

During the implementation process, existing solutions were analyzed, technical requirements were developed, and hardware and software logic were created. The test results confirmed the effectiveness of lighting adaptation and the potential for the system to be implemented in domestic and commercial spaces.

The system is based on an ESP32 microcontroller, which provides periodic data collection from illumination and color temperature sensors, processing via a built-in neural network, and control of LED lighting. The system operates locally, without cloud services, in a low-power mode.

The project implemented a local neural network for color temperature prediction without a connection to the cloud, achieved prediction accuracy with a deviation of less than 2%, and ensured stable autonomous operation of the system without the need for user intervention throughout the entire operating cycle.

Keywords: ESP32, Illumination, Automation, Neural network, Energy saving.

Зм.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045420.002 ТЗ	Портативна система	4		
			моніторингу розумного			
			освітлення на основі ESP32			
			Технічне завдання			
	A4	ІАЛЦ.045420.003 ТП	Портативна система	1		
			моніторингу розумного			
			освітлення на основі ESP32			
			Відомість технічного проєкту			
	A4	ІАЛЦ.045420.004 ПЗ	Портативна система	63		
			моніторингу розумного			
			освітлення на основі ESP32			
			Пояснювальна записка			
	A4	ІАЛЦ.045420.005 Е2	Портативна система	1		
			моніторингу розумного			
			освітлення на основі ESP32			
			Схема функціональна			
	A4	ІАЛЦ.045420.006 Д1	Портативна система	1		
			моніторингу розумного			
			освітлення на основі ESP32			
			Блок-схема алгоритму			
			функціонування системи			

					ІАЛЦ. 045420.001 ОА								
Зм.	Арк.	№ докум.	Підпис	Дата	Портативна система моніторингу розумного освітлення на основі ESP32				Літ.	Аркуш	Аркушів		
Розробив	Гармаш Д.О.										1	2	
Перевірив	Сергієнко П.А.								КПІ ім. Ігоря Сікорського ФПМ КВ-13				
Н.контр.	Клятченко Я.М.												
Затвердив	Романкевич В.О.				Опис альбома								

[illegible]

3MICT

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

					ІАЛЦ. 045420.002 ТЗ													
Зм.	Арк.	№ докум.	Підпис	Дата	Портативна система моніторингу розумного освітлення на основі ESP32 Технічне завдання													
Розробив	Гармаш Д.О.																	
Перевірив	Сергієнко П.А.																	
Н.контр.	Клятченко Я.М.																	
Затвердив	Романкевич В.О.				Літ.			Аркуш			Аркушів							
											1				4			
					КПІ ім. Ігоря Сікорського ФПМ КВ-13													

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Портативна система моніторингу розумного освітлення на основі ESP32»

Галузь застосування: покращення умов освітлення в робочих та освітніх приміщеннях

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 розробка автоматизованої системи моніторингу освітлення з гнучкістю, забезпечуваною штучним інтелектом

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами розробки є інформація, взята з:

- офіційної технічної документації до мікроконтролера ESP32, датчика освітленості, датчика кольору температури та дисплея;
- науково-технічної літератури, публікацій, електронних статей з відкритих джерел Інтернету, пов'язаних з програмуванням мікроконтролерів;
- відеоматеріалів навчального та демонстраційного характеру, що включають в себе відеолекції та практичні приклади розробок на базі мікроконтролерів ESP32

					<i>ІАЛЦ. 045420.002 ТЗ</i>	Арк.
						2
Зм.	Арк	№ докум.	Підпис	Дата		

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється:

- Здатність отримання з датчиків освітленості та кольорової температури;
- Відображення поточних значень на дисплеї;
- Передача даних до веб-інтерфейсу або хмарного серверу;
- Відправка через HTTP-запити даних до веб-інтерфейсу або хмарного серверу;
- Візуалізація історії вимірювань у разі зникнення з'єднання;
- Автоматичне підключення до заданої Wi-Fi мережі при увімкненні живлення.

5.2 Мінімальні вимоги до апаратного забезпечення:

- Мікроконтролер ESP32 з підтримкою живлення 3.3 В або 5 В;
- Наявність двох інтерфейсів I2C, одного інтерфейсу SPI та одного інтерфейсу UART;
- Вбудована флеш-пам'ять від 4 Мб, оперативна пам'ять – від 520 Кб;
- TFT-дисплей з роздільною здатністю не менше 128x160 пікселів;
- Процесор Intel Core Duo T2300;
- Оперативна пам'ять: від 4 Гб;
- Простір на диску: від 2 Гб;
- Тип ОС: 64-розрядна

5.3 Вимоги до апаратного та програмного забезпечення користувача

- Операційна система Windows 10/11, Ubuntu 20.04+ або MacOS X 10.15+;
- Програмне забезпечення: Visual Studio Code та фреймворк ESP-IDF

					<i>ІАЛЦ. 045420.002 ТЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		3

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1	Вивчення літератури за тематикою проєкту	03.01.2025
2	Розроблення та узгодження технічного завдання	17.01.2025
3	Аналіз існуючих рішень	19.02.2025
4	Підготовка матеріалів першого розділу дипломного проєкту	05.05.2025
5	Підготовка матеріалів другого розділу дипломного проєкту	12.05.2025
6	Підготовка матеріалів третього розділу дипломного проєкту	18.05.2025
7	Підготовка графічної частини дипломного проєкту	24.05.2025
8	Оформлення документації дипломного проєкту	26.05.2025
9	Попередній огляд матеріалів диплому на кафедрі	27.05.2025

					<i>ІАЛЦ. 045420.002 ТЗ</i>	Арк.
						4
Зм.	Арк	№ докум.	Підпис	Дата		

[illegible]

Пояснювальна записка

до дипломного проєкту

на тему: «Портативна система моніторингу розумного освітлення на основі ESP32»

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	3
ВСТУП.....	5
1. АНАЛІЗ НАЯВНИХ РІШЕНЬ ТА ОГЛЯД ЛІТЕРАТУРИ.....	7
1.1 Значення моніторингу освітлення в сучасному середовищі	7
1.2 Класифікація пристроїв для моніторингу освітлення	8
1.2.1 За типом монтажу.....	8
1.2.2 За типом вимірювального сигналу	9
1.2.3 За способом живлення	10
1.2.4 За способом обробки та передачі даних	10
1.2.5 За рівнем «розумності».....	11
1.2.6 За сферою застосування.....	11
1.3 Порівняння зі схожими рішеннями на ринку.....	12
1.4 Потенційні проблеми стандартних систем моніторингу	15
1.5 Світовий ринок сенсорів освітлення	17
1.6 Порівняння розглянутих рішень згідно з класифікаційними ознаками....	18
2. ОГЛЯД ОБРАНИХ ТЕХНОЛОГІЙ ТА АПАРАТНИХ ПРИСТРОЇВ	20
2.1 Загальний принцип роботи та обґрунтування вибору ESP32.....	20
2.2 Опис мікроконтролера ESP32 CH340C 30P Type-C	24
2.3 Опис датчика GY-30 BH1750FVI	26
2.4 Опис датчика TCS34725	28
2.5 Опис дисплея 1.8" TFT IPS ST7789	30
2.6 Проектування апаратної частини.....	31
2.7 Опис та обґрунтування вибору мови програмування C.....	35
2.8 Опис та обґрунтування потреби проекту в ШІ	37
2.9 Опис інструментів розробки	40

					<i>ІАЛЦ. 045420.004 ПЗ</i>		
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Розробив	Гармаш Д.О.				<i>Портативна система моніторингу розумного освітлення на основі ESP32</i> <i>Технічне завдання</i>	<i>Літ.</i>	<i>Аркуш</i>
Перевірів	Сергієнко П.А.						
						1	63
Н.контр.	Клятченко Я.М					КПІ ім. Ігоря Сікорського ФПМ КВ-13	
Затвердив	Романкевич В.О.						

3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	
МОНІТОРИНГУ РОЗУМНОГО ОСВІТЛЕННЯ НА ОСНОВІ ESP32	43
3.1 Налаштування роботи ESP-IDF та структура проєкту	43
3.2 Програмна реалізація мікроконтролеру ESP32 CP340C TYPE-C	44
3.3 Програмна реалізація роботи сенсору GY-30 BH1750FVI	45
3.4 Програмна реалізація роботи сенсору TCS34725	47
3.5 Програмна реалізація роботи дисплею	47
3.6 Програмна реалізація модулю штучного інтелекту	50
4. ОЦІНКА РОБОТИ РОЗРОБЛЕНОЇ СИСТЕМИ	52
4.1 Загальний принцип роботи та характеристика	52
4.2 Персоналізація освітлення за допомогою алгоритмів ШІ	55
4.3 Сценарій використання в домашніх умовах.....	56
4.4 Сценарій використання в робочих умовах	56
4.5 Поведінка в нестандартних ситуаціях.....	57
4.6 Потенційне розширення можливостей.....	58
ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	61

ДОДАТКИ

Додаток А. ІАЛЦ.045420.005 Е2. Схема функціональна.

Додаток Б. ІАЛЦ.045420.006 Д1. Блок-схема алгоритму функціонування системи.

Додаток В. ІАЛЦ.045420.007 Д2. Структурна схема взаємодії модулів підпрограм.

Додаток Г. ІАЛЦ.045420.008 Д3. Структурна схема зв'язків програмних та апаратних модулів.

Додаток І. Презентація.

Додаток Д. Лістинг коду.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						2
Зм.	Арк	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

DIY (Do It Yourself) – тип розробки без залучення спеціалістів

лк (люкси) – одиниця вимірювання освітленості (лм/м²)

CRI (Color Renrereng Index) – індекс точності передачі кольору світлом

PAR (Photosynthetic Active Radiation) – фотосинтетичне активне випромінювання

В (Вольт) – одиниця вимірювання напруги

USB (Universal Serial Bus) – послідовний інтерфейс для підключення периферії

IoT (Internet of Things) – концепція об'єднання фізичних пристроїв у мережу для автоматизованого обміну даними та керування

УФ (Ультрафіолет) – електромагнітне випромінювання з довжиною хвилі <400 нм

RISC – Reduced Instruction Set Computer

CISC – Complex Instruction Set Computer

3V3 – напруга 3.3 Вольта

GND – Ground

PMU – Process Management Unit

ROM – Read-Only Memory

GPIO (General Purpose Input/Output) – універсальні порти вводу/виводу

UART (Universal Asynchronous Receiver/Transmitter) – асинхронний серійний інтерфейс передачі даних

I2C (Inter-Integrated Circuit) – синхронна двопровідна шина з адресацією пристроїв

MOSI (Master Out Slave In) – лінія передачі даних від ведучого до веденого

MISO (Master In Slave Out) – лінія зчитування даних веденим у ведучого

SCK, SCLK (Serial Clock) – лінія активації вибраного SPI-пристрою

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						3
Зм.	Арк	№ докум.	Підпис	Дата		

SPI (Serial Peripheral Interface) – синхронна послідовна шина з кількома лініями

CS (Chip Select) – лінія активації вибраного SPI-пристрою

SRAM (Static Random Access Memory) – енергонезалежна статична оперативна пам'ять

AES (Advanced Encryption Standard) – блоковий симетричний алгоритм шифрування даних

RSA (Rivest, Shamir, Adleman) – алгоритм асиметричного шифрування з відкритим ключем

SHA (Secure Hash Algorithm) – криптографічна хеш-функція фіксованої довжини

WPA (Wi-Fi Protected Access) – протокол шифрування бездротових з'єднань

TLS (Transport Layer Security) – протокол захищеної передачі даних в мережі

АЦП – Аналогово-Цифровий Перетворювач

STL – Standard Template Library

ШІ – Штучний інтелект

RTC (Real-Time Clock) – автономний модуль точного хронометрування

MQTT (Message Queuing Telemetry Transport) – брокерський протокол телеметрії з низьким навантаженням

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						4
Зм.	Арк	№ докум.	Підпис	Дата		

ВСТУП

Освітлення є не просто фізичним явищем, а одним із пріоритетних факторів, що впливають на сучасну людину, середовище та розвиток технологій. Цей вплив легко недооцінити, бо освітлення має пряме відношення не тільки до сприйняття навколишнього середовища, фізіологічного та емоційного стану людини, ефективності праці, безпеки, екології та навіть до стратегічних аспектів в умовах війни. Це стосується таких галузей, як архітектура, торгівля, агросектор, побут тощо. Однак надмірна або нераціональна кількість світла призводить до світлового забруднення, порушень в функціонуванні організму людини та техногенного навантаження на довкілля.

У сучасному світі зростання кількості світлових пристроїв та вартості енергоносіїв набуває все більшого значення. За даними Міжнародного Енергетичного Агентства (ІЕА), станом на 2024-й рік освітлення споживає приблизно 12% світлової електроенергії, з яких значна частина витрачається нераціонально. Цей показник зменшується останні роки завдяки впровадженню LED-освітлення та інших енергоефективних технологій. Актуальним стає пошук доступних та ефективних рішень, які здатні динамічно реагувати на зміну умов і потреб, зберігаючи комфорт користувача одночасно знижуючи енергоспоживання. Таким рішенням є розумні системи освітлення, які здатні не лише вимірювати, а і реагувати на рівень освітленості, регулювати кольорову температуру та прогнозувати оптимальні параметри світла. Таким чином, запропонована тема поєднує в собі запит на мобільність, інтелектуальну обробку даних, простоту використання та модернізації, а також повну автономність роботи без залежності від наявності інтернету, що повністю відповідає вимогам до сучасних розумних систем освітлення.

Основу такої системи доцільно реалізувати на базі мікроконтролерів ESP32, STM32, Raspberry Pi Pico та ін., та сенсорів освітлення, які доступні у продажу та не потребують вузькопрофесійних знань. Метою проєкту є

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						5
Зм.	Арк	№ докум.	Підпис	Дата		

створення такого пристрою, який буде одночасно портативним, гнучким, у налаштуванні та здатним працювати повністю локально. Під час дослідження проводилось вивчення наукових джерел, моделювання, програмування, складання прототипу, тестування в реальних умовах та аналіз результатів.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						6
Зм.	Арк	№ докум.	Підпис	Дата		

1. АНАЛІЗ НАЯВНИХ РІШЕНЬ ТА ОГЛЯД ЛІТЕРАТУРИ

1.1 Значення моніторингу освітлення в сучасному середовищі

У сучасному світі світло стало не лише джерелом комфорту, а й фактором, який прямо впливає на здоров'я, енергоефективність, продуктивність праці та екологічний баланс. Дослідження показують, що недостатнє або надмірне освітлення призводить до зорової втоми, головного болю, зниження концентрації, порушень сну та загального погіршення психофізіологічного стану людини [1]. У випадках системного впливу це переростає в хронічні захворювання або втрату працездатності. Це стосується не тільки офісних працівників, але й робітників промислових підприємств, персоналу в лікарнях, учасників освітнього процесу в навчальних закладах та ін..

Ще одна важлива проблема — світлове забруднення. За даними дослідження Kuva et al. (2017), загальна яскравість нічного освітлення на планеті зростає щороку на 2,2%, що погіршує екологічну ситуацію та порушує добові біоритми людей і тварин [1]. У великих містах нічна яскравість освітлення у рази перевищує безпечні межі, і це негативно позначається на якості сну, який в свою чергу сприяє розвитку хронічної втоми, гормональному дисбалансу та порушенню психологічного стану.

Неефективне або неконтрольоване освітлення — це також значне енергоспоживання. У середньому до 40% енергоспоживання будівель припадає на освітлення [2]. В багатьох випадках ці витрати можна було б скоротити за рахунок адаптивного керування. Для цього необхідні системи моніторингу, які можуть оперативно реагувати на зміни умов — рівень природного освітлення, присутність людей у приміщеннях. Більшість таких систем є стаціонарними, вимагають централізованих хабів, або нераціональні у використанні для звичайного користувача через економічну недоцільність.

Особливого значення контроль освітлення набуває в умовах підвищеної складності та відповідальності, умовах нестабільного або відсутнього

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						7
Зм.	Арк	№ докум.	Підпис	Дата		

енергопостачання: польові лабораторії, військові об'єкти, мобільні медичні пункти, сільськогосподарські об'єкти та ін.. В цих умовах відсутнє мережеве енергопостачання, доступ до мережі Інтернет. Використовуються автономні, портативні пристрої для генерування електроенергії. Комерційні рішення, як правило, орієнтовані на стаціонарне використання, потребують мережевого живлення, зв'язку з хмарними платформами, інтегровані в загальні інженерні мережі, що обмежує їх використання в польових умовах.

1.2 Класифікація пристроїв для моніторингу освітлення

Під час розробки будь-якого пристрою для управління освітленістю важливо спиратись не лише на технічні характеристики, а й розуміти, як класифікуються подібні пристрої, чим вони відрізняються один від одного та в яких сферах найчастіше застосовуються, адже кінцевий вибір користувача напряму залежить від сценаріїв його використання та фінансових можливостей. У сучасних системах моніторингу освітлення можна виділити кілька ключових критеріїв класифікації, які дозволяють систематизувати пристрої відповідно до їх функцій, принципів роботи, сфер застосування та специфіки технічної реалізації.

1.2.1 За типом монтажу

Стаціонарні пристрої встановлюються на постійній основі в певному місці — наприклад, на стелі чи на стіні приміщення. Такі рішення найчастіше входять до складу автоматизованих систем освітлення або моніторингу мікроклімату, зокрема у «розумному будинку» чи системах промислової автоматизації. Вони зазвичай не є автономними, бо потребують живлення від електромережі й дротового підключення до контролера або хабу.

