

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА**

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**

до проведення лабораторних занять, організації самостійної роботи  
з навчальної дисципліни

**«ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ»**

*(для здобувачів першого (бакалаврського) рівня вищої освіти  
денної форми навчання зі спеціальності 141 – Електроенергетика,  
електротехніка та електромеханіка,  
освітньо-професійна програма «Світлотехніка та дизайн світлового  
середовища»)*

**Харків**  
**ХНУМГ ім. О. М. Бекетова**  
**2024**

Методичні рекомендації до проведення лабораторних занять, організації самостійної роботи з навчальної дисципліни «Програмування мікроконтролерів» (для здобувачів першого (бакалаврського) рівня вищої освіти денної форми навчання зі спеціальності 141 – Електроенергетика, електротехніка та електромеханіка, освітньо-професійна програма «Світлотехніка та дизайн світлового середовища») / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова ; уклад. : А. І. Колесник, В. М. Поліщук. – Харків : ХНУМГ ім. О. М. Бекетова, 2024. – 44 с.

Укладачі: канд. техн. наук А. І. Колесник,  
канд. техн. наук, доц. В. М. Поліщук

Рецензент

**Л. А. Назаренко**, доктор технічних наук, професор Харківського національного університету міського господарства імені О. М. Бекетова

*Рекомендовано кафедрою світлотехніки і джерел світла, протокол № 3  
від 28 грудня 2022 р.*

Методичні рекомендації призначені для здобувачів спеціальності 141 – Електроенергетика, електротехніка та електромеханіка. Подано засоби та послідовність виконання лабораторних робіт, список рекомендованих джерел, наведено приклади оформлення звіту та питання для самостійного опрацювання.

## ЗМІСТ

ВСТУП.....	4
1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДИСЦИПЛІНИ .....	5
2 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ .....	6
Лабораторна робота 1 .....	6
Лабораторна робота 2 .....	12
Лабораторна робота 3 .....	18
Лабораторна робота 4 .....	23
Лабораторна робота 5 .....	28
Лабораторна робота 6 .....	33
3 РЕКОМЕНДАЦІЇ ЩОДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ .....	38
ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ .....	41
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ.....	42

## ВСТУП

Самостійна робота студента є основним засобом оволодіння матеріалом дисципліни, засвоєння необхідних умінь і навичок у час, вільний від обов'язкових навчальних занять. Під час такої роботи використовується навчальна, спеціальна література, а також тексти лекцій.

Основні види самостійної роботи студента: підготовка та виконання лабораторних робіт; підготовка до проміжного і поточного контролю; виконання завдань для самостійного опрацювання; вивчення питань для поглибленого вивчення дисципліни; виконання самодіагностики.

Метою викладання навчальної дисципліни «Програмування мікроконтролерів» є формування знань щодо основних видів та принципів керування системами освітлення за допомогою мікроконтролера; оволодіння елементною базою та типами мікроконтролерів, використання їх у системах керування освітленням та світлотехнічними пристроями; ознайомлення з мовою програмування та апаратною підтримкою мікроконтролерних пристроїв.

У методичних рекомендаціях розглянуто програмно-апаратний комплекс «*Arduino*». Описано особливості реалізації основних функцій для проектування мікропроцесорних систем на базі плат *Arduino* для управління об'єктом середньої складності. Детально розглянуті основні функції, які використовуються для керування периферією, форматування даних та обміну даними. Описано основні інтерфейси, що використовуються в мікроконтролерах *Arduino*. Наведено шість лабораторних робіт різного ступеня складності, які базуються на програмно-апаратному комплексі «*Arduino*». Завдяки виконанню курсу лабораторних робіт студенти зможуть писати і налагоджувати програми у середовищі розробки «*Arduino*». У результаті вивчення предмета студенти повинні знати:

- класифікацію мікроконтролерних пристроїв;
- принципи дії мікроконтролерів;
- основи мови програмування;
- сферу застосування системи управління освітлювальними приладами.

Програма навчальної дисципліни складається з таких змістових модулів (ЗМ):

ЗМ 1 Методи оцінки перешкод в електромережі від роботи систем освітлення.

