

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Циклова комісія з інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Голова циклової комісії,
спеціаліст в/к

_____ С.О. Сабанов

ВИПУСКНА РОБОТА МОЛОДШОГО СПЕЦІАЛІСТА

ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС
"КОНТРОЛЕР ДЛЯ СВІТЛОДІОДНИХ RGB-СТРІЧОК"

Виконав

ст. гр. КІ – 118К9

В.Р. Воробйов

Керівник

професор

С.О. Сабанов

Запоріжжя

2023

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»
Циклова комісія з інформаційних технологій

ЗАТВЕРДЖУЮ

Голова циклової комісії,
спеціаліст в/к

_____ С.О. Сабанов

03 жовтня 2022 р.

З А В Д А Н Н Я
НА ВИПУСКНУ РОБОТУ МОЛОДШОГО СПЕЦІАЛІСТА

студенту гр. КІ-118К9,

спеціальності 124 - «Комп'ютерна інженерія»

Воробйову Владиславу Руслановичу

1.Тема: «Програмно-апаратний комплекс "Контролер для світлодіодних RGB-стрічок"»

затверджена наказом № 09.2-19 від 04 березня 2022 р.

2. Термін здачі студентом закінченої роботи: 12 січня 2023 р.

3. Перелік питань, що підлягають розробці:

1. Провести огляд предметної області та аналітику проблеми.
2. Провести огляд та аналіз популярних аналогів, зробити висновки про вимоги до проекту.
3. Провести вибір та огляд апаратної складової проекту.
4. Провести вибір та огляд програмної складової проекту.
5. Здійснити проектування розробляємої системи.
6. Здійснити програмування мікроконтролеру розробляємої системи.
7. Здійснити зборку та налагодження розробленої системи.

8. Оформити звіт за результатами роботи

4. Календарний графік підготовки випускної роботи молодшого спеціаліста

№ етапу	Зміст	Терміни виконання	Готовність по графіку %, підпис керівника	Підпис керівника про повну готовність етапу, дата
1	Формулювання (корегування) теми випускної роботи молодшого спеціаліста та збір практичного матеріалу за темою випускної роботи	20.10.2022		
2	I атестація I розділ випускної роботи молодшого спеціаліста	27.10.2022		
3	II атестація II розділ випускної роботи молодшого спеціаліста	17.11.2022		
4	III атестація III розділ випускної роботи молодшого спеціаліста, висновки та рекомендації, додатки, реферат	29.12.2022		
5	Перевірка випускної роботи молодшого спеціаліста програмою «Антиплагіат»	29.12.2022		
6	Доопрацювання випускної роботи молодшого спеціаліста, підготовка презентації, отримання відгуку керівника і рецензії	10.01.2023		
7	Попередній захист випускної роботи молодшого спеціаліста	13.01.2023		
8	Подача випускної роботи молодшого спеціаліста на кафедру	за 3 дні до захисту		
9	Захист випускної роботи молодшого спеціаліста	20.01.2023		

Дата видачі завдання: 03 жовтня 2022 р.

Керівник

_____ (підпис)

С.О. Сабанов
(ініціали, прізвище)

Завдання отримав до виконання

_____ (підпис студента)

В.Р. Воробйов
(ініціали, прізвище)

РЕФЕРАТ

Випускна робота молодшого спеціаліста містить 70 сторінок, 5 таблиць, 30 рисунків, 2 додатка, 24 бібліографічних посилань.

Метою роботи є розробка системи освітлення зі світлодіодної RGB-стрічки з мікроконтролерним керуванням.

У випускній роботі розглядаються основні системи сучасного освітлення та принципи їхнього функціонування. Основну увагу приділено світлодіодним стрічкам: типам, моделям, характеристикам та методам управління.

Наведено опис процесу розробки і реалізації робочої моделі освітлювальної системи на базі RGB LED-стрічки, що працює під управлінням мікропроцесора.

В роботі використані спеціалізовані середовища розробки та програмування, зокрема Proteus Design Suite 8.13 та Arduino IDE 1.8.19.

ARDUINO IDE, ARDUINO UNO, ATMEGA328P, LED, RGB-СТРІЧКА, SMD,
WS1228B, КОЛЬОРОВА МОДЕЛЬ, МІКРОКОНТРОЛЕР, СИСТЕМА
ОСВІТЛЕННЯ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	10
РОЗДІЛ 1 ТЕХНОЛОГІЧНІ ОСНОВИ ПОБУДОВИ СИСТЕМ ОСВІТЛЕННЯ	11
1.1 Типи джерел освітлення.....	11
1.2 Типи світлодіодних стрічок.....	13
1.3 Кольорові моделі.....	15
1.3.1 Кольорова модель RGB.....	15
1.3.2 Кольорова модель CMYK.....	17
1.3.3 Кольорова модель HSB.....	18
1.3.4 Кольорова модель LAB.....	19
1.4 Огляд існуючих аналогів.....	20
1.5 Опис розробляемого проекту.....	23
1.6 Висновки за першим розділом.....	24
РОЗДІЛ 2 ВИБІР ТА ОБГРУНТУВАННЯ СКЛАДОВИХ ПРОГРАМНО- АПАРАТНОГО КОМПЛЕКСУ.....	25
2.1 Вибір платформи та мікроконтролеру.....	25
2.2 Переривання в Arduino.....	28
2.3 Вибір системи освітлення.....	31
2.3.1 Принцип роботи SMD RGB світлодіоду WS1228B.....	31
2.3.2 Огляд адресної світлодіодної стрічки WS1228B.....	33
2.3.3 Опис роботи LED-світлодіоду.....	35
2.4 Підключення натискної кнопки.....	38
2.5 Робота зі змінними резисторами.....	42
2.5.1 Робота з фоторезистором.....	42

2.5.2 Робота з потенціометром.....	44
2.6 Робота з інфрачервоним приймачем.....	45
2.7 Робота з датчиком руху.....	47
2.8 Вибір програмного інструментарію.....	49
2.8.1 Вибір середовища розробки.....	49
2.8.2 Вибір мови програмування.....	53
2.8.3 Вибір середовища проектування та моделювання.....	55
2.9 Висновки за другим розділом.....	56
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ.....	58
3.1 Проектування апаратного комплексу системи.....	58
3.2 Програмування комплексу системи.....	63
3.2.1 Використання зовнішніх бібліотек.....	63
3.2.2 Зчитування кнопок пульту дистанційного керування.....	66
ВИСНОВКИ.....	69
ПЕРЕЛІК ПОСИЛАНЬ.....	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

Слово/словосполучення	Скорочення
A	
Ампер	A
Alternating Current (змінний струм)	AC
C	
Сяан, Magenta, Yellow, Black color	СМУК
D	
Direct Current (постійний струм)	DC
	DIP
E	
Electrically Erasable Programmable Read-Only Memory	EEPROM
G	
Ground	GND
H	
Hue, Saturation, Brightness	HSB
Hue, Saturation, Lightness	HSL
I	
Інтегроване середовище розробки	IDE
Infrared (інфрочервоний)	IR
Inter-Integrated Circuit	I2C
L	
light-emitting diode	LED
M	
Microcontroller Unit	MCU
O	
Operating system	OS

Organic light-emitting diode	OLED
P	
Personal computer	PC
R	
Red, Green, Blue	RGB
S	
Surface Mounted Device	SMD
Serial Peripheral Interface	SPI
Static Random Access Memory	SRAM
T	
Transistor-Transistor Logic	TTL
U	
Universal Asynchronous Receiver/Transmitter	UART
Universal Serial Bus	USB
V	
Volt	V
I	
Інфрачервоне випромінювання	ІЧ
К	
кілоОМ (одиниця вимірювання)	кОМ
Л	
Люкс	лк
М	
міліампер	мА
Мегагерци	МГц
Мілікандели	мкд
Мікросекунди	мкс

П	
Програмне забезпечення	ПЗ
С	
Сантиметр	см
Ц	
Цифро-аналоговий перетворювач	ЦАП
Ш	
Широтно-імпульсна модуляція	ШІМ

ВСТУП

У наш час елементи RGB – освітлення набувають широкого поширення у повсякденному житті людини. Насамперед використання гірлянд, стрічок та аналогічних приладів актуально під час свят, розважальних заходів, тощо. Однак теперішня ситуація в Україні з постійними знеструмленнями надає таким приладам ще одного застосування – як основне освітлення під час постійних виключень електричного струму. Тому створення такої системи є актуальною задачею.

На сьогоднішній день надзвичайно широке застосування знайшли світлодіодні стрічки, які з'явилися навіть раніше, ніж світлодіодні лампи. Ними виконують декоративне підсвічування стель, різних ніш, робочих зон. Світлодіодну енергозберігаючу стрічку часто встановлюють у вітрини, стелажі та інше торгове обладнання для підсвічування товару, замість люмінесцентних світильників, що морально застарівають і постійно виходять з ладу. При використанні стрічки зі зміною кольору (RGB стрічки) вдається отримати унікальні світлові ефекти. Яскравість і кольоровість такої стрічки може змінюватися з пультом дистанційного керування. Монохромними стрічками оформлюють кафе, ресторани, інші місця відпочинку, а у вологозахищеному виконанні таку стрічку використовують як вуличне декоративне підсвічування. Іншими словами, завдяки різноманітності варіантів виконання світлодіодна стрічка стала незамінною і використовується в найрізноманітніших випадках оформлення.

Елементна база стрічок дозволяє створювати прості й надійні системи керування ними. Особливо виділяються прилади з мікроконтролерним управлінням, через те, що користувач має можливість власноруч обирати режими роботи приладу, створювати безліч кольорових схем, здійснювати персональні налаштування, та зберігати їх у пам'яті.

В роботі надається опис розробки однієї з можливих конструкцій системи управління RGB-стрічкою, що базується на широко розповсюджених недорогих компонентах.

РОЗДІЛ 1

ТЕХНОЛОГІЧНІ ОСНОВИ ПОБУДОВИ СИСТЕМ ОСВІТЛЕННЯ

1.1 Типи джерел освітлення

Освітлення робочої зони і робочих місць може бути природним і штучним [1].

Види природного освітлення визначаються архітектурою будівель. Природне світло буває:

- бічним – таким, що проходить через віконні отвори;
- верхнім – що надходять крізь засклений дах або технологічний отвір;
- комбінованим – поєднує вищеперелічені варіанти.

Можливі типи освітлення:

- Основне (рівномірне) – світловий потік поширюється по всьому простору приміщення з однаковою рівномірністю. Це функціональне рішення реалізуються за допомогою установки стельових світильників, наприклад точкових, з лінзами, що розсіюють, і кутом розсіювання до 360 градусів.
- Локалізоване (місцеве) – спосіб для акцентного підсвічування окремих ділянок, наприклад місця відпочинку, робочого столу, кухонної стільниці. Застосовуються освітлювальні джерела із вузьким пучком світіння не більше 30-60 градусів.

Можна розглянути ряд ламп для освітлення та подачі світла штучно (рис. 1.1):

- натрієві лампи (рис. 1.1 (а));
- ртутні галогенні лампи (рис. 1.1 (б));
- лампи розжарювання (рис. 1.1 (в));

- люмінесцентні лампи (рис. 1.1 (г));
- світлодіодна лампа (рис. 1.1 (д));
- світлодіодні стрічки (рис. 1.1 (е)).



(а)



(б)



(в)



(г)



(д)



(е)

Рисунок 1.1 – Види ламп та систем освітлення: (а) – натрієва лампа; (б) – галогенна лампа; (в) – лампа розжарювання; (г) – люмінісцентні лампи; (д) – світлодіодна лампа; (е) – RGB – стрічка

1.2 Типи світлодіодних стрічок

Сучасний світлотехнічний ринок на сьогоднішній день пропонує безліч найрізноманітніших типів світлодіодних стрічок. Традиційно стрічка представляє собою гнучку тонку друковану плату з нанесеними на ній методом поверхневого монтажу (частіше званим SMD) різного розміру світлодіодами. Як правило, світлодіодна стрічка з протилежного боку оснащена клеючою основою за принципом двостороннього липкого скотчу, але у виняткових випадках крім або замість скотчу стрічка поставляється зі

спеціальними кріпленнями (переважно з вологозахищеними світлодіодними стрічками). У більшості випадків, світлодіодна стрічка йде бухтами по 5 метрів. Кожна бухта має заводські припаяні струмопідвідні дроти (найчастіше і на початку бухти, і в кінці).

Основні параметри, за якими світлодіодні стрічки відрізняються одна від одної:

- тип світлодіодів;
- напруга;
- кількість світлодіодів;
- ступінь захисту від вологи та пилу;
- колір.

У стрічках використовують такі основні типи SMD світлодіодів:

- SMD3010 – світлодіод розміром 3x1мм (використовуються в торцевих (бічного світіння) стрічках);
- SMD3528 – світлодіод розміром 3,5x2,8мм (найпоширеніші, використовуються у стрічках горизонтального світіння);
- SMD5050 – світлодіод розміром 5x5мм (світлодіоди підвищеної яскравості за рахунок того, що всередині одного діода використовується 3 кристали).

Від загальної кількості світлодіодів на друкованій платі залежить яскравість та потужність усієї стрічки. Як правило, у маркуванні світлодіодної стрічки вказується тип використовуваних світлодіодів, їх кількість та загальна споживана потужність. Світлодіодні стрічки зі світлодіодами SMD3528 виготовляються з кількістю в 60, 120 і 240 світлодіодів на метр, а світлодіодні стрічки з діодами SMD5050 – 30, 60 і 120 світлодіодів на метр.

Виникають випадки, коли світлодіодну стрічку необхідно застосовувати в умовах підвищеної вологості або при перепадах температур

(у тюнінгу машин та мотоциклів, при монтажі стрічки у ванних кімнатах, на балконах, басейнах, на вулиці, у світлових коробах тощо). Саме в таких ситуаціях використовують світлодіодну стрічку з різним ступенем захисту. Ступінь захисту – це міжнародний параметр і позначається як IP.

Стрічка буває:

- IP20 – невологозахищена відкрита світлодіодна стрічка. Таку стрічку, як правило, використовують для підсвічування вітрин, стель, полиць, меблів, інтер'єрної реклами та декоративного підсвічування.
- IP65 – світлодіодна стрічка, яка зверху покрита епоксидною смолою. Таку стрічку найчастіше використовують для підсвічування салону автомобіля, ландшафту, світлових коробів, ванних кімнат, тобто в приміщеннях з підвищеною вологістю. У зв'язку з тим, що плату захищено силіконом тільки зверху, для агресивних середовищ вона не підійде.
- IP68 – світлодіодна стрічка з максимальним ступенем захисту. Друкована плата такої стрічки повністю залита з двох сторін силіконом і, більше того, додатково захищена силіконовим прозорим чохлам. Вологозахищену світлодіодну стрічку сміливо можна використовувати під днищем автомобіля, занурювати у воду, застосовувати для світлового оформлення фонтанів, акваріумів та в інших відкритих і схильних до вологи і пилу місцях.

Категорії розподілу діодів показано на рисунку 1.2.

Світлодіодні стрічки мають такі кольори світіння: білий (який відрізняється відтінками: теплий білий і холодний білий), червоний, жовтий, синій, зелений та багатобарвний RGB. Унікальність різнокольорової світлодіодної стрічки RGB в тому, що вона може світитися різними кольорами: помаранчевим, рожевим, фіолетовим, бузковим, смарагдовим, блакитним і багатьма іншими кольорами. Різні відтінки створюються завдяки

керуваному контролеру, за допомогою пульта якого можна задавати і варіювати не тільки колір стрічки, але і динаміку, порядок і швидкість зміни кольору.



