

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему: Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega

Виконав(ла): студент(ка) IV курсу, групи СІс-44

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

	<u>Городецька Я.Р.</u> (прізвище та ініціали)
Керівник	<u>Лецишин Ю.З.</u> (прізвище та ініціали)
Нормоконтроль	<u>Тиш Є.В.</u> (прізвище та ініціали)
Завідувач кафедри	<u>Осухівська Г.М.</u> (прізвище та ініціали)
Рецензент	<u>Мудрик І.Я.</u> (прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студенту Городецькій Яніні Романівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega

Керівник роботи Лецишин Юрій Зіновійович, к.т.н., доцент кафедри КС  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 28 » 02 2023 року № 4/7-238

2. Термін подання студентом завершеної роботи \_\_\_\_\_

3. Вихідні дані до роботи Методи перетворення звукових сигналів, плата макетування Arduino Mega, бездротова передача даних

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз вимог та особливостей побудови системи віддаленого керування світлогенеруючою матрицею. 2. Проектування апаратного забезпечення системи віддаленого керування світлогенеруючою матрицею. 3. Реалізація програмного забезпечення та тестування системи віддаленого керування світлогенеруючою матрицею. 4. Безпека життєдіяльності, основи охорони праці. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Блок-схема алгоритму роботи системи

2. Структурна схема комп'ютеризованої системи

3. Схема електрично-принципова спроектованої системи

4. Фізична реалізація готової комп'ютеризованої системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., д.т.н., проф. каф. МТ</i>		

7. Дата видачі завдання 02.03.2023р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка та аналіз технічного завдання</i>	<i>02.03-05.03.2023</i>	
2	<i>Аналіз принципів функціонування систем віддаленого керування світлогенеруючою матрицею</i>	<i>06.03-14.03.2023</i>	
3	<i>Обґрунтування вибору апаратного складової комп'ютерної системи</i>	<i>15.03-23.03.2023</i>	
4	<i>Проектування структури комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею</i>	<i>24.03-03.04.2023</i>	
5	<i>Аналіз параметрів звукових сигналів у цифровому форматі та бездротового зв'язку</i>	<i>04.04-20.04.2023</i>	
6	<i>Розробка алгоритмів та програмного забезпечення для віддаленого керування світлогенеруючою матрицею</i>	<i>21.04-19.05.2023</i>	
7	<i>Розробка інструкцій з використання комп'ютерної системи за допомогою додатку</i>	<i>20.05-26.05.2023</i>	
8	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>27.05-30.05.2023</i>	
9	<i>Оформлення кваліфікаційної роботи</i>	<i>30.05-09.06.2023</i>	
10	<i>Попередній захист кваліфікаційної роботи</i>	<i>10.06-19.06.2023</i>	
11	<i>Захист кваліфікаційної роботи</i>	<i>20.06-23.06.2023</i>	

Студент

\_\_\_\_\_ (підпис)

*Городецька Яніна Романівна*

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

*Лецишин Юрій Зіновійович*

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Городецька Яніна Романівна // ТНТУ, спеціальність 123 «Комп'ютерна інженерія» // Тернопіль, 2023 // с. – 88, рис. – 20, табл. – 2, аркушів А1 – 4, бібліогр. – 16.

Ключові слова: система, звук, віддалене керування, матриця, перетворення.

Під час проєктування кваліфікаційної роботи виконано зазначеної мети щодо реалізації комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega за допомогою методів та засобів бездротового зв'язку і спектрального аналізу аналогових звукових сигналів.

Апаратне забезпечення проєкту містить Bluetooth-модуль, мікрофонний модуль з автоматичним регулюванням коефіцієнта підсилення, світлогенеруючу матрицю та мікроконтролер Arduino Mega 2560, який виконує налаштування та розподілення команд до основних компонентів. Програмне забезпечення розроблено як і для функціонування системи в цілому, так і для підтримки зв'язку між користувачем, за допомогою мови програмування C/C++ та бібліотек, які знаходяться у відкритому доступі.

## ABSTRACT

Computerized system of remote control of light-generating matrix based on Arduino Mega // Bachelor's thesis // Horodetska Yanina Romanivna // TNTU, specialty 123 «Computer engineering» // Ternopil, 2023 // p. – 88, fig. – 20, tab. – 2, posters A1 – 4, ref. – 16.

Keywords: system, sound, remote control, matrix, conversion.

During the design of the qualification work, the stated goal of implementing a computerized system of remote control of the light-generating matrix based on Arduino Mega using methods and means of wireless communication and spectral analysis of analog sound signals was fulfilled.

The hardware of the project includes a Bluetooth module, a microphone module with automatic gain adjustment, a light-generating matrix and an Arduino Mega 2560 microcontroller, which performs configuration and distribution of commands to the main components. The software is developed both for the functioning of the system as a whole, and for maintaining communication between the user, using the C/C++ programming language and libraries that are in public access.

## ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ СКОРОЧЕНЬ .....	8
ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ВИМОГ ТА ОСОБЛИВОСТЕЙ ПОБУДОВИ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ СВІТЛОГЕНЕРУЮЧОЮ МАТРИЦЕЮ.....	11
1.1 Аналіз вимог до технічного проектування систем віддаленого керування світлогенеруючою матрицею .....	11
1.1.1 Призначення системи.....	11
1.1.2 Мета створення системи.....	12
1.1.3 Основні задачі та функції об'єкту .....	13
1.1.4 Вимоги до системи.....	14
1.2 Аналіз технологій віддаленого керування.....	17
РОЗДІЛ 2 ПРОЄКТУВАННЯ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ СВІТЛОГЕНЕРУЮЧОЮ МАТРИЦЕЮ.....	20
2.1 Розробка структурної схеми комп'ютеризованої системи .....	20
2.2 Обґрунтування вибору апаратного забезпечення .....	21
2.2.1 Мікроконтролер Arduino Mega 2560 .....	21
2.2.2 Мікрофонний модуль MAX9814 .....	24
2.2.3 Bluetooth-модуль HC-05 .....	26
2.2.4 Матриця з керованих світлодіодів WS2812B.....	28
2.2.5 Фоторезистор PDV-P8103 .....	29
2.3 Реалізація алгоритму комп'ютеризованої системи .....	30
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ СВІТЛОГЕНЕРУЮЧОЮ МАТРИЦЕЮ.....	35
3.1 Обґрунтування середовища розробки.....	35
3.2 Моделювання проєкту та реалізація програмного забезпечення.....	38
3.3 Тестування спроектованої комп'ютеризованої системи.....	45
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ ....	47
4.1 Вимоги пожежної безпеки при гасінні електротехніки .....	47
4.2 Санітарно-гігієнічні вимоги до умов праці .....	48

					КС КРБ 123.038.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega	Літ.	Арк.	Аркушів
Розроб.		Городецька Я.Р.					5	2
Перевір.		Лецишин Ю.З.				ТНТУ, каф. КС, гр. СІ-41		
Реценз.		Мудрик І.Я.						
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

4.3 Вплив кольору на покращення умов праці та підвищення продуктивності виробництва .....	50
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
Додаток А.....	55
Додаток Б .....	66
Додаток В .....	68
Додаток Г .....	70

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

## ПЕРЕЛІК ОСНОВНИХ УМОВНИХ СКОРОЧЕНЬ

FFT	Fast Furier Transform
FHT	Fast Hartley Transform
DFT	Discrete Fourier Transform
IoT	Internet of Things
IIoT	Industrial internet of things
GaAs	Gallium arsenide
GaP	Gallium phosphide
AGC	Automatic Gain Control

					КС КРБ 123.038.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



## ВСТУП

У сучасному світі комп'ютеризація та автоматизація процесів знаходяться в центрі уваги. Віддалене керування системами стає все більш популярним і знайшло широке застосування в таких галузях, як промисловість, транспорт, сільське господарство, медицина та багато інших. Використання комп'ютерів та мікроконтролерів дозволяє забезпечити ефективність та точність управління, а також зменшити залежність від присутності оператора на місці.

Для віддаленого керування системами використовуються різноманітні технології і протоколи. Наприклад, можуть використовуватись бездротові з'єднання, такі як Wi-Fi, Bluetooth або мережа зв'язку мобільного оператора. Це дозволяє забезпечити зв'язок між комп'ютером або іншим пристроєм та системою, яку потрібно керувати.

Одним із цікавих напрямків у цій області є керування світлогенеруючими матрицями. Основні функції, які будуть доступні користувачеві, включають встановлення різних візуальних ефектів, зміну кольорів, регулювання яскравості та швидкості анімацій. За допомогою додаткових датчиків або модулів можна також реалізувати автоматичне керування матрицею на основі зовнішніх умов, наприклад, освітлення чи температури.

Крім функціональності, велика увага буде приділена ергономіці та зручності використання системи. Будуть розроблені інтерфейси для комп'ютера та мобільних пристроїв, що дозволять зручно керувати матрицею за допомогою інтуїтивно зрозумілих елементів управління. Це підвищить доступність та зручність навіть для користувачів без спеціальних знань у галузі програмування або електроніки. Завдяки інтерфейсам буде можливо швидко налаштувати бажані ефекти та параметри відображення.

Світлогенеруючі матриці мають широкий спектр застосувань:

– можуть бути використані у великих рекламних таблах, щоб створити яскраві, живі та виразні зображення або відеоролики. Вони здатні привернути

					КС КРБ 123.038.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

увагу перехожих, особливо в нічний час, завдяки своїм високим рівням яскравості та контрастності.

– можуть використовуватись для створення освітлення з динамічними зображеннями або анімаціями на фасадах будівель. Це може створити вражаючий ефект та додати художню атмосферу до архітектурних споруд.

– можуть бути використані на транспортних засобах, таких як автобуси, трамваї або поїзди, для відображення інформації про маршрути, зупинки, рекламні акції або навіть для створення художніх проєкцій на зовнішній поверхні транспортних засобів.

– можуть бути використані на виставках, конференціях або інших подіях, щоб створити ефектні інтерактивні експозиції, динамічні фони або візуальні ефекти, які підсилюють враження від події.

У майбутньому розвиток віддаленого керування системами ймовірно приведе до впровадження ще більш передових технологій, таких як IoT, PoT та III. Це дозволить створити розумні системи, які зможуть самостійно адаптуватись до змінних умов і оптимізувати свою роботу. Тому розробка комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею є доволі актуальною і може легко використовуватись для подальших покращень.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

# РОЗДІЛ 1 АНАЛІЗ ВИМОГ ТА ОСОБЛИВОСТЕЙ ПОБУДОВИ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ СВІТЛОГЕНЕРУЮЧОЮ МАТРИЦЕЮ

## 1.1 Аналіз вимог до технічного проектування систем віддаленого керування світлогенеруючою матрицею

### 1.1.1 Призначення системи

Основним призначенням комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею є розпізнавання звуку, що подається, та команд з віддаленого пристрою для виведення бажаної інформації.

Проектована система повинна забезпечувати коректне опрацювання сигналів та перетворення їх у відповідну ілюмінацію. Попри це вона зобов'язана пристосовуватись до навколишнього середовища, а саме фільтрувати зайві шуми для чистоти відображення і корегувати яскравість відповідно до пори доби.

Перевагами застосування такої системи віддаленого керування світлогенеруючою матрицею є довший термін працездатності та більший спектр використання завдяки своїй адаптивності.

Більшість подібних систем є широкопоширеними, тому різні варіації спроектованих рішень можна зустріти у всіх галузях. Перспективи роботи над такими проектами є забезпечення можливості постійного розширення функціоналу та автоматизації вже наявних концепцій для використання у IoT-технологіях.

У даній роботі необхідно сконструювати таку розробку, яка б підтримувала велику площу світлогенеруючої матриці, при цьому функціонувала з найменшими помилками та затратами в часі, а також була в загальному доступі для користувачів будь-якого рівня у сфері програмування електроніки.

					КС КРБ 123.038.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Городецька Я.Р.			Аналіз вимог та особливостей побудови системи віддаленого керування світлогенеруючою матрицею	Літ.	Арк.	Аркушів
Перевір.		Лецишин Ю.З.					10	9
Реценз.		Мудрик І.Я.				ТНТУ, каф. КС, гр. СІ-41		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

Проект повинен використовувати таке апаратне забезпечення, яке встановить надійний та безперебійний зв'язок з користувачем, здатне сприймати аналогові сигнали і перетворювати їх у цифрові завдяки розбиттю на спектр. Окрім цього, повинне правильно керувати та розподіляти вхідні запити без сторонніх завад.

### 1.1.2 Мета створення системи

Мета проєктування та реалізації комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею полягає у створенні розробки ефективних рішень, як на рівні апаратного, так і програмного забезпечення, які б могли задовільнити потреби довільних користувачів, а саме миттєво налаштовувати режими роботи і мати постійний доступ до системи.

Для того, щоб досягти заданої мети потрібно розв'язати наступні задачі, такі як:

- аналіз природи звукових сигналів;
- аналіз методів і засобів перетворення аналогових звукових сигналів у цифровий формат;
- аналіз технологій та протоколів бездротового з'єднання;
- визначення існуючих рішень та виокремлення саме тих, які відповідатимуть встановленим вимогам для проєктування системи;
- побудова концептуальної моделі комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею;
- обґрунтування та дослідження необхідних технічних характеристик апаратного забезпечення для проєктування системи;
- аналіз програмних засобів для роботи із звуковими сигналами, бездротовим з'єднанням та навколишнім освітленням;
- реалізація програмного забезпечення відповідно до визначених потреб;
- аналіз результатів проєктування та тестування готової спроектованої системи.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.1.3 Основні задачі та функції об'єкту

Найбільш важливі функції та завдання, які повинна забезпечувати комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega, полягають у:

- можливості отримання аналогового звукового сигналу;
- якісно відфільтровувати сторонні шуми;
- перетворювання отриманого звуку у цифровий формат;
- підтримувати постійний зв'язок між системою і пристроєм користувача по Bluetooth;
- адаптовуватись під навколишнє освітлення.

Програмне обладнання системи відповідальне за виконання різносторонніх задач, тому складається з декількох частин. Одна з них призначена для налаштування апаратного забезпечення, тоді як інша виконує управління роботи робочого процесу.

Враховуючи, що для нормального відображення будь-якого мінімального зображення або спектру, або анімації потрібно достатньо велику кількість керуючих світлодіодів, для проектування доцільно обрати мікроконтролер зі здатністю контролювати такі габарити. Також він відповідальний за звукові сигнали, тому має містити в собі аналогово-цифровий перетворювач. В той час світлогенеруюча матриця має легко програмуватись та здатна відтворювати потрібні ілюмінації.

Окрім цього, планується, що реалізація такого проєкту ще повинна містити такі компоненти апаратної складової системи, як:

- мікрофон з підсилювачем;
- фоторезистор для фіксування денного світла;
- Bluetooth-модуль для віддаленого управління;
- ПК або інший мобільний пристрій для з'єднання з Bluetooth-модулем.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

#### 1.1.4 Вимоги до системи

Загалом, до проєкту на тему «Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega» висувають перелік вимог, а саме:

- до чутливості та якості розпізнавання звукових сигналів;
- до стабільного бездротового з'єднання між користувачем та Bluetooth-модулем;
- до швидкодії та живлення процесу управління компонентами;
- до ефективності функціонування мікроконтролера;
- до продуктивності і зручності налаштування системи.

Вимоги до вибору мікрофону зумовлені його габаритами, типом з'єднання до системи, функціональністю та калібруванням підсилювання.

Мікроконтролер мусить містити достатньо технічних характеристик, зокрема тактову частоту процесора, оперативну пам'ять та аналогово-цифровий перетворювач, для відтворення всіх зазначених функцій та завдань.

Програмне забезпечення повинне бути адаптивне до апаратної складової та зручне для використання і налаштування потрібних параметрів керування світлогенеруючою матрицею.

Структура комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega повинна містити в собі як апаратне забезпечення, так і програмне.

У складі апаратної частини системи рекомендується використати такі компоненти, як:

- плата програмування Arduino Mega 2560;
- мікрофонний модуль з підсилювачем;
- фоторезистор;
- Bluetooth-модуль.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

До складу програмного забезпечення рекомендується вибрати середовище програмування мікроконтролера Arduino. Для подальших налаштувань використовується мобільний застосунок під операційну систему Android.

Комп'ютерна система має відповідати критеріям надійності для даного класу систем, а також забезпечувати захист на різноманітних етапах взаємодії між компонентами.

Основним способом зв'язку усіх компонентів у системі, як додаткових, так і основних, проводиться за допомогою провідників для забезпечення надійності передачі.

З'єднанням між смартфоном користувача та системою відбувається завдяки бездротового зв'язку за участі протоколів Bluetooth. У випадку необхідності проведення системних налаштувань з'єднання встановлюється до лінійного інтерфейсу ПК через СОМ-порт.

Діагностичні заходи, такі як загальний огляд цілісності компонентів комп'ютеризованої системи повинні проходити відповідно до зазначених графіків та інструкцій з експлуатації модулів.

Діагностика цілісності програмного забезпечення проводиться кожного разу при ввімкненні системи перед виконанням основних функцій. При виявленні збоїв у роботі система повідомляє про знайдені неточності, тому необхідно негайно усунути всі перешкоди до повноцінного функціонування.

Перспективами розвитку даної комп'ютеризованої системи є імплементація як додатковий модуль у існуючу більшу систему для загального чи особистого користування або використання у різноманітних галузях з більшою або меншою кількістю керуючих світлодіодів матриці чи інших компонентів.

Також система підлягає модернізації у випадку недостатньої ефективності або для того, щоб доповнити функціональність, яка би більше підходила під вимоги та потреби користувача.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

До необхідних складових надійності комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega належать:

- стабільність працездатності протягом визначеного часового періоду;
- стабільне функціонування мікроконтролера Arduino Mega 2560;
- середній наробіток між відмовами становить до 50000 годин;
- здатність пристосовуватися до нових звукових сигналів.

Вимоги до функціонування та виконання конкретних задач, які виконує комп'ютеризована система, включають в себе:

- можливість одержання аналогового звукового сигналу;
- здатність перетворювати одержаний звук у цифровий вигляд;
- здатність фільтрувати звук від шумів;
- можливість розбивати сигнал на смуги;
- можливість зчитувати денну яскравість;
- здатність створювати бездротовий зв'язок між користувацьким

пристроєм по Bluetooth;

- можливість налаштування підсилення звуку;
- здатність віддаленого налаштування світлогенеруючої матриці.

Апаратне забезпечення комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega обов'язково повинне містити:

- плату Arduino Mega з мікроконтролером ATmega 2560;
- мікрофонний модуль із підсилювачем;
- Bluetooth-модуль;
- світлогенеруючу матрицю;
- фоторезистор;
- ПК та/або смартфон.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Для зв'язку по Bluetooth вимагається використовувати мобільний пристрій на базі операційної системи Android 7.0 і вище. Специфікації ПК, необхідні для конфігурації комп'ютерної системи, обробки команд мають бути у відповідності з такими технічними характеристиками:

- частота процесора не менше 2 ГГц;
- дисковий простір не менше 100 Гб;
- обсяг оперативної пам'яті 8 ГБ і більше.

Щоб реалізувати програмне забезпечення для Arduino Mega, потрібно використовувати таке середовище, яке б могло забезпечити зв'язок між всіма компонентами структури системи. Для цього доцільним вирішенням буде скористатись Arduino IDE за допомогою мови програмування C/C++.

## 1.2 Аналіз технологій віддаленого керування

У сучасному світі, де вирішальними факторами є ефективність, екологічність та сталість розвитку, технології постійно змінюються, адаптуючись до нових викликів. Одним з важливих напрямків технологічних змін є розвиток нових джерел світла, що забезпечують більше ефективності та безпеки, ніж традиційні лампи розжарювання.

Однією з таких технологій є світлогенеруючі матриці - високотехнологічні системи, які використовують різні типи світлодіодів для створення світла. Вони обіцяють переваги в енергоефективності, довговічності, гнучкості використання та зменшенні впливу на навколишнє середовище.

Перші в історії світлодіоди були створені на основі GaAs, GaP та інших напівпровідникових сплавів у 1961 році. Їх почали використовувати, а потім виготовляти комерційно, як індикатори двоє працівників компанії Texas Instruments Боб Б'ярд і Гарі Пітман. Наступним інноватором став Нік Голоняк, який зі своєю групою розробив перший світлодіод, який працює у видимому для людини діапазоні.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Після цього еволюція світлодіодів розвилась і до нашого часу, де стало доступно об'єднувати їх в стрічки чи матриці та застосовувати в набагато більших масштабах.

Можливість зобразити потрібну ілюмінацію чи текст на матриці значно розширювала перспективи використання, але зі збільшенням потреб виникали нові труднощі, а саме в керуванні подібними системами. Тому згодом стало доступно налаштовувати пристрої віддалено.

Розглянемо кілька загальних технологій що використовуються для віддаленого керування світловими системами.

Wi-Fi – це бездротова технологія, яка використовує радіохвилі для надання мережевого з'єднання, доступу до Інтернету та обміну даними між різними пристроями. Вона зазвичай використовується в межах будівель, хоча може мати значний радіус дії на відкритому просторі.

DMX, або Digital Multiplex, - це протокол, який зазвичай використовується для контролю освітлення в театрах, концертних залах і нічних клубах. DMX контролери відправляють сигнали до освітлювальних приладів, щоб керувати їхніми параметрами, такими як яскравість, колір, швидкість обертання тощо.

Z-Wave та Zigbee - це два стандарти бездротового з'єднання, які часто використовуються в пристроях "розумного дому". Обидва стандарти дозволяють пристроям комунікувати між собою та з центральним контролером, дозволяючи автоматизувати різні процеси в будинку, такі як управління освітленням, температурою, безпекою та іншими системами. Z-Wave використовує радіохвилі, а Zigbee - технологію меш-мережі для встановлення зв'язку між пристроями.

Останньою із відомих та надійних технологій бездротового зв'язку є Bluetooth, яка є найкращим рішенням для віддаленого керування майбутньої системи. Ця технологія була розроблена для обміну даними між різними пристроями, що знаходяться на невеликій відстані один від одного. Ім'я технології походить від давнього скандинавського короля Харальда "Bluetooth"

					КС КРБ 123.038.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Гормссона, який об'єднав Данію і Норвегію. Як і король, технологія Bluetooth об'єднує різні пристрої.

Основні аспекти технології Bluetooth:

– Bluetooth дозволяє різним пристроям спілкуватися між собою через короткодіапазову радіосистему.

– Bluetooth працює в частотному діапазоні 2,4 ГГц, який є вільним для використання в більшості країн світу.

– Bluetooth LE (Low Energy) було розроблено для систем, що вимагають низького споживання енергії. Це робить Bluetooth ідеальним для пристроїв, які працюють від батарей.

– Bluetooth включає набір "профілів", що визначають, як пристрої можуть спілкуватися один з одним. Наприклад, профіль гарнітури (HSP) визначає, як бездротова гарнітура може спілкуватися з мобільним телефоном.

– Стандартна дальність дії Bluetooth зазвичай становить близько 10 метрів, але з версією 5.0 та його режимом дальньої дії це може бути значно більше. На офіційному сайті доступний інструмент визначення діапазону відстані між двома пристроями залежно від характеристик та умов поширення.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2 ПРОЄКТУВАННЯ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ СВІТЛОГЕНЕРУЮЧОЮ МАТРИЦЕЮ

### 2.1 Розробка структурної схеми комп'ютеризованої системи

Як і зазначалось раніше, для досягнення мети майбутньої комп'ютеризованої системи, в її структурі пропонується використати такі компоненти, як:

- Arduino Mega 2560;
- мікрофонний модуль;
- Bluetooth-модуль;
- матриця з керованих світлодіодів;
- фоторезистор.

Для того, щоб майбутня система коректно функціонувала, при з'єднанні основних компонентів потрібно слідувати таким правилам:

- мікрофонний модуль та фоторезистор під'єднати до аналогових портів мікроконтролера;
- світлогенеруючу матрицю під'єднати до цифрового порта мікроконтролера;
- Bluetooth-модуль під'єднати до послідовних портів мікроконтролера, які будуть з'єднуватись за протоколом SPI.

Детально про особливості підключення розглянемо в наступному розділі після вибору апаратного забезпечення.

Переглянути як виглядає готова спроектована схема структури комп'ютеризованої системи можна на рис. 2.1.

					КС КРБ 123.038.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проекування апаратного забезпечення системи віддаленого керування світлогенеруючою матрицею	Літ.	Арк.	Аркушів
Розроб.		Городецька Я.Р.						
Перевір.		Лецишин Ю.З.					19	15
Реценз.		Мудрик І.Я.				ТНТУ, каф. КС, гр. СІ-41		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

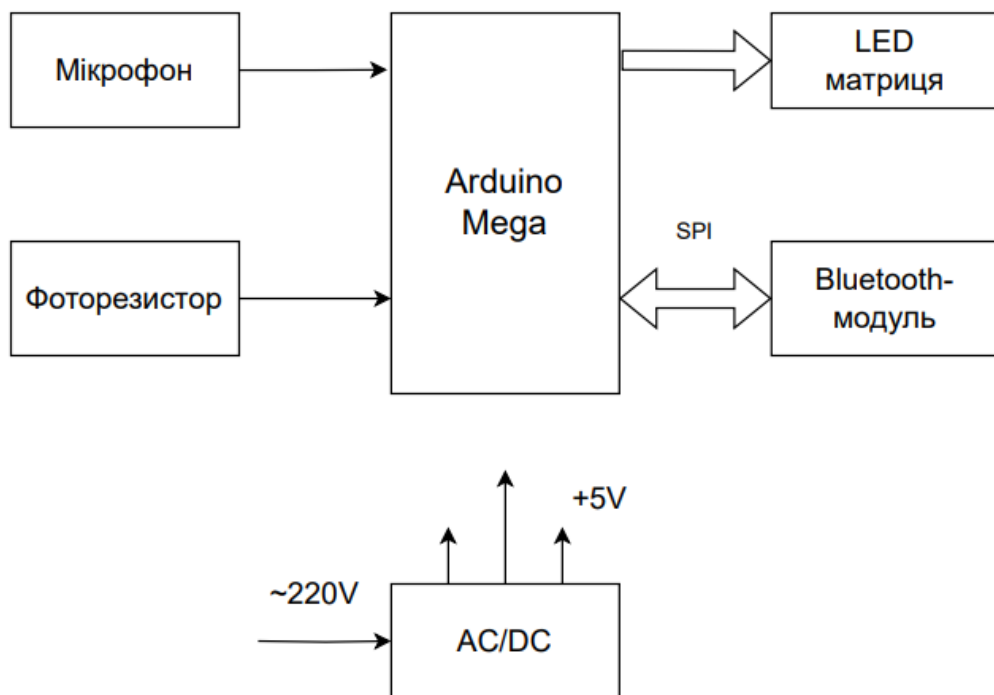


Рисунок 2.1 – Структурна схема спроектованої системи

Далі визначимо які саме компоненти підходять під сформовану структуру системи.

## 2.2 Обґрунтування вибору апаратного забезпечення

### 2.2.1 Мікроконтролер Arduino Mega 2560

Головним компонентом всієї системи є мікроконтролерна плата Arduino Mega 2560. З її допомогою відбувається зв'язок та управління всіх модулів та перетворення звукового сигналу в цифровий. Вона виробляється компанією Arduino, яка займається створенням відкритого апаратного забезпечення та програмного забезпечення для розробки та прототипування електронних пристроїв. Плата може живитися як від USB-порту, так і від зовнішнього джерела живлення. Вхідна напруга може вар'юватися від +7 до +12 V [1].

Основні технічні характеристики плати наведено у табл. 2.1.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Основні технічні характеристики Arduino Mega 2560

Характеристика	Значення
Вбудований LED-пін	13
Цифрові I/O піни	54
Аналогові вхідні піни	16
PWM піни	15
Серійна комунікація (UART)	4
I2C та SPI	Так
Напруга живлення I/O	+5 V
DC струм на I/O пін	20 mA

Вигляд самої плати зображено на рис. 2.2.



Рисунок 2.2 – Вигляд мікроконтролерної плати Arduino Mega 2560

Як видно на рисунку, ATmega2560 є основним мікроконтролером на платі. Він є 8-бітним мікроконтролером з родини AVR, вироблений Atmel та вперше представлений у 2006 році. Із особливостей ATmega2560 можна відзначити наявність внутрішнього кварцового генератора, можливість роботи від широкого діапазону напруг живлення (+1.8-5.5 V), а також наявність різних режимів енергозбереження.

Усі виводи ATmega2560 показано на рис. 2.3.

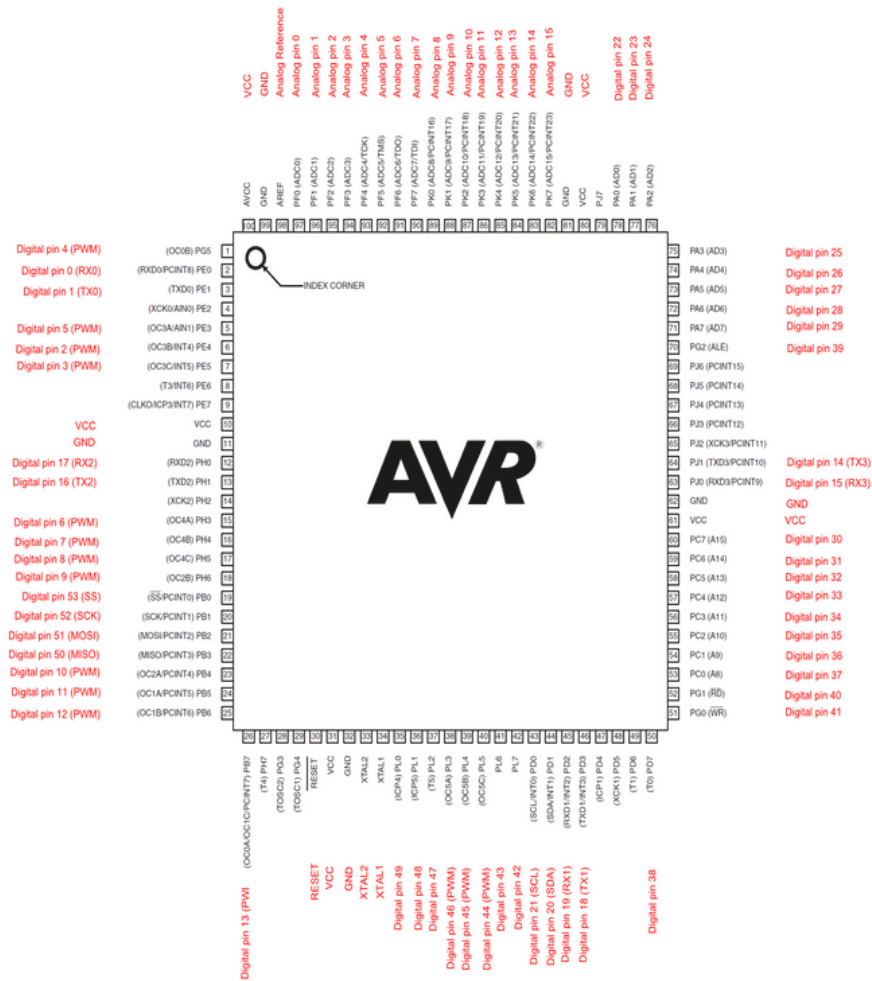


Рисунок 2.3 – Виводи мікроконтролера ATmega2560

Основні технічні характеристики мікроконтролера ATmega2560 наведено в табл. 2.2.

Таблиця 2.2 – Основні технічні характеристики мікроконтролера ATmega2560

Характеристика	Значення
Архітектура	8-бітна AVR
Частота тактування	до 16 МГц
Флеш-пам'ять	256 КБ
EEPROM	4 КБ

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

КС КРБ 123.038.00.00 ПЗ

Арк.

22

Продовж. табл. 2.2

Характеристика	Значення
SRAM	8 КБ
GPIO (вхід/вихід)	86 портів
АЦП	16 каналів
16-бітні таймери	4
8-бітні таймери	2
Інтерфейси	USART, SPI, I2C
UART	4

Згідно теми кваліфікаційної роботи саме цей мікроконтролер здатний виконати всі визначені до системи завдання. Перейдемо до наступного компонента.

### 2.2.2 Мікрофонний модуль MAX9814

Як і описувалось раніше, ще одним важливим компонентом комп'ютеризованої системи є мікрофонний модуль з підсилювачем. Для описаних завдань було обрано MAX9814.

MAX9814 – це мікросхема мікрофонного підсилювача з автоматичним контролем затухання, вироблена компанією Maxim Integrated. Цей модуль є високоякісним, низькошумовим мікрофонним підсилювачем, який ідеально підходить для використання в портативних пристроях, таких як мобільні телефони, ноутбуки та інші мобільні пристрої [2].

Як виглядає сама мікросхема представлено нижче на рис. 2.4.





Рисунок 2.4 – Мікрофонний модуль MAX9814

Так як цей модуль обрано з метою якісно розпізнавати звукові сигнали, розглянемо детальніше підсилювальні властивості MAX9814.

Він має три режими підсилення, які вибираються шляхом налаштування вхідного піну, що відомий як GAIN. Перечислимо ці режими

– 40dB Gain є найнижчим режимом підсилення MAX9814. В цьому режимі, підсилювач використовується для невеликих сигналів, які потребують значного підсилення для нормальної роботи. Пін GAIN повинен бути підключений до VDD для вибору цього режиму.

– 50dB Gain є середнім рівнем підсилення MAX9814. Він використовується для сигналів, які потребують помірною підсилення. Щоб вибрати цей режим, пін GAIN повинен бути підключений до землі.

– 60dB Gain є найвищим режимом підсилення MAX9814. Він використовується для дуже слабких сигналів, які потребують великого підсилення для нормальної роботи. Щоб вибрати цей режим, пін GAIN повинен бути не під'єднаний.