Портативні пристрої — це компактні або малі електронні системи, місце розташування яких можна легко змінювати, закріплювати за допомогою

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						8
Зм.	Арк	№ докум.	Підпис	Дата		

засобів тимчасового монтажу в необхідних місцях. Такі пристрої зазвичай живляться від акумуляторів або батарей та мають бездротові інтерфейси для обміну даними. Вони застосовуються для тимчасового або локального управління освітленням в будь-яких умовах. Прикладами можуть бути стандартні люксеметри або DIY-пристрої на базі Arduino чи ESP32, які використовуються для освітніх або прикладних цілей [3].

Вбудовані сенсори — це пристрої, інтегровані у більші системи або техніку: наприклад, датчик освітлення в смартфоні, що автоматично регулює яскравість екрана, або світловий сенсор у корпусі термостата чи смарт-лампи. Вони зазвичай не мають власного інтерфейсу й функціонують як частина комплексу.

1.2.2 За типом вимірювального сигналу

Оскільки від цього залежать можливості пристрою та сфери його застосування:

Люксеметри — це найпоширеніший тип пристроїв, які вимірюють освітленість у люксах. Ці прилади широко застосовуються для перевірки відповідності умов освітлення до нормативів, як у побуті, так і в промислових умовах. Вони мають просту будову: фотодіод з фільтром і цифровий (або аналоговий) індикатор.

Спектрометри — прилади, що дають набагато глибше уявлення про світло. Вони вимірюють спектральний склад освітлення, включаючи колірну температуру, індекс передачі кольору (CRI), домінуючі довжини хвиль тощо. Такі пристрої використовують у фотостудіях, лабораторіях, музейній справі.

RGB-сенсори — здатні фіксувати інтенсивність основних кольорів (Red, Green, Blue) і розраховувати відтінок білого або колірну температуру. їх часто використовують у мобільних пристроях або в системах “розумного світла” Прикладом такого сенсора є TCS34725 — цифровий сенсор кольору з можливістю вимірювання освітленості.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						9
Зм.	Арк	№ докум.	Підпис	Дата		

УФ-сенсори — застосовуються в сферах, де є необхідно знати рівень ультрафіолетового випромінювання: в медицині, сільському господарстві, промисловості.

PAR-датчики — спеціалізовані сенсори, що визначають рівень світла, корисного для фотосинтезу (в діапазоні 400-700 нм). Активно застосовуються в агротехнологіях і теплицях.

1.2.3 За способом живлення

Сенсори з живленням від мережі, працюють від постійного джерела живлення — зазвичай від мережі 220 В. Це найчастіше стаціонарні системи «розумних будинків» та промисловості.

Сенсори з живленням від автономних чи енергонезалежних джерел енергії. Вони отримують живлення від батарейок, акумуляторів або сонячних панелей. Саме такі рішення мають найбільшу актуальність у мобільних, віддалених або військових умовах, де підключення до мережі неможливе або небажане.

Сенсори з гібридним типом живлення, поєднують в собі як живлення від мережі, так і живлення від альтернативних джерел енергії (акумулятори, батареї тощо). Використовуються для забезпечення безперервного вимірювання в системах, де важлива безперервність вимірювання.

1.2.4 За способом обробки та передачі даних

Дротові системи — класичний варіант з підключенням через кабелі (USB, аналогові виходи, інтерфейси типу RS-485). Вони стабільні, але обмежені в мобільності.

Бездротові системи — використовують Wi-Fi, Bluetooth або інші протоколи (Zigbee, LoRa). Це рішення, як правило, застосовується в розумних будинках, теплицях або автономних IoT-системах.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						10
Зм.	Арк	№ докум.	Підпис	Дата		

IoT-рішення — здатні передавати дані в хмарні сервіси, інтегруватися з веб-інтерфейсами, смартфонами або серверами. Прикладами є датчики INNOPRO, Xiaomi, Shelly, які обмінюються даними з додатками або хмарою.

1.2.5 За рівнем «розумності»

За участі людини — найпростіші прилади, де сенсори фіксують рівень освітленості та надають дані, а користувач самостійно аналізує дані та приймає рішення через зовнішні системи.

Автоматизовані пристрої — фіксують, самостійно обробляють та передають дані, здійснюють обмежені автоматичні дії (наприклад, регулюють рівень світла в залежності від освітлення приміщення).

Автоматизовані пристрої з використанням ШІ — новітній тренд, де штучний інтелект або алгоритми машинного навчання фіксують, фільтрують, обробляють, аналізують зміну умов та прогнозують майбутній рівень освітленості. Можуть приймати рішення без втручання людини, також можуть давати рекомендації користувачеві.

1.2.6 За сферою застосування

Побутове використання — регулювання рівня освітлення для комфорту, економії енергоспоживання. Використовуються у системах «розумного будинку», побутовій електроніці та техніці.

Промислове та комерційне використання — енергоефективне управління великими робочими зонами, офісами, торговими центрами для забезпечення нормативних вимог освітлення.

Спеціалізоване використання — застосування сенсорів освітлення для вирішення вузько направлених задач у наукових дослідженнях, де потрібна висока точність вимірювання випромінювання. Використовується у медицині, аграрному секторі тощо.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						11
Зм.	Арк	№ докум.	Підпис	Дата		

Після ознайомлення з представленими на ринку пропозиціями та порівнявши їх з наведеною вище класифікацією, можна зробити висновок, що більшість рішень на ринку:

- або мають високу вартість, що унеможливлює поширене застосування;
- або обмежені стаціонарним способом встановлення, що унеможливлює їхнє використання;
- або мають зайвий функціонал, який не буде мати практичного застосування;
- або є складними у використанні, бо потребують спеціального налаштування та інтеграції з централізованими хабами.

1.3 Порівняння зі схожими рішеннями на ринку

Для контролю освітленості приміщень на ринку доступно безліч різних смарт-пристроїв. Кожен із розглянутих пристроїв має різноманітні технічні характеристики та особливості реалізації. Щоб зрозуміти переваги і недоліки кожного продукту, розглянемо більш детально наступні рішення:

- датчик освітленості INNOPRO S-LUX-ZB;
- датчик освітленості Xiaomi Mijia Smart Light Sensor;
- датчик освітленості Philips Hue Motion Sensor;
- датчик освітленості Shelly Motion 2.

Всі чотири моделі належать до одного класу пристроїв — датчиків навколишнього освітлення з можливістю інтеграції в «розумний дім». Використовуючи ці сучасні смарт-пристрої в приміщеннях або на вулиці, можна автоматично регулювати освітлення, вмикати/вимикати світло залежно від часу доби або рівня освітленості, запускати сценарії автоматизації для енергозбереження та підвищення рівня комфорту.

INNOPRO S-LUX-ZB – це сенсор з розмірами 88x26x21 мм для вимірювання рівня освітленості в діапазоні до 83000 лк (рисунк 1.1).

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						12
Зм.	Арк	№ докум.	Підпис	Дата		



Рисунок 1.1 – датчик освітленості INNOPRO S-LUX-ZB

Живлення забезпечується від двох батарейок типу ААА. Передача даних можлива через Wi-Fi або ZigBee. ZigBee призначений для енергоощадного зв'язку між великою кількістю смарт-пристроїв у локальній мережі через шлюз, тоді як Wi-Fi забезпечує пряме, швидке з'єднання з інтернетом, але з вищим енергоспоживанням. Інтегрується в системи «розумного дому» TuYa Smart або Smart Life, які використовуються для створення автоматизованих сценаріїв. Доступна сумісність із Alexa та Google Assistant [4].

Xiaomi Mijia Smart Light Sensor — це сенсор з розмірами 40x40x12 мм для вимірювання рівня освітленості в діапазоні до 83000 лк (рисунок 1.2).



Рисунок 1.2 – Xiaomi Mijia Smart Light Sensor

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		13

Він працює від батареї типу CR2450 і передає дані через ZigBee 3. Підключення здійснюється через шлюз до застосунку Mi Home. Сенсор має лише датчик освітлення, і призначений для запуску сценаріїв автоматизації в «розумному домі», обмежений закритою екосистемою Xiaomi та потребою у хабі [5].

Модель Philips Hue Motion Sensor має розміри 62x62x28 мм, окрім сенсора освітленості пристрій має ще й сенсори руху та температури (рисунок 1.3).



Рисунок 1.3 – Philips Hue Motion Sensor

Тим самим керує рівнем освітленості в приміщенні в залежності від рівня навколишнього світла та присутності людини. Цей пристрій отримує живлення від двох батарейок AAA та є частиною екосистеми Philips Hue, яка передбачає керування через мережевий шлюз Hue Bridge. Hue Bridge підключається до домашнього Wi-Fi-роутера і утворює локальну ZigBee-мережу. Використовуючи Hue Bridge користувач має змогу створювати складні сценарії автоматизації, задавати умови активації, переглядати дані з сенсорів у реальному часі, встановлювати денні/нічні режими тощо. Крім цього, має повну сумісність з популярними помічниками — такими як Amazon Alexa, Google Assistant та Apple HomeKit, за допомогою яких можливо використовувати голосові команди для керування світлом та створювати

мультиплатформенні сценарії без взаємодії з смартфоном [6].

Shelly Motion 2 може використовуватись як охоронний елемент, як сенсор системи освітлення або кліматичної системи, також як частина комплексної логіки «розумного будинку» (рисунок 1.4).



Рисунок 1.4 – Shelly Motion 2

Має розміри 89.3x41.4x22.9 мм, об'єднує в собі кілька сенсорів: освітленості, руху та температури. Вимірює рівень освітлення в діапазоні до 65000 лк. Працює від вбудованого літій-іонного акумулятора 6500 мА*год. Передача даних відбувається через Wi-Fi-з'єднання. Сумісний з системами Home Assistant, Node-RED, Google Home та Alexa. Має повноцінний локальний API прямо на пристрої та не потребує хмарного сервісу або додаткових хабів [7].

1.4 Потенційні проблеми стандартних систем моніторингу

Системи автоматизації, моніторингу та керування освітленням дедалі

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						15
Зм.	Арк	№ докум.	Підпис	Дата		

частіше впроваджують не лише в комерційних об'єктах, а й у житлових будинках. Стрімке зростання доступності смарт-пристроїв серед широкого кола користувачів стало наслідком зниження вартості цих систем на ринку. У багатьох випадках це досягається шляхом спрощення функціональності, а саме обмежується найпростішими можливостями, а це істотно впливає на якість роботи систем моніторингу.

Сенсори вимірюють лише загальний рівень освітленості в люксах і не здатні визначати колірну температуру та спектральний склад, тому вони непридатні до точного налаштування «теплого» чи «холодного» світла в приміщенні. Не мають функції калібрування та компенсації температури. Як наслідок, світлодіод або фотодіод сенсора нездатний показувати коректні значення при різних температурах в приміщенні. За межами середнього діапазону сенсори мають високу похибку від $\pm 15\text{-}25\%$. З часом багато сенсорів схильні до старіння фоточутливих компонентів. У результаті стають менш чутливими до змін рівня освітленості та дають неточні значення.

Для економії енергії сенсор знаходиться в режимі сну та оновлення даних відбувається один раз в певний проміжок часу, тим самим унеможливорює швидке реагування в сценаріях автоматизації. Для більшої доступності сенсори живляться від батарейок, які, на відміну від акумуляторів, не можна заряджати.

Більшість продуктів не мають локального управління та працюють лише за наявності інтернет-з'єднання, тому дані доступні лише через хмарний додаток. Відсутня сумісність між сенсорами та платформами різних виробників. Кожен виробник реалізовує власний закритий протокол обміну даними або інтегрує сенсор лише в свою екосистему. Технології зв'язку Wi-Fi, ZigBee, Bluetooth також не гарантують повної взаємодії між сенсорами та платформою та можуть втрачати сигнал у приміщеннях з залізобетонними стінами та перекриттями. Наприклад, для ZigBee-пристрою може знадобитися окремий хаб кожного бренду, навіть якщо технічно протокол один і той самий. Тому розробки на основі доступних компонентів з відкритим API дають

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						16
Зм.	Арк	№ докум.	Підпис	Дата		

споживачам більше гнучкості під час використання.

У системі «розумного будинку» багато пристроїв збирають та передають особисту інформацію на власні хмарні сервіси, викликаючи занепокоєння щодо конфіденційності особистих даних. Наприклад: за даними досліджень, пристрої Amazon Alexa збирає 28 із 32 можливих типів даних, а Google Home збирають 22 типи даних [8]. Ці дані можуть зберігатися в інших країнах, без повного контролю з боку користувача. Гарантія компанії щодо конфіденційності не унеможливило зламу, витоку або можливості використання даних у рекламних цілях.

1.5 Світовий ринок сенсорів освітлення

Світове споживання систем моніторингу освітлення демонструє стабільне зростання. Європа і Азія виступають ключовими регіонами збуту. Ринок Азії перевищує європейський за обсягом і демонструє вищі темпи зростання. В 2024 році частка Європи в світовому попиті оцінюється в 22%, тоді як на частку Азії припадає близько 29%. Середньорічні темпи зростання становили близько 6-7% [9].

Уряди країн Європи запровадили суворі норми енергоефективності та стандарти освітлення, тим самим мотивуючи користувачів використовувати прилади для контролю освітленості. Стабільний попит підтримується проектами «розумних будинків» та зеленого будівництва. Ринок ЄС і надалі буде підтримуватися регуляторними вимогами та модернізацією інфраструктури освітлення. Очікується впровадження систем автоматичного регулювання освітлення в офісах і громадських приміщеннях, що вимагатиме використання якісних люксометрів для налаштування та моніторингу освітленості [10].

У країнах Східної та Південної Азії стрімка урбанізація, будівництво сучасних хмарочосів, розбудова смарт-міст, увага до аграрних і інфраструктурних проектів та наявність великої кількості локальних

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						17
Зм.	Арк	№ докум.	Підпис	Дата		

виробників доступних за ціною приладів стимулює попит на системи моніторингу освітленості для оптимізації енергоспоживання.

Світовий ринок характеризується відносно невеликою кількістю гравців, тому нові компанії (особливо з Китаю та Тайваню) будуть виходити з інноваційними моделями люкметрів з розширеним функціоналом. Впровадження штучного інтелекту відкриває нові можливості ринку сенсорів освітлення та дозволяє зменшити енергоспоживання, підвищити рівень комфорту та знизити потреби в постійному втручанні людини. Поєднання систем моніторингу з локальними або хмарними сервісами штучного інтелекту визначатиме конкурентоспроможність нових пристроїв на ринку у найближчі роки. Тим самим зробить моніторинг освітлення ще доступнішим для широкого кола споживачів. Аналітики прогнозують стабільне щорічне зростання ~6-8% у наступні 5 років, підкріплене тенденціями енергоефективності та цифровізації [10]. В продажі присутня продукція як великих електронних корпорацій, так і спеціалізованих компаній вимірювальної техніки. Серед лідерів називають: HTC Instruments (ІндіяТайвань), UNI-T (Китай), Ніюкі (Японія), Chauvin Amoux (Франція), Extech (FLIR Systems, США) та ін..

1.6 Порівняння розглянутих рішень згідно з класифікаційними ознаками

Згідно з раніше поданою класифікацією пристроїв для моніторингу освітлення, розробка орієнтується на:

- Портативний формфактор (за типом монтажу);
- Автономне живлення та роботу без хабів (за типом живлення);
- Локальну обробку даних без хмарних сервісів (за способом обробки);
- Використання RGB-датчика для аналізу кольорової температури (за типом вимірювання);
- Високий ступінь автономії з елементами ШІ (за рівнем «розумності»);
- Застосування у побуті та офісних просторах (за сферою використання).

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						18
Зм.	Арк	№ докум.	Підпис	Дата		

Відповідність зазначеним критеріям у контексті розглянутих пристроїв узагальнено в Таблиці 1.1:

Таблиця 1.1 – Порівняння пристроїв за класифікаційними ознаками

Пристрій	Тип монтажу	Тип сигналу	Живлення	Тип обробки та передачі даних	Рівень «розумності»	Сфера застосування
INNOPRO S-LUX-ZB	Стационарний	Люксметр	Дві ААА батарейки	ZigBee, Wi-Fi	Автоматизований	Побут, офіс
Xiaomi Mijia Light Sensor	Стационарний	Люксметр	CR2450 батарейка	ZigBee	Автоматизований	Побут
Philips Hue Motion Sensor	Стационарний	Люксметр + рух + температура	Дві ААА батарейки	ZigBee через Hue Bridge	Автоматизований	Побут
Shelly Motion 2	Стационарний	Люксметр + рух + температура	Акумулятор 6500 мА*год	Wi-Fi, локальний API	Автоматизований	Побут, офіс, охорона

2. ОГЛЯД ОБРАНИХ ТЕХНОЛОГІЙ ТА АПАРАТНИХ ПРИСТРОЇВ

2.1 Загальний принцип роботи та обґрунтування вибору ESP32

ESP32 - це сімейство мікроконтролерів виробництва китайської компанії Espressif Systems Co., Ltd., яке випускається з використанням або одноядерного або двоядерного процесора, зазвичай під управлінням операційної системи реального часу FreeRTOS. Мікроконтролери ESP32 побудовані на базі 32-розрядної архітектури Xtensa гарвардського типу з набором команд RISC з можливістю створювати спеціалізовані процесорні ядра, які оптимізовані для роботи з аудіо, відео, обробкою сигналів, керуванням штучним інтелектом та мережами тощо [11].