Рисунок 1.2 – Категорії розподілу діодів для різних типів захисту стрічок

Залежно від інтенсивності світіння діодів кожного з кольорів, можна домогтися різних відтінків та отримати цікаве освітлення з використанням всього лише одного пристрою.

1.3 Кольорові моделі

Кольорова модель – абстрактна модель опису представлення кольорів у вигляді наборів певної кількості чисел. Зазвичай представлена трьома або чотирма значеннями, що названі колірними компонентами або колірними координатами [3].

Множина кольорів моделі визначає колірний простір системи. Існує основна група набору колірних моделей:

- RGB;
- CMYK;
- Lab;
- HSB;
- HSL;
- Grayscale та ін.

1.3.1 Кольорова модель RGB

RGB або «Red, Green, Blue» – стара колірна модель, в якій кожен колір представлений, як сума червоного, зеленого та синього кольорів. Розтлумачити або перекласти слова з англійської:

- ✓ Red – червоний;
- ✓ Green – зелений;
- ✓ Blue – синій.

Виведення зображень на екрани виконується саме за допомогою цієї моделі. Три основних кольори продуцують ще три проміжних відтінки, як показано на рисунку 1.3. Найчастіше цей процес ілюструють трьома колірними колами. Вибір основних відтінків обумовлений специфікою будови людського ока [4]. Також на рисунку можна розгледіти ще набір майже чотирьох складових кольорів:

- ✓ Magenta – бузковий, пурпуровий.
- ✓ Yellow – жовтий.
- ✓ White – білий.
- ✓ Cyan – блакитний або колір моря.

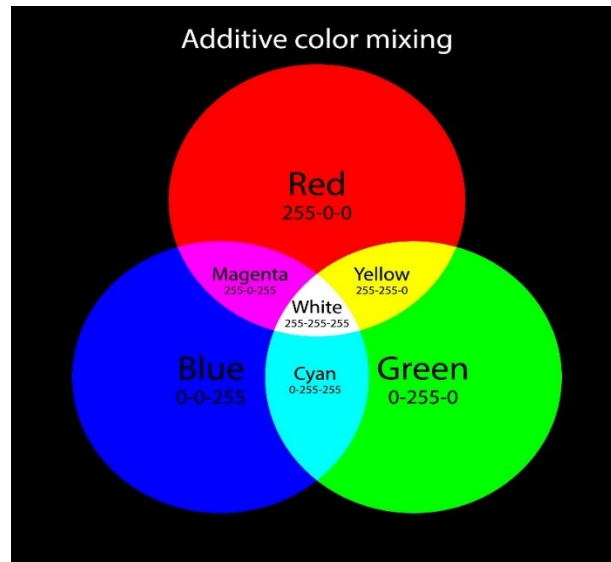


Рисунок 1.3 – Проста колірна модель RGB

При змішуванні одного з трьох основних кольорів (основними кольорами вважаються червоний, зелений та синій) – наприклад, синього (B) та червоного (R), ми отримуємо пурпуровий (M), при змішуванні зеленого (G) та червоного (R) – жовтий (Y), при змішуванні зеленого (G) та синього (B) – блакитного (C). При змішуванні всіх трьох колірних компонент ми отримуємо білий колір (W).

1.3.2 Кольорова модель СМҮК

Субтрактивна колірна модель СМҮК описує синтез друкарських фарб. Для опису одного пікселя стандартного зображення використовується 32 біта інформації (4 байти). Як випливає з назви аббревіатури колірної моделі, СМҮК – це скорочення від назви чотирьох фарб, що використовуються в друку: синьої (Cyan), пурпурової (Magenta), жовтої (Yellow), і чорної (black). [5]

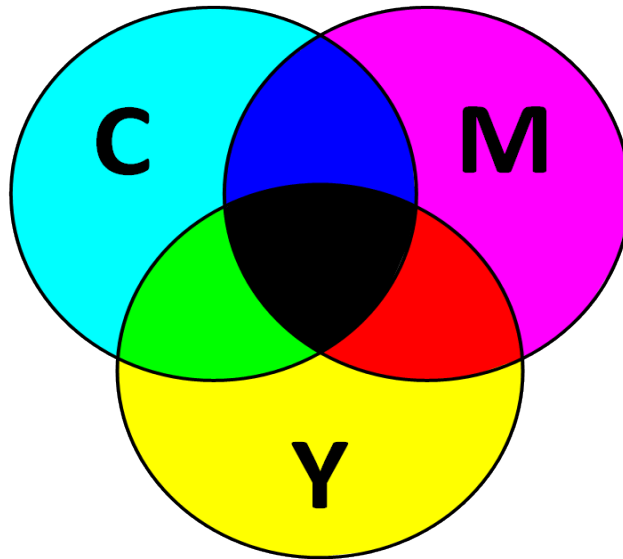


Рисунок 1.4 – Система моделі СМУК

Теоретично, для отримання будь-якого кольору, який знаходиться в рамках колірної моделі, достатньо 3 фарб без використання чорної (малюнок 1.4). Практично ж, через фізичні недоліки фарб (насамперед, недосконалість використаного в фарбах пігменту, в основному, через синю складову) при змішуванні 100% суан, magenta і yellow у друку виходить темний брудно-коричневий колір. Саме тому для "підтримки" чорної складової і був введений четвертий компонент друкованих фарб – чорна фарба. При кольороподілі (переході до субтрактивної моделі СМУК) деяка частина "сірого" кольору, утвореного трьома компонентами фарб Суан, Magenta, Yellow, частково замінюється чорною фарбою.

1.3.3 Кольорова модель HSB

HSB (Hue – Saturation – Brightness; Відтінок – Насиченість – Яскравість) – колірна модель, заснована на фізичному сприйнятті колірної інформації людським оком. Завдання кольору в цій колірній моделі виконується за допомогою трьох каналів, що видно з її назви. Перший

параметр – Hue (Відтінок) – задає колір, що відображається від об'єкта або випромінюється цим об'єктом. Конкретний колір задається в градусах, що визначає розташування цього кольору на стандартному колі кольору (рисунок 1.5). [6].

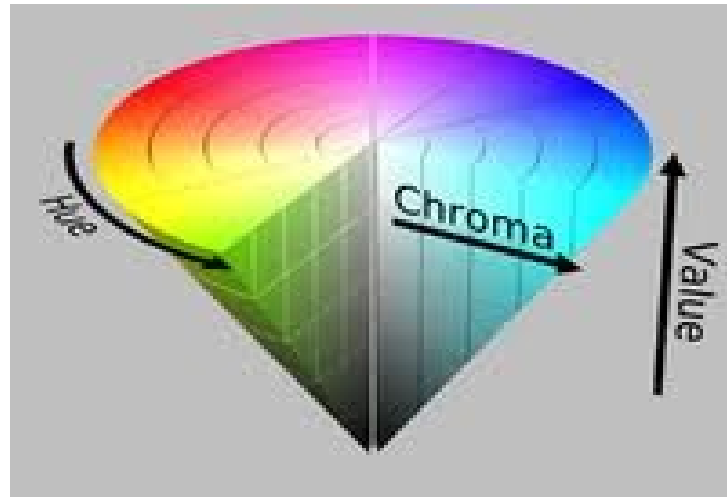


Рисунок 1.5 – Кольорова модель HSB

Значення цього параметра може змінюватися від 0 до 360 (параметр В на малюнку, змінюється по всій довжині кола).

Другий параметр – Saturation (Насиченість) визначає активність цього відтінку. Змінюється від 0 до 100 %. При нульовому значенні параметра будь-який відтінок є просто сірим нейтральним кольором. При 100% – максимальна насиченість будь-якого кольору в цій точці.

Третій параметр – Brightness (Яскравість, параметр на малюнку, збільшується знизу вгору) є відносним значенням яскравості, що вимірюється у відсотках від 0 до 100. При 0% колір точки буде чорним, при 100% - білим.

Пряма підтримка цієї колірної моделі у переважній більшості графічних редакторів відсутня, тобто. у цій колірній моделі не можна редагувати або створювати зображення, але існує низка фільтрів і прийомів впливу на зображення в цій моделі.

1.3.4 Кольорова модель LAB

L^*a^*b – Lightness – a-component – b-component – колірна модель, заснована на значеннях яскравості (канал Lightness – Яскравість), та двох колірних компонентах, що визначають положення потрібного кольору в координатах по двох осях (Component a – вісь зеленого – червоного кольору, Component b – вісь синього – жовтого кольору, значення змінюються від -127 до +127).

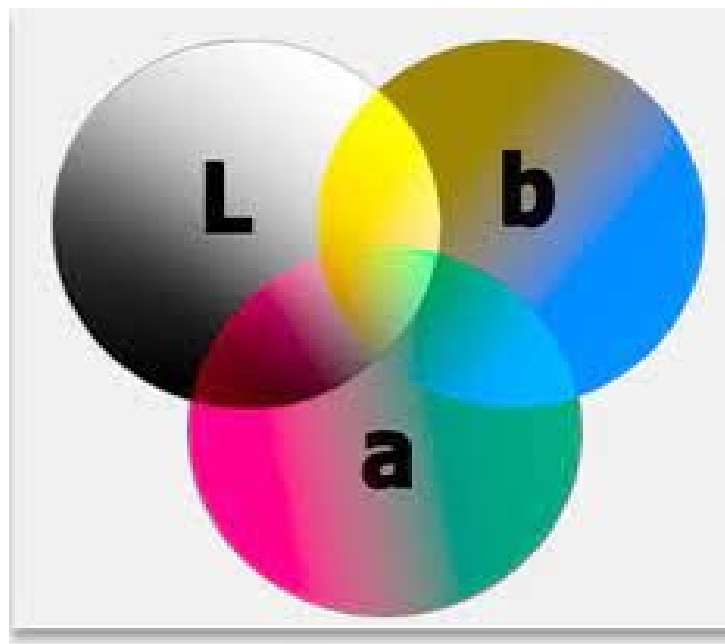


Рисунок 1.6 – Кольорова модель LAB

Модель має широкий колірний кут охоплення і може описувати кольори, відтворення яких неможливе ні в умовах друку, ні на моніторі. Растрові зображення в цій колірній моделі зберігаються, наприклад, у файлах формату Kodak Photo CD, а сама колірна модель Lab в основному використовується як проміжна модель для зберігання растрових зображень.

Модель є також і апаратно-незалежною моделлю, що дозволяє описувати колір у ній без прив'язки до якогось конкретного пристрою відтворення. [7]

1.4 Огляд існуючих аналогів

У даний час на ринку представлено велику кількість різноманітних приладів освітлення на базі світлодіодних стрічок, вартість яких починається з 200 гривень. Ці найдешевші моделі складаються в основному з наступних елементів: сама світлодіодна стрічка різних видів та метражу, простого контролера, системи керування (у більшості випадків IR-пульт), блока живлення (220AC/12DC).

Розглянемо найпоширенішу модель світлодіодної стрічки Solar RGB 3528 10 м з контролером та пультом дистанційного керування від відомого виробника Solar Electric за ціною 700 грн [8].

Стрічка складається з гнучкої основи, на яку нанесено струмопровідні доріжки, світлодіоди, токообмежуючі резистори та керуючі елементи. Ширина стрічки становить 1 см при товщині близько 2 мм. Стрічку фасують в бобіни по 10 метрів. LED стрічка має високий рівень яскравості і великий кут світіння, що дорівнює 120° , тож відмінно підійде для підсвічування.



Рисунок 1.7 – Світлодіодна стрічка Solar RGB 3528

На стрічці розташовані LED, кожен з яких складається з трьох кольорів: червоного, синього та зеленого. Особливістю даної стрічки є можливість вибору кольорів і відтінків освітлення. Багатобарвна світлодіодна стрічка SMD 3528 RGB працює від напруги 12V постійного струму, що забезпечує безпеку та економічність використання. Технічні характеристики наведені в таблиці 1.1

Таблиця 1.1

Технічні характеристики багатобарвної світлодіодної стрічки
SMD 3528 RGB

Режими	перемикання (швидко), перемикання (повільно), плавне перемикання (швидко), плавне перемикання (повільно), статичний режим
Кількість кольорів та відтінків	16
Джерело живлення	постійний струм 12 V, 2 A

Кут свічення	120 градусів
Модель світлодіодів	SMD 3528
Розмір	довжина: 10 м, ширина: 1 см, висота: 0.25 см
Комплектація	стрічка (10м), контролер, блок живлення, ІЧ-пульт

Однак дана система (як і багато інших) мають ряд недоліків:

- напруга живлення складає 12 Вольт, що майже унеможливорює використання автономних систем (PowerBank), які можуть надавати напругу 5 Вольт;
- відсутні енергозощаджувальні механізми включення / виключення за наявності людей у приміщенні, доцільність освітлення вдень тощо;
- мають лише встановлені алгоритми світіння, які користувачів вже не здивовують.

Враховуючі вищенаведене, доцільно створити нову систему на базі використання мікроконтролера, систем сенсорів та актуаторів з новими схемами освітлення.

1.5 Опис розробляемого проекту

Наступним кроком є схематичне представлення розробляемого проекту, який представлено на рисунку 1.8. Він складається з наступних блоків:

- ✓ блок мікроконтролера – центральний блок, що здійснює керування та взаємодію всієї системи;
- ✓ система датчиків та сенсорів – необхідна для отримання даних зовні для роботи системи;

- ✓ блок взаємодії з користувачем – передбачає механізми вводу та виводу інформації для користувача;
- ✓ система освітлення – складається зі світлодіодної стрічки;
- ✓ система живлення – необхідна для повноцінної роботи системи.



Рисунок 1.8 – Проект розробляємої системи

Для отримання інформації від користувача передбачено тактові кнопки та пульт дистанційного керування. Передбачено три службових світлодіоди для виводу для користувача необхідної інформації про стан системи.

Для визначення присутності користувачів бажано додати до системи датчик руху. Для визначення додаткових умов роботи системи передбачено використання змінних резисторів (фоторезистора та потенціометра).

1.6 Висновки за першим розділом

В основу роботи кольорової світлодіодної стрічки покладено традиційну модель опису палітри RGB, що використовується для випромінюючих світло пристроїв.

Гнучка світлодіодна стрічка RGB – це освітлювальний модуль, який складається з гнучкої основи, нанесених на неї струмопровідних доріжок, резисторів та світлодіодів.

Електричні параметри та елементи стрічки дозволяють створити систему керування на базі контролера з використанням мікропроцесора, що дозволить відпрацьовувати довільні алгоритми управління діодами стрічки. Проект такої системи включає в себе блок мікроконтролера, систему датчиків та сенсорів, блок взаємодії з користувачем, систему освітлення та живлення.

РОЗДІЛ 2

ВИБІР ТА ОБГРУНТУВАННЯ СКЛАДОВИХ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

2.1 Вибір платформи та мікроконтролеру

Для даного проекту доцільним буде використання платформи ARDUINO UNO R3 на базі мікроконтролеру ATmega328p.

Технічні характеристики контролера:

- тактова частота 16 МГц;
- 14 цифрових портів, що працюють на вхід/вихід (6 з них підтримують режим широтно-імпульсної модуляції);
- USB порт;
- 6 аналогових входів;
- роз'єм живлення;
- роз'єм внутрішньо схемного програмування;
- кнопка RESET. [9]



Рисунок 2.1 – Зовнішній вид плати Arduino Uno

У плати є всі компоненти, що необхідні для забезпечення роботи мікроконтролера. Живляча напруга подається через USB кабель, наприклад, від комп'ютера.