Важливо зазначити, що MAX9814 має AGC, який автоматично налаштовує підсилення в залежності від вхідного сигналу. Це означає, що незалежно від вибраного режиму підсилення, AGC автоматично налаштує підсилення для оптимального використання динамічного діапазону [3].

					КС КРБ 123.038.00.00 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Як саме виглядає підсилення на графіку вхідних та вихідних сигналів зображено на рис. 2.5.

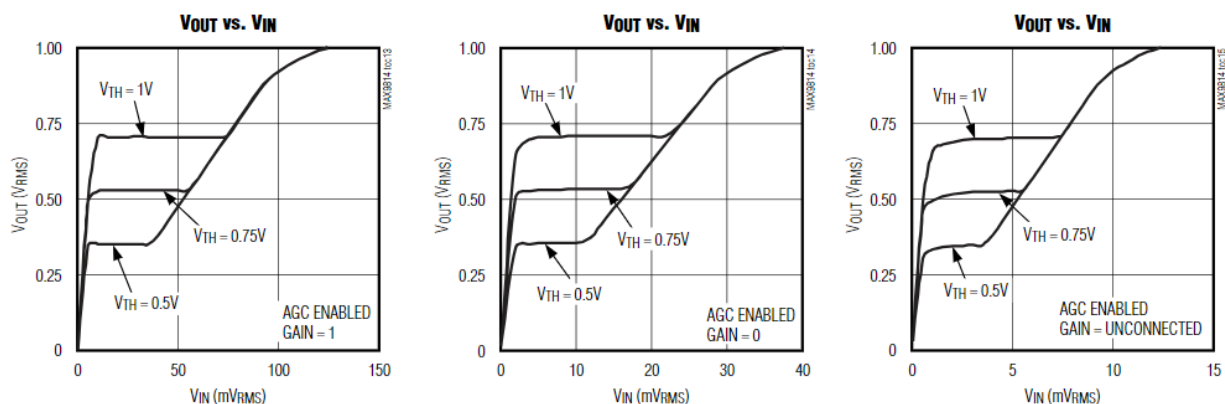


Рисунок 2.5 – Вплив підсилення на вхідні та вихідні сигнали

Отже, дослідивши особливості даного мікрофонного модуля, прийнято рішення додати його до системи. Розглянемо наступний компонент.

### 2.2.3 Bluetooth-модуль HC-05

Наступна не менш важлива складова апаратної частини системи є Bluetooth-модуль HC-05. Це периферійний пристрій, який дозволяє створювати бездротові з'єднання між двома пристроями. Цей модуль часто використовується в проєктах, заснованих на Arduino, для створення бездротового з'єднання між Arduino і смартфоном або іншим пристроєм Bluetooth. Зовнішній вигляд модуля зображено на рис. 2.6.

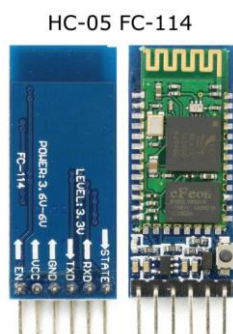


Рисунок 2.6 – Зовнішній вигляд Bluetooth-модуля HC-05

Разом з цим модулем серед технічних спеціалістів популярний ще одна схожа схема під назвою HC-06. Розглянемо основні відмінності між ними:

– HC-05 може працювати як в режимі master, так і в режимі slave. Це означає, що HC-05 може ініціювати з'єднання з іншими пристроями Bluetooth або приймати вхідні з'єднання. З іншого боку, HC-06 працює тільки в режимі slave і може тільки приймати вхідні з'єднання.

– HC-05 має більше опцій конфігурації, включаючи швидкість передачі даних, режим роботи, пароліні налаштування тощо. HC-06 має менше опцій для налаштування.

– Через свою здатність працювати в режимі master і slave, HC-05 може бути сумісним з більшою кількістю пристроїв, ніж HC-06.

– Зазвичай HC-05 є трохи дорожчим, ніж HC-06, через свої додаткові можливості.

– Обидва модулі використовують Bluetooth 2.0, який вже вважається дещо застарілим, але все ще широко використовується для простих проєктів з бездротового з'єднання.

Отже, враховуючи перелічені відмінності, доходимо висновку, що HC-05 переважає над HC-06, адже має більший функціонал. Розглянемо технічні особливості обраного модуля.

HC-05 працює з напругою від +3,6 до +6 V, але логічні рівні TTL – це +3,3 V. Це означає, що ви можете живити модуль напругою до +6 V, але при передачі даних між модулем і мікроконтролером, як-то Arduino, вам потрібно використовувати +3,3 V логічний рівень. Передача даних відбувається зі швидкістю від 9600 до 1382400 baud [5].

HC-05 може працювати в двох режимах: командний режим та режим зв'язку. В командному режимі модуль приймає AT-команди для налаштування параметрів модуля. В режимі зв'язку модуль може передавати та отримувати дані через Bluetooth. Для того, щоб ідентифікувати в якому режимі зараз знаходиться модуль, достатньо звернути увагу на його світлодіод. Якщо HC-05 перебуває в

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

командному режимі, світлодіод загоряється лише раз кожні 2 секунди. Якщо HC-05 знаходиться в режимі зв'язку та очікує на з'єднання, тоді світлодіод швидко загоряється і затухає. Якщо з'єднання встановлено, світлодіод загоряється по два рази кожні 2 секунди [6].

#### 2.2.4 Матриця з керованих світлодіодів WS2812B

WS2812B – це тип світлодіоду, який включає в себе вбудований мікроконтролер, що дозволяє йому бути програмованим. Колір кожного світлодіода визначається шляхом зміни значень червоного, зеленого та синього (RGB) каналів світлодіоду. Ці канали можуть бути налаштовані на будь-яке значення від 0 до 255, що дозволяє створювати більше 16 мільйонів можливих кольорів.

Загальний вигляд матриці можна переглянути на рис. 2.7.

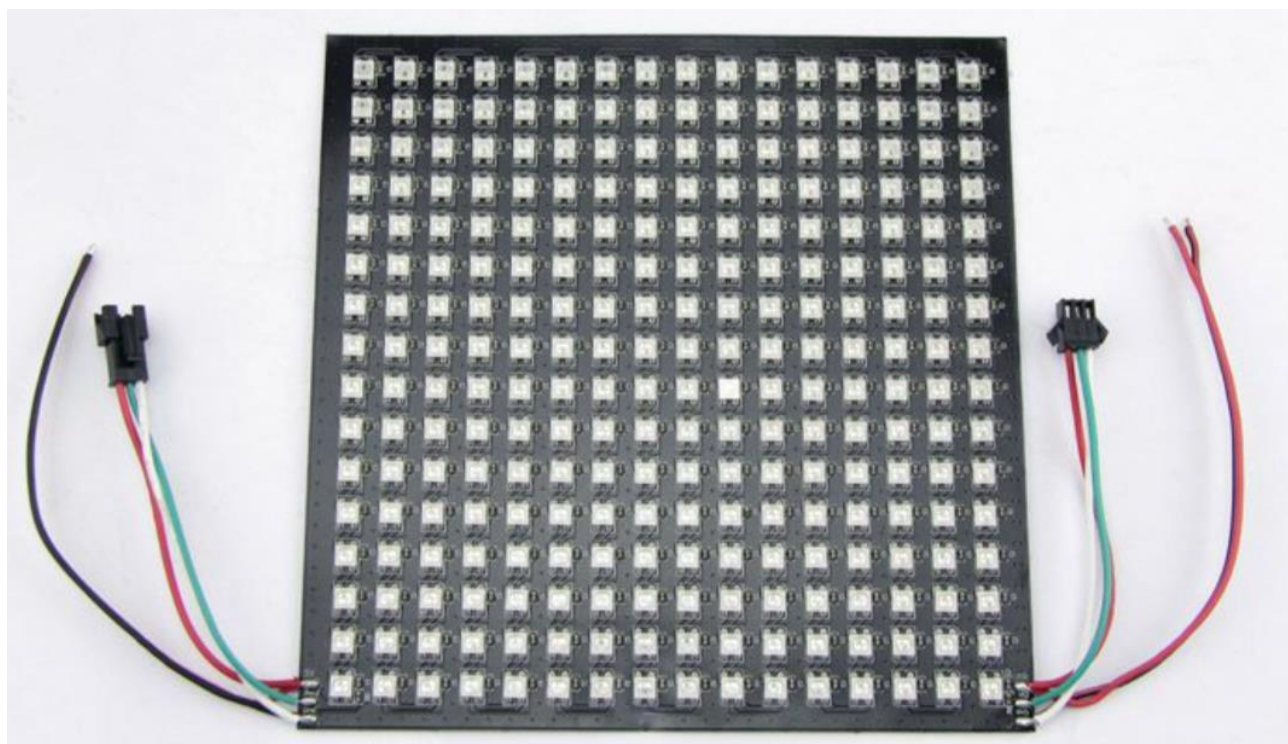


Рисунок 2.7 – Загальний вигляд матриці з керованих світлодіодів WS2812B

Кожен світлодіод в масиві має розмір 5050 (5мм x 5мм), включаючи керуючий мікročип та RGB чіп в одному пакеті. Світлодіоди розташовані так, що є 16 світлодіодів на лінію, і 16 ліній на панелі. Відстань між кожним світлодіодом становить 1 см.

У світлодіодів є наступні параметри:

- Синій: довжина хвилі 465-467 нм, світлова інтенсивність 180-200 mcd, робоча напруга +3.0-3.4 V

- Зелений: довжина хвилі 522-525 нм, світлова інтенсивність 660-720 mcd, робоча напруга +3.0-3.4 V

- Червоний: довжина хвилі 620-625 нм, світлова інтенсивність 390-420 mcd, робоча напруга +2.0-2.2 V

В обраній матриці світлодіоди з'єднані зигзагоподібно. Також метод з'єднання є послідовним, що зображено на рис. 2.8.

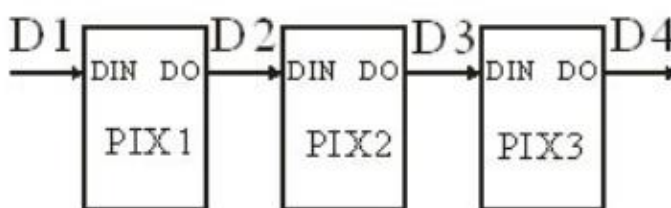


Рисунок 2.8 – Метод з'єднання світлодіодів

Отже, доходимо висновку, що саме ці світлодіоди є практичними і задовільняють визначені потреби. Розглянемо останній головний компонент системи.

### 2.2.5 Фоторезистор PDV-P8103

Фоторезистор PDV-P8103 є кадмієвим сульфідним (CdS) фотопровідним фоторезистором, розробленим для виявлення світла в діапазоні від 400 до 700 нм. Ці світлозалежні резистори доступні в широкому діапазоні значень опору і упаковані в двопровідний керамічний корпус, покритий пластиком.

Загальний вигляд фоторезистора зображено на рис. 2.9.

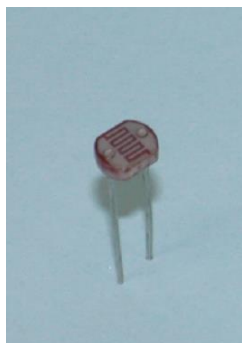


Рисунок 2.9 – Загальний вигляд фоторезистора PDV-P8103

Основні характеристики фоторезистора PDV-P8103 включають:

- максимальна прикладна напруга: +150 V;
- постійна потужність розсіювання: 100 мВт/°С;
- діапазон робочих та зберігальних температур: від -30 до +75 °С;
- температура паяння: до +260 °С.

Він також має наступні специфічні характеристики при тестуванні:

- темний опір (після 10 секунд при 10 люксах при 2856 °К): 0,5 МОм;
- освітлений опір (при 10 люксах при 2856 °К): від 16 до 33 КОм;
- чутливість: 0,75 Вт/люкс;
- діапазон спектрального застосування: від 400 до 700 нм;
- пік спектрального застосування: 520 нм;
- час зростання: 60 мс;
- час спаду: 25 мс.

### 2.3 Реалізація алгоритму комп'ютеризованої системи

Після того як апаратне забезпечення задовільнило всі зазначені вимоги, наступним кроком проектування системи є складання алгоритму. Щоб майбутній

					КС КРБ 123.038.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

проект зміг реалізувати всі задачі, які від нього вимагають, прийнято рішення розділити систему на два режими.

Один із режимів буде відповідати за виконання функцій аналізатора спектру, а саме за допомогою мікрофону вловлювати звук та відображати його на матриці у вигляді певних смуг. Щоб налаштувати чутливість сприймання звуку, використаємо потенціометр.

Другий режим відповідатиме за зв'язок між Bluetooth-модулем та користувацьким пристроєм. В мобільному додатку користувач зможе надсилати спеціальні команди за допомогою яких на певний час відображатиметься певний режим ілюмінації.

Як описувалось раніше в попередніх розділах, додатково матриця зможе підлаштовувати свою яскравість під пору доби за допомогою фоторезистора. З його допомогою система буде приймати певне значення навколишнього освітлення та налаштовувати яскравість матриці, надаючи їй протилежне значення.

Далі розглянемо детальніше процес відображення ілюмінації та методи реалізації світломузики.

Світломузика - це використання освітлення та/або відео для доповнення живої музики або запису. Світломузика може бути використана для створення синхронізованого шоу, в якому зміни в освітленні або відео відповідають ритму, темпу або іншим елементам музики. Але для того, щоб поєднати музику та зображення, яке має йому відповідати, проєктована система, зазвичай, повинна містити мікрофон. Він в свою чергу сприймає вхідні звукові сигнали як аналогові, подає на АЦП, де за допомогою FFT, звук перетворюється в цифровий сигнал.

FFT – це алгоритм, який використовується для виконання DFT більш ефективно, ніж було б можливо, використовуючи прямий підхід до DFT. DFT – це математичний інструмент, що перетворює послідовність з комплексних чисел (сигнал в часовому просторі) на іншу послідовність комплексних чисел (сигнал

					КС КРБ 123.038.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

в частотному просторі). Це допомагає розглянути сигнал в частотному домені, який може бути більш зручним для аналізу в ряді ситуацій.

Прямий підхід до обчислення DFT вимагає великої кількості обчислень, в порядку  $N^2$ , де  $N$  – кількість точок у вхідному сигналі. FFT здатний виконати це ж саме перетворення, використовуючи значно менше обчислень, в порядку  $N \log(N)$ .

Принцип дії FFT можна розбити на наступні етапи:

– Вхідний сигнал розбивається на дві підпоследовності: одну для парних індексів і одну для непарних індексів, використовуючи властивість періодичності DFT.

– FFT застосовується рекурсивно до кожної з цих підпоследовностей, використовуючи властивість симетрії DFT.

– Результати з кожної підпоследовності комбінуються, щоб отримати кінцевий результат, знову використовуючи властивість симетрії DFT.

Цей процес продовжується рекурсивно, розбиваючи кожну підпоследовність на ще менші підпоследовності, поки не буде досягнуто одного елемента. В результаті, кількість обчислень зменшується з  $O(N^2)$  до  $O(N \cdot \log N)$ , де  $N$  – кількість точок у DFT.

Важливо зазначити, що хоча FFT значно зменшує кількість обчислень, він дає точно такий же результат, як і DFT. Це означає, що FFT можна використовувати для обчислення DFT без втрати точності.

Після того, як зі звукового сигналу було отримано спектр частот, його розбивають на смуги і візуалізують на матриці або дисплеї. Таким чином, одержуємо один із методів створення світломузики.

Проаналізувавши перелік задач, які повинна виконувати комп'ютеризована система, було складено блок-схему алгоритму функціонування. На рис. 2.10 зображено підпрограму, яка відповідає за алгоритм відображення ілюмінації в аналізаторі спектру, а основний алгоритм представлено на рис. 2.11.