У RISC-архітектурі виконання інструкцій відбувається поетапно та з розподілом функцій між компонентами. Спочатку програмні інструкції завантажуються з основної пам'яті (Main Memory) до кешу інструкцій (Instruction Cache), наявність якого значно прискорює доступ до них. З кешу інструкції передаються до апаратного блоку керування (Hardwired Control Unit), який декодує команду та генерує відповідні сигнали управління для виконання дії. Далі ці сигнали надходять у виконавчий блок (Data Path), який реалізує операцію — арифметичну, логічну чи зчитування запис даних. Якщо для виконання потрібні дані, виконавчий блок звертається до кешу даних (Data Cache), який зберігає інформацію з основної пам'яті, яка часто використовується. У разі відсутності цієї інформації у кеші даних здійснюється звернення до основної пам'яті. Після обробки результати можуть бути записані назад у кеш даних або до основної пам'яті (рисунок 2.1).

RISC-архітектура навмисно обмежує набір команд, щоб кожна інструкція виконувалася максимально швидко — зазвичай за один такт процесора. Цей підхід протилежний до CISC, де команди можуть бути складнішими, але повільнішими [12].

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						20
Зм.	Арк	№ докум.	Підпис	Дата		

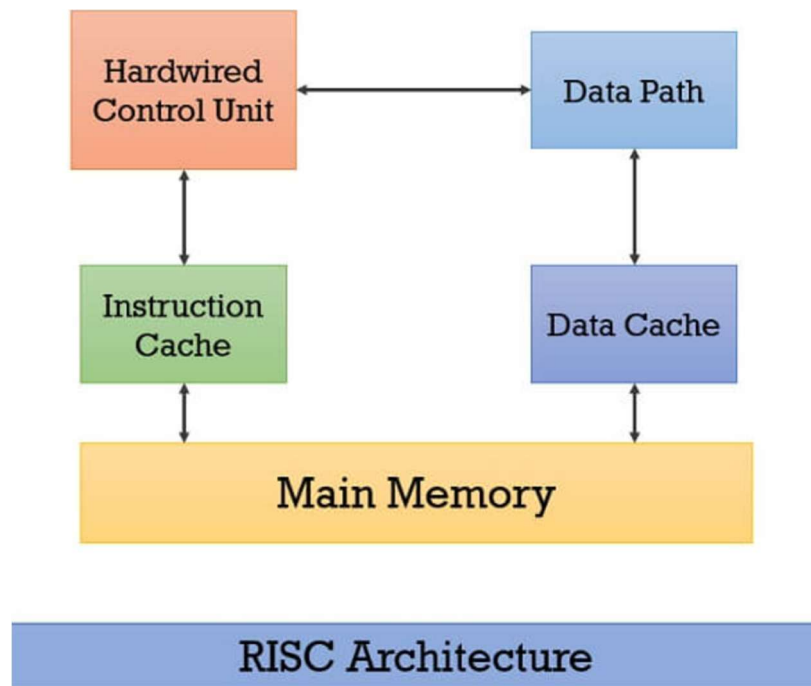


Рисунок 2.1 – Схема роботи RISC-архітектури

Робота мікроконтролера ESP32 починається з того, що на нього поступає стабільне живлення значенням 3.3 В. Живлення подається на піни основного живлення та заземлення. Якщо напруга нестабільна або нижча за мінімум, то ESP32 буде перезавантажуватись або не запускатись взагалі. Внутрішній модуль Brownout Detector весь час вимірює напругу живлення, тим самим запобігаючи уникненню ситуацій помилок доступу до флеш-пам'яті, некоректного виконання команд процесора та порушення роботи SRAM. Після подачі живлення відбувається активація блоку керування живленням (PMU), який розподіляє подачу напруги на внутрішні блоки ядра, пам'ять та периферію, задає послідовності запуску для вмикання компонентів в коректному порядку та керує споживанням енергії. Далі ESP32 запускає внутрішній генератор частоти для синхронізації роботи процесора, таймерів та периферії, а потім переходить на точніший зовнішній кварцовий генератор [13].

Отримавши живлення, запускається ROM Bootloader, який вбудований в постійну пам'ять мікроконтролера. Ця програма автоматично запускається при кожному увімкненні та перезапуску ESP32, обирає джерело прошивки, готує її

до запуску або вмикає режим перепрошивки. Для цього вона зчитує піни GPIO2 для вибору режиму завантаження та піни GPIO2 чи GPIO15 залежно від режиму конфігурації плати або модуля [14]. Наступним кроком ROM Bootloader зчитує заголовок прошивки з SPI Flash, перевіряє її цілісність та виконує дешифрування (Flash Encryption), після чого завантажує код у оперативну пам'ять і передає управління основній програмі.

Після завантаження прошивки контроль виконання передається до початкового коду ініціалізації (start-up code), який відповідає за підготовку системи до роботи. Потім ініціалізується SRAM, формуються stack та heap, здійснюється початкове налаштування глобальних і статичних змінних шляхом обнулення або копіювання з флеш-пам'яті до оперативної. Паралельно налаштовуються базові зовнішні інтерфейси GPIO, UART, I2C та інші, необхідні для подальшої взаємодії з периферійними пристроями. Після цього запускається FreeRTOS, яка керує виконанням задач (потоків), організацією планування, обробкою подій та управлінням ресурсами мікроконтролера.

FreeRTOS виконує налаштування периферійних модулів, драйверів та мережевих компонентів, конфігурування портів GPIO як входи або виходи згідно до призначення, активацію UART для виведення діагностичних повідомлень та ініціалізацію інтерфейсів SPI та I2C для взаємодії з підключеними сенсорами, дисплеями та іншими зовнішніми пристроями.

Інтерфейс SPI працює за принципом master-slave, де мікроконтролер (master) керує передачею даних до підключених пристроїв (slaves). Для обміну використовується кілька ліній: одна передає дані (MOSI), інша подає тактові сигнали (SCK або SCLK), а також використовується пін вибору пристрою (CS). Коли мікроконтролер активує CS, обраний пристрій починає приймати дані по MOSI. Тактові імпульси SCK чи SCLK синхронізують обмін, передаючи кожен біт у заданому ритмі. У деяких випадках використовується зворотна лінія (MISO), щоб пристрій міг надсилати дані назад до мікроконтролера (рисунок 2.2) [15].

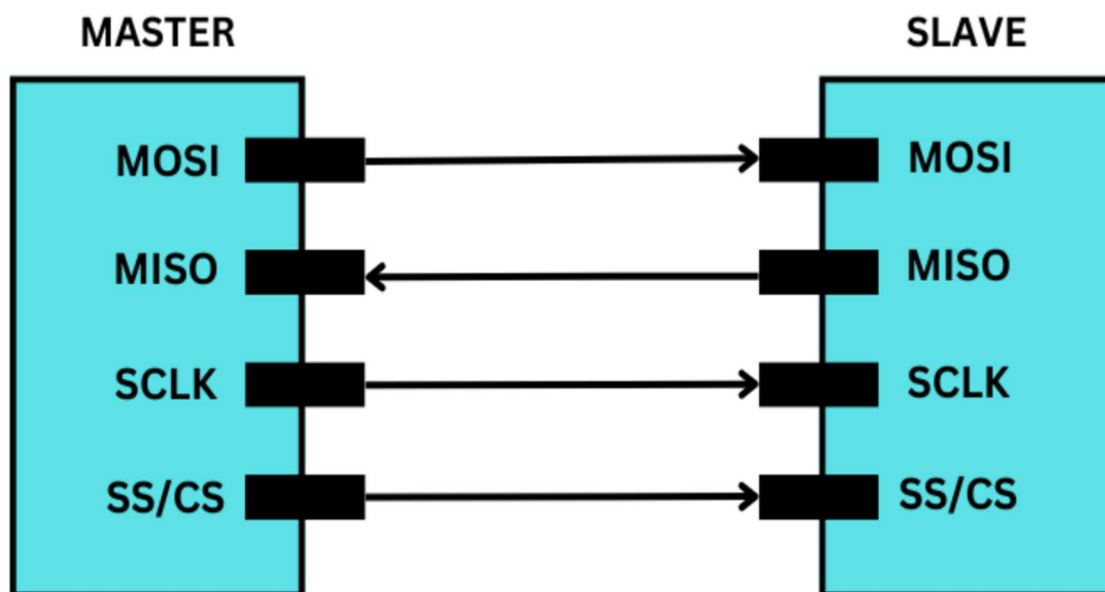


Рисунок 2.2 – Схема роботи SPI

В свою чергу протокол I2C на відміну від SPI використовує лише два дроти — один для даних, інший для тактового сигналу. Він дозволяє підключити багато пристроїв на одну спільну шину, кожен з яких має свою унікальну адресу. SPI натомість потребує більше дротів (мінімум чотири) і окремого керуючого піну (CS) для кожного пристрою, але забезпечує значно вищу швидкість передачі даних. Тому I2C зручніший для простих і малошвидкісних сенсорів, де важлива економія пінів, а SPI краще підходить для дисплеїв, карт пам'яті та інших "важких" пристроїв, де потрібна швидкість.

Після запуску FreeRTOS встановлюється мережеве з'єднання через Wi-Fi у режимі клієнта (Station mode) або точки доступу (Access Point mode), при цьому здійснюється отримання IP-адреси через DHCP та активація WPA2-шифрування з передачею сертифікатів. Завершивши початкову конфігурацію, FreeRTOS створює окремі задачі (tasks), кожна з яких відповідає за конкретну функцію. Наприклад: періодичне зчитування даних з сенсора, передавання інформації через MQTT, обробка HTTP-запитів тощо. Створивши задачі, FreeRTOS керує їхнім плануванням шляхом надавання їм відповідних пріоритетів, обсягів пам'яті та часу виконання для забезпечення стабільної та паралельної роботи.

Схожим чином працюють і інші мікроконтролери таких сімейств як STM32, ESP8266 чи Arduino. На відміну від інших, більшість мікроконтролерів ESP32 мають вбудовані модулі Wi-Fi та Bluetooth, двоядерний процесор, апаратне шифрування та більше пам'яті, в результаті чого не потребують зовнішніх модулів для складних задач на відміну від STM32, де потрібні додаткові плати для бездротового зв'язку. Завдяки цьому ESP32 дозволяє створювати компактніші, дешевші та енергоефективніші пристрої з меншими затратами на апаратну частину.

2.2 Опис мікроконтролера ESP32 CH340C 30P Type-C

Під час реалізації проекту було прийнято рішення використати компактну, найпопулярнішу, бюджетну та найдоступнішу плату розробки ESP32 CH340C 30P Type-C (рисунок 2.3).

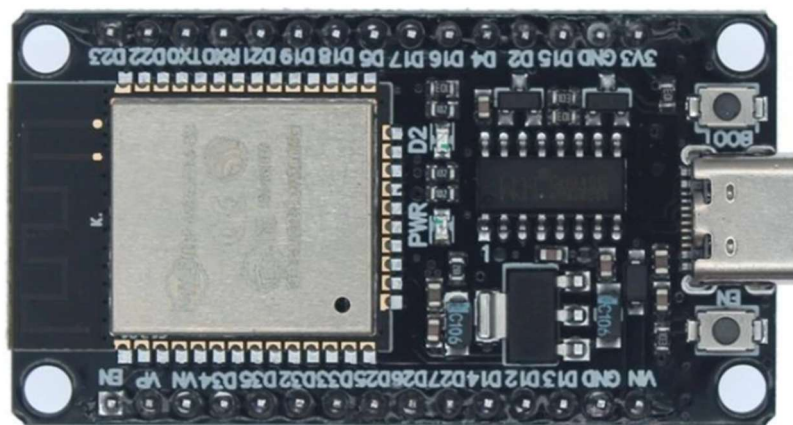


Рисунок 2.3 – Плата, побудована на мікроконтролері EPS32

Особливістю цієї плати є використання вбудованого чипу CH340C, який забезпечує просте підключення до ПК без зовнішніх програматорів та виконує функції USB-UART перетворювача. Плата використовує ESP32-WROOM-32, який є популярним універсальним бездротовим модулем на базі чипа ESP32

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		24

(Espressif) з вбудованим Wi-Fi та Bluetooth та обладнаний двома 32-бітними ядрами Xtensa LX6, які можуть працювати паралельно на частоті до 240 МГц та мають архітектуру з однорівневим кешем, підтримкою переривань, енергозберігаючими режимами, а також високою швидкістю доступу до внутрішньої SRAM-пам'яті [13].

Мікроконтролер оснащено 448 Кб ROM для зберігання завантажувача, 520 Кб оперативної пам'яті SRAM для виконання коду та обробки даних, 8 Кб швидкої та 8 Кб повільної пам'яті RTC SRAM для використання в режимі енергозбереження. Також є 1 Кб пам'яті eFuse, з яких 256 біт призначені для зберігання MAC-адреси та параметрів захисту, 4 Мб SPI Flash для постійного зберігання програм і даних з можливістю розширення до 16 Мб [13].

На платі доступно 30 виводів GPIO, розведених по 15 з кожного боку, кнопки EN (скидання) і BOOT (режим прошивки), а також індикатор живлення (LED). Форм-фактор сумісний із макетними платами: розміри — приблизно 51.5 x 28.5 мм, крок виводів — 2.54 мм. Плата підтримує режими енергозбереження Light-sleep і Deep-sleep, при цьому у найекономнішому режимі споживання знижується до <5 мкА.

Присутня можливість оновлення прошивки по Wi-Fi та підключення зовнішньої антени замість PCB-антени, підтримка Flash Encryption та Secure Boot.

Іншими додатковими інтерфейсами та бездротовими можливостями модуля є [13]:

- 3 послідовних UART порти;
- 4 SPI-інтерфейси для високошвидкісного обміну даними;
- 2 I2C-інтерфейси для підключення сенсорів та контролерів;
- I2S інтерфейс для обробки аудіо сигналів;
- CAN-шина для промислових застосувань;
- SDIO для роботи з картами пам'яті;
- Інфрачервоний передавач і приймач (IR TX/RX);

- 16 каналів PWM для керування двигунами чи яскравістю світлодіодів;
- 18 каналів 12-бітного АЦП для зчитування аналогових сигналів;
- 2 канали ЦАП на GPIO25 та GPIO26;
- 10 ємнісних сенсорних каналів;
- Вбудований температурний сенсор;
- Вбудований сенсор Холла (магнітного поля);
- Підтримка Wi-Fi 802.11 b/g/n (2.4 ГГц) зі швидкістю до 150 Мбіт/с;
- Підтримка Bluetooth 4.2 (Classic BR/EDR + BLE);
- Апаратне шифрування: AES, RSA, SHA;
- Захист бездротового з'єднання: WPA/WPA2, TLS 1.2.

2.3 Опис датчика GY-30 BH1750FVI

GY-30 – це модуль на базі цифрового датчика освітленості BH1750FVI (рисунок 2.4), який призначений для вимірювання рівня освітленості. Сенсор орієнтований на видимий спектр. Завдяки цьому він підходить для вимірювання освітлення в побутових та природних умовах, де джерелом освітлення є сонячне світло та штучні лампи з «теплим» спектром світла, видає результат у люксах. Має базовий ІЧ-фільтр, який не повністю блокує випромінювання поза межами видимого спектра. І тому використання у середовищі з високим рівнем ІЧ- або УФ- випромінюванням не рекомендується, оскільки цей сенсор може видавати неточні або спотворені результати. На відміну від аналогових фоторезисторів, цей сенсор одразу забезпечує цифровий вихід через інтерфейс I2C, завдяки чому усуваються похибки, пов'язані з аналоговими перетвореннями на стороні контролера, спрощується підключення та підвищується точність вимірювань. GY-30 BH1750FVI використовується в системах «розумний дім», побутовій техніці, вуличному освітленні тощо.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						26
Зм.	Арк	№ докум.	Підпис	Дата		

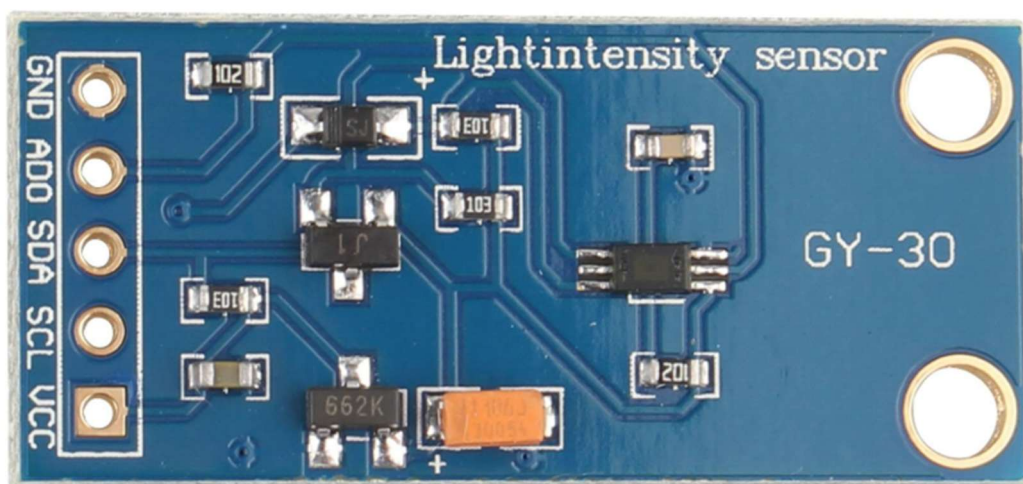


Рисунок 2.4 – Датчик освітленості GY-30 BH1750FVI

Сенсор працює наступним чином: під час вимірювання фотодіод створює фотострум, пропорційний кількості світла, що на нього потрапляє. Який подається на інтегруючий операційний підсилювач (AAIP), де накопичується протягом певного часу. Накопичений сигнал оцифровується 16-бітним АЦП, в результаті чого формується цифрове значення, що відповідає рівню освітленості. Під час вимірювання BH1750FVI накопичує сигнал протягом певного фіксованого часу. Вимірювання можна здійснювати в кількох режимах: High Resolution (час 120 мілісекунд, точність до 1 лк), High Resolution 2 (час 120 мілісекунд, точність 0.5 лк) або Low Resolution (час 16 мілісекунд, але з кроком 4 лк). Сенсор усереднює стрибки коливань освітленості та ефективно відфільтровує шум, зумовлений мерехтінням ламп. Отримане цифрове значення зберігається у внутрішньому регістрі і готове для зчитування.

Даний датчик має наступні технічні характеристики [16]:

- Напруга живлення: 3.3–5 В (вбудований стабілізатор: чіп працює на 2.4–3.6 В);
- Інтерфейс: I2C (100 або 400 кГц), адреса: 0x23 або 0x5C;
- Діапазон вимірювання: ~1–65 535 лк (до 100 000 як при зміні

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		27

чутливості);

- Роздільна здатність: 16-бітний АЦП;
- Точність: $\pm 20\%$ (калібрований виробником);
- Спектральна чутливість: близька до людського ока (~ 560 нм);
- Фільтрація мерехтіння: усереднення сигналу, ігнорує 50/60 Гц;
- Споживання струму: ~ 120 μA в активному режимі, ~ 0.01 μA в режимі сну;

- Цифровий вихід: безпосередньо у люксах, 16-біт;
- Робочий діапазон температур: $-40\dots+85$ $^{\circ}\text{C}$.

2.4 Опис датчика TCS34725

TCS34725 — це цифровий датчик кольору температури світла (рисунок 2.5), здатний вимірювати інтенсивність червоного, зеленого, синього та загального (прозорого) світла. Завдяки вбудованому інфрачервоному фільтру, сенсор забезпечує точне визначення кольорів, наближене до сприйняття людського ока [17].

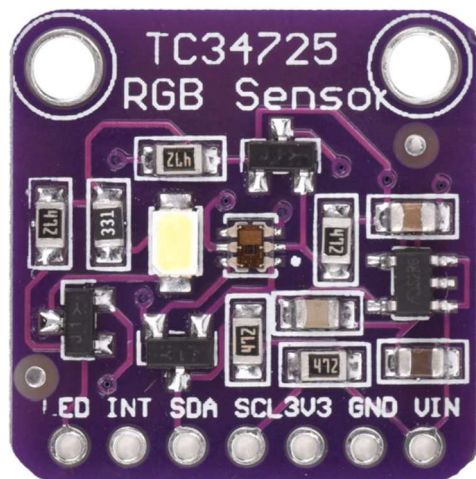


Рисунок 2.5 – датчик кольору TCS34725

Принцип роботи датчика TCS34725 базується на використанні масиву

					ІАЛЦ. 045420.004 ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		28

фотодіодів: К, Г, В, С. Для визначення кольорових компонентів використовуються три типи фотодіодів, що реагують на червоне R (Red), зелене G (Green) та синє B (Blue) світло. Останній тип, С (Clear), пропускає весь спектр світла без кольорової фільтрації і фіксує загальну інтенсивність. Світло, що потрапляє на поверхню сенсора, проходить через вбудований інфрачервоний фільтр, який відсікає ІЧ-випромінювання для уникнення спотворення вимірювань і забезпечення більш точного визначення кольору подібно до того, як це робить людське око. Кожен фотодіод генерує струм, пропорційний інтенсивності світла в певному спектральному діапазоні. Далі сигнал підсилюється і оцифровується 16-бітним АЦП окремо для кожного з чотирьох каналів. Під час оцифровування використовується програмована інтеграція світла — сенсор накопичує світлову інформацію протягом заданого часу (від 2.4 до 614 мс), що забезпечує точність навіть при слабкому освітленні і дозволяє зменшити вплив мерехтіння джерел світла.

Оцифровані значення зберігаються у внутрішніх регістрах і доступні для зчитування через I2C-інтерфейс. За бажанням, ці значення можна використовувати для обчислення кольору, колірної температури (в Кельвінах) або освітленості (в люксах), що особливо зручно у системах адаптивного освітлення або розпізнавання кольору.

Даний датчик має наступні технічні характеристики [17]:

- Тип сенсора: цифровий RGB + Clear датчик кольору;
- Спектральні канали: Red $\approx$ 615 нм, Green $\approx$ 525 нм, Blue $\approx$ 465 нм, Clear — загальна освітленість;
- Розрядність АЦП: 16 біт на кожен канал;
- Динамічний діапазон: до 3 800 000 : 1;
- ІЧ-фільтр: вбудований (блокує інфрачервоне світло);
- Інтерфейс: I2C, швидкість до 400 кГц;
- I2C-адреса: фіксована — 0x29.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						29
Зм.	Арк	№ докум.	Підпис	Дата		

2.5 Опис дисплея 1.8" TFT IPS ST7789

З метою виводу найважливішої частини даних було прийнято рішення використати невеликий за розмірами 1.8-дюймовий TFT-дисплей на контролері ST7789 із роздільною здатністю 128x160 пікселів. Дисплей має невеликі розміри (рисунок 2.6), низьке енергоспоживання та сумісність з мікроконтролерами ESP32.

Принцип роботи дисплея з вбудованим контролером та драйвером ST7789 полягає в керуванні кожним пікселем екрана за допомогою вбудованих транзисторів (TFT), які змінюють прозорість рідкокристалічного шару для проходження світла з підсвітки.

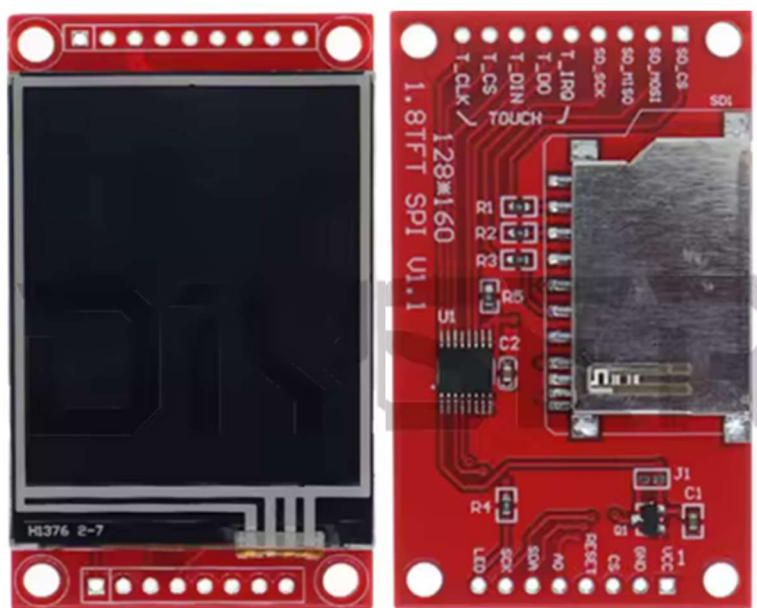


Рисунок 2.6 – 1.8" IPS дисплей TFT ST7789

Дані про колір кожного пікселя надсилаються мікроконтролером по SPI-інтерфейсу, що дозволяє швидко оновлювати зображення. Ці дані надсилаються у внутрішню пам'ять дисплея (Graphic RAM) та самостійно оброблюються контролером ST7789, де кожному пікселю відповідає 16-бітне значення кольору (5 біт червоного, 6 біт зеленого, 5 біт синього). Після цього дисплей оновлює зображення, регулюючи, які пікселі повинні пропускати

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		30

червоне, зелене чи синє світло, а IPS-структура забезпечує стабільну передачу кольору незалежно від кута огляду. Дисплей оснащений сенсорним екраном в комплекті зі стилусом для кращої взаємодії. Модель цього дисплея має вбудований картридер з SPI-сумісним слотом для microSD-карт, який дозволяє мікроконтролеру читати та зберігати дані безпосередньо з карти пам'яті.

Також дисплей має наступні технічні характеристики [17]:

- Глибина кольору: до 16 біт (65 536 кольорів, формат RGB565)
- Інтерфейс підключення: SPI (4-провідний)
- Живлення: 3.3 В логіка (або 5 В — якщо на платі є стабілізатор)
- Типове споживання: ~30-50 мА (разом із підсвіткою)
- Сумісність: Arduino, ESP32, STM32, Raspberry Pi
- Контролер: ST7789
- Підсвітка: білий світлодіод (LED), керується окремим піном
- Час відгуку: приблизно 10–20 мс
- Кути огляду: ~160 градусів (завдяки IPS)

2.6 Проектування апаратної частини

Для того, щоб зібрати пристрій для подальшого використання, треба коректно з'єднати між собою піни компонентів.

Підключення датчику освітленості GY-30 BH1750FVI до мікроконтролеру ESP32 CH340C TYPE-C реалізовано так, як показано в таблиці 2.1.

Таблиця 2.1 – Відповідність пінів ESP32 та GY-30 BH1750FVI

Пін ESP32	Пін GY-30
3V3	VCC
GND	GND
GND	ADD

Продовження таблиці 2.1

Пін ESP32	Пін GY-30
GPIO21	SDA
GPIO22	SCL

Пін 3V3 мікроконтролера ESP32 підключено до VCC модуля GY-30, який живить внутрішній датчик BH1750FVI. Оскільки ESP32 працює з логікою 3.3 В, подача такого робочого живлення є оптимальною для забезпечення коректної роботи GY-30, оскільки немає потреби використовувати стабілізатор напруги.

Пін GND ESP32 з'єднано з GND GY-30 для утворення спільної землі (нульового потенціалу) в електричному колі. Це критично важливо для того, щоб сигнали даних між пристроями мали спільний відлік напруги, інакше обмін не буде працювати або може спричинити пошкодження.

Окремо пін ADD модуля GY-30 також підключено до GND. Цей вивід використовується для встановлення I2C-адреси сенсора BH1750FVI. Якщо ADD підключено до GND — використовується адреса 0x23, яка є типовою у більшості програмних бібліотек. Якщо ж підключити ADD до VCC — адреса змінюється на 0x5C. Тому з'єднання ADD з GND дозволяє уникнути конфліктів і не вимагає додаткової конфігурації в коді.

Пін GPIO21 ESP32 підключено до SDA на GY-30 — це лінія передачі даних протоколу I2C. GPIO21 є стандартним піном для SDA в ESP32, і саме по цьому каналу датчик передає виміряне значення освітленості у вигляді цифрового сигналу.

Пін GPIO22 ESP32 підключено до SCL на GY-30 — це лінія тактової синхронізації для I2C. GPIO22 є типовим піном SCL на ESP32, і по ньому мікроконтролер надсилає тактові імпульси, які синхронізують обмін інформацією між ним і датчиком BH1750FVI.

Підключення датчику кольору температури світла TCS34725 до мікроконтролера ESP32 CH340C TYPE-C реалізовано так, як показано в

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						32
Зм.	Арк	№ докум.	Підпис	Дата		

таблиці 2.2.

Таблиця 2.2 – Відповідність пінів ESP32 та TCS34725

Пін ESP32	TC34725
3V3	3V3
GND	GND
GPIO21	SDA
GPIO22	SCL

Пін 3V3 на платі ESP32 підключено до 3V3 на модулі TCS34725. 3.3 В є робочою напругою для датчика TCS34725. Також TCS34725 підтримує можливість живлення від 5 В через пін VIN, оскільки має стабілізатор напруги.

Пін GND ESP32 з'єднано з GND модуля TCS34725 для створення спільного нульового потенціалу, який необхідний для функціонування протоколу I2C.

Пін GPIO21 на ESP32 підключено до SDA на TCS34725. Цей пін відповідає за передачу даних у рамках шини I2C. Через GPIO21 мікроконтролер отримує значення з чотирьох каналів.

Пін GPIO22 ESP32 підключено до SCL на TCS34725. Цей пін є лінією синхронізації для передачі даних у I2C. GPIO22 на ESP32 використовується як стандартний пін SCL для шини I2C і відповідає за формування тактових імпульсів, які синхронізують обмін даними з датчиком TCS34725.

Підключення дисплею 1.8" TFT IPS ST7789 до мікроконтролеру ESP32 CH340C TYPE-C реалізовано так, як показано в таблиці 2.3.

Таблиця 2.3 – Відповідність пінів ESP32 та 1.8" TFT IPS ST7789

Пін ESP32	Пін дисплея
3V3	LED
GPIO18	SCK
GPIO23	SDA

Продовження таблиці 2.3

Пін ESP32	Пін дисплея
GPIO2	A0 (DC)
GPIO4	RESET
GPIO15	CS
GND	GND
3V3	VCC

Пін 3V3 на ESP32 підключено до пину VCC дисплея. Логіка дисплея 3.3 В, тобто на цій напрузі працюють всі сигнали (SCK, SDA, RESET, A0/DC, CS), але на пін VCC можна підключати не тільки 3.3 В, а ще й 5 В, бо модуль має вбудований стабілізатор живлення. Враховуючи логіку ESP32, обидва компоненти мають пряму сумісність.

Пін 3V3 ESP32 також підключено до пину LED на дисплеї. Цей пін відповідає за живлення білого світлодіоду підсвітки екрану. На пін LED можна подавати як 3.3 В, так і 5 В. За бажанням яскравість підсвітки можна регулювати, підключивши LED до GPIO ESP32 з PWM (широотно-імпульсною модуляцією), що дозволить її регулювання програмним шляхом. Також можна знизити значення струму, використавши резистор.

Пін GND ESP32 з'єднано з піном GND дисплея. Це обов'язкове з'єднання, бо без спільного GND сигнали можуть бути спотвореними або нестабільними. Відсутність спільного GND повністю унеможливить роботу дисплея, навіть якщо решта сигнальних пінів підключено правильно.

Пін GPIO18 ESP32 з'єднано з піном SCK дисплея. Це лінія SPI-тактів (Serial Clock), яка визначає ритм обміну даними. GPIO18 є стандартним апаратним SPI-піном для SCK на ESP32, що дозволяє використовувати апаратне прискорення та стабільну передачу.

Пін GPIO23 ESP32 підключено до пину SDA на дисплеї, який у SPI-протоколі є лінією MOSI. Через цей пін ESP32 надсилає байти інформації, які можуть бути командами керування або даними пікселів. GPIO23 — типовий

SPI-пін MOSI на ESP32, і використання його відповідає стандартній схемі.

Пін GPIO2 ESP32 підключено до піну AO (або DC) дисплея. Цей пін визначає тип переданого байта: чи інформація, що надходить через SPI, є командою або графічними даними. GPIO2 обраний як вільний керуючий пін. У коді ЕІН налаштовується як DC-пін і відповідає за перемикання між режимами.

Пін GPIO4 ESP32 з'єднано з піном RESET дисплея. Він використовується для апаратного скидання ST7789 перед початком ініціалізації. Це допомагає уникнути “зависання” або неправильного запуску дисплея. GPIO4 — звичайний вільний пін, який конфігурується в бібліотеці як Reset.

Пін GPIO15 ESP32 підключено до піну CS дисплея. Цей пін активує або деактивує пристрій на SPI-шині. При наявності кількох SPI-пристроїв, CS дозволяє ESP32 звертатися лише до обраного пристрою. GPIO15 тут використовується як керуючий пін вибору дисплея.

Схему з'єднання всіх компонентів показано в додатку А.

2.7 Опис та обґрунтування вибору мови програмування C

Сімейство мікроконтролерів ESP32 підтримує низку поширених мов програмування, серед яких найвживанішими є C, C++ та MicroPython. Кожна мова має свої переваги та недоліки, які визначають її доцільність залежно від вимог до реалізації проекту [18].

Мова програмування C є однією з найстаріших та найпоширеніших мов програмування у сфері розробки для мікроконтролерів. Її головна перевага — максимальна продуктивність. Оскільки C — це компільована мова, програми на ній виконуються безпосередньо на апаратному рівні, що забезпечує високу швидкість роботи та ефективне використання ресурсів. Це особливо важливо для задач, де критичні час відгуку та обмежені ресурси. Крім того, мова C надає повний контроль над апаратною частиною, що дозволяє оптимізувати

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						35
Зм.	Арк	№ докум.	Підпис	Дата		

програми для конкретних потреб. Проте, С має і свої недоліки. Вона вимагає глибоких знань у галузі програмування та розуміння апаратної архітектури. Управління пам'яттю в С здійснюється вручну, що може призвести до помилок, таких як витоки пам'яті або переповнення буфера. Також, відсутність об'єктно-орієнтованих конструкцій ускладнює розробку великих проектів.

Мова програмування С++ характеризується високою складністю внаслідок великої кількості мовних конструкцій і концепцій, зокрема шаблонів, багаторівневої та множинної спадковості, перевантаження операторів, механізмів обробки виключень тощо. Особливої уваги заслуговує підтримка множинного наслідування, яка, хоч і забезпечує додаткову гнучкість у проектуванні, водночас створює потенційні конфлікти під час спадкування методів та атрибутів. Це ускладнює підтримку проектів і порушує передбачуваність реалізації шаблонів проектування (патернів), зокрема в ситуаціях, коли виникає так звана «проблема діаманта». У вбудованих системах, зокрема на мікроконтролерах ESP32, використання складних мовних засобів С++, таких як STL, може суттєво збільшити розмір прошивки. В умовах обмеження апаратних ресурсів, зокрема на мікроконтролерах ESP32, це стає серйозною проблемою, оскільки надмірний обсяг коду може не вміститись у доступну пам'ять мікроконтролера або вплинути на стабільність роботи пристрою. Крім того, використання динамічного керування пам'яттю без застосування додаткових технік, таких як RAII чи smart pointers, може призвести до фрагментації пам'яті, витоків і нестабільної роботи системи. Окрему складність становить мовна еволюція С++, яка є результатом поступового об'єднання концепцій з кількох мов програмування (С, Ada, Simula, JavaScript та ін.). Кожен новий стандарт мови (С++98, С++11, С++14, С++17, С++20) додає нові конструкції, що іноді змінюють трактування наявних підходів, а відтак ускладнюють підтримку коду й створюють плутанину щодо застосування тих чи інших ідіом. Відсутність усталеного й універсального стилю програмування на С++ у сфері вбудованих систем

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						36
Зм.	Арк	№ докум.	Підпис	Дата		

призводить до несумісності між проектами та додаткових витрат на адаптацію коду. Хоча застосування C++ у проектах на базі ESP32 є технічно можливим, воно потребує значних зусиль для забезпечення стабільності, продуктивності та сумісності системи.

MicroPython — це легка версія Python, адаптована для мікроконтролерів, включаючи ESP32. Її головна перевага — простота та швидкість розробки. Завдяки інтерпретованому характеру, розробники можуть швидко писати та тестувати код без необхідності компіляції. Це особливо корисно для прототипування та навчання. Крім того, MicroPython має інтерактивну консоль (REPL), що дозволяє виконувати команди в реальному часі. Проте, MicroPython має обмеження. Оскільки це інтерпретована мова, програми на MicroPython виконуються повільніше порівняно з C або C++. Це може бути критично для задач, де важлива швидкодія. Крім того, MicroPython споживає більше пам'яті, що може бути проблемою для ресурсно обмежених систем. Також, не всі стандартні бібліотеки та функціональність доступні в MicroPython, що обмежує його застосування в деяких проектах.

Вибір мови C для розробки на ESP32 обґрунтований її високою продуктивністю та ефективністю. Програми на C компілюються в машинний код, що дозволяє максимально ефективно використовувати ресурси мікроконтролера. Це особливо важливо для задач, де критичні час відгуку та обмежені ресурси. Крім того, C надає повний контроль над апаратною частиною, що дозволяє оптимізувати програми для конкретних потреб. Хоча C вимагає глибших знань та уважності при розробці в порівнянні з MicroPython, її переваги в продуктивності та контролі роблять її ідеальним вибором для серйозних проектів на ESP32.

2.8 Опис та обґрунтування потреби проекту в ШІ

У даному проекті основою для прийняття рішень виступають дані, отримані з датчиків освітлення та кольору температури світла, підключених до

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						37
Зм.	Арк	№ докум.	Підпис	Дата		

мікроконтролера ESP32. Дані надходять у режимі реального часу та постійно змінюються залежно від рівня освітленості. Для коректної обробки таких змінних і часто непередбачуваних сигналів потрібен гнучкий механізм, здатний аналізувати динаміку, виявляти патерни або реагувати на нестандартні ситуації — саме цю роль може виконувати алгоритм машинного навчання.

Якщо треба розширити функціональність проекту, систему також можна доповнити підключенням смарткамери, мікрофону або PIR-датчика. Обробка звуку або зображення вручну (тобто написанням умовних операторів і порогів) майже неможлива або вкрай неефективна без ШІ. Наприклад, якщо потрібно виявляти присутність людини за голосом або жестами — це завдання вже виходить за межі простого кодування й потребує застосування методів комп'ютерного зору або розпізнавання звуку. У таких випадках використання нейронних мереж або попередньо натренованих моделей дозволяє впровадити в систему здатність до класифікації подій чи об'єктів.

Ще одним важливим джерелом для прийняття рішень у проекті є лог-файли та історія попередніх подій. Система, що веде запис змін стану пристроїв, дій користувача або реакцій на певні події, накопичує цінний масив даних, який може бути використаний для тренування ШІ-моделей. Наприклад, якщо зафіксувати закономірність у включенні вимкненні освітлення залежно від часу доби, колірної температури світла та рівня освітлення, система зможе з часом навчитися прогнозувати подальші дії та діяти автоматично. Це створює передумови для переходу від реактивної логіки до проактивного функціонування пристрою.

Використання штучного інтелекту у проекті значно підвищує точність прийняття рішень. Замість того, щоб покладатися на жорсткі пороги або прості «if»-умови, ШІ аналізує контекст і обирає найкращий варіант дії на основі накопиченого досвіду чи навчання. Наприклад, замість простого «увімкни світло при освітленні нижче 200 люкс», система зможе враховувати ще й час доби (не вмикати світло вдень, навіть якщо погода хмарна), рух у приміщенні (аби не вмикати світло даремно), попередній стан (чи світло вже було

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						38
Зм.	Арк	№ докум.	Підпис	Дата		

вимкнено вручну), колірну температуру (встановлювати тепле світло ввечері для зменшення навантаження на очі) тощо — і приймати рішення, які набагато ближчі до людського способу мислення. Це особливо важливо в середовищах, де вхідні дані нестабільні або суперечливі.

Використання штучного інтелекту дозволяє системі працювати автономно та адаптуватися до змін без участі користувача, що значно підвищує її ефективність та зручність у використанні. Завдяки здатності до самонавчання пристрій може самостійно регулювати свою поведінку — наприклад передбачати пікові навантаження на електромережу та коригувати роботу пристроїв для зменшення споживання, враховувати особливості приміщення (розташування вікон, відбиття світла) з метою створення оптимального освітлення — тим самим реагуючи на реальні умови, а не лише на заздалегідь прописані сценарії. Це дає змогу системі ефективно пристосовуватись до нових ситуацій, зберігаючи стабільність роботи навіть тоді, коли початкові умови змінюються або взагалі не були передбачені під час розробки.

Якщо не використовувати ШІ, розробнику доводиться реалізовувати логіку у вигляді жорстких сценаріїв з фіксованими умовами, що часто призводить до громіздкого, складного коду, який важко підтримувати. Зі зростанням кількості змінних і взаємозв'язків між ними така система стає ненадійною і непередбачуваною. Впровадження ШІ дозволяє скоротити кількість умов і винятків, делегуючи адаптивну логіку моделі, яка самостійно навчається на основі реальних ситуацій. В результаті система буде працювати ефективніше, точніше реагувати та не потребувати частого втручання в код.

Зрештою, без ШІ пристрій був би лише автоматизованим виконавцем команд, а не самостійною, розумною системою. Він не міг би прогнозувати події, виявляти аномалії, формувати поведінку залежно від ситуації. Замість «розумного пристрою» це була б проста схема, яка робить лише те, що в неї закладено. Саме тому впровадження ШІ в проект не просто виправдане, а — необхідне для переходу на новий рісень автономності та гнучкої взаємодії з

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						39
Зм.	Арк	№ докум.	Підпис	Дата		

оточенням.

2.9 Опис інструментів розробки

Для розробки будь-якого проєкту необхідно використовувати спеціальний набір програмного забезпечення, який називається тулчейном. Його призначення полягає в забезпеченні всіх необхідних інструментів для написання, відлагодження та тестування коду. Такий набір повинен відповідати технічним вимогам і покривати всі етапи розробки продукту — від початкових етапів до фінального запуску.

У процесі реалізації проєкту було прийнято рішення використовувати редактор коду Visual Studio Code. Це одне з найпопулярніших середовищ для розробників, що поєднує простоту інтерфейсу з широкими можливостями для розширень. Завдяки підтримці плагінів, VS Code легко можна адаптувати під будь-які потреби — від веб-розробки до низькорівневого програмування мікроконтролерів. Для роботи з мікроконтролером ESP32 використовується офіційне розширення ESP-IDF Extension, яке інтегрує повний стек сучасних інструментів розробки від компанії Espressif прямо в середовище редактора VS Code.

Крім Visual Studio Code, можна використовувати й інші інструменти розробки такі як Sublime Text 3, Arduino IDE, Eclipse IDE тощо.

Arduino IDE орієнтована на простоту й зручний дизайн, однак це середовище використовує спрощену модель програмування (наприклад, єдиний цикл `loop()`), не має можливості встановлення ESP-IDF та гнучкої підтримки багатозадачності і лише частково відкриває доступ до низькорівневих функцій чипа мікроконтролера.

Eclipse IDE є важчим і складнішим у налаштуванні середовищем, яке споживає більше системних ресурсів і має менш інтуїтивний інтерфейс порівняно з легким і гнучким VS Code. Для роботи Eclipse IDE з ESP-IDF потрібні додаткові плагіни та ручне конфігурування середовища розробки, що

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						40
Зм.	Арк	№ докум.	Підпис	Дата		

ускладнює початкові налаштування для недосвідчених розробників.

Sublime Text 3 — це редактор коду, що має низькі системні вимоги та може бути альтернативою Visual Studio Code, але поступається йому за функціональністю. Він також потребує встановлення додаткових плагінів та ручного налаштування, що ускладнює процес збірки, прошивки, налагодження та конфігурації проектів.

ESP-IDF (Espressif IoT Development Framework) — це офіційний фреймворк для програмування пристроїв на базі мікроконтролерів ESP32, ESP32-S3, ESP32-C3 та інших. Він побудований на основі CMake, напряду підтримує FreeRTOS та дозволяє розробляти як прості, так і комерційні складні IoT-рішення. Під час розробки програми, ESP-IDF використовується для створення прошивки, налаштування периферійних інтерфейсів, організації роботи із сенсорами, реалізації логіки обробки даних та передачі інформації через бездротові інтерфейси (Wi-Fi, Bluetooth). За допомогою цього фреймворку реалізовано повноцінну підтримку багатозадачності, управління пам'яттю, взаємодію з апаратним забезпеченням, а також використано вбудовані API для підключення до мережі, роботи з файловою системою та реалізації веб-сервера. ESP-IDF активно підтримується для забезпечення кращої інтеграції з сучасними інструментами та новими мікроконтролерами ESP32.

Після встановлення ESP-IDF у VS Code, користувач отримує доступ до всіх необхідних функцій: автоматичного налаштування середовища, запуску проектів, меню-конфігуратора параметрів (menuconfig), збирання прошивки, запису у флеш-пам'яць мікроконтролера, моніторингу UART-порту, а також налагодження з використанням GDB та трасування. Це суттєво пришвидшує процес розробки, дозволяє уникати помилок і забезпечує зручність при роботі з багатьма конфігураційними файлами, драйверами й компонентами. Крім того, VS Code надає зручний перегляд структури проекту, підсвічування синтаксису, перевірку коду в реальному часі, автодоповнення, інтеграцію з системами контролю версій та інші корисні функції, які роблять розробку

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						41
Зм.	Арк	№ докум.	Підпис	Дата		

більш ефективною.

Ще одним важливим інструментом, який активно використовується в цьому проєкті, є Git. Git дозволяє зберігати історію змін у коді, створювати гілки для нових функцій, відслідковувати помилки та легко повертатись до попередніх робочих версій програми. У рамках роботи над програмною частиною, Git використовувався для організації репозиторію проєкту, фіксації змін після кожного важливого етапу розробки, а також для зручного експорту чи передавання проєкту на інші комп'ютери. Завдяки інтеграції Git у VS Code можна виконувати всі основні дії (commit, push, pull, merge) без використання терміналу.

Усі три компоненти — Visual Studio Code, ESP-IDF і Git — створюють зручне та ефективне середовище розробки, яке підходить як для особистих експериментів, так і для професійних комерційних рішень, дозволяють зосередитися на розробці функціоналу пристрою та мінімізують технічні складнощі з інструментами.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						42
Зм.	Арк	№ докум.	Підпис	Дата		

3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ РОЗУМНОГО ОСВІТЛЕННЯ НА ОСНОВІ ESP32

3.1 Налаштування роботи ESP-IDF та структура проєкту

Для коректної роботи з мікроконтролером ESP32 та забезпечення стабільної взаємодії з периферією було здійснено налаштування середовища ESP-IDF. Конфігурація виконувалась через інтегровану систему меню `menuconfig`, яка дозволяє гнучко управляти параметрами прошивки, режимами роботи SPI Flash та іншими параметрами.

У розділі `Serial flasher config` було задано такі параметри:

- Flash SPI mode: QIO – вибрано режим, що дозволяє одночасну передачу даних чотирма лініями, що прискорює обмін із зовнішньою Flash-пам'яттю.
- Flash Sampling Mode: STR Mode – встановлено односторонній режим семплювання даних, який є стабільним для більшості стандартних мікросхем пам'яті.
- Flash SPI speed: 80 MHz – задано підвищену швидкість обміну даними для прискорення завантаження прошивки та зчитування з пам'яті, при цьому мікроконтролер стабільно витримує роботу на цій частоті.
- Flash size: 4 MB – обраний стандартний обсяг вбудованої пам'яті для прошивки, достатній для зберігання коду, бібліотек та ресурсів інтерфейсу.
- Disable download stub: залишено неактивним - завантажувальний stub використовується за замовчуванням для оптимізації передачі даних під час прошивки.

Сам проєкт реалізований в наступній файловій структурі:

- `CmakeLists.txt` – конфігураційний файл збірки проєкту для системи Cmake.
- `gy_30.c` – реалізація драйвера сенсора освітленості люксметра.
- `gy_30.h` – заголовковий файл з визначеннями функцій для люксметра.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						43
Зм.	Арк	№ докум.	Підпис	Дата		

- `i2c_module.c` – функції ініціалізації та обміну по шині I2C.
- `i2c_module.h` – інтерфейс I2C-функцій та структура конфігурації пристрою.
- `idf_component.yml` – метадані компонента для ESP-IDF (назва, залежності, версії).
- `led.c` – функції керування RGBW-світлодіодною стрічкою.
- `led.h` – заголовковий файл для API керування світлодіодами.
- `main.cc` – головний файл програми, координує роботу всіх модулів.
- `neural_network.cc` – логіка обробки інференсу TFLite-моделі штучного інтелекту.
- `neural_network.h` – заголовок інтерфейсу до модуля нейронної мережі.
- `real_time.c` – обробка часу доби, синхронізація через NTP.
- `real_time.h` – заголовок функцій для отримання й обробки часу.
- `st7789.c` – драйвер дисплея st7789 для виводу тексту.
- `st7789.h` – оголошення функцій і констант для дисплея ST7789.
- `tcs34725.c` – реалізація драйвера сенсора кольору TCS34725.
- `tcs34725.h` – інтерфейс функцій зчитування кольорових даних.
- `wifi_sta.c` – ініціалізація Wi-Fi у режимі станції, підключення до мережі
- `wifi_sta.h` – оголошення функцій підключення до Wi-Fi.
- `natural_light_model.h` – заголовочний файл, що містить байтовий масив моделі TFLite у форматі C.

Структурну схему взаємодії модулів підпрограм показано в додатку В.

3.2 Програмна реалізація мікроконтролера ESP32 CH340C TYPE-C

Мікроконтролер ESP32 реалізовано як центральний керуючий вузол, який відповідає за збір та керування периферійними пристроями.

Централізований збір даних здійснюється з двох сенсорів: GY-30 BH1750FVI, який вимірює рівень освітленості, та TCS34725, що аналізує

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						44
Зм.	Арк	№ докум.	Підпис	Дата		

колірне середовище й обчислює колірну температуру. Обидва сенсори підключені до ESP32 через інтерфейс I2C, для якого у коді реалізовано окремий модуль `i2c_module.c`. У функції `I2C_Bus_Init()` відбувається конфігурація шини, після чого кожен сенсор додається до I2C-шини як окремий пристрій із зазначенням адреси, частоти та параметрів зв'язку.

Після збору даних із сенсорів, мікроконтролер відображає їх на дисплеї, підключеному до ESP32 через SPI-шину. Ініціалізація дисплея реалізована в модулі `st7789.c`, де також інтегрована графічна бібліотека LVGL. Завдяки їй користувацький інтерфейс оновлюється в реальному часі: створюються текстові мітки, які відображають актуальні значення освітленості (у люксах) та температури кольору (у Кельвінах). Для оновлення інтерфейсу створено окреме фонове завдання `IvgLtask`, яке працює з таймером бібліотеки LVGL та забезпечує стабільне відображення.

3.3 Програмна реалізація роботи сенсору GY-30 BH1750FVI

Сенсор GY-30 BH1750FVI взаємодіє з мікроконтролером ESP32 реалізований в файлах `gy_30.h` та `gy_30.c`. У заголовочному файлі визначено параметри підключення, адреса пристрою (0x23), увімкнення (POWER_ON), скидання (RESET) та запуск вимірювання в безперервному режимі з високою точністю (CONT_H_MODE). Тоді як в файлі `gy_30.c` реалізовано ініціалізацію I2C-шини, передачу команд вмикання, скидання, запуску вимірювання та зчитування сирих даних та обчислення рівня освітленості в люксах.

Після запуску пристрою викликається функція `BH1750FVI_Init()`, яка налаштовує роботу сенсора. У ній спочатку виконується виклик `BH1750FVI_Add_To_I2C_Bus()`, що додає сенсор на I2C-шину за допомогою функції `i2c_master_bus_add_device(...)`. У цьому процесі встановлюється адреса пристрою, частота обміну, а також використовуються визначені пари пінів SCL (GPIO 22) та SDA (GPIO 21). Одразу після цього мікроконтролер надсилає команду увімкнення через `BH1750FVI_Send_CMD(POWER_ON)`, що

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						45
Зм.	Арк	№ докум.	Підпис	Дата		

переводить сенсор у робочий режим.

Для отримання поточних даних про рівень освітленості використовується функція BH1750FVI_Get_Data(). Вона надсилає команду BH1750FVI_Send_CMD(CONT_H_MODE) для запуску вимірювання у безперервному режимі з високою точністю, після чого зчитує два байти даних і обчислює значення освітленості у люксах. Це значення згодом використовується для аналізу умов навколишнього середовища або відображення на дисплеї. Датчик має шість режимів, описаних в таблиці 3.1.

Таблиця 3.1 – Типи режимів вимірювання BH1750FVI

Команда	Режим	Точність	Час вимірювання
CONT_H_MODE (0x10)	Безперервний	Висока (1 лк)	~120 мс
CONT_H_MODE2 (0x11)	Безперервний	Дуже висока (0.5 лк)	~120 мс
CONT_L_MODE (0x13)	Безперервний	Низька (4 лк)	~16 мс
ONETIME_H_MODE (0x20)	Одноразовий	Висока (1 лк)	~120 мс
ONETIME_H_MODE2 (0x21)	Одноразовий	Дуже висока (0.5 лк)	~120 мс
ONETIME_L_MODE (0x23)	Одноразовий	Низька (4 лк)	~16 мс

Після цього з датчика зчитуються два байти даних за допомогою функції i2c_master_receive(...). Ці байти об'єднуються у одне 16-бітове число за допомогою побітового зсуву ((buffer[0] << 8) | buffer[1]).

На завершення це число ділиться на 1.2 – згідно з документацією це потрібно для перевodu в люкси, тобто реальні значення освітленості.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						46
Зм.	Арк	№ докум.	Підпис	Дата		

Отриманий результат передається далі в основну програму, де виводиться на екран.

3.4 Програмна реалізація роботи сенсору TCS34725

Сенсор TCS34725 реалізовано у файлах tcs34725.c та tcs34725.h. Основна ініціалізація виконується функцією TCS34725_Init(), яка додає сенсор до I2C-шини (TCS34725_Add_To_I2C_Bus()), встановлює час інтеграції, підсилення та активує живлення з автоматичним режимом вимірювання.

Зчитування даних виконується функцією TCS34725_Get_Data(reg), яка надсилає команду на читання вказаного регістра та отримує два байти, що об'єднуються у 16-бітове значення. Функція TCS34725_Get_All_Data() послідовно читає значення всіх чотирьох каналів (Clear, Red, Green, Blue) і зберігає їх у структурі tcs34725_data_t.

На основі отриманих значень RGB виконується обчислення колірної температури у функції TCS34725_Get_Color_Temperature(). Значення RGB перетворюються у XYZ-координати у просторі CIE 1931, після чого розраховуються координати білої точки та співвідношення n. Кінцевий результат - CCT (Correlated Color Temperature) - визначається за формулою МакКамі, що дозволяє отримати значення в кельвінах.

Отримані дані можуть бути використані в основному циклі програми для відображення на екрані, прийняття рішень або автоматичного керування освітленням.

3.5 Програмна реалізація роботи дисплею

За реалізацію логіки дисплея відповідають файли st7789.c та st7789.h. Основна логіка зводиться до двох ключових етапів:

1. Ініціалізація дисплею як пристрою через SPI.
2. Ініціалізація графічної бібліотеки LVGL та зв'язування з

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						47
Зм.	Арк	№ докум.	Підпис	Дата		

дисплеєм.

Процес ініціалізації відбувається наступним чином (рисунок 3.1):

- Виконується базова ініціалізація графічної бібліотеки.
- Створюється дисплейний об'єкт, який логічно відповідає фізичному екрану пристрою. Він визначає, як інтерфейс буде пов'язаний з апаратним модулем.
- Виділяються два буфери оперативної пам'яті для збереження графічної інформації з метою уникнення мерехтіння при оновленні екрану та для плавного виводу зображень та тексту.
- Буфери встановлюються як основні для створеного дисплейного об'єкта. З такою прив'язкою бібліотека керує їхнім вмістом автоматично та забезпечується швидке перемикання між активним та фоновим буфером.
- Увімкнюється згладжування країв об'єктів.
- Визначається формат кольору, який підтримується дисплеєм.
- Задається фактична роздільна здатність екрана та його орієнтація.

```
void ST7789_LVGL_Init()
{
    ESP_LOGI(tag, "LVGL initialization");
    lv_init();

    ESP_LOGI(tag, "Creating display");
    disp = lv_display_create(ST7789_DISPLAY_HEIGHT, ST7789_DISPLAY_WIDTH);

    ESP_LOGI(tag, "Setting buffers");
    const uint16_t disp_bufsize = ST7789_DISPLAY_WIDTH * ST7789_DISPLAY_HEIGHT / 8;
    disp_buf_1 = heap_caps_malloc(disp_bufsize, sizeof(uint16_t), MALLOC_CAP_DMA);
    disp_buf_2 = heap_caps_malloc(disp_bufsize, sizeof(uint16_t), MALLOC_CAP_DMA);

    ESP_LOGI(tag, "Display configuration");
    lv_display_set_antialiasing(disp, true);
    lv_display_set_color_format(disp, LV_COLOR_FORMAT_RGB565);
    lv_display_set_resolution(disp, ST7789_DISPLAY_HEIGHT, ST7789_DISPLAY_WIDTH);
    lv_display_set_rotation(disp, LV_DISPLAY_ROTATION_0);
    lv_display_set_buffers(disp, disp_buf_1, disp_buf_2, disp_bufsize * sizeof(uint16_t), LV_DISPLAY_RENDER_MODE_PARTIAL);
    lv_display_set_flush_cb(disp, my_flush_cb);
    lv_display_set_default(disp);
    //lv_display_set_offset(disp, 1, 2);
}
```

Рисунок 3.1 – Ініціалізація дисплею

Процес ініціалізації графічної бібліотеки відбувається наступним чином (рисунок 3.2):

					ІАЛЦ. 045420.004 ПЗ	Арк.
						48
Зм.	Арк	№ докум.	Підпис	Дата		

- Встановлюється функція, яка відповідатиме за оновлення зображення на екрані. Вона викликається щоразу, коли частина інтерфейсу потребує оновлення.
- В межах цієї функції виводиться графічний блок із буфера на дисплей. Зазвичай це частина екрану, яку потрібно перерисувати. Таким чином оновлення виконується лише там, де це потрібно.
- Після завершення виводу система повідомляє бібліотеку про готовність, що дозволяє уникнути дублювання запитів на оновлення та забезпечує стабільність в роботі інтерфейсу.
- Встановлюється активний дисплей, який буде використовуватись за замовчуванням, щоб працювати з інтерфейсом без постійного вказування конкретного об'єкта.
- Ініціюється базова демонстрація або елемент інтерфейсу
- Запускається механізм таймера, що відповідає за регулярне оновлення. Він слідкує за змінами у графічному інтерфейсі, і в разі потреби викликає оновлення зображення на дисплеї.

```
void ST7789_Init()
{
    ESP_LOGI(tag, "Initialization");
    ESP_ERROR_CHECK(ST7789_SPI_Init());
    ESP_ERROR_CHECK(ST7789_Panel_IO_Handle_Init());
    ESP_ERROR_CHECK(ST7789_Panel_Handle_Init());

    ESP_LOGI(tag, "Driver configuration");
    ESP_ERROR_CHECK(esp_lcd_panel_reset(st7789_panel_handle));
    ESP_ERROR_CHECK(esp_lcd_panel_init(st7789_panel_handle));
    ESP_ERROR_CHECK(esp_lcd_panel_invert_color(st7789_panel_handle, false));
    ESP_ERROR_CHECK(esp_lcd_panel_mirror(st7789_panel_handle, true, false));
    ESP_ERROR_CHECK(esp_lcd_panel_swap_xy(st7789_panel_handle, true));
    ESP_ERROR_CHECK(esp_lcd_panel_set_gap(st7789_panel_handle, 1, 2));
    ESP_ERROR_CHECK(esp_lcd_panel_disp_on_off(st7789_panel_handle, true));
}
```

Рисунок 3.2 – фрагмент коду інтеграції ST7789 з бібліотекою LVGL

						Арк.
						49
Зм.	Арк	№ докум.	Підпис	Дата	ІАЛЦ. 045420.004 ПЗ	

Повна схема роботи дисплея має вигляд, представлений на рисунку 3.3.

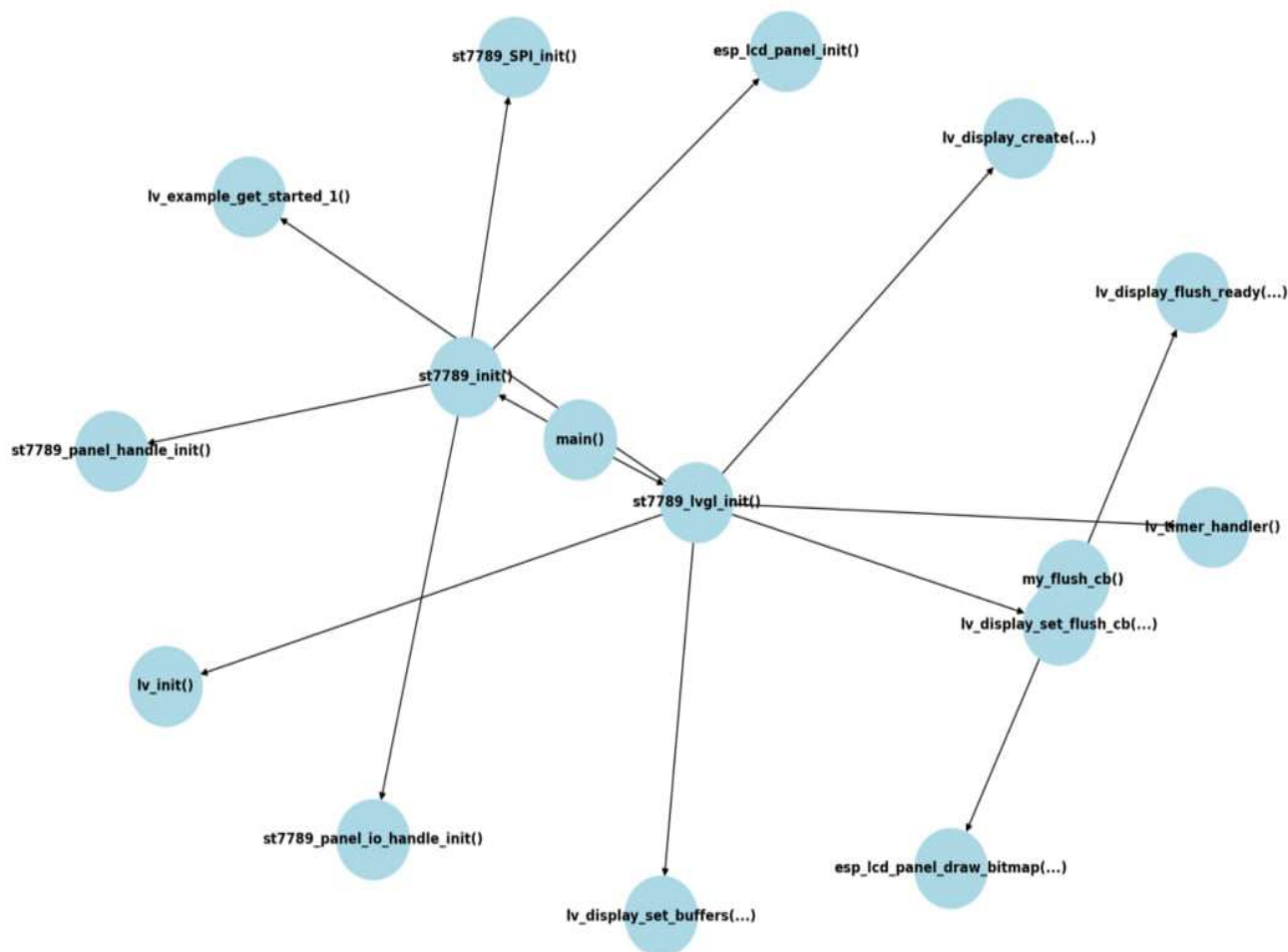


Рисунок 3.3 – Схема викликів функцій під час роботи дисплея

Таким чином, залежність між програмними та апаратними модулями буде виглядати так, як показано в додатку Г.

3.6 Програмна реалізація модулю штучного інтелекту

Для реалізації інтелектуального керування освітленням у системі впроваджено окремий модуль штучного інтелекту на основі нейронної мережі мовою Python. Модель приймає три вхідні параметри:

- Час доби (0-1440 хв).
- Рівень освітленості (0-10000 люкс);
- Поточну кольорову температуру (1000-10000K).

					ІАЛЦ. 045420.004 ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		50

Цільовим виходом є бажана кольорова температура, яка змінюється відповідно до природних умов: теплі тони вранці та ввечері (~3000-4000K), холодні вдень (до 6500K) та приглушено теплі вночі (~2200K).

Синтетичний дата сет генерується функцією `generate_synthetic_data()` з урахуванням добових закономірностей та впливу освітленості. Дані нормалізуються через `Normalization().adapt(X)` для стабільності навчання. Модель має архітектуру з нормалізаційним шаром, двома призованими шарами (16 і 32 нейрони з ReLu) та одним вихідним лінійним шаром. Компіляція виконується за допомогою `model.compile()` з оптимізатором Adam, функцією втрат MSE та метрикою MAE.

Навчання запускається через `model.fit()` з валідаційною вибіркою та callback-ом ранньої зупинки. Результати тренування зберігаються в об'єкті `history` для подальшого аналізу. Після навчання модель конвертується до формату TensorFlow Lite з повним INT8-квантуванням. Для цього використовується `TFLiteConverter.from_keras_model()` із заданням представницького датасету `representative_dataset()`, а також параметрів `converter.optimizations`, `inference_input_type` та `inference_output_type`. Фінальний байтів масив зберігається у .tflite-файл.

Час інференсу не перевищує 10 мс, а розмір моделі після квантування становить близько 3-5 Кб. Модель інтегрується у вбудоване середовище у вигляді C-масиву без залежності від зовнішніх бібліотек, що забезпечує автономну, швидкодіючу та стійку до збоїв систему, яка може бути легко перенавчена аб адаптована під нові умови без зміни базової структури.

Враховуючи всі вище наведені дані, блок-схема алгоритму функціонування системи виглядатиме так, як показано в додатку Б.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						51
Зм.	Арк	№ докум.	Підпис	Дата		

4. ОЦІНКА РОБОТИ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1 Загальний принцип роботи та характеристика

Розроблена система є мікроконтролерним пристроєм, призначеним для вимірювання параметрів освітлення у приміщенні та адаптації штучного освітлення до природнього. В результаті реалізації розроблена система функціонує у повністю автономному режимі з урахуванням енергозбереження та мінімального втручання користувача. (рисунок 4.1).

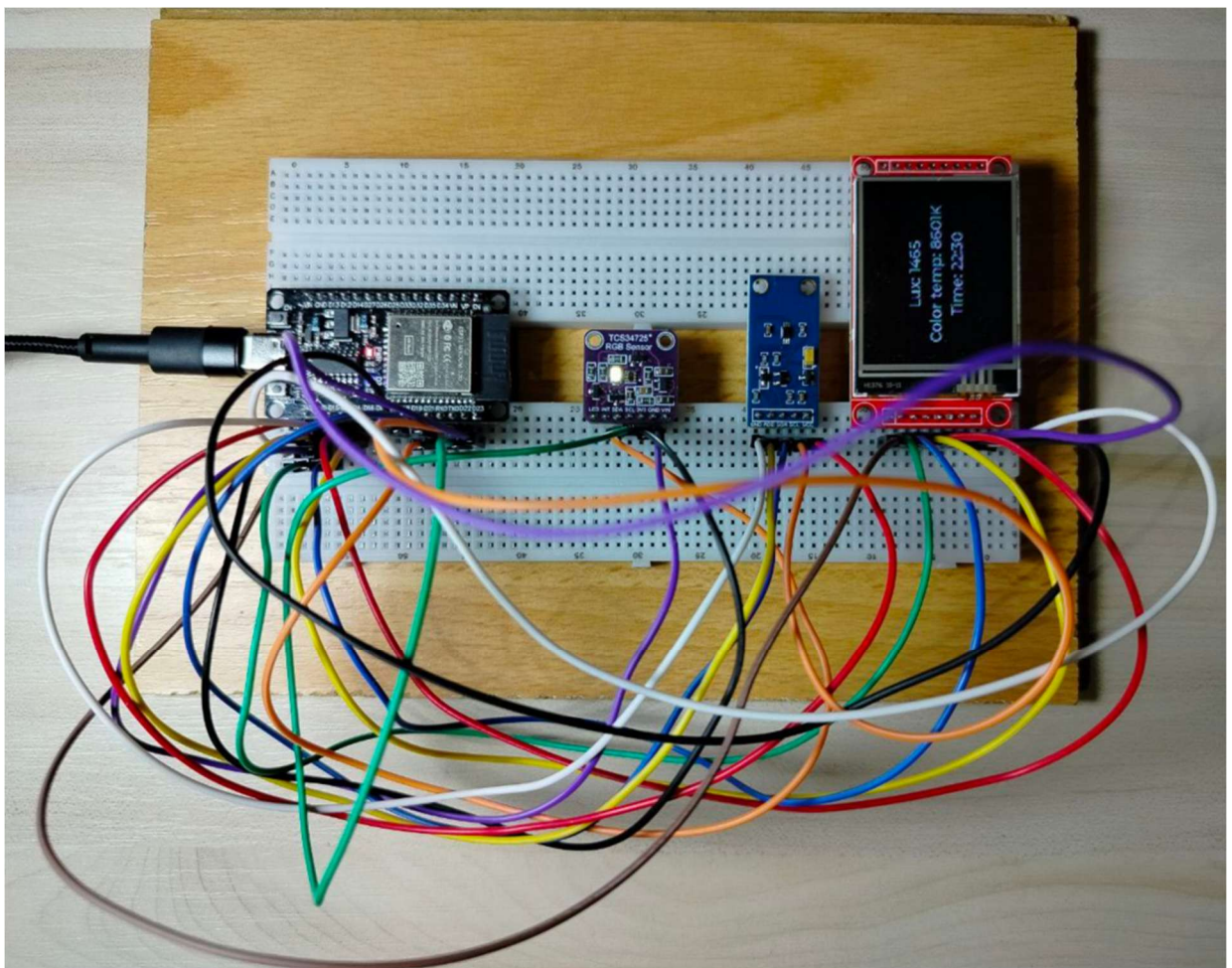


Рисунок 4.1 – Портативна система моніторингу розумного освітлення на основі ESP32

Після ввімкнення пристрою користувачу не потрібно виконувати жодних додаткових дій – система самостійно переходить у режим циклічного

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

вимірювання та реагування. Протягом більшої частини часу мікроконтролер знаходиться у режимі сну, пробуджуючись лише на короткі проміжки для збору даних, оновлення інформації на екрані та регулюванні штучного освітлення за необхідності.