Плату можна програмувати безпосередньо з середовища Arduino IDE. Резидентний завантажувач забезпечує управління програмуванням по протоколу STK500. Апаратний програматор в цьому випадку не потрібен. Також можна запрограмувати мікроконтролер через роз'єм ICSP (внутрішньосхемне програмування) Завантажувач при цьому не використовується.

Опис функціонального призначення портів плати наведено в додатку А.

Arduino UNO R3 є нащадком контролера Arduino Diecimila та Duemilanove. Вони відрізняються лише USB-UART мостом. За комерційною змовою компаній Atmel та Arduino, остання перейшла на встановлення у свої плати Arduino, програмно залежні мости на мікроконтролерах ATmega16U2, на місце чіпів компанії FTDI FT232R. Arduino Uno можна живити як від USB, так і від зовнішнього джерела живлення, при підборі джерела живлення також враховуйте споживання підключених до Arduino UNO додаткових плат та модулів. Контролер Ардуіно Uno здатний працювати при підключенні до нього зовнішньої напруги на контакт Vin від 6 до 20 В. Підключення напруги більше 12 В, без додаткового охолодження стабілізатора може призвести мікросхему стабілізатора до перегріву та пошкодження. Для підключення живлення можуть бути використані наступні виводи:

- ✓ Vin – Живлення плати від зовнішнього джерела.
- ✓ 5 V – Вихід стабілізатора напруги плати. На ньому напруга 5 В
- ✓ V 3 – Напруга 3,3 В від стабілізатора напруги на платі.
- ✓ GND – Загальний провід.

- ✓ IOREF – Виведення інформації про робочу напругу плати. Плата розширення може визначити значення сигналу і переключитися на режим живлення 5 В або 3,3 В.

У мікроконтролера три типи пам'яті:

- 32 кБ флеш (FLASH);
- 512 кБ оперативної пам'яті (SRAM);
- 1 кБ незалежній пам'яті (EEPROM);

Кожен з 14 цифрових виводів може бути використаний в якості виходу або входу. Рівень напруги на виводах 5 В. Гранично допустиме значення струму складає 40 мА. Кожен вивід має внутрішній підтягаючий резистор опором 20-50 кОм. Резистор може бути відключений програмно.

Світлодіод, підключений до виходу 13. Світиться при високому рівні сигналу на виводі. Інтерфейс TWI: вивід A4 або SDA і A5 або SCL.

У плати Arduino UNO є 6 аналогових входів, позначених A0-A5. Роздільна здатність аналогового цифрового перетворення 10 розрядів. За замовчуванням, вхідна напруга вимірюється щодо землі в діапазоні 0-5 В, але може бути змінено за допомогою виведення AREF і програмних установок. Ще 2 виведення плати мають функції: AREF. Низький рівень на цьому виходу викликає скидання мікроконтролера.

Модуль Arduino UNO має засоби для зв'язку з комп'ютером, з іншою платою UNO або з іншими мікроконтролерами. Для цього на платі існує інтерфейс UART з логічними рівнями TTL (5 В), пов'язаний з виходами 0 (RX) і 1 (TX). Мікросхема ATmega16U2 на платі пов'язує UART інтерфейс з USB портом комп'ютера. При підключенні до порту комп'ютера, з'являється віртуальний COM порт, через який програми комп'ютера працюють з Ардуіно.

2.2 Переривання в Arduino

Переривання є механізмом, за допомогою якого можна тимчасово призупинити нормальну послідовність дій (нормальне функціонування) і змусити його обслуговувати переривання, яке в даному випадку має найвищий пріоритет серед усіх дій, які він виконує. Наприклад, нормальне функціонування мікропроцесора може перервати який-небудь датчик і запустити послідовність дій, що є у програмі (функції, процедурі) обслуговування переривань (ISR – Interrupt Service Routine). Після виконання програми обслуговування переривання мікропроцесор може повернутися до нормального функціонування.

Переривання в Arduino можна поділити на кілька видів:

- ✓ Апаратні переривання – виникають при настанні якоїсь зовнішньої події, наприклад, стан контакту зміниться з LOW на HIGH або з HIGH на LOW..
- ✓ Програмні переривання – виникають під час виконання будь-яких інструкцій у тілі програми. Наприклад, переривання від таймерів Arduino є програмними перериваннями.
- ✓ Внутрішні переривання – генеруються у випадку порушення у виконанні програми (наприклад, при переповненні розрядної сітки, зверненні до адреси, що є неприпустима, при некоректній операції тощо).

Апаратні переривання обробляються на апаратному рівні, тому їхня обробка відбувається дуже швидко. Ці переривання можна налаштувати на події збільшення або падіння рівня або просто на низький рівень напруги.

Номери контактів для використання зовнішніх переривань – 2 та 3 [11]. У платах Arduino можна значно збільшити кількість контактів для обробки переривань. У платах Arduino, побудованих на основі мікроконтролерів ATmega168/328, кожен з 20 сигнальних контактів можна налаштувати на

обробку переривання. Їх також можна налаштувати на події збільшення чи падіння рівня.

2.3 Вибір системи освітлення

Основне призначення проекту – сучасний прилад для роботи з освітленням. Світлодіодні стрічки зараз є актуальними приладами для даних систем. Тому необхідно розглянути принципи роботи таких стрічок та здійснити вибір необхідної до проекту стрічки.

2.3.1 Принцип роботи SMD RGB світлодіоду WS1228B

WS2812B – це RGB світлодіод із вбудованим ШИМ контролером в одному корпусі (рис. 2.3).



Рисунок 2.3 – Зовнішній вид LED світлодіоду WS 2812B у SMD корпусі

Відмінністю від інших RGB світлодіодів є наявність адресації – тобто кожним світлодіодом у схемі можна керувати індивідуально. Для цього використовується послідовна однопровідна шина, вхід DIn підключається до мікроконтролера, а вихід DOut – до входу наступного світлодіода, при цьому в один ланцюжок можна з'єднати до 1000 таких світлодіодів [12].

Характеристики світлодіоду наведені у таблиці 2.1.

Таблиця 2.1

Базові характеристики SMD світлодіоду WS1228B

Характеристика	Значення
Робоча напруга	3.5...5.3В
Максимальний струм	60 мА
Інтенсивність випромінювання червоний колір (625 Нм)	550-700 мкд
Інтенсивність випромінювання зелений колір (520 Нм)	1100-1400 мкд
Інтенсивність випромінювання синій колір (470 Нм)	200-400 мкд
Кут випромінювання	120 °
Тип корпусу	5050

Алгоритм управління світлодіодами WS2812B реалізований так, що після того, як перший світлодіод приймає 24 інформаційні біти з інформацією про яскравість кожного з вбудованих діодів, виконується запис цих бітів у внутрішню пам'ять. Всі наступні імпульси передаються на вихід по ланцюжку іншим світлодіодам. Після виконання передачі для всіх світлодіодів витримується технологічна пауза протягом 50 мікросекунд. Це потрібно для оновлення стану світлодіодів.

2.3.2 Огляд адресної світлодіодної стрічки WS1228B

За допомогою адресної світлодіодної стрічки можна створювати світломузику, розумне підсвічування для телевізора, рядки, що біжать, та інші проекти, в яких потрібно відобразити інформацію на широкому екрані. Завдяки вбудованим контролерам, можна керувати кожним із світлодіодів стрічки окремо, керуючи ними як пікселями на екрані [13].



Рисунок 2.4 – Зовнішній вид адресної світлодіодної стрічки WS1228B

ШИМ драйвер в адресній стрічці компактний і розміщується прямо в корпусі світловипромінюючого діода.

Стрічка оснащена чотирма виходами: живлення; вихід передачі; загальний контакт; вхід передачі.

Один адресний світлодіод в максимумі споживає струм в 60 міліампер. Діапазон робочих температур становить -25° – $+80^{\circ}$. Напруга живлення становить $5 \pm 0,5$ В.

ШИМ драйвери стрічки 8-бітні – для кожного кольору можливі 256 градацій яскравості. Для встановлення яскравості потрібно 3 байти інформації – по 8 біт з кожного світлодіода. Інформація передається за однолінійним протоколом з фіксованою швидкістю. Нулі та одиниці кодуються високим та низьким рівнем сигналу по лінії.

1 біт передається за 1,25 мкс. Весь пакет із 24 біт для одного світлодіода передається за 30 мкс.

Характеристики стрічки наведені у таблиці 2.2.

Таблиця 2.2

Базові характеристики стрічки WS1228B

Характеристика	Значення
Кількість світлодіодів	60 шт / метр
Розмір світлодіода	5,0 x 5, 0 мм
Напруга живлення	5 В
Потужність	14.5 Вт/метр
Яскравість	720 Лм/метр
Кут випромінювання	120 °
Ступінь захисту	IP 65
Ширина підкладки	10 мм
Робоча температура	від -40° до 60°С

Стрічка поставляється в котушках по 5 метрів, при замовленні менше котушки, ціна відрізаної стрічки, кратно метру, збільшується на 20%

2.3.3 Опис роботи LED-світлодіоду

Світлодіод має два електроди: позитивний (анод) та негативний (катод) (рисунок 2.5).

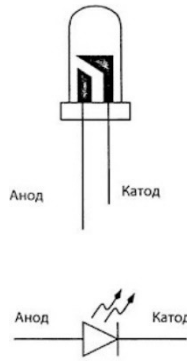


Рисунок 2.5 – Зовнішній вигляд та позначення світлодіода

Світлодіод завжди включається в схему через резистор (рис. 2.6). Основна функція резистора полягає в обмеженні струму і контролі над ним.



Рисунок 2.6 – Вигляд та схематичне зображення резисторів

Важливою характеристикою діоду, у тому числі і світловипромінювального, є струм живлення та напруга. Дані світлодіоди у середньому розраховані на максимальний струм 20 мА. Напруга варіюється в залежності від струму та матеріалу світлодіоду, що зумовлює колір світіння. Дані з характеристик світлодіодів наведені у таблиці 2.2.

Таблиця 2.2

Характеристики світлодіодів за напругами

Колір світлодіоду	Матеріал	Пряма напруга при 20 мА	
		Типове значення (В)	Діапазон (В)
ІЧ	GaAs, GaAlAs	1,2	1,1-1,6
Червоний	GaAsP, GaP, AlInGaP	2,0	1,5-2,6

Колір	Матеріал	Пряма напруга при 20 мА	
		В	В
Світлодіодний Помаранчевий	GaAsP, GaP, AlGaInP	2,0	1,7-2,8
Жовтий	GaAsP, AlInGaP, GaP	2,0	1,7-2,5
Зелений	GaP, InGaN	2,2	1,7-4,0
Блакитний	ZnSe, InGaN	3,6	3,2-4,5
Білий	Синій/УФ діод з люмінофором	3,6	2,7-4,3

Сигнальні порти Arduino Uno (цифрові та аналогові) можуть видавати до 40 мА (HIGH – для цифрових та 1024 – для аналогових) та напругу у 5 В. Безпосереднє підключення світлодіоду може призвести до виходу LED-індикаторів з ладу.

Щоб цього не відбувалось необхідно обмежити послідовний електричний ланцюг за струмом за допомогою резистору (рис. 2.7).

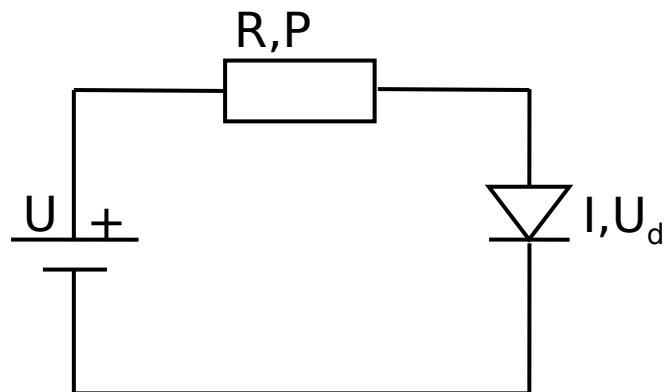


Рисунок 2.7 – Схема підключення діоду та резистору

Опір (R) та потужність теплового розсіювання (P) можна знайти за допомогою формул 2.1 та 2.2.

$$R = \frac{U - U_d}{I} \quad (2.1)$$

$$P = (U - U_d) \cdot I \quad (2.2)$$

Скористаємось формулами (2.1, 2.2) та вирахуємо опір та потужність резистора, необхідного для підключення червоного світлодіоду до Arduino:

$$R = (5-2)V/0,02A=150 \text{ Ом}; P=(5-2)V*0,02A=0,06\text{Вт}$$

Якщо підключати резистори більшого опору, то світіння світлодіоду буде більш млявим. У нашому випадку підходить резистор 220 Ом з потужністю 0,25 Вт.

Схема підключення світлодіодів до Ардуіно зображена на рисунку 2.8.

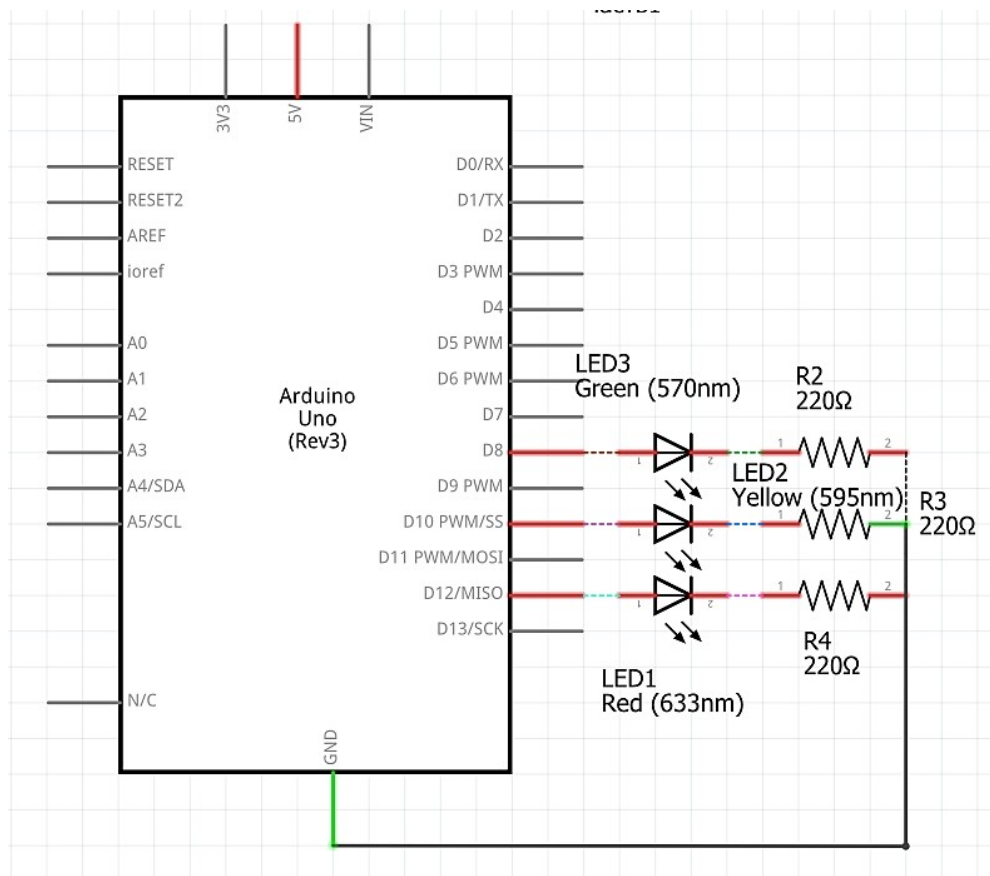


Рисунок 2.8 – Схема підключення світлодіоду до Ардуіно

2.4 Підключення натискної кнопки

В схемах, де використовуються кнопки зазвичай присутні підтягуючий

(включений між провідником і живленням – pull-up) та стягуючий (між провідником і землею – pull-down) резистори.