ЗМ 2 Основи програмування на C, C++ на ПК.

ЗМ 3 Програмування мікроконтролерів для керування світловими приладами.

# 1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДИСЦИПЛІНИ

Для засвоєння дисципліни студенту необхідно вивчити такі теми:

*Змістовий модуль 1* Методи оцінки перешкод в електромережі від роботи систем освітлення.

Тема 1 Чинники виникнення перешкод в електромережі.

Тема 2 Імпульсні та високочастотні перешкоди.

Тема 3 Наявності спотворень якості електроенергії.

Тема 4 Програмні засоби підтримки проєктування інтелектуальної системи освітлення.

*Змістовий модуль 2* Основи програмування на C, C++ на ПК.

Тема 5 Апаратні засоби підтримки мікроконтролерів AVR.

Тема 6 Стандарти мов програмування C/C++.

Тема 7 Програмування мікроконтролерів AVR.

Тема 8 Ведення в систему вводу-виводу C/C++.

Тема 9 Особливості мікроконтролера *Arduino*.

*Змістовий модуль 3* Програмування мікроконтролерів для керування світловими приладами.

Тема 10 Алгоритм системи віддаленого управління освітленням.

Тема 11 Програмування мікроконтролера для підключення системи освітлення.

Тема 12 Тестування системи.

Для опрацювання зазначених тем необхідно скористатися джерелами, що рекомендується наприкінці методичних рекомендацій або на сайті дистанційного курсу.

Під час вивчення тем модулів студенти виконують лабораторні роботи і отримують практичні навички у вирішенні і виконанні практичного завдання.

*Мета лабораторного практикуму:*

– закріпити і поглибити теоретичні знання, отримані на лекційних заняттях;

– набути вміння самостійно працювати з матеріалом, а також викладати свої думки з новим матеріалом як в усній, так і письмовій формі.

Лабораторні роботи виконуються індивідуально. На першому занятті проводиться інструктаж з техніки безпеки в лабораторії і інструктаж з пожежної безпеки з використанням цифрової платформи для наукової освіти «*Labster*». Під час підготовки до роботи потрібно ознайомитися з темою та метою заняття, які належить досліджувати, використовуючи методичні рекомендації до лабораторної роботи і рекомендовані джерела.

Потім після короткого інструктажу, проведеного викладачем, необхідно розпочати виконувати лабораторну роботу. Лабораторну роботу рекомендується намагатися виконувати самостійно. Наприкінці заняття необхідно показати

викладачеві отримані результати для перевірки. В умовах дистанційного навчання лабораторну роботу проводить викладач у застосунку «*Microsoft Teams*» з використанням необхідного обладнання, студенти можуть відтворити завдання у віртуальній програмі за необхідністю.

Здача робіт передбачає перевірку отриманих результатів, правильності оформлення звіту і захист роботи. При захисті роботи слід чітко орієнтуватися в теоретичному матеріалі, викладеному в методичних вказівках до виконання роботи і на лекціях, і в порядку проведених досліджень, вміти обґрунтовувати і оцінювати отримані результати, робити на їх основі висновки. Для отримання заліку по роботі необхідно продемонструвати розуміння суті поставленого завдання у вигляді відповідей на поставлені викладачем питання.

Тематика лабораторних завдань спрямована на закріплення і поглиблення теоретичних знань, отриманих на лекціях, вироблення умінь і практичних навичок складання ефективної системи керування освітлювальною установкою, вибору складових елементів системи, аналізу заходів, спрямованих на підвищення енергоефективності освітлення. В ході виконання лабораторного завдання студенти повинні набути вміння самостійно працювати з джерелами літератури за новими темами, а також викладати свої судження, як в усній, так і письмовій формі у вигляді відповідей на контрольні питання.

Для виконання лабораторних завдань необхідно мати методичні посібники, підручники та джерела, що рекомендовані викладачем, використання дистанційного курсу. Підсумкова атестація з лабораторного практикуму проводиться за сукупністю всіх виконаних завдань.

## