

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

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр Коваль

«__» _____ 2021 р.

Дипломна робота

на здобуття ступеня бакалавра

**спеціальності 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення розподілених
систем»**

**на тему: «Автоматизована система керування мультizonальним освітленням
будівлі»**

Виконала:

студентка IV курсу, групи ТІ-72

Васіна Єлизавета Олександрівна

Керівник:

Ст. викладач

Сарибога Ганна Володимирівна

Консультант:

Рецензент:

Ст. викладач

Наливайчук Миколай Васильович

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

Студент (-ка) _____

Київ — 2021 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

спеціальності 121 «Інженерія програмного забезпечення»

освітня програма «Інженерія програмного забезпечення розподілених систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) Олександр Коваль

” ____ ” _____ 2021р.

ЗАВДАННЯ

на дипломну роботу студенту

Васіной Єлизаветі Олександрівні

(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система керування мультизональним освітленням будівлі

керівник роботи Сарибога Ганна Володимирівна, ст. викладач.
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”24” травня 2021р. № 1267-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи мова програмування Dart та C, фреймворк Flutter, елементи Arduino, середовище розробки Android Studio

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) розглянути існуючі системи освітлення, виявити їх переваги та недоліки, на основі чого розробити власний мобільний застосунок для автоматизованої системи мультизонального освітлення будівлі; реалізувати програмне забезпечення, призначенням якого є врегулювання освітлення в приміщенні відповідно часу доби або вручну та створення комфортних умов освітлення для певного виду діяльності людей; здійснити тестування системи.

5. Перелік ілюстративного матеріалу:

Мета та задачі. Огляд існуючих рішень. Актуальність. Основний функціонал системи. Засоби розробки. Діаграма розгортання. Use-Case Діаграма. Діаграма класів. Діаграма компонентів. Апаратна частина програмного забезпечення. Вікна мобільного застосунку. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ” ____ ” _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	16.02.2020	
2.	Вивчення та аналіз задачі	09.10.2020 – 23.01.2021	
3.	Розробка архітектури та загальної структури системи	09.03.2021 – 27.03.2021	
4.	Розробка структур окремих підсистем	28.03.2021 – 01.04.2021	
5.	Програмна реалізація системи	05.04.2021 – 28.04.2021	
6.	Оформлення пояснювальної записки	20.05.2021 – 04.06.2021	
7.	Захист програмного продукту	14.05.2021	
8.	Передзахист	28.05.2021	
9.	Захист	15.06.2021	

Студентка _____
(підпис)

Керівник роботи _____
(підпис)

Васіна Є.О.
(прізвище та ініціали,)

Сарибога Г.В.
(прізвище та ініціали,)

АНОТАЦІЯ

Основною метою даного проекту є аналіз існуючих систем керування освітленням в приміщенні і розробка власної системи керування освітленням певної будівлі через використання мобільного додатку та веб-інтерфейсу за допомогою Wi-Fi та Bluetooth.

Призначенням даної системи є врегулювання освітлення в приміщенні відповідно часу доби або вручну та створення комфортних умов освітлення для певного виду діяльності людей.

Дана система передбачає наступний функціонал:

1. Повний контроль рівня освітленості в кожному приміщенні будівлі
2. Управління інтенсивністю, температурою та яскравістю світла вручну
3. Автоматична адаптація світла за допомогою прив'язки до часу доби
4. Встановлення автоматичного включення або виключення за допомогою прив'язки до астрономічного сходу або заходу сонця
5. Можливість використання трьох мов: англійська, українська та російська.

Пояснювальна записка складається зі вступу, чотирьох розділів, висновку, списку використаних джерел; містить 59 сторінки, 23 рисунки та 3 додатки. Список використаних джерел включає 22 бібліографічних найменувань.

Ключові слова: система керування освітленням, світлодіоди, мобільний додаток, властивості світла, будівля, апаратна частина.

ABSTRACT

The main goal of this project is to analyze the existing lighting control systems in the room and develop your own lighting control system for a particular building through the use of a mobile application and web interface using Wi-Fi and Bluetooth.

The purpose of this system is to adjust the lighting in the room according to the time of day or manually and create comfortable lighting conditions for a particular type of human activity.

This system provides the following functionality:

1. Full control of the level of illumination in each room of the building
2. Control the intensity, temperature and brightness of light manually
3. Automatic adaptation of light by reference to the time of day
4. Set automatic on or off by linking to the astronomical sunrise or sunset
5. Ability to use three languages: English, Ukrainian and Russian.

The explanatory note consists of the introduction, four sections, conclusions, the list of the used sources; contains 59 pages, 23 pictures and 3 appendices. The list of used sources contains 22 bibliographic names.

Keywords: lighting control system, LEDs, mobile application, light properties, building, hardware.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП.....	9
1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Поняття і можливості системи керування освітленням.....	11
1.2 Основні функції системи керування освітленням.....	12
1.3 Керування зонами освітлення	14
1.4 Огляд існуючих рішень систем управління освітленням	16
1.4.1 Провідні системи управління освітлення	17
1.4.2 Бездротові системи управління освітленням	18
1.4.3 Приклади популярних систем освітлення сьогодні.....	19
1.4.4 Системи освітлення для учбових закладів	23
Висновки до розділу	25
2. ОСНОВНІ ПРИЦИПИ ПРОЕКТУВАННЯ СИСТЕМ КЕРУВАННЯ ОСВІТЛЕННЯМ	27
2.1 Рівномірність освітленості та контраст яскравості.....	27
2.2 Зовнішній вигляд кольору та відблиски	29
2.3 Мерехтіння та моделювання предметів.....	30
2.4 Проектування загального освітлення	32
2.5 Світлодіодні світильники	34
2.6 Візуалізація кольорів	37
3. ОПИС ВИКОРИСТАНИХ ПРОГРАМНИХ ЗАСОБІВ	40
3.1 Засоби розробки	40

	7
3.1.1 Мова програмування Dart	40
3.1.2 Фреймворк Flutter	41
3.1.3 Елементи Arduino	42
3.2 Середовище розробки	44
4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	45
4.1 Опис функціональності системи	46
4.2 Структура інформаційної бізнес–моделі	47
4.3 Структура діаграми компонентів	48
4.4 Опис UI flow діаграми	50
4.5 Апаратна частина програмного забезпечення	51
4.6 Інструкція з використання програмного продукту	53
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
ДОДАТОК А	60
ДОДАТОК Б	62
ДОДАТОК В	75

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- SPD** – Spectral Power Distribution (спектральний розподіл потужності)
CCT – Correlated Color Temperature (корельована температура кольору)
CRI – Color Rendering Index (індекс кольоропередачі)
UI – User Interface (користувацький інтерфейс)
IoT – Internet of Things (інтернет речей)
ПЗ – програмне забезпечення

ВСТУП

У теперішні часи близько 40 відсотків генеруючої в світі електроенергії і до 37 відсотків всіх електричних ресурсів використовується в житлових і громадських будівлях, при цьому значну частку в енергоспоживанні становить енергія на освітлення.

Скорочення витрат електроенергії на ці цілі можливо двома основними шляхами: зниженням номінальної потужності освітлення або зменшенням часу використання світильників. Зменшення встановленої потужності освітлення в першу чергу означає перехід до більш ефективних джерела світла, що дає потрібний світловий потік при істотно меншим споживанні електроенергії. Зменшення часу використання світильників може бути досягнуто впровадженням сучасних систем управління і контролю освітлювальних мереж.

Тому в останнє десятиліття тема автоматизації управління внутрішнім освітленням стала дуже актуальною. Сучасні системи поєднують в собі значні можливості економії електроенергії з максимальною зручністю для користувачів.

Як правило вони виконують наступні функції:

- Повна підтримка штучної освітленості в приміщенні на заданому рівні. Досягається це введенням в систему управління освітленням фотоелемента, що знаходиться усередині приміщення.
- Облік природної освітленості в приміщенні. Незважаючи на наявність у переважній більшості приміщень природного освітлення у світлий час доби, потужність освітлювальної установки розраховується без його урахування.
- Облік часу доби і дня тижня. Додаткова економія енергії в освітленні може бути досягнута відключенням освітлювальної установки в певні години доби, а також у вихідні і святкові дні. Цей захід дозволяє ефективно боротися із забудькуватістю

людей, що не відключають освітлення на робочих місцях перед своїм відходом. Для її реалізації автоматизована система управління освітленням має бути обладнана власним годинником реального часу.

- Облік присутності людей в приміщенні. При устаткуванні системи управління освітленням датчиком присутності можна включати і відключати світильники залежно від того, чи є люди в цьому приміщенні.

- Безпроводне керування освітлювальною установкою. Хоча така функція не є автоматизованою, вона часто присутня в автоматизованих системах управління освітленням завдяки тому, що її реалізація на базі електроніки системи управління освітленням дуже проста, а сама функція додає значну зручність в управлінні освітлювальною установкою.

Отже, у багатьох будинках різного призначення: промислових, житлових, адміністративних, освітлення становить більшу частину від загальної споживаної електроенергії. А енергозбереження в освітлювальних установках істотно впливає на витрату електроенергії, тому проблеми її якості і раціональні методи експлуатації є надзвичайно важливими.

Записка містить 4 розділи:

У першому розділі описується предметна область та постановка задачі автоматизованої системи управління мультизональним освітленням будівлі.

У другому розділі описуються основні принципи проектування систем освітлення.

У третьому розділі описані засоби реалізації програмної системи.

У четвертому розділі дано проектування та опис реалізованого програмного продукту та його використання користувачем.

1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Тенденція до IoT (Інтернету речей) відкрила нові перспективи для управління освітленням. Індустрія освітлення зазнає радикальні перетворення, викликані концепцією освітлення, орієнтованої на людину, і величезним розвитком розумних міст, розумних будинків і автоматизації будівель на базі Інтернету речей. Системи інтелектуального освітлення все частіше покладаються на повсюдну бездротовий зв'язок для реалізації розширених функцій. Інтеграція датчиків, сумісність пристроїв і розробка програмного забезпечення - ось деякі з нових завдань, які представляє Інтернет речей [1].

Світло є частиною нашого повсякденного життя, і деякі навіть забули про його важливість. З цієї причини існує потреба оновити спосіб використання та взаємодії з вогнями. Тема автоматизованих систем управління освітленням дуже актуальна. Ці системи є необхідністю майбутнього та сьогодення будь-якого будинку.

1.1 Поняття і можливості системи керування освітленням

Система управління освітленням - це електронна мережа, яка дозволяє автоматично регулювати включення, виключення, інтенсивність світла. Контроль над освітленням ведеться з використанням різних протоколів. Кожен з них - як мова програмування, за допомогою якого прилади спілкуються між собою.

Система дистанційного керування освітленням в поєднанні з енергоефективними світлодіодними світильниками дозволяє заощадити від 30 до 80% електроенергії. Показник буде залежати від застосовуваних налаштувань і датчиків, інтенсивності природного світла, частоти руху персоналу і техніки.

Управління освітленням відіграє важливу роль в системах освітлення, дозволяючи користувачам вручну або автоматично.

Системи управління освітленням роблять будівлю більш ефективною, оскільки існує централізоване управління кожним аспектом освітлення. Завдяки

інтелектуальній технології освітлення ви можете керувати всіма світильниками просто натисканням кнопки, будь то на смартфоні чи планшеті. Тому ви не зобов'язані ходити, перемикаючи кожне світло. Ви можете встановити умови, в яких слід вмикати та вимикати світло. Автоматизоване освітлення зробить місце більш ефективним, коли мова зайде про економію енергії. Ви можете встановити датчики вакансій у свою систему, щоб увімкнути освітлення лише тоді, коли хтось знаходиться в кімнаті, і вимкнути їх, коли нікого немає.

Контроль освітлення також гарантує, що у будівлі буде безпека. Це головним чином завдяки зовнішньому освітленню. Їх можна запрограмувати таким чином, щоб увімкнути при заході сонця і вимкнути при сході сонця. Таким чином, кожен куточок приміщення буде освітлений, крім того, ви зможете супроводжувати систему освітлення датчиками руху, які зможуть зробити їх розумнішими та відчувати будь-яку форму вторгнення.

Розумні технології по-своєму стильні, більшість із них зроблені таким чином, що вони можуть відповідати вашому декору як інтер'єру, так і екстер'єру. Використання центрального управління гарантує, що більше немає перемикачів, оскільки ви перейдете на розумні пристрої для управління світлом.

1.2 Основні функції системи керування освітленням

Органи управління освітленням - це пристрої та системи введення / виведення. Система управління отримує інформацію, вирішує, що з нею робити, а потім відповідно регулює потужність освітлення. На рисунку 1.2.1 ми бачимо основний контур освітлення (перемикач). Потужність рухається по ланцюгу, щоб підсилити групу вогнів. Ця система освітлення забезпечує освітлення [2].



Рисунок 1.2.1 – Основний контур освітлення

1. Перемикання: одним з основних результатів є перемикання. На рисунку 1.2.2 ми бачимо перемикач, розміщений на лінії між джерелом живлення та навантаженням. Коли перемикач замикається (тобто перемикач перемикається в положення "УВІМК."), схема завершується, дозволяючи потужності надходити до навантаження. Коли він розмикається, ланцюг розривається (перемикається у положення «ВИМК.»), що порушує живлення навантаження. Це робить перемикач контролером живлення.

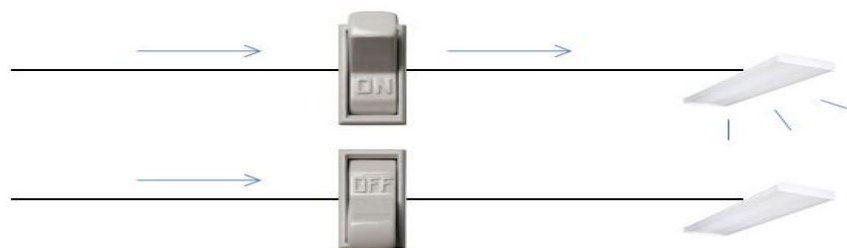


Рисунок 1.2.2 - Перемикання

2. Затемнення: інший основний вихід - затемнення. Якщо використовується диммер-перемикач, крім УВІМК. / ВИМК., він може змінити струм, що протікає через навантаження під час стану УВІМК., що підвищує або знижує світловіддачу. На рисунку 1.2.3 ми бачимо диммер, розміщений на лінії, при цьому вихідний сигнал постійно затемнюється в діапазоні затемнення навантаження.

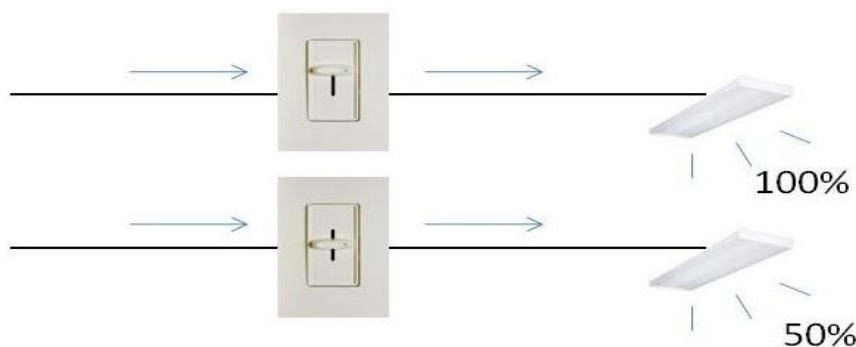


Рисунок 1.2.3 - Затемнення

3. Контроль кольору: зі світлодіодами досить економічно надавати користувачам можливість регулювати колір освітлення. З регульованими білими світлодіодними пристроями окремо затемнені масиви теплих та холодно-білих світлодіодів дозволяють користувачам регулювати температуру джерела світла.

Можуть бути додані інші кольори, щоб покращити доступний колірний спектр та забезпечити хорошу передачу кольорів.

Ще два підходи - приглушення до тепла (світлодіодні вироби, які приглушуються до дуже теплого білого кольору, подібного до розжарювання), і повнокольорова настройка (окремо регульовані світлодіодами червоного, зеленого та синього кольорів плюс жовтий або білий та потенційно інші кольори).

1.3 Керування зонами освітлення

Зонування є важливим аспектом проектування системи управління освітленням, оскільки зонування - це механізм, за допомогою якого елементи керування освітленням призначаються світловим навантаженням. Зона управління визначається як одне або декілька джерел світла, керованих одночасно одним керуючим виходом. Зони можуть бути організовані відповідно до енергетичних норм, бажаної економії енергії та гнучкості, загального освітлювального обладнання (наприклад, флуоресцентного та світлодіодного), характеристик простору (наприклад, меблі та оздоблення), завдань, наявності денного світла та графіків освітлення.

Зони освітлення просто означають ділянки світла, які ви хочете використовувати разом. Зони дозволяють контролювати більше одного світла або освітлювальної зони одночасно, тобто вам не потрібно обійтися окремо, контролюючи кожне світло окремо. Однозонове управління освітленням просто означає можливість контролю освітлення в одній зоні. Наприклад, якщо у вас є кілька

областей освітлення на кухні, але вас задовольняє контроль і затемнення всіх ліхтарів разом, ви можете використовувати однозонове управління освітленням.