Підтягуючий резистор гарантує на логічному вході, з яким з'єднаний провідник, високий (у першому випадку) або низький (у другому випадку) рівень у випадках, коли провідник не з'єднаний із логічним виходом, коли приєднаний вихід знаходиться у високоомному стані та коли розімкнено ключовий елемент на приєднаному логічному виході, який налагоджено, як відкритий вивід ключа. Ключовим елементом може бути польовий чи біполярний транзистор, а відкритим виводом – колектор, емітер, стік чи витік.

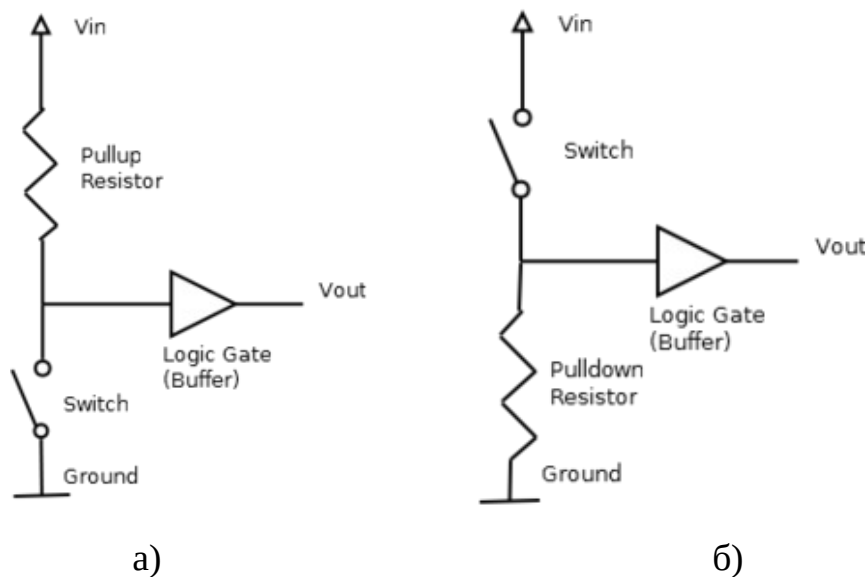


Рисунок 2.9 – Схема з підтягуючим (а) та стягуючим (б) резисторами

Ланцюг з резистором, що підтягує, можна порівняти з ділянкою напруги з двох резисторів – великого підтягуючого і дуже маленького на місці кнопки або відкритого стоку.

При натисканні на кнопку і її відпусканні виникає т.зв. "брязкіт" (bounce). Це багаторазове перемикання стану кнопки за короткий проміжок часу (порядку кількох мілісекунд), перш ніж вона прийме сталий стан. Це небажане явище виникає в момент перемикання кнопки через пружність

матеріалів кнопки або через мікроіскри, що виникають при електричному контакті.

З ефектом брязкоту контактів можна боротися або апаратними, або програмними методами.

Одним із схемотехнічних способів боротьби з брязкотом контактів є використання RS-тригера. Ця схема використовується у випадку, коли кнопка або інший механічний датчик виконані у вигляді групи контактів, що перемикаються.

RS-тригер складається з двох логічних елементів І-НЕ і має вхід установки S (від англ. Set –встановлювати) і вхід скидання R (від Reset). На обидва входи через струмообмежуючі резистори подано напругу живлення. На вході RS-тригера, який в даний момент не підключений до рухомого контакту, присутній сигнал логічної одиниці.

Якщо рухомий контакт замикає вхід землі, то ньому формується рівень логічного нуля. При натисканні та відпусканні кнопки (або при спрацьовуванні іншого механічного датчика) за допомогою рухомого контакту один, то інший вхід тригера підключається до землі.

Робота схеми залежить від числа і тривалості імпульсів, викликаних брязкотом контактів.

Ще одним способом боротьби з брязкотом контактів є використання RC-фільтрів для згладжування коливань. Згладжений сигнал потім подається на вхід тригера Шмідта або іншого логічного елемента з високоімпедансним входом.

Найпростіший програмний спосіб боротьби з брязкотом контактів – це використання затримок. Брязкіт контактів призводить до того, що на входному піні замість зміни стану з одиниці в нуль при натисканні кнопки, ми отримуємо цілу серію імпульсів. Щоб позбавитися їх паразитного впливу потрібно виявити натискання кнопки, призупинити виконання програми і реалізувати деяку затримку. Час затримки необхідно вибрати таким чином,

щоб він перевищував брязкіт контактів. Таку ж процедуру затримки потрібно реалізувати після виявлення відпускання кнопки.

Також для боротьби з брязкотом контактів для Arduino існує спеціальна бібліотека, яка називається Bounce2. Завантажити її можна з репозиторію GitHub. Ця бібліотека включає такі методи:

- ✓ `Bounce ()` – ініціалізація об'єкта `Bounce`.
- ✓ `void interval (unsigned long interval)` – встановлює час антибрязкоту в мілісекундах.
- ✓ `void attach (int pin)` – встановлює пін, до якого підключена кнопка і підключає на цьому висновку вбудований підтягуючий резистор.
- ✓ `int update()` – оскільки `Bounce` не використовує переривання Arduino, ви «оновлюєте» об'єкт до того, як зчитуєте його стан і це потрібно робити постійно (наприклад, усередині `loop`). Метод `update` оновлює об'єкт і повертає `TRUE (1)`, якщо стан піна змінилося (кнопка була натиснута або ж, навпаки, відпущена) і `FALSE (0)` в іншому випадку. Виклик методу `update` всередині `loop` необхідно робити лише один раз.
- ✓ `int read ()` – повертає оновлений стан піна.

За промовчанням бібліотека `Bounce` використовує інтервал стабілізації (`stable interval`) для реалізації антибразгату. Це простіше для розуміння і дозволяє не знати тривалість брязкоту [15].

2.5 Робота зі змінними резисторами

У проект доцільно включити змінні резистори: фоторезистор для контролю рівня освітлення та потенціометр, для використання користувачем.

2.5.1 Робота з фоторезистором

Фоторезистор – це напівпровідниковий прилад (датчик), який при опроміненні світлом змінює свій внутрішній опір.

На відміну від фотоелементів інших типів (фотодіодів та фототранзисторів) даний прилад не має р-п переходу. Це означає, що фоторезистор може проводити струм незалежно від його напрямку і може працювати не тільки в ланцюгах постійного струму, де є постійна напруга, але і зі змінними струмами.

Конструкція різних моделей фоторезисторів може відрізнятися формою матеріалу корпусу. Але в основі кожного такого приладу лежить основа, найчастіше керамічна, вкрита шаром напівпровідникового матеріалу. Поверх цього напівпровідника наносять змійкою тонкий шар золота, платини або іншого корозійностійкого металу. Шари наносяться шляхом напилення. [16].

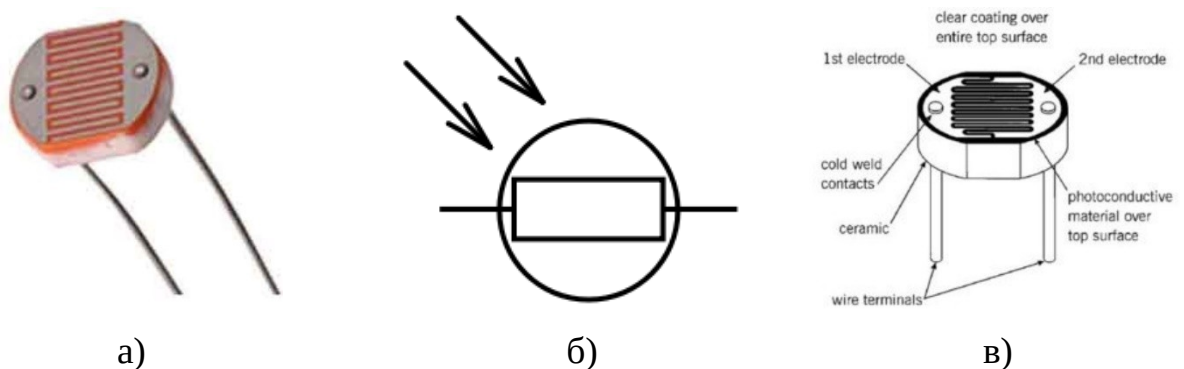


Рисунок 2.10 – Вид (а), схематичне зображення (б), та конструкція (в) фоторезистора

У неактивному стані напівпровідник виявляє властивості діелектрика. Для того щоб він проводив струм, необхідний вплив на речовину

зовнішнього стимулятора. Таким стимулятором може бути термічна дія або світлова.

Під дією фотонів світла напівпровідник насичується електронами, у результаті він стає здатним проводити електричний струм. Чим більше електронів утворюється, тим менший опір струму має напівпровідниковий матеріал.

На цьому принципі базується робота фоторезисторів. Утворенню електронів сприяє як видимий спектр світла, так і не видимий. Причому фоторезистор чутливіший до інфрачервоних променів, що мають велику енергію. Низьку чутливість до видимого світла виявляють чисті матеріали.

Фоторезистор має спектральну чутливість. Якщо довжина світлових хвиль знаходиться поза зоною провідності, то прилад перестає реагувати на такі промені. Освітленість у таких випадках вже не може впливати на струмопровідність виробу. Вибір спектральних характеристик залежить від умов експлуатації виробу та розв'язуваних завдань. Якщо інтенсивностей випромінювання замало стабільної роботи пристрою, його ефективність можна підвищити шляхом підбору чутливих елементів, з відповідним напівпровідниковим шаром.

2.5.2 Робота з потенціометром

Потенціометр (англ. potentiometer) – змінний або регульований резистор. Він відрізняється від постійного тим, що його опір можна плавно змінювати практично від нуля до певного значення. Зміна відбувається за допомогою механічного переміщення повзунка. [17].

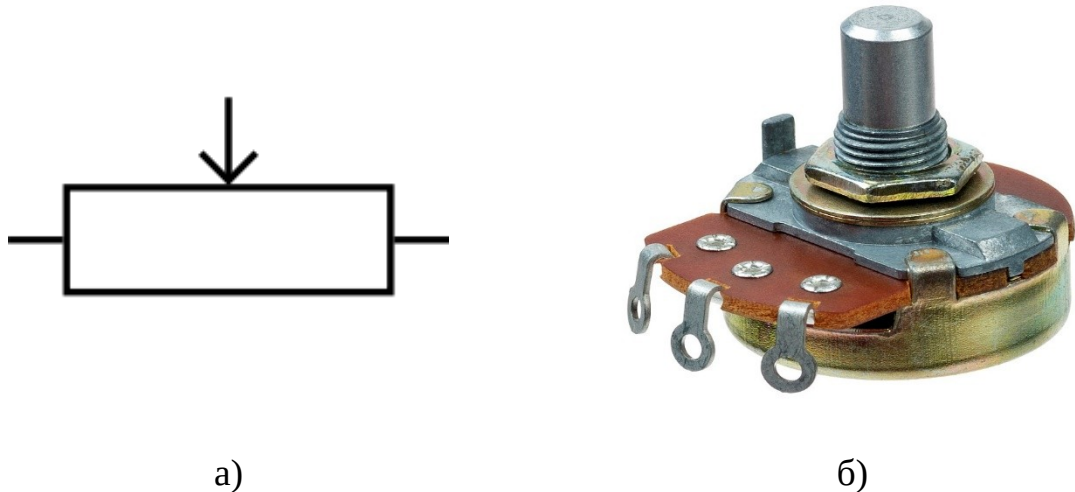


Рисунок 2.11 – Схематичне позначення а) та типовий вигляд потенціометра

Змінний резистор може бути двох типів: дротяним та плівковим. У дротяних на діелектричну трубку намотаний дріт, уздовж неї переміщується металевий пересувний контакт – повзунок. Його розташування і визначає опір елемента. Витки дроту укладені впритул один до одного, але вони розділені шаром лаку з високими діелектричними властивостями.

Існує багато різних конструкцій змінних резисторів, але їх можна розділити на дві основні групи, які відрізняються траєкторією руху ковзного контакту: потенціометри з лінійним рухом рухомої системи та потенціометри з круговим рухом рухомої системи.

Здвоєні, чотиривірні і т.д. потенціометри використовуються в аудіо техніці для зміни рівня гучності та тембру в кількох аудіо каналах одночасно (стерео, квадро і т.д.).

Співвісні потенціометри застосовуються для керування різними функціями, наприклад, гучністю та балансом в автомобільній та переносній радіоапаратурі.

Потенціометри з вимикачем зазвичай застосовують для керування гучністю та увімкненням/вимкненням за допомогою однієї ручки.

Потенціометри, з фіксацією середнього становища, використовуються регулювання стероебаланса, тембру та інших подібних функцій з одночасним отриманням тактильних відчуттів в нейтральному положенні.

Змінні резистори можуть відрізнятися розміром і мати різний діаметр валу, що дозволяє при проектуванні радіоапаратури вибирати потенціометри виходячи і з розмірів корпусу пристрою.

На відміну від постійних резисторів, у регульованих не два виводи, а як мінімум три. На електричних схемах змінні та підстроювальні резистори позначаються прямокутниками як постійні, але мають додатковий вивід, який схематично представлений як ламана лінія, що упирається в середину зображення. Щоб можна було відрізнити змінний від підстроювального, у змінного на кінці третього введення малюють стрілку, підстроювальний зображується довшою перпендикулярною лінією без стрілки.

Вибирати змінний резистор необхідно не тільки за стандартними параметрами – опору, потужності, що розсіюється, і допустимої похибки, а й за такими:

- Діапазон зміни опорів.
- Робоча температура.
- Тепловий опір. Показує, наскільки збільшується опір при нагріванні.
- Ефективний кут повороту регулятора.

2.6 Робота з інфрачервоним приймачем

Інфрачервоні промені – найбільш дешевий спосіб для дистанційного керування пристроєм у невидимому діапазоні світла. Майже всі аудіо та відео пристрої управляються ІЧ променями. У зв'язку з широким поширенням використовуваних необхідних компонентів, ІЧ управління стало

дуже дешевим, що робить його ідеальним у любителів використовувати для власних проектів [18].

Модуляція необхідна для того, щоб сигнал виділявся на тлі шуму. З модуляцією сигнал ІЧ блимає з певною частотою. ІЧ-приймач буде налаштований на цю частоту, тому він може ігнорувати решту.

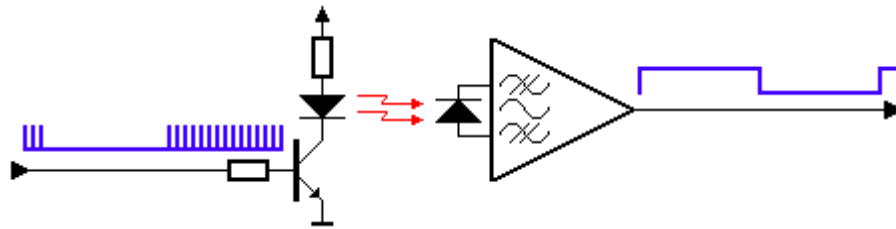


Рисунок 2.12 – Схема роботи системи

На рисунку 2.12 ліворуч передавач, що модулює сигнал за допомогою ІЧ-світлодіода. Сигнал реєструється у приймачі з іншого боку.

На рисунку 2.13 представлено типову блок-схему ІЧ-приймача. Отриманий ІЧ-сигнал із фотодіода виявлення (на лівій стороні діаграми). Цей сигнал посилюється та обмежується у перших 2-х етапах. Обмежувачем є АРП, щоб отримати постійний рівень імпульсу, незалежно від відстані до пульта. Далі з АРП сигнал надходить на смуговий фільтр (BPF). Смужний фільтр налаштований на частоту модуляції пульта. Загальний діапазон частот від 30 кГц до 60 кГц для споживчої електроніки. Наступний етап: детектор, інтегратор та компаратор. Ціль цих трьох блоків для виявлення присутності частоти модуляції. Ця частота модуляції є вихід компаратора як низький сигнал.

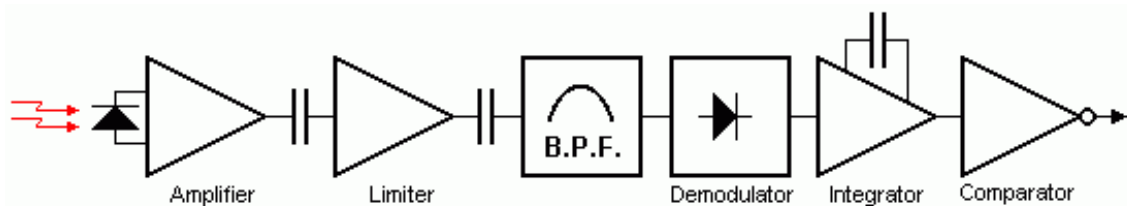


Рисунок 2.13 – Схема роботи ІЧ приймача

ІЧ приймач (IR receiver) на базі датчика 1838 призначений для прийому даних від пультів управління, що працюють в інфрачервоному діапазоні на частоті 33-43 кГц. Загальні характеристики:

- ✓ робоча напруга: 2.7 .. 5.5 В
- ✓ споживаний струм: 1.4 мА
- ✓ частота: 38 КHZ
- ✓ кут чутливості: 45 градусів
- ✓ дальність прийому: до 20 м

Приймач TL1838 VS1838B 1838 працює з більшістю пультів дистанційного керування (рисунок 2.14).

Виводи приймача інфрачервоного випромінювання TL1838 VS1838B 1838 :

- 1 - вихід;
- 2 - земля;
- 3 - плюс 5V.

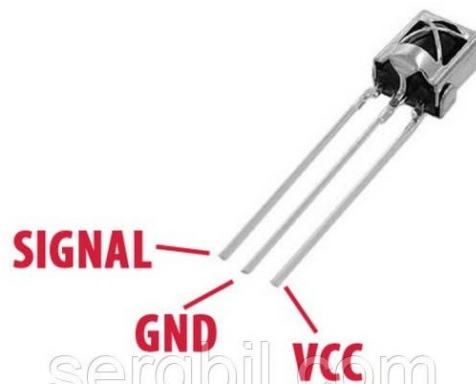


Рисунок 2.14 – Схема контактів для підключення

2.7 Робота з датчиком руху

Датчик руху (PIR-сенсор, пасивний інфрачервоний датчик чи піроелектричний сенсор) – реагує на рух. Датчик руху Ардуїно дозволяє

відстежити переміщення в закритій зоні об'єктів, що випромінюють тепло (люди, тварини). Такі системи часто застосовують у побутових умовах, наприклад, для включення освітлення у під'їзді. Малі габарити, низька вартість, простота експлуатації та відсутність складнощів у підключенні дозволяє використовувати такі датчики у системах сигналізації різного типу.

Конструкція PIR-сенсора складається з піроелектричного елемента, що відрізняється високою чутливістю (деталь циліндричної форми, в центрі якої розташований кристал) до наявності в зоні впливу певного рівня інфрачервоного випромінювання. Що більша температура об'єкта, то більше випромінювання. Зверху PIR-датчика встановлюється напівсфера, розділена на кілька ділянок (лінз), кожна з яких забезпечує фокусування випромінювання теплової енергії різні сегменти датчика руху. Найчастіше як лінза застосовують лінзу Френеля, яка за рахунок концентрації теплового випромінювання дозволяє розширити діапазон чутливості інфрачервоного датчика руху Ардуїно.

PIR-сенсор конструктивно поділено на дві половини. Це пов'язано з тим, що з пристрою сигналізації важливо саме наявність руху у зоні чутливості, а чи не сам рівень випромінювання. Тому частини встановлені таким способом, що при уловлюванні одного більшого рівня випромінювання на вихід буде подаватися сигнал зі значенням high або low.

Основними технічними характеристиками датчика руху Ардуїно є:

Зона виявлення об'єктів, що рухаються, становить від 0 до 7 метрів;

- діапазон кута стеження – 110 °;
- напруга живлення – 4.5-6;
- робочий струм – до 0,05 мА;
- температурний режим - від -20 ° до +50 ° С;
- регульований час затримки від 0,3 до 18 с.

Модуль, на якому встановлено інфрачервоний датчик руху, включає додаткову електричну обв'язку з запобіжниками, резисторами і конденсаторами. [19].

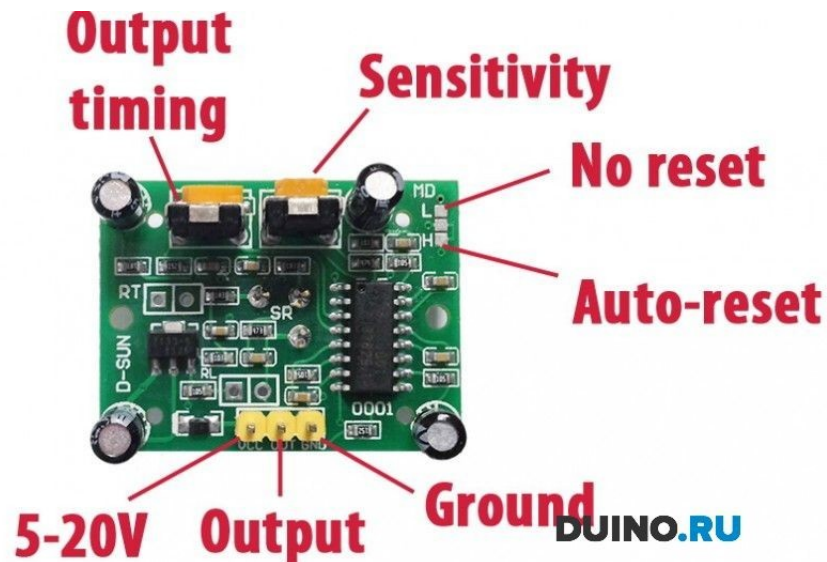


Рисунок 2.15 – Датчик руху HC-SR501

Коли пристрій встановлений у порожній кімнаті, доза випромінювання, одержувана кожним елементом, постійна, як і напруга. З появою в кімнаті людини, він насамперед потрапляє у зону огляду першого елемента, у якому з'являється позитивний електричний імпульс. Коли людина переміщається по кімнаті, разом із нею переміщається і теплове випромінювання, що потрапляє вже другого сенсор. Цей PIR-елемент генерує негативний імпульс. Різноспрямовані імпульси реєструються електронною схемою датчика, що робить висновок, що у полі зору PIR-sensor Arduino знаходиться людина.

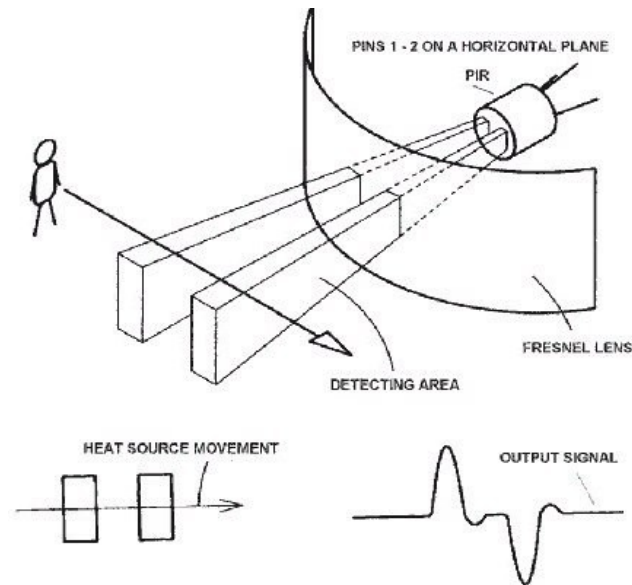


Рисунок 2.16 – Принцип роботи датчика

Для надійного захисту від зовнішніх шумів, перепадів температури та вологості елементи датчика встановлюються в герметичний металевий корпус. На верхній частині корпусу по центру знаходиться прямокутник, виконаний з матеріалу, що пропускає інфрачервоне випромінювання (найчастіше на основі силікону). Чутливі елементи встановлюються за пластиною.

2.8 Вибір програмного інструментарію

Не менш важливим питанням щодо використання програмних засобів для проектування системи, написання відповідного набору команд та прошивання платформи. Тому розглянемо основні програмні компоненти нижче.

2.8.1 Вибір середовища розробки

Так як для даного проекту застосовується платформа Ардуіно то доречним буде використання відомого засобу – Arduino IDE.

Абревіатура IDE розшифровується як Integrated Development Environment, у перекладі – інтегроване середовище розробки.

Arduino IDE дозволяє складати програми за допомогою зручного текстового редактора, компілювати їх у машинний код, і завантажувати на всі версії плати Arduino. [20].

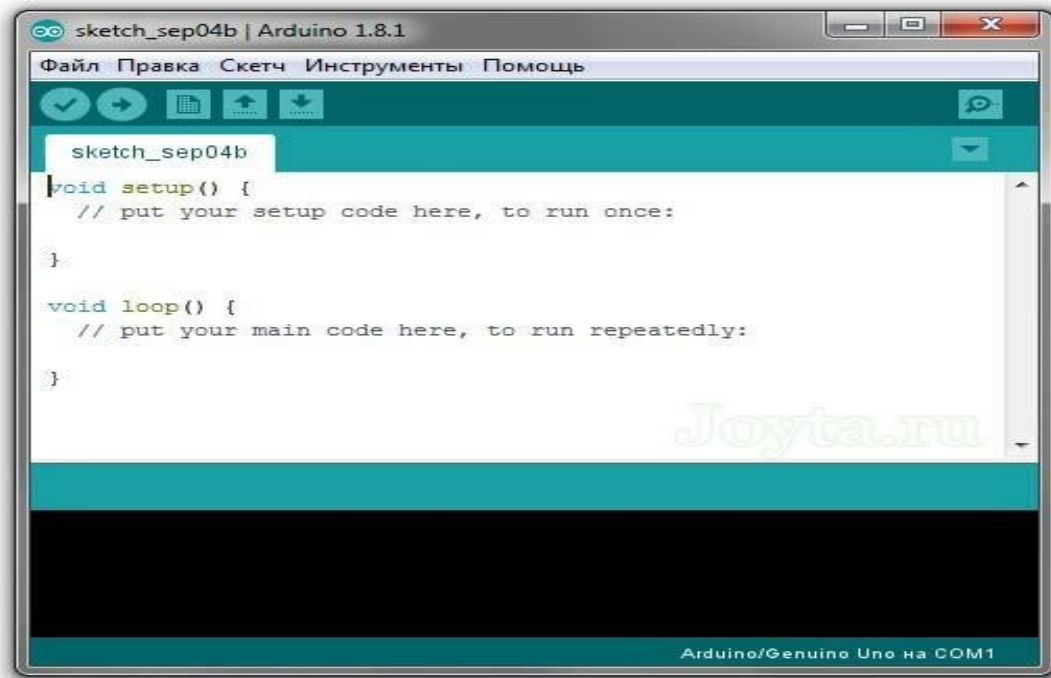


Рисунок 2.17 – Інтерфейс Arduino IDE

Ардуїно IDE, як середовище програмування, призначене для непрофесійного використання у роботобудуванні та створення найпростіших автоматичних систем управління. Спектр завдань, які буде здатний виконувати готовий пристрій, визначається набором плат, що входять до його складу.

Для роботи з середовищем необхідно підключити зібраний виріб до комп'ютера за допомогою кабелю USB. Далі в диспетчері пристроїв необхідно перевірити, до якого порту підключений мікроконтролер. Якщо він не відображається або написано, що пристрій не розпізнаний – значить неправильно встановлений драйвер або плата потребує діагностики. Наступним кроком буде запуск мови програмування Arduino IDE. У меню

потрібно вибрати вкладку інструменти. При її натисканні відкриється список, у якому необхідно вибрати пункт порт. Там потрібно вибрати порт, вказаний у диспетчері пристроїв. Кінцевим пунктом є вибір плати, яку ми використовуватимемо для завантаження скетчів.

2.8.2 Вибір мови програмування

Основою для Ардуїно є мова C/C++, проте для роботи з мікроконтролерами використовуються й інші, наприклад Processing – невелика субмова програмування, заснована частково на Java (концепція) та частково на C++ (синтаксис, зручність, інше, інше). На Processing можна легко та швидко створити мультимедійний додаток для різних платформ. Програма Arduino називається скетч.

2.8.3 Вибір середовища проектування та моделювання

Для проектування та моделювання розробленої системи необхідно обрати САПР, що задовільнить вимогам проекту. Для цього можна скористатись системою Proteus Design Suite 8.x від компанії LabCenter Electronics.

Вона працює за стандартами PSpice, на основі яких функціонує логіка всіх персональних комп'ютерів сучасності, а також програмованих пристроїв:

- мікроконтролери;
- мікропроцесори;
- цифрові сигнальні процесори;

- та інше.

Інтерфейс Proteus дуже наочний (рис. 2.21). Всі, хто на практиці знайомий із побудовою мікросхем, досить швидко освоїться тут. До складу комплексу також входить довідкова інформація, де ви знайдете відповіді на свої запитання, а також ознайомитеся зі специфікаціями.

Є бібліотека демонстраційних проектів. Там реалізовані типові ситуації, з прикладу яких зможете навчатися і навіть реалізувати власні розробки. Це заощадить час і дозволить швидше втягнутися у процес.

Передбачено і редактора для написання програми для контролерів. У результаті в одному середовищі ви зможете спроектувати схему, подивитися, як вона виглядатиме в готовому вигляді, і написати програмне забезпечення для її правильного функціонування [22].