При першому підключенні до Wi-Fi мікроконтролер отримує точний час за протоколом NTP для реалізації сценаріїв адаптації освітлення до часу доби чи прив'язки до режиму дня користувача. Для роботи цієї функції потрібно вказати назву мережі Wi-Fi (SSID) та пароль. Після цього синхронізація часу відбуватиметься автоматично.

Конструкція пристрою є модульною та зручною в обслуговуванні. Усі компоненти взаємодіють між собою за допомогою двох макетних плат та стандартних роз'ємів, що дозволяє легко замінити будь-який елемент – сенсор, дисплей чи навіть певний дрiт – без необхідності паяння або спеціальної апаратури.

Результати тестування з наведено в таблиці 4.1. Для зручності аналізу деякі значення були округлені до цілого.

Таблиця 4.1 – Таблиця вхідних та вихідних значень нейромережі

Час (хв)	Освітленість (люкс)	Поточна КТ	Очікувана КТ	Прогнозована КТ	Різниця
21	11016	6873	2208	2248	40
87	1899	40994	2261	2248	13
121	189	48555	2214	2248	34
130	11394	1585	2281	2248	33
251	23897	5675	2287	2248	39
276	43323	40757	2244	2248	4
330	53810	44974	2286	2248	38
343	41606	19457	2225	2248	23

Продовження таблиці 4.1

Час (хв)	Освітленість (люкс)	Поточна КТ	Очікувана КТ	Прогнозована КТ	Різниця
385	21777	43573	2240	2248	8
458	8433	45758	2288	2247	41
466	19118	59900	2271	2248	24
747	2612	161	2200	2247	47
769	3890	23939	2223	2247	14
805	14502	11653	2259	2247	12
860	41434	58707	2261	2247	14
871	50680	26531	2279	2247	32
955	8792	52995	2214	2247	33
1044	2747	58822	2247	2247	0
1095	18431	23483	2294	2247	47
1123	31551	35920	2284	2247	37
1126	27480	56193	2273	2247	26
1130	18942	54638	2213	2247	34
1184	43021	9692	2264	2246	18
1215	10627	13417	2228	2246	18
1238	35773	17159	2277	2246	31
1294	25658	55387	2299	2246	53
1332	3556	3943	2252	2246	6
1337	43001	63889	2270	2246	24
1396	1267	14541	2239	2246	7
1437	30740	1021	2288	2246	42

Табличні дані в графічному вигляді представлено на рисунку 4.2:

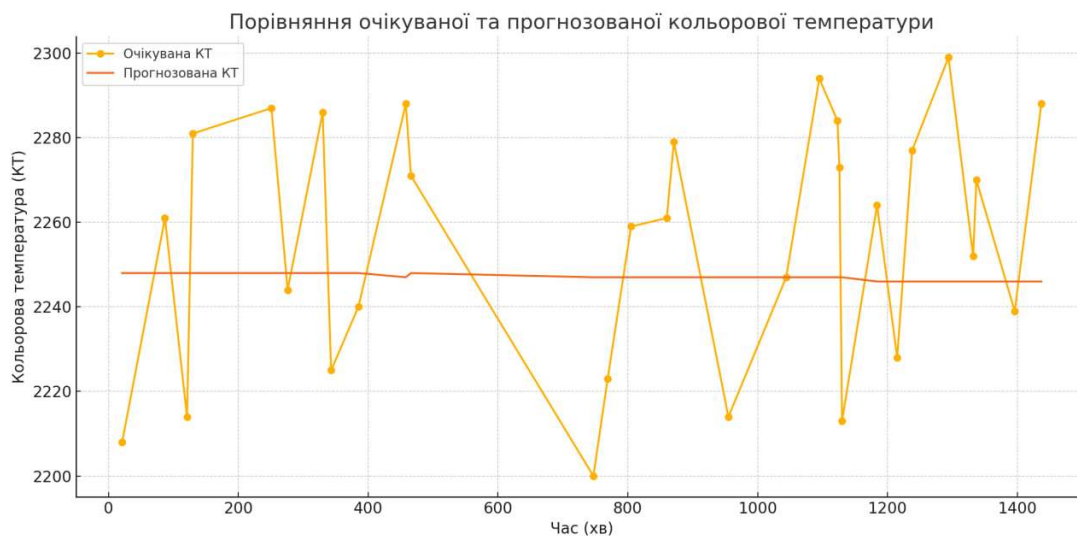


Рисунок 4.2 – Порівняння очікуваної та прогнозованої кольорової температури

4.2 Персоналізація освітлення за допомогою алгоритмів ШІ

Розроблена система автоматично адаптує освітлення під поточні умови та потреби користувача завдяки алгоритмам штучного інтелекту. Поведінка системи змінюється залежно від часу доби, яскравості навколишнього середовища та вже ввімкненого освітлення.

У ранковий час система активує теплі відтінки світла для створення м'якої атмосфери для пробудження. Вдень світло поступово стає яскравішим і холоднішим для сприяння зосередженню та природному ритму активності. Увечері кольорова температура знову переходить до теплішої, сприяючи релаксації. Вночі освітлення значно приглушується, щоб не викликати дискомфорту очей при випадковій активації.

Якщо за вікном похмуро або приміщення погано освітлене, штучне світло автоматично компенсує нестачу, адаптуючи яскравість і спектр. Якщо в кімнаті вже увімкнене освітлення, нейронна мережа враховує це при розрахунку наступного стану.

Модель може бути перенавчена. Наприклад, для тих, хто надає перевагу лише теплому або лише холодному світлу або хоче підлаштувати освітлення під розпорядок дня.

4.3 Сценарій використання в домашніх умовах

У типових умовах житла система виконує роль інтелектуального регулятора освітлення без участі користувача. Її доцільно встановлювати у кімнатах з природнім освітленням – спальнях, дитячих кімнатах або вітальнях, де протягом доби змінюється доступ до сонячного світла. Система працює безперервно, автоматично контролюючи освітлення навіть при зміні умов: від світанку до пізньої ночі:

- З 6:00 до 8:00 при кімнатному освітленні < 200 люкс пристрій створює м'яке світло з кольоровою температурою 3000-3300K.

- З 10:00 до 14:00 при зовнішній освітленості > 5000 люкс штучне освітлення автоматично зменшується до 10-15% яскравості або повністю вимикається.

- У вечірній період, після заходу сонця, світло стає теплішим (2800-3200K) з яскравістю до 70% достатньою для побутових потреб.

У разі короточасного вимкнення електроенергії пристрій автоматично відновлює свій стан: після подачі живлення від зчитує поточні умови й відновлює керування освітленням вже за 1-2 секунди.

У разі відключення сенсора або його відмови, користувач бачить на дисплеї відповідне повідомлення. Пристрій переходить у резервний режим керування світлом лише за часом доби, що гарантує мінімальну втрату функціональності.

4.4 Сценарій використання в робочих умовах

У службових приміщеннях головною перевагою системи є автоматичне підтримання оптимального рівня освітлення відповідно до умов навколишнього середовища без потреби в ручному втручанні або централізованій системі керування.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						56
Зм.	Арк	№ докум.	Підпис	Дата		

Особливо ефективною система є в кабінетах без вікон або з обмеженим природним освітленням. У такому середовищі вона виконує роль імітатора динамічного денного світла: змінює яскравість та температуру кольору відповідно до поточного часу.

У робочому режимі:

- З 9:00 до 17:00 система утримує яскраве біле світло в межах 5000-6000K, спрямоване на підтримку концентрації та зменшення зорової втоми;
- У перерви з 13:00 до 14:00 інтенсивність освітлення тимчасово знижується для створення спокійнішої атмосфери;
- У випадку зміни кількості людей у приміщенні або включення додаткового світла система адаптує режим, аби уникнути перенавантаження освітленням.

Пристрій працює автономно, не потребуючи централізованого сервера чи окремої IT-інфраструктури. Для налаштування достатньо ввести SSID Wi-Fi та пароль. Надалі всі дії виконуються локально, без залежності від підключення до інтернету.

Середній час реакції системи у робочих умовах – 300-600 мс після пробудження. За результатами випробувань, середнє відхилення між очікуваною та фактичною кольоровою температурою не перевищує 180K.

4.5 Поведінка в нестандартних ситуаціях

Система спроектована з урахуванням стійкості до помилок та непередбачених подій.

У разі повного зникнення живлення пристрій автоматично вимикається. Після відновлення енергії контролер самостійно проходить повний цикл ініціалізації: запускається основна логіка, виконується спроба підключення до раніше збереженої мережі Wi-Fi, синхронізується час (якщо доступний інтернет) та запускається зчитування з датчиків. Усі критичні параметри

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						57
Зм.	Арк	№ докум.	Підпис	Дата		

включно з поточним режимом освітлення не потребують збереження у Flash-пам'яті.

У разі втрати зв'язку з одним із сенсорів система ідентифікує це через зворотній код виклику функцій зчитування. Наприклад, при обриві дроту або відмові TCS34725 на дисплеї з'являється повідомлення відповідне повідомлення, а адаптація продовжується тільки на основі рівня освітленості та часу доби. Якщо недоступний люксметр GY-30, то кольорова температура прогнозується з поправкою на фіксований рівень яскравості. Якщо недоступні обидва сенсори, система переходить у режим часової автоматизації, при якому доба розбивається на п'ять інтервалів, кожному з яких відповідає визначення значення CCT.

У разі втрати Wi-Fi або інтернету система не припиняє роботу. RTC-таймер ESP-32 продовжує облік часу і, навіть з урахуванням похибки в 2 хвилини на добу, точність достатня для коректного освітлення у побутовому середовищі. Система намагається постійно повторювати синхронізацію, а виконує повторне з'єднання лише один раз на 6 годин для економії енергії.

У разі збоїв у моделі III система переходить в fallback-режим. У цьому випадку діапазони освітлення жорстко зашиті в прошивку, наприклад:

- 05:00 – 08:00 ➔ 3200K;
- 08:00 – 17:00 ➔ 6000K;
- 17:00 – 22:00 ➔ 3500K;
- 22:00 – 05:00 ➔ 2200K.

4.6 Потенційне розширення можливостей

Хоч основний функціонал реалізований та протестований, система залишається відкритою до вдосконалення.

Одним із найперспективніших напрямків розвитку системи є інтеграція веб-інтерфейсу для віддаленого керування та моніторингу. Для максимальної ефективності найкращим варіантом буде класичний фронтенд-стек

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						58
Зм.	Арк	№ докум.	Підпис	Дата		

HTML+CSS+JavaScript. HTML забезпечує структурну розмітку, CSS — оформлення і адаптивність, а JavaScript відповідає за динаміку та взаємодію з ESP32. Такий підхід не вимагає встановлення прт-пакетів, компіляторів або сторонніх бібліотек, що критично важливо для мікроконтролеру. Уся логіка працює на клієнтській стороні, а ESP32 лише обробляє HTTP-запити, що мінімізує навантаження на мікроконтролер.

Окремий вектор розвитку — підключення системи до хмарних платформ таких як Firebase, Blynk або власні MQTT-брокери. Це дасть змогу зберігати історію показників, реалізувати push-сповіщення, керувати освітленням не тільки в межах Wi-Fi, а й через мобільний інтернет, та інтегрувати систему з іншими IoT- пристроями — наприклад, розумними розетками або голосовими асистентами.

Також можливо реалізувати більш гнучкі сценарії керування освітленням. Наприклад створення індивідуальних профілів для кожного члена сім'ї, що активуються за допомогою розкладу або навіть Bluetooth-маячків.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						59
Зм.	Арк	№ докум.	Підпис	Дата		

ВИСНОВКИ

У межах дипломного проєкту було реалізовано портативну автономну систему адаптивного освітлення на базі мікроконтролера ESP32 з акумуляторним живленням, локальною обробкою даних та бездротовим зв'язком. В основі пристрою використовуються два сенсори: люксметр GY-30 BH1750FVI та датчик кольорової температури TCS34725 для визначення спектру природнього світла. Компоненти змонтовано на макетних платах з модульними з'єднаннями, що забезпечує легке обслуговування, заміну та модернізацію компонентів.

Основу адаптивного керування складає нейронна мережа, навчена на великому наборі синтетичних даних, які моделюють зміну природного освітлення протягом доби. У процесі генерації враховувалися як хронологічні закономірності, так і зміни кольорової температури залежно від рівня освітленості.

У результаті система продемонструвала стабільну роботу в умовах змінного зовнішнього освітлення включно з різкими перепадами та частковим затіненням датчиків, здатність до адаптації в реальному часі завдяки оновленню кожні 200 мс без затримок, ефективне функціонування без підключення до зовнішніх контролерів та мережевих ресурсів, повну інтеграцію сенсорної частини та мікроконтролера з мінімальними апаратними залежностями, гнучку модульну архітектуру з легкою заміною або вдосконаленням окремих компонентів системи, низьку собівартість та придатність до використання у «розумних» системах освітлення в побуті, освітніх закладах, офісних та промислових приміщеннях.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						60
Зм.	Арк	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kyba, C. C. M., et al. (2017) Artificially lit surface of Earth at night increasing in radiance and extent // ScienceAdvances. – Режим доступу: <https://www.science.org/doi/10.1126/sciadv.1701528>. – Дата звернення: травень 2025.
2. Martinez-Molina, A., Hernandez-Callejo, L., Varela, F. (2022) Digital Twin-Based Assessment Framework for Energy Savings in University Classroom Lighting // MDPI. – Режим доступу: <https://www.mdpi.com/2075-5309/12/5/544>. – Дата звернення: травень 2025.
3. Mobile device [Електронний ресурс] // Wikipedia. – Режим доступу: https://en.wikipedia.org/wiki/Mobile_device. – Дата звернення: травень 2025.
4. Luminance Sensor S-LUX-ZB [Електронний ресурс] // Innoprotech. – Режим доступу: <https://innoprotech.cc/product/luminance-sensor-s-lux-zb/>. – Дата звернення: травень 2025.
5. Mi Smart Light Detection Sensor [Електронний ресурс] // Mi-Home.pl. – Режим доступу: <https://mi-home.pl/products/mi-smart-light-detection-sensor>. – Дата звернення: травень 2025.
6. Philips Motion Sensor Detecteur de Mouvement [Електронний ресурс] // Amazon. – Режим доступу: <https://www.amazon.fr/Philips-Motion-Sensor-D%C3%A9tecteur-Mouvement/dp/B0B5LQQ7GW>. – Дата звернення: травень 2025.
7. Shelly Detecteur de Mouvement Domotique, Detecteur de Temperature Compatible [Електронний ресурс] // Amazon. – Режим доступу: <https://www.amazon.fr/Shelly-Domotique-D%C3%A9tecteur-temp%C3%A9rature-Compatible/dp/B0B8TLQ6M4>. – Дата звернення: травень 2025.
8. Privacy Risks in Smart Home Apps: A Closer Look at Data Collection [Електронний ресурс] // ComplexDiscovery. – Режим доступу:

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

<https://complexdiscovery.com/privacy-risks-in-smart-home-apps-a-closer-look-at-data-collection/>. – Дата звернення: травень 2025.

9. Ambient Light Sensor Market [Електронний ресурс] // Gmmsights. – Режим доступу: <https://www.gminsights.com/industry-analysis/ambient-light-sensor-market>. – Дата звернення: травень 2025.

10. Europe Lux Intensity Meter Market Size, Segments, Forecast [Електронний ресурс] // LinkedIn. – Режим доступу: <https://www.linkedin.com/pulse/europe-lux-intensity-meter-market-size-segments-forecast-zalbf/>. – Дата звернення: травень 2025.

11. A Guide for the ESP32 Microcontroller Series [Електронний ресурс] // DigiKey. – Режим доступу: <https://www.digikey.com/en/maker/blogs/2024/a-guide-for-the-esp32-microcontroller-senes>. – Дата звернення: травень 2025.

12. CISC Processor [Електронний ресурс] // ElectronicDesk. – Режим доступу: <https://electronicsdesk.com/cisc-processor.html>. – Дата звернення: травень 2025.

13. ESP32 Datasheet [Електронний ресурс] // Espressif.com. – Режим доступу: https://www.espressif.com/sites/default/files/documentation/esp32_datash_eet_en.pdf. – Дата звернення: травень 2025.

14. Boot Mode Selection [Електронний ресурс] // Docs.espressif. – Режим доступу: <https://docs.espressif.com/projects/esptool/en/latest/esp32/advanced-topics/boot-mode-selection.html>. – Дата звернення: травень 2025.