Мультизональне управління освітленням - це система, яка дозволяє індивідуально контролювати різні зони освітлення. Це було б корисно, якщо ви встановили освітлення в декількох зонах кімнати, але не завжди хочете, щоб кожен шар вмикався або вимикався одночасно. Наприклад, якщо ви встановлюєте на кухню зони освітлення завдань, а також шари зовнішнього освітлення (докладніше про різницю дізнайтеся тут), вам потрібно мати можливість увімкнути зовнішнє освітлення окремо від освітлення завдання, щоб створити м'яке та розслаблююче атмосфера в кімнаті. На Рисунку 1.3.1 ми можемо побачити план будівлі, її головні зони, якими керують користувачі за допомогою системи мультизонального освітлення [3]:

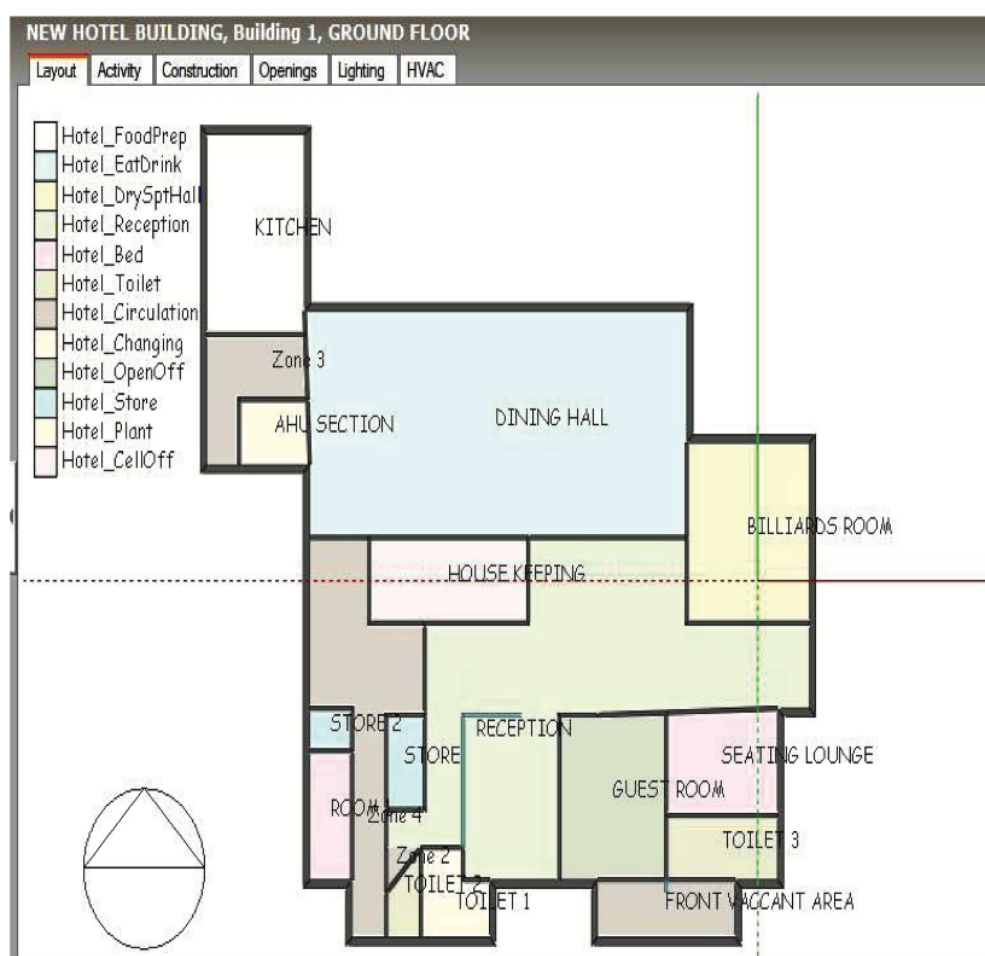


Рисунок 1.3.1 – Зони освітлення

Менші зони управління (більша деталізація зон у просторі чи будівлі) забезпечують більшу гнучкість та, як правило, більшу економію енергії. З цієї причини більшість енергетичних кодексів регулюють зонування контролю, встановлюючи обмеження площі. Традиційно контроль зонування та майбутнього перепланування обмежувався електропроводкою освітлювальних схем.

Досягнення комунікацій дозволяють порівняно економне зонування настільки гранульованого, як окремі світильники або баласты / драйвери, а також зонування та перепланування з використанням програмного забезпечення замість жорсткої проводки.

1.4 Огляд існуючих рішень систем управління освітленням

Різні програми та програмне забезпечення підтримують впровадження систем управління освітленням. Найнадійніше програмне забезпечення доступне для централізованих інтелектуальних мережевих систем управління освітленням [4]. Перебуваючи на сервері або в хмарі, програмне забезпечення може надавати безліч функцій, таких як:

- 1) виявити контрольні точки (пристрої тощо);
- 2) призначити контрольні точки зонам;
- 3) програмні послідовності роботи для зон;
- 4) калібрувати датчики;
- 5) контролювати пункти управління та видавати службові попередження / тривоги;
- 6) фіксувати та відображати споживання енергії та інші записані дані;
- 7) резервне копіювання даних та журналів подій та створення користувачів / рівнів доступу.

Існує дуже багато систем керування освітлення сьогодні. Розглянемо найбільш популярні з них.

1.4.1 Провідні системи управління освітлення

Провідна система управління освітленням передає керуючий сигнал по кабелю.

Переваги:

- відносно невисока вартість;
- незалежність від перешкод і відстані;
- надійність.

Недоліки:

- потрібно сигнальний кабель;
- його необхідно монтувати в стінах окремо від кабелю живлення так, щоб уникнути наведень.

Види провідних протоколів залежно від можливостей діляться на: домашні та професіональні.

Провідне управління світлом для будинку коштує порівняно недорого, просте в підключенні і налаштуванні.

PUSH (DIM): Стандарт має найпростіший спосіб відправки команд - за допомогою кнопки. Кнопка розмикає контур управління, тим самим включає, вимикає і регулює яскравість світильника. При цьому до однієї кнопці можна підключити обмежене число приладів.

1(0)-10V: Управління світильниками здійснюється за допомогою сигналу напругою від 0 до 10 В. При цьому 10 В забезпечує максимальну яскравість, 0 В - відключення світла.

TRIAC: Працює на основі сімісторного диммера - спеціального світлорегулятора. Його перевага - не вимагає додаткових проводів; управляє світлом, використовуючи тільки кабель живлення. При організації мережі важливо врахувати сумісність обладнання, інакше світильники мерехтіти і шуміти.

Професійне управління світлом більш дороге і складне в монтажі.

DMX-512: Один з найпопулярніших стандартів, активно використовується в сценічному обладнанні. Дозволяє за допомогою однієї лінії зв'язку управляти одночасно 512 каналами. Здатний об'єднати різні типи пристроїв від різних виробників.

DALI: Широко поширений в Європі протокол. Має на увазі використання єдиного керуючого кабелю (шина DALI), до якого приєднуються всі керуючі і все керовані пристрої. Таке рішення дозволяє побудувати гнучку мережу, але має обмеження по роботі з групами світильників.

KNX: Конкурент DALI, але працює не тільки зі світильниками. KNX підтримують більшість розумних приладів від різних виробників: від регуляторів на акумуляторі до моторів на шторах. Прекрасний вибір для тих, хто хоче пов'язати системи управління освітленням і розумного будинку.

Найчастіше середовищем для передачі даних служить спеціально виділена провідна шина; іноді застосовується радіоканал. Система децентралізована, тобто поломка одного приладу не вплине на роботу мережі.

1.4.2 Бездротові системи управління освітленням

Бездротові системи управління освітленням передають сигнал за допомогою радіоканалу.

Переваги:

- швидкий монтаж;
- не потрібно кабель;
- світильники передають сигнал один одному по ланцюжку.

Недоліки:

- більш висока вартість приладів;
- прийом сигналу обмежений відстанню;

- можливі перешкоди.

Види бездротових протоколів:

RF (радіо хвилі): Працює як побутової радіоприймач - за допомогою радіохвиль. Радіус роботи приладів в приміщеннях з перешкодами - в середньому 25 метрів. Дальність дії також можна збільшувати, використовуючи спеціальні пристрої-підсилювачі.

Zigbee: Популярний бездротовий стандарт, який дозволяє приладам «спілкуватися» між собою. Дані передаються по ланцюжку від світильника до світильника. У разі поломки приладу, автоматизована система управління освітленням перебудує ланцюжок і продовжить передачу даних.

BLL (Blue Light Link): Протокол від найбільшого європейського виробника блоків живлення TCI. Компанія розвиває бездротові mesh-мережі - надкомпактне рішення для установки в світильники. Стандарт також підтримує мобільний додаток. Його перевага в тому, що не треба встановлювати окремий пристрій управління світильником - передавач вже вбудований в блок живлення.

Кількість протоколів щороку зростає. Будь-який виробник освітлювальних приладів і блоків живлення прагне випустити свою версію.

1.4.3 Приклади популярних систем освітлення сьогодні

Система Interact For Schools - розумне освітлення для шкіл вищого класу. Розробники вважають, що розумне освітлення може створити розумніші умови для навчання. Тому вони розробили програмне забезпечення для співробітників, які тепер можуть відрегулювати освітлення в класі відповідно до навчального завдання або часу доби за допомогою інтуїтивно зрозумілого додатка для смартфонів. Датчики руху та автоматизовані графіки освітлення допомагають бути більш енергоефективними та економічно ефективними, тоді як інформаційна панель Interact

Pro дозволяє відстежувати статистику для подальшої оптимізації операцій та економії. Нова система Interact Pro полегшує як ніколи насолоджуватися перевагами розумного освітлення. Це економічно вигідне та бездротове рішення для кожного кроку розумного освітлення вашого замовника: починаючи з простого освітлення та програми, отримуючи більше переваг, просто додавши шлюз.

Система є повністю масштабованою, тому завжди можливо модернізувати її пізніше, коли змінюються вимоги бізнесу, без необхідності заміни світлових точок - що дозволяє модернізувати своє освітлення з інтелектуального на супер-розумний. Переваги системи:

- Ідеально підходить для проектів будь-якого розміру та масштабу;
- Відсутність обмежень на світлові точки - ідеально підходить для офісів, промислових майданчиків, складів, шкіл та багатьох інших областей застосування;
- Працює з широким асортиментом ламп та світильників Philips Interact Ready, розумних датчиків та вимикачів, щоб покрити всі потреби в застосуванні.

На Рисунку 1.4.3.1 можна побачити приклад роботи додатку системи Interact For Schools, який призначений для керування зонами освітлення в класі:

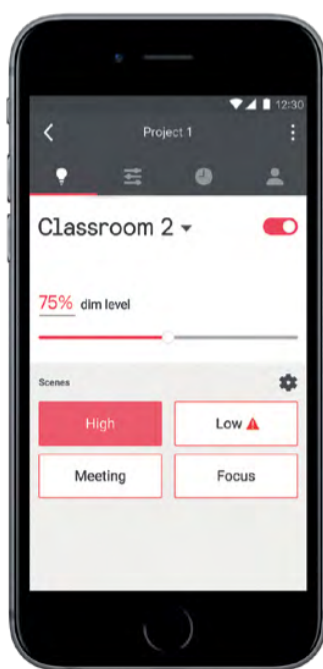


Рисунок 1.4.3.1 – Приклад роботи додатку Interact For Schools

Система Casambi – це розробка для офісних приміщень. За думкою розробників цього програмного забезпечення, сучасний офіс - це не просто робоче місце, де ми сидимо за письмовим столом. Люди переходять між зустрічами, семінарами, творчими сесіями та зоною відпочинку для кава-брейків та спілкування. Тому найбільш відповідне освітлення є необхідним елементом створення гостинного та продуктивного офісного середовища для працівників. Враховуючи, що освітлення як основна необхідність є застарілою концепцією, було доведено, що освітленість, орієнтована на людину, з відповідними відтінками та реакцією на природне та навколишнє світло збільшує продуктивність на 12%. Розробники зробили розумне освітлення, яке забезпечує різні рівні освітлення, останні можна виділити для різних завдань. Це може бути для різних сценаріїв, таких як яскраве, яскраве світло, якщо потрібна хороша видимість. І все ж, якщо відбувається робочий шопінг, можливо, потрібно менше світла та розслаблених або енергійних кольорних температур.

Схеми освітлення можуть враховувати навіть потреби людей, наприклад, різницю у віці та необхідність більшої видимості для старших демографічних груп.

Система Leviton - Розумні датчики Wallbox. Розумне, просте зондування та регулювання затемнення або перемикання. Розумні датчики Wallbox поєднують визначення зайнятості / вільних місць з затемненням 0-10 В або перемиканням ON / OFF, забезпечуючи просте рішення для легкої економії енергії, місцевого контролю та відповідності коду. Розумні датчики Wallbox пропонують легке програмування та налаштування. Це включає стандартний режим за замовчуванням для автоматичного ввімкнення / автоматичного вимкнення з 10-хвилинним тайм-аутом, а також зручне програмування кнопок для налаштувань загальної кімнати. Додаток Smart Sensor також можна використовувати для налаштування та включення додаткових функцій, таких як чутливість датчика, тайм-аути, денне освітлення, утримання, часткове увімкнення, часткове вимкнення тощо.

Приклад роботи додатку можна побачити на Рисунок 1.4.3.2:



Рисунок 1.4.3.2 – Приклад роботи системи Leviton

Система HealWell – це унікальна система освітлення, розроблена на основі досліджень, дозволяє підвищити комфорт пацієнтів в лікарні. HealWell використовує інноваційне освітлення, щоб забезпечити здоровий сон, знизити тривожність і поліпшити настрій пацієнтів і персоналу. Наукові дані свідчать, що світло значно впливає на наше здоров'я і самопочуття. Він не тільки впливає на настрій, а й управляє природними біоритмами. Пацієнти проводять в лікарнях добу безперервно. Імітуючи природне чергування дня і ночі, штучне освітлення здатне підтримувати здоровий цикл сну і неспання. Система HealWell розроблена спеціально з цією метою і автоматично змінює яскравість і відтінок світла протягом дня відповідно до обстановки на вулиці.

Випробування підтвердили, що світло позитивно впливає на пацієнтів. В результаті порівняння палат зі звичайним освітленням і палат, оснащених системою HealWell, були виявлені наступні переваги системи HealWell:

- Поліпшення настрою пацієнтів і персоналу;
- Збільшення тривалості сну пацієнтів;
- Зменшення часу засипання пацієнтів;

- Поліпшення самопочуття пацієнтів по Госпітальній шкалою тривоги і депресії.

1.4.4 Системи освітлення для учбових закладів

Роль освітлення у засвоєнні знань та процесі навчання є фундаментальною. Це дозволяє візуально дослідити фізичні характеристики досліджуваних, а також виявити поняття з письмових та графічних зображень на папері, комп'ютері та проекції. Світло також створює сцену для прослуховування, словесного спілкування, розвитку соціальних навичок та розуміння ситуацій. Як важливий елемент дизайну, який суттєво впливає на те, наскільки простір відповідає потребам студентів та викладачів, освітлення в класі повинно підтримувати здоров'я, добробут та ефективність, забезпечуючи комфортне, привабливе середовище для студентів та викладачів. Окрім підвищення рівня задоволеності мешканців та підтримки освітнього досвіду в освітленому просторі, освітлення в школах та навчальних закладах повинно забезпечуватися в рамках жорстких обмежень кодексу.

Навчальні заклади варіюються від початкових (початкових) шкіл, середніх шкіл, середніх шкіл до університетів та коледжів. Хоча ці заклади мають різні типи приміщень, спільним для всіх є те, що більшість навчальних та навчальних занять проходить у класах. Клас загального призначення має площу не менше 32 квадратних метрів (350 кв. Футів) і вміщує від 20 до 75 учнів. Типовий клас має прямокутний план підлоги, що забезпечує кращі оглядові площі, ніж квадратний. Навчальний простір розроблений з оглядовими лініями, паралельними вікнам, які забезпечують проникнення денного світла (мансардного вікна) у простір і дають сенсорну стимуляцію та візуальний контакт із зовнішнім світом. Контрольні носії, такі як штори або жалюзі, використовуються для зменшення зовнішньої освітленості, щоб вони знаходились в рівновазі з яскравістю інтер'єру, або для усунення денного світла, коли це не потрібно. Бічне освітлення з використанням денного світла через вікна забезпечує загальне освітлення більшої частини навчального дня. Однак штучне

освітлення відіграє ключову роль, коли потрібне збалансоване, послідовне та кероване візуальне середовище.

Планування класу, як правило, поділяється на зону для студентів та зону для вихователя. Студентська зона завжди вимагає загального освітлення, тоді як зона вихователя вимагає додаткового освітлення для забезпечення вертикальної освітленості на навчальних дошках та забезпечення гарного моделювання людських рис викладача. Найпоширенішим інструкторським інструментом у класах є навчальні дошки, до яких належать темно-сірі та зелені дошки (дошки) та дошки для сухого стирання, такі як дошки та сірі дошки. Відеоекрани для презентації проєктованих носіїв часто використовують для комп'ютерних інструктажів. Це вимагає мінімізації освітленості на проєкційному екрані, тоді як над студентською зоною повинно бути достатньо освітленого світла для конспектування. Клас може бути комп'ютеризованим середовищем, де головним завданням буде мінімізація відображення VDT - візуального дисплейного терміналу. Читаність на екрані може зменшитися завдяки відбитим зображенням, що створюються світильниками, вікнами та оточуючими поверхнями з високою яскравістю.

Освітлення в класі можна вважати якісним, якщо воно дозволяє студентам та викладачам точно та комфортне виконувати візуальні завдання. Основою дизайну освітлення є інтеграція потреб людини, архітектури, економіки та навколишнього середовища. Пріоритетом освітлення в класі є задоволення людських потреб, таких як видимість, виконання завдань, візуальний комфорт, соціальне спілкування, здоров'я, безпека та добробут. Ці різні потреби людини повинні бути належним чином збалансовані для формування стимулюючого навчального середовища, а також враховувати економічні, екологічні та архітектурні міркування. Досягнення якісного освітлення передбачає не лише належне освітлення, щоб зробити певне завдання видимим. Існує багато факторів, що впливають на здатність людей бачити і виконувати завдання, сім найважливіших - це відблиски, однорідність освітленості,

контраст яскравості, мерехтіння, кольоровий вигляд, моделювання обличч та предметів та завуальовані відбиття.

Висновки до розділу

Якісне освітлення є важливим аспектом нашого повсякденного життя і часто сприймається як належне. Контроль світла - це здатність регулювати рівень і якість світла в даному просторі для конкретних завдань або ситуацій. Правильний контроль за світлом не тільки покращує досвід, він допомагає економити енергію, використовуючи світло, коли і де воно найбільше потрібно.

Вимоги людини та якість робочого середовища часто виражаються термічним та візуальним комфортом. Оптимальні умови теплового комфорту можна легко охарактеризувати як нейтральне сприйняття внутрішнього середовища, коли мешканці не відчують потреби в змінах у бік теплих або холодних умов. Однак візуальний комфорт описується непросто. Замість посилення на стан нейтрального сприйняття внутрішнього середовища, воно сприймається як отримання повідомлення. Такі аспекти, як денне освітлення, відблиски, коефіцієнти освітленості, інтенсивність світла та контакт зовні, впливають на наше сприйняття візуального комфорту.

Елементи управління освітленням традиційно використовувались для створення настрою. Сьогодні вони також використовуються як частина високоякісної енергоефективної системи освітлення, яка об'єднує денне світло та електричні джерела світла, щоб забезпечити комфортне та візуально цікаве середовище для мешканців простору.

Управління електричним освітленням підходить для різноманітних приміщень, від туалетів до великих відкритих кабінетів, від конференц-залів до навчальних класів. Їх можна поєднувати з денним освітленням, щоб забезпечити гнучкість,

економію енергії та екологічні переваги. Хоча засоби управління освітленням все ще найчастіше використовують у комерційних будівлях, вони також все частіше використовуються в житлових приміщеннях.

Електричні регулятори освітлення використовуються в проектах освітлення для досягнення високоякісної енергоефективної системи освітлення. Визначення шаруватої, інтегрованої в денне світло системи освітлення та управління дає мешканцям контроль освітлення, забезпечуючи при цьому належні рівні освітлення, мінімізуючи відблиски, збалансовуючи яскравість поверхні та покращуючи навколишню архітектуру.

При правильному використанні елементів керування електричним освітленням буде збережено енергію, а термін служби ламп та баластів може бути продовжений.

Основною перевагою системи управління освітленням перед автономним управлінням освітленням чи звичайним ручним перемиканням є можливість управління окремими вогнями або групами вогнів з одного пристрою інтерфейсу користувача. Ця можливість контролювати кілька джерел світла від пристрою користувача дозволяє створювати складні сцени освітлення. У кімнаті може бути встановлено кілька сцен, кожна створена для різних видів діяльності в кімнаті. Основною перевагою систем управління освітленням є зниження споживання енергії. Більший термін служби лампи також досягається при затемненні та вимкненні світла, коли він не використовується.

2. ОСНОВНІ ПРИЦИПИ ПРОЕКТУВАННЯ СИСТЕМ КЕРУВАННЯ ОСВІТЛЕННЯМ

На освітлення припадає близько третини споживання електроенергії в комерційних будівлях. Хоча в останні роки економія енергії була досягнута завдяки встановленню більш ефективних джерел світла та світильників, управління світлом зараз є однією з найбільших можливостей для зменшення енергетичних витрат. Надійні, економічно вигідні пристрої управління доступні. Однак відсутність простих у використанні та об'єктивних вказівок щодо проектування перешкоджає застосуванню вдосконалених засобів управління.

Для полегшення проектування елементів керування освітленням, таких як датчики заповнення, таймери та фотосенсиори, необхідно створити більш надійні та зручні установки управління, збільшити використання елементів керування освітленням. Розглянемо основні властивості, які необхідні при проектування автоматизованих систем керування освітленням.

2.1 Рівномірність освітленості та контраст яскравості

Освітленість - це кількість світла, що падає на поверхню. Найпоширеніші зони в будівлі вимагають освітленості робочого простору в діапазоні від 150 lx до 250 lx. Рівномірне горизонтальне освітлення в зоні користувача усуває тіні, що впливають на видимість його основної роботи, і дозволяє гнучко використовувати простір під час зміни місця його розташування. У приміщеннях також дуже важлива вертикальна освітленість та освітленість на інших площинах між горизонтальною та вертикальною. Відношення мінімальної освітленості до середньої освітленості над поверхнею завдання, наприклад горизонтальна освітленість на робочих столах та вертикальна освітленість на інших предметах не повинна бути нижче 1: 1.4.

Яскравість - це кількість світла, що надходить від поверхні або точки. Це функція освітленості поверхні та відбиття поверхні, що означає, що яскравість може

бути збільшена за рахунок збільшення кількості світла, що потрапляє на поверхню предмета, або збільшення відбивної здатності поверхні [5]. Детальніше про це можна побачити на Рисунку 2.1.1:

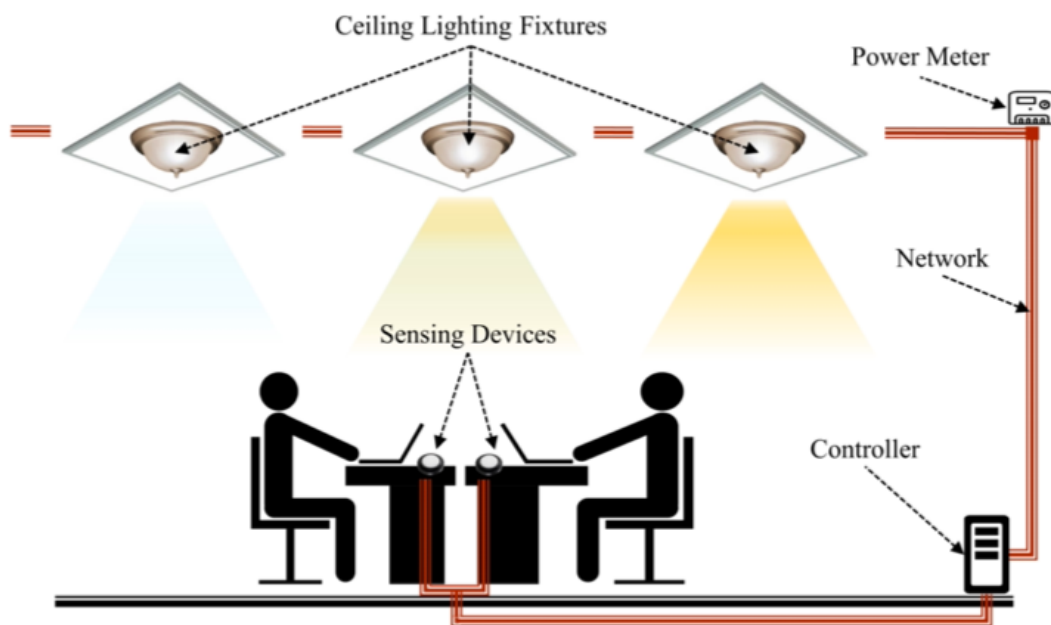


Рисунок 2.1.1 – Компоненти автоматизованої системи освітлення

Щоб підтримувати прийнятний контраст, відбивна здатність предмету повинна підтримуватися в межах від 5% до 20%. Для порівняння, наприклад, у зоні навчання, дощі потрібно 70% відбивної здатності, щоб привернути увагу. Відбивна здатність робочих поверхонь (робочих столів) повинна знаходитися в межах від 25% до 40%, щоб можна було досягти комфортного балансу яскравості. Стіни та стелі, як правило, мають світло-матове покриття. Вони створюють взаємовідбивання світла, які можуть забезпечити ефективне використання світла для покращеної горизонтальної та вертикальної освітленості, мінімізуючи відбиті відблиски. Людське око реагує на яскравість, а не на освітленість. Саме яскравість призводить до відчуття яскравості. На здатність бачити деталі сильно впливає зв'язок між яскравістю об'єкта та його безпосереднім фоном. Відповідний контраст між деталлю завдання та його фоном може створити візуальний інтерес та забезпечити візуальні сигнали. Однак занадто великі коливання яскравості створюють труднощі з адаптацією та візуальний

дискомфорт. Верхня межа коефіцієнта яскравості між завданням та безпосереднім оточенням становить 3: 1 (темніше оточення) або 1: 3 (світліше оточення).

2.2 Зовнішній вигляд кольору та відблиски

Колір є найважливішим елементом освітлення. Він має цілісний зв'язок зі світлом з точки зору візуального, емоційного та біологічного ефектів. Ступінь впливу на світло зорової продуктивності, настрою, атмосфери, здоров'я та самопочуття залежить від SPD світла (спектрального розподілу потужності), що випромінюється джерелом світла. Джерело світла можна охарактеризувати за кольоровою температурою та за показниками передачі кольорів, які обидва визначаються SPD. Зовнішній вигляд кольорів об'єктів є продуктом взаємодії між SPD джерела світла та спектральною функцією відбиття об'єктів. Деякі приміщення можуть вимагати освітлення, яке точно передає кольори. Передача кольору - лише один із аспектів освітлення. Важливіше поглянути на спектральний розподіл потужності світла та інтуїтивно зрозуміти, як колір світла впливатиме на поведінку, задоволеність, психологічні реакції та здоров'я. Колір джерел світла, будь то "теплий" чи "холодний" на вигляд, надзвичайно впливає на здоров'я людини, продуктивність та добробут.

Відблиски виникають, коли яскравість або коефіцієнт освітленості надмірно перевищують яскравість або коефіцієнт яскравості, до яких пристосовані очі. Наслідки відблисків включають інвалідність (зменшення видимості та зорових характеристик) та дискомфорт (неприємне відчуття яскравості, яке не обов'язково заважає зоровій діяльності або видимості).

Відблиски можуть бути наслідком потрапляння світла в око безпосередньо від джерела світла (прямий відблиск) або спричиненого відбиттями високої яскравості від відбивальної поверхні (відбиті відблиски). На Рисунку 2.2.1 можна це детально побачити:



Рисунок 2.2.1 – Прогнозування відблисків

Для прогнозування дискомфортих відблисків у внутрішніх приміщеннях верхнім освітлювальним приладам може бути призначений Єдиний рейтинг відблисків UGR або візуальна вірогідність комфорту VCP [6]. Максимальний UGR 19 або мінімальний VCP 70 вважається прийнятним для читання, письма та комп'ютерних завдань. Коли бажаний більш високий рівень візуального комфорту, слід вибирати світильники з UGR 16 або VCP 80.

2.3 Мерехтіння та моделювання предметів

Мерехтіння - це амплітудна модуляція світла, яка відволікає увагу і має ряд негативних наслідків. Як люмінесцентні, так і світлодіодні світильники, які працюють від неякісних джерел живлення, можуть виробляти частоту вдвічі більше, ніж частота електромережі (тобто 120 Гц або 100 Гц). Мерехтіння, як правило,

помітне на частотах вище 70 Гц. Однак мерехтіння, яке не помітне для людського ока, все одно може викликати реакцію нервової системи. Викликає занепокоєння як видиме, так і непомітне мерехтіння. Різний вплив від людини до людини, мерехтіння може спричинити перенапруження очей, нездужання, нудоту, зниження зорової працездатності, панічні атаки, головні болі, мігрень, епілептичні напади та свідчення обтяжуючих аутичних станів.

Наприклад, у навчальних закладах, де діти чи молодь перебувають протягом тривалого періоду щодня, слід здійснювати суворий контроль мерехтіння. Відсоток мерехтіння бажано не повинен перевищувати 4% при 120 Гц або 3% при 100 Гц, що надзвичайно безпечно для всіх груп населення. Максимально допустиме значення - 10% при 120 Гц або 8% при 100 Гц.

Відблиски - це плями з високою яскравістю (яскраві зображення джерела світла), що відбиваються на дзеркальних поверхнях, таких як екрани комп'ютерів або глянцеві матеріали для читання. Завуальовані відбиття від первинних джерел світла (вдови або світильники) або вторинних джерел світла (відбитих) зменшують контрастність завдання і затемнюють деталі. Щоб джерела світла не створювали дзеркального або дифузного відбиття в очах людини, треба встановити світильник із розподілом світла з мінімальним випромінюванням світла під проблемними кутами.

Моделювання обличчя та об'єктів є важливим фактором освітлення в закладах. Взаємодія світла і тіні на обличчі може допомогти спілкуванню користувачів, полегшуючи читання губ, а інтерпретацію жестів обличчя. Освітлення може додати форму та глибину візуальній сцені, розкрити текстуру та деталі об'єктів, створити бажаний візерунок та виділити яскраві моменти та візуальні інтереси. Сильне спрямоване освітлення може спричинити неприємні глибокі тіні, тоді як надзвичайно розсіяне освітлення робить обличчя або предмети плоскими або нецікавими. Тому бажана відповідна суміш спрямованого та розсіяного освітлення.

2.4 Проектування загального освітлення

Загальне освітлення є основним джерелом освітлення в приміщеннях. Він забезпечує простір загальним освітленням, одночасно слугуючи основним джерелом освітлення певної роботи людини. Загальне освітлення може бути здійснено за допомогою систем освітлення, встановлених на стелі з прямим, непрямим або комбінованим прямим / непрямим розподілом. Пряме освітлення забезпечує безперебійне світло від світильника до горизонтальної площини основних предметів. Непряме освітлення розподіляє світло до стелі, яке, в свою чергу, відбиває світло вниз. Пряме / непряме освітлення забезпечує розподіл світла як вниз, так і вгору. Системи прямого освітлення ефективно подають світло, але можуть створювати різкі тіні, завуальовані відблиски та небажані візуальні ефекти, такі як темні стелі та гребінці на верхніх поверхнях стін. При освітленні, спрямованому на стелі, системи непрямого освітлення рівномірно розподіляють світло до надмірно яскравого у полі зору. Однак непряме освітлення робить простір нудним і позбавленим відблисків та візуальних інтересів. Пряме / непряме освітлення поєднує в собі переваги прямого та непрямого освітлення, щоб забезпечити збалансоване розподілення світла для поліпшення візуального комфорту, рівномірного освітлення на горизонтальних робочих поверхнях та посилення вражень від простору, пильності та чіткості зору.

Незважаючи на стурбованість ефектом відблисків та ефектів печер, пряме освітлення є майже універсальним вибором у приміщеннях просто тому, що більшість з них мають низьку висоту стелі. Пряме освітлення, як правило, забезпечується у вигляді вбудованого освітлення, прихованого освітлення або підвісного освітлення. Світильники для прямого освітлення можуть бути розроблені різних форм і розмірів. У закладах загальноповиваними освітлювальними приладами є прямокутні трофери, призначені для установки в стельові стелі та лінійні освітлювальні прилади, призначені для вбудованих, поверхневих та прихованих установок. Трофери доступні у формі об'ємних, параболічних, дифузійних / лінзових, і світлодіодних панелей з підсвічуванням. Лінійні світильники поставляються в

секціях стандартної довжини, таких як секції 4, 8 або 12 футів, або в конфігурації безперервного пробігу.

Протягом останніх декількох десятиліть освітлення приміщень будівлі було майже ексклюзивною провінцією технології флуоресцентного освітлення. Флуоресцентна лампа використовує електрику для збудження парів ртуті в скляній трубці. Пари ртуті розряджаються для випромінювання ультрафіолетового світла, яке потім призводить до флуоресценції фосфорного покриття, виробляючи світло у видимому спектрі. Флуоресцентні лампи отримали широке застосування завдяки своїй високій світловіддачі, розсіяному розподілу світла та тривалому терміну експлуатації.

Однак використання люмінесцентних ламп є суперечливим. Флуоресцентні лампи мають багато недоліків, таких як ультрафіолетове випромінювання, тривалий час пуску, радіоперешкоди, висока крихкість, гармонічні спотворення, обмежений діапазон робочих температур та зменшення тривалості життя через часті перемикання. Тим не менше, найбільш негативним впливом флуоресцентного освітлення є те, що воно суттєво знизило якість внутрішнього освітлення та створило ризик для здоров'я. Надмірна увага до світлової ефективності призвела до того, що більшість люмінесцентних освітлювальних приладів погано виконували відтворення кольорів і забезпечували надмірно високу кольорову температуру (6000 K - 6500 K), що могло мати руйнівний вплив на циркадний ритм людини і викликало занепокоєння. небезпеки синього світла. Оскільки для люмінесцентної лампи необхідний баласт, щоб регулювати струм, що подається через електроди лампи, виникає проблема мерехтіння. Що стосується якості світла, флуоресцентне освітлення є особливо поганим початком в історії штучного освітлення для внутрішніх приміщень.

2.5 Світлодіодні світильники

Освітлення на основі світлодіодної технології LED швидко набирає популярності. Світлодіоди стали переважаючим джерелом світла для будь-якої уявної програми освітлення. Світлодіод - це напівпровідниковий пристрій, який перетворює електричну енергію безпосередньо в фотони. Напівпровідниковий пристрій має р-n-перехід, утворений протилежно легованими шарами напівпровідникового матеріалу, такого як нітрид галію індію (InGaN). Коли р-n-перехід зміщений у прямому напрямку, електрони та дірки вводяться в активну область і рекомбінують для генерації світла. Світлодіодна технологія усунула багато недоліків звичайних технологій і пропонує обіцянку високої ефективності, тривалого терміну служби, високої спектральної універсальності, виняткової керованості (увімкнення / вимкнення / затемнення), високої гнучкості в оптичному дизайні та високої стійкості до ударів та вібрацій. Світлодіоди видають потужність випромінювання лише у видимому спектрі (зазвичай від 400 до 700 нм). Відсутність ультрафіолетового випромінювання та інфрачервоного випромінювання робить цю технологію особливо придатною для використання людьми зі специфічною чутливістю або в ситуаціях, коли оптичне випромінювання від традиційних джерел світла може становити небезпеку для людини.

Тривалий термін служби та висока енергоефективність є головними перевагами світлодіодів. Це призводить до поширеної помилки, що тривалий термін експлуатації та висока світловіддача світлодіодних систем освітлення - це звичайно. Флуоресцентний світильник використовує набір ламп, наприклад, лінійний T5 (діаметр 5/8 дюйма), T8 (діаметр 1 дюйм) і T12 (діаметр 1 1/2 дюйма), стандартизований для всієї галузі та у виробників зі схожими термінами служби, світловіддачі та підтримка просвіту. Світильник в основному служить монтажною рамою для світильників і забезпечує обмежений контроль розподілу світла. На відміну від цього, світлодіодний світильник, як правило, є високо інженерною системою, яка цілісно інтегрує світлодіоди з тепловими, електричними та оптичними

підсистемами, щоб забезпечити прийнятний продукт. Ефективність системи та термін експлуатації світлодіодного світильника значною мірою залежать від конструкції та конструкції системи. Термін експлуатації світлодіодного світильника базується на тому, що світильник вперше вимагає технічного обслуговування, що, ймовірно, може бути пов'язано із знеціненням просвіту, зміною кольору, несправністю або навіть різкими несправностями світлодіодних драйверів.

Світлодіоди є найефективнішим джерелом світла, доступним сьогодні. Однак, як і раніше, більше половини електричної потужності, що подається на світлодіоди, перетворюється на тепло. На відміну від ламп розжарювання та галогенних ламп, які випромінюють тепло з ламп у вигляді інфрачервоної енергії, тепло, що генерується світлодіодами, потрапляє в напівпровідникові пакети і повинно розсіюватися через сам світильник. Надмір накопичення тепла в світлодіодах може прискорити процес деградації мікросхеми, люмінофора та пакувальних матеріалів. Показано, що підвищені температури переходів спричиняють багато механізмів руйнування, таких як зародження та зростання дислокацій в активній діодній ділянці, погіршення квантової ефективності фосфору та зміна кольору капсулюючих та пластикових корпусів. Отже, ефективне управління тепловою енергією має вирішальне значення для роботи світлодіодів відповідно до їх номінального терміну служби. Теплова конструкція є найважливішою частиною дизайну світильника. Всі матеріали та компоненти на тепловому шляху від напівпровідникової матриці через друковану плату до навколишнього середовища повинні мати низький тепловий опір. Ефективність теплової конструкції по суті залежить від здатності радіатора розсіювати тепло за допомогою теплопровідності та конвекції. Накладні світильники, такі як трофери та лінійні підвіски, зазвичай забезпечують достатній об'єм для створення достатньої площі поверхні, яка полегшує теплообмін.

Найчастіше точкою несправності або несправності в світлодіодній системі є світлодіодний драйвер. Оскільки світлодіоди чутливі навіть до дуже невеликих змін струму та напруги, схеми драйверів світлодіодів повинні бути налаштовані для

регулювання виходу при постійному струмі при напрузі живлення або змінах навантаження. Робота світлодіодів з належним струмом приводу також є частиною управління теплом. Перевищення того, що призначено для світлодіода, збільшить температуру переходу і зменшить внутрішню квантову ефективність світлодіодів. Основні показники продуктивності драйверів фокусуються на їх здатності регулювати потужність світлодіода або струни (або струни) світлодіодів належним чином та ефективно, забезпечуючи при цьому високий коефіцієнт потужності та низьке THD у широкому діапазоні вхідної напруги. Драйв повинен також забезпечити захисні функції від перевантаження, розімкнення та короткого замикання, а також перехідне придушення напруги та інтелектуальний захист від перегріву. Однак деякі виробники освітлення невпинно скорочують витрати, недопрацьовуючи схеми драйверів. Це не тільки призводить до порушення надійності схеми драйвера, але і робить проблему мерехтіння, оскільки недорогі драйвери часто забезпечують неповне придушення пульсацій. Як правило, неприпустимо, щоб значення пульсацій вихідного струму перевищувало $\pm 10\%$.

Оптичний дизайн стає головним пріоритетом у проектуванні світлодіодних систем. Рівномірне освітлення на великій площі або площині завдань вимагає використання великої кількості світлодіодів середньої потужності. Висока інтенсивність цих мініатюрних джерел світла робить пом'якшення відблисків пріоритетом. Світлодіодні світильники мають різноманітні характеристики розподілу, які досягаються за допомогою оптичних компонентів, таких як дифузори, лінзи, відбивачі та жалюзі. Пряме відблиск від світлодіодів можна пом'якшити, розсіюючи яскравість на великих поверхнях. Лінзи, що містять ряд невеликих призм, можуть зменшити освітленість світильника під кутами огляду поблизу горизонталі. Відображення - це широко застосовувана техніка регулювання світлового потоку від світлодіодів. Об'ємні трофери - це тип "відбитих прямих" світильників, які відбивають світло від внутрішньої поверхні вбудованого корпусу, тоді як світлодіодні модулі, що випромінюють світло вгору, екрановані або затемнені в металевих кошиках з дифузним акрилом. Світлодіодні підсвічування на кромці

світлодіодні інжектори вводять світло в світловодну пластину LGP, яка потім рівномірно розподіляє світло до дифузора через повне внутрішнє відбиття TIR. Здатність забезпечувати рівномірне освітлення, не створюючи надмірно яскравого освітлення, робить ці вбудовані світильники робочим конем у навчальних закладах.

2.6 Візуалізація кольорів

Як і у випадку з флуоресцентним освітленням, компроміс між якістю кольорів та світловою ефективністю залишився в еру світлодіодного освітлення. Білі світлодіоди, як правило, є світлодіодами з перетвореним фосфором, які використовують світло короткої хвилі, випромінюване від світлодіодних штампів, для накачування люмінофорів (люмінесцентних матеріалів). Більшість світлодіодів з перетвореним фосфором - це сині світлодіоди насоса, які частково перетворюють електролюмінесценцію. Синій світлодіодний насос високої кольоропередачі вимагає перетворення дуже великої частини випроміненого короткохвильового світла вниз. Цей процес перетворення світла насоса в люмінофор (фотолюмінесценція) передбачає велику кількість втрат енергії Стокса. Перетворення світлової ефективності випромінювання LER за чутливістю ока є неефективним у порівнянні зі спектральним розподілом світла довшої хвилі. Складаючи ці ефекти, світлова ефективність світлодіодів з високою кольоровою передачею, які мають SPD, більш рівномірно розподілений по всьому видимому спектру, порівняно низька, ніж світлодіоди з низьким рівнем кольоропередачі, які перенасичені на синій та зеленій довжинах хвиль.

В результаті спрямування на високоефективне освітлення та зниження витрат, більшість світлодіодних світильників, що використовуються в закладах, мають світлодіоди з CRI (індексом кольоропередачі) 80, що є прийнятним (але далеко не добрим). Зокрема, світло, що випромінюється від цих світильників, не має достатньої довжини хвилі, яка надає насичені кольори. Щоб у приміщенні було приємне відчуття і щоб кольори здавалися природними, джерело світла повинен спроможний

викликати візуальну реакцію на всі довжини хвиль у видимому спектрі. Наприклад, навчальні заклади заслуговують на освітлення з високою якістю кольорів, напр. CRI - 90. Хоча сині світлодіоди насоса можуть бути розроблені для забезпечення чудової передачі кольору, фіолетові світлодіоди насоса були розроблені спеціально для отримання білого світла широкого спектру, яке достатньо випромінює потужність у видимому спектрі.

ССТ (корельована температура кольору) джерела світла призначена для характеристики кольору світла (наприклад, теплого або прохолодного) [7]. Біле світло, що демонструє теплий тон, має ССТ в діапазоні від 2700 до 3200 К. Біле світло з ССТ в діапазоні від 3500 К до 4100 К зазвичай називають таким, що має "нейтральний білий" вигляд. Біле світло з ССТ вище 4100 К називається "холодним білим" зовнішнім виглядом. Не всі білі світлі рівні, незалежно від того, чи є біле світло теплим чи холодним, не тільки візуально впливає на наше сприйняття та емоційно впливає на наш настрій, але також впливає на цілий ряд нейроендокринних та нейро-поведінкових реакцій. Як правило, більш холодний білий колір відповідає відносно високому відсотку синього світла в спектрі, а теплий білий означає низький синій компонент у спектрі.

Дослідження встановили, що синє світло може стимулювати властиві до фото чутливих фоторецепторів гангліозних клітин сітківки (ipRGC) у шарі клітин сітківки. IpRGC перетворюють світло на нейронні сигнали для біологічного годинника. Потім біологічний годинник, розміщений у надхіазматичних ядрах (SCN), регулює температуру тіла та виділяє ендокринні гормони, такі як мелатонін та кортизол [8]. Досить висока доза біоактивного синього світла спричинить головний біологічний годинник для програмування людського тіла на денний режим. Виявлено, що вплив синього випромінювання стимулює вироблення таких гормонів, як кортизол, для реагування на стрес та настороженості; серотонін для контролю імпульсів та тяги до вуглеводів; і дофамін для задоволення, настороженості та координації м'язів. Моделюючи денну фізіологічну реакцію, вплив біоактивного синього світла також призводить до пригнічення гормону, що стимулює сон, мелатоніну. Оскільки він

підтримує концентрацію, пильність і ефективність, тому яскраве біле світло з високосиніми компонентами часто використовується протягом годин навчання.

Зазвичай для денного освітлення в освітніх приміщеннях вибирають холодне біле світло з ССТ близько 4100 К. Максимальний ССТ для внутрішнього освітлення загалом не повинен перевищувати 5400 К, що є видимою кольоровою температурою сонячного світла, що сяє прямо над головою. Однак запровадження флуоресцентного освітлення супроводжувало різке підвищення кольорових температур для внутрішнього освітлення. Джерела світла, що виробляють біле світло з довжинами хвиль, накопичених на блакитному кінці спектра, мають найбільшу світлову ефективність завдяки мінімальній фотолюмінесценції та чутливості очей у цій спектральній смузі. Це робить ССТ в діапазоні від 6000 К до 6500 К загальним вибором для освітнього освітлення [9].

Однак оптичне випромінювання з такою надзвичайно високою ССТ здається різким і часто спричиняє спотворення кольорів через відсутність довжин хвиль для надання насичених кольорів. Найголовніше, що вплив синього випромінювання в надзвичайно високій дозі протягом дня може призвести до перенапруження людського організму і ускладнює підтримку плавних добових ритмів.

3. ОПИС ВИКОРИСТАНИХ ПРОГРАМНИХ ЗАСОБІВ

Аналізуючи поставлену задачу, вирішено розроблювати програмне забезпечення у вигляді мобільного додатку. Це оптимальне рішення, тому що мета роботи передбачає застосування програми користувачами, які знаходяться в будівлі відділено від системи освітлення.

3.1 Засоби розробки

Логіка додатку та внутрішні методи реалізовані за допомогою мови програмування Dart. Середовищем розробки було обрано Android Studio. Для створення користувацького інтерфейсу та для зручності написання та побудови структури проекту було використано фреймворк Flutter.

Щодо серверної та апаратної частини були використані наступні елементи: Wi-fi та Bluetooth контролер esp32, світлодіодні стрічки, фоторезистори, годинник реального часу, макетна плата та інші допоміжні елементи. Усі функції з'єднання та функціонування елементів були реалізовані за допомогою мови програмування C у середовищі Arduino.

3.1.1 Мова програмування Dart

Dart - це оптимізована клієнтом мова для розробки швидких додатків на будь-якій платформі. Його мета - запропонувати найпродуктивнішу мову програмування для розробки мультиплатформ у поєднанні з гнучкою платформою виконання для фреймворків програм [10].

Мови визначаються їх технічною оболонкою - рішеннями, зробленими під час розвитку, що формують можливості та сильні сторони мови. Dart розроблений для технічного пакету, який особливо підходить для розробки клієнтів, надаючи пріоритет як розробці (допоміжна гаряча перезавантаження), так і високоякісному

виробничому досвіду для широкого спектру цілей компіляції (Інтернет, мобільні та настільні комп'ютери).

Мова Dart безпечна для друку; вона використовує перевірку статичного типу, щоб значення змінної завжди відповідало статичному типу змінної. Іноді це називають набором звуку. Незважаючи на те, що типи є обов'язковими, анотації типів є необов'язковими через умовивід типу. Система набору тексту Dart також гнучка, дозволяючи використовувати динамічний тип у поєднанні з перевітками виконання, що може бути корисним під час експериментів або для коду, який повинен бути особливо динамічним [11].

Dart також складає основу Flutter. Dart надає мову та час роботи, які працюють із програмами Flutter, але Dart також підтримує багато основних завдань розробника, таких як форматування, аналіз та тестування коду.

3.1.2 Фреймворк Flutter

Flutter - це міжплатформенний набір інтерфейсів, призначений для повторного використання коду в таких операційних системах, як iOS та Android, а також дозволяє додаткам взаємодіяти безпосередньо з базовими службами платформи. Мета полягає в тому, щоб дозволити розробникам доставляти високопродуктивні додатки, які почуваться природно на різних платформах, враховуючи відмінності там, де вони існують, спільно використовуючи якомога більше коду [12].

Під час розробки програми Flutter працюють у віртуальній машині, яка пропонує гаряче перезавантаження змін без необхідності повної перекомпіляції. Для випуску програми Flutter компілюються безпосередньо до машинного коду, будь то інструкції Intel x64 або ARM, або до JavaScript, якщо націлено на Інтернет. Фреймворк має відкритий код, має дозвільну ліцензію BSD і має процвітаючу екосистему сторонніх пакетів, які доповнюють основні функціональні можливості бібліотеки [13].

Програми Flutter упаковуються так само, як і будь-яка інша рідна програма. Специфічний для платформи вбудовувач забезпечує точку входу; узгоджується з

базовою операційною системою для доступу до таких послуг, як рендеринг поверхонь, доступність та введення; та керує циклом подій повідомлення. Embedder написаний мовою, яка підходить для платформи: в даний час Java та C ++ для Android, Objective-C / Objective-C ++ для iOS та macOS та C ++ для Windows та Linux. Використовуючи embedder, код Flutter може бути інтегрований до існуючої програми як модуль, або код може представляти собою весь вміст програми. Flutter включає ряд вбудованих програм для загальних цільових платформ, але існують і інші вбудовувальні програми [14].