Рисунок 2.10 – Блок-схема алгоритму відображення ілюмінації

Змн.	Арк.	№ докум.	Підпис	Дата

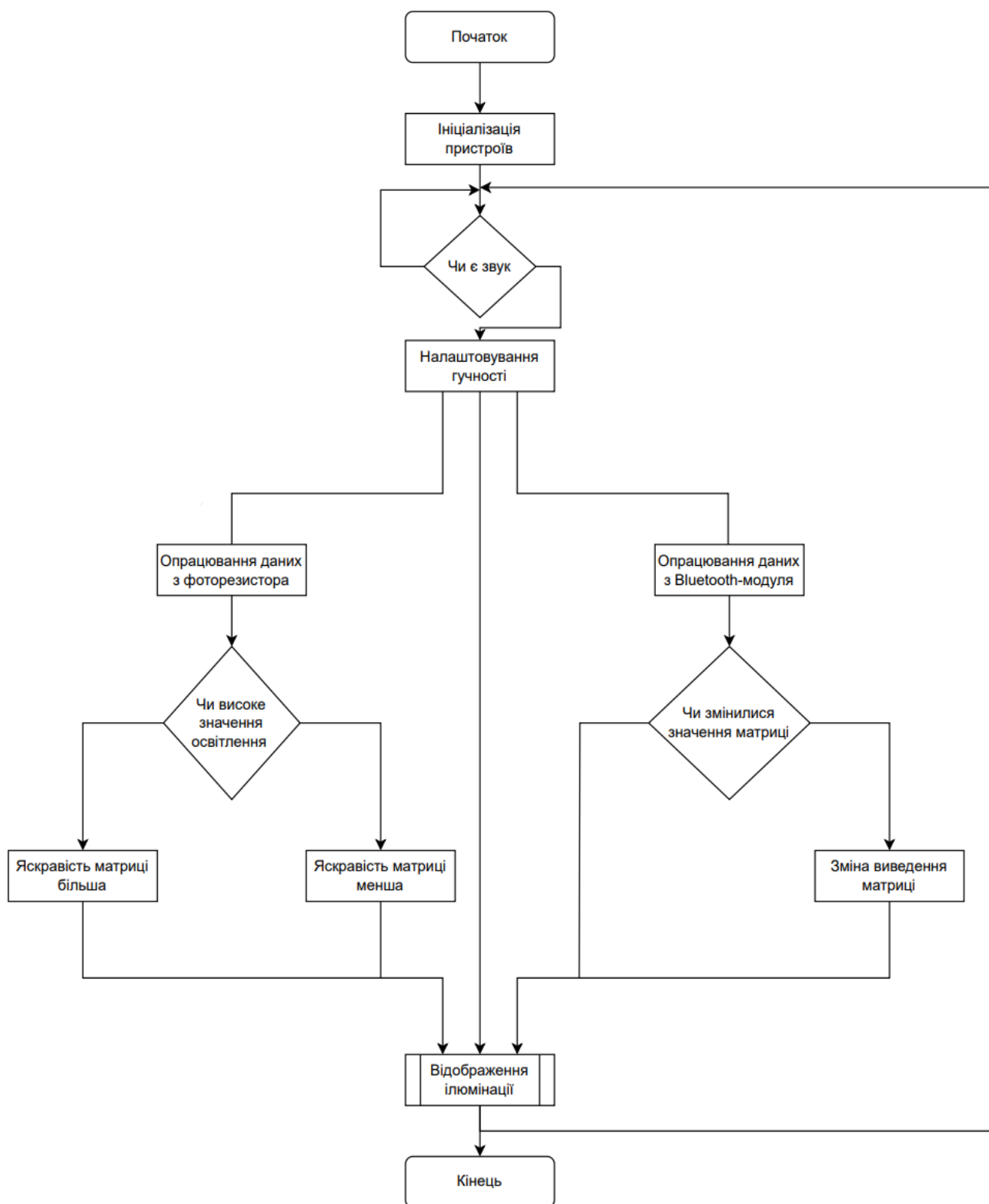


Рисунок 2.11 – Блок-схема основного алгоритму системи

Після обґрунтування апаратного забезпечення та визначення принципу роботи системи можна переходити до реалізації самого проєкту.

Змн.	Арк.	№ докум.	Підпис	Дата

## РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ СВІТЛОГЕНЕРУЮЧОЮ МАТРИЦЕЮ

### 3.1 Обґрунтування середовища розробки

Наступним кроком після розробки алгоритму доцільним буде реалізовувати саме програмне забезпечення. Найкращим варіантом для середовища розробки є Arduino IDE.

Arduino IDE (Integrated Development Environment) – це основний засіб розробки для мікроконтролерів Arduino. Це програмне забезпечення, що дає можливість писати код, компілювати його і завантажувати на плати Arduino. Він був спеціально розроблений, щоб люди, які тільки починають свою співпрацю з мікроконтролерами, змогли проходити цей шлях легко, але все ще має всі необхідні функції для більш досвідчених розробників.

Arduino IDE містить в собі всі необхідні нам характеристики:

- текстовий редактор;
- компіляція та завантаження;
- підтримка бібліотек;
- серійний монітор;
- підтримка різних плат Arduino;
- скетчі.

Arduino IDE доступний для Windows, Mac OS X, Linux та є у відкритому доступі для завантаження. На даний момент існують дві версії Arduino IDE: 1.x.x та 2.x. IDE 2.x – це нова головна версія, яка працює швидше і має більш потужні можливості, ніж IDE 1.x.x. Вона має більш сучасний редактор і більш чутливий

					<b>КС КРБ 123.038.00.00 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Городецька Я.Р.			Реалізація програмного забезпечення та тестування системи віддаленого керування світлогенеруючою матрицею	Літ.	Арк.	Аркушів
Перевір.		Лецишин Ю.З.					34	12
Реценз.		Мудрик І.Я.				ТНТУ, каф. КС, гр. СІ-41		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

інтерфейс, а також включає розширені функції для допомоги користувачам з кодуванням і відлагодженням. Для розробки програмного забезпечення доцільніше буде скористуватись новішою версією 2.x. Загальний вигляд середовища розробки зображено на рис. 3.1.

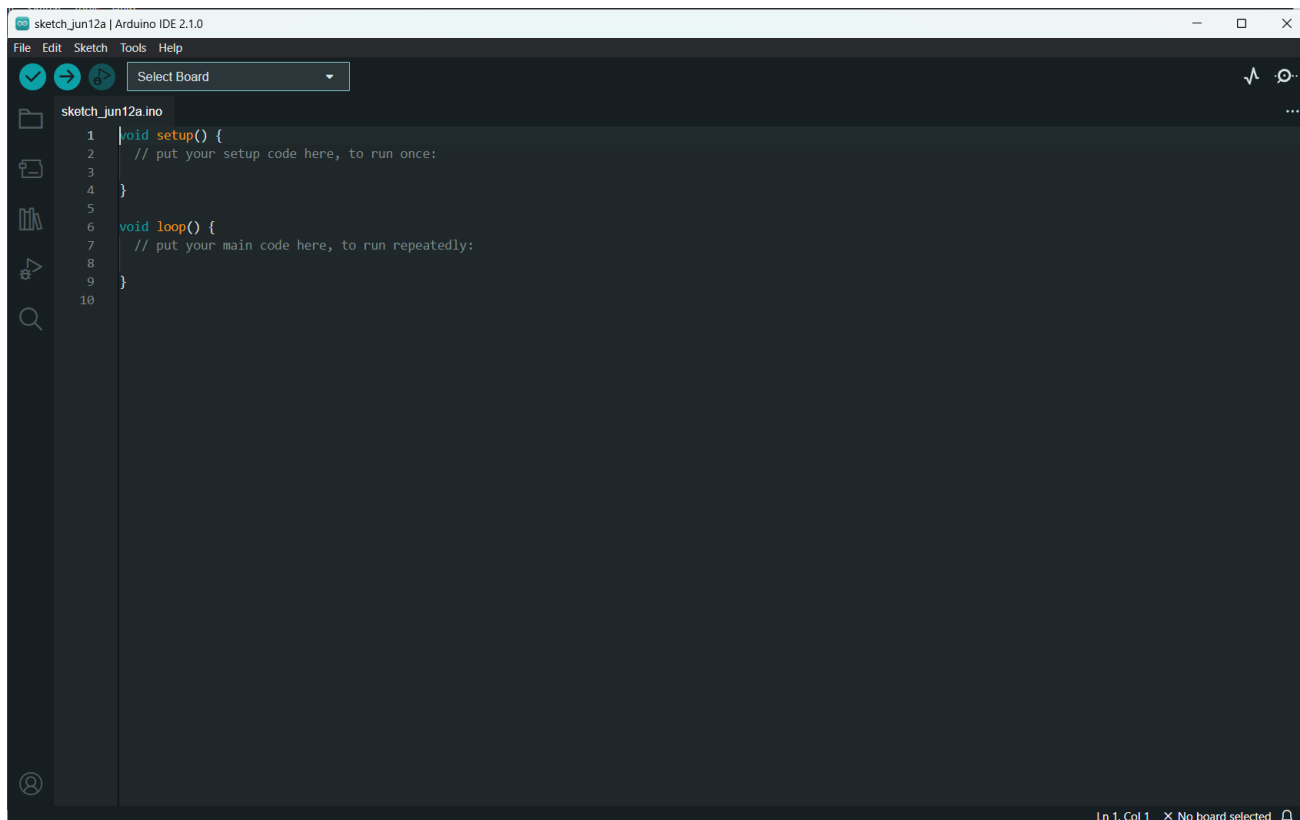


Рисунок 3.1 – Зовнішній вигляд середовища розробки Arduino IDE

Також було прийнято рішення реалізовувати програмну складову системи за допомогою діалекту мови програмування C++ під назвою Arduino Language. Ця мова проста для вивчення та не потребує додаткових ресурсів. Так як основа взята з мови програмування C++, Arduino Language включає в себе всі ті самі елементи, а саме:

- функції(невід’ємними функціями є setup() та loop());
- змінні;
- оператори

					КС КРБ 123.038.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

- контроль потоки(цикли);
- бібліотеки.

Важливо зазначити, що програмування для Arduino вимагає розуміння як програмування, так і основ електроніки, оскільки потрібно взаємодіяти з різними видами апаратного забезпечення через програмний код.

Основними бібліотеками для програмної складової системи є FastLED та FHT[12-13].

FastLED – це високопродуктивна бібліотека для Arduino, призначена для управління світлодіодними лентами і матрицями. FastLED має багато корисних функцій, включаючи:

- підтримку великої кількості видів світлодіодних стрічок (WS2812B, WS2811, APA102, і багато інших);
- високопродуктивне, оптимізоване для швидкості управління світлодіодами;
- гнучке управління кольорами, включаючи підтримку RGB та HSV кольорових просторів;
- можливість створення складних анімацій та ефектів.

FHT – це бібліотека для Arduino, яка дозволяє виконувати швидке перетворення Гартлі на вхідних даних. Це аналог FFT, але відрізняється тим, що воно виробляє реальні, а не комплексні виходи. Рівень шуму на FHT трохи кращий порівняно з FFT і становить близько -78 дБ. Детальніше про принцип роботи цього алгоритму наведено в описі лістингів коду програми.

FHT використовується для аналізу частотних характеристик сигналів. Це особливо корисно для аудіо аналізу, включаючи спектральний аналіз, обробку сигналів та визначення тону. Вона також може використовуватися для створення візуалізацій музики, коли використовується разом з бібліотекою FastLED.

## 3.2 Моделювання проєкту та реалізація програмного забезпечення

Після обґрунтування апаратного забезпечення та визначення алгоритму роботи наступним кроком є проєктування самої системи.

Так як наша система містить багато різносторонніх складових, доцільніше буде розділити програмне забезпечення на декілька підпрограм. Перш за все в головному файлі оголошуємо глобальні змінні, які будуть використовуватись у всій програмі. Перелік використовуваних глобальних змінних зображено на рис. 3.2.

```
1 #define BT_MODE 1 // using Bluetooth (0 no, 1 yes)
2 #define D_EFFECT_SPEED 100
3
4 #define AUTOPLAY_PERIOD 10
5 #define IDLE_TIME 10
6
7 #define LED_PIN 2
8 #define PHOTO_PIN A2
9 #define POT_PIN 7
10 #define BRIGHTNESS 60
11
12 #define WIDTH 16
13 #define HEIGHT 16
14
15 #define MATRIX_TYPE 0 // matrix type: 0 - zigzag, 1 - progressive
16 #define CONNECTION_ANGLE 0
17 #define STRIP_DIRECTION 0
18
19 #define GLOBAL_COLOR_1 CRGB::Green
20 #define GLOBAL_COLOR_2 CRGB::Orange
21
22 #define NUM_LEDS WIDTH * HEIGHT
23 #define FHT_N 256
24 #define LOG_OUT 1
25
26 #define COLOR1 CRGB::Green
27 #define COLOR2 CRGB::Yellow
28 #define COLOR3 CRGB::Orange
29 #define COLOR4 CRGB::Red
30
31 #define INPUT_GAIN 1.5
32 #define LOW_PASS 30
33 #define MAX_COEF 1.1
34 #define NORMALIZE 0
35
36 #define SMOOTH 0.3
37 #define DELAY 4
38
39 #define DEF_GAIN 50
40 #define MANUAL_GAIN 0
41 #define AUTO_GAIN 1
42
43 #define MAX_DOTS 1
44 #define MAX_COLOR CRGB::Red
45 #define FALL_DELAY 50
46 #define FALL_PAUSE 700
```

Рисунок 3.2 – Лістинг використовуваних глобальних змінних

Розпочнемо з частини віддаленого керування. Перш за все потрібно визначити якою саме інформацією зможе керувати користувач. Основними параметрами, які найчастіше може міняти користувач, є режими ілюмінацій, яскравість та швидкість матриці. Перечислимо назви доступних режимів роботи:

- снігопад;
- матриця;
- зорепад;
- бігаюча кулька;
- бігаючі кульки;
- хвилі;
- веселка;
- вогонь.

Переглянути лістинг коду роботи цих режимів доступно у додатку Г. Для прикладу лістинг коду одного з режимів представлено на рис.3.3.

```
37 void snowRoutine() {
38     if (effectTimer.isReady()) {
39         for (byte x = 0; x < WIDTH; x++) {
40             for (byte y = 0; y < HEIGHT - 1; y++) {
41                 drawPixelXY(x, y, getPixColorXY(x, y + 1));
42             }
43         }
44
45         for (byte x = 0; x < WIDTH; x++) {
46             if (getPixColorXY(x, HEIGHT - 2) == 0 && (random(0, 10) == 0))
47                 drawPixelXY(x, HEIGHT - 1, 0xE0FFFF - 0x101010 * random(0, 4));
48             else
49                 drawPixelXY(x, HEIGHT - 1, 0x000000);
50         }
51     }
52 }
```

Рисунок 3.3 – Лістинг коду роботи режиму ілюмінації під назвою “Снігопад”

Наступним кроком у проектуванні є під'єднання Bluetooth-модуля HC-05. Важливою особливістю, на яку потрібно звернути увагу це те, що сама схема може живитись з вхідною напругою від +3,6 до 6 V, але на вивід RX, який відповідає за приймання даних, потрібно подавати живлення лише з +3.3 V. В іншому випадку модуль не зможе розпізнати дані, які надходять на нього, і може вийти з ладу. Тому при під'єднанні потрібно використати дільник напруги за допомогою відповідних резисторів.

Існує ще один важливий аспект при підключенні до мікроконтролера. На відміну від менших популярних плат, наприклад, таких як Uno чи Nano, у Arduino Mega наявно кілька портів для послідовної комунікації: основні (RX0||TX0) та додаткові (RX1||TX1, RX2||TX2, RX3||TX3). Основні порти використовуються для комунікації з комп'ютером, тому не рекомендується під'єднувати сторонні пристрої саме до них. Також окрім наведених пінів в мікроконтролері можна використовувати і декілька інших портів, як RX, а саме: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69) [7]. Через це виводи Bluetooth-модуля потрібно підключати в такому порядку:

- RX від HC-05 через дільник напруги підключаємо до цифрового піну Arduino 10;
- TX від HC-05 під'єднуємо до RX1 Arduino.

Таким чином система зможе безпечно отримувати та передавати дані по Bluetooth без переривання зв'язку між мікроконтролером та ПК.

Наступною частиною програмної складової є зв'язок між смартфоном та Bluetooth-модулем. Розглянемо кроки налагодження комунікації з HC-05:

Спочатку пристрій користувача повинен під'єднатися до модуля по Bluetooth, як і до будь-якого звичайного Bluetooth-пристрою. Для цього заходимо в налаштування, знаходимо модуль у списку доступних пристроїв, як показано на рис. 3.4.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39



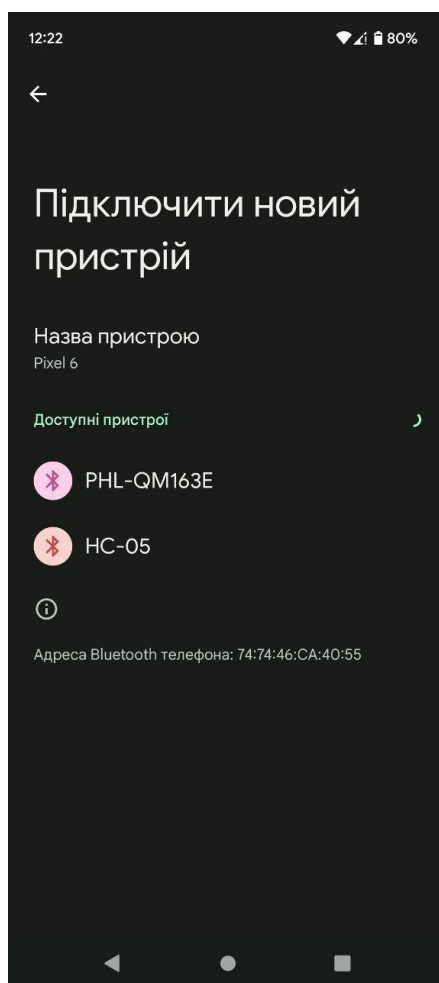


Рисунок 3.4 – Вигляд НС-05 у списку шуканих Bluetooth-пристроїв

Далі для під'єднання потрібно ввести пароль. За замовчуванням він дорівнює 1234 або 0000. Після з'єднання НС-05 переходить у збережені пристрої та очікує подальших дій.