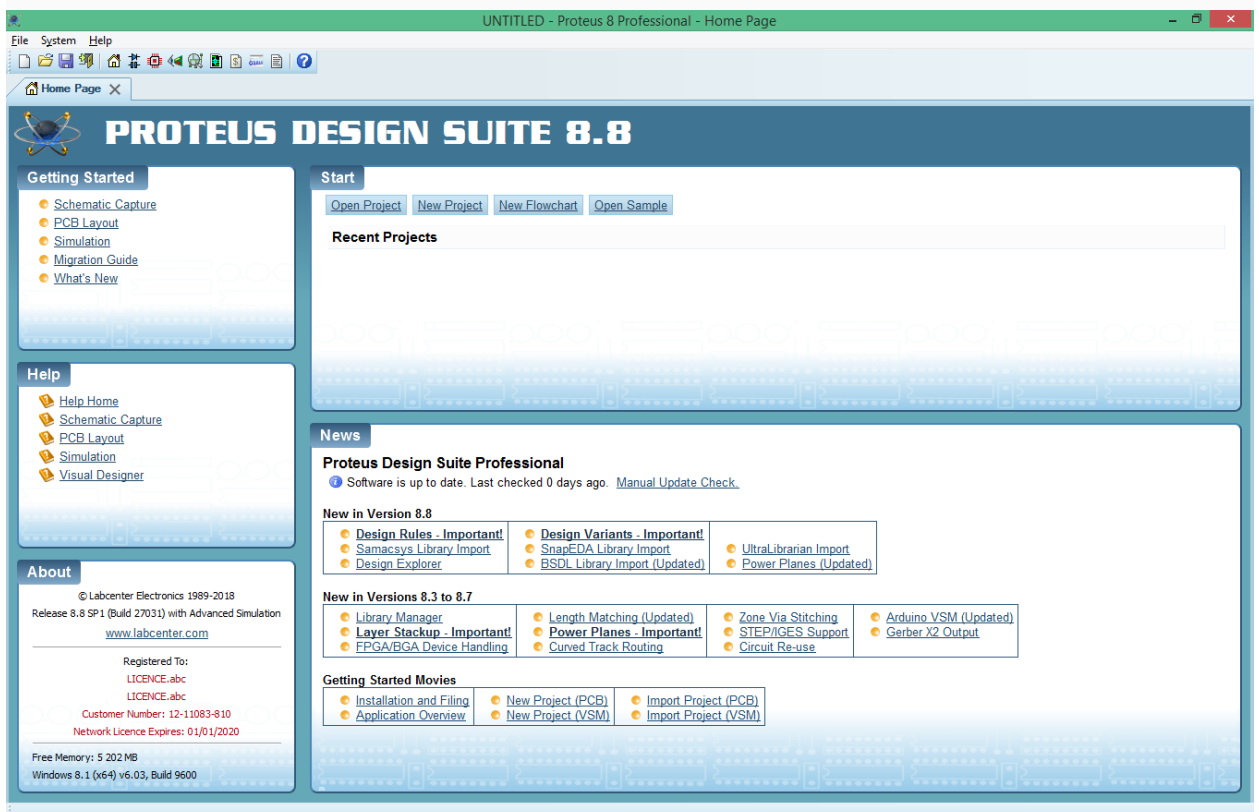


Рисунок 2.18 – Стартова сторінка САПР Proteus 8.x

Є підтримка плагінів і компіляторів, що підключаються. Це значно розширить функціонал та сфери застосування.

2.9 Висновки за другим розділом

У відповідності до структурної схеми для програмно-апаратного комплексу, було підібрано такі основні компоненти:

1. контролер Arduino UNO R3 на базі мікроконтролеру ATmega328P;
2. світлодіодна стрічка WS1228B;
3. ІЧ приймач приймач TL1838 VS1838B 1838;
5. датчик руху HC-SR501

Платформа Arduino Uno, має всі можливості для реалізації поставлених в роботі завдань: зчитування показів датчиків та керування виходами, до яких через схеми узгодження підключається стрічка. Алгоритм вимірювання передбачає можливість програмного задіяння таких компонентів MCU і доповнень в схемотехніці платформи:

- ЦАП;
- блок обробки переривань;
- цифрові входи/виходи;
- аналогові входи.

Моделювання роботи системи велося за допомогою програми Proteus Design Suite 8.13

У якості середовища розробки програмного забезпечення було вирішено використовувати Arduino IDE, мова програмування яка використовується в Arduino IDE – Processing.

РОЗДІЛ 3

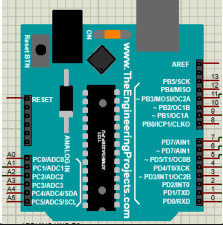


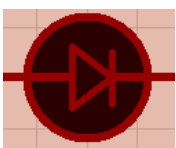
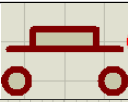
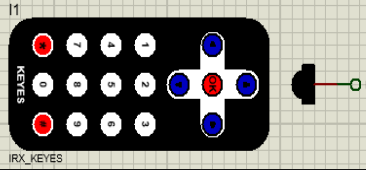
РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

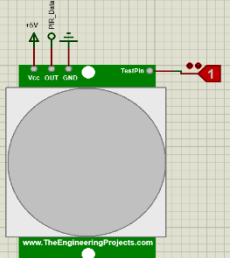
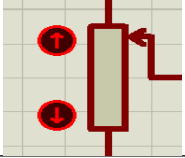
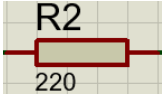



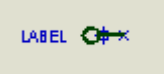
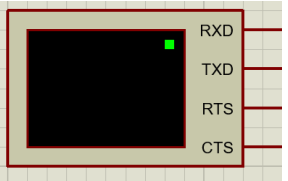
3.1 Проектування апаратного комплексу системи

Для проектування скористаємось компонентами САПР Proteus, що наведені у таблиці 3.1.

Таблиця 3.1

Перелік необхідних компонентів проекту

№ з/п	Тип деталі	Розташування	Підключення	Вигляд
1	Arduino UNO R3	DEVICES/ Arduino UNO R3	Головний модуль проекту	
2	Фото-резистор	DEVICES/TORH LDR	До Порту A1	
3	RGB Стрічка	DEVICES/ WS2812x8	До Порту D11	
4	LEDS: RED GREEN BLUE	DEVICES/ LED-RED LED-GREEN LED-BLUE	До Порту D4 До Порту D5 До Порту D6	
5	Кнопка механічна	DEVICES/ BUTTON	До Порту D2	
6	ІК-приймач та пульт	DEVICES/ IRX_KEYES	До Порту D9	

№ з/п	Тип деталі	Розташування	Підключення	Вигляд
7	ІК-Датчик руху	DEVICES/PIR Sensor2	До Порту D7	
8	Потенціометр	DEVICES/POT	До Порту A2	
9	Резистор	DEVICES/RES	Опір у електричних ланцюгах	
10	Заземлення	TERMINALS/GROUND	Для заземлення у електричних ланцюгах	
11	Живлення	TERMINALS/POWER	Для живлення електричних ланцюгів	
12	Логічний сигнал	DEVICES/LOGICSTATE	Для отримання логічного сигналу	
13	З'єднувач	TERMINALS/DEFAULT	Для розриву у електричних схемах	
14	Термінал	INSTRUMENTS/VIRTUAL TERMINAL	Для моделювання роботи послідовного порту	

В додатку А зображена схема проекту, що побудована в середовищі САПР Proteus. Центральним вузлом даної схеми є прототип плати ARDUINO UNO R3, на якому присутні виводи цифрових та аналогових портів (ARD1). Живлення та заземлення у цій моделі плати не передбачено. Дана модель

приладу не входить до офіційного ПЗ системи, тому її необхідно скачати та встановити до бібліотек з сайту консорціуму розробників [23]. При визові контекстного меню даного компонента можна завантажити до нього прошивку з кодом проекту у бінарному файлі (експорт бінарного файлу).

Світлодіоди (червоний – D1, зелений – D2 та синій – D3) підключені до цифрових портів 4, 5, 6 відповідно. До катодів кожного зі світлодіодів підключено резистор номіналом 220 Ом та заземлення.

Натискна кнопка (BTN1) підключена до цифрового порту 2 зі стягуючим резистором R5 номіналом 10 кОм. На інший контакт кнопки подається живлення +5V.

Змінний резистор – потенціометр RV1 підключено до аналогового порту A1. Інші його контакти під'єднано до живлення (+5V) та заземлення. Фоторезистор підключено до порту A2 з підтягуючим резистором R4 на 10 кОм.

Віртуальний термінал відповідно підключено до портів 0 та 1 плати для моделювання роботи апаратного серійного порту.

Інфрачервоний датчик руху представлено у проекті як PIR1. Це також пристрій, бібліотеку якого необхідно завантажити з того ж ресурсу що і Ардуіно. Він має чотири входи:

- TestPin – підключаємо логічний сигнал для імітації роботи датчику;
- Vcc – до живлення (+5V);
- OUT – підключаємо до порту 7 через з'єднувач PIR_Data;
- GND – до заземлення.

Модель приймача з пультом дистанційного керування I1 підключаємо до порту 11 через з'єднувач Remote.

Модель світлодіодної стрічки, що складається із трьох блоків по вісім світлодіодів під'єднано до порту 9.

Для достовірності роботи системи та правильності підключення пристроїв схеми можна скористатись програмою, яка наведена у лістингу:

```

#include <Adafruit_NeoPixel.h>
#include <IRremote.h> //бібліотека для роботи з ІЧ
#приймачем
#define LEDRED 4
#define LEDGREEN 5
#define LEDBLUE 6
#define buttonPin 2
#define POT A1
#define PHOTO A2
#define PIR_Data 7
#define PIN 9
#define Remote 11
IRrecv irrecv(Remote); //об'єкт класу
Adafruit_NeoPixel strip (numPixels, PIN, NEO_GRB +
NEO_KHZ800);
decode_results results;
uint8_t numPixels = 24;
int ledSwitch=0;
void setup() {
  pinMode(LEDRED, OUTPUT);
  pinMode(LEDGREEN, OUTPUT);
  pinMode(LEDBLUE, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(POT, INPUT);
  pinMode(PHOTO, INPUT);
  pinMode(PIR_Data, INPUT);
  irrecv.enableIRIn(); //Активація інфрачервоного
  //приймача

  Serial.begin(9600);
  Serial.println("Test");
  strip.begin();
  strip.show();
  uint8_t brightness=200;
  strip.setBrightness(brightness);
  for (int i=0;i<numPixels;i++)
    strip.setPixelColor(i, strip.Color(rand()
%brightness, rand()%brightness, rand()%brightness));
  strip.show();
  digitalWrite(LEDRED, HIGH);
  digitalWrite(LEDGREEN, HIGH);
  digitalWrite(LEDBLUE, HIGH);
  delay(100);
  digitalWrite(LEDRED, LOW);
  digitalWrite(LEDGREEN, LOW);
  digitalWrite(LEDBLUE, LOW);
}

```

```

void loop() {
  if (digitalRead(buttonPin))
  {
    ledSwitch++;
    Serial.print("ButtonPressed  ");
    Serial.println(ledSwitch);
    Serial.print(analogRead(POT));
    Serial.print(' ');
    Serial.print(analogRead(PHOTO));
    Serial.print(' ');
    Serial.print(digitalRead(PIR_Data));
    Serial.println(' ');
    delay(200);
  }
  if (irrecv.decode(&results))
  {
    Serial.print("0x");
    Serial.println(results.value, HEX);
    irrecv.resume();
  }
}

```

Дана програма розрахована на перевірку портів, світлодіодів, датчиків та кнопки. Даний код необхідно транслювати до бінарного файлу а потім завантажити файл з розширенням .hex до моделі Program File:(рис. 3.2).

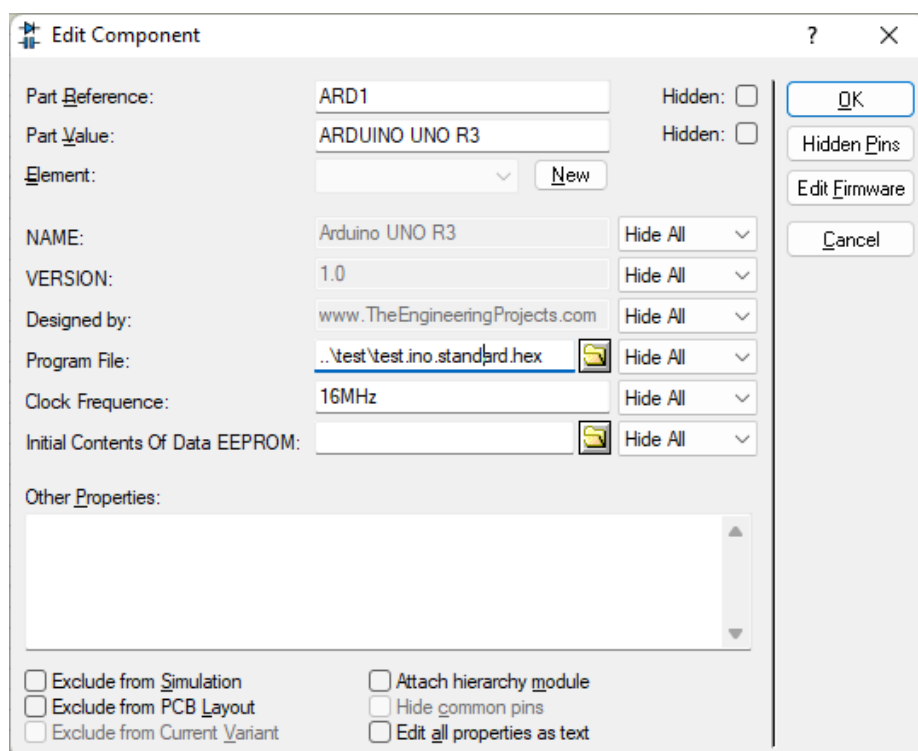


Рисунок 3.2 – Діалогове вікно налаштування Ардуіно

Потім за допомогою вбудованих засобів моделювання (Run the Simulation) можна перевірити вірність отриманих результатів програмування та проектування (додаток Б).

Як ми бачимо на віртуальному терміналі відображається службова інформація. З проектом у режимі симуляції можна взаємодіяти (натискати кнопки пульта, механічні кнопки, змінювати значення датчиків та сенсорів). При цьому інформація відображається на екрані. А саме кількість натисків, значення зі змінних резисторів та код кнопки дистанційного керування.

3.2 Програмування комплексу системи

3.2.1 Використання зовнішніх бібліотек

До більшості портів ми будемо безпосередньо звертатись у програмі за допомогою методів зчитування/запису, однак для управління складними сигналами інфрачервоного приймача сигналів та контролера світлодіодної стрічки необхідно використовувати зовнішні бібліотеки, що можуть просто та зрозуміло керувати процесом роботи даних приладів. Екосистема Arduino заснована на Open Source проектах, тому навіть для даних систем можна зустріти багато бібліотек, отже їхній вибір повинен бути чітким та доцільним.

Для управління інфрачервоним приймачем ми будемо використовувати бібліотеку IRremote (<IRremote.h>) версії 3.6.1, з офіційною інформацією якої можна ознайомитись на платформі GitHub [24].

Дана бібліотека признана багатьма розробниками як одна з повних, стабільних, з великою кількістю прикладів використання. Згідно до визначення самих розробників проекту, це бібліотека для інфрачервоного дистанційного керування для Arduino: надсилає та приймає інфрачервоні сигнали за допомогою кількох протоколів. Вона підтримує ООП парадигму


```

uint32_t decodedRawDataArray[RAW_DATA_ARRAY_SIZE];
// 32 bit decoded raw data,
to
//be used for send function.
irparams_struct *rawDataPtr; // Pointer of the raw
timing
//data to be decoded.
//Mainly the data buffer
//filled by receiving ISR.
};

```

Для роботи зі світлодіодною стрічкою було прийнято рішення скористатись бібліотекою NeoPixel від відомого розробника Adafruit (<Adafruit_NeoPixel.h>). З офіційною інформацією даної бібліотеки можна ознайомитись на платформі GitHub [25].

Бібліотека Arduino для керування однопровідними світлодіодними пікселями та стрічками, такими як цифрова світлодіодна стрічка Adafruit 60 LED/meter, Adafruit FLORA RGB Smart Pixel, Adafruit Breadboard RGB Smart Pixel, Adafruit NeoPixel Stick і Adafruit NeoPixel.

Це також відома бібліотека що має зручний інструментарій взаємодії. По-перше конструктор класу Adafruit_NeoPixel має декілько перевантажень, але доречно скористатись наступним:

```

Adafruit_NeoPixel::Adafruit_NeoPixel(uint16_t n,
int16_t p, neoPixelType t)
: begun(false), brightness(0), pixels(NULL),
endTime(0) {
updateType(t);
updateLength(n);
setPin(p);
#ifdef ARDUINO_ARCH_RP2040
// Find a free SM on one of the PIO's
sm = pio_claim_unused_sm(pio, false); // don't panic
// Try pio1 if SM not found
if (sm < 0) {
pio = pio1;
sm = pio_claim_unused_sm(pio, true); // panic if no
SM is free
}
init = true;
#endif
}

```

Він має три параметри, де n – кількість світлодіодів, p – порт підключення, t – тип стрічки.

Також у проекті необхідно скористатись основними методами даної бібліотеки:

```
void Adafruit_NeoPixel::begin(void) {
  if (pin >= 0) {
    pinMode(pin, OUTPUT);
    digitalWrite(pin, LOW);
  }
  begun = true;
}
```

`show()` – для відображення освітлення стрічки;

`setBrightness()` – для управління яскравістю

`setPixelColor()` – для встановлення кольору світлодіоду.

3.2.2 Зчитування кнопок пульту дистанційного керування

Задля правильного керування пристроєм необхідно зчитати коди пульту дистанційного керування, зображеного на рисунку 3.4.



Рисунок 3.4 – Пристрій дистанційного керування

Для цього скористуємось прикладом керування пульту дистанційного керування з бібліотеки IRremote:

```
#include "PinDefinitionsAndMore.h"
#include <IRremote.hpp>

void setup() {
    Serial.begin(115200);
    // Just to know which program is running on my Arduino
    Serial.println(F("START " __FILE__ " from "
__DATE__ "\r\nUsing library version "
VERSION_IRREMOTE));

    // Start the receiver and if not 3. parameter
    //specified, take LED_BUILTIN pin from the internal
    //boards definition as default feedback LED
    IrReceiver.begin(IR_RECEIVE_PIN,
ENABLE_LED_FEEDBACK);

    Serial.print(F("Ready to receive IR signals of
protocols: "));
    printActiveIRProtocols(&Serial);
    Serial.print(F("at pin "));
    Serial.println(IR_RECEIVE_PIN);
}

void loop() {
    /*
    * Check if received data is available and if yes,
    * try to decode it.
    * Decoded result is in the
    * IrReceiver.decodedIRData structure.
    * E.g. command is in
    * IrReceiver.decodedIRData.command
    * address is in command is in
    * IrReceiver.decodedIRData.address
    * and up to 32 bit raw data in
    * IrReceiver.decodedIRData.decodedRawData
    */
    if (IrReceiver.decode()) {

        // Print a short summary of received data
        //IrReceiver.printIRResultShort(&Serial);
        if (IrReceiver.decodedIRData.protocol == UNKNOWN)
        {
            //We have an unknown protocol here, print more info
            IrReceiver.printIRResultRawFormatted(&Serial, true);
        }
    }
}
```

```

Serial.println();

/*
 * !!!Important!!! Enable receiving of the next
 * value,
 * since receiving has stopped after the end of
 * the current received data packet.
 */
IrReceiver.resume(); // Enable receiving of the
next value

/*
 * Finally, check the received data and perform
 * actions according to the received command
 */
if (IrReceiver.decodedIRData.command == 0x10) {
    // do something
} else if (IrReceiver.decodedIRData.command ==
0x11) {
    // do something else
}
}
}
}

```

Після виконання даного коду з пультом дистанційного керування було отримано значення кодів у шістнадцятиричному форматі, що представлено у таблиці 3.2.

Таблиця 3.2

Перелік кодів зчитування дистанційного пульта керування

Const	Значення	Const	Значення
B_BR_UP	F700FF	B_BR_DOWN	F7807F
B_COL_R	F720DF	B_COL_G	F7A05F
B_COL1	F710EF	B_COL2	F7906F
B_COL4	F730CF	B_COL5	F7B04F
B_COL7	F708F7	B_COL8	F78877
B_COL10	F728D7	B_COL11	F7A857
B_OFF	F740BF	B_ON	F7C03F
B_COL_B	F7609F	B_COL_W	F7E01F

B_COL3	F750AF	B_REG1	F7D02F
B_COL6	F7708F	B_REG2	F7F00F
B_COL9	F748B7	B_REG3	F7C837
B_COL12	F76897	B_REG4	F7E817

Дані коди будуть використані у подальшому програмуванні

ВИСНОВКИ

Електричні параметри та елементи гнучкої світлодіодної стрічки RGB дозволяють створити систему керування що може відпрацьовувати довільні алгоритми управління світловими ефектами.

В роботі розроблено проект такої системи, що включає в себе блок мікроконтролера, систему датчиків та сенсорів, блок взаємодії з користувачем, систему освітлення та живлення.

Основними компонентами системи керування стрічкою є:

1. контролер Arduino UNO R3 на базі мікроконтролеру ATmega328P;
3. ІЧ приймач TL1838 VS1838B 1838;
5. датчик руху HC-SR501

Алгоритм роботи системи використовує такі блоки MCU і доповнень в схемотехніці платформи:

- ЦАП;
- блок обробки переривань;
- цифрові входи/виходи;
- аналогові входи.

Моделювання роботи системи проведено за допомогою програми Proteus Design Suite 8.13. У якості середовища розробки програмного забезпечення використано Arduino IDE з мовою програмування Processing.

ПЕРЕЛІК ПОСИЛАНЬ

1. Освітлення виробничих приміщень. Студентська бібліотека [Електронний ресурс]. Режим доступу [www. URL = https://buklib.net/books/35234/](http://www.buklib.net/books/35234/) - 10.11.22
2. Типи та види світлодіодних стрічок. Інтернет-магазин освітлення [Електронний ресурс]. Режим доступу [www. URL = https://5watt.ua/uk/blog/statti/tipi-ta-vidi-svitlodiodnikh-strichok](http://www.5watt.ua/uk/blog/statti/tipi-ta-vidi-svitlodiodnikh-strichok) - 10.11.22
3. Колірна модель. Вікіпедія – вільна енциклопедія [Електронний ресурс]. Режим доступу [www. URL = https://uk.wikipedia.org/wiki/Колірна_модель](http://www.wikipedia.org/wiki/Колірна_модель)
4. RGB. Вікіпедія – вільна енциклопедія [Електронний ресурс]. Режим доступу [www. URL = https://uk.wikipedia.org/wiki/RGB](http://www.wikipedia.org/wiki/RGB) - 15.11.22
5. СМΥК. Вікіпедія – вільна енциклопедія [Електронний ресурс]. Режим доступу [www. URL = https://uk.wikipedia.org/wiki/СМΥК](http://www.wikipedia.org/wiki/СМΥК) - 15.11.22
6. HSV. Вікіпедія – вільна енциклопедія [Електронний ресурс]. Режим доступу [www. URL = https://uk.wikipedia.org/wiki/HSV](http://www.wikipedia.org/wiki/HSV) - 15.11.22
7. LAB. Вікіпедія – вільна енциклопедія [Електронний ресурс]. Режим доступу [www. URL = https://uk.wikipedia.org/wiki/Lab](http://www.wikipedia.org/wiki/Lab) - 15.11.22
8. Світлодіодна стрічка Solar 3528 60 Led комплект RGB (5945/T). Інтернет магазин [Електронний ресурс]. Режим доступу [www. URL = https://epicentrk.ua/ua/shop/mplc-svitlodiodna-strichka-solar-rgb-3528-10-m-z-kontrolerom-ta-pul-tom-distantiinogo-keruvannya-bilii-08180unk-1ec80f2e-10fe-6d92-a669-3d1d199326f7.html](http://www.epicentrk.ua/ua/shop/mplc-svitlodiodna-strichka-solar-rgb-3528-10-m-z-kontrolerom-ta-pul-tom-distantiinogo-keruvannya-bilii-08180unk-1ec80f2e-10fe-6d92-a669-3d1d199326f7.html) - 17.11.22
9. Основи мікропроцесорної техніки: лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 171 «Електроніка» / В.С.Баран, Г.Г.Власюк, Ю.О.Оникієнко, О.І.Смоленська ; КПІ ім. Ігоря Сікорського. – Електронні текстові

- данні (1 файл: 3,42 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 140 с.
10. Переривання. Словник з інформатики [Електронний ресурс]. Режим доступу [www. URL = it.slovnik.ukr/index.php/Переривання](http://www.it.slovnik.ukr/index.php/Переривання) - 20.11.22
 11. Глухов О.В., Кравчук О.О., Левченко Є.В. Вивчення властивостей мікроконтролерів і електронних систем на базі платформи Ардуіно: навч. посібник для студентів ВНЗ. Харків: ХНУРЕ, 2019. – 192 с. ISBN 978-966-659-273-9
 12. WS2812B Intelligent control LED integrated light source. Магазин з продажу комп'ютерної техніки [Електронний ресурс]. Режим доступу [www. URL = https://greenchip.com.ua/23-0-444-2.html](https://greenchip.com.ua/23-0-444-2.html) - 22.11.22
 13. АДРЕСНА СВІТЛОДІОДНА СТРИЧКА WS2812B 60 IP65 1МІнтернет магазин Схема [Електронний ресурс]. Режим доступу [www. URL https://sxema.com.ua/ua/p1277819438-adresnaya-svetodiodnaya-lenta.htmlhttps://sxema.com.ua/ua/p1277819438-adresnaya-svetodiodnaya-lenta.html](https://sxema.com.ua/ua/p1277819438-adresnaya-svetodiodnaya-lenta.html) - 30.11.22
 14. Жеребцов О.А. Програмування мікроконтролерів: Технологічна карта з проведення лабораторної роботи за темою «Програмування LED – індикаторів за допомогою мікроконтролеру». Запоріжжя. ЗІЕІТ. 2022. 15с.
 15. Jeremy Blum. Exploring Arduino: Tools and Techniques for Engineering Wizardry [Електронний ресурс]. Режим доступу [www. URL = https://www.pdfdrive.com/exploring-arduino-tools-and-techniques-for-engineering-wizardry-d181896366.html](https://www.pdfdrive.com/exploring-arduino-tools-and-techniques-for-engineering-wizardry-d181896366.html) - 402 pp. – 22.12.22
 16. Фоторезистор – Вікіпедія. [Електронний ресурс]. Портал вільної енциклопедії // Режим доступу: <https://uk.wikipedia.org/wiki/Фоторезистор> – 26.12.22

17. Потенціометр – Вікіпедія. [Електронний ресурс]. Портал вільної енциклопедії // Режим доступу: <https://uk.wikipedia.org/wiki/Потенціометр> – 26.12.22
18. Інфрачервоне_випромінювання Вікіпедія – вільна енциклопедія [Електронний ресурс]. Режим доступу [www. URL = https://uk.wikipedia.org/wiki/Інфрачервоне_випромінювання](https://uk.wikipedia.org/wiki/Інфрачервоне_випромінювання) - 02.01.23
19. PIR датчик руху Arduino HC-SR501. Wiki ТНТУ 3.0 - відкритті знання та навчання [Електронний ресурс]. Режим доступу [www. URL = https://wiki.tntu.edu.ua/PIR_датчик_руху_Arduino_HC-SR501](https://wiki.tntu.edu.ua/PIR_датчик_руху_Arduino_HC-SR501) - 05.01.23
20. Arduino Software [Електронний ресурс]. Режим доступу [www. URL = https://www.arduino.cc/en/software](https://www.arduino.cc/en/software) - 06.01.23
21. Методичні вказівки до лабораторних робіт з навчальної дисципліни "Електроніка та мікропроцесорні системи" для здобувачів вищої освіти першого (бакалаврського) рівня за спеціальністю 015.10 "Професійна освіта. Комп'ютерні технології" денної та заочної форм навчання /Василець С.В., Василець К.С. - Рівне: НУВГП, 2019 - 134 с.
22. Proteus Tutorials. Офіційний портал розробників [Електронний ресурс]. Режим доступу [www. URL = https://www.labcenter.com/tutorials/](https://www.labcenter.com/tutorials/) - 08.01.23
23. THE ENGINEERING PROJECTS A platform for engineers & technical professionals to share their engineering projects, solutions & experiences with TEP Community & support open source [Електронний ресурс]. Режим доступу [www. URL = https://www.theengineeringprojects.com/2015/12/arduino-library-proteussimulation.html](https://www.theengineeringprojects.com/2015/12/arduino-library-proteussimulation.html) - 10.01.23
24. Arduino-IRremote. GitHub. Developer Platform [Електронний ресурс]. Режим доступу [www. URL = https://github.com/Arduino-IRremote/Arduino-IRremote](https://github.com/Arduino-IRremote/Arduino-IRremote) - 10.01.23

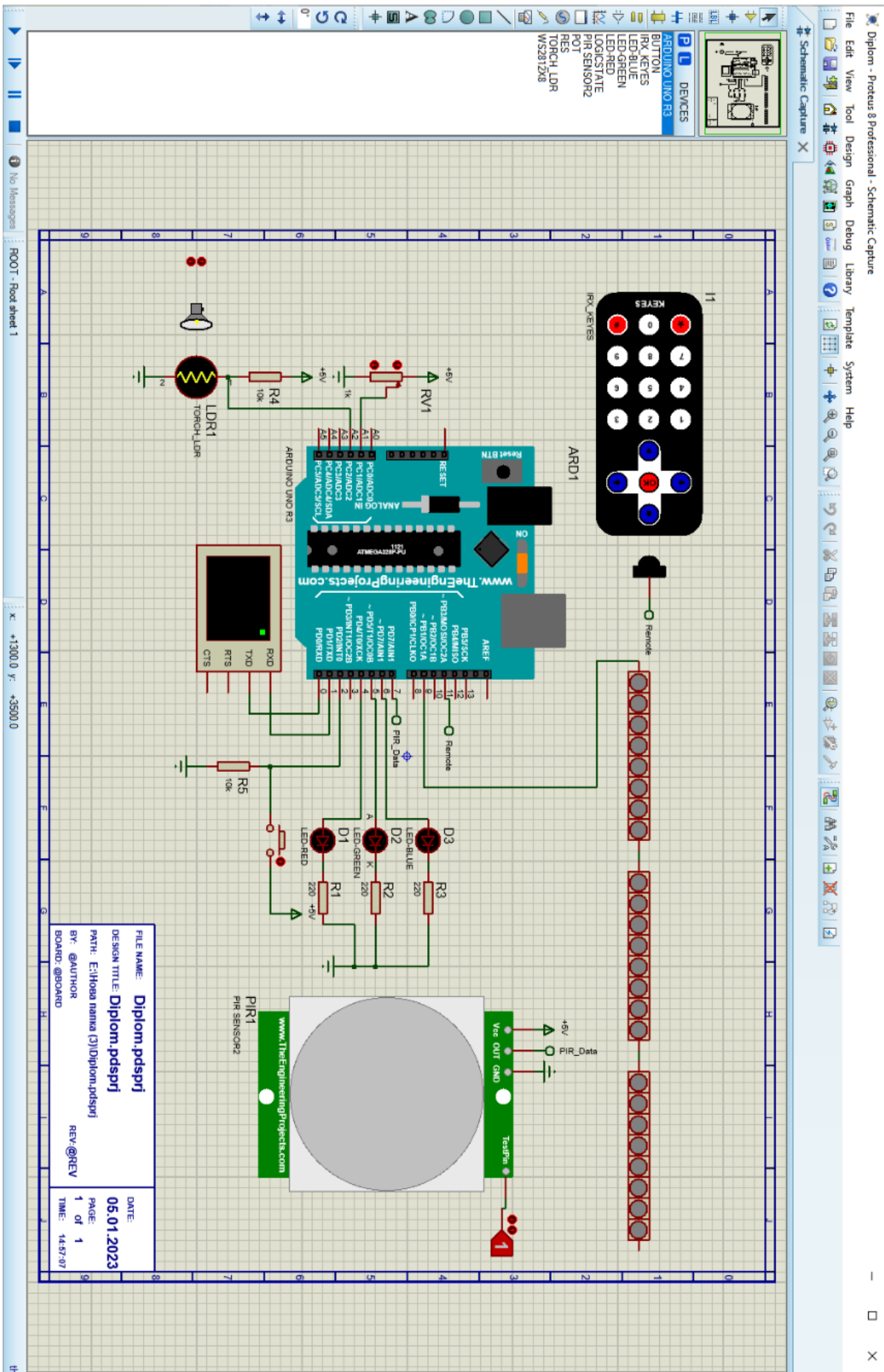
25. Adafruit_NeoPixel. GitHub. Developer Platform [Электронный ресурс].

Режим доступа www. URL =

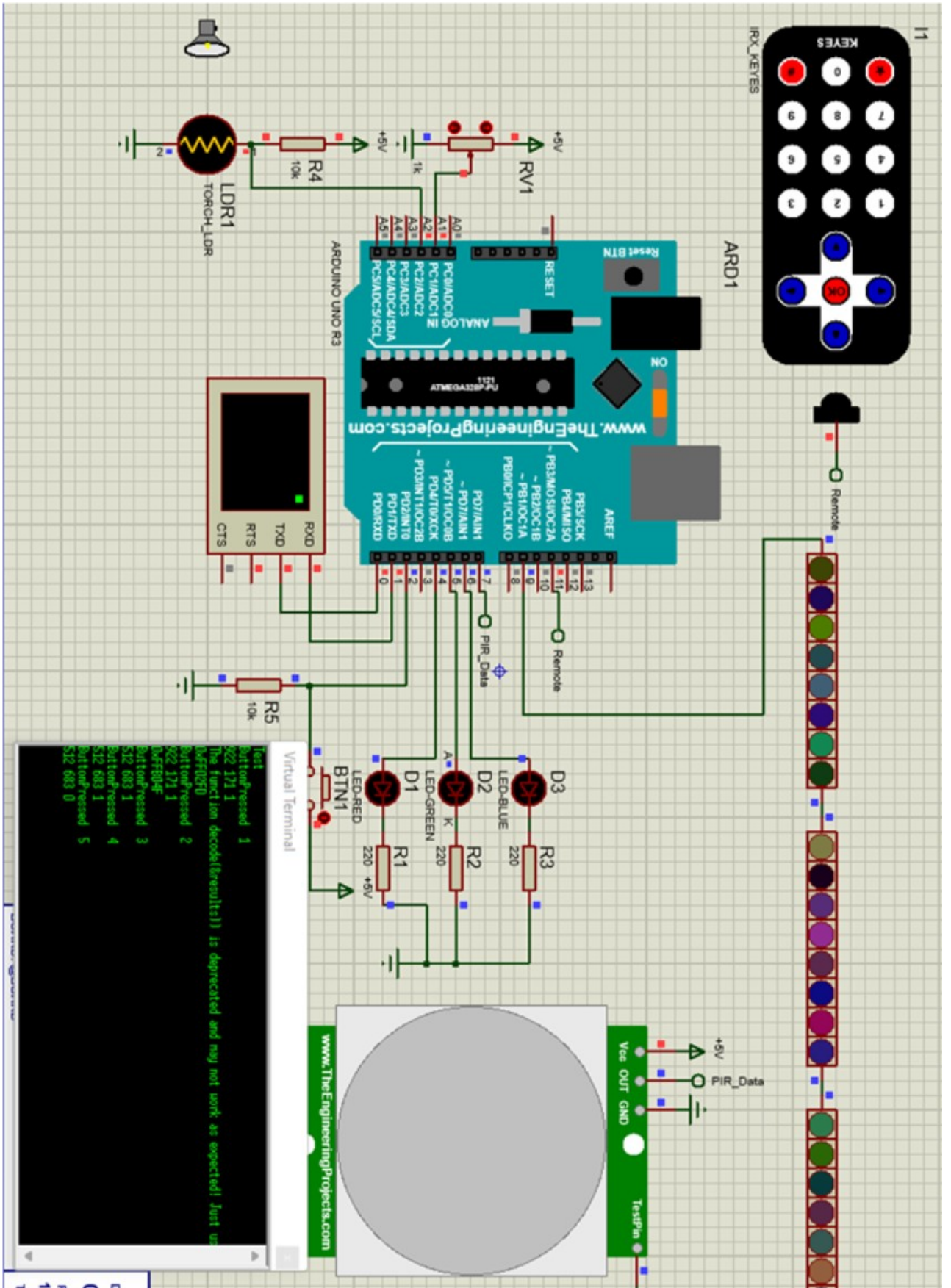
https://github.com/adafruit/Adafruit_NeoPixel - 10.01.23

ДОДАТКИ

ДОДАТОК А
Проектування схеми підключення у Proteus



ДОДАТОК Б
Приклад симуляції тестової програми у Proteus



Додаток Г

Код програми управління контролером

```

#include Adafruit_NeoPixel.h
#include GRGB.h
#include EEPROM.h
#include avreeprom.h
#define RED 4
#define BLUE 6
#define GREEN 5
#define PIN 11
#define NUMPIXELS 60

const uint8_t totalplans = 10;
enum regime{Auto=0,Manual=1,Setting=2,Off=3};
Adafruit_NeoPixel strip (NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
uint8_t prevec[5]={0,1,2,3,4};
uint32_t wtime=0,ltime=0;
bool isSelect=false;
volatile int regimeSelect = 0; переменная-счётчик
int mancurplan=0;
int autocurplan=0;
int writenum=0;
Функції роботи стрічки
void striprandom(int del,int n)
{
  for (int i=0;in;i++)

  strip.setPixelColor(i, strip.Color(random(256),random(256),random(256)));
  strip.show();
  delay(del);
  strip.clear();
}
void zat(int del,int n)
{strip.clear();
  for (int j=0;j10;j++)
  {
  for(int i=0;in;i++) strip.setPixelColor(i, strip.Color(j,j,j));
  strip.show();
  delay(del);
  }
  strip.clear();
  for (int j=10;j=0;j--)
  {
  for(int i=0;in;i++) strip.setPixelColor(i, strip.Color(j,j,j));
  strip.show();
  delay(del);
  }
  strip.clear();
}

```

```

}
void striprgb(int del,int n)
{strip.clear();
int i=0;
while(in

{strip.setPixelColor(i, strip.Color(255,0,0));
i++;
strip.setPixelColor(i, strip.Color(0,255,0));
i++;
strip.setPixelColor(i, strip.Color(0,0,255));
i++;
}
strip.show();
delay(del);
}
void stripblack(int del,int n)
{
for (int i=0;in;i++) strip.setPixelColor(i, strip.Color((0),(0),(0)));
strip.show();
delay(del);
}
void stripmove(int n)
{strip.clear();
for (int i=0;in;i++)
{strip.setPixelColor(i, strip.Color(60,0,60));
delay(40);
if (i0) strip.setPixelColor(i-1, strip.Color(0,0,0));
strip.show();
}
}
for (int i=n-1;i=0;i--)
{strip.setPixelColor(i, strip.Color(60,0,60));
delay(40);
if (in-1) strip.setPixelColor(i+1, strip.Color(0,0,0));
strip.show();
}
}
void stripmove2(int n)
{strip.clear();
for (int i=0;in;i++)
{strip.setPixelColor(i, strip.Color(60,0,0));
strip.setPixelColor(n-(i+1), strip.Color(0,60,0));
delay(40);
if (i0) strip.setPixelColor(i-1, strip.Color(0,0,0));
if (i=n-1) strip.setPixelColor(n-i, strip.Color(0,0,0));
strip.show();
}
}
for (int i=n-1;i=0;i--)

```

```

    {strip.setPixelColor(i, strip.Color(60,0,0));
    strip.setPixelColor(n-(i+1), strip.Color(0,60,0));
    delay(40);
    if (in-1) strip.setPixelColor(i+1, strip.Color(0,0,0));
    if (i=0) strip.setPixelColor(n-i-2, strip.Color(0,0,0));
    strip.show();
    }
}
void stripmove4(int n)
{strip.clear();
for (int i=0;i=n2;i++)
{strip.setPixelColor(n2 - i, strip.Color(60,0,0));
strip.setPixelColor(n2 + i, strip.Color(0,0,60));
delay(40);
strip.show();
}
for (int i=0;i=n2;i++)
{strip.setPixelColor(i, strip.Color(0,0,0));
strip.setPixelColor(n-i, strip.Color(0,0,0));
delay(40);
strip.show();
}
for (int i=0;i=n2;i++)
{strip.setPixelColor(n2 - i, strip.Color(60,0,0));
delay(40);
strip.show();
}

for (int i=0;i=n2;i++)
{strip.setPixelColor(i, strip.Color(0,0,0));
delay(40);
strip.show();
}
}
void stripmove3(int del,int n)
{
strip.clear();
for(int i=0;i=n2;i++)
{
strip.setPixelColor(i, strip.Color(60,30,0));
strip.setPixelColor(i+n2, strip.Color(0,0,0));
}
strip.show();
delay(del);
for(int i=0;i=n2;i++)
{

strip.setPixelColor(i, strip.Color(0,0,0));
strip.setPixelColor(i+n2, strip.Color(0,0,60));
}
}

```

```

strip.show();
delay(del);

}
void randomshow(int del,int n)
{strip.clear();
int pos=random(n+1);
strip.setPixelColor(pos, strip.Color(60,0,60));
delay(del);
strip.show();
strip.setPixelColor(pos, strip.Color(0,0,0));
delay(del);
strip.show();
}
void barmove(int n)
{strip.clear();
for (int i=0;in;i++)
{strip.setPixelColor(i, strip.Color(60,0,60));
strip.show();
delay(40);
}

for (int i=n-1;i=0;i--)
{strip.setPixelColor(i, strip.Color(0,0,0));
strip.show();
delay(40);
}
}
void colorchange(int n)
{
strip.clear();
int red=random(80);
int green=random(80);
int blue=random(80);
for (int i=0;in;i++)
strip.setPixelColor(i, strip.Color(red,green,blue));

strip.show();
delay(5000);
stripblack(10,NUMPIXELS);
}
void setup() {
Serial.begin(9600); открыли порт для связи
подключили кнопку на D2 и GND
attachInterrupt(0, isr, CHANGE);
attachInterrupt(1, isr2, CHANGE);
D2 это прерывание 0
обработчик - функция buttonTick
FALLING - при нажатии на кнопку будет сигнал 0, его и ловим

```

```

strip.begin();
stripblack(100,NUMPIXELS);
for(int i=0;i5;i++) eeprom_write_byte(i,i);
for(int i=0;i5;i++) prevec[i]=eeprom_read_byte(i);
}
volatile uint32_t debounce,debounce2;
void isr() {
    оставим 100 мс таймаут на гашение дребезга
    CHANGE не предоставляет состояние пина,
    придётся узнать его при помощи digitalRead
    if (millis() - debounce = 100 && digitalRead(2)) {
    debounce = millis();
    ваш код по прерыванию по высокому сигналу
    counter++; + нажатие
    switch(regimeSelect)
    {
    case Auto autocurplan=0;break;
    case Manual mancurplan++;
    if (mancurplan=totalplans)mancurplan=0;break;
    case SettingisSelect=true;
    writenum++;
    if (writenum=5)writenum=0;break;
    }
    }
}
void isr2() {
    оставим 100 мс таймаут на гашение дребезга
    CHANGE не предоставляет состояние пина,
    придётся узнать его при помощи digitalRead
    if (millis() - debounce2 = 200 && digitalRead(3)) {
    debounce2 = millis();
    ваш код по прерыванию по высокому сигналу
    regimeSelect++; + нажатие
    if (regimeSelect3)regimeSelect=0;
    }
}
void loop() {
wtime=millis()601000;
Serial.print(wtime);
Serial.print( );
Serial.println(prevec[autocurplan]);
switch(regimeSelect)
{case Auto {digitalWrite(RED,HIGH);
digitalWrite(GREEN,LOW);
digitalWrite(BLUE,LOW);
if(wtime-ltime)
{
ltime=wtime;
autocurplan++;
if (autocurplan4)autocurplan=0;break;
}
}
}

```

```

}
switch(prevec[autocurplan])
{
case 0stripmove(NUMPIXELS); strip.clear();break;
case 1stripmove2(NUMPIXELS); strip.clear();break;
case 2randomshow(50,NUMPIXELS); break;
case 3stripmove3(500,NUMPIXELS); strip.clear();break;
case 4stripmove4(NUMPIXELS); strip.clear();break;
case 5striprandom(1000,NUMPIXELS);
strip.clear();

stripblack(1000,NUMPIXELS);
break;
case 6barmove(NUMPIXELS); strip.clear(); break;
case 7striprgb(1000,NUMPIXELS); strip.clear(); break;
case 8colorchange(NUMPIXELS); strip.clear(); break;
case 9zat(200,NUMPIXELS); strip.clear();break;
}
}break;
case Manual {digitalWrite(RED,LOW);
digitalWrite(GREEN,HIGH);
digitalWrite(BLUE,LOW);
switch(mancurplan)
{
case 0stripmove(NUMPIXELS); strip.clear();break;
case 1stripmove2(NUMPIXELS); strip.clear();break;
case 2randomshow(50,NUMPIXELS); break;
case 3stripmove3(500,NUMPIXELS); strip.clear();break;
case 4stripmove4(NUMPIXELS); strip.clear();break;
case 5striprandom(1000,NUMPIXELS);
strip.clear();
stripblack(1000,NUMPIXELS);
break;
case 6barmove(NUMPIXELS); strip.clear(); break;
case 7striprgb(1000,NUMPIXELS); strip.clear(); break;
case 8colorchange(NUMPIXELS); strip.clear(); break;
case 9zat(200,NUMPIXELS); strip.clear();break;
}

}break;
case Setting {digitalWrite(RED,LOW);
digitalWrite(GREEN,LOW);
digitalWrite(BLUE,HIGH);
stripblack(100,NUMPIXELS);
strip.setPixelColor(writenum, strip.Color(50,50,50));
strip.show();
delay(500);
}break;
case Off {digitalWrite(RED,LOW);
digitalWrite(GREEN,LOW);

```



```
digitalWrite(BLUE,LOW);
stripblack(1000,NUMPIXELS);
strip.clear();
if(isSelect)
{ eeprom_write_byte(writenum,mancurplan);prevec[writenum]=mancurplan;isSelect=false;}
}break;
}
}
```