15. Understanding SPI [Електронний ресурс] // Medium. – Режим доступу: <https://medium.com/@shathiralakdilu/understanding-spi-6c7ea9eb8e91>. – Дата звернення: травень 2025.

16. BH1750FVI Datasheet [Електронний ресурс] // Alldatasheet. – Режим доступу: <https://www.alldatasheet.com/datasheet-pdf/pdf/338083/ROHM/BH1750FVI.html>. Дата звернення: травень 2025.

17. TCS34725 Sensor [Електронний ресурс] // Esphome. – Режим доступу: <https://esphome.io/components/sensor/tcs34725.html>. – Дата звернення: травень 2025.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						62
Зм.	Арк	№ докум.	Підпис	Дата		

18. Espressif Documentation [Електронний ресурс] // Espressif Systems.
– Режим доступу: <https://docs.espressif.com/>. – Дата звернення: червень 2025.

					<i>ІАЛЦ. 045420.004 ПЗ</i>	Арк.
						63
Зм.	Арк	№ докум.	Підпис	Дата		

ДОДАТОК А

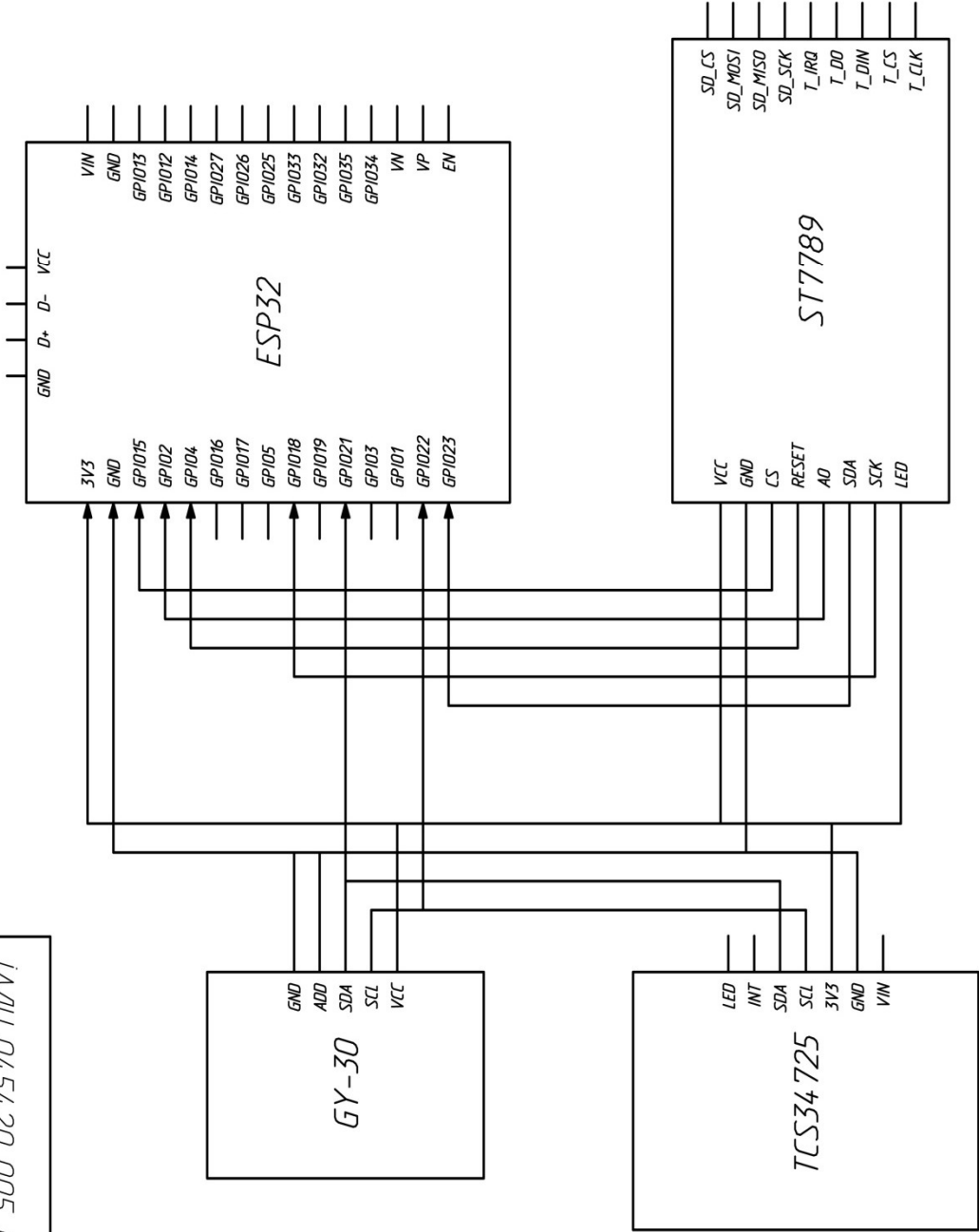
Портативна система моніторингу розумного освітлення на основі ESP32

Схема функціональна

ІАЛЦ.045420.005 Е2

Аркушів: 1

Київ – 2025



ESP32	Мікроконтролер з Wi-Fi та Bluetooth
GY-30	Сенсор освітленості, цифровий люксметр
TCS34725	RGB-сенсор, вимірює колір і КТ
ST7789	Кольоровий TFT-дисплей для виводу даних

ІА/ПЦ.045420.005 Е2										
				Портативна система моніторингу розумного освітлення на основі ESP32. Схема функціональна				Лист	Маса	Масштаб
								Аркуш 1		
								КПІ ім. Ігоря Сікорського, ФЛМ, КВ-13		

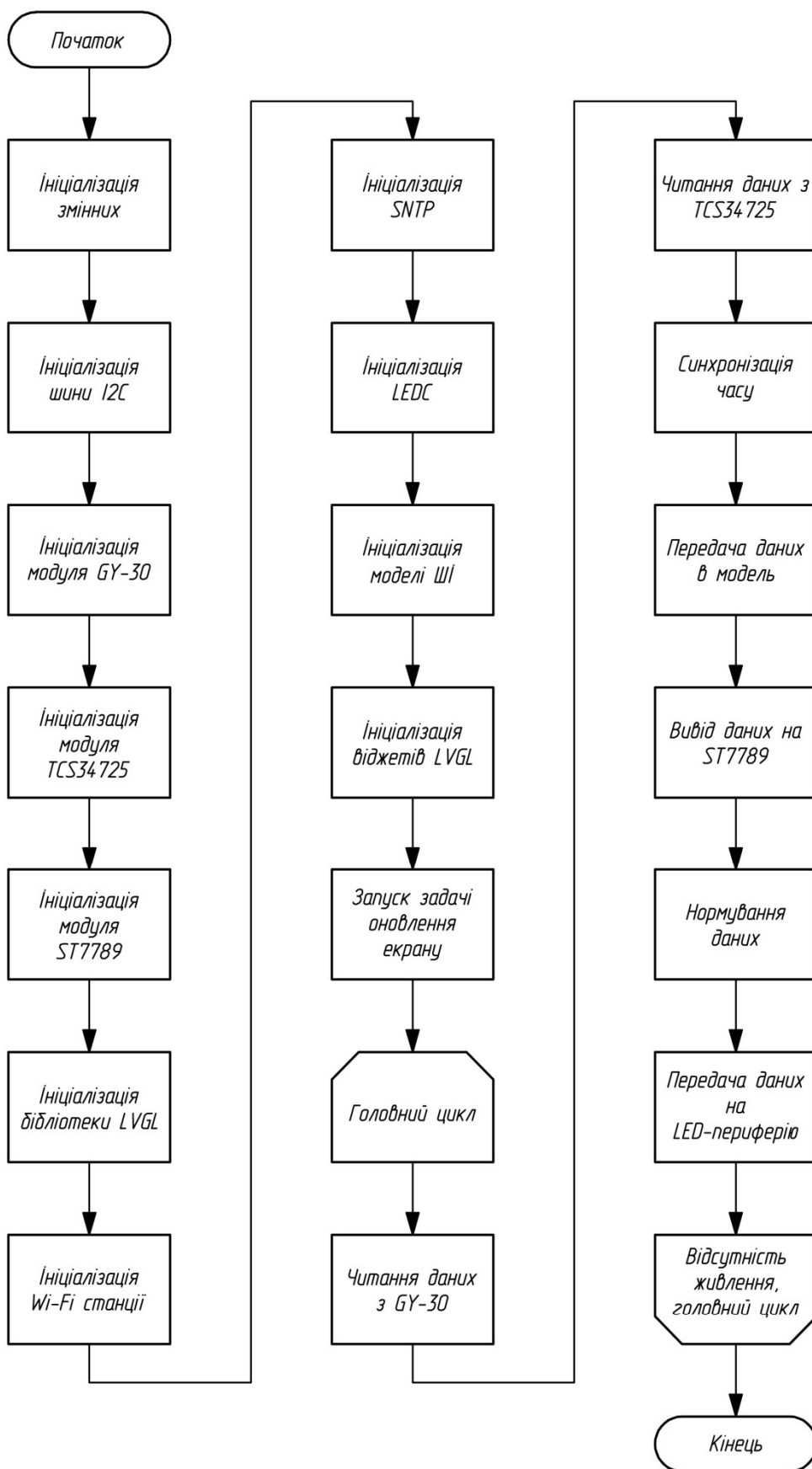
ДОДАТОК Б

Портативна система моніторингу розумного освітлення на основі ESP32

Блок-схема алгоритму функціонування системи

ІАЛЦ.045420.006 Д1

Аркушів: 1



Зм.	Арк.	N докум	Підпис	Дата	
Розроб.		Гармаш Д.О.			
Перев.		Сергієнко П.А.			
Т. контр.					
Н. контр.		Клятенко Я.М.			
Затв.		Романкевич В.О.			

ІАЛЦ.045420.006 Д1

Портативна система моніторингу розумного освітлення на основі ESP32.
Блок-схема алгоритму функціонування системи

Літ.			Маса		Масштаб	
Аркуш 1				Аркуш 1		
КПІ ім. Ігоря Сікорського, ФПМ, КВ-13						

ДОДАТОК В

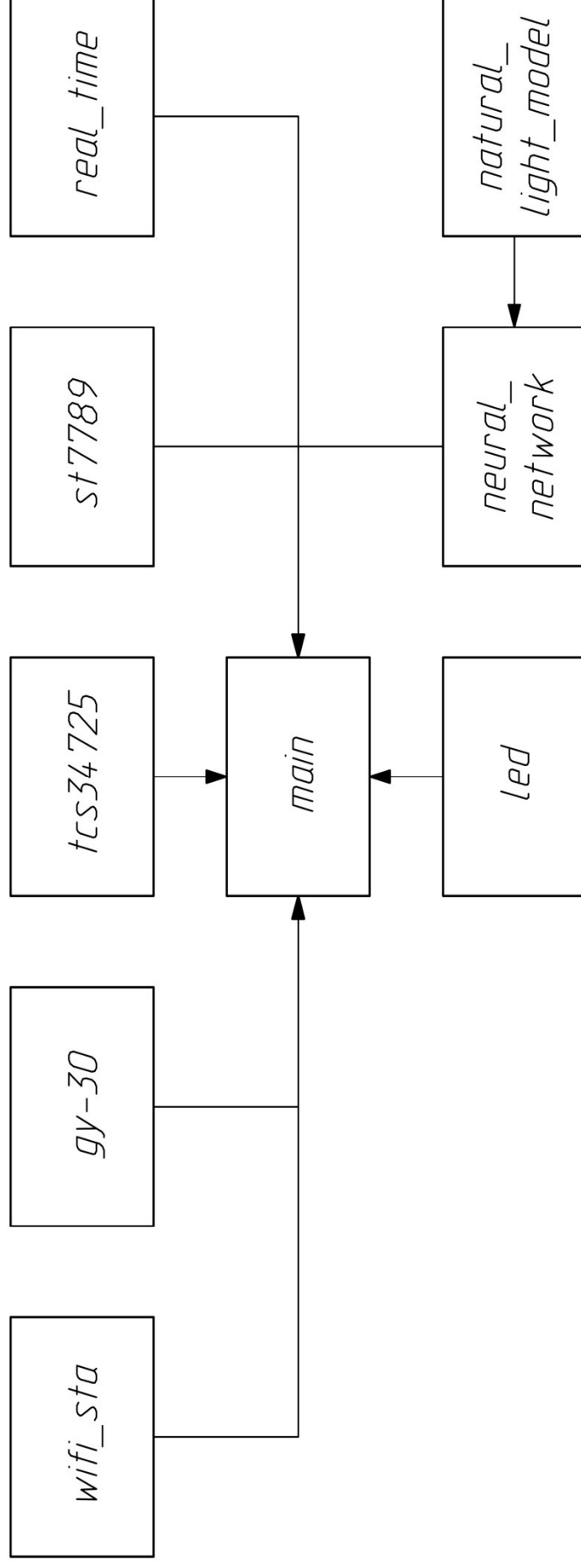
Портативна система моніторингу розумного освітлення на основі ESP32

Структурна схема взаємодії модулів підпрограм

ІАЛЦ.045420.007 Д2

Аркушів: 1

IA/ML.045420.007 A2

[illegible]

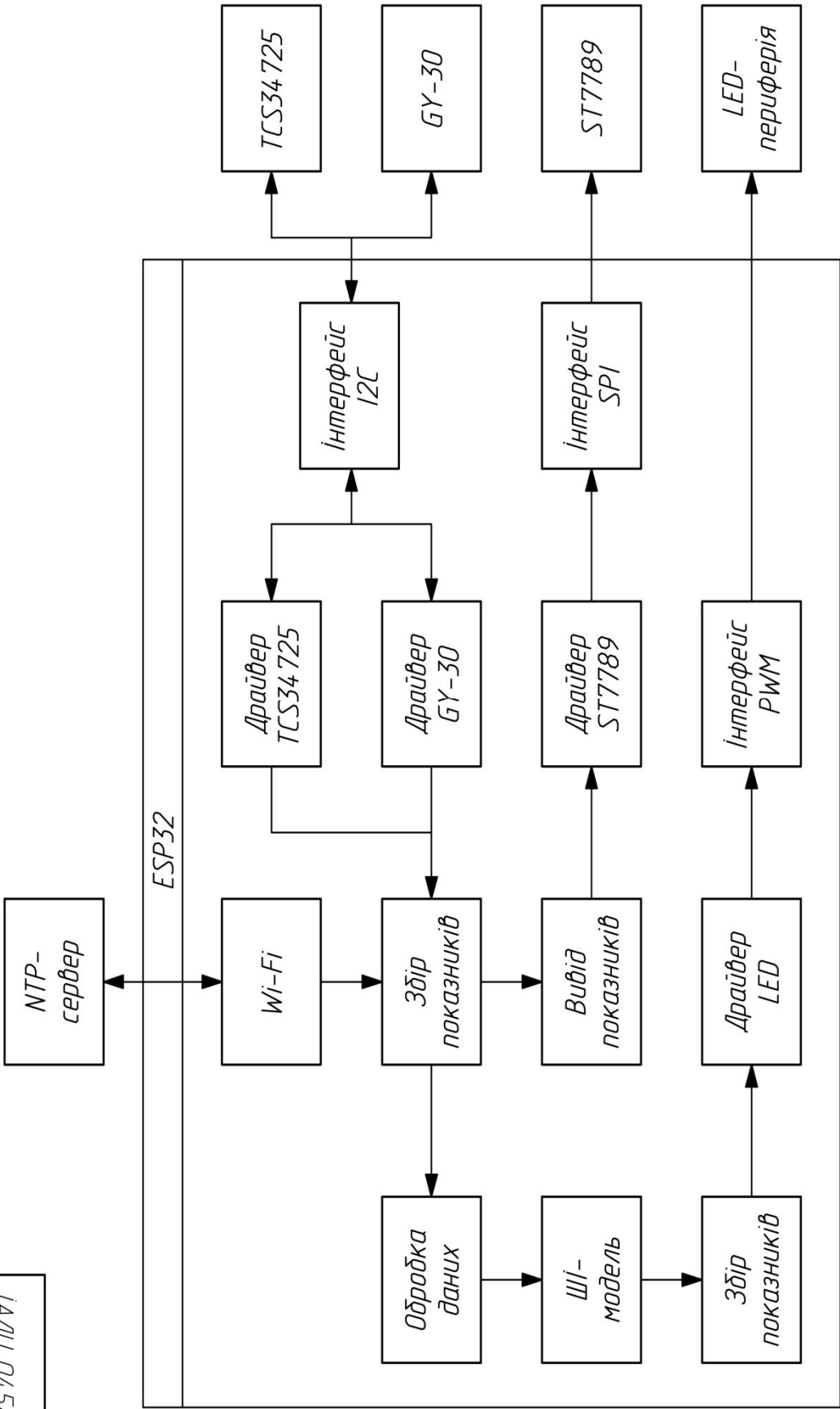
ДОДАТОК Г

Портативна система моніторингу розумного освітлення на основі ESP32

Структурна схема зв'язків програмних і апаратних модулів

ІАЛЦ.045420.008 ДЗ

Аркушів: 1



ІА/ПЦ.045420.008 ДЗ											
Портативна система моніторингу розмірного освітлення на основі ESP32. Структурна схема зб'язків програмних та апаратних модулів					Лист		Маса		Масштаб		
					Аркуш 1		Аркуш 1				
					КПІ ім. Ігоря Сікорського, ФЛМ, КВ-13						

ДОДАТОК Г

Портативна система моніторингу розумного освітлення на основі ESP32

Презентація

Аркушів: XX

ДОДАТОК Д

Портативна система моніторингу розумного освітлення на основі ESP32

Лістинг коду

Аркушів: XX