Для того, щоб комунікувати з модулем через смартфон, використаємо додаток під назвою Bluetooth Serial Monitor. Це спеціально розроблений монітор саме для Arduino, щоб ним могли користуватись так само, як з Serial Monitor в Arduino IDE. Цей додаток знаходиться у відкритому доступі для користування в Google Play Market. Як виглядає інтерфейс для користувача зображено на рис. 3.5.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

## User Interface

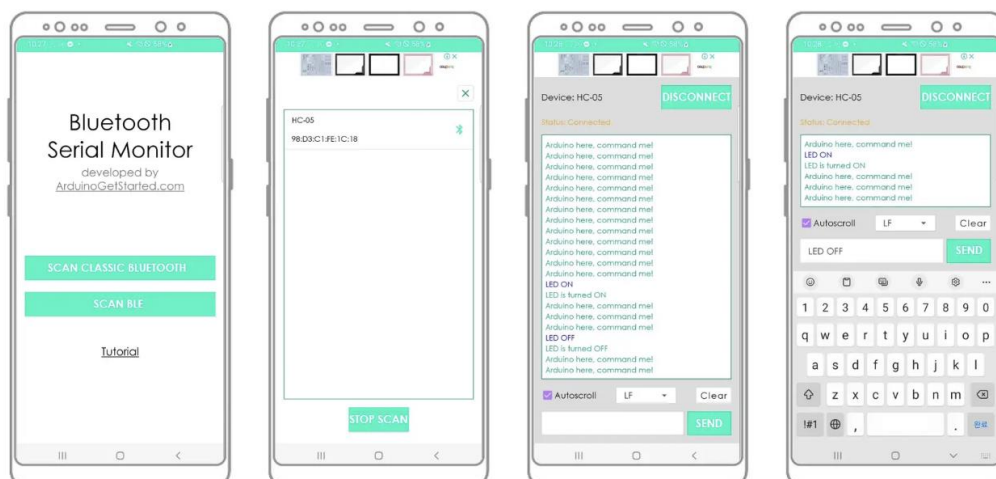


Рисунок 3.5 – Інтерфейс користувача в додатку Bluetooth Serial Monitor

Після завантаження в Android-додатку, який використовує Bluetooth, потрібно повторно встановити з'єднання з Bluetooth-модулем. Це здійснюється через системний API Android для Bluetooth, для того, щоб передавати необхідні дані. Додаток має отримати об'єкт BluetoothAdapter, який представляє Bluetooth-адаптер пристрою. За допомогою цього об'єкта додаток може виконати пошук доступних Bluetooth-пристроїв і встановити з'єднання з вибраним пристроєм. Додаток або обирає самостійно з уже збережених пристроїв, або потрібно самому ще раз вказати який конкретно пристрій під'єднується, у випадку якщо в переліку є кілька пристроїв.

Після встановлення з'єднання, Android-додаток і Bluetooth-модуль можуть обмінюватися даними. Це відбувається через BluetoothSocket, який служить каналом для передачі даних між двома пристроями. Додаток може використовувати потоки введення/виведення (InputStream/OutputStream), які надає BluetoothSocket, для надсилання і отримання даних.

Дані, які передаються між Android-додатком і Bluetooth-модулем, зазвичай відформатовані у вигляді байтових масивів. Формат цих даних залежить від конкретного застосування.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Після завершення обміну даними, додаток має коректно закрити з'єднання з Bluetooth-модулем. Це зазвичай включає закриття BluetoothSocket і звільнення всіх ресурсів, пов'язаних з Bluetooth.

Таким чином для спілкування зі системою користувачу лише потрібно правильно надсилати команди, попередньо визначені у коді програми. Для цього команда повинна відповідати певній структурі, а саме:

- початковий символ “\$”;
- значення якої саме команди хоче викликати користувач;
- додатковий параметр(номер режиму чи значення яскравості);
- кінцевий символ “;”.

Якщо користувач не буде дотримуватись цих правил, Bluetooth-модуль просто ігноруватиме та стиратиме введені команди, адже вони не збігаються з визначеною структурою. Як саме працює комунікація з HC-05 наведено в лістингах у додатку Г. Перелік значень та команд, які їм відповідають, продемонстровано нижче у лістингу на рис. 3.6.

```
59 void parsing() {
60     if (recievedFlag) {
61         recievedFlag = false;
62         BTcontrol = true;
63         idleTimer.reset();
64
65         switch (intData[0]) {
66             case 1:
67                 globalBrightness = intData[1];
68                 breathBrightness = globalBrightness;
69                 FastLED.setBrightness(globalBrightness);
70                 FastLED.show();
71                 break;
72             case 2:
73                 if (intData[1] == 0) {
74                     effect = intData[2];
75                     gameFlag = false;
76                     loadingFlag = true;
77                     breathBrightness = globalBrightness;
78                     FastLED.setBrightness(globalBrightness);
79                     globalSpeed = intData[3];
80                     gameTimer.setInterval(globalSpeed * 4);
81                 }
82                 else if (intData[1] == 1)
83                     effectsFlag = !effectsFlag;
84                 break;
85             case 3: globalSpeed = intData[1];
86                 if (effectsFlag)
87                     effectTimer.setInterval(globalSpeed);
88                 break;
89         }
90         lastMode = intData[0];
91     }
```

Рисунок 3.6 – Перелік значень та команд, які їм відповідають

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Далі при під'єднанні на виводах для живлення світлогенеруючої матриці важливо встановити великоємнісний конденсатор задля стабільної роботи. Це забезпечить уникнення мигання світлодіодів. За такою ж логікою потрібно підключити вивід для вихідного сигналу з мікрофонного модуля. Це допоможе відфільтрувати непотрібні шуми з вхідного звуку.

Остання частина програмного забезпечення відповідальна за робота аналізатора спектру. Як описувалось раніше у розділі 1, принцип роботи побудований за алгоритмом FFT, але дещо модифікований. Було прийнято рішення використати FHT, який є ідентичним до попереднього, але працює набагато швидше, оскільки вимагає лише реальних арифметичних обчислень у порівнянні зі складними арифметичними обчисленнями, які вимагає FFT. Реалізація цієї частини також представлена в лістингу в додатку Г.

Підключивши решту необхідних компонентів, отримуємо схему з'єднань готової до програмування системи, яка зображена на рис. 3.7.

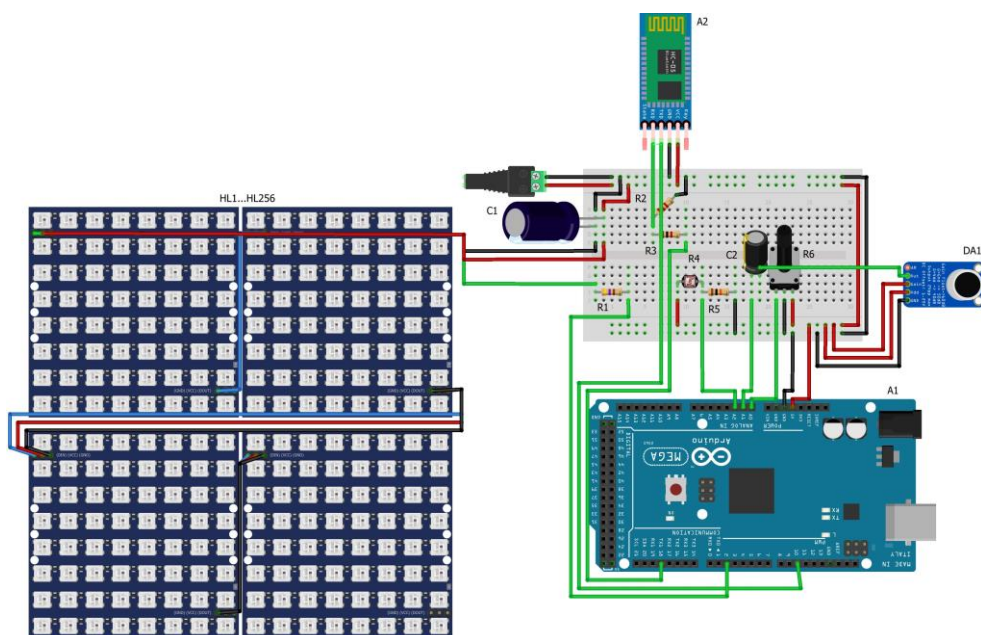


Рисунок 3.7 – Схема з'єднань комп'ютеризованої системи

Після проектування системи перейдемо до кінцевого завдання, а саме тестування.

### 3.3 Тестування спроектованої комп'ютеризованої системи

Завершальним етапом проекту є тестування готового продукту, щоб перевірити чи відповідає проект усім попередньо визначеним вимогам.

Для початку визначимо, що має включати тестування частини, яка відповідає за аналізатор спектру.

Перевірка точності аналізатора спектрів. Включає порівнювання його вимірювання зі стандартними відомими значеннями. Для цього можна скористатись онлайн-ресурсами для відтворення сигналів будь-якої частоти. Для прикладу представлено тонгенератор SPLAudio на рис. 3.8.

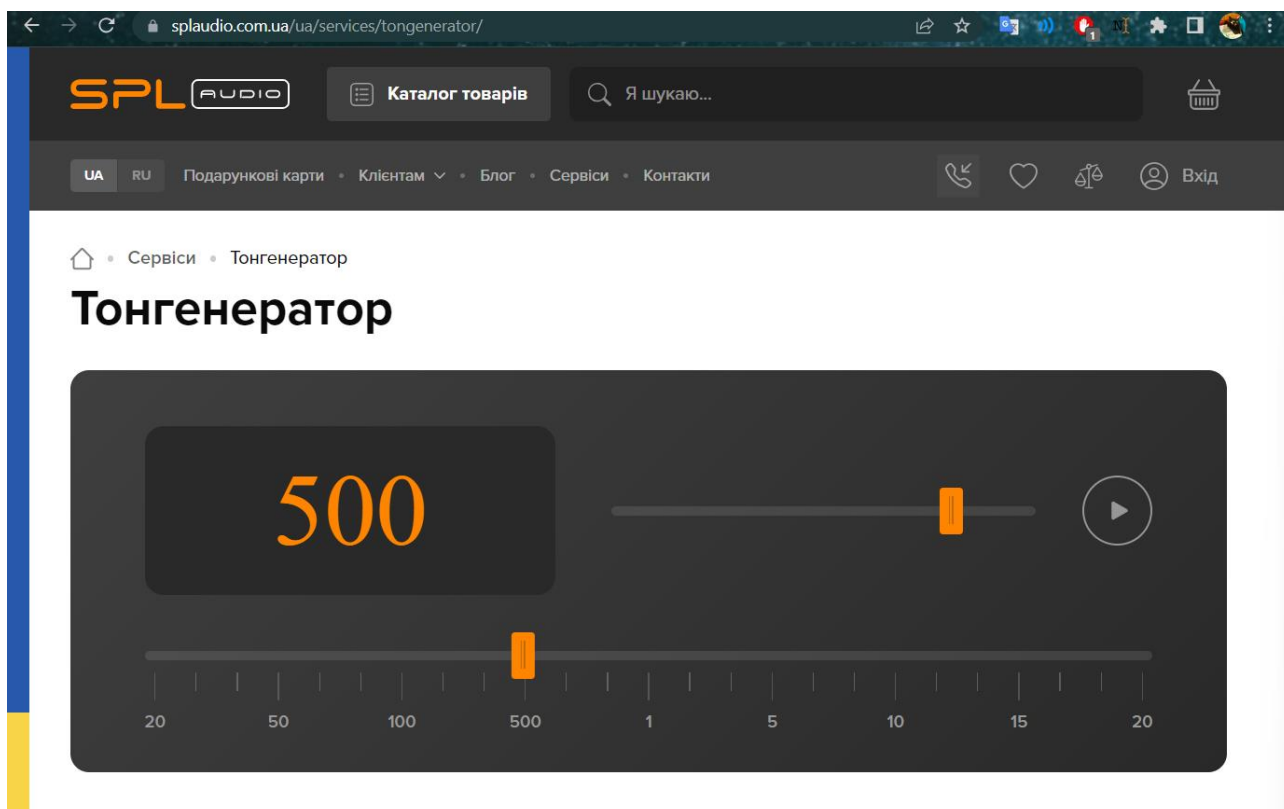


Рисунок 3.8 – Онлайн-тонгенератор SPLAudio

Перевірка діапазону. Необхідно перевірити, чи здатний аналізатор спектрів працювати в усьому заданому діапазоні частот. Завдяки цьому користувач буде знати нижню та верхню межі.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Перевірка роздільної здатності. Допомагає визначити, наскільки добре він може розрізняти два близькі спектральні компоненти.

Перевірка динамічного діапазону. Вимірює здатність аналізатора спектрів виявляти слабкі сигнали в присутності сильних.

Перевірка стабільності. Важливо перевірити стабільність аналізатора спектрів за різних умов роботи. Це містить в собі:

- рівень шуму довкілля;
- відстань джерела звуку;
- гучність джерела звуку;
- стабільність джерела звуку.

Після усіх перевірок важливо знову виконати калібрування аналізатора спектрів за допомогою потенціометра, щоб гарантувати точність його роботи в майбутньому.

Наступною частиною тестування є перевірки роботи Bluetooth-модуля. Важливо відмітити такі аспекти функціонування, як:

- працездатність на великих відстанях;
- поведінка роботи при некоректному вводі команд;
- поведінка роботи при неправильному з'єднанні;
- стабільність з'єднання;
- швидкість обробки команд;
- стійкість до перешкод.

Кінцевим кроком тестування залишається перевірити функціональність фоторезистора, а саме:

- чутливість сприймання освітлення;
- працездатність при швидкій зміні освітлення.

					КС КРБ 123.038.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Вимоги пожежної безпеки при гасінні електротехніки

Пожежна безпека – це критично важливий аспект при вирішенні будь-якого питання, пов'язаного з електротехнікою, особливо при гасінні пожеж. Пожежі в електротехніці часто виникають через коротке замикання або перевантаження електричних кабелів. Працівники носять відповідний захисний одяг, включаючи захисні окуляри та рукавиці, якщо це можливо.

Перш ніж починати гасіння пожежі, якщо це безпечно, потрібно відключити електроенергію. Це зменшує ризик електричного удару та допомагає зупинити поширення вогню.

Через наявність високої напруги, матеріали, що використовуються для гасіння, можуть стати провідниками електрики і спричинити повторну пожежу. Крім того, існує ймовірність загоряння ізоляції кабелів без доступу до кисню, що може спричинити повторне займання.

Типи вогнегасників:

- хімічно-пінні;
- повітряно-пінні;
- вуглекислотні;
- аерозольні;
- порошкові.

Не всі вогнегасники придатні для гасіння пожеж в електротехніці. Водяні вогнегасники не рекомендується використовувати для гасіння електротехніки через ризик електричного удару. Засоби пінного і водяного пожежогасіння використовуються тільки для знеструмлених електроустановок. Вогнегасники

					КС КРБ 123.038.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Городецька Я.Р.			Безпека життєдіяльності, основи охорони праці	Літ.	Арк.	Акрушів
Перевір.		Лецишин Ю.З.					46	5
Консульт.		Пилипець М.І.				ТНТУ, каф. КС, гр. СІ-41		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

класу С або D, які використовують хімічні речовини для гасіння вогню, є найкращим вибором. Приклад використання вогнегасника зображено на рис. 4.1.



Рисунок 4.1 – Правила користування вогнегасником

Після того, як пожежа була ліквідована, необхідно вжити заходів, щоб запобігти повторному загоранню. Це включає охолодження області вогнегасником, відключення всієї електротехніки в області та виклик професіоналів для огляду і ремонту пошкоджень.

Усі вимоги та заходи боротьби з пожежами сформовано згідно з наказом № 1417 від 30.12.2014 р [14].

#### 4.2 Санітарно-гігієнічні вимоги до умов праці

Санітарно-гігієнічні вимоги до умов праці – це набір правил і рекомендацій, які регулюють використання робочих місць та обов'язкові умови

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47



праці, щоб забезпечити безпеку і здоров'я працівників. Ці правила включають, але не обмежуються, вимогами до вентиляції, освітлення, ергономіки, чистоти, звуку та вібрації, температури, вологості, а також захисту від шкідливих речовин і виробничого шуму.

Згідно з вимогами існують такі ключові аспекти, як:

– Робоче місце достатньо провітрюється, щоб запобігти накопиченню шкідливих газів, пилу або диму. Це забезпечується природною або штучною вентиляцією.

– Освітлення на робочому місці мусить бути достатньо яскравим, щоб працівники могли безпечно та ефективно виконувати свої обов'язки. Залежно від характеру роботи, доповнюється додатковим освітленням згідно з ДБН В.2.5-28 : 2018. „Природне і штучне освітлення” [15].

– Умови праці налаштовують так, щоб зменшити ризик виникнення професійних захворювань, зокрема проблем з м'язами і кістками. Це включає в себе правильні налаштування робочого місця, такі як висота столу або крісла, а також перерви для відпочинку.

– Якщо на робочому місці використовуються шкідливі речовини, обов'язково вжити відповідні заходи безпеки. До цього входять використання особистого захисного обладнання, такого як респіратори або рукавиці, а також забезпечення належного зберігання та утилізації цих речовин.

– Умови температури і вологості на робочому місці створюють комфортними і безпечними для працівників. Якщо робота виконується в екстремальних умовах, вживають відповідні заходи для захисту працівників, такі як забезпечення води для пиття при високих температурах або теплої одягу при низьких.

– Шум і вібрації на робочому місці контролюються, щоб запобігти пошкодженню слуху працівників або виникненню інших проблем зі здоров'ям. До цього відносяться використання ОЗО, такого як вушні затички або антивібраційні рукавиці, а також регулярні перерви від впливу шуму або вібрації.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

У разі підозри на професійне захворювання у працівника, відповідальні керівники зобов'язані застосувати наступні заходи:

- негайно направити працівника на медичне обстеження. Здійснюється воно у спеціалізованій установі з професійних захворювань або в звичайному медичному закладі, але з консультацією спеціаліста з профзахворювань.
- Працівників повідомляють про потенційні ризики, пов'язані з їхнім здоров'ям на роботі, та про те, як можна запобігти захворюванням.
- Провести оцінку робочого місця та процесів. Це включає аналіз матеріалів, з якими працюють, все обладнання, навантаження та інші фактори, які можуть спричинити захворювання.
- Застосувати необхідні заходи для усунення або зменшення впливу шкідливих факторів, якщо такі виявлено. Це містить зміну робочих процесів, заміну матеріалів або впровадження додаткових засобів захисту;
- Провести додаткові тренінги з охорони праці, щоб забезпечити, що всі працівники знають про заходи безпеки та вміють їх використовувати.
- Керівники звертаються до відповідного органу охорони праці з інформацією про випадок.

#### 4.3 Вплив кольору на покращення умов праці та підвищення продуктивності виробництва

Колір впливає на людську психологію та поведінку, а тому має значний вплив на продуктивність праці та якість робочого середовища. Правильний вибір кольорів зменшує візуальне перевантаження та сприяє комфорту. Від нейтральних кольорів очі менше втомлюються, особливо при довгій роботі з комп'ютером.

Колір є важливим аспектом ергономіки, оскільки він впливає на видимість, орієнтацію та безпеку на робочому місці. Яскраві кольори підсилюють видимість особливо важливих об'єктів або зон.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Створення здорового та комфортного робочого середовища є важливим для добробуту працівників. Кольори, такі як синій та зелений, відомі своїм заспокійливим впливом, і допомагають знизити рівень стресу. Використання теплих кольорів, таких як оранжевий, червоний та жовтий, викликає відчуття тепла, комфорту та щастя. Використання цих кольорів піднімає настрій працівників та сприяє позитивному робочому середовищу. Крім того, спокійні кольори, такі як лавандовий, м'який синій або блідо-зелений, сигналізують працівникам, що ці простори призначені для відпочинку та перезарядки під час перерви.

Кольори відіграють важливу роль у стимулюванні продуктивності працівників. Зелений та синій, покращують концентрацію, що робить їх ідеальними для місць, де виконуються завдання, таких як індивідуальні робочі місця або конференц-зали. Яскраві та насичені кольори, такі як оранжевий, червоний та жовтий, можуть сприяти креативному мисленню та формуванню інноваційних ідей. Ці кольори включають в простір для мозкових штурмів або спільні простори для стимулювання креативного вирішення проблем.

Вплив основних кольорів на організм людини наведено на рис. 4.2.

Колір	Артеріальний тиск	Пульс	Частота дихань	Емоційний стан	Час реакції	Працездатність
Червоний	Підвищується	Збільшується	Збільшується	Збуджує, стимулює	Зменшується	Спочатку підвищується ненадовго, а потім зменшується наполовину
Оранжевий	Мало підвищується	Трохи збільшується	Трохи збільшується	Збуджує, бадьорить	Трохи зменшується	Так само, але менш виражено
Жовтий	Не змінюється	Не змінюється	Не змінюється	Урівноважує	Не змінюється	Суттєво не змінюється
Зелений	Не змінюється	Не змінюється	Не змінюється	Урівноважує	Не змінюється	Трохи підвищується
Блакитний	Дещо знижується	Трохи сповільнюється	Трохи сповільнюється	Заспокоює	Суттєво не змінюється	Трохи підвищується
Синій	Знижується	Сповільнюється	Сповільнюється	Заспокоює, трохи загальмовує	Трохи сповільнюється	Підвищується на 3–9 %
Фіолетовий	Знижується	Сповільнюється	Сповільнюється	Пригнічує	Виразно сповільнюється	Стабільно знижується

Рисунок 4.2 – Вплив основних кольорів на організм людини

Таким чином важливо вибрати правильний колір для оформлення робочого місця залежно від потреб та цілей.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

## ВИСНОВКИ

Під час проєктування кваліфікаційної роботи виконано зазначеної мети щодо реалізації комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega за допомогою методів та засобів бездротового зв'язку і спектрального аналізу аналогових звукових сигналів. Також були виконані перелічені завдання до конструювання системи, а саме:

- проаналізовано методи і засоби перетворення аналогових звукових сигналів у цифровий формат та технології і протоколи бездротового з'єднання;
- спроектовано концептуальну схему структури комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею;
- обґрунтовано вибір апаратного забезпечення та програмних засобів для роботи із звуковими сигналами, бездротовим з'єднанням та навколишнім освітленням;
- реалізовано програмне забезпечення відповідно до визначених потреб;
- аналіз результатів проєктування та тестування готової спроектованої системи.

Апаратне забезпечення проєкту містить Bluetooth-модуль, мікрофонний модуль з автоматичним регулюванням коефіцієнта підсилення, світлогенеруючу матрицю та мікроконтролер Arduino Mega 2560, який виконує налаштування та розподілення команд до основних компонентів. Програмне забезпечення розроблено як і для функціонування системи в цілому, так і для підтримки зв'язку між користувачем, за допомогою мови програмування C/C++ та бібліотек, які знаходяться у відкритому доступі.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Arduino Mega 2560 R3. URL: <https://docs.arduino.cc/hardware/mega-2560> (дата звернення 15.03.2023р.)
2. MAX9814 datasheet. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/max9814.pdf> (дата звернення 16.03.2023р.)
3. Adafruit AGC Electret Microphone Amplifier - MAX9814. URL: <https://learn.adafruit.com/adafruit-agc-electret-microphone-amplifier-max9814/overview> (дата звернення 04.04.2023р.)
4. Онлайн-тонгенератор SPL Audio. URL: <https://splaudio.com.ua/ua/services/tongenerator/> (дата звернення 05.04.2023р.)
5. HC-05 datasheet. URL: [https://components101.com/sites/default/files/component\\_datasheet/HC-05%20Datasheet.pdf](https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf) (дата звернення 17.03.2023р.)
6. HC-03/05 Embedded Bluetooth Serial Communication Module AT command set. URL: [http://elec Freaks.com/store/download/datasheet/Bluetooth/HC-0305%20serail %20module%20AT%20commamnd%20set%20201104%20revised.pdf](http://elec Freaks.com/store/download/datasheet/Bluetooth/HC-0305%20serail%20module%20AT%20commamnd%20set%20201104%20revised.pdf) (дата звернення 06.04.2023р.)
7. Arduino Serial Communication. URL: <https://reference.arduino.cc/reference/en/language/functions/communication/serial/> (дата звернення 07.04.2023р.)
8. Лецишин Ю. З. Розробка системи зв'язку як інтегрованого елементу роботизованих систем. Проблеми створення, розвитку та застосування високотехнологічних систем спеціального призначення з урахуванням досвіду антитерористичної операції. Збірник тез доповідей XXI Всеукраїнської науково-практичної конференції. Житомир, 2016. С. 102.
9. Лецишин Ю. З. Створення вбудованих систем на базі структурно-параметричних моделей цифрових каналів зв'язку : Лецишин Ю.З., Назаревич Т.О., Міська І.В. VIII Науково-технічна конференція «Інформаційні моделі, системи та технології». Тернопіль, 2020. С. 127.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

10. Марків В.А., Осухівська Г.М., Лещишин Ю.З., Луцків А.М. Комп'ютерна система аутентифікації осіб : Матеріали ХХ наукової конференції ТНТУ ім. І. Пулюя. Тернопіль, 2017. С. 90–91.

11. Шевчук Є.Ю. Методи та засоби оброблення голосових сигналів для корекції вимови людини. Тернопіль: 2017. 103 с.

12. FastLED library source. URL: <https://github.com/FastLED/FastLED> (дата звернення 21.04.2023р.)

13. Arduino FHT library. URL: <http://wiki.openmusiclabs.com/wiki/ArduinoFHT> (дата звернення 22.04.2023р.)

14. Правила пожежної безпеки в Україні. URL: <https://zakon.rada.gov.ua/laws/show/z0252-15#Text> (дата звернення 27.05.2023р.)

15. ДБН В.2.5-28 : 2018. „Природне і штучне освітлення”. URL: [https://ledeffect.com.ua/images/\\_\\_branding/dbn2018.pdf](https://ledeffect.com.ua/images/__branding/dbn2018.pdf) (дата звернення 28.05.2023р.)

16. Осухівська Г.М., Тиш Є.В., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2022. 28 с.

					КС КРБ 123.038.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Додаток А.  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя  
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

"\_\_" \_\_\_\_\_ 2023 р.

КОМП'ЮТЕРИЗОВАНА СИСТЕМА ВІДДАЛЕНОГО КЕРУВАННЯ  
СВІТЛОГЕНЕРУЮЧОЮ МАТРИЦЕЮ НА ОСНОВІ ARDUINO MEGA

**ТЕХНІЧНЕ ЗАВДАННЯ**

на 10 листках

**Вид робіт:**

Кваліфікаційна робота

**На здобуття освітнього ступеня «Бакалавр»**

**Спеціальність 123 «Комп'ютерна інженерія»**

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-41

\_\_\_\_\_ к.т.н, доц. Лещишин Ю.З.

\_\_\_\_\_ Городецька Я.Р.

«\_\_\_\_» \_\_\_\_\_ 2023 р.

«\_\_\_\_» \_\_\_\_\_ 2023 р.

**Тернопіль 2023**



## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.038.00.00.

### 1.2 Виконавець

Студентка групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Городецька Яніна Романівна.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№ 4.7-238 від 28.02.2023 р.).

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 28.02.2023 р.

Плановий термін завершення кваліфікаційної роботи – 09.06.2023 р.

### 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Основним призначенням комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею є розпізнавання звуку, що подається, та команд з віддаленого пристрою для виведення бажаної інформації.

Проектована система повинна забезпечувати коректне опрацювання сигналів та перетворення їх у відповідну ілюмінацію. Попри це вона зобов'язана пристосовуватись до навколишнього середовища, а саме фільтрувати зайві шуми для чистоти відображення і корегувати яскравість відповідно до пори доби.

Перевагами застосування такої системи віддаленого керування світлогенеруючою матрицею є довший термін працездатності та більший спектр використання завдяки своїй адаптивності.

Більшість подібних систем є широкопоширеними, тому різні варіації спроєктованих рішень можна зустріти у всіх галузях. Перспективи роботи над такими проєктами є забезпечення можливості постійного розширення функціоналу та автоматизації вже наявних концепцій для використання у IoT-технологіях.

У даній роботі необхідно сконструювати таку розробку, яка б підтримувала велику площу світлогенеруючої матриці, при цьому функціонувала з найменшими помилками та затратами в часі, а також була в загальному доступі для користувачів будь-якого рівня у сфері програмування електроніки.

Проект повинен використовувати таке апаратне забезпечення, яке встановить надійний та безперебійний зв'язок з користувачем, здатне сприймати аналогові сигнали і перетворювати їх у цифрові завдяки розбиттю на спектр. Окрім цього, повинне правильно керувати та розподіляти вхідні запити без сторонніх завад.

## 2.2 Мета створення системи

Мета проєктування та реалізації комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею полягає у створенні розробки ефективних рішень, як на рівні апаратного, так і програмного забезпечення, які б могли задовільнити потреби довільних користувачів, а саме миттєво налаштувати режими роботи і мати постійний доступ до системи.

Для того, щоб досягти заданої мети потрібно розв'язати наступні задачі, такі як:

- аналіз природи звукових сигналів;
- аналіз методів і засобів перетворення аналогових звукових сигналів у цифровий формат;
- аналіз технологій та протоколів бездротового з'єднання;
- визначення існуючих рішень та виокремлення саме тих, які відповідатимуть встановленим вимогам для проєктування системи;
- побудова концептуальної моделі комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею;
- обґрунтування та дослідження необхідних технічних характеристик апаратного забезпечення для проєктування системи;
- аналіз програмних засобів для роботи із звуковими сигналами, бездротовим з'єднанням та навколишнім освітленням;
- реалізація програмного забезпечення відповідно до визначених потреб;
- аналіз результатів проєктування та тестування готової спроектованої системи.

## 2.3 Характеристика об'єкту

### 2.3.1 Основні задачі та функції об'єкту

Найбільш важливі функції та завдання, які повинна забезпечувати комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega, полягають у:

- можливості отримання аналогового звукового сигналу;
- якісно відфільтрувати сторонні шуми;
- перетворення отриманого звуку у цифровий формат;
- підтримувати постійний зв'язок між системою і пристроєм користувача по Bluetooth;
- адаптуватись під навколишнє освітлення.

Програмне обладнання системи відповідальне за виконання різносторонніх задач, тому складається з декількох частин. Одна з них призначена для налаштування апаратного забезпечення, тоді як інша виконує управління роботи робочого процесу.

Враховуючи, що для нормального відображення будь-якого мінімального зображення або спектру, або анімації потрібно достатньо велику кількість керуючих світлодіодів, для проєктування доцільно обрати мікроконтролер зі здатністю контролювати такі габарити. Також він відповідальний за звукові сигнали, тому має містити в собі аналогово-цифровий перетворювач. В той час світлогенеруюча матриця має легко програмуватись та здатна відтворювати потрібні ілюмінації.

Окрім цього, планується, що реалізація такого проєкту ще повинна містити такі компоненти апаратної складової системи, як:

- мікрофон з підсилювачем;
- фоторезистор для фіксування денного світла;
- Bluetooth-модуль для віддаленого управління;
- ПК або інший мобільний пристрій для з'єднання з Bluetooth-модулем.

### 3 Вимоги до системи

#### 3.1 Вимоги до системи в цілому

Загалом, до проєкту на тему «Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega» висувають перелік вимог, а саме:

- до чутливості та якості розпізнавання звукових сигналів;
- до стабільного бездротового з'єднання між користувачем та Bluetooth-модулем;
- до швидкодії та живлення процесу управління компонентами;
- до ефективності функціонування мікроконтролера;
- до продуктивності і зручності налаштування системи.

Вимоги до вибору мікрофону зумовлені його габаритами, типом з'єднання до системи, функціональністю та калібруванням підсилювання.

Мікроконтролер мусить містити достатньо технічних характеристик, зокрема тактову частоту процесора, оперативну пам'ять та аналогово-цифровий перетворювач, для відтворення всіх зазначених функцій та завдань.

Програмне забезпечення повинне бути адаптивне до апаратної складової та зручне для використання і налаштування потрібних параметрів керування світлогенеруючою матрицею.

##### 3.1.1 Вимоги до структури та функціонування системи

Структура комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega повинна містити в собі як апаратне забезпечення, так і програмне.

У складі апаратної частини системи рекомендується використати такі компоненти, як:

- плата програмування Arduino Mega 2560;
- мікрофонний модуль з підсилювачем;

- фоторезистор;
- Bluetooth-модуль.

До складу програмного забезпечення рекомендується вибрати середовище програмування мікроконтролера Arduino. Для подальших налаштувань використовується мобільний застосунок під операційну систему Android.

Комп'ютерна система має відповідати критеріям надійності для даного класу систем, а також забезпечувати захист на різноманітних етапах взаємодії між компонентами.

### 3.2 Вимоги до способів та засобів зв'язку між компонентами системи

Основним способом зв'язку усіх компонентів у системі, як додаткових, так і основних, проводиться за допомогою провідників для забезпечення надійності передачі.

З'єднанням між смартфоном користувача та системою відбувається завдяки бездротового зв'язку за участі протоколів Bluetooth. У випадку необхідності проведення системних налаштувань з'єднання встановлюється до лінійного інтерфейсу ПК через СОМ-порт.

#### 3.2.1 Вимоги по діагностуванню системи

Діагностичні заходи, такі як загальний огляд цілісності компонентів комп'ютеризованої системи повинні проходити відповідно до зазначених графіків та інструкцій з експлуатації модулів.

Діагностика цілісності програмного забезпечення проводиться кожного разу при ввімкненні системи перед виконанням основних функцій. При виявленні збоїв у роботі система повідомляє про знайдені неточності, тому необхідно негайно усунути всі перешкоди до повноцінного функціонування.

### 3.2.2 Перспективи розвитку, модернізація системи

Перспективами розвитку даної комп'ютеризованої системи є імплементація як додатковий модуль у існуючу більшу систему для загального чи особистого користування або використання у різноманітних галузях з більшою або меншою кількістю керуючих світлодіодів матриці чи інших компонентів.