3.1.3 Елементи Arduino

Контролер ESP32 - високоінтегрований, сумісний (Wi-Fi + Bluetooth) чіп, виконаний для рішення, що вимагає мінімальних показників енергопостачання. ESP32 розроблений для носіїв електроніки та доданого інтернету речовини, виконаний в супер-мініатюрному корпусі 6 x 6 мм, що вимагає інтеграції близько 10-ти зовнішніх компонентів. Він володіє хорошим функціоналом та багатообіцяючими можливостями [15]. Співвідношення в одному чипі Wi-Fi та Bluetooth, двома процесорними ядрами та багатого набору периферій може зробити ESP32 лідером у своєму сегменті. ESP32 пропонує повторно здійснити революцію у світі IoT, як у свій час, коли його молодший брат ESP8266. На рисунку 3.1.3.1 можна побачити, як виглядає цей контролер:

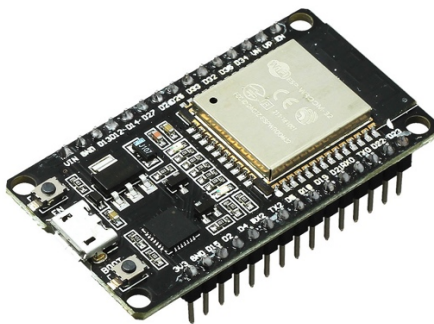


Рисунок 3.1.3.1 – Контролер ESP32

Годинник реального часу представляє собою невелику плату, що містить мікросхему DS1307. Крім цього, на платі практично можна знайти механізм установок батарейки живлення. Такі плати часто застосовуються для вивчення часу, дат, днів тижня та інших хронометричних параметрів. Модулі працюють від автономного харчування - батареї, акумулятори та продовжують проводити відлік, навіть якщо на Ардуїно відключено харчування. Вони засновані на підключеному модулі Arduino RTC [16]. Розглянути цю плату можна на Рисунку 3.1.3.2:

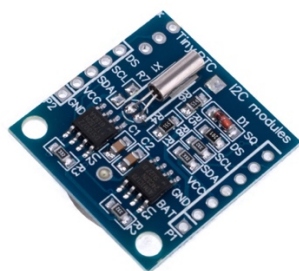


Рисунок 3.1.3.2 – Годинник реального часу

Світлодіодна стрічка RGB WS2812B - кожен піксель містить у собі червоний, зелений та синій світлодіоди та контролер ШИМ [17], за допомогою якого можна керувати яскравістю кожного світлодіоду та отримувати безліч різноманітних квітів з основних, наприклад Рисунок 3.1.3.3:



Рисунок 3.1.3.3 – Світлодіодна стрічка

Також використовувались багато допоміжних елементів. А саме, макетна плата, фоторезистори, дроти мама-мама, мама-тато, тато-тато і т.д.

3.2 Середовище розробки

Android Studio - середовище, що створює середню розробку програм для мобільних платформ Android. Переваги цього середовища наступні:

- Гнучкість середовища розробки;
- Більший набір функцій;
- Процес розробки, який підстроюється під розробника.

Коли створюється додаток та утиліт для операційної системи Android, програмне забезпечення користувача може спостерігати за змінами в проектах, в режимі реального часу.

Android Studio - універсальна розробка середовища, так як дозволяє оптимізувати роботу майбутніх додатків для роботи не тільки на смартфонах, але і на планшетах, і портативних ПК, що працюють на основі розглянутої операційної системи [18]. Середина розробки для додатків Android Studio останньої версії стала по суті зручною навіть для початківців розробників. У програмі реалізовані всі сучасні засоби для упаковки коду, його маркування. Затребувана множина творців ПЗ (програмного забезпечення) функція Drag-n-Drop, яка переносить компоненти у середу розробки безпосередньо.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Система була спроектована відповідно до основних вимог компонентного програмування. Кожний модуль системи відповідає за реалізацію чіткого та мінімального переліку функцій та взаємодії з іншими компонентами за допомогою гнучкого інтерфейсу.

Для роботи додатку необхідно мати сучасний телефон з Android або iOS операційною системою.

Шар представлення, реалізований у вигляді мобільного додатку, сполучається контролером із шаром сервісу, в якому можна відокремити самостійні елементи, відповідальні за керування мультизональним освітленням.

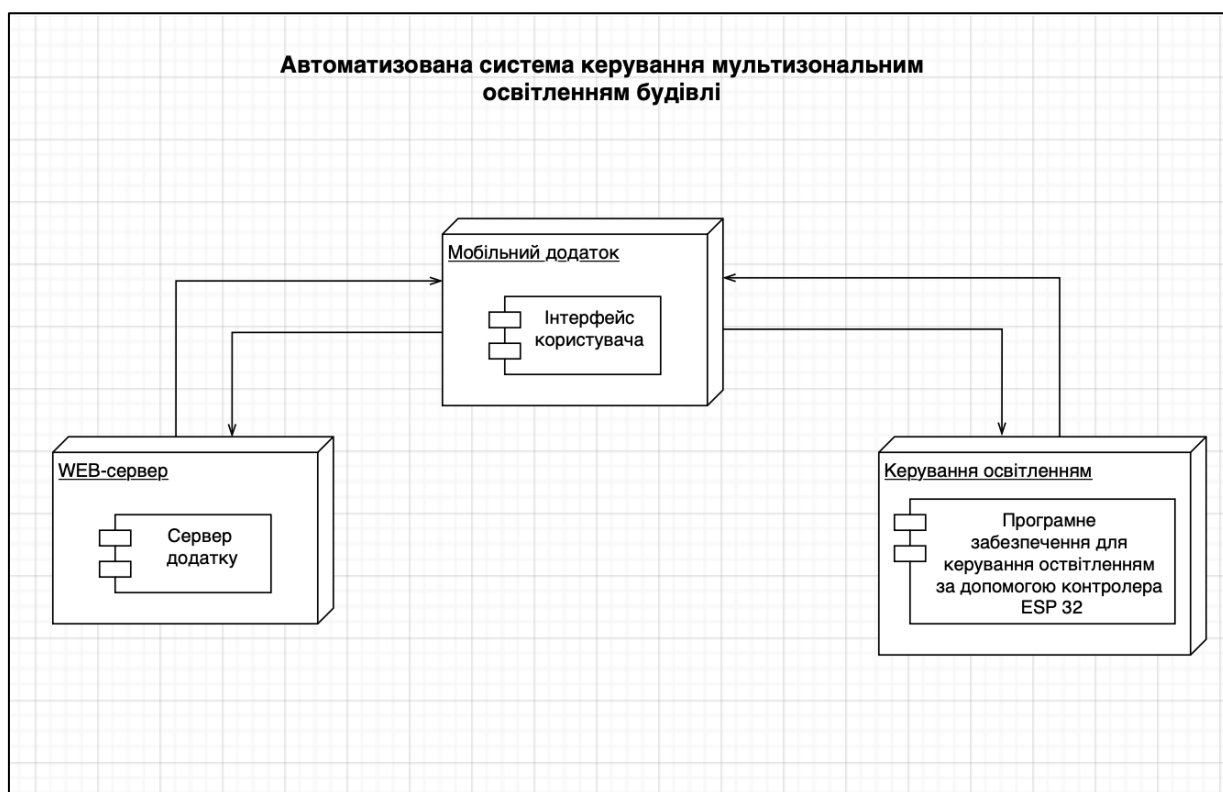


Рисунок 4.1 — Загальна архітектура створеної системи

На рисунку 4.1 зображена принципова архітектура системи .

4.1 Опис функціональності системи

Головний актор системи може виконувати дії трьох типів:

- контроль системи освітлення будівлі;
- корегування освітлення будівлі;
- налаштування певних функцій.

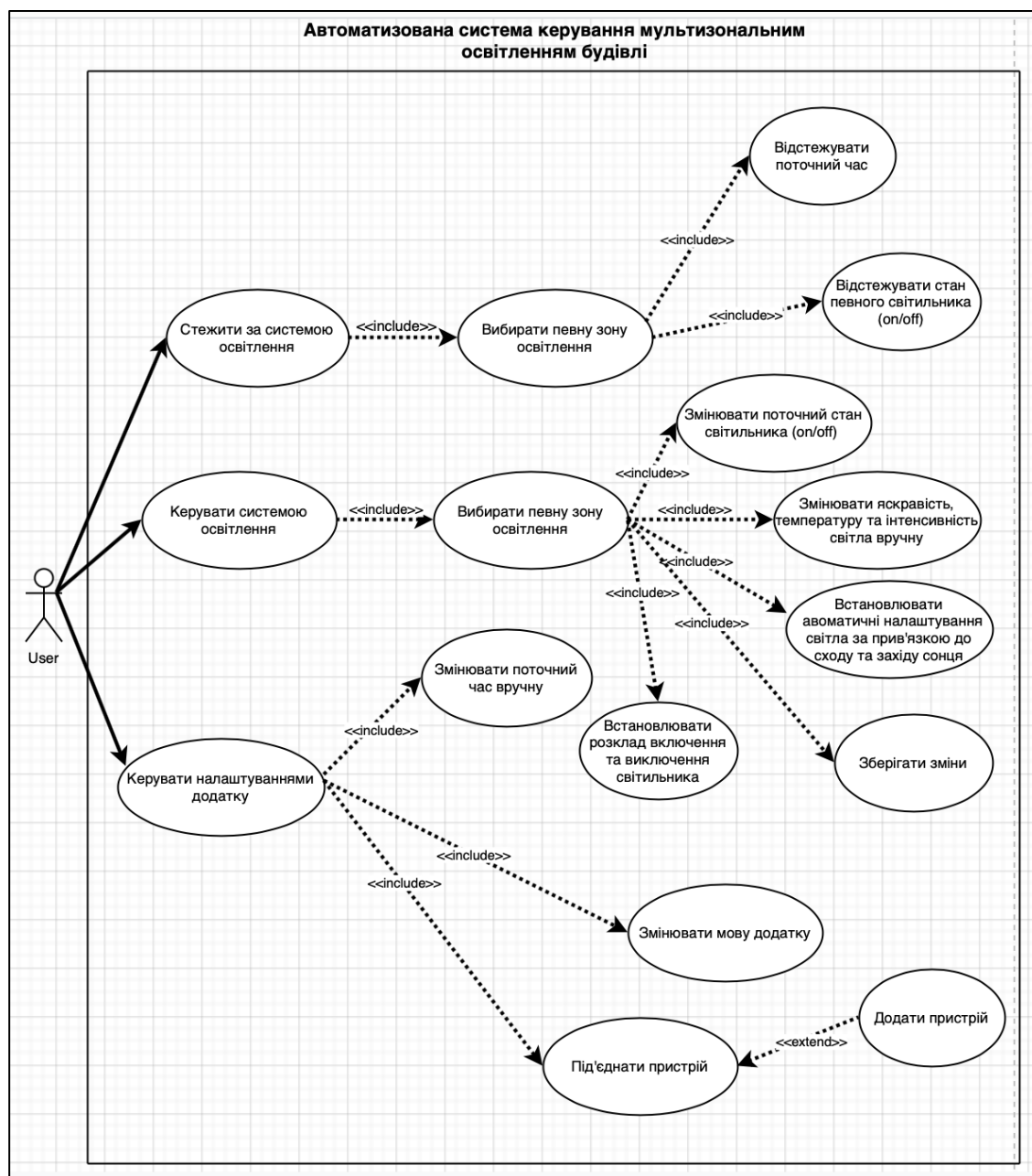


Рисунок 4.1.1 — Діаграма прецедентів системи

В залежності від вибору типу дії користувач в змозі виконувати наступне:

А) Вибирати певну зону, відстежувати поточний час та відстежувати стан певного світильника;

Б) Вибирати певну зону, змінювати поточний стан світильника, змінювати яскравість, температуру та інтенсивність світла вручну, встановлювати автоматичні налаштування світла за прив'язкою до сходу та заходу сонця, встановлювати розклад включення та виключення світильника і зберігати налаштування;

В) Змінювати поточний час вручну, змінювати мову додатку, під'єднати світлодіодний пристрій та додавати новий.

Діаграма прецедентів, що зображена на рисунку 4.2, описує набір дій, які створюють цілісний та повний функціональний сценарій роботи користувача зі системою.

4.2 Структура інформаційної бізнес–моделі

Головною складовою системи, де реалізовані всі сервіси, які вона може надавати користувачам, є її бізнес–модель [19]. Дана частина системи реалізує функції:

- Керування різними складовими: температурою, яскравістю та інтенсивністю світла;
- Моніторинг поточного стану зон освітлення;
- Налаштування автоматичного включення або виключення світильників.

Дані про користувача зберігаються у класі, що містить такі поля даних: особисті дані користувача, пароль персонального кабінету, зона керування світлом.

Розглянемо Рисунок 4.2.1:

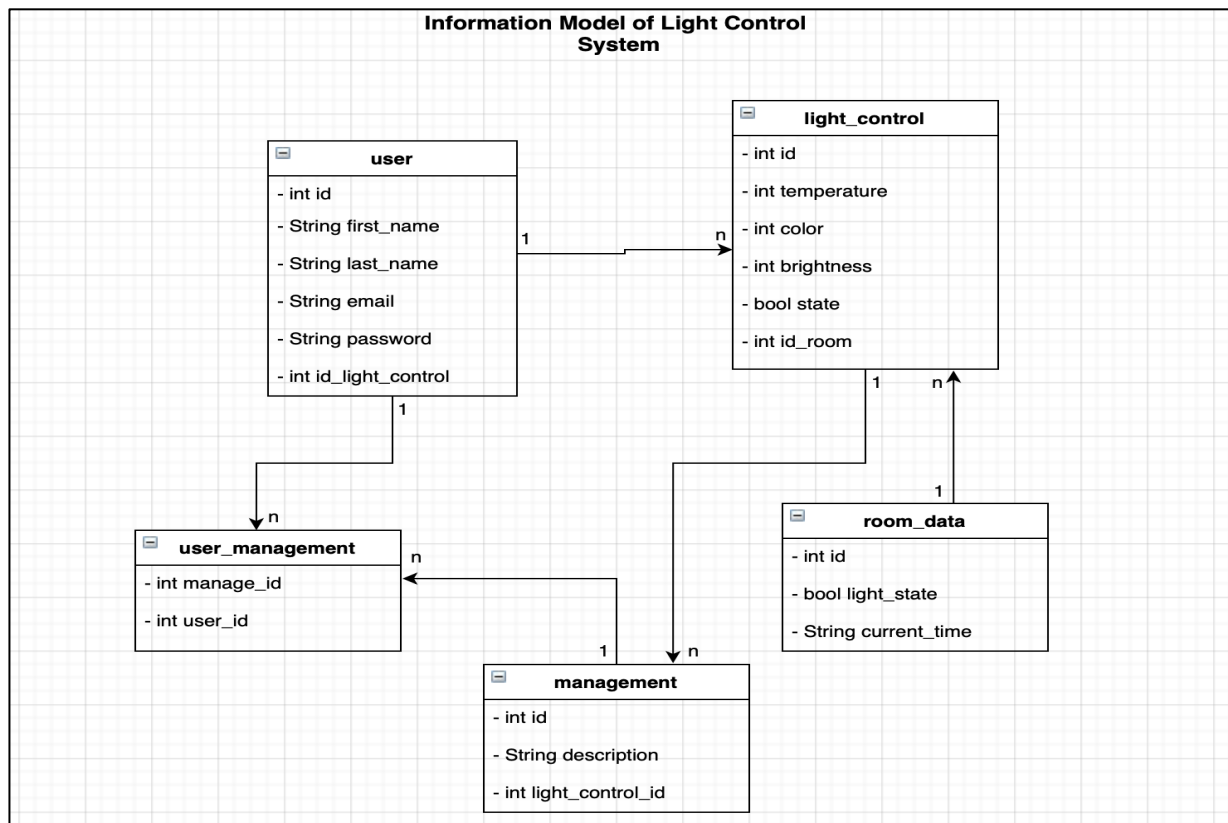


Рисунок 4.2.1— Інформаційна модель системи

Щодо об'єкта світла, то кожна світлодіодна стрічка має: значення температури, певний колір, інтенсивність, зона знаходження. Взаємозв'язок класів можна побачити на діаграмі вище.

4.3 Структура діаграми компонентів

Діаграма компонентів, також відома як діаграма компонентів UML, описує організацію та з'єднання фізичних компонентів у системі. Діаграми компонентів часто складаються, щоб допомогти деталям реалізації моделі та переконатися, що кожен аспект необхідних функцій системи [20].

Компонент є логічним блоком системи, трохи вищою абстракцією, ніж класи. Він представлений у вигляді прямокутника з вкладками або словом, написаним над назвою компонента, щоб допомогти відрізнити його від класу.

На Рисунку 4.3.1 – Діаграма компонентів системи, ми можемо почати нашу діаграму:

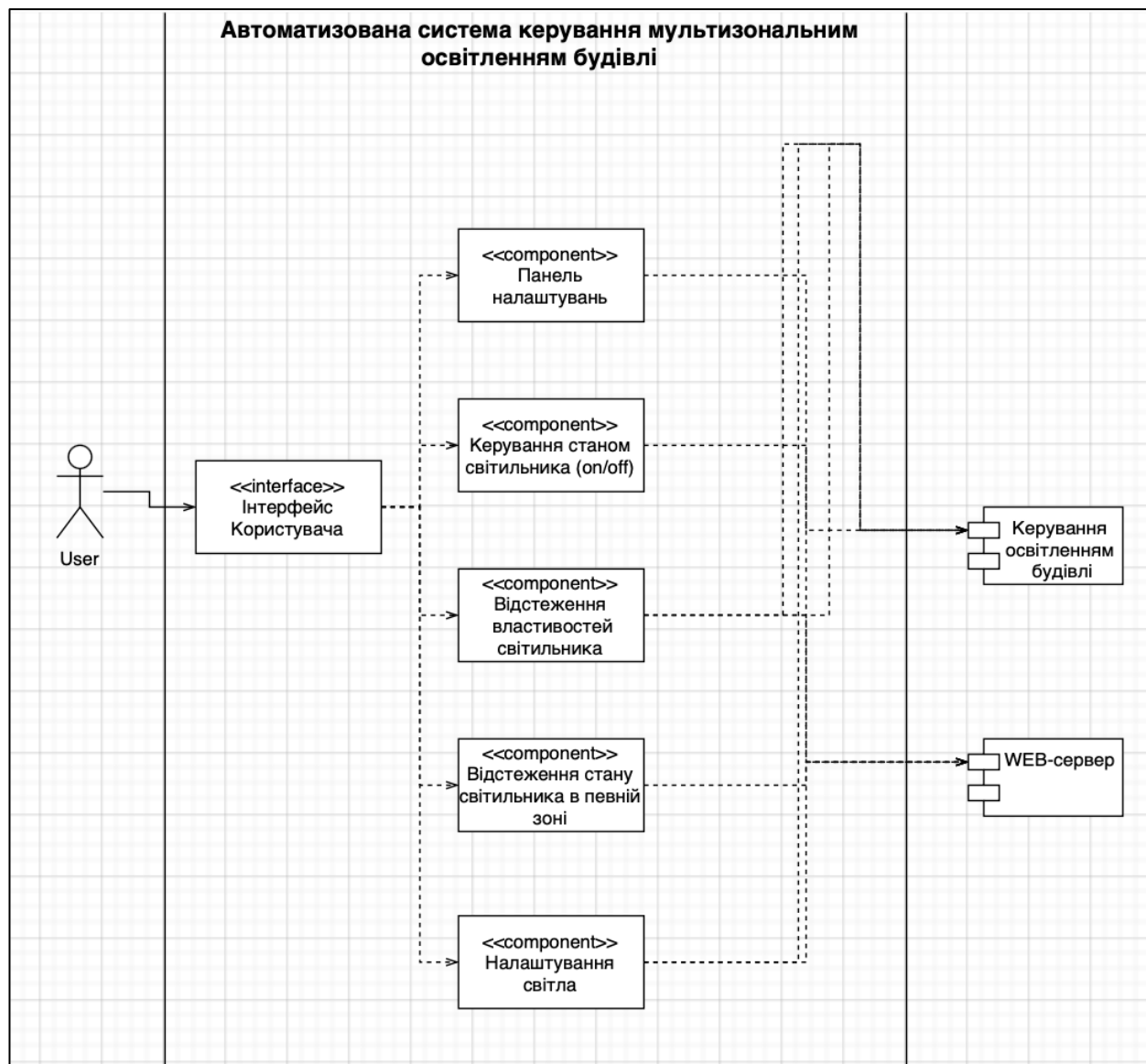


Рисунок 4.3.1 – Діаграма компонентів системи

Наш користувач за допомогою елемента інтерфейсу з'єднується з наступними компонентами: панель налаштувань, керування станом світильника, відстеження властивостей світильника, відстеження стану світильника в певній зоні та налаштування світла. Останні в свою чергу пов'язанні з елементом керування освітленням будівлі та елементом веб-серверу.

4.4 Опис UI flow діаграми

UI (User Interface) діаграми, їх ще називають діаграмами потоку інтерфейсу, є прекрасним засобом вивчення вашого користувацького інтерфейсу. Діаграми потоків користувацького інтерфейсу, або розкадровками, діаграмами потоків інтерфейсів, діаграмами навігації та картами контекстної навігації - дозволяють моделювати взаємозв'язки на високому рівні між основними елементами користувацького інтерфейсу і тим самим задавати основні питання зручності використання [21].

Діаграми потоку інтерфейсу користувача зазвичай використовуються для однієї з двох цілей. По-перше, вони використовуються для моделювання взаємодії користувачів із вашим програмним забезпеченням, як визначено в одному випадку використання. Наприклад, варіант використання може стосуватися декількох екранів і надає уявлення про те, як вони використовуються. На основі цієї інформації ви можете розробити діаграму потоку користувацького інтерфейсу, яка відображає поведінковий погляд на варіант одноразового використання [22].

На Рисунку 4.4.1 наводиться приклад UI flow діаграми нашої системи:

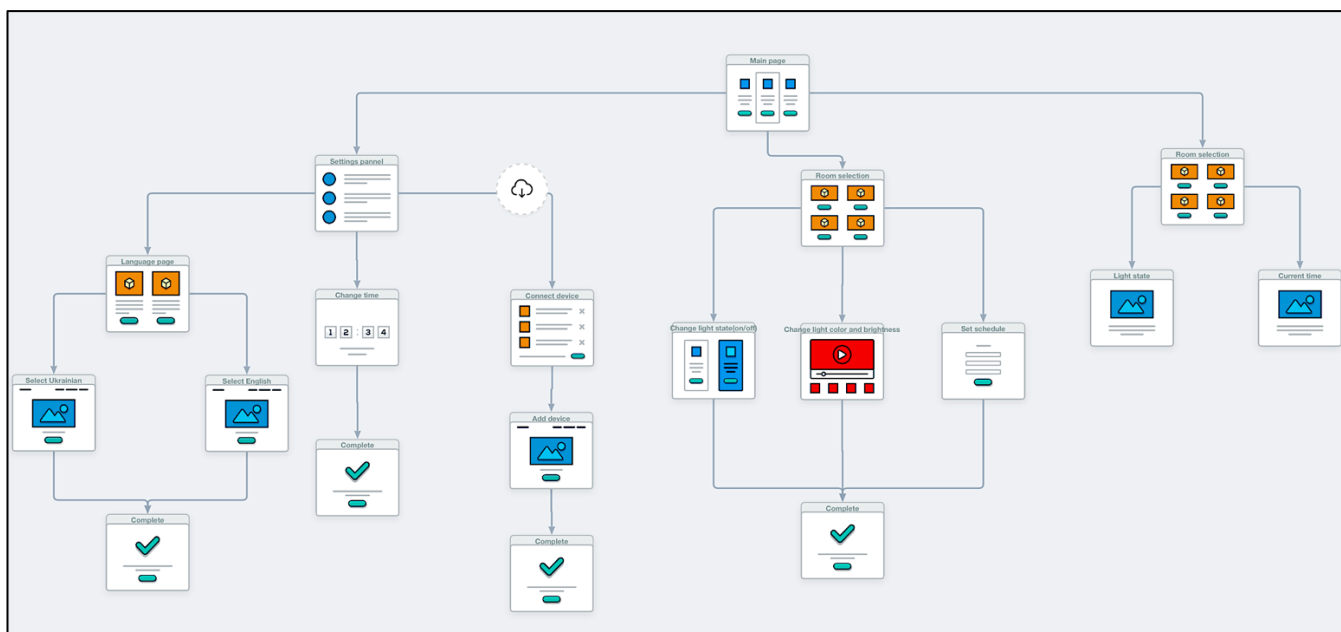


Рисунок 4.4.1 – UI flow діаграма

На цій діаграмі ми можемо побачити приблизний інтерфейс та взаємозв'язок наших усіх сторінок мобільного додатку, їх основні функції. На основі цього в подальшому розроблювалась верстка ПЗ.

4.5 Апаратна частина програмного забезпечення

За допомогою наступних елементів: Wi-fi та Bluetooth контролер esp32 – це наш головний компонент, світлодіодні стрічки, фоторезистори, годинник реального часу, макетна плата та інші допоміжні елементи, такі як дроти Мама-Мама, Мама-Тато, Тато-Тато та інші, - ми збираємо наступну схему:

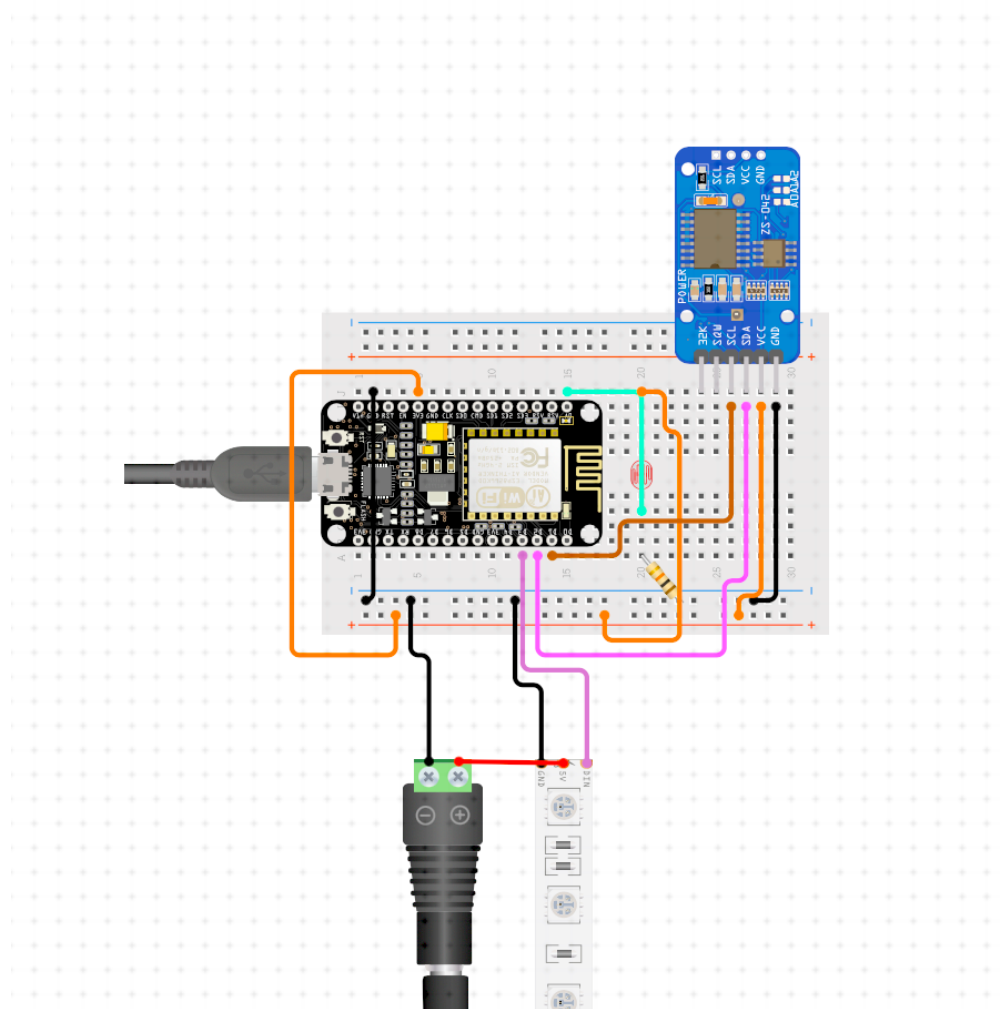


Рисунок 4.5.1 – Схема під'єднання світлодіодного пристрою

На Рисунку 4.5.2 та Рисунку 4.5.3 – Апаратна частина програмного забезпечення, можна побачити, як виглядає наша схема та всі її головні елементи наочно:

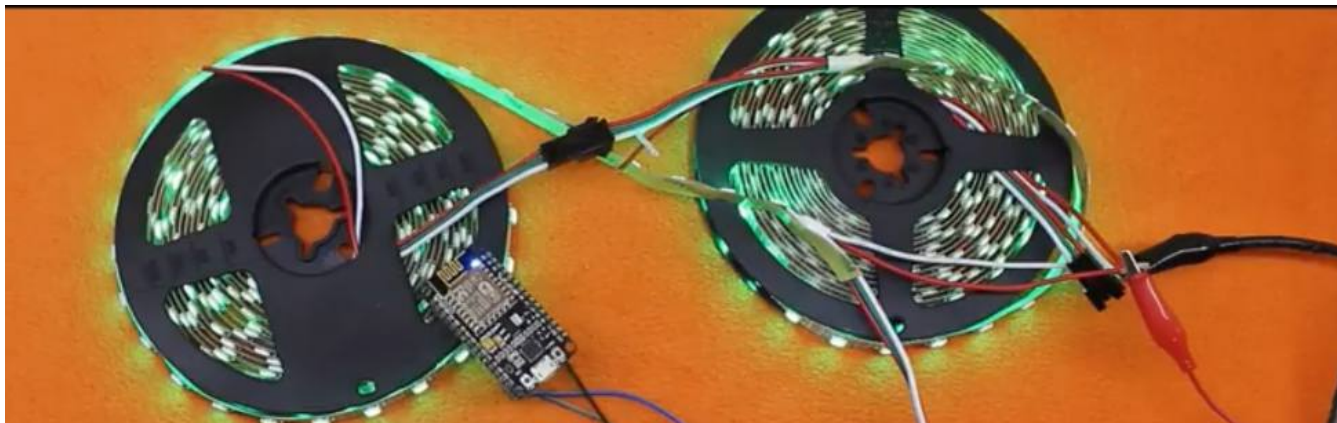


Рисунок 4.3.2 – Апаратна частина програмного забезпечення



Рисунок 4.5.3 – Апаратна частина програмного забезпечення

Тобто, ми маємо дві системи. З мобільного пристрою наш користувач відправляє команди на плату Arduino, Arduino обробляє ці команди і ми бачимо корегування світла наочно. Ми можемо спостерігати за зміною кольору, температури та яскравості нашого світлодіодного пристрою. Також, підлаштовувати його до певного часу доби, включати та виключати вручну і т.д. Наш мобільний застосунок написаний за допомогою Flutter, а Arduino на C. Системи спілкуються за допомогою Wi-Fi та Bluetooth.

4.6 Інструкція з використання програмного продукту

Коли користувач відкриває додаток, він потрапляє на головну сторінку, яка містить меню – налаштування та зони будівлі. Виглядає це наступним чином:

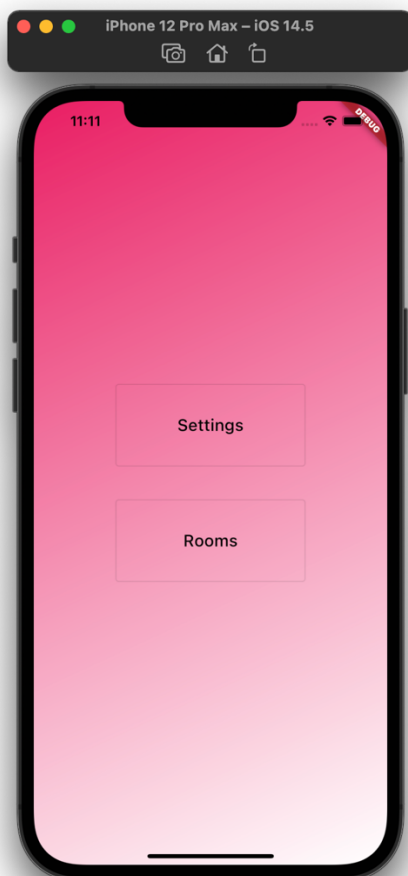


Рисунок 4.6.1 – Головний екран

Коли користувач переходить в налаштування, він може обрати мову додатку – це російська, англійська або українська, встановити поточний час доби вручну двома способами та під'єднати світлодіодний пристрій. Детально, як це відбувається можна побачити на Рисунку 4.6.2 – Меню налаштувань:

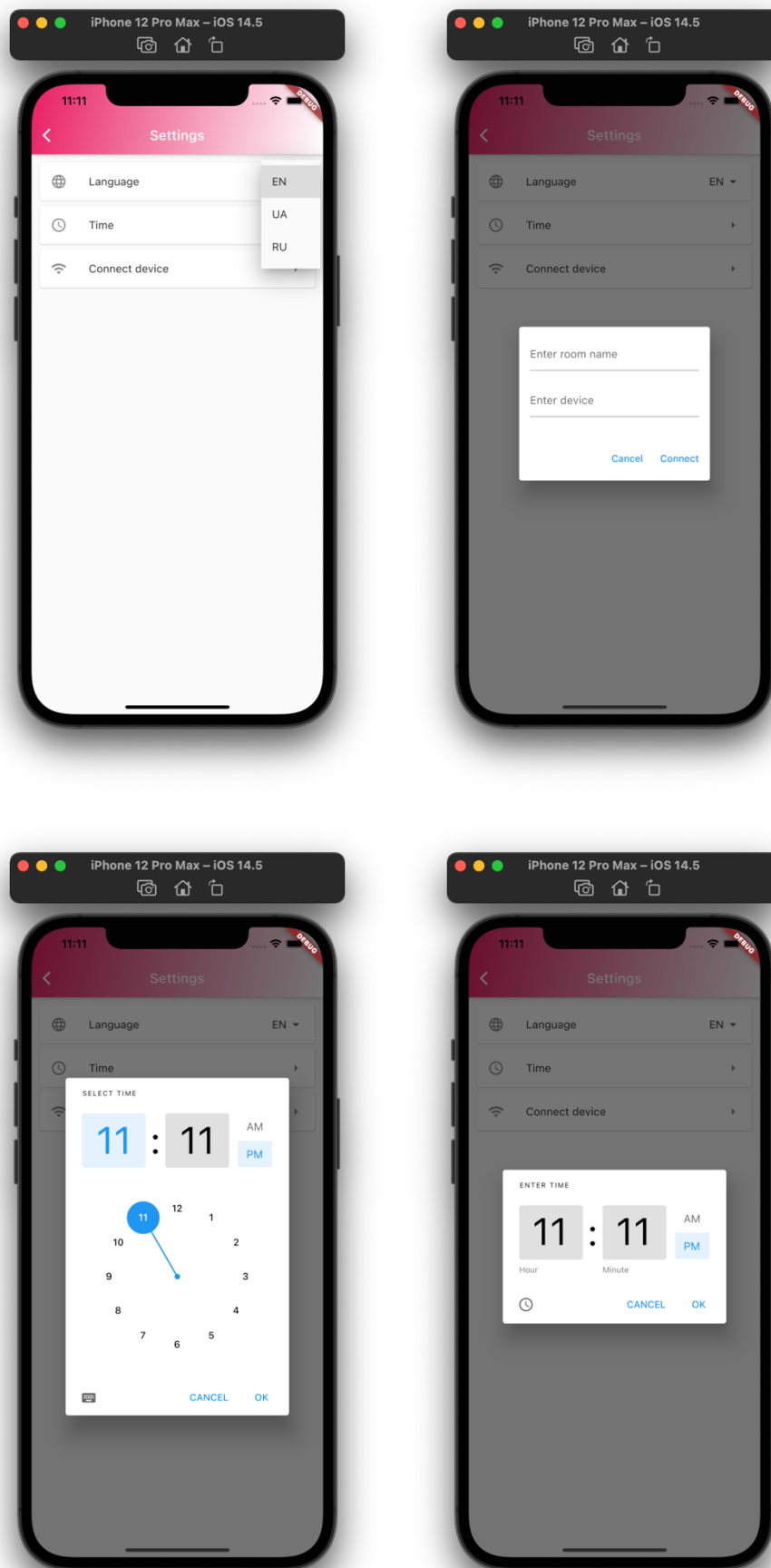


Рисунок 4.6.2 – Меню налаштувань

Повернемося на головну сторінку, звідки перейдемо до наших зон будівлі, користувач побачить поточний склад кімнат, в яких розташовані наші світлодіодні стрічки:

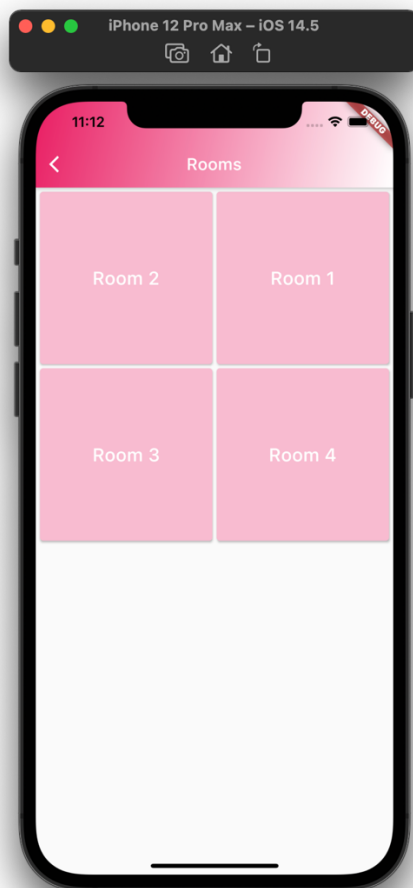


Рисунок 4.6.3 – Зони будівлі

Коли ми переходимо у певну зону, то можемо обрати певний світлодіодний пристрій, який нас цікавить та налаштувати яскравість, температуру або колір вручну. Крім цього ми можемо побачити поточний стан пристрою, увімкнений він або ні. В залежності який стан, ми можемо його змінити та світло буде адаптовано автоматично, дивлячись який зараз час доби. Також користувач може поставити таймер, коли потрібно світло увімкнути або навпаки. Детальніше це на Рисунку 4.3.4 – Налаштування світла:

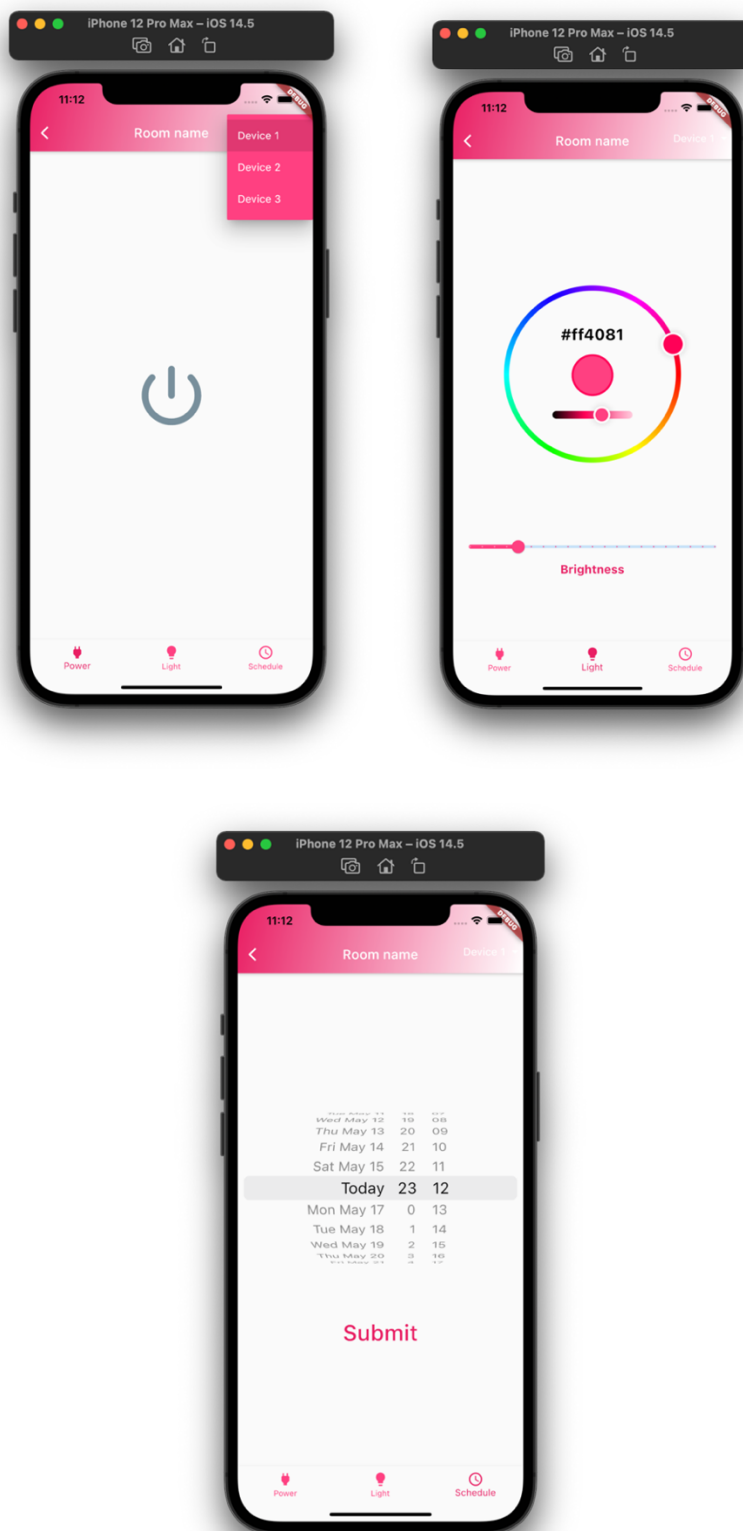


Рисунок 4.6.4 – Налаштування світла

З усіх сторінок ми можемо повернутися до головного меню і виконувати операції, які потрібні користувачу.

ВИСНОВКИ

У ході дипломної роботи була досліджена актуальність теми автоматизованих систем керування мультizonальним освітленням, на основі чого був розроблений певний програмний продукт – мобільний додаток. Він дозволяє ефективно автоматизувати процедуру корегування освітленням в певній будівлі.

На виході ми отримали програмне забезпечення, яке має наступні функції: повний контроль рівня освітленості в кожному приміщенні будівлі, управління інтенсивністю, температурою та яскравістю світла вручну, автоматична адаптація світла за допомогою прив'язки до часу доби та встановлення автоматичного включення або виключення за допомогою прив'язки до астрономічного сходу або заходу сонця.

Обрано якісні засоби розробки та надійну архітектуру, щоб система могла працювати з великою кількістю користувачів та при цьому зберігати криптографічну стійкість та безперебійність роботи.

Розроблений застосунок може бути використаний різними користувачами. Розроблена система показала високу ступінь ефективності та має широкий спектр потенційного використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision architectural elements and future directions", *Future Generation Computer Systems*, 2013. — 45 p.
2. Вступ до засобів управління освітленням [Електронний ресурс] – К.: Ділує, 2019. — 3 с. — Режим доступу до ресурса: <https://lightingcontrolsassociation.org/2017/07/21/introduction-to-lighting-controls>.
3. Neelam Verma, Anjali Jain. " Optimized Automatic Lighting Control in a Hotel Building for Energy Efficiency", *International Conference on Power Energy*, 2018. — 170 p.
4. H. Park et al. "Soummotulti-modal Illumimote: Multi-modal and high-fidelity light sensor module for wireless sensor networks", *IEEE Sensors Journal*, 2006. — 170 p.
5. Hajjaj, M., Miki, M., & Shimohara, K. "The Effect of Using the Intelligent Lighting System to Deduct the Power Consumption at the Office", *IEEE 7th Conference on Systems, Process and Control (ICSPPC)*, 2019. — 66 p.
6. Smart Wallbox Sensors [Електронний ресурс]. — 2020. — Режим доступу до ресурса: <https://www.leviton.com/en/products/brands/smart-sensors>.
7. Smart lightning for top class schools [Електронний ресурс]. — 2018. — Режим доступу до ресурса: <https://www.interact-lighting.com/global/what-is-possible/interact-pro/schools>.
8. Understanding Building Automation and Control Systems [Електронний ресурс]. — 2013. — Режим доступу до ресурсу: http://www.kmcccontrols.com/products/Understanding_Building_Automation_and_Control_Systems.aspx.
9. J. Burke et al. "Embedding expression: Pervasive computing architecture for art and entertainment", *Elsevier PMC Journal*, 2006. — 17 p.

10. Карабин, Петр Язык программирования Dart: Создание интерактивных приложений для Internet / Петр Карабин. — М.: Познавательная книга плюс, 2010. — 224 с.
11. T. Ball and S. Rajamani. "The SLAM Toolkit. In Proceedings of CAV'2001 (13th Conference on Computer Aided Verification) ", volume 2102 of Lecture Notes in Computer Science, Paris, 2001. — 260—264 pp.
12. Flutter Documentation [Электронный ресурс]. — 2020. — Режим доступа до ресурса: <https://flutter.dev/docs>.
13. Konstantin Ignatiev, Denis Grafov, Nikita Kuzmin. "Experience of Developing a Mobile Application Using Flutter", Information Science and Applications, 2019. — 572 p.
14. Payne R. "Beginning App Development with Flutter: Create Cross-Platform Mobile Apps / Rap Payne", Apress, 2019. — 334 p.
15. Michael McRoberts. "Beginning Arduino", Apress, 2010. — 50 p.
16. Huaguo J. "Study and Application of Encoding and Decoding Algorithms for Colored Two-dimensional Code on Mobile Terminals / Jia Huaguo ", Hangzhou, Zhejiang University of Technology, 2009. — 17 p.
17. David Kushner. "The Making of Arduino", IEEE Spectrum, 2015. — 20 p.
18. Adam Gerber, Clifton Craig. "Learn Android Studio. Build Android Apps quickly and effectively", Apress, 2015. — 129 p.
19. Сервис-ориентированная архитектура ИС [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: https://studopedia.su/10_104529_servis-orientirovannaya-arhitektura-is.html.
20. Zaccagnino C. "Programming Flutter: Native, Cross-Platform Apps the Easy Way (The Pragmatic Programmers) / Carmine Zaccagnino", Pragmatic Bookshelf, 2020. — 330 p.
21. Grade Booch, James Rumbaugh, Ivar Jacobson. "The Unified Modeling Language Reference Manual", Addison-Wesley, 2001. — 33 p.
22. Philippe Kruchten. "The Rational Unified Process -An Introduction", 2nd edition, Addison-Wesley, 2000. — 170 p.

ДОДАТОК А

Автоматизована система керування мультизональним освітленням будівлі

Специфікація

УКР.НТУУ “КПІ” _ТЕФ_АПЕПС_ТІ72127

Аркушів 2

Київ — 2021

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ ТІ72127	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ ТІ ТІ72127	arduino_part.ino	Компонент, що містить підключення апаратної частини та основні функції врегулювання освітлення
УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ ТІ ТІ72127	SettingsPage.dart	Компонент налаштувань мобільного додатку

ДОДАТОК Б

Автоматизована система керування мультизональним освітленням будівлі

Текст програми

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ТІ72127

Аркушів 13

Київ — 2021

arduino_part.ino

```
#include <ESP8266WiFi.h>

#define maxPMW 255 //Максимальный ШИМ

///

```



```

String parseB;
for (int i = pos1 + 1; i < pos2; i++) {
    parseB += header[i];
}
modeLED = parseB.toInt(); //После парсинга получаем режим
//Serial.println(parseB);
}
//GET /?value=180& HTTP/1.1
if (header.indexOf("GET /?value=") >= 0) { //Если пользователь потянул слайдер
    int pos1 = 0, pos2 = 0;
    pos1 = header.indexOf('=');
    pos2 = header.indexOf('&');

    String getH, getS, getV, getSp;

    for (int i = pos1 + 1, bb = 0; i < pos2; i++) {
        if (header[i] == '_') {
            bb++;
            continue;
        }
        switch (bb) {
            case 0:
                getH += header[i];
                break;
            case 1:
                getS += header[i];
                break;
            case 2:
                getV += header[i];
                break;
            case 3:
                getSp += header[i];
                break;
        }
    }

    //Serial.println(getH);
    //Serial.println(getS);
    //Serial.println(getV);
    //Serial.println(getSp);
    if (modeLED == 1) valueStringH = getH; //Если первый режим
    valueStringS = getS;
    valueStringV = getV;
    valueStringSp = getSp;
    HSVtoRGB(getH.toInt(), getS.toInt(), getV.toInt()); //Перевод в модель RGB
}
else { //Если пользователь написал что-то в адресную строку или только открыл
страницу
    if (true) { //Сделано блоком для удобства скрытия
        client.println("HTTP/1.1 200 OK");
        client.println("Content-type:text/html");
        client.println("Connection: close");
    }
}

```

```

// "Соединение: отключено"
client.println();
// показываем веб-страницу:
client.println("<!DOCTYPE html><html>");
client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");
client.println("<link rel=\"icon\" href=\"data:;\">");
client.println("<style>body { text-align: center; font-family: \"Trebuchet MS\", Arial;
margin-left:auto; margin-right:auto;}");
client.println(".inputH { background: linear-gradient(to right, hsl(0, 100%, 50%)
0%,hsl(14.4,100%,51%) 4%,hsl(28.8,100%,51%) 8%,hsl(43.2,100%,51%) 12%,hsl(57.6,100%,51%)
16%,hsl(72,100%,51%) 20%,hsl(86.4,100%,51%) 24%,hsl(100.8,100%,51%) 28%,hsl(115.2,100%,51%)
32%,hsl(129.6,100%,51%) 36%,hsl(144,100%,51%) 40%,hsl(158.4,100%,51%)
44%,hsl(172.8,100%,51%) 48%,hsl(187.2,100%,51%) 52%,hsl(201.6,100%,51%)
56%,hsl(216,100%,51%) 60%,hsl(230.4,100%,51%) 64%,hsl(244.8,100%,51%)
68%,hsl(259.2,100%,51%) 72%,hsl(273.6,100%,51%) 76%,hsl(288,100%,51%)
80%,hsl(302.4,100%,51%) 84%,hsl(316.8,100%,51%) 88%,hsl(331.2,100%,51%)
92%,hsl(345.6,100%,51%) 96%,hsl(360,100%,51%) 100%);}");
client.println(".inputS { background: linear-gradient(to right, #ffffff 0%, hsl(" +
valueStringH + ", 100%, 100%) 100%);}");
client.println(".inputV { background: linear-gradient(to right, #000000 0%, #ffffff
100%);}");
client.println(".inputSp { background: linear-gradient(to right, #ffbb00, #ff0000);}");
client.println(".divColor{ width: 50px;height: 50px;background: hsl(" + valueStringH + ",
100%, 50%);border-radius: 45px;margin: auto;}");
client.println("input[type=range] { -webkit-appearance: none; border-color: #000000;
border: 1px solid #000000;}");
client.println("input[type=range]::-webkit-slider-thumb {-webkit-appearance:
none;background-color: #ecf0f1;border: 1px solid #bdc3c7;width: 20px;height: 20px;border-radius:
10px;cursor: pointer;}");
client.println(".button { background-color: rgb(83, 1, 116); border: none; border-radius:
5%; color: white; padding: 15px 44px; text-align: center; text-decoration: none; display: inline-
block; font-size: 16px; transition-duration: 0.4s;}");
client.println(".button:hover {background-color: rgb(114, 1, 207); color: white;}");
client.println(".button:active { background-color: #5f00cc; box-shadow: 0 5px #666;
transform: translateY(4px); }");
client.println(".slider { width: 300px; border-radius: 40px; } ");
client.println("p { display: block; margin: 5px;}");
client.println("html { background-color: #0d1117; color: rgb(155, 2, 216);}");
client.println("span{ color:#9538ff; font-family: 'Franklin Gothic Medium', 'Arial
Narrow', Arial, sans-serif;}");
client.println("h1{ font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman',
serif;}");
client.println("</style>");

// веб-страница:
client.println("</head> <body><h1>LED WIFI </h1 > ");
client.println("<div class = \"divColor\"></div>");
client.println("<p>Hue: <span id=\"sliderH\">\" + valueStringH + "</p>");

```

```

        client.println("<p><input type=\"range\" min=\"0\" max=\"360\" class=\"slider inputH\"
id=\"HSlider\" onchange=\"SliderH(this.value)\" value=\"\" + valueStringH + "\"/></p>");
        client.println("<p>Saturation: <span id=\"sliderS\">\" + valueStringS + "</p>");
        client.println("<p><input type=\"range\" min=\"0\" max=\"100\" class=\"slider inputS\"
id=\"SSlider\" onchange=\"SliderS(this.value)\" value=\"\" + valueStringS + "\"/></p>");
        client.println("<p>Brightness: <span id=\"sliderV\">\" + valueStringV + "</p>");
        client.println("<p><input type=\"range\" min=\"0\" max=\"100\" class=\"slider inputV\"
id=\"VSlider\" onchange=\"SliderV(this.value)\" value=\"\" + valueStringV + "\"/></p>");
        client.println("<p>Speed: <span id=\"sliderSp\">\" + valueStringSp + "</p>");
        client.println("<p><input type=\"range\" min=\"0\" max=\"1000\" class=\"slider inputSp\"
id=\"SpSlider\" onchange=\"SliderSp(this.value)\" value=\"\" + valueStringSp + "\"/></p>");
        client.println("<p> <input type=\"button\" value=\"1\" class=\"button\"
onclick=\"buttonClick(this.value)\"> <input type=\"button\" value=\"2\" class=\"button\"
onclick=\"buttonClick(this.value)\"> <input type=\"button\" value=\"3\" class=\"button\"
onclick=\"buttonClick(this.value)\"> </p>");
        client.println("<p> <input type=\"button\" value=\"4\" class=\"button\"
onclick=\"buttonClick(this.value)\"> <input type=\"button\" value=\"5\" class=\"button\"
onclick=\"buttonClick(this.value)\"> <input type=\"button\" value=\"6\" class=\"button\"
onclick=\"buttonClick(this.value)\"> </p>");
        client.println("<p> <input type=\"button\" value=\"7\" class=\"button\"
onclick=\"buttonClick(this.value)\"> <input type=\"button\" value=\"8\" class=\"button\"
onclick=\"buttonClick(this.value)\"> <input type=\"button\" value=\"9\" class=\"button\"
onclick=\"buttonClick(this.value)\"> </p>");

        client.println("<script>var slideH = document.getElementById(\"HSlider\");");
        client.println("var slH = document.getElementById(\"sliderH\");");
        client.println("slH.innerHTML = slideH.value;");
        client.println("slideH.oninput = function() {slH.innerHTML = this.value; fetch(\"/?value=\" +
this.value + \"_\" + slideS.value + \"_\" + slideV.value + \"_\" + slideSp.value + \"&\");");
        client.println("document.getElementsByClassName('divColor')[0].style.background=\"hsl(\"+slideH.v
alue+\", 100%, 50%)\");");
        client.println("slideS.style.background = \"linear-gradient(to right, #ffffff 0%,
hsl(\"+slideH.value+\", 100%, 50%) 100%)\");");
        client.println("function SliderH(pos) {slH.innerHTML = pos; }");
        client.println("var slideS = document.getElementById(\"SSlider\");");
        client.println("var slS = document.getElementById(\"sliderS\");");
        client.println("slS.innerHTML = slideS.value;");
        client.println("slideS.oninput = function() {slS.innerHTML = this.value; fetch(\"/?value=\" +
slideH.value + \"_\" + this.value + \"_\" + slideV.value + \"_\" + slideSp.value + \"&\"); }");
        client.println("function SliderS(pos) {slS.innerHTML = pos; }");
        client.println("var slideV = document.getElementById(\"VSlider\");");
        client.println("var slV = document.getElementById(\"sliderV\");");
        client.println("slV.innerHTML = slideV.value;");
        client.println("slideV.oninput = function() {slV.innerHTML = this.value; fetch(\"/?value=\" +
slideH.value + \"_\" + slideS.value + \"_\" + this.value + \"_\" + slideSp.value + \"&\"); }");
        client.println("function SliderV(pos) {slV.innerHTML = pos; }");
        client.println("var slideSp = document.getElementById(\"SpSlider\");");
        client.println("var slSp = document.getElementById(\"sliderSp\");");
        client.println("slSp.innerHTML = slideSp.value;");

```

```

        client.println("slideSp.oninput = function() {slSp.innerHTML = this.value; fetch(\"/?value=\"
+ slideH.value + \"_\" + slideS.value + \"_\" + slideV.value + \"_\" + this.value + \"&\"); }");
        client.println("function SliderSp(pos) {slSp.innerHTML = pos; } ");
        client.println("function buttonClick(text){fetch(\"/?b=\" + text + \"&\");}");
        client.println("</script>");
        client.println("</body></html>");
        // конец HTTP-ответа задается
        // с помощью дополнительной пустой строки:
        client.println();
        // выходим из цикла while():
    }
}
break; //завершить цикл
}
}
}
header = ""; //Очистка строки передачи данных
//client.stop();
//Serial.println("Client disconnected.");
//Serial.println("");
}
}
void HSVtoRGB(int H, int S, int V) { //Перевод из HSV в RGB
    if (H == 360) {
        H = 0;
    }
    int Hi = (H / 60);
    int Vmin = ((100 - S) * V) / 100;
    int a = ((H % 60) * (V - Vmin)) / 60;
    int Vinc = Vmin + a;
    int Vdec = V - a;
    int r, g, b;
    r = g = b = 0;
    switch (Hi) {
        case 0:
            r = V;
            g = Vinc;
            b = Vmin;
            break;
        case 1:
            r = Vdec;
            g = V;
            b = Vmin;
            break;
        case 2:
            r = Vmin;
            g = V;
            b = Vinc;
            break;
        case 3:
            r = Vmin;
            g = Vdec;

```

```

    b = V;
    break;
case 4:
    r = Vinc;
    g = Vmin;
    b = V;
    break;
case 5:
    r = V;
    g = Vmin;
    b = Vdec;
    break;
}
//Serial.println("");
globalR = (r * maxPMW) / 100;
globalG = (g * maxPMW) / 100;
globalB = (b * maxPMW) / 100;
//Serial.print("R = ");
//Serial.println(globalR);
//Serial.print("G = ");
//Serial.println(globalG);
//Serial.print("B = ");
//Serial.println(globalB);
//Serial.println("");
}
void turnRGB(int r, int g, int b) { //Отображение на ленте цвета
    analogWrite(ledPinR, maxPMW - r);
    analogWrite(ledPinG, maxPMW - g);
    analogWrite(ledPinB, maxPMW - b);
}
void timerRunLed() {
    if (millis() > (runTimer + valueStringSp.toInt()/2)) {
        valueStringH = valueStringH.toInt() + 1;
        if (valueStringH == "361")valueStringH = "0";
        HSVtoRGB(valueStringH.toInt(), valueStringS.toInt(), valueStringV.toInt());
        //Serial.println(valueStringH);
        runTimer = millis();
    }
}
void runningLights() {
    int curModeColor[] = {0, 120, 240};
    if (millis() > (runTimer + valueStringSp.toInt())) {
        runTimer = millis();
        modeCurrentColor++;
    }
    if(modeCurrentColor > 2)modeCurrentColor = 0;
    HSVtoRGB(curModeColor[modeCurrentColor], valueStringS.toInt(), valueStringV.toInt());
}
void police(){
    if(millis() < ((runTimer+valueStringSp.toInt()))){
        valueStringH = "0";
        valueStringV = "100";
    }
}

```

```

    }
    if(millis()>((runTimer+valueStringSp.toInt())) &&
millis()<((runTimer+valueStringSp.toInt())*1.5){
        valueStringH = "0";
        valueStringV = "0";
    }
    if(millis()>(runTimer+valueStringSp.toInt()*1.5 &&
millis()<(runTimer+valueStringSp.toInt()*2.5){
        valueStringH = "0";
        valueStringV = "100";
    }
    if(millis()>(runTimer+valueStringSp.toInt()*2.5 &&
millis()<(runTimer+valueStringSp.toInt()*3.5){
        valueStringH = "0";
        valueStringV = "0";
    }
    if(millis()>(runTimer+valueStringSp.toInt()*7 && millis()<(runTimer+valueStringSp.toInt()*8){
        valueStringH = "240";
        valueStringV = "100";
    }
    if(millis()>(runTimer+valueStringSp.toInt()*8 &&
millis()<(runTimer+valueStringSp.toInt()*8.5){
        valueStringH = "240";
        valueStringV = "0";
    }
    if(millis()>(runTimer+valueStringSp.toInt()*8.5 &&
millis()<(runTimer+valueStringSp.toInt()*9.5){
        valueStringH = "240";
        valueStringV = "100";
    }
    if(millis()>(runTimer+valueStringSp.toInt()*9.5 &&
millis()<(runTimer+valueStringSp.toInt()*10.5){
        valueStringH = "240";
        valueStringV = "0";
    }
    if(millis()>(runTimer+valueStringSp.toInt()*10.5)
        runTimer = millis();
        HSVtoRGB(valueStringH.toInt(), valueStringS.toInt(), valueStringV.toInt());
    }
    void runningLights2(){
        int curModeColor[] = {0, 60, 120,180,240,300};
        if (millis() > (runTimer + valueStringSp.toInt())) {
            runTimer = millis();
            modeCurrentColor++;
        }
        if(modeCurrentColor >5)modeCurrentColor = 0;
        HSVtoRGB(curModeColor[modeCurrentColor], valueStringS.toInt(), valueStringV.toInt());
    }
    void runningLights3(){
        if (millis() > (runTimer + valueStringSp.toInt())) {
            valueStringH = valueStringH.toInt() + 10;
            if (valueStringH > "361")valueStringH = "0";

```

SettingsPage.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:liza/entity/Device.dart';
import 'package:liza/entity/Room.dart';
import 'package:liza/service/SharedPreferences.dart';

class SettingsPage extends StatefulWidget {
  @override
  State<StatefulWidget> createState() => _SettingsPageState();
}

class _SettingsPageState extends State<SettingsPage> {
  List<String> languages = ['EN', 'UA', 'RU'];
  String currentLanguage;
  String _roomName;
  String _deviceName;

  @override
  void initState() {
    super.initState();
    currentLanguage = languages[0];
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Settings'),
        flexibleSpace: Container(
          decoration: BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topLeft,
              end: Alignment.bottomRight,
              colors: [Colors.pink, Colors.white])),
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView(
          children: [
            Card(
              child: ListTile(
                leading: Icon(Icons.language),
                title: Text('Language'),
                trailing: DropdownButton<String>(
                  value: currentLanguage,
                  icon: const Icon(Icons.arrow_drop_down),
                  iconSize: 24,
                  elevation: 16,
```

```

onChanged: (String newValue) {
  setState(() {
    currentLanguage = newValue;
  });
},
underline: Container(),
items:
  languages.map<DropdownMenuItem<String>>((String value) {
return DropdownMenuItem<String>(
  value: value,
  child: Text(value),
);
}).toList(),
),
),
),
TextButton(
  onPressed: () {
    // Navigator.of(context).pushNamed(Routes.TIME);
    showTimePicker(
      context: context,
      initialTime: TimeOfDay(
        hour: DateTime.now().hour,
        minute: DateTime.now().minute),
    );
  },
  child: Card(
    child: ListTile(
      leading: Icon(Icons.access_time),
      title: Text('Time'),
      trailing: Icon(Icons.arrow_right)),
    ),
  style: TextButton.styleFrom(
    padding: EdgeInsets.zero,
  )),
TextButton(
  onPressed: () {
    // Navigator.of(context).pushNamed(Routes.TIME);
    showDialog(
      context: context,
      builder: (context) {
        return AlertDialog(
          contentPadding: const EdgeInsets.all(16.0),
          content: Container(
            height: 130,
            child: Column(
              children: [
                TextFormField(
                  decoration: const InputDecoration(
                    hintText: "Enter room name",
                  ),
                  onChanged: (value) => _roomName = value,

```

```

    ),
    Padding(
      padding: EdgeInsets.symmetric(vertical: 10)),
    TextFormField(
      decoration: const InputDecoration(
        hintText: "Enter device",
      ),
      onChanged: (value) => _deviceName = value,
    ),
  ],
),
),
actions: <Widget>[
  new TextButton(
    child: const Text('Cancel'),
    onPressed: () {
      Navigator.pop(context);
    },
  ),
  new TextButton(
    child: const Text('Connect'),
    onPressed: () {
      save();
      Navigator.of(context).pop();
    },
  ),
],
);
},
);
},
child: Card(
  child: ListTile(
    leading: Icon(Icons.wifi),
    title: Text('Connect device'),
    trailing: Icon(Icons.arrow_right)),
  ),
  style: TextButton.styleFrom(
    padding: EdgeInsets.zero,
  ),
),
],
),
),
);
}

```

```

Future<void> save() async {
  var value = await SharedPreferencesService.getJsons("rooms");

```

```

  List<Room> rooms =
    value == null ? [] : value.map<Room>((e) => Room.fromJson(e)).toList();
  Room room = rooms.firstWhere((element) => element.name == _roomName,
    orElse: () => null);

```

```
if (room != null) {
    room.addDevice(Device()..name = _deviceName);
    rooms.removeWhere((element) => element.name == _roomName);
    rooms.add(room);
} else {
    room = Room()
        ..name = _roomName
        ..devices = [Device()..name = _deviceName];
    rooms.add(room);
}
SharedPreferencesService.putJsons(
    "rooms", rooms.map((e) => e.toJson()).toList());
}
```

ДОДАТОК В

Автоматизована система керування мультизональним освітленням будівлі

Опис програми

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ТІ72127

Аркушів 9

Київ — 2021

АНОТАЦІЯ

Додаток містить опис мобільного застосунку для автоматизованої системи керування мультizonальним освітленням будівлі за допомогою Wi-Fi та Bluetooth.

Головним призначенням мобільного додатку є врегулювання освітлення в приміщенні відповідно часу доби або вручну та створення комфортних умов освітлення для певного виду діяльності людей.

Мобільний застосунок розроблений мовою програмування Dart з використанням технології Flutter у середовищі Android Studio. Для апаратної частини програмного комплексу використовувались елементи Arduino.

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ.....	78
2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ	79
3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ	80
4. ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ	81
5. ВИКЛИК І ЗАВАНТАЖЕННЯ.....	82
6. ВХІДНІ ДАНІ ТА ВИХІДНІ ДАНІ	83

ЗАГАЛЬНІ ВІДОМОСТІ

У додатку міститься опис основних компонентів мобільного застосунку для автоматизованого керування освітленням в будівлі. Додаток Б містить програмний код цих компонентів.

Для роботи мобільного застосунку потрібно мати мобільний пристрій на платформі Android версії Nougat (7.1.1) і вище, або на платформі iOS версії 8 і вище. Також необхідно мати включеним Wi-Fi та Bluetooth.

Мобільний застосунок розроблений мовою програмування Dart з використанням технології Flutter у середовищі Android Studio. Для апаратної частини програмного комплексу використовувались елементи Arduino.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблений мобільний застосунок надає користувачу наступний функціонал:

1. Повний контроль рівня освітленості в кожному приміщенні будівлі;
2. Управління інтенсивністю, температурою та яскравістю світла вручну;
3. Автоматична адаптація світла за допомогою прив'язки до часу доби;
4. Встановлення автоматичного включення або виключення за допомогою прив'язки до астрономічного сходу або заходу сонця;
5. Можливість використання трьох мов: англійська, українська та російська.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Додаток був спроектований відповідно до основних вимог компонентного програмування. Кожний модуль системи відповідає за реалізацію чіткого та мінімального переліку функцій та взаємодії з іншими компонентами за допомогою гнучкого інтерфейсу.

Для роботи додатку необхідно мати сучасний телефон з Android або iOS операційною системою.

Шар представлення сполучається контролером із шаром сервісу, в якому можна відокремити самостійні елементи, відповідальні за керування мультизональним освітленням.

Програмне забезпечення складається з двох систем. З мобільного пристрою користувач відправляє команди на плату Arduino, Arduino обробляє ці команди і ми бачимо корегування світла наочно. Мобільний застосунок написаний за допомогою Flutter, а Arduino на C. Системи спілкуються за допомогою Wi-Fi та Bluetooth.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для роботи мобільного застосунку потрібно мати мобільний пристрій на платформі Android версії Nougat (7.1.1) і вище, або на платформі iOS версії 8 і вище. Також необхідно мати включеним Wi-Fi та Bluetooth.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Для завантаження мобільного застосунку на Android необхідно відкрити проект програмного забезпечення в Android Studio та запустити віртуальний пристрій, або під'єднати мобільний пристрій через шнур до комп'ютера. Для інсталяції застосунку на пристрій на платформі iOS потрібно відкрити та запустити програмний проект в середовищі розробки XCode.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для програми є інформація, яку користувач вводить в додатку: поточний час, обраний пристрій, мова додатку, розклад включення або виключення, а також до вхідних даних відноситься поточний стан світильника та всі його значення-властивості: інтенсивність, яскравість та температура світла.

Вихідними даними програмного модуля є зміни, які робить користувач протягом використання додатку. Наприклад, змінений поточний час або розклад, оновлена температура та інтенсивність і т.д.