

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматики
Кафедра автоматизації та інтелектуальних інформаційних технологій

Бакалаврська кваліфікаційна робота на тему:
ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС УПРАВЛІННЯ
ІНТЕЛЕКТУАЛЬНОЮ ПІДСВІТКОЮ ДЛЯ МОНІТОРУ

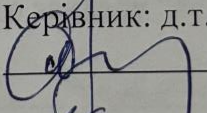
Виконав: студент 4 курсу, групи АКІТР-23мс
спеціальності 174 – Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка

(шифр і назва напрямку підготовки, спеціальності)


Олександр БАЗЕЛЮК

(прізвище та ініціали)

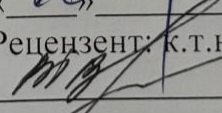
Керівник: д.т.н., професор, зав. кафедри АІТ


Олег БІСІКАЛО

(прізвище та ініціали)

« 16 » 06 2025 р.

Рецензент: к.т.н., доцент кафедри САІТ


Ілона ВАРЧУК

(прізвище та ініціали)

« 17 » 06 2025 р.


Допущено до захисту

Завідувач кафедри АІТ

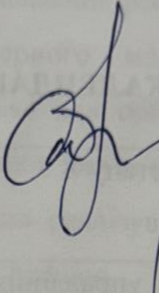
д.т.н., проф. Олег БІСІКАЛО

(прізвище та ініціали)

« 18 » 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 17 «Електроніка, автоматизація та електронні комунікації»
Спеціальність – 174 «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка»
Освітньо-професійна програма – «Інтелектуальні комп'ютерні
системи управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ
проф., д.т.н. Олег БІСІКАЛО
"14" 06 2025 року

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

БАЗЕЛЮКУ Олександр

1. Тема роботи: Програмно-апаратний комплекс управління інтелектуальною підсвіткою для монітору.

керівник роботи: Олег БІСІКАЛО, д.т.н., професор

затверджені наказом вищого навчального закладу "20" 03 2025 року №97

2. Строк подання студентом роботи 12 червня 2025 р

3. Вихідні дані до роботи:

кількість світлодіодів – 60шт.;

діапазон контролю зображення – ліва, права, верхня та нижня границі;

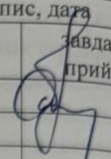
мова програмування – об'єктно-орієнтована

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

вступ; аналіз предметної області інтелектуальної mat-підсвітки комп'ютерного монітора; розробка апаратної частини програмно-апаратного комплексу управління інтелектуальною підсвіткою для монітору; розробка програмної частини програмно-апаратного комплексу управління інтелектуальною підсвіткою для моніторур; висновки; перелік джерел посилань

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
схема загального алгоритму функціонування комплексу; схема монтажна комплексу інтелектуальної підсвітки монітора; зовнішній вигляд готового комплексу.

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	БІСІКАЛО Олег, д.т.н., професор		

6. Дата видачі завдання 22.03.2025р.

КАЛЕНДАРНИЙ ПЛАН

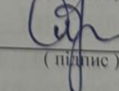
№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1.	аналіз предметної області управління інтелектуальною підсвіткою для монітору	02.10.2025	24.03.2025	Вик.
2.	розробка апаратної частини програмно-апаратного комплексу управління інтелектуальною підсвіткою для монітору	02.04.2025	31.04.2025	Вик.
3.	розробка програмної частини програмно-апаратного комплексу управління інтелектуальною підсвіткою для монітору	10.05.2025	23.05.2025	Вик.
4.	Розробка інструкції користувача.	24.05.2025	03.06.2025	Вик.
5.	Висновки та аналіз отриманих результатів	10.06.2025	15.06.2025	Вик.
6.	Оформлення матеріалів до захисту БКР	16.06.2025	23.06.2025	Вик.

Студент


(підпис)

Олександр БАЗЕЛ
(прізвище та ініціали)

Керівник проєкту (роботи)


(підпис)

Олег БІСІКАЛО
(прізвище та ініціали)

АНОТАЦІЯ

УДК 519.87

Бакалаврська кваліфікаційна робота складається з 69 сторінок формату А4, на яких є 43 рисунки, список використаних джерел містить 18 найменувань.

В даній бакалаврській кваліфікаційній роботі досліджено процес розробки інтелектуальної підсвітки комп'ютерного монітора. Проаналізовано сучасні системи-аналоги, що використовуються для організації підсвітки комп'ютерного монітора. Наведено переваги і недоліки існуючих аналогів. Досліджено способи вирішення даної задачі. Запропоновано реалізувати інтелектуальну підсвітку для комп'ютерного монітору на базі плати Arduino.

Результатом дипломного дослідження є пристрій, який виконує функцію інтелектуальної підсвітки комп'ютерного монітора, що реалізовано на платі Arduino. Програмна частина реалізована за допомогою мови Wiring.

Ключові слова: Arduino, підсвітка, монітор, Ambibox, Ambilight, Wiring.

ABSTRACT

Bachelor's qualification work consists of 77 pages of A4, which has 43 drawings, the list of sources used contains 18 names.

This bachelor's qualification has investigated the process of developing intellectual backlighting. Modern analog systems used to organize a computer monitor are analyzed. The advantages and disadvantages of existing analogues are given. The ways of solving this problem are investigated. It is proposed to implement the intellectual backlights of the computer monitor based on the Arduino board.

The result of a diploma research is a device that performs the function of intellectual illumination of a computer monitor implemented on the Arduino board. The software part is implemented with Wiring.

Keywords: Arduino, backlight, monitor, Ambibox, Ambilight, Wiring.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНОЮ ПІДСВІТКОЮ ДЛЯ МОНІТОРА	6
1.1 Дослідження аспектів світлодіодного освітлення	6
1.2 Аналіз основних способів розробки інтелектуальної підсвітки	7
1.3 Висновки до розділу 1	11
2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНОЮ ПІДСВІТКОЮ ДЛЯ МОНІТОРУ	12
2.1 Принцип роботи інтелектуальної підсвітки для монітора.....	12
2.2 Вимоги до обладнання та вибір елементів для інтелектуальної підсвітки комп'ютерного монітора.....	17
2.3 Розробка схеми пристрою інтелектуальної підсвітки монітора	23
2.4 Висновки до розділу 2.....	30
3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНОЮ ПІДСВІТКОЮ МОНІТОРА	31
3.1 Обґрунтування вибору середовища розробки	31
3.2 Установка середовища Arduino IDE	33
3.3 Вибір мови програмування.....	49
3.4 Програмування контролера Arduino.....	50
3.5 Налаштування програмного забезпечення AmbiBox.....	54
3.6 Висновки до розділу 3.....	57
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	59
ДОДАТОК А (обов'язковий) Ілюстративна частина.....	61
ДОДАТОК Б (обов'язковий) Лістинг основної функції програмного забезпечення.....	66
ДОДАТОК В (обов'язковий) Протокол перевірки кваліфікаційної роботи.....	69

ВСТУП

Актуальність даного дослідження визначається потребою створення комфортного освітлення зони, розташованої позаду комп'ютерного монітору. Оптимальний ефект досягається у випадку, коли кольорова гама, що відображається на екрані, дублюється позаду нього, висвітлюючи задню область за пристроєм [1].

Тривала робота за комп'ютером призводить до дискомфорту очей – зображення на моніторі стає розмитим, розрізнення кольорів погіршується, виникає відчуття "піску в очах". Проблеми із зоровим сприйняттям можуть виникати через некоректні налаштування монітора, невідповідну модель, надмірно велику діагональ екрана або недостатню відстань між користувачем та монітором.

Для зменшення загального навантаження на зір необхідно коректно налаштувати такі параметри [2]:

- яскравість і контрастність;
- роздільну здатність;
- частоту оновлення;
- відстань від очей до монітору.

Надмірна яскравість екрана спричиняє осліплюючий ефект. Аналогічно з контрастністю – інтенсивні кольори швидко викликають втому очей, тому рекомендується встановлювати середні значення яскравості та контрастності або використовувати автоматичне налаштування [2].

Використання "нативної" роздільної здатності екрана забезпечує чіткість зображення, що не перевантажує органи зору. Допускається тимчасове збільшення або зменшення стандартної роздільної здатності для ігор чи перегляду фільмів, проте згодом варто повернутися до стандартного режиму, який забезпечує оптимальну чіткість та якість зображення без надмірного навантаження на очі. Рекомендована відстань між екраном і користувачем становить 50-60 см (дорівнює відстані витягнутої руки). За умови поганої

видимості шрифту чи зображення з такої відстані, доцільніше збільшити розмір шрифту та масштабувати зображення [2].

Деякі пристрої пропонують різноманітні режими налаштування екрану: стандартний, динамічний, кіно тощо. Оптимальним вважається стандартний колірний профіль, який забезпечує помірні яскравість, контрастність та інтенсивність кольорів.

Обов'язковими елементами зорової гігієни є регулярні перерви під час роботи та правильно організоване освітлення.

Концепція "розширення" зображення за допомогою відповідного освітлення була розроблена компанією Philips. Система аналізує кольорову палітру кадру на екрані монітора чи телевізора та розсіює світло по всьому периметру задньої частини пристрою. Таким чином, поверхня стіни за корпусом телевізора динамічно освітлюється, створюючи світловий ореол, що підсилює інтенсивність зображення на екрані та візуально збільшує його розміри.

Застосування такого адаптивного фонового підсвічування нівелює межі телевізора, розширюючи зону перегляду. Адаптивне підсвічування зменшує зорове напруження, знижує ймовірність головного болю, запобігає ефекту сухості очей. Підсилюється ефект присутності при перегляді фільмів, особливо помітний під час трансляції спортивних змагань [3].

Важливо зазначити, що зображення з ТБ-тюнера обробляється на рівні операційної системи, тому зі звичайними телевізійними каналами функція підсвічування не активується. Підсвічування функціонує винятково в додатках ОС Android (HDTVideobox тощо), які не використовують захищений контент (Netflix). При інтеграції підсвічування з комп'ютерним монітором ці обмеження не застосовуються [4].

Метою даного бакалаврської кваліфікаційної роботи є автоматизація роботи інтелектуальної підсвітки комп'ютерного монітору.

Для реалізації поставленої мети необхідно виконати наступні завдання:

- провести аналіз сучасних технологічних принципів функціонування підсвітки моніторів;

- дослідити існуючі аналоги інтелектуальної підсвітки моніторів;
- спроектувати апаратну частину інтелектуальної підсвітки комп'ютерного монітору на базі Arduino;
- розробити програмне забезпечення для роботи інтелектуальної підсвітки комп'ютерного монітору на платформі Arduino;
- виготовити прототип інтелектуальної підсвітки комп'ютерного монітору;
- здійснити тестування створеної інтелектуальної підсвітки комп'ютерного монітору.

Об'єкт дослідження – процес розробки інтелектуальної підсвітки комп'ютерного монітора.

Предмет дослідження – програмні та апаратні засоби для створення інтелектуальної підсвітки комп'ютерного монітора.

Методи дослідження бакалаврської роботи базуються на застосуванні технологій конструювання електронних пристроїв з використанням мікроконтролера Arduino, а також методик програмування мікроконтролерів Arduino на мові C для програмної імплементації інтелектуальної підсвітки комп'ютерного монітору.

Практична цінність отриманих результатів – розроблено інтелектуальну підсвітку комп'ютерного монітору, яка функціонує як додаткове джерело освітлення та забезпечує візуальне збільшення діагоналі монітору.

Апробація результатів роботи. Результати роботи були апробовані на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» (м. Вінниця, Україна, 2025 р.) [1].

Публікації. За результатами досліджень опубліковано тези доповіді на науково-практичній конференції [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНОЮ ПІДСВІТКОЮ ДЛЯ МОНІТОРА

1.1 Дослідження аспектів світлодіодного освітлення

Світлодіодне освітлення вважається одним із найперспективніших напрямків у галузі технологій штучного освітлення. Цей інноваційний підхід ґрунтується на використанні світлодіодів як основних джерел світла. Еволюція світлодіодного освітлення безпосередньо корелює з технологічним прогресом самого світлодіода.

Інноваційні технології дозволили розробити так звані над'яскраві світлодіоди, спеціально призначені для систем штучного освітлення. Ці компоненти характеризуються підвищеною яскравістю та ефективністю, що робить їх оптимальними для застосування у різноманітних освітлювальних конфігураціях – від домашнього освітлення до великомасштабних промислових об'єктів.

Світлодіодні пристрої освітлення демонструють найбільший функціональний потенціал з позиції енергоефективності, енергозбереження та практичного використання [5]. Вони вирізняються виключною ефективністю енергоспоживання та забезпечують суттєву економію електроенергії порівняно з традиційними джерелами світла.

Вже сьогодні існують сфери застосування, де світлодіодним світильникам немає рівноцінних альтернатив. Зокрема, у промисловому виробництві, шоу-індустрії, розважальному секторі та рекламних послугах, де потрібні освітлювальні пристрої з можливістю регулювання кольорових параметрів. Під регулюванням кольору мається на увазі здатність миттєво (за кілька секунд) змінювати спектр випромінювання світильника відповідно до потреб користувача. Ця властивість швидкої модифікації кольору практично недосяжна не лише для традиційних ламп розжарювання, але й для сучасних люмінесцентних ламп.

У сучасному технологічному ландшафті простежується тенденція до

оснащення джерел світла різноманітними електронними керуючими та допоміжними компонентами. Електронні модулі, що використовуються в конструкції, як правило, забезпечують додаткову функціональність виробу, підвищують рівень автоматизації, покращують користувацький досвід, сприяють вдосконаленню дизайну та оптимізують енергоспоживання. В окремих випадках вони також забезпечують більш ефективне виконання основних функцій освітлювальних пристроїв.

Ця тенденція відображає еволюцію сучасних технологічних рішень, орієнтованих на оптимізацію функціональних можливостей та забезпечення покращеного користувацького досвіду. Завдяки інтеграції електронних компонентів, джерела світла набувають інтелектуальних властивостей та здатності до адаптації відповідно до вимог сучасного споживача.

1.2 Аналіз основних способів розробки інтелектуальної підсвітки

Телевізори та монітори пройшли стрімкий еволюційний шлях, результатом якого стали надзвичайно тонкі та легкі сучасні пристрої. Паралельно з цим їх функціональні можливості значно розширилися порівняно з нещодавнім минулим – сьогодні будь-який сучасний телевізор може легко підключатися до мережі Інтернет, надаючи користувачеві доступ до всіх переваг технології SMART-TV. Логічним наслідком розширення функціоналу телевізора стало ускладнення його внутрішньої архітектури.

Це означає, що при виявленні несправностей, ремонт сучасного телевізора становить значні труднощі, тому оптимальним рішенням є звернення до кваліфікованого спеціаліста. Проте нерідко користувачі, наважившись на самостійний ремонт, намагаються усунути несправності телевізора власними силами. Найчастіше такі спроби стосуються проблем із підсвічуванням екрана. Здавалося б, проста заміна діодів виявляється насправді трудомістким процесом.

Ми звикли оперувати таким параметром телевізора як роздільна здатність зображення – HD, Full HD або Ultra HD. Різні формати характеризуються різною

щільністю пікселів, і чим вища ця щільність, тим якісніше зображення. Хоча це твердження частково відповідає дійсності, якість та насиченість зображення на екрані сучасного телевізора значною мірою залежить від яскравості підсвічування дисплея [6].

Підвищення якості та яскравості підсвічування безпосередньо впливає на яскравість відтворюваного на телевізорі зображення. При якісному та інтенсивному підсвічуванні кольори виглядають більш насиченими та привабливими. Яскравість підсвічування екрана вимірюється в нітах – чим вище значення цього параметра, тим яскравішим буде підсвічування телевізора. Однак, незважаючи на критичну важливість цієї характеристики екрана, виробники зазвичай не акцентують увагу споживачів на даному показнику.

Цілком очевидно, що без належного підсвічування екрана неможливо отримати не лише якісне зображення, але й взагалі будь-яке видиме зображення на екрані. Відповідно, якщо система підсвічування виходить з ладу, зображення на екрані стає невидимим. Для діагностики такої несправності можна увімкнути телевізор, потім активувати ліхтарик на смартфоні та спрямувати його на екран – при несправному підсвічуванні буде помітне надзвичайно тьмяне, ледь розрізняване зображення.

Принцип дії підсвічування базується на комплексі світлодіодів, які функціонують синхронно, і, як правило, вихід з ладу навіть одного елемента призводить до припинення роботи всієї системи підсвічування (рис. 1.1).

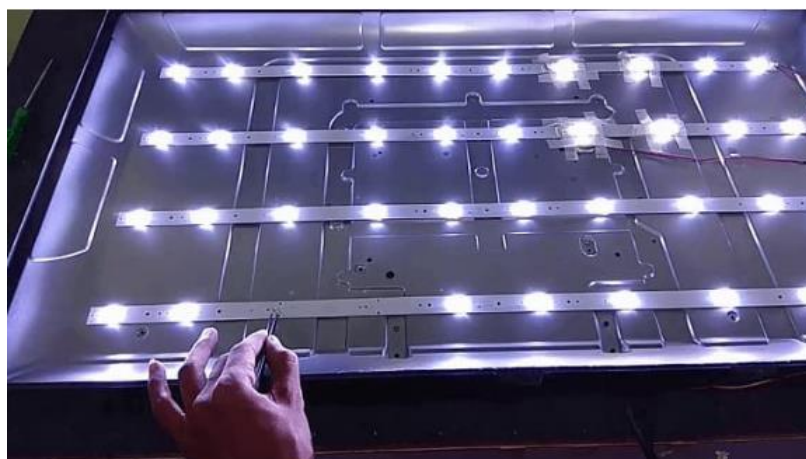


Рисунок 1.1 – Світлодіодна підсвітка матриці монітора чи телевізора

Однією з найбільш передових технологій, що суттєво покращує візуальне сприйняття та занурення під час перегляду відео в домашніх умовах, є система фонових освітлення. Сьогодні ця інноваційна розробка реалізована лише в обмеженій лінійці телевізорів бренду Philips і наразі не впроваджена в моделях інших виробників [7].

Метою даного дипломного проєкту є спроба реалізувати цю сучасну функцію в більш широкому колі телевізійних пристроїв, попри деякі технічні обмеження.

Ambilight (скорочення від англійського Ambient Lighting Technology – «Технологія атмосферного освітлення») – це запатентована розробка компанії Philips Electronics, яка полягає у створенні динамічного підсвічування навколо телевізійного екрана (рис. 1.2) [6].



Рисунок 1.2 – Приклад роботи технології Ambilight від компанії Philips

Ця система встановлена в РК-телевізори та працює за принципом аналізу кольорів поточного зображення на екрані. На основі цього аналізу відбувається проєкція м'якого світла навколо рамки пристрою, що візуально продовжує межі зображення на стіні позаду. Таке світлове обрамлення значно підсилює яскравість картинки і створює ефект занурення [8].

З часом Ambilight пройшла кілька стадій модернізації. Наприклад, версії Ambilight 2, Surround та Full Surround реалізують багатоканальне підсвічування,

яке гармонійно доповнює палітру та світлову насиченість екрана.

Ambilight Surround – третє покоління технології — включає додаткові світлодіодні елементи у верхній частині телевізора, завдяки чому зона підсвічування розширюється. Варто зазначити, що кожен світловий елемент функціонує незалежно.

Ambilight Full Surround – четверта і вдосконалена версія – оснащена освітлювальними компонентами по всьому периметру дисплея. Центральний блок керування, тобто процесор системи, аналізує зображення принаймні у чотирьох окремих ділянках екрана. У моделях з цією технологією реалізована спеціальна панель, що створює ілюзію, ніби телевізор «ширяє» в ореолі світла.

Подальшим кроком у розвитку цієї технології стала Ambilight Spectra – вона включає вдосконалені алгоритми аналізу відеозображення, а також модернізовані LED-елементи для ще більш точного підсвічування. Її впроваджено в серії телевізорів Philips під назвою Aurea [9].

Ключовими параметрами, якими керується система Ambilight, є колірна палітра, рівень яскравості, контрастність зображення, напрям та інтенсивність світлового потоку, а також динаміка його зміни у часі. Ці характеристики відіграють вирішальну роль у створенні максимально реалістичного, комфортного та захопливого візуального середовища під час перегляду контенту. Завдяки гнучкому керуванню зазначеними параметрами система може адаптувати світловий супровід до типу відео, жанру фільму чи навіть кольорової гами окремої сцени, що значно підсилює емоційне сприйняття.

Для досягнення оптимального візуального ефекту важливо правильно налаштувати систему ще на етапі встановлення. Зокрема, необхідно враховувати колір стіни, яка розташована безпосередньо позаду телевізора: світлі нейтральні кольори, такі як білий, сірий чи бежевий, краще відображають підсвічування і не спотворюють кольорову температуру Ambilight. У випадку, якщо стіна пофарбована у темні чи насичені кольори, рекомендується знизити яскравість або скористатися спеціальними налаштуваннями для компенсації впливу кольору фону.

Крім того, варто дотримуватись рекомендованої відстані між телевизором і стіною, яка, за даними виробника, має становити приблизно 25 см. Це забезпечує рівномірний розподіл світла по поверхні стіни та сприяє збереженню глибини візуального ефекту. Надто мала відстань може призвести до перенасичення світлом і втрати дифузії, а надто велика — до зменшення інтенсивності підсвітки та втрати ефекту «занурення». У сукупності всі ці аспекти налаштування дозволяють реалізувати повний потенціал технології Ambilight та забезпечити неперевершене враження від перегляду фільмів, телепередач чи ігор.

Окрім цього, система має додаткову функцію Lounge light, яку можна активувати в режимі очікування. Вона дозволяє використовувати підсвічування як м'яке фонове освітлення приміщення.

1.3 Висновки до розділу 1

У першому розділі було виконано огляд предметної галузі інтелектуального підсвічування моніторів. Розглянуто основні аргументи на користь впровадження такої функції, а також її переваги з точки зору користувачького досвіду.

Також проаналізовано наявні комерційні рішення для підсвітки області позаду монітора та визначено типові характеристики подібних систем.

На основі проведеного аналізу сформульовано технічні вимоги до майбутньої системи підсвічування монітора:

- вона повинна бути реалізована на основі апаратної платформи Arduino;
- необхідно використовувати доступні за вартістю апаратні компоненти;
- програмна частина має забезпечувати простий та інтуїтивно зрозумілий інтерфейс для користувача.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНОЮ ПІДСВІТКОЮ ДЛЯ МОНІТОРУ

2.1 Принцип роботи інтелектуальної підсвітки для монітора

У 2007 році компанія Philips отримала патент на досить просту, але водночас новаторську розробку – технологію фонового підсвічування під назвою Ambilight. Суть цієї технології полягає в адаптивному освітленні, яке не тільки знижує втомлюваність очей під час перегляду в умовах низького освітлення, а й створює ефект глибшого занурення у візуальний контент, візуально розширюючи межі зображення. Ambilight швидко знайшла застосування не лише для перегляду відео або фото, а й у сфері геймінгу. Відтоді вона стала впізнаваним елементом телевізорів Philips (рис. 2.1).

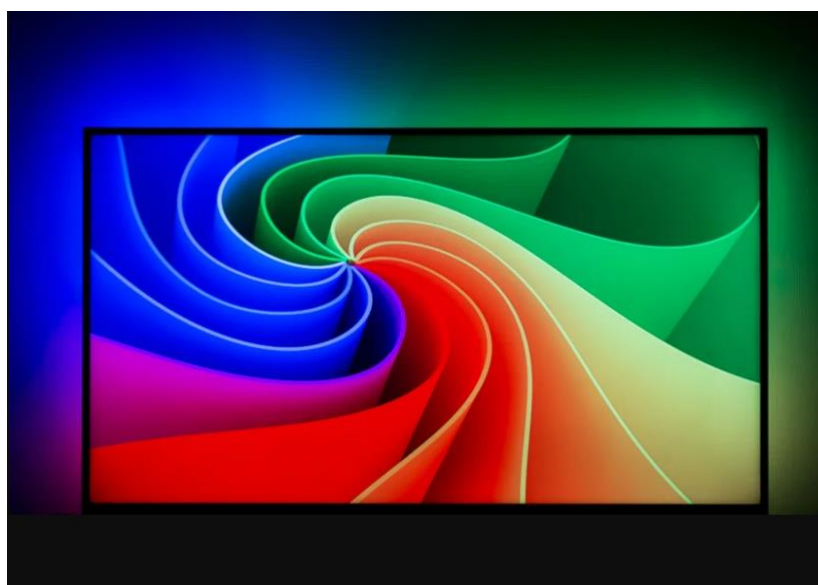


Рисунок 2.1 – Приклад роботи інтелектуальної підсвітки монітора

З моменту запровадження цієї технології Philips ретельно контролює її використання, не допускаючи до ринку конкурентів із подібними рішеннями. Теоретично існує можливість отримати дозвіл на використання Ambilight шляхом ліцензування, однак умови настільки суворі, що інші компанії не поспішають

долучатися. Деякі дрібні виробники пробували розробляти альтернативні варіанти у формі окремих комплектів, але Philips, як правило, негативно реагує на подібні ініціативи [10]. В ідеальному випадку, лише після закінчення терміну дії патенту або його оновлення інші виробники отримають змогу створювати схожі продукти – орієнтовно після 2027 року.

Для звичайного користувача обмеження, пов'язані з патентом, не є критичними. Кожен охочий може створити подібну систему власноруч. Далі буде детально розглянуто варіант самостійного виготовлення динамічного підсвічування за зразком Ambilight (далі – просто Ambilight).

Існує кілька основних методів реалізації цієї технології, і вибір залежить від джерела відеосигналу. Найдоступнішим і найефективнішим є використання комп'ютера з ОС Windows, macOS або Linux як джерела зображення. Особливо популярними є бюджетні Windows-системи на базі процесорів Atom, ціна яких починається від 70\$, і вони чудово підходять для цієї задачі.

Конструкція обладнання в цьому варіанті є уніфікованою для всіх зазначених платформ. Саме така конфігурація буде реалізована в межах цього проєкту. Основне програмне забезпечення – AmbiBox для Windows. Для macOS і Linux доцільно використовувати альтернативу – програму Prismatik.

Другий спосіб передбачає використання медіапристроїв на Android, які доволі численні. Цей варіант вважається найскладнішим у реалізації. Підсвічування працює лише в межах медіацентру Kodi (та сумісних додатках), і найчастіше система функціонує лише з вимкненим апаратним декодуванням, що створює серйозні обмеження. Крім того, апаратні характеристики значно впливають на стабільність роботи [11].

Третій підхід – повністю автономна система, незалежна від джерела відео. Цей варіант найдорожчий, однак і найгнучкіший, оскільки дозволяє зчитувати сигнал безпосередньо з HDMI-кабелю. Для цього потрібен мінікомп'ютер з достатнім рівнем продуктивності (наприклад, Raspberry Pi), HDMI-розгалужувач, HDMI-RCA AV-конвертер і пристрій захоплення відео через USB 2.0. Саме така схема дозволяє використовувати Ambilight з будь-яким джерелом сигналу – ТВ-

приставками, Android-боксами, Apple TV, ігровими консолями на зразок Xbox One чи PlayStation 4 тощо.

У варіанті з підтримкою роздільної здатності 1080p60 вартість комплектуючих (без урахування світлодіодної стрічки) складе приблизно 70\$. У разі необхідності обробки зображення у форматі 2160p60 – близько 100\$.

Щоб зрозуміти, як працює Ambilight, варто розглянути її базовий принцип: спеціальний модуль обробляє зображення, що виводиться на екран, виділяє кольори по краях матриці й передає відповідні сигнали на адресну RGB-стрічку, розміщену позаду дисплея. Захоплення відео та оновлення світіння відбуваються кілька разів на секунду, тому для стабільної роботи потрібна висока обчислювальна потужність. Чим більша роздільна здатність зображення, тим складніший і ресурсомісткіший процес його аналізу.

Існують два основних типи технічних реалізацій Ambilight: на базі зовнішнього декодера або через телевізійну приставку. Перший тип – це окремий пристрій із входом і виходом HDMI, який зазвичай містить одноплатний комп'ютер (часто на основі Raspberry Pi). Такий пристрій під'єднується між джерелом відео (ігровою консоллю, приставкою тощо) та телевізором. Всередині здійснюється обробка сигналу, а результат передається на екран (рис. 2.2).



Рисунок 2.2 – Приклад організації підсвітки Ambilight

Основна перевага цієї конструкції – простота інсталяції: не потрібно встановлювати складне ПЗ або налаштовувати систему. Пристрій просто вмикається в HDMI-ланцюг. Водночас, такий підхід має і свої недоліки: якість зображення, як правило, знижується. Якщо обчислювальних ресурсів недостатньо, система може не впоратися з передачею відео у форматі 4K, часто немає підтримки HDR, що знижує насиченість і контрастність. Крім того, можливі зниження частоти кадрів або втрата багатоканального звуку, що негативно впливає на ефект присутності.

Ці проблеми виникають або через обмежену потужність пристрою, або через застарілі HDMI-інтерфейси. Недостатньо продуктивні процесори й графічні модулі можуть не впоратися з обробкою відео високої якості. Також старі або бюджетні пристрої можуть мати HDMI-порти попередніх версій, що не підтримують сучасні технології (4K, HDR, високий FPS). Тому перед вибором такого рішення важливо оцінити не лише ціну, а й технічні характеристики. Також потрібно впевнитися, що пристрій сумісний із усіма елементами вашої мультимедійної системи, аби уникнути зайвих витрат і проблем у майбутньому.

Другий варіант реалізації підсвічування базується на використанні телевізійної приставки з попередньо встановленим програмним забезпеченням для захоплення відео. У цьому випадку потрібен лише невеликий контролер, що управляє адресною стрічкою (рис. 2.3).



Рисунок 2.3 – Приклад організації підсвітки на базі спеціального ПЗ

Перевагою є невисока вартість системи. Однак до недоліків можна віднести підвищені вимоги до апаратного забезпечення приставки: вона повинна бути достатньо потужною, щоб виконувати два завдання одночасно — відтворювати контент і здійснювати його захоплення.

Крім того, технічна реалізація деяких процесорів (наприклад, Amlogic) створює захисний шар при декодуванні відео, що блокує можливість зчитування зображення. Такі обмеження потребують часу на пошук налаштувань, сумісних плеєрів і відповідного програмного забезпечення [12].

Перед тим як зібрати таку систему, варто впевнитися, що всі компоненти присутні. Обов'язковою умовою є використання зовнішньої приставки — вбудовані функції Smart-TV не підійдуть. Підійдуть Raspberry Pi, Windows-комп'ютери, адаптовані як приставки, або сучасні Android-бокси.

Також важливо, щоб телевізор мав вузькі рамки — інакше світловий ефект буде порушено, і між зображенням і підсвіткою з'явиться розрив. Окрім того, навколо телевізора повинен бути вільний простір не менше 30–40 см, щоби світло рівномірно розсіювалося [13].

Нарешті, обрана приставка повинна мати відповідну продуктивність — дешеві стік-пристрої, як правило, не відповідають необхідним технічним вимогам. Перед придбанням комплектуючих також слід визначитися з кількістю світлодіодів на метр (30 або 60) і загальною довжиною стрічки:

- для екранів до 22" достатньо 1 метра стрічки;
- для 22"– 32" – 2 метри;
- для 32"– 43" – 3 метри;
- для 43"– 60" – 4 метри;
- понад 60" – до 5 метрів.

Для точного визначення довжини стрічки рекомендується виміряти периметр телевізора по задній панелі. Важливо також визначити, чи потрібне підсвічування нижньої грані екрану. При розташуванні телевізора на підставці ефект від світлодіодів знизу майже непомітний, але при кріпленні на настінний кронштейн обов'язково слід враховувати всі чотири сторони екрану.

Метою даної дипломної роботи є автоматизація функціонування аналога пристрою інтелектуального підсвічування монітора або телевізора, який матиме значно нижчу вартість порівняно з комерційними аналогами.

2.2 Вимоги до обладнання та вибір елементів для інтелектуальної підсвіткi комп'ютерного монітора

Для практичної реалізації пристрою інтелектуальної підсвіткi монітора необхідно використати три основні компоненти: адресну RGB-світлодіодну стрічку, блок живлення та мікроконтролер Arduino.

Контролер WS2811 є триканальним драйвером RGB-світлодіодів, що підтримує одно-дротову передачу даних та адресацію кожного світлодіода. Компонент WS2812B є вдосконаленим рішенням – це SMD-світлодіод типу 5050 з вбудованим контролером WS2811. У подальшому ці компоненти умовно називаються відповідно стрічками WS2811 та WS2812B.

Світлодіодна стрічка WS2812B являє собою гнучку плату, на якій послідовно розміщено адресні RGB-світлодіоди WS2812B (рис. 2.4). Вона функціонує при напрузі живлення 5 В. Доступні варіанти з різною щільністю елементів: 30, 60, 74, 90 або 144 світлодіоди на метр. Підкладка може бути білого або чорного кольору, залежно від потреб монтажу.



Рисунок 2.4 – Приклад RGB-стрічки WS2812B

Стрічки також класифікуються за ступенем захисту:

- IP20–30 – базовий захист від твердих частинок;
- IP65 – захист від пилу та водяних струменів (частково волога середовище);

– IP67 – захист при короткочасному зануренні у воду (до 1 метра).

У свою чергу, стрічка WS2811 складається з послідовно з'єднаних RGB-світлодіодів, кожен з яких керується окремим мікроконтролером WS2811 (рис. 2.5). Існують модифікації з живленням на 5 В або 12 В. За характеристиками щільності та захисту цей тип не поступається WS2812B.



Рисунок 2.5 – Приклад RGB-стрічки WS2811

Для більш масштабних проєктів можуть застосовуватися стрічки WS2811 з потужними світлодіодами великого формату (рис. 2.6). Вони забезпечують підвищену яскравість, однак переважно застосовуються для підсвічування великих екранів або панелей.



Рисунок 2.6 – Приклад RGB-стрічки WS2811 з потужними світлодіодами

Живлення стрічки є критичним елементом. При наявності блоку живлення на 5 В доцільно використовувати WS2812B, тоді як для джерел з напругою 12 В

оптимальним вибором буде WS2811. Якщо кількість світлодіодів у системі не перевищує 120 одиниць, з огляду на спрощення схеми живлення та управління, доцільно застосовувати WS2812B. У випадку перевищення цієї межі рекомендовано використовувати WS2811.

Щодо ступеня захисту – у більшості побутових випадків оптимальним вибором є рівень IP65. Такі стрічки мають силіконове покриття (або шар епоксидної смоли) з лицьового боку та самоклеючу основу 3М з тильного боку, що спрощує монтаж та обслуговування.

Оптимальна щільність розміщення світлодіодів для даного проєкту становить 30–60 одиниць на метр. Збільшення щільності підвищує деталізацію динамічного підсвічування (роздільну здатність системи Ambilight) та яскравість, але водночас вимагає більш потужного джерела живлення. Максимальна кількість світлодіодів, передбачена в рамках даного проєкту, обмежена 60 одиницями.

Для досягнення максимального візуального ефекту та повного занурення під час перегляду контенту, рекомендується встановлювати світлодіодну стрічку системи Ambilight по всьому периметру задньої частини монітора або телевізора. Це забезпечує рівномірне підсвічування фону навколо пристрою, що не лише покращує естетику, а й зменшує навантаження на очі при тривалому перегляді.

Оптимальним варіантом вважається чотиристороннє розміщення стрічки — по верхньому, обом боковим та нижньому краях корпусу. Такий спосіб монтажу створює найповніший ефект розширення зображення за межі екрана, особливо помітний у темному приміщенні.

У випадках, коли телевізор або монітор встановлено на підставці, що унеможливорює кріплення стрічки знизу, або коли нижня частина пристрою щільно прилягає до поверхні меблів, можливе тристороннє встановлення — по верхньому та обох бокових краях. Це також забезпечує привабливий світловий ефект, хоча й дещо менш інтенсивний у порівнянні з повним контуром.

Під час монтажу важливо враховувати рівномірність розташування стрічки, відстань до стіни, а також тип поверхні позаду екрана — світлі стіни краще відбивають світло, посилюючи Ambilight-ефект. Для досягнення найкращих

результатів варто дотримуватись рекомендацій виробника щодо способу кріплення та налаштування системи.

У реалізованій модифікації системи було використано білу стрічку WS2812B з рівнем захисту IP65, щільністю 60 світлодіодів/м (рис. 2.7, 2.8). Для стабільної роботи такої стрічки необхідний блок живлення з вихідною напругою 5 В і струмом не менше 2 А, рекомендовано – 3,6 А для забезпечення запасу потужності.



Рисунок 2.7 –RGB-стрічка WS2812B у вигляді мотка 2м



Рисунок 2.8 –RGB світлодіодна стрічка WS2812B

Підбір блоку живлення здійснюється відповідно до параметрів напруги та потужності. Для світлодіодної стрічки типу WS2812B потрібна стабільна напруга у 5 Вольт. У випадку з моделлю WS2811 – можлива робота як від 5В, так і від 12В. Найбільше енергоспоживання одного світлодіода WS2812B становить близько 0,3 Вт, аналогічний показник має і WS2811. Таким чином, загальна потужність блоку живлення має бути не меншою за добуток кількості світлодіодів (N) на 0,3 Вт: $P \geq N \times 0,3 \text{ Вт}$ [15].

Розглянемо приклад: для екрана з діагоналлю 42 дюйми при застосуванні WS2812B із щільністю 30 світлодіодів на один метр, для повного обрамлення монітора знадобиться близько 3 метрів стрічки. У такому випадку рекомендовані параметри джерела живлення складатимуть 5В з вихідним струмом щонайменше 6А. У поточному проєкті використовується один метр WS2812B із загальною кількістю 60 світлодіодів, отже, оптимальні характеристики живлення: 5В / 4А.

Для забезпечення живлення можна скористатись окремим зовнішнім джерелом із відповідними характеристиками, який необхідно придбати додатково (рис. 2.9). Як альтернатива, допускається застосування лабораторного джерела живлення або навіть блоку від стаціонарного ПК [15].



Рисунок 2.9 – Зовнішній блок живлення 5В/4А

У деяких випадках дозволено використовувати зарядні пристрої для мобільних телефонів, якщо вони відповідають потрібним параметрам вихідного струму. Типові сучасні зарядні блоки забезпечують вихідну потужність у межах 5–20 Вт, видаючи струм 1–3А, що робить їх сумісними з невеликими електронними пристроями: наприклад, з міні-комп'ютерами, маршрутизаторами чи портативними колонками.

Водночас важливо враховувати, що не всі зарядні пристрої демонструють однакову якість та ефективність у роботі з різною електронікою. Варто бути

уважними до якості самого зарядного пристрою і до кабелів, які використовуються. Дешеві або несертифіковані адаптери можуть продукувати нестабільний струм, що може призвести до перегріву, збоїв у роботі чи навіть пошкодження пристрою. Тому рекомендується обирати продукцію перевірених брендів, які мають сертифікати якості.

Ще один критичний аспект – безпечність експлуатації. Більшість сучасних зарядних пристроїв оснащуються засобами захисту від короткого замикання, перегріву та перенавантаження. Використання такого обладнання значно знижує ризики та забезпечує надійну роботу системи.

Таким чином, грамотний вибір джерела живлення забезпечує стабільну роботу системи освітлення, продовжує ресурс компонентів і знижує ймовірність виникнення аварійних ситуацій (рис. 2.10).



Рисунок 2.10 – Блок живлення з роз'ємами USB 5В/4А

У випадку використання джерела живлення з напругою 12В можна підключити негерметичний блок живлення потужністю 15Вт у перфорованому металевому корпусі зі стабілізованою вихідною напругою 12В постійного струму. У цій моделі не передбачено підстроювальний резистор для регулювання вихідної напруги. Підключення до мережі 220 В, а також підключення споживачів перетвореної напруги здійснюється за допомогою гвинтових клем.

Цей блок живлення вирізняється компактними розмірами серед аналогічних за потужністю моделей, що дозволяє використовувати його навіть в умовах

обмеженого простору. Блок має захист від короткого замикання та перевантаження, що знижує ризик його виходу з ладу. Під час експлуатації без активної системи охолодження необхідно забезпечити природну циркуляцію повітря в місці встановлення. Не допускається монтаж у закритих просторах без вентиляції (рис. 2.11).



Рисунок 2.11 – Блок живлення 12В/15W

Для створення системи інтелектуальної підсвітки монітора потрібні наступні апаратні елементи:

- мікроконтролер Arduino – 1 шт.;
- блок живлення з USB-роз'ємами, 5В / 4А – 1 шт.;
- RGB-світлодіодна стрічка WS2812B довжиною 1 м;
- набір з'єднувальних кабелів та USB-шнур з відповідним роз'ємом.

2.3 Розробка схеми пристрою інтелектуальної підсвітки монітора

Однією з найвідповідальніших частин при побудові такої системи є забезпечення якісного й стабільного живлення стрічки. При великій довжині і високому струмі споживання (особливо при роботі на 5 В) нерідко виникає падіння напруги, що спричиняє нестабільну роботу світлодіодів. Значна частина типових несправностей пов'язана саме з порушеннями в системі

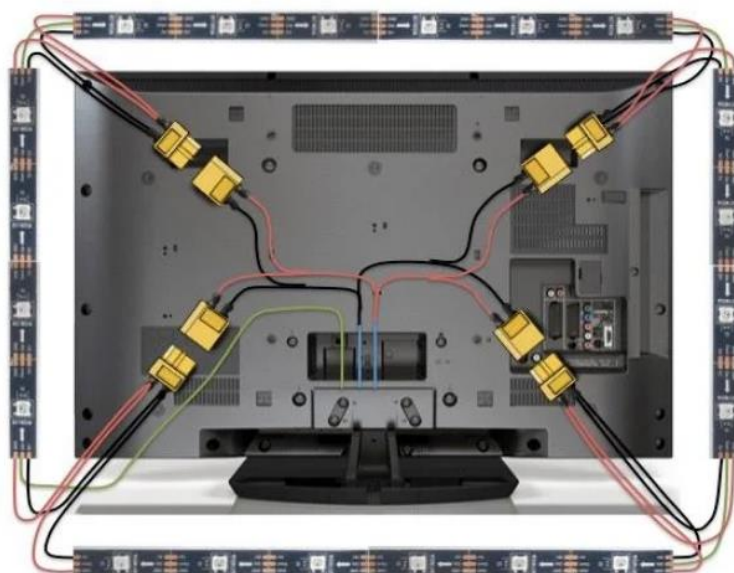


Рисунок 2.13 – Організація живлення довгої світлодіодної стрічки на 4 сторони монітора

Для правильного підведення живлення необхідно використовувати дроти відповідного перерізу: чим менший номер калібру AWG, тим більшу силу струму може витримати провідник.

У разі використання плати Arduino UNO для управління підсвіткою, конструктивне рішення пристрою повинно відповідати принциповій схемі, зображеній на рисунку 2.14.

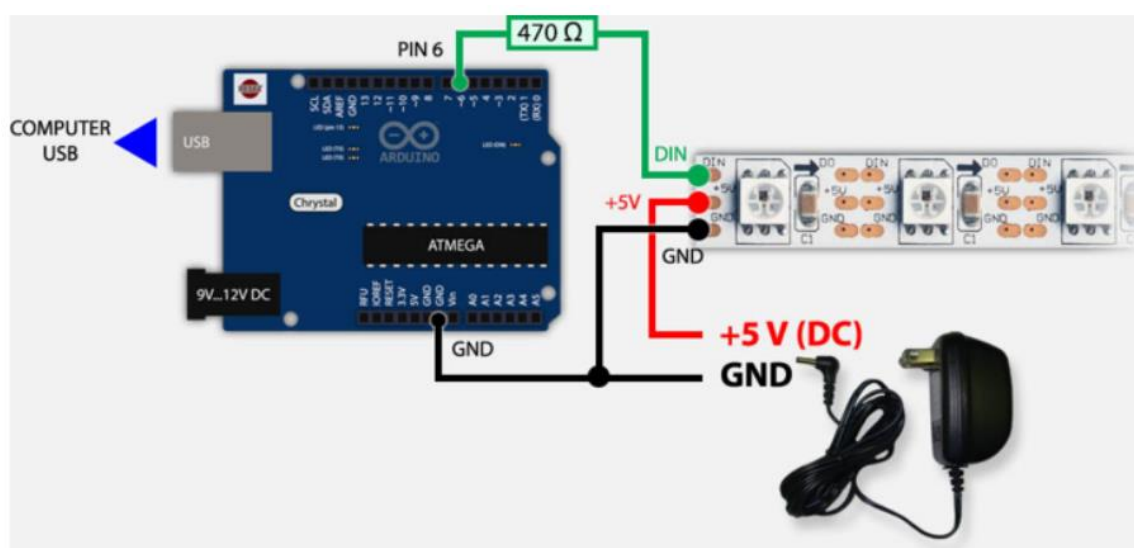


Рисунок 2.14 – Схема пристрою інтелектуальної підсвітки на платі Arduino UNO

Для підключення стрічки до плати Arduino використовується два основні контакти: GND (земля), який необхідно під'єднати до відповідного роз'єму контролера, і DATA – інформаційний вивід, що має бути підключений до цифрового порту №6 через резистор з номіналом у межах 300–550 Ом (оптимально – 470 Ом).

Що стосується мікроконтролера Arduino Nano, то його можна розміщувати у будь-якому зручному корпусі. Як приклад, для цього може підійти навіть звичайна упаковка з-під шоколадного «Кіндер-сюрпризу». Важливо розташовувати сам контролер якомога ближче до стрічки, щоб мінімізувати довжину лінії передавання сигналу DATA. За умови правильного підключення усіх компонентів, зовнішній вигляд готового пристрою інтелектуальної -підсвітки зображено на рисунку 2.15.

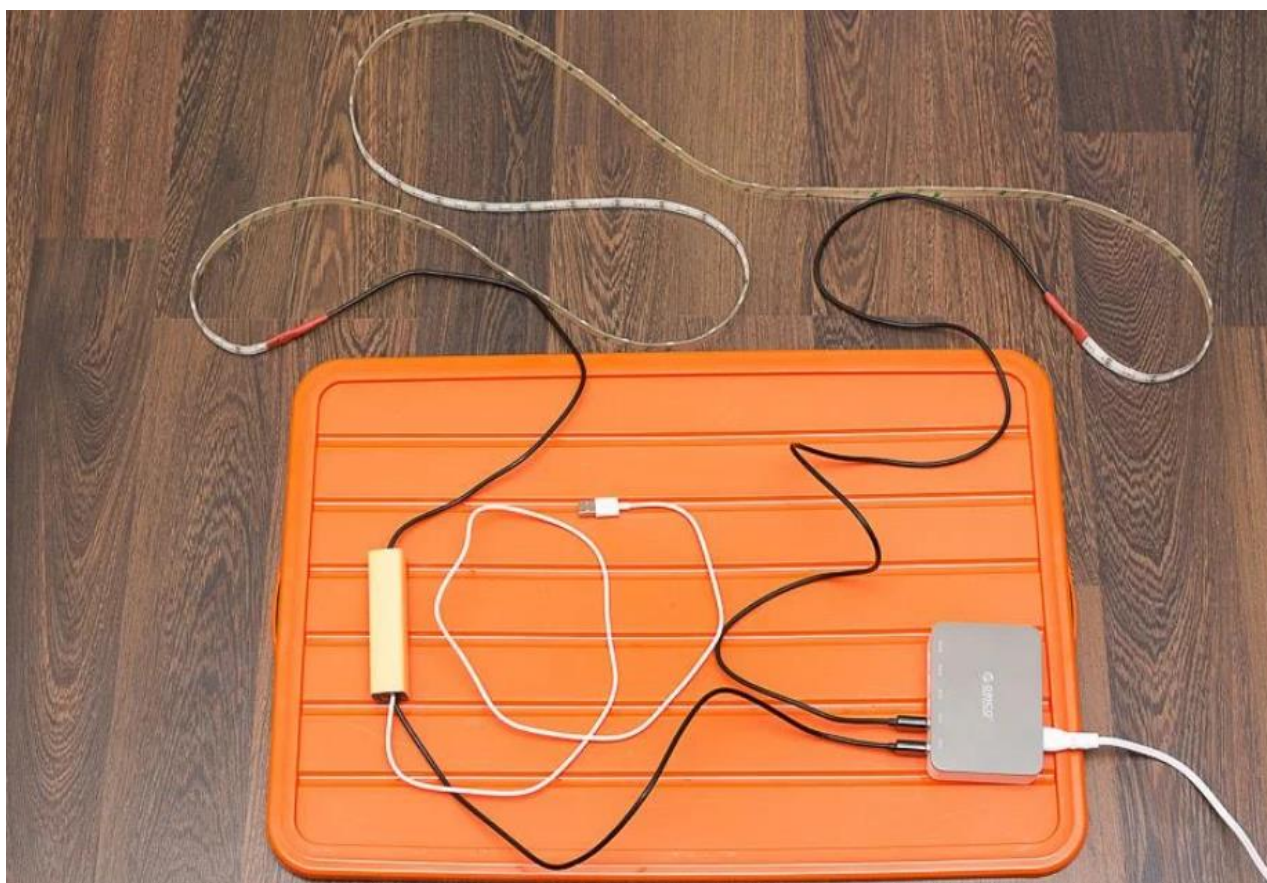


Рисунок 2.15 – Загальний вигляд пристрою інтелектуальної підсвітки для монітору

Живлення RGB-світлодіодної стрічки забезпечується двома чорними USB-кабелями, а білий USB-кабель виконує функцію підключення до комп'ютера. Місця з'єднання проводів зі стрічкою обов'язково повинні бути заізолювані, що досягається шляхом використання термоусадкових трубок.

Ще одним важливим і практичним завданням під час монтажу LED-стрічки є забезпечення її вигину під прямим кутом — наприклад, при обрамленні телевізора або монітора по периметру. Особливу складність становить реалізація такого вигину у випадках, коли використовується світлодіодна стрічка з високою щільністю розміщення світлодіодів, зокрема 60 світлодіодів на метр або більше. Через малу відстань між діодами стрічка стає менш гнучкою, а надмірний вигин може призвести до пошкодження провідників або навіть до виходу з ладу окремих елементів.

У таких ситуаціях стрічку рекомендується розрізати в місцях, позначених виробником як безпечні для обрізки, зазвичай через кожні 3–5 см, залежно від моделі. Після цього відрізані частини з'єднуються між собою за допомогою коротких гнучких дротів. Важливо дотримуватись полярності та не допускати перехресного підключення контактів: +12V або +5V, а також каналів R, G, B або Data/Clock для цифрових стрічок. Для підвищення надійності та довговічності таких з'єднань рекомендується застосовувати пайку, після чого всі оголені контакти слід заізолювати термоусадковими трубками, що не лише запобігає короткому замиканню, але й забезпечує механічний захист.

Альтернативним рішенням, особливо для користувачів без досвіду роботи з паяльником, є застосування спеціалізованих кутових з'єднувачів. Такі компоненти мають три або чотири контакти, залежно від типу стрічки (RGB, RGBW, адресна тощо), і дозволяють швидко та безпечно з'єднати дві частини стрічки під кутом 90 градусів. З'єднувачі часто мають заціпний механізм або притискні контакти, що полегшує монтаж і не потребує пайки. Водночас варто пам'ятати, що електричний контакт у таких з'єднувачах менш надійний, ніж при пайці, тому їх рекомендується використовувати лише в умовах стаціонарного встановлення без значних механічних навантажень. Правильний вибір способу

з'єднання та його якісне виконання є запорукою стабільної роботи всієї системи підсвічування, без мерехтіння, втрат кольору або перебоїв у передачі сигналу (рис. 2.16).

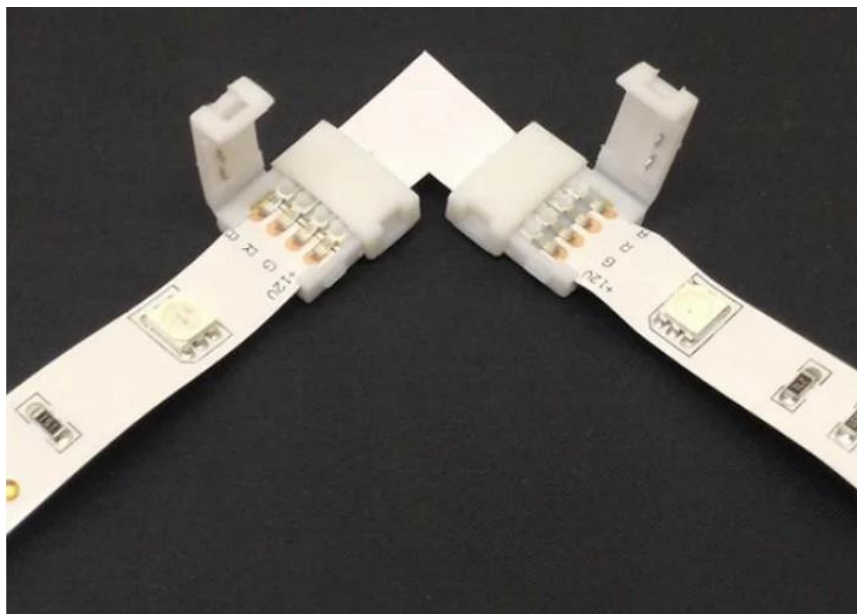


Рисунок 2.16 – Кутові конектори для світлодіодних стрічок

Якщо ж світлодіодна стрічка має нижчу щільність, наприклад 30 світлодіодів на метр, це відкриває низку переваг під час її монтажу. Через збільшені проміжки між окремими світлодіодами така стрічка є значно гнучкішою, як у фізичному, так і в практичному сенсі. Зменшення кількості компонентів на одиницю довжини знижує напруження в шарі провідників, що дозволяє стрічці легко згинатися навіть на малих радіусах без ризику пошкодження електричних доріжок чи самих світлодіодів.

Завдяки цій гнучкості значно полегшується формування прямих кутів, обходження виступів, заокруглень або інших особливостей поверхні. У більшості випадків немає необхідності розрізати стрічку та здійснювати повторне під'єднання контактів — це економить час, зменшує кількість монтажних матеріалів (провідників, термоусадок, з'єднувачів) і знижує загальний рівень складності робіт. Особливо це є цінним у випадках, коли встановлення проводиться в важкодоступних місцях, наприклад, за меблями, всередині ніш, за

вигнутими телевізорами або вздовж декоративних конструкцій зі складною геометрією.

До того ж, стрічки з меншою щільністю споживають менше енергії, що може бути важливим у проектах із обмеженою потужністю джерел живлення. Хоча така стрічка забезпечує меншу яскравість та однорідність світіння в порівнянні з більш щільними аналогами, її все одно доцільно використовувати там, де не потрібна надмірна інтенсивність світла, а важливішими є гнучкість, простота монтажу та економічність.

Таким чином, вибір світлодіодної стрічки з нижчою щільністю є раціональним рішенням у багатьох дизайнерських і технічних задачах, особливо якщо ключовим пріоритетом є зручність монтажу на складних або нерівних поверхнях.

Крім того, такий підхід зменшує ризик пошкодження провідників чи втрати контакту, що часто трапляється при некоректному розрізанні та з'єднанні. Хоча при меншій щільності зменшується загальна яскравість підсвічування, для декоративних або допоміжних цілей така стрічка може стати практичним і зручним вибором (рис. 2.17).



Рисунок 2.17 – Організація кута 90^0 на світлодіодній стрічці

У разі правильної збірки пристрою для інтелектуальної підсвітки монітора та його акуратного встановлення на тильну сторону екрана, кінцевий вигляд конструкції відповідатиме зображенню на рисунку 2.18.



Рисунок 2.18 – Зовнішній вигляд встановленого на монітор пристрою інтелектуальної підсвітки

2.4 Висновки до розділу 2

На основі проведеного аналізу предметної області та вивчення характеристик існуючих систем підсвічування комп'ютерних моніторів було виконано проєктування і реалізацію інтелектуальної підсвітки для монітора комп'ютера. Обґрунтовано вибір технічних засобів, які використовувалися під час розробки. Розроблено функціональну схему пристрою інтелектуальної підсвітки монітора, а також схему інтеграції цього пристрою з монітором і блоком живлення. Надано детальний опис послідовності етапів створення запропонованого інноваційного рішення.

3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНОЮ ПІДСВІТКОЮ МОНІТОРА

3.1 Обґрунтування вибору середовища розробки

У процесі створення технічного рішення для RGB-світлодіодної інтелектуальної підсвітки дисплея як основний елемент керування було обрано мікроконтролер Arduino UNO. Відповідно до цього вибору, процес програмування даного пристрою буде здійснюватися шляхом використання спеціалізованого програмного середовища Arduino IDE.

Перед початком створення програмного коду і процедурами програмування мікроконтролера важливо провести необхідні підготовчі заходи. На персональному комп'ютері потрібно встановити відповідне програмне забезпечення - драйвер для віртуального COM-порту (послідовного з'єднання) та інтегроване середовище програмування для платформи Arduino.

Першочерговим завданням є інсталяція драйверного забезпечення для мікросхеми перетворювача USB-UART інтерфейсу. Якщо використовується оригінальна плата Arduino або її високоякісний аналог, слід встановити драйвер для чіпа FT232RL. Проте, цей етап може бути пропущений, адже необхідне драйверне забезпечення вже входить до складу інсталяційного пакету середовища розробки Arduino.

У разі застосування сумісних плат виникає необхідність встановлення драйверів для відповідних мікросхем, що використовуються в таких пристроях. Зазвичай це чіпи CH340, CP2102 або PL2303. Програмування мікроконтролера перетворювача інтерфейсів виконано таким способом, що в диспетчері пристроїв плата розпізнається як віртуальний COM-порт (емульований послідовний інтерфейс).

Подальшим етапом є інсталяція програмного середовища Arduino, яке створено на базі програми Processing. Processing являє собою інтуїтивне

інтегроване середовище розробки, що було спеціально розроблено для користувачів, які не володіють поглибленими знаннями в галузі програмування, але потребують створення власних програмних продуктів.

Останню стабільну версію середовища Arduino можна отримати з офіційного веб-ресурсу проекту за посиланням www.arduino.cc. Програмне забезпечення доступне у вигляді інсталятора або як архівний файл з переносною версією, яка не вимагає процедури встановлення. Інсталятор забезпечить покроковий процес встановлення на вашому персональному комп'ютері, в той час як переносну версію необхідно лише розпакувати для подальшого застосування без інсталяції.

Надзвичайно важливо регулярно оновлювати програмне забезпечення Arduino IDE, оскільки команда розробників систематично вносить нові бібліотеки, драйверне забезпечення та усуває виявлені недоліки. Це гарантує сумісність з найновішими варіантами апаратного забезпечення та підвищує загальну стабільність роботи середовища. Для оновлення Arduino IDE слід перевіряти доступність нових версій на офіційному веб-сайті та завантажувати найсвіжішу версію, дотримуючись аналогічних кроків, як і при первинному встановленні. Таким способом, забезпечивши правильну інсталяцію драйверного забезпечення та Arduino IDE, ми створюємо стійку платформу для розробки та апробації проектів на базі Arduino.

IDE (Integrated Development Environment) – інтегроване середовище розробки Arduino є кросплатформним застосунком, створеним на мові програмування Java, який забезпечує користувачів зручним інструментарієм для розробки, компіляції та завантаження програм на плати Arduino. Дане середовище складається з кількох основних компонентів:

- текстовий редактор для створення та модифікації програм (скетчів) для Arduino з підтримкою виділення синтаксису, автоматичного доповнення коду та інших функціональних можливостей, що полегшують процес створення коду. Це робить програмування більш доступним і зрозумілим навіть для користувачів-початківців;

- компілятор, який перетворює створений код у формат, придатний для розуміння мікроконтролером на платі Arduino. Після розробки програми можна оперативно скомпілювати її для виявлення помилок і переконання у готовності коду до завантаження;

- модуль передачі програмного коду, який після успішної компіляції завантажує програму на плату Arduino. Цей компонент використовує послідовний інтерфейс для передачі скомпільованого коду на мікроконтролер. Після завантаження мікроконтролер виконує програму в автономному режимі.

Середовище Arduino IDE створено з урахуванням потреб новачків, які не володіють глибокими знаннями в програмуванні. Мова програмування Arduino схожа на ту, що застосовується в проекті Wiring. З технічної точки зору це C/C++ з інтеграцією певних бібліотек та деякими обмеженнями (зокрема, відсутні можливості багатопотокового програмування і неповна об'єктна модель C++). Програмний код обробляється препроцесором, а згодом компілюється за допомогою вільно розповсюджуваного компілятора AVR-GCC

3.2 Установка середовища Arduino IDE

Щоб розпочати створення системи інтелектуального освітлення, потрібно встановити на персональний комп'ютер спеціалізоване програмне середовище для розробки застосунків Arduino – Arduino IDE. Дане програмне рішення сумісне з наступними операційними платформами:

- Windows (підтримуються версії від XP до Windows 11);
- Linux (доступні збірки для архітектур x32, x64 та ARM);
- Mac OS X.

Завантаження програмного забезпечення здійснюється через офіційний веб-ресурс за адресою: <https://www.arduino.cc/en/Main/Software>, де потрібно обрати і скачати встановлювальний файл актуальної версії додатку (рис. 3.1)..



Рисунок 3.1 – Вікно завантаження інтегрованого середовища програмування Arduino IDE

Після обрання відповідного інсталяційного файлу для конкретної операційної платформи на екрані браузера з'явиться діалогове вікно з двома варіантами: безплатне завантаження або варіант із добровільним фінансовим внеском.

Необхідно скористатися опцією "just download" (рис 3.2), позначеною червоною рамкою у правому нижньому секторі, щоб отримати інсталятор без обов'язкового благодійного внеску.



Рисунок 3.2 – Вікна завантаження інсталятора

Після активації кнопки "just download" потрібно обрати директорію для

розміщення встановлювального файлу, щоб по завершенні процесу завантаження мати можливість легко знайти та активувати його для подальшого встановлення Arduino IDE (рис. 3.3).

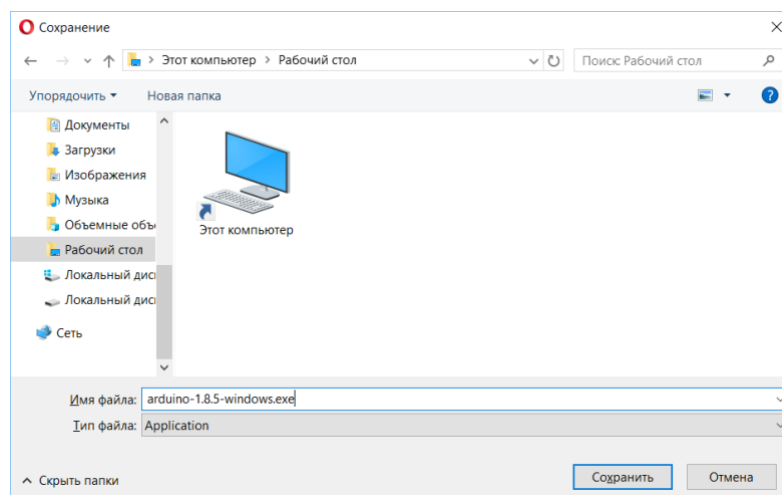


Рисунок 3.3 – Вікно збереження інсталлятора

Під час активації встановлювального файлу на екрані з'явиться діалог (рис. 3.4) з текстом ліцензійних умов від розробників Arduino IDE. Подальший процес встановлення програмного продукту активується натисканням кнопки "I Agree", яка обведена червоною рамкою у правому нижньому розділі вікна.

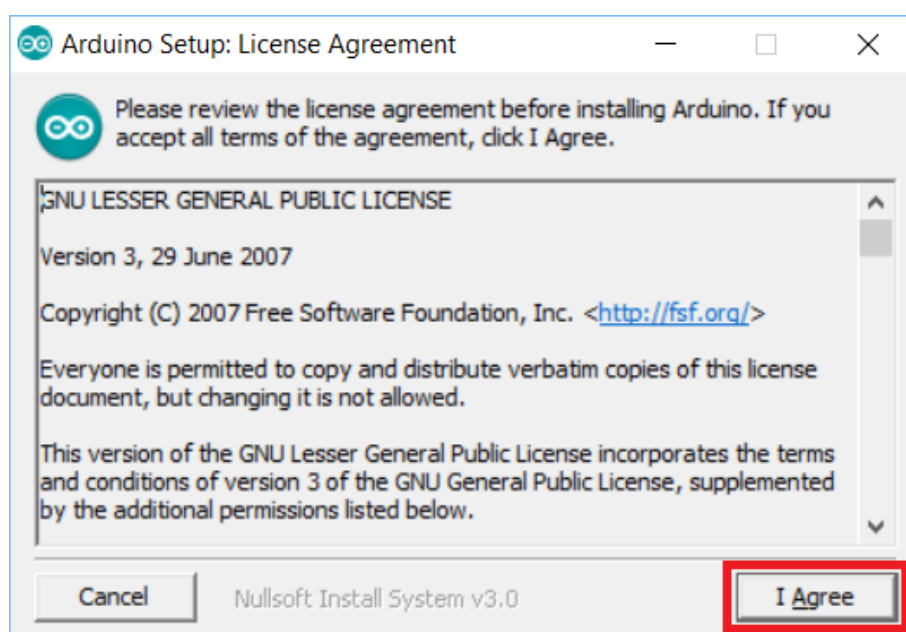


Рисунок 3.4 – Погодження з ліцензійними правами Arduino IDE

Подальший крок передбачає відбір потрібних модулів Arduino IDE для їх інсталяції на робочу станцію. Для цього слід позначити усі доступні компоненти згідно з рисунком 3.5, після чого активувати кнопку "Next", виділену червоною рамкою у правому нижньому розділі інтерфейсу.

Наступним етапом є визначення цільової директорії для розміщення програмного продукту. На рисунку 3.6 демонструється типовий маршрут встановлення на системному диску C:. Продовження процедури відбувається через натискання кнопки "install", позначеної червоною рамкою у правому нижньому куті діалогового вікна.

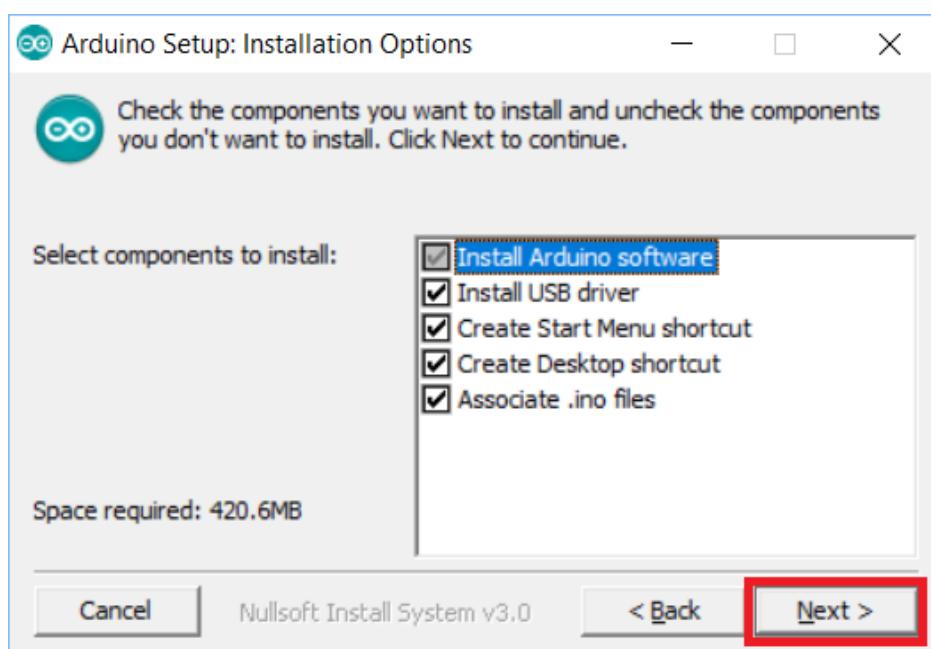


Рисунок 3.5 – Вікно вибору компонентів Arduino IDE для їх встановлення

Протягом процесу встановлення Arduino IDE система може потребувати підтвердження для інсталяції драйверів портів зв'язку. У такій ситуації слід надати програмному забезпеченню відповідні дозволи для їх встановлення. Рисунок 3.7 ілюструє фінальну стадію встановлення додатку і завершення роботи з інсталятором через активацію кнопки "Close".

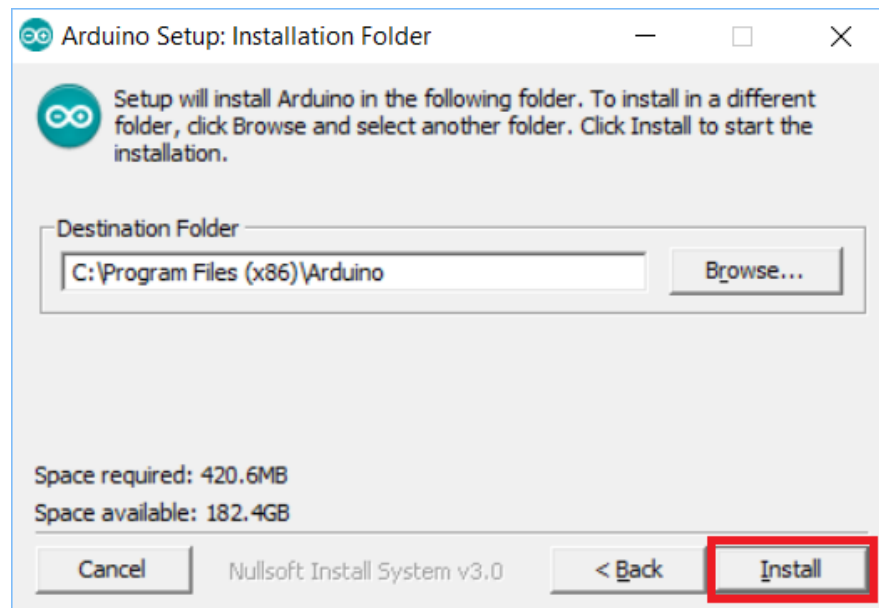


Рисунок 3.6 – Вікна вибору папки для встановлення програми Arduino IDE

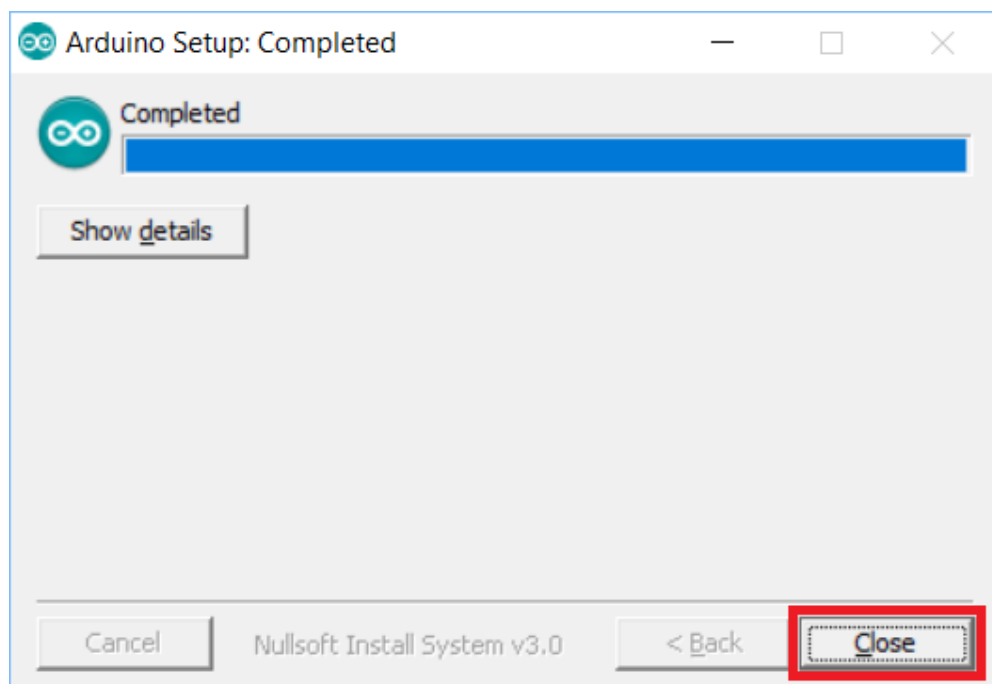


Рисунок 3.7 – Вікно завершення встановлення Arduino IDE

Після виконання попередніх кроків на робочому столі має з'явитись ярлик "Arduino" (рис. 3.8).

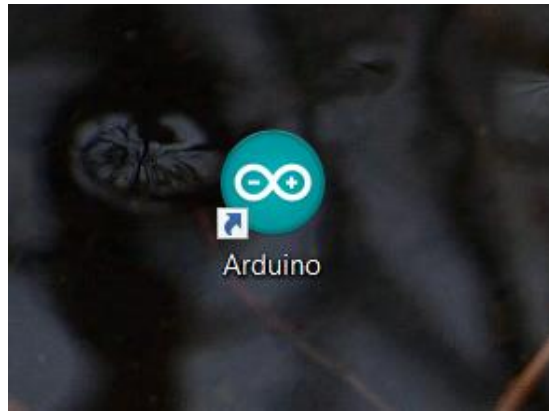


Рисунок 3.8 – Ярлик додатку Arduino IDE на робочому столі

Після завершення процедури завантаження та встановлення програмного продукту Arduino IDE, запуск здійснюється шляхом подвійного натискання лівою кнопкою миші на піктограмі "Arduino", внаслідок чого буде активовано робочий інтерфейс Arduino IDE (рис. 3.9).

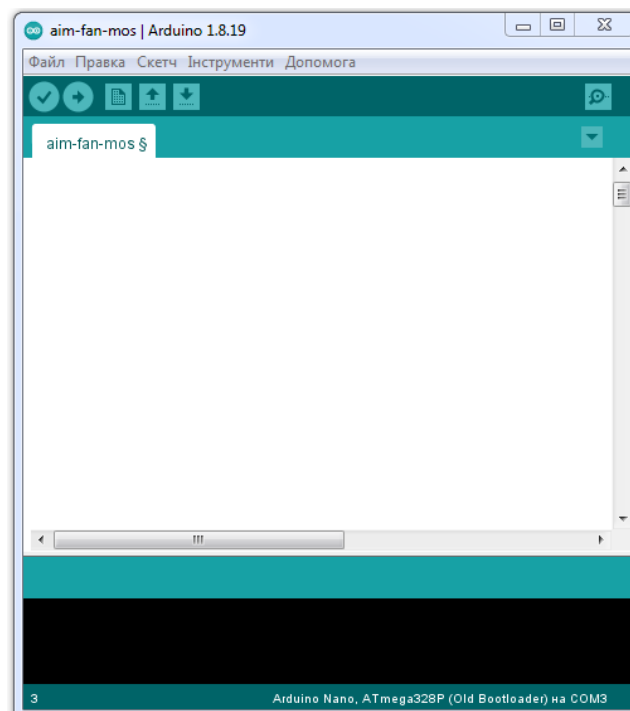


Рисунок 3.9 – Вікно додатку Arduino IDE

По завершенні процедури встановлення програмного середовища Arduino IDE необхідно здійснити фізичне з'єднання плати Arduino Uno з персональним комп'ютером.

З цією метою встановлюємо з'єднання між Arduino Uno та комп'ютером за допомогою USB-кабелю. На мікроконтролерній платі Arduino Uno активується індикатор "ON" і розпочне періодичне спалахування світлодіод "L" (рис. 3.10). Такі візуальні сигнали свідчать про успішну подачу електроживлення на плату та ініціацію виконання мікроконтролером Arduino Uno заздалегідь "прошитої" базової програми "Blink" (функція періодичного блимання світлодіодом).

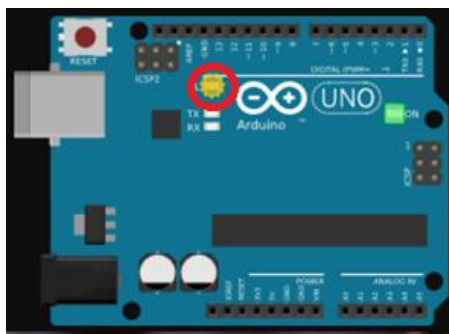


Рисунок 3.10 – Сигнали про ввімкнення живлення Arduino UNO

Для належного налаштування Arduino IDE на взаємодію з платою Arduino Uno потрібно визначити, який саме номер COM-порту операційна система призначила пристрою Arduino. З цією метою необхідно відкрити "Диспетчер пристроїв Windows" (рис. 3.11) і розгорнути розділ "Порти (COM и LPT)".

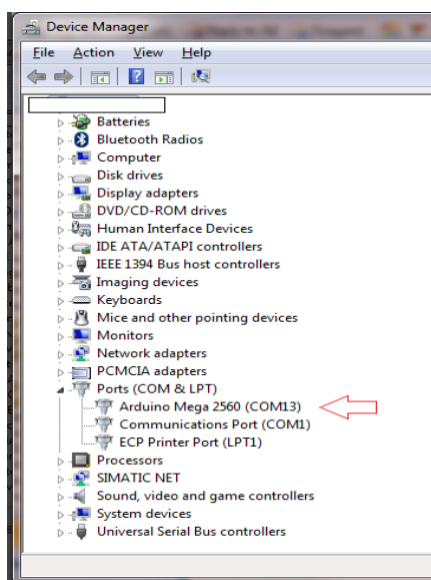


Рисунок 3.11 – Диспетчер пристроїв Windows

У даному випадку операційна платформа розпізнала мікроконтролерну плату Arduino Uno як COM-пристрій, автоматично підбрала відповідний драйвер і присвоїла даному COM-порту ідентифікаційний номер 13. При підключенні альтернативної плати Arduino операційна система призначить їй відмінний номер. Тому при роботі з множинними платами Arduino критично важливо уникати плутанини в номерах COM-портів.

Для конфігурування Arduino IDE для ефективної взаємодії з мікроконтролерною платою Arduino Uno потрібно зазначити місцезнаходження плати Arduino Uno у програмному середовищі Arduino IDE через її COM-порт "COM1".

З цією метою переходимо до розділу меню "Сервис" → "Последовательный порт" і обираємо порт "COM1" (рис. 3.12). Після цього Arduino IDE успішно розпізнаватиме місцезнаходження плати на порту "COM1".

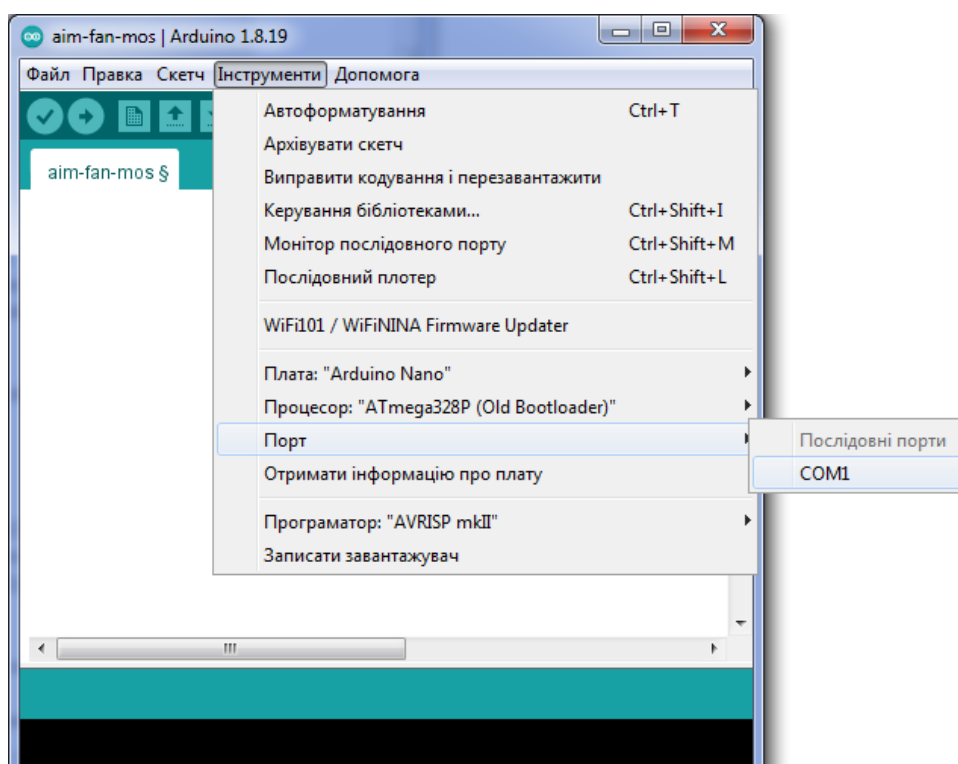


Рисунок 3.12 – Вибір COM-порту

Для забезпечення коректної компіляції програмного коду потрібно зазначити, що програмне середовище Arduino IDE взаємодіє саме з

мікроконтролером Arduino Uno. Це включає декілька фундаментальних етапів для гарантування належної взаємодії програмного забезпечення з апаратною платформою.

З цією метою переходимо до розділу меню "Сервис" → "Плата" і обираємо модель "Arduino Uno" (рис. 3.13).

Правильне налаштування програмного середовища Arduino IDE для взаємодії з мікроконтролером Arduino Uno є фундаментально важливим для успішної компіляції та завантаження програмного коду. Дотримання зазначених рекомендацій сприятиме запобіганню типових проблем та забезпечить стабільне функціонування створюваного проекту.

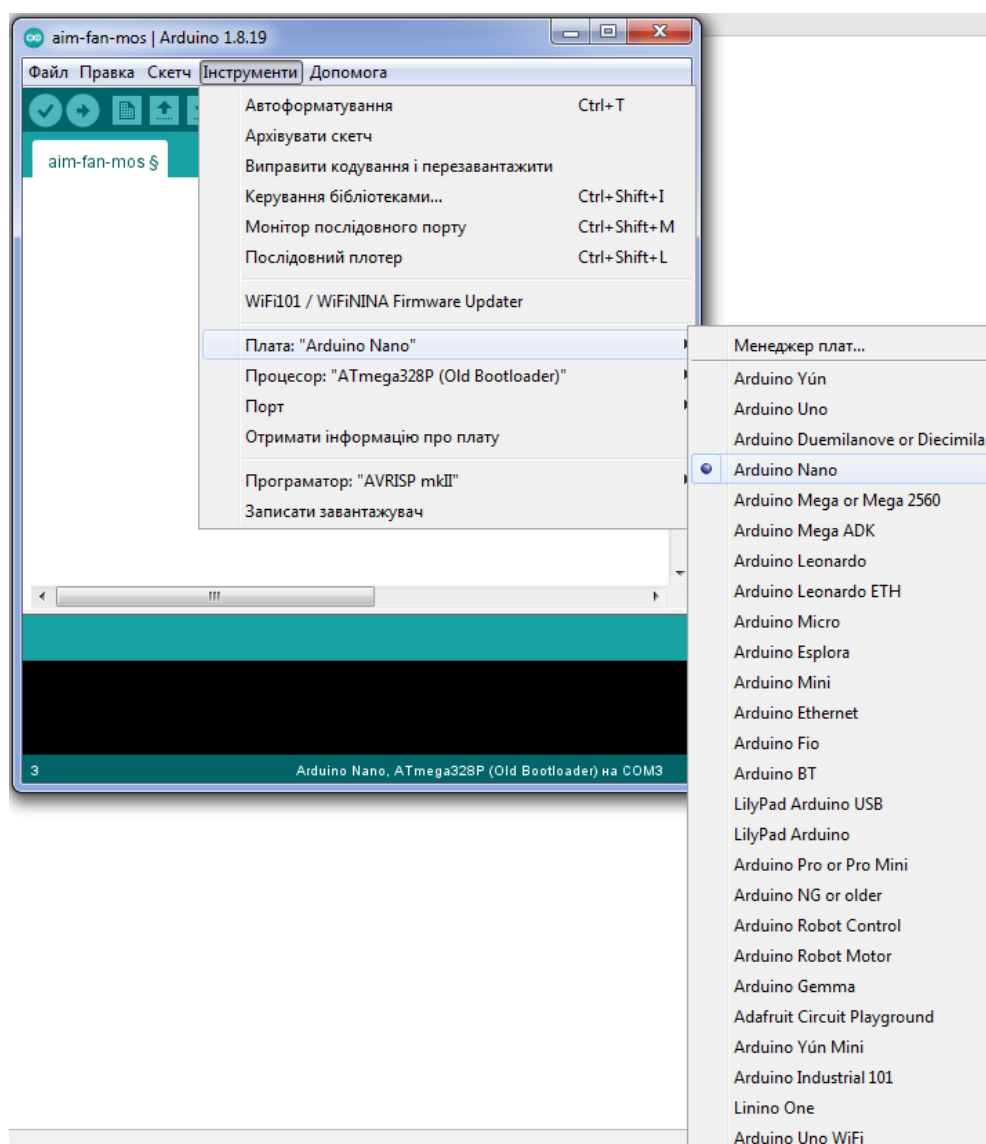


Рисунок 3.13 – Вибір плати Arduino UNO

Програмне середовище Arduino IDE включає готові зразки попередньо розроблених рішень для потенційного швидкого впровадження цих прикладів у основну програмну логіку.

Щоб верифікувати функціональність мікроконтролерної плати Arduino Uno, потрібно завантажити до неї перший тестовий програмний код. Для завантаження початкової програми обираємо, наприклад, скетч під назвою "Blink" (рис. 3.14).

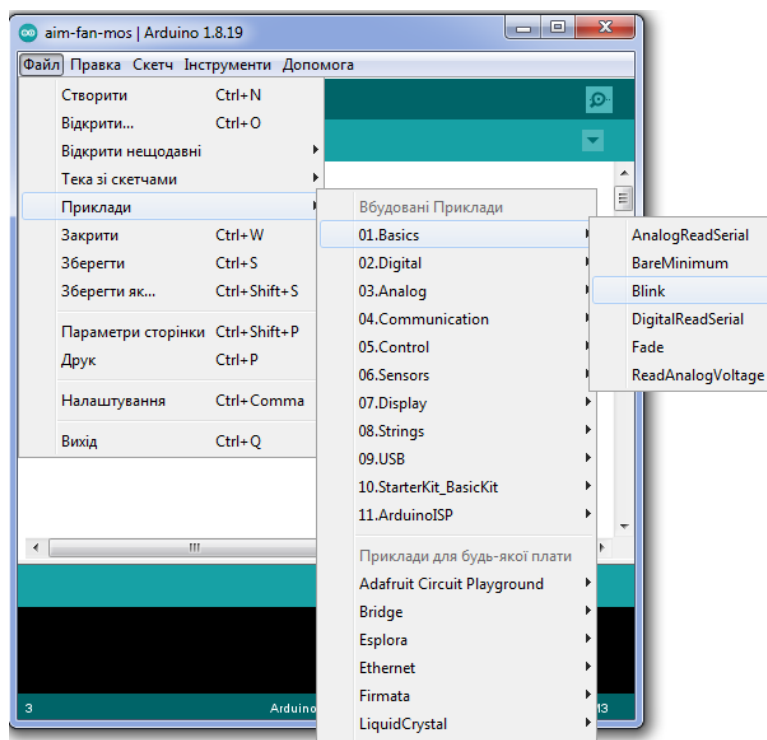


Рисунок 3.14 – Вибір програми “Blink”

Для завантаження програми “Blink” необхідно натиснути кнопку “Вивантажити” (рис. 3.15).

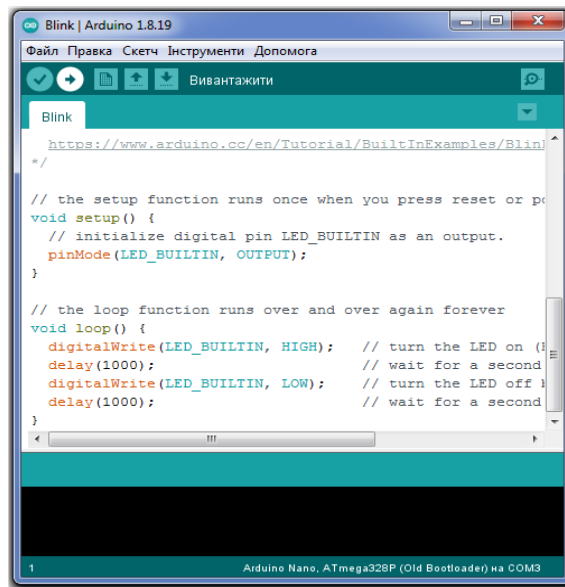


Рисунок 3.15 – Завантаження програми “Blink”

Після встановлення або розпакування архіву з програмою Arduino запускаємо файл Arduino.exe. Для зручності можна створити ярлик на робочому столі. Середовище Arduino IDE містить різноманітні інструменти та налаштування, які спрощують взаємодію користувача з програмою (рис. 3.16).

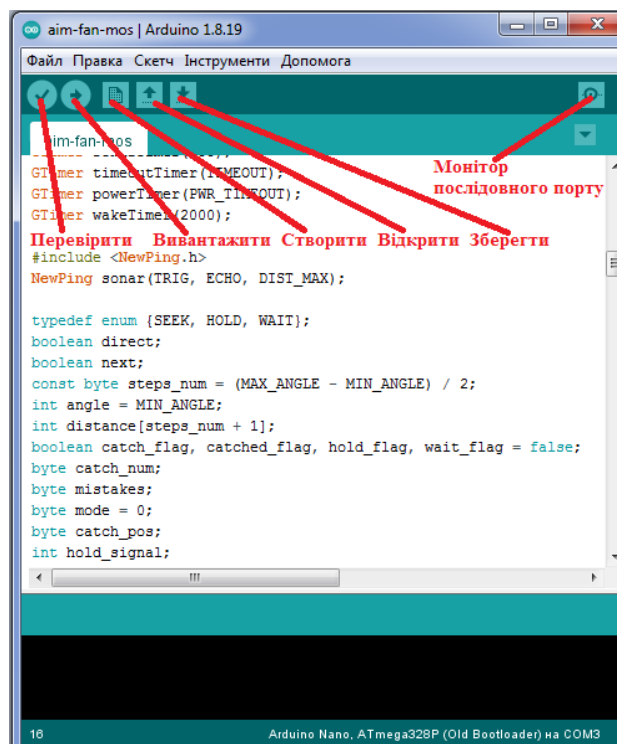


Рисунок 3.16 - Середовище розробки Arduino

Розглянемо докладніше можливості середовища програмування Arduino IDE. Ця платформа забезпечує повний набір засобів і функцій, необхідних для зручної взаємодії з мікроконтролерними платами Arduino. Основні інструменти доступні через меню, розташоване у верхній частині вікна програми. Безпосередньо під цим меню знаходиться панель швидкого доступу до основних команд:

- «Перевірити» – виконує аналіз коду на предмет синтаксичних та інших помилок, дозволяючи виявити їх ще до завантаження програми на пристрій;
- «Завантажити» – створює виконуваний файл і передає його на мікроконтролерну плату, після чого програма починає виконуватись;
- «Створити» – відкриває новий скетч (файл для написання коду), що особливо корисно при запуску нового проєкту;
- «Відкрити» – дозволяє переглядати й завантажувати раніше збережені файли для подальшого редагування або компіляції;
- «Зберегти» – зберігає зміни у поточному файлі, що є важливим для уникнення втрати інформації;
- «Монітор послідовного порту» – викликає вбудований термінал для спілкування з платою через інтерфейс UART. Це зручно для налагодження і спостереження за даними у режимі реального часу.

Кожна команда має відповідну іконку на панелі, що забезпечує оперативний доступ до функцій. Меню та інструментальна панель створюють комфортне середовище для програмування, де користувач може зосередитись на розробці та тестуванні коду.

Середовище Arduino IDE також підтримує додаткові можливості — зокрема, менеджери плат та бібліотек. Вони дають змогу встановлювати нові або оновлювати наявні компоненти, які забезпечують підтримку широкого спектру мікроконтролерів. Такий інструмент є корисним як для новачків, так і для професіоналів, оскільки дозволяє реалізовувати проєкти будь-якого рівня складності.

Програма в середовищі Arduino IDE завжди базується на базовій структурі,

що складається з двох обов'язкових функцій — `setup()` і `loop()`. Відсутність хоча б однієї з них призведе до помилки компіляції. Наприклад:

```
void setup() {
  Serial.begin(9600); Serial.println("loop");
}

void loop() {
  delay(1000);
  Serial.println("loop");
}
```

Функції в Arduino – це окремі блоки коду, які мають унікальні імена та визначаються фігурними дужками. До таких блоків можна звертатися з різних частин програми.

Функція `setup()` виконує одноразове налаштування системи – тут програміст задає початкові параметри роботи пристрою й об'єктів, що використовуватимуться надалі.

Функція `loop()` є безперервним циклом, у якому реалізується основна логіка програми. Після ініціалізації мікроконтролер виконує інструкції, що містяться в `setup()`, а далі безперервно переходить до обробки команд, зазначених у `loop()`.

Розглянемо приклад, у якому керується вбудованим світлодіодом на платі (контакт №13), а також передається текст у монітор порту:

```
int led_pin = 13;

void setup() {
  Serial.begin(9600);
  pinMode(led_pin, OUTPUT);
  Serial.println("setup");
}

void loop() {
  digitalWrite(led_pin, HIGH);
  delay(1000);
  digitalWrite(led_pin, LOW);
```

```

    delay(1000);
    Serial.println("loop");
}

```

У цьому коді мікроконтролер чергує вмикання та вимикання світлодіода з інтервалом в одну секунду, супроводжуючи це повідомленнями у терміналі.

Розглянемо ще один приклад – використання LCD-дисплея для виведення повідомлення. Перед основними функціями йде підключення бібліотеки та створення об'єкта дисплея, який підключено до певних пінів:

```

#include "LiquidCrystal.h"
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
    lcd.begin(16, 2);
}
void loop() {
    lcd.print("Hello, world!");
    delay(1000);
}

```

Розглянемо складніший приклад структури програми з підключенням бібліотеки та створенням екземпляра класу. У цьому прикладі програма щосекунди виводить повідомлення «Hello, world!» на LCD-дисплей. Звернемо увагу на типові елементи коду, які часто зустрічаються у скетчах Arduino.

Спочатку здійснюється підключення бібліотеки LiquidCrystal, яка забезпечує взаємодію з символьними рідкокристалічними дисплеями. Далі оголошується об'єкт класу LiquidCrystal з параметрами, що вказують номери контактів мікроконтролера, до яких підключено дисплей. У функції setup() викликається метод lcd.begin(16, 2) — це початкова ініціалізація дисплея з вказанням кількості символів у рядку (16) та кількості рядків (2). Це є обов'язковим налаштуванням перед початком роботи з дисплеєм.

Основна частина програми знаходиться в функції loop(), де виконується періодичне (раз на секунду) виведення повідомлення на екран.

Ще одним прикладом є варіант із використанням користувацької функції `myfunc()`. У цьому випадку структура скетчу зберігається, але в тілі `loop()` замість прямого виклику команд відбувається звернення до функції, яка реалізує виведення повідомлення:

```
#include "LiquidCrystal.h"

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
}

void loop() {
  myfunc();
  delay(1000);
}

void myfunc(){
  lcd.print("Hello, world!");
}
```

В Arduino-скетчах часто використовуються коментарі, які допомагають краще зрозуміти призначення окремих частин коду. Коментарі ігноруються компілятором і можуть бути:

- однорядковими, які починаються з `//`;
- багаторядковими, які обмежуються символами `/* */..`.

```
/*
```

```
Blink
```

```
Turns on an LED on for one second, then off for one second, repeatedly.
```

```
*/
```

```
int led_pin = 13;
```

```
// setup function
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  pinMode(led_pin, OUTPUT);
```

```

Serial.println("setup");
}
// loop function
void loop() {
    digitalWrite(led_pin, HIGH);
    delay(1000);
    digitalWrite(led_pin, LOW);
    delay(1000);
    Serial.println("loop");
}

```

Таким чином, перед початком роботи з Arduino необхідно виконати початкові налаштування середовища. Насамперед слід вибрати відповідну модель плати (у нашому випадку – Arduino Nano) та встановити потрібний інтерфейс підключення. Якщо потрібна інша плата, її можна обрати зі списку доступних моделей у меню середовища Arduino IDE (рис. 3.17).

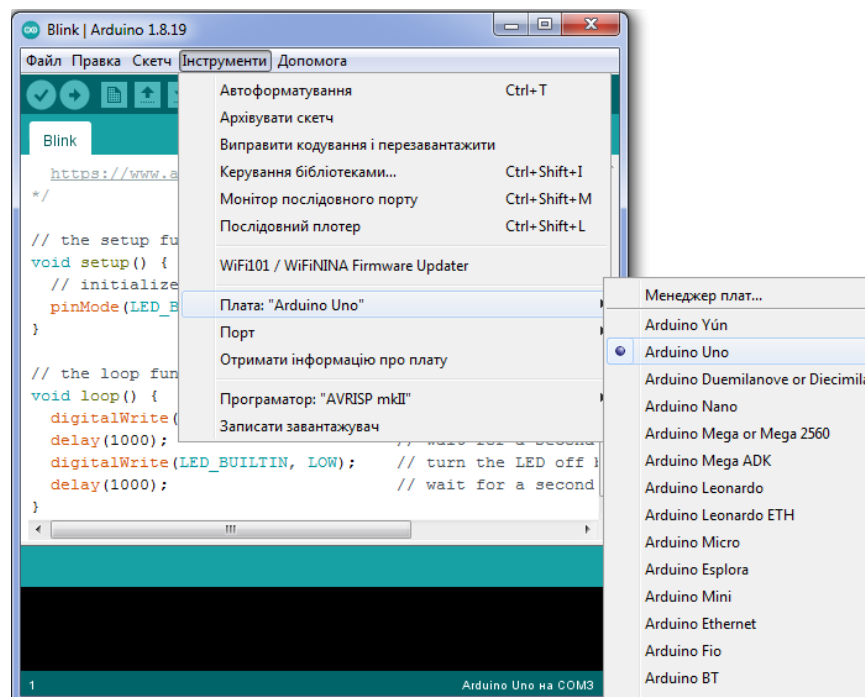


Рисунок 3.17 - Вибір плати мікроконтролера Arduino

3.3 Вибір мови програмування

Програмне забезпечення для роботи системи смарт-підсвітлення монітора було створено з використанням мови програмування C++. Мова C++ була створена як універсальна мова з використанням статичних типів даних, що забезпечує ефективність та портативність мови C. C++ була розроблена для комплексної підтримки різноманітних підходів до програмування (процедурного підходу, абстракції даних, об'єктно-орієнтованого підходу та узагальненого програмування).

Вона створена для забезпечення програмістові свободи вибору, навіть коли це може призвести до вибору неоптимальних рішень. C++ розроблена для максимального збереження сумісності з мовою C, що гарантує плавний перехід від програмування на C. Вона уникає особливостей, що залежать від платформи або не є стандартизованими. Не створює зайвого навантаження на програму, яка не застосовує спеціальні можливості.

Створена для роботи без складного середовища програмування. C++ вважається однією з найефективніших мов програмування.

Переваги C++ у порівнянні з C:

- покращена безпека;
- можливість розробки узагальненого коду через використання шаблонів; — підтримка об'єктно-орієнтованого підходу;
- управління ресурсами через використання RAII;
- спрощення коду завдяки перевантаженню функцій та операторів;
- більш ефективна обробка помилок через використання винятків.

Недоліки застосування мови C++:

- тривалий процес компіляції;
- збільшений розмір генерованого машинного коду.

3.4 Програмування контролера Arduino

Здійснюємо завантаження та розархівування середовища Arduino IDE. Отримуємо бібліотеку FastLED та переміщуємо директорію FastLED до папки libraries (Arduino IDE). Активуємо Arduino IDE та завершуємо роботу програми. У директорії Документи буде сформовано папку Arduino. У даній папці формуємо директорію Adalight і переміщуємо туди скетч Adalight.ino.

Здійснюємо підключення мікроконтролера Arduino через USB-з'єднання. Драйверне забезпечення (послідовний інтерфейс CH340) встановлюється в автоматичному режимі. У випадку відсутності автоматичної інсталяції, в каталозі Arduino IDE розміщена папка Drivers з необхідними компонентами.

Запускаємо Arduino IDE та відкриваємо файл Adalight.ino (рис. 3.18).

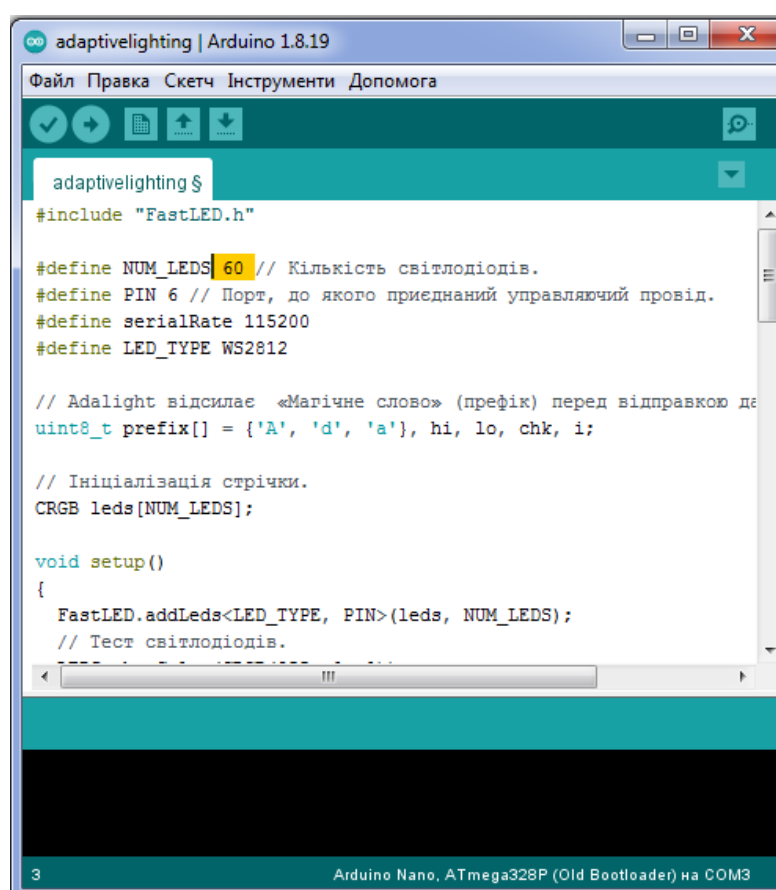


Рисунок 3.18 – Скетч Adalight.ino

Модифікуємо кількість світлодіодів у програмному коді. У нашому проекті

їх 60.

```
#include "FastLED.h"
```

```
#define NUM_LEDS 60 // Кількість світлодіодів.
```

```
#define PIN 6 // Порт, до якого приєднаний управляючий провід.
```

```
#define serialRate 115200
```

```
#define LED_TYPE WS2812
```

```
// Adalight відсилає «Магічне слово» (префікс) перед відправкою даних.
```

```
uint8_t prefix[] = {'A', 'd', 'a'}, hi, lo, chk, i;
```

```
// Ініціалізація стрічки.
```

```
CRGB leds[NUM_LEDS];
```

```
void setup()
```

```
{
```

```
  FastLED.addLeds<LED_TYPE, PIN>(leds, NUM_LEDS);
```

```
  // Тест світлодіодів.
```

```
  LEDS.showColor(CRGB(255, 0, 0));
```

```
  delay(500);
```

```
  LEDS.showColor(CRGB(0, 255, 0));
```

```
  delay(500);
```

```
  LEDS.showColor(CRGB(0, 0, 255));
```

```
  delay(500);
```

```
  LEDS.showColor(CRGB(0, 0, 0));
```

```
  Serial.begin(serialRate);
```

```
  Serial.print("Ada\n"); // Відсилаємо «магічне слово» для з'єднання з програмою.
```

```
}
```

```
void loop()
```

```
{
```

```

// Чекаємо перший байт із слова.
for(i = 0; i < sizeof prefix; ++i)
{
    waitLoop: while (!Serial.available());
    // Тест наступного байта із слова.
    if(prefix[i] == Serial.read()) continue;
    // Інакше починаємо все спочатку.
    i = 0;
    goto waitLoop;
}
// Чекаємо старший і молодший байт, а також контрольну суму.
while (!Serial.available());
hi = Serial.read();
while (!Serial.available());
lo = Serial.read();
while (!Serial.available());
chk = Serial.read();
// Якщо контрольна сума не співпала, починаємо спочатку.
if (chk != (hi ^ lo ^ 0x55))
{
    i = 0;
    goto waitLoop;
}
memset(leds, 0, NUM_LEDS * sizeof(struct CRGB));
// Отримуємо данні і налаштовуємо відповідний світлодіод.
for (uint8_t i = 0; i < NUM_LEDS; i++)
{
    byte r, g, b;
    while(!Serial.available());
    r = Serial.read();

```

```

while(!Serial.available());

g = Serial.read();

while(!Serial.available());

b = Serial.read();

leds[i].r = g;

leds[i].g = r;

leds[i].b = b;

}

// Світимо новий стан стрічки.

FastLED.show();

}

```

Наступним етапом є запис програми в контролер Arduino. Для цього обираємо меню «Інструменти > Плата > Arduino UNO». У розділі «Інструменти > Порт >» обираємо необхідний COM-порт (там буде відображений потрібний варіант). Активуємо кнопку «Вивантажити» (рис. 3.19).

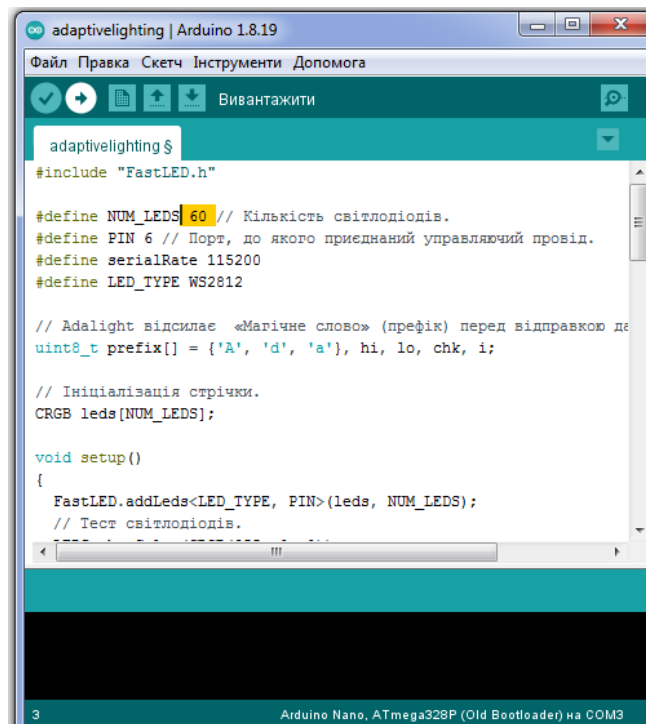


Рисунок 3.19 – Запис скетчу Adalight.ino в контролер Arduino UNO

Програмне забезпечення автоматично проінформує користувача про завершення процесу завантаження (тривалість процесу - декілька секунд).

На цьому етапі необхідно від'єднати контролер Arduino UNO від USB-порту та підключити повторно. Світлодіодна стрічка послідовно засвітиться червоним, зеленим та синім кольорами – це означає, що Arduino UNO активувався та готовий до експлуатації.

3.5 Налаштування програмного забезпечення AmbiBox

Наступним кроком є інсталяція програмного забезпечення AmbiBox, яке необхідно попередньо завантажити з офіційного джерела. Після завершення встановлення запускаємо програму. У головному вікні слід перейти до розширених налаштувань, натиснувши на кнопку «Більше налаштувань». У цьому розділі потрібно задати ключові параметри системи: як пристрій керування вказати — Adalight, обрати відповідний COM-порт, до якого підключено мікроконтролер, та ввести точну кількість світлодіодів у стрічці. Окрім цього, налаштовується кількість кадрів за секунду, які програма буде захоплювати з монітора для аналізу (рис. 3.20).

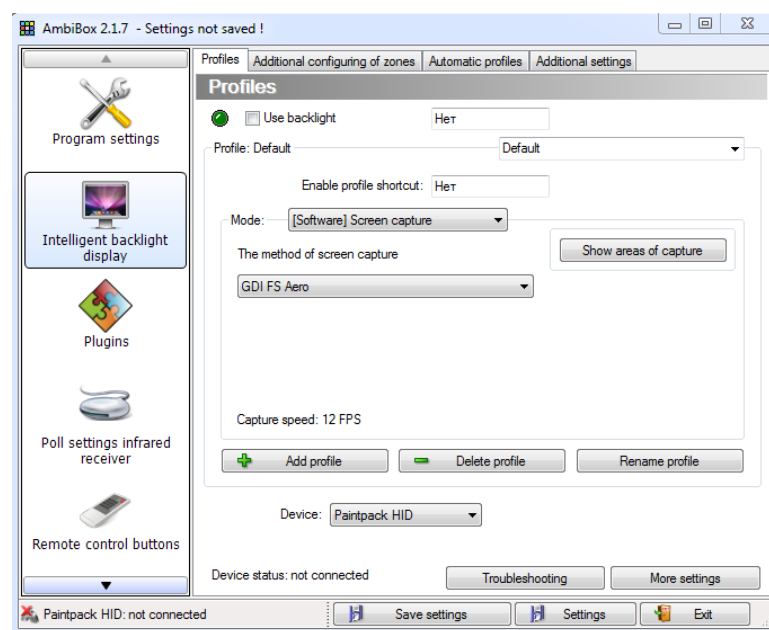


Рисунок 3.20 – Вікно налаштувань програми AmbiBox

Подальші дії передбачають конфігурацію зон захоплення зображення. Для цього потрібно активувати опцію «Показати зони захоплення», а потім запустити «Майстер налаштування зон». Цей інструмент дозволяє користувачу визначити конфігурацію та схему розташування світлодіодної стрічки по периметру монітора (рис. 3.21).

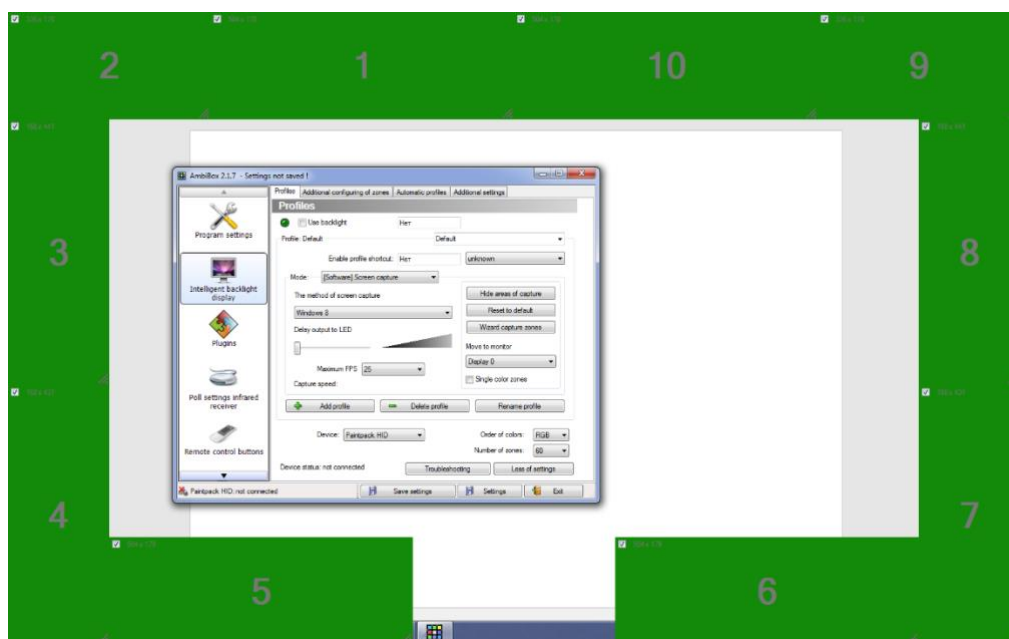


Рисунок 3.21 – Вікно конфігурації світлодіодної стрічки у програмі AmbiBox

Щоб зберегти внесені зміни, необхідно послідовно натиснути кнопки «Застосувати» та «Зберегти налаштування». На цьому процес базової конфігурації можна вважати завершеним. Проте, варто зазначити, що функціонал AmbiBox надає значно ширші можливості для детального налаштування. Користувач може гнучко змінювати розміри кожної зони захоплення, проводити корекцію кольорової палітри стрічки, а також регулювати такі параметри, як динаміка зміни кольору, насиченість, відтінки та загальна яскравість підсвічування (рис. 3.22).

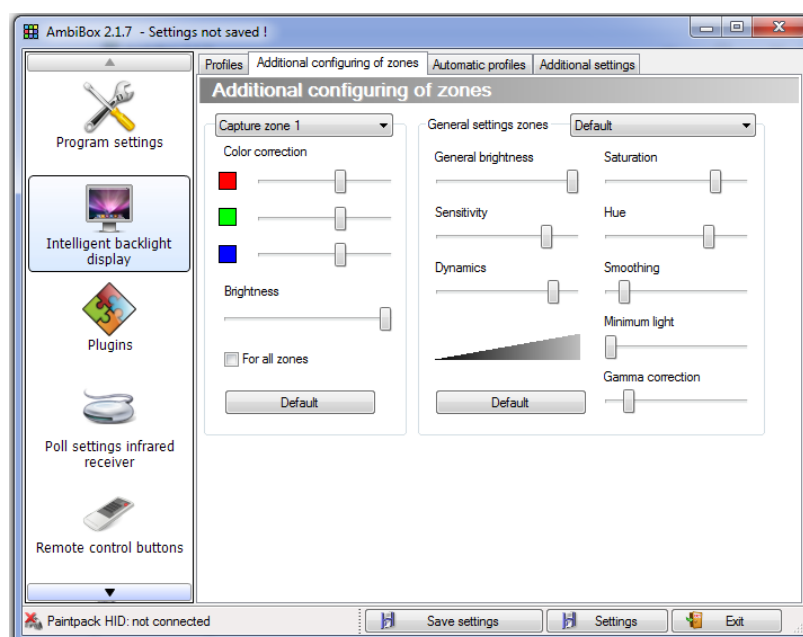


Рисунок 3.22 – Додаткові параметри налаштування у програмі AmbiBox

Керування роботою системи Ambilight за допомогою програмного забезпечення є інтуїтивно зрозумілим та не викликає труднощів навіть у користувачів без технічної підготовки. Програма, яка найчастіше використовується для взаємодії зі стрічкою — це AmbiBox, що забезпечує широку функціональність та зручне налаштування ефектів підсвічування.

Активація створеного профілю підсвічування здійснюється у кілька простих кроків. Після запуску операційної системи Windows програма AmbiBox автоматично розміщується в області сповіщень (системному треї), яка знаходиться у правому нижньому куті екрана — поряд з годинником. Для увімкнення підсвічування достатньо двічі натиснути лівою кнопкою миші на іконку AmbiBox. Після цього обрана світлодіодна стрічка активується та миттєво починає роботу відповідно до збереженого профілю налаштувань.

Користувач може заздалегідь налаштувати один або кілька профілів, кожен з яких міститиме певні параметри: режим роботи (наприклад, розширення зображення з екрана, фіксоване світло, динамічна анімація), кольорову палітру, швидкість реакції на зміну зображення, яскравість та інші ефекти. Це дозволяє швидко перемикатися між різними режимами — наприклад, для перегляду фільмів, гри або прослуховування музики.

Таким чином, завдяки зрозумілому інтерфейсу та простоті взаємодії, система Ambilight з AmbiBox дозволяє швидко й легко адаптувати підсвічування до будь-яких умов, забезпечуючи максимальний комфорт та візуальне задоволення під час користування комп'ютером чи перегляду контенту.

Вимкнення підсвічування здійснюється аналогічним чином — повторним подвійним кліком по іконці (рис. 3.23).



Рисунок 3.23 – Зовнішній вигляд монітора зі смарт-підсвітленням

3.6 Висновки до розділу 3

У третьому розділі детально описано процес створення програмного забезпечення для управління інтелектуальною підсвіткою комп'ютерного монітора. Розглянуто етапи встановлення, налаштування та основні параметри ПЗ, що забезпечує взаємодію з Arduino-платами. Також представлено процес конфігурації програмного компонента AmbiBox. Крім цього, виконано тестування розробленої системи та проведено аналіз результатів її роботи.

ВИСНОВКИ

Під час виконання бакалаврської кваліфікаційної роботи було розроблено апаратну та програмну частини програмно-апаратного комплексу управління інтелектуальною підсвіткою для монітору.

Проведено аналіз класифікації сучасних технологій підсвітлення екранів моніторів і телевізорів. Доведено доцільність створення нової розробки для розв'язування поставленого завдання. Описано принцип роботи розумної підсвітки для комп'ютерного монітора. Детально описано вибір апаратних складових для створення пристрою інтелектуальної підсвітки.

Описано бібліотеки, які використовує програмно-апаратний комплекс інтелектуальної підсвітки для забезпечення нормального функціонування. Обґрунтовано вибір мови програмування Wiring. Здійснено розробку і описано процес виготовлення програмно-апаратного комплексу інтелектуальної підсвітки для монітора. Було реалізовано програмну частину проєкту інтелектуальної підсвітки для монітора.

Проведено тестування розробленого програмно-апаратного комплексу інтелектуальної підсвітки для монітору та проаналізовано результати його роботи, які доводить що він повністю виконує всі свої функції.

Отже всі поставлені задачі виконано, мету роботи досягнуто.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бісікало О.В., Базелюк О.І. Організація інтелектуальної підсвітки для моніторів. Матеріали Міжнародної науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» – [Електронний ресурс] . – <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/25526/21145>.
2. Використання технології фонового підсвічування для телевізорів фірми Phillips Ambilight – Вікіпедія: веб-сайт. - Режим доступу: <https://uk.wikipedia.org/wiki/Ambilight>
3. Unified Modeling Language – Вікіпедія: веб-сайт. - Режим доступу: URL: [https://uk.wikipedia.org/wiki/ Unified_Modeling_Language](https://uk.wikipedia.org/wiki/Unified_Modeling_Language).
4. Microsoft Visual Studio – Вікіпедія: веб-сайт. Режим доступу: URL: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
5. Ситник В. Ф., Писаревська Т. А., Єршоміна Н. В., Краєва О. С. Основи інформаційних систем: Навчальний посібник Київ, 2005. 308 с.
6. Josuttis, Nicolai M. The C++ standard library : a tutorial and reference / Nicolai M. Josuttis.—2nd ed. с. 78-80.
7. Свароп С. Г. A Bite Of Python v2.01. 2013. 159 с. с. 20-23. PyCharm [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/pycharm/>
8. Chollet Francois Deep Learning with Python. 2018. — 400 с.
9. Miguel Grinberg Flask Web Development. Published by O'Reilly Media, Inc. 2018.
10. C++ 17 STL. Стандартна бібліотека шаблонів. Яцек Галовіц. 2018. ISBN: 978-5-4461-0680-6
11. Парадигми програмування. Елементи програм C++ [Electronic resource]. – Mode of access: WWW/URL: http://om.univ.kiev.ua/users_upload/15/upload/file/prog_lecture_01.pdf. – Title from the screen.
12. Кравець П. Об'єктно-орієнтоване програмування : навч. посібник / П.О. Кравець. – Львів: Видавництво Львівської політехніки, 2012. – 624 с.

13. C Tutorial [Electronic resource]. – Mode of access : WWW/URL : <https://www.tutorialspoint.com/cprogramming/index.htm>. – Title from the screen.
14. Arduino Pro Mini [Електронний ресурс] // doc.arduino.ua – 2021. – Режим доступу до ресурсу: <https://doc.arduino.ua/ru/hardware/ProMini>.
15. Arduino Nano [Електронний ресурс] // arduino.ua – 2021. – Режим доступу до ресурсу: [https://arduino.ua/prod166-arduino-nano-v3-0-avr-atmega 328p-s-raspayannimi-razemami](https://arduino.ua/prod166-arduino-nano-v3-0-avr-atmega-328p-s-raspayannimi-razemami).
16. Muresan D. D., Parks T. W., Adaptive Principal Components and Image Denoising, in: Image Processing, 2003, Proceedings 2003 IEEE International Conference on Image Processing (ICIP), 14-17 Sept. 2003, V. 1, pp. I-101-104.
17. Офіційний сайт мови Python [Електронний ресурс]. Режим доступу до матеріалу <https://www.python.org/>
18. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт (проектів) для студентів спеціальностей: 126 «Інформаційні системи та технології», 151 «Автоматизація та комп'ютерно-інтегровані технології» [Електронний ресурс] / Уклад. К. В. Овчинников, О. В. Бісікало. – Вінниця : ВНТУ, 2023. – 50 с

ДОДАТОК А (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

**«ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС УПРАВЛІННЯ
ІНТЕЛЕКТУАЛЬНОЮ ПІДСТВІТКОЮ МОНІТОРУ**



Рисунок А.1 – Приклад роботи технології Ambilight від компанії Philips



Рисунок А.2 – Приклад організації підсвітки Ambilight



Рисунок А.3 – Апаратні засоби для створення інтелектуальної підсвітки комп'ютерного монітора

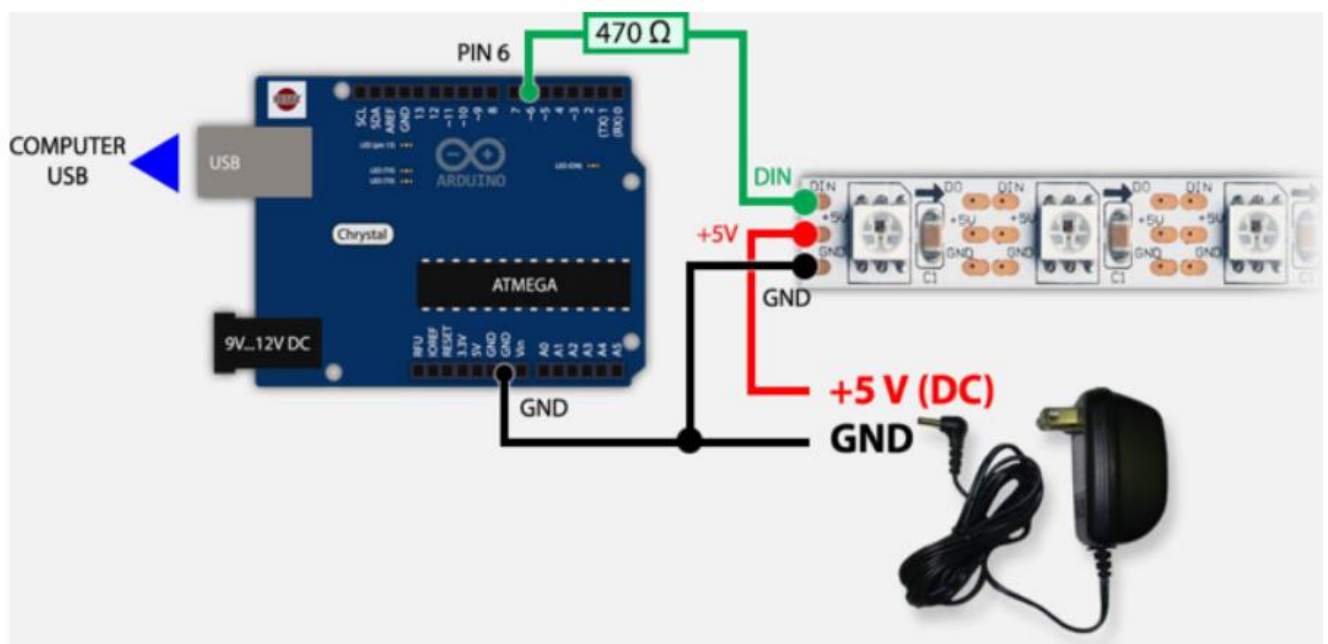


Рисунок А.4 – Структурна схема пристрою інтелектуальної підсвітки комп'ютерного монітора

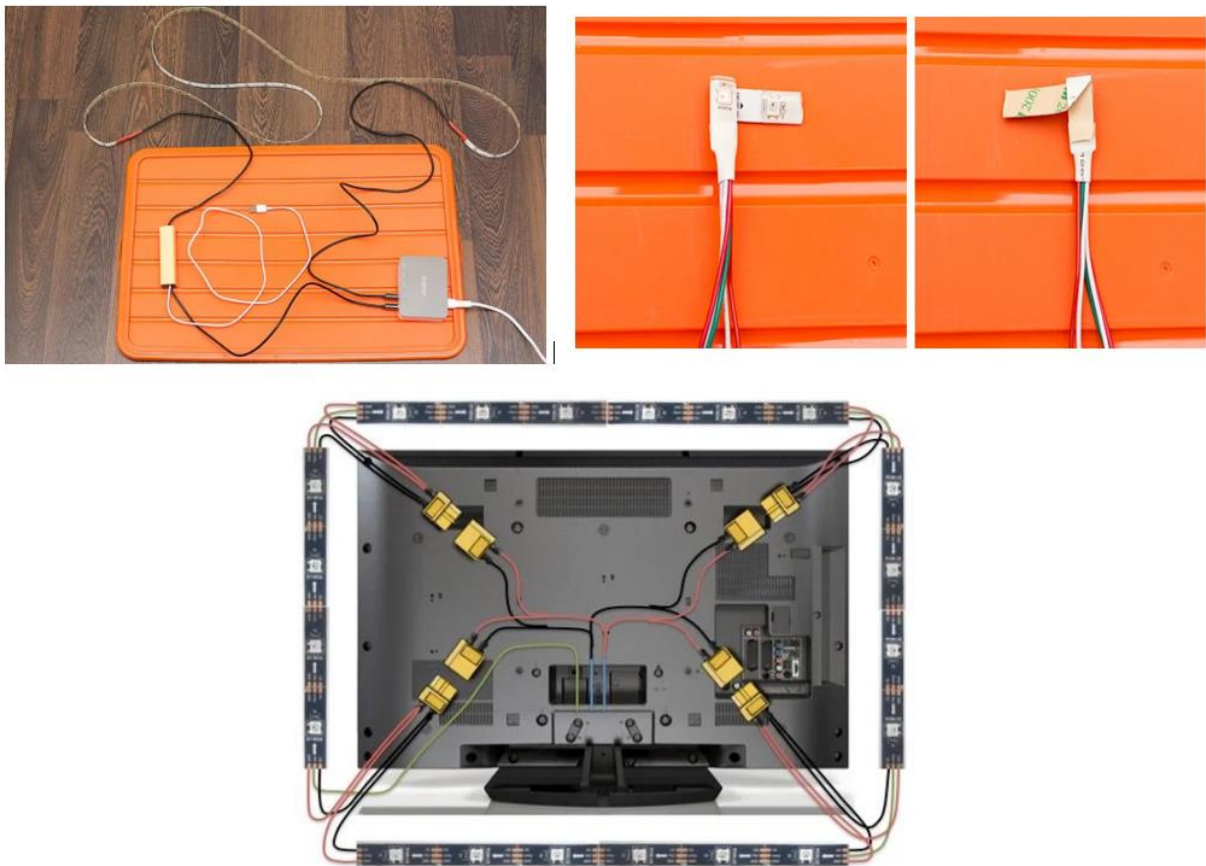


Рисунок А.5 – Етапи створення інтелектуальної підсвітки комп'ютерного монітора

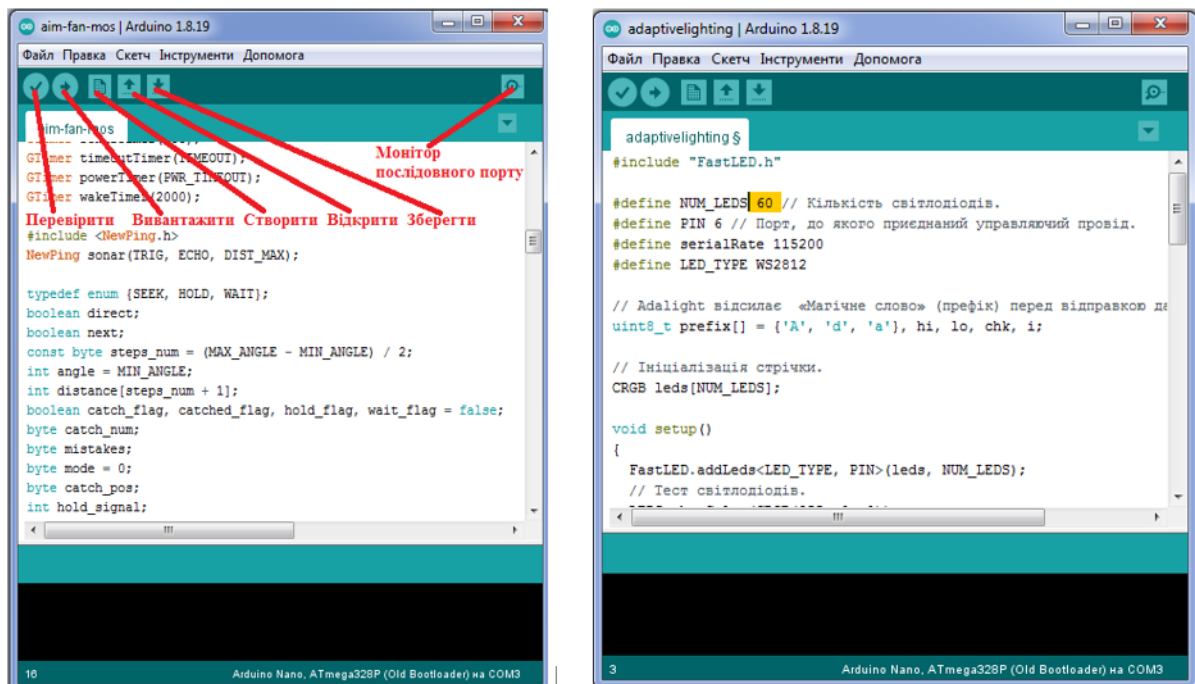


Рисунок А.6 – Програмні засоби створення інтелектуальної підсвітки комп'ютерного монітора

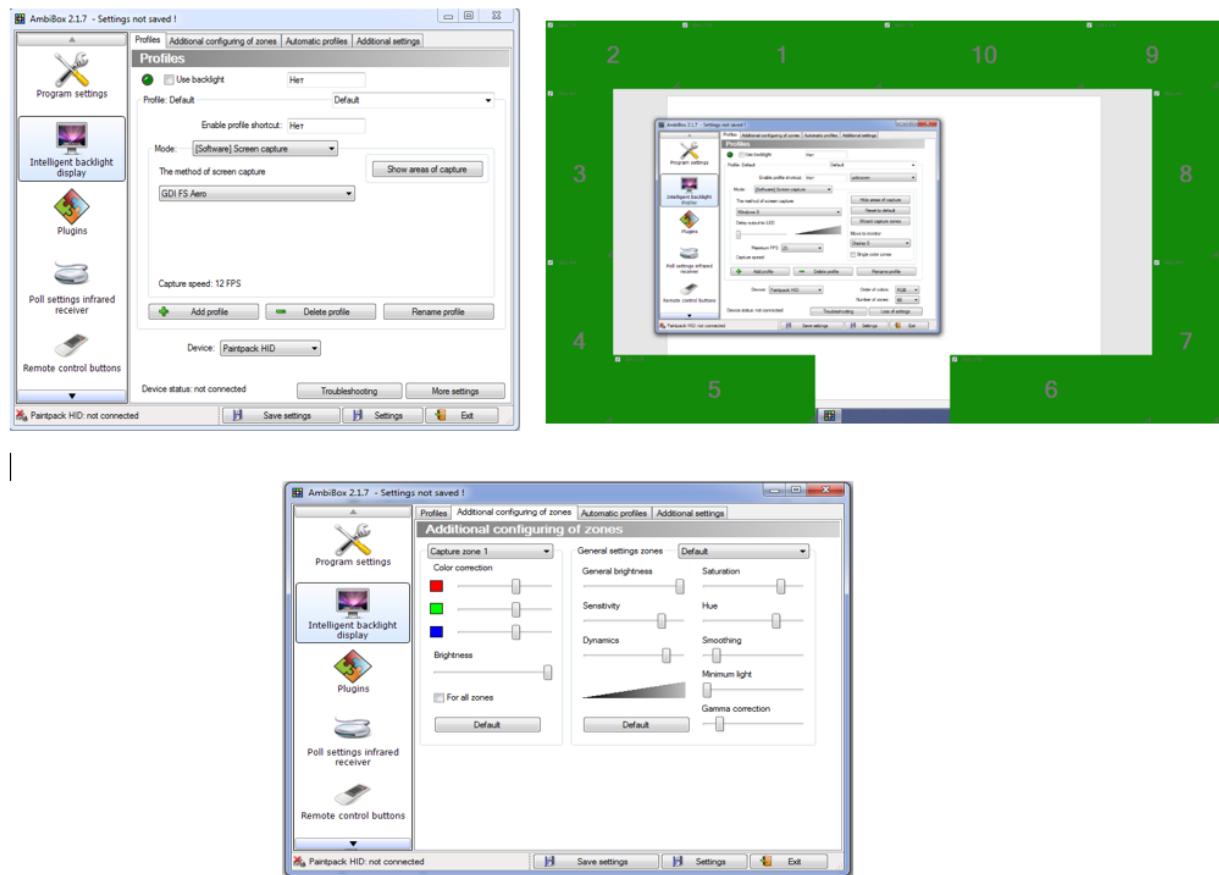


Рисунок А.7 – Програмне забезпечення AmbiBox



Рисунок А.8 – Зовнішній вигляд монітора з розробленою інтелектуальною підсвіткою

ДОДАТОК Б (обов'язковий)

Лістинг основної функції програмиого забезпечення

```
#include "FastLED.h"

#define NUM_LEDS 60 // Кількість світлодіодів
#define PIN 6 // Порт, до якого приєднаний управляючий провід
#define serialRate 115200
#define LED_TYPE WS2812

// Adalight відсилає «Магічне слово» (префік) перед відправкою даних
uint8_t prefix[] = {'A', 'd', 'a'}, hi, lo, chk, i;

// Ініціалізація стрічки
CRGB leds[NUM_LEDS];

void setup()
{
    FastLED.addLeds<LED_TYPE, PIN>(leds, NUM_LEDS);
    // Тест світлодіодів
    LEDS.showColor(CRGB(255, 0, 0));
    delay(500);
    LEDS.showColor(CRGB(0, 255, 0));
    delay(500);
    LEDS.showColor(CRGB(0, 0, 255));
    delay(500);
    LEDS.showColor(CRGB(0, 0, 0));
    Serial.begin(serialRate);
    Serial.print("Ada\n"); // Відсилаємо «магічне слово» для з'єднання з
програмою
}
```

```

void loop()
{
    // Чекаємо перший байт із слова
    for(i = 0; i < sizeof prefix; ++i)
    {
        waitLoop: while (!Serial.available());
        // Тест наступного байта із слова
        if(prefix[i] == Serial.read()) continue;
        // Інакше починаємо все спочатку
        i = 0;
        goto waitLoop;
    }
    // Чекаємо старший і молодший байт, а також контрольну суму
    while (!Serial.available());
    hi = Serial.read();
    while (!Serial.available());
    lo = Serial.read();
    while (!Serial.available());
    chk = Serial.read();
    // Якщо контрольна сума не співпала, починаємо спочатку
    if (chk != (hi ^ lo ^ 0x55))
    {
        i = 0;
        goto waitLoop;
    }
    memset(leds, 0, NUM_LEDS * sizeof(struct CRGB));
    // Отримуємо данні і налаштовуємо відповідний світлодіод
    for (uint8_t i = 0; i < NUM_LEDS; i++)
    {
        byte r, g, b;

```

```
while(!Serial.available());
r = Serial.read();
while(!Serial.available());
g = Serial.read();
while(!Serial.available());
b = Serial.read();
leds[i].r = g;
leds[i].g = r;
leds[i].b = b;
}
// Світимо новий стан стрічки
FastLED.show();
}
```

ДОДАТОК В (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Програмно-апаратний комплекс управління інтелектуальною підсвіткою для монітору

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра АІТ
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 5,12 %

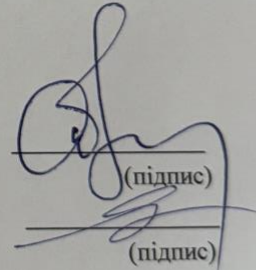
Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

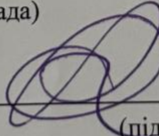
- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Олег БІСІКАЛО, зав. кафедри. АІТ
(прізвище, ініціали, посада)

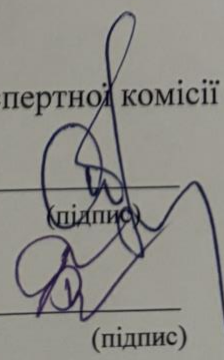
Любов ЯНЧУК, секретар кафедри АІТ
(прізвище, ініціали, посада)


(підпис)

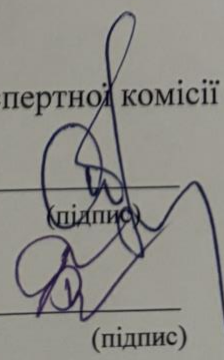
Особа, відповідальна за перевірку 
(підпис)

Костянтин ОВЧИННИКОВ
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
(підпис)

Олег БІСІКАЛО
(прізвище, ініціали, посада)

Здобувач 
(підпис)

Олександр БАЗЕЛЮК
(прізвище, ініціали)