Також система підлягає модернізації у випадку недостатньої ефективності або для того, щоб доповнити функціональність, яка би більше підходила під вимоги та потреби користувача.

### 3.2.3 Вимоги до надійності системи

До необхідних складових надійності комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega належать:

- стабільність працездатності протягом визначеного часового періоду;
- стабільне функціонування мікроконтролера Arduino Mega 2560;
- середній наробіток між відмовами становить до 50000 годин;
- здатність пристосовуватися до нових звукових сигналів.

### 3.2.4 Вимоги до функцій та задач, які виконує система

Вимоги до функціонування та виконання конкретних задач, які виконує комп'ютеризована система, включають в себе:

- можливість одержання аналогового звукового сигналу;
- здатність перетворювати одержаний звук у цифровий вигляд;
- здатність фільтрувати звук від шумів;
- можливість розбивати сигнал на смуги;
- можливість зчитувати денну яскравість;
- здатність створювати бездротовий зв'язок між користувацьким пристроєм по Bluetooth;
- можливість налаштування підсилення звуку;
- здатність віддаленого налаштування світлогенеруючої матриці.

### 3.2.5 Вимоги до апаратного забезпечення

Апаратне забезпечення комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею на основі Arduino Mega обов'язково повинне містити:

- плату Arduino Mega з мікроконтролером ATmega 2560;
- мікрофонний модуль із підсилювачем;
- Bluetooth-модуль;
- світлогенеруючу матрицю;
- фоторезистор;
- ПК та/або смартфон.

Для зв'язку по Bluetooth вимагається використовувати мобільний пристрій на базі операційної системи Android 7.0 і вище. Специфікації ПК, необхідні для конфігурації комп'ютерної системи, обробки команд мають бути у відповідності з такими технічними характеристиками:

- частота процесора не менше 2 ГГц;
- дисковий простір не менше 100 Гб;
- обсяг оперативної пам'яті 8 Гб і більше.

### 3.2.6 Вимоги до програмного забезпечення

Щоб реалізувати програмне забезпечення для Arduino Mega, потрібно використовувати таке середовище, яке б могло забезпечити зв'язок між всіма компонентами структури системи. Для цього доцільним вирішенням буде скористатись Arduino IDE за допомогою мови програмування C/C++.

## 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
  - графічного матеріалу:
1. Блок-схема алгоритму роботи системи



2. Структурна схема комп'ютеризованої системи
3. Схема електрично-принципова спроектованої системи
4. Фізична реалізація готової комп'ютеризованої системи
5. Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка та аналіз технічного завдання	28.02-05.03.2023
2	Аналіз принципів функціонування систем віддаленого керування світлогенеруючою матрицею	06.03-14.03.2023
3	Обґрунтування вибору апаратного складової комп'ютерної системи	15.03-23.03.2023
4	Проектування структури комп'ютеризованої системи віддаленого керування світлогенеруючою матрицею	24.03-03.04.2023
5	Аналіз параметрів звукових сигналів у цифровому форматі та бездротового зв'язку	04.04-20.04.2023
6	Розробка алгоритмів та програмного забезпечення для віддаленого керування світлогенеруючою матрицею	21.04-19.05.2023
7	Розробка інструкцій з використання комп'ютерної системи за допомогою додатку	20.05-26.05.2023
8	Безпека життєдіяльності, основи охорони праці	27.05-30.05.2023
9	Оформлення кваліфікаційної роботи	30.05-09.06.2023
10	Попередній захист кваліфікаційної роботи	10.06-19.06.2023
11	Захист кваліфікаційної роботи	20.06-23.06.2023

## 6. Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

## Додаток Б

Перелік елементів електрично-принципової схеми

<i>Позн.</i>	<i>Найменування</i>	<i>К-сть</i>	<i>Примітка</i>
A1	Плата Arduino Mega 2560 R3	1	
A2	Bluetooth-модуль HC-05	1	
C1	Конденсатор 100мФ	1	
C2	Конденсатор 1000мФ	1	
DA1	Мікрофонний модуль із підсилювачем MAX9814	1	
HL1...HL256	Керований світлодіод WS2812B	256	
R1	Фоторезистор 16 ~ 33 кОм	1	
R2	Потенціометер 0.2Вт 100кОм 20%	1	
R3	Резистор 2Вт 10 кОм 5%	1	
R4	Резистор 2Вт 470 Ом 5%	1	
R5	Резистор 2Вт 1 кОм 5%	1	
R6	Резистор 2Вт 2.2 кОм 5%	1	

**КС КРБ 123.038.00.00 ПЕ**

<i>Зм.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>
Розробив		Городецька Я.Р.		
Перевірів		Лецишин Ю.З.		
Рецензент		Мудрик І.Я.		
Н. контр.		Тиш С.В.		
Зав. каф.		Осухівська Г.		

Комп'ютеризована система  
віддаленого керування  
світлогенеруючою матрицею на основі  
Arduino Mega  
Перелік елементів

<i>Лім</i>	<i>Аркуш</i>	<i>Аркушів</i>
н	I	I
ТНТУ, каф. КС, гр. СІ-41		

## Додаток В

### Перелік елементів схеми з'єднань

Позн.	Найменування	К-сть	Примітка
A1	Плата Arduino Mega 2560 R3	1	
A2	Bluetooth-модуль HC-05	1	
C1	Конденсатор 1000мФ	1	
C2	Конденсатор 100мФ	1	
DA1	Мікрофонний модуль із підсилювачем MAX9814	1	
HL1...HL256	Керований світлодіод WS2812B	256	
R1	Резистор 2Вт 470 Ом 5%	1	
R2	Резистор 2Вт 2.2 кОм 5%	1	
R3	Резистор 2Вт 1 кОм 5%	1	
R4	Фоторезистор 16 ~ 33 кОм	1	
R5	Резистор 2Вт 10 кОм 5%	1	
R6	Потенціометер 0.2Вт 100кОм 20%	1	

<b>КС КРБ 123.038.01.00 ПЕ</b>				
<i>Зм.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>
<i>Розробив</i>		Городецька Я.Р.		
<i>Перевірів</i>		Лецишин Ю.З.		
<i>Рецензент</i>		Мудрик І.Я.		
<i>Н. контр.</i>		Тиш Є.В.		
<i>Зав. каф.</i>		Осухівська Г.		
Комп'ютеризована система віддаленого керування світлогенеруючою матрицею на основі Arduino Mega <i>Перелік елементів</i>				
		<i>Лім</i>	<i>Аркуш</i>	<i>Аркушів</i>
		н	1	1
ТНТУ, каф. КС, гр. СІ-41				

## Додаток Г

### Лістинг коду системи

```
#define BT_MODE 1 // using Bluetooth (0 no, 1 yes)
#define D_EFFECT_SPEED 100

#define AUTOPLAY_PERIOD 10
#define IDLE_TIME 10

#define LED_PIN 2
#define PHOTO_PIN A2
#define POT_PIN 7
#define BRIGHTNESS 60

#define WIDTH 16
#define HEIGHT 16

#define MATRIX_TYPE 0 // matrix type: 0 - zigzag, 1 -
progressive
#define CONNECTION_ANGLE 0
#define STRIP_DIRECTION 0

#define FONT_TYPE 1

#define GLOBAL_COLOR_1 CRGB::Green
#define GLOBAL_COLOR_2 CRGB::Orange

#define NUM_LEDS WIDTH * HEIGHT
#define FHT_N 256
#define LOG_OUT 1

#define COLOR1 CRGB::Green
#define COLOR2 CRGB::Yellow
#define COLOR3 CRGB::Orange
#define COLOR4 CRGB::Red

#define INPUT_GAIN 1.5
#define LOW_PASS 30
#define MAX_COEF 1.1
#define NORMALIZE 0

#define SMOOTH 0.3
#define DELAY 4

#define DEF_GAIN 50
#define MANUAL_GAIN 0
#define AUTO_GAIN 1

#define MAX_DOTS 1
```

```

#define MAX_COLOR CRGB::Red
#define FALL_DELAY 50
#define FALL_PAUSE 700

byte posOffset[17] = {2, 3, 4, 6, 8, 10, 12, 14, 16, 20, 25,
30, 35, 60, 80, 100, 120};
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))

#include "FastLED.h"
#include <FHT.h>
CRGB leds[NUM_LEDS];

byte globalBrightness = BRIGHTNESS;
byte globalSpeed = 200;
uint32_t globalColor = 0x00ff00;
byte breathBrightness;
boolean loadingFlag;
byte frameNum;
boolean idleState = false;
boolean BTcontrol = false;
int gain = DEF_GAIN;
unsigned long gainTimer, fallTimer;
byte maxValue;
float k = 0.05, maxValue_f = 0.0;
int maxLevel[16];
byte posLevel_old[16];
unsigned long timeLevel[16], mainDelay;
boolean fallFlag;

void setup() {
    sbi(ADCSRA, ADPS2);
    cbi(ADCSRA, ADPS1);
    sbi(ADCSRA, ADPS0);

    analogReference(EXTERNAL);

    Serial.begin(9600);

    FastLED.addLeds<WS2812, LED_PIN, GRB>(leds,
NUM_LEDS).setCorrection( TypicalLEDStrip );
    FastLED.setBrightness(BRIGHTNESS);
    FastLED.clear();
    FastLED.show();

    randomSeed(analogRead(0) + analogRead(1));
}

void loop() {
    customRoutine();
    bluetoothRoutine();
}

```

```

int inputval = analogRead(LDR);
int outval = map(inputval, 0, 380, 255, 0);

BRIGHTNESS = constrain(outval, 0,255);

if (millis() - mainDelay > DELAY) {
  mainDelay = millis();

  analyzeAudio();

  for (int i = 0; i < 128; i++) {
    if (fht_log_out[i] < LOW_PASS)
      fht_log_out[i] = 0;

    fht_log_out[i] = (float)fht_log_out[i] * INPUT_GAIN;

    if (NORMALIZE)
      fht_log_out[i] = (float)fht_log_out[i] / ((float)1 +
(float)i / 128);
  }

  maxValue = 0;
  FastLED.clear();
  for (byte pos = 0; pos < WIDTH; pos++) {
    int posLevel = fht_log_out[posOffset[pos]];
    byte linesBetween;
    if (pos > 0 && pos < WIDTH) {
      linesBetween = posOffset[pos] - posOffset[pos - 1];
      for (byte i = 0; i < linesBetween; i++) {
        posLevel += (float) ((float)i / linesBetween) *
fht_log_out[posOffset[pos] - linesBetween + i];
      }
      linesBetween = posOffset[pos + 1] - posOffset[pos];
      for (byte i = 0; i < linesBetween; i++) {
        posLevel += (float) ((float)i / linesBetween) *
fht_log_out[posOffset[pos] + linesBetween - i];
      }
    }

    if (posLevel > maxValue)
      maxValue = posLevel;

    posLevel = posLevel * SMOOTH + posLevel_old[pos] * (1
- SMOOTH);
    posLevel_old[pos] = posLevel;

    posLevel = map(posLevel, LOW_PASS, gain, 0, HEIGHT);
    posLevel = constrain(posLevel, 0, HEIGHT);

    if (posLevel > 0) {

```



```

    for (int j = 0; j < posLevel; j++) {
        uint32_t color;
        if (j < 5) color = COLOR1;
        else if (j < 10) color = COLOR2;
        else if (j < 13) color = COLOR3;
        else if (j < 15) color = COLOR4;

        if (pos % 2 != 0)
            leds[pos * WIDTH + j] = color;
        else
            leds[pos * WIDTH + WIDTH - j - 1] = color;
    }
}

if (posLevel > 0 && posLevel > maxLevel[pos]) {
    maxLevel[pos] = posLevel;
    timeLevel[pos] = millis();
}

if (maxLevel[pos] >= 0 && MAX_DOTS) {
    if (pos % 2 != 0)
        leds[pos * WIDTH + maxLevel[pos]] = MAX_COLOR;
    else
        leds[pos * WIDTH + WIDTH - maxLevel[pos] - 1] =
MAX_COLOR;
}

    if (fallFlag) {
        if ((long)millis() - timeLevel[pos] > FALL_PAUSE) {
            if (maxLevel[pos] >= 0)
                maxLevel[pos]--;
        }
    }
}
FastLED.show();

fallFlag = 0;
if (millis() - fallTimer > FALL_DELAY) {
    fallFlag = 1;
    fallTimer = millis();
}

if (MANUAL_GAIN)
    gain = map(analogRead(POT_PIN), 0, 1023, 0, 150);

if (AUTO_GAIN) {
    if (millis() - gainTimer > 10) {
        maxValue_f = maxValue * k + maxValue_f * (1 - k);

        if (maxValue_f > LOW_PASS)
            gain = (float) MAX_COEF * maxValue_f;
    }
}

```

```

        else gain = 100;
        gainTimer = millis();
    }
}
}
}

void analyzeAudio() {
    for (int i = 0 ; i < FHT_N ; i++) {
        int sample = analogRead(AUDIO_IN);
        fht_input[i] = sample; // put real data into bins
    }
    fht_window(); // window the data for better frequency
response
    fht_reorder(); // reorder the data before doing the fht
    fht_run(); // process the data in the fht
    fht_mag_log(); // take the output of the fht
}

#if (BT_MODE == 1)
#define PARSE_AMOUNT 4
#define header '$'
#define divider ' '
#define ending ';'

byte prevY = 0;
byte prevX = 0;
boolean effectsFlag;
byte effect;
byte intData[PARSE_AMOUNT];
uint32_t prevColor;
boolean recievedFlag;
byte lastMode = 0;
boolean parseStarted;

void bluetoothRoutine() {
    parsing();

    if (!parseStarted) {
        if (effectsFlag)
            effects();
    }
}

void effects() {

    switch (effect) {
        case 0: brightnessRoutine();
            break;
        case 1: colorsRoutine();
    }
}

```

```

        break;
    case 2: snowRoutine();
        break;
    case 3: ballRoutine();
        break;
    case 4: rainbowRoutine();
        break;
    case 5: rainbowColorsRoutine();
        break;
    case 6: fireRoutine();
        break;
    case 7: matrixRoutine();
        break;
    case 8: ballsRoutine();
        break;
    case 9: wavesRoutine();
        break;
    case 10: starfallRoutine();
        break;
    }
    FastLED.show();
}

byte index;
String string_convert = "";
enum modes {NORMAL, COLOR, TEXT} parseMode;

void parsing() {
    if (recievedFlag) {
        recievedFlag = false;
        BTcontrol = true;
        idleTimer.reset();

        switch (intData[0]) {
            case 1:
                drawPixelXY(intData[1], intData[2],
gammaCorrection(globalColor));
                FastLED.show();
                break;
            case 2:
                fillAll(gammaCorrection(globalColor));
                FastLED.show();
                break;
            case 3:
                FastLED.clear();
                FastLED.show();
                break;
            case 4:
                globalBrightness = intData[1];
                breathBrightness = globalBrightness;
                FastLED.setBrightness(globalBrightness);

```

```

        FastLED.show();
        break;
    case 8:
        if (intData[1] == 0) {
            effect = intData[2];
            gameFlag = false;
            loadingFlag = true;
            breathBrightness = globalBrightness;
            FastLED.setBrightness(globalBrightness); //
возвращаем яркость
            globalSpeed = intData[3];
            gameTimer.setInterval(globalSpeed * 4);
        }
        else if (intData[1] == 1)
            effectsFlag = !effectsFlag;
        break;
    case 15: globalSpeed = intData[1];
        if (effectsFlag)
            effectTimer.setInterval(globalSpeed);
        break;
    }
    lastMode = intData[0];
}

if (Serial.available() > 0) {
    char incomingByte;
    if (parseMode == TEXT) {
        runningText = Serial.readString();
        incomingByte = ending;
        parseMode = NORMAL;
    } else {
        incomingByte = Serial.read();
    }
    if (parseStarted) {
        if (incomingByte != divider && incomingByte != ending)
        {
            string_convert += incomingByte;
        } else {
            if (index == 0) {
                byte thisMode = string_convert.toInt();
                if (thisMode == 0 || thisMode == 5) parseMode =
COLOR;
                else if (thisMode == 6) parseMode = TEXT;
                else parseMode = NORMAL;
            }

            if (index == 1) {
                if (parseMode == NORMAL) intData[index] =
string_convert.toInt();
                if (parseMode == COLOR) globalColor =
strtol(&string_convert[0], NULL, 16);

```

```

        } else {
            intData[index] = string_convert.toInt();
        }
        string_convert = "";
        index++;
    }
}
if (incomingByte == header) {
    parseStarted = true;
    index = 0;
    string_convert = "";
}
if (incomingByte == ending) {
    parseMode == NORMAL;
    parseStarted = false;
    recievedFlag = true;
}
}
}
}
#elif (BT_MODE == 0)
void bluetoothRoutine() {
    return;
}
#endif

GTimer_ms autoplayTimer(AUTOPLAY_PERIOD * 1000);
byte thisMode = 0;

#define MODES_AMOUNT 8

void customModes() {
    switch (thisMode) {
        case 0: snowRoutine();
            break;
        case 1: matrixRoutine();
            break;
        case 2: starfallRoutine();
            break;
        case 3: ballRoutine();
            break;
        case 4: ballsRoutine();
            break;
        case 5: wavesRoutine();
            break;
        case 6: rainbowRoutine();
            break;
        case 7: fireRoutine();
            break;
    }
    FastLED.show();
}
}

```

```

void customRoutine() {

    if (idleState) {
        if (autoplayTimer.isReady()) {
            thisMode++;
            if (thisMode >= MODES_AMOUNT) thisMode = 0;
            loadingFlag = true;
        }
    } else {
        if (idleTimer.isReady()) {
            idleState = true;
            autoplayTimer.reset();
            gameDemo = true;
            BTcontrol = false;
        }
    }
}

#define WAVES_AMOUNT 2

#define BALLS_AMOUNT 3
#define CLEAR_PATH 1
#define BALL_TRACK 1
#define DRAW_WALLS 1
#define TRACK_STEP 40

#define BALL_SIZE 3
#define RANDOM_COLOR 1

#define SPARKLES 1
#define HUE_ADD 0

#define TAIL_STEP 30
#define SATURATION 150

boolean brightnessDirection;
void brightnessRoutine() {
    if (effectTimer.isReady()) {
        if (brightnessDirection) {
            breathBrightness += 2;
            if (breathBrightness > globalBrightness - 1) {
                brightnessDirection = false;
            }
        } else {
            breathBrightness -= 2;
            if (breathBrightness < 1) {
                brightnessDirection = true;
            }
        }
    }
}

```

```

    FastLED.setBrightness(breathBrightness);
}
}

void snowRoutine() {
    if (effectTimer.isReady()) {
        for (byte x = 0; x < WIDTH; x++) {
            for (byte y = 0; y < HEIGHT - 1; y++) {
                drawPixelXY(x, y, getPixColorXY(x, y + 1));
            }
        }

        for (byte x = 0; x < WIDTH; x++) {
            if (getPixColorXY(x, HEIGHT - 2) == 0 && (random(0,
10) == 0))
                drawPixelXY(x, HEIGHT - 1, 0xE0FFFF - 0x101010 *
random(0, 4));
            else
                drawPixelXY(x, HEIGHT - 1, 0x000000);
        }
    }
}

int coordB[2];
int8_t vectorB[2];
CRGB ballColor;

void ballRoutine() {
    if (effectTimer.isReady()) {
        if (loadingFlag) {
            for (byte i = 0; i < 2; i++) {
                coordB[i] = WIDTH / 2 * 10;
                vectorB[i] = random(3, 8);
                ballColor = CHSV(random(0, 9) * 28, 255, 255);
            }
            loadingFlag = false;
        }
        for (byte i = 0; i < 2; i++) {
            coordB[i] += vectorB[i];
            if (coordB[i] < 0) {
                coordB[i] = 0;
                vectorB[i] = -vectorB[i];
                if (RANDOM_COLOR) ballColor = CHSV(random(0, 9) *
28, 255, 255);
            }
        }
        if (coordB[0] > (WIDTH - BALL_SIZE) * 10) {
            coordB[0] = (WIDTH - BALL_SIZE) * 10;
            vectorB[0] = -vectorB[0];
            if (RANDOM_COLOR) ballColor = CHSV(random(0, 9) * 28,
255, 255);
        }
    }
}

```

```

    }
    if (coordB[1] > (HEIGHT - BALL_SIZE) * 10) {
        coordB[1] = (HEIGHT - BALL_SIZE) * 10;
        vectorB[1] = -vectorB[1];
        if (RANDOM_COLOR) ballColor = CHSV(random(0, 9) * 28,
255, 255);
    }
    FastLED.clear();
    for (byte i = 0; i < BALL_SIZE; i++)
        for (byte j = 0; j < BALL_SIZE; j++)
            leds[getPixelNumber(coordB[0] / 10 + i, coordB[1] /
10 + j)] = ballColor;
    }
}

void rainbowRoutine() {
    if (effectTimer.isReady()) {
        hue++;
        for (byte i = 0; i < WIDTH; i++) {
            CHSV thisColor = CHSV((byte)(hue + i * float(255 /
WIDTH)), 255, 255);
            for (byte j = 0; j < HEIGHT; j++)
                leds[getPixelNumber(i, j)] = thisColor;
        }
    }
}

unsigned char matrixValue[8][16];
unsigned char line[WIDTH];
int pcnt = 0;

//these values are substracted from the generated values to
give a shape to the animation
const unsigned char valueMask[8][16] PROGMEM = {
    {32 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 32 , 32 , 0 , 0 , 0 ,
, 0 , 0 , 0 , 32 },
    {64 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 64 , 64 , 0 , 0 , 0 ,
, 0 , 0 , 0 , 64 },
    {96 , 32 , 0 , 0 , 0 , 0 , 32 , 96 , 96 , 32 , 0 , 0 ,
, 0 , 0 , 32 , 96 },
    {128, 64 , 32 , 0 , 0 , 32 , 64 , 128, 128, 64 , 32 , 0
, 0 , 32 , 64 , 128},
    {160, 96 , 64 , 32 , 32 , 64 , 96 , 160, 160, 96 , 64 , 32
, 32 , 64 , 96 , 160},
    {192, 128, 96 , 64 , 64 , 96 , 128, 192, 192, 128, 96 , 64
, 64 , 96 , 128, 192},
    {255, 160, 128, 96 , 96 , 128, 160, 255, 255, 160, 128, 96
, 96 , 128, 160, 255},
    {255, 192, 160, 128, 128, 160, 192, 255, 255, 192, 160,
128, 128, 160, 192, 255}

```



```

};

//these are the hues for the fire,
//should be between 0 (red) to about 25 (yellow)
const unsigned char hueMask[8][16] PROGMEM = {
  {1 , 11, 19, 25, 25, 22, 11, 1 , 1 , 11, 19, 25, 25, 22,
  11, 1 },
  {1 , 8 , 13, 19, 25, 19, 8 , 1 , 1 , 8 , 13, 19, 25, 19, 8
  , 1 },
  {1 , 8 , 13, 16, 19, 16, 8 , 1 , 1 , 8 , 13, 16, 19, 16, 8
  , 1 },
  {1 , 5 , 11, 13, 13, 13, 5 , 1 , 1 , 5 , 11, 13, 13, 13, 5
  , 1 },
  {1 , 5 , 11, 11, 11, 11, 5 , 1 , 1 , 5 , 11, 11, 11, 11, 5
  , 1 },
  {0 , 1 , 5 , 8 , 8 , 5 , 1 , 0 , 0 , 1 , 5 , 8 , 8 , 5 , 1
  , 0 },
  {0 , 0 , 1 , 5 , 5 , 1 , 0 , 0 , 0 , 0 , 1 , 5 , 5 , 1 , 0
  , 0 },
  {0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 0
  , 0 }
};

void fireRoutine() {
  if (loadingFlag) {
    loadingFlag = false;
    FastLED.clear();
    generateLine();
    memset(matrixValue, 0, sizeof(matrixValue));
  }
  if (effectTimer.isReady()) {
    if (pcnt >= 100) {
      shiftUp();
      generateLine();
      pcnt = 0;
    }
    drawFrame(pcnt);
    pcnt += 30;
  }
}

// Randomly generate the next line (matrix row)

void generateLine() {
  for (uint8_t x = 0; x < WIDTH; x++) {
    line[x] = random(64, 255);
  }
}

//shift all values in the matrix up one row

```

```

void shiftUp() {
    for (uint8_t y = HEIGHT - 1; y > 0; y--) {
        for (uint8_t x = 0; x < WIDTH; x++) {
            uint8_t newX = x;
            if (x > 15) newX = x - 15;
            if (y > 7) continue;
            matrixValue[y][newX] = matrixValue[y - 1][newX];
        }
    }

    for (uint8_t x = 0; x < WIDTH; x++) {
        uint8_t newX = x;
        if (x > 15) newX = x - 15;
        matrixValue[0][newX] = line[newX];
    }
}

// draw a frame, interpolating between 2 "key frames"
// @param pcnt percentage of interpolation

void drawFrame(int pcnt) {
    int nextv;

    //each row interpolates with the one before it
    for (unsigned char y = HEIGHT - 1; y > 0; y--) {
        for (unsigned char x = 0; x < WIDTH; x++) {
            uint8_t newX = x;
            if (x > 15) newX = x - 15;
            if (y < 8) {
                nextv =
                    (((100.0 - pcnt) * matrixValue[y][newX]
                     + pcnt * matrixValue[y - 1][newX]) / 100.0)
                    - pgm_read_byte(&(valueMask[y][newX]));

                CRGB color = CHSV(
                    HUE_ADD +
                    pgm_read_byte(&(hueMask[y][newX])), // H
                    255, // S
                    (uint8_t)max(0, nextv) // V
                );

                leds[getPixelNumber(x, y)] = color;
            } else if (y == 8 && SPARKLES) {
                if (random(0, 20) == 0 && getPixColorXY(x, y - 1) !=
0) drawPixelXY(x, y, getPixColorXY(x, y - 1));
                else drawPixelXY(x, y, 0);
            } else if (SPARKLES) {

                // старая версия для яркости
                if (getPixColorXY(x, y - 1) > 0)
                    drawPixelXY(x, y, getPixColorXY(x, y - 1));
            }
        }
    }
}

```

```

        else drawPixelXY(x, y, 0);
    }
}

//first row interpolates with the "next" line
for (unsigned char x = 0; x < WIDTH; x++) {
    uint8_t newX = x;
    if (x > 15) newX = x - 15;
    CRGB color = CHSV(
        HUE_ADD +
        pgm_read_byte(&(hueMask[0][newX])), // H
        255, // S
        (uint8_t)((100.0 - pcnt) *
matrixValue[0][newX] + pcnt * line[newX]) / 100.0) // V
    );
    leds[getPixelNumber(newX, 0)] = color;
}
}

void matrixRoutine() {
    if (loadingFlag) {
        loadingFlag = false;
        FastLED.clear();
    }
    if (effectTimer.isReady()) {
        for (byte x = 0; x < WIDTH; x++) {
            uint32_t thisColor = getPixColorXY(x, HEIGHT - 1);
            if (thisColor == 0)
                drawPixelXY(x, HEIGHT - 1, 0x00FF00 * (random(0, 10)
== 0));
            else if (thisColor < 0x002000)
                drawPixelXY(x, HEIGHT - 1, 0);
            else
                drawPixelXY(x, HEIGHT - 1, thisColor - 0x002000);
        }

        for (byte x = 0; x < WIDTH; x++) {
            for (byte y = 0; y < HEIGHT - 1; y++) {
                drawPixelXY(x, y, getPixColorXY(x, y + 1));
            }
        }
    }
}

int coord[BALLS_AMOUNT][2];
int8_t vector[BALLS_AMOUNT][2];
CRGB ballColors[BALLS_AMOUNT];

void ballsRoutine() {

```

```

if (loadingFlag) {
    loadingFlag = false;
    for (byte j = 0; j < BALLS_AMOUNT; j++) {
        int sign;

        coord[j][0] = WIDTH / 2 * 10;
        random(0, 2) ? sign = 1 : sign = -1;
        vector[j][0] = random(4, 15) * sign;
        coord[j][1] = HEIGHT / 2 * 10;
        random(0, 2) ? sign = 1 : sign = -1;
        vector[j][1] = random(4, 15) * sign;
        ballColors[j] = CHSV(random(0, 9) * 28, 255, 255);
    }
}
if (effectTimer.isReady()) {
    if (!BALL_TRACK)
        FastLED.clear();
    else {
        fader();
    }

    for (byte j = 0; j < BALLS_AMOUNT; j++) {

        if (DRAW_WALLS) {
            uint32_t thisColor = getPixColorXY(coord[j][0] / 10
+ 1, coord[j][1] / 10);
            if (thisColor == globalColor/* && vector[j][0] >
0*/) {
                vector[j][0] = -vector[j][0];
            }
            thisColor = getPixColorXY(coord[j][0] / 10 - 1,
coord[j][1] / 10);
            if (thisColor == globalColor/* && vector[j][0] <
0*/) {
                vector[j][0] = -vector[j][0];
            }
            thisColor = getPixColorXY(coord[j][0] / 10,
coord[j][1] / 10 + 1);
            if (thisColor == globalColor/* && vector[j][1] >
0*/) {
                vector[j][1] = -vector[j][1];
            }
            thisColor = getPixColorXY(coord[j][0] / 10,
coord[j][1] / 10 - 1);
            if (thisColor == globalColor/* && vector[j][1] <
0*/) {
                vector[j][1] = -vector[j][1];
            }
        }

        for (byte i = 0; i < 2; i++) {

```

```

        coord[j][i] += vector[j][i];
        if (coord[j][i] < 0) {
            coord[j][i] = 0;
            vector[j][i] = -vector[j][i];
        }
    }

    if (coord[j][0] > (WIDTH - 1) * 10) {
        coord[j][0] = (WIDTH - 1) * 10;
        vector[j][0] = -vector[j][0];
    }
    if (coord[j][1] > (HEIGHT - 1) * 10) {
        coord[j][1] = (HEIGHT - 1) * 10;
        vector[j][1] = -vector[j][1];
    }
    leds[getPixelNumber(coord[j][0] / 10, coord[j][1] /
10)] = ballColors[j];
    }
}
}
}

```

```

#define DEG_TO_RAD 0.017453

```

```

int t;

```

```

byte w[WAVES_AMOUNT];

```

```

byte phi[WAVES_AMOUNT];

```

```

byte A[WAVES_AMOUNT];

```

```

CRGB waveColors[WAVES_AMOUNT];

```

```

void wavesRoutine() {

```

```

    if (loadingFlag) {

```

```

        loadingFlag = false;

```

```

        for (byte j = 0; j < WAVES_AMOUNT; j++) {

```

```

            w[j] = random(17, 25);

```

```

            phi[j] = random(0, 360);

```

```

            A[j] = HEIGHT / 2 * random(4, 11) / 10;

```

```

            waveColors[j] = CHSV(random(0, 9) * 28, 255, 255);

```

```

        }

```

```

    }

```

```

    if (effectTimer.isReady()) {

```

```

        for (int i = WIDTH - 1; i > 0; i--)

```

```

            for (int j = 0; j < HEIGHT; j++)

```

```

                drawPixelXY(i, j, getPixColorXY(i - 1, j));

```

```

        t++;

```

```

        if (t > 360) t = 0;

```

```

        for (byte i = 0; i < HEIGHT; i++) {

```

```

            drawPixelXY(0, i, 0x000000);

```

```

        }

```

```

        for (byte j = 0; j < WAVES_AMOUNT; j++) {

```

```

        float value = HEIGHT / 2 + (float)A[j] *
sin((float)w[j] * t * DEG_TO_RAD + (float)phi[j] *
DEG_TO_RAD);
        leds[getPixelNumber(0, (byte)value)] = waveColors[j];
    }
}

void fader() {
    for (byte i = 0; i < WIDTH; i++) {
        for (byte j = 0; j < HEIGHT; j++) {
            fadePixel(i, j, TRACK_STEP);
        }
    }
}

void fadePixel(byte i, byte j, byte step) {
    uint32_t thisColor = getPixColorXY(i, j);

    if (thisColor == 0) return;

    byte rgb[3];
    rgb[0] = (thisColor >> 16) & 0xff;
    rgb[1] = (thisColor >> 8) & 0xff;
    rgb[2] = thisColor & 0xff;

    byte maximum = max(max(rgb[0], rgb[1]), rgb[2]);
    float coef = 0;

    if (maximum >= step)
        coef = (float)(maximum - step) / maximum;

    for (byte i = 0; i < 3; i++) {
        if (rgb[i] > 0) rgb[i] = (float)rgb[i] * coef;
        else rgb[i] = 0;
    }
    leds[getPixelNumber(i, j)] = CRGB(rgb[0], rgb[1], rgb[2]);
}

void starfallRoutine() {
    if (effectTimer.isReady()) {
        for (byte i = HEIGHT / 2; i < HEIGHT; i++) {
            if (getPixColorXY(0, i) == 0
                && (random(0, 60) == 0)
                && getPixColorXY(0, i + 1) == 0
                && getPixColorXY(0, i - 1) == 0)
                leds[getPixelNumber(0, i)] = CHSV(random(0, 200),
SATURATION, 255);
        }
        for (byte i = 0; i < WIDTH / 2; i++) {
            if (getPixColorXY(i, HEIGHT - 1) == 0

```

```
        && (random(0, 60) == 0)
        && getPixColorXY(i + 1, HEIGHT - 1) == 0
        && getPixColorXY(i - 1, HEIGHT - 1) == 0)
    leds[getPixelNumber(i, HEIGHT - 1)] = CHSV(random(0,
200), SATURATION, 255);
}

for (byte y = 0; y < HEIGHT - 1; y++) {
    for (byte x = WIDTH - 1; x > 0; x--) {
        drawPixelXY(x, y, getPixColorXY(x - 1, y + 1));
    }
}

for (byte i = HEIGHT / 2; i < HEIGHT; i++) {
    fadePixel(0, i, TRACK_STEP);
}
for (byte i = 0; i < WIDTH / 2; i++) {
    fadePixel(i, HEIGHT - 1, TAIL_STEP);
}
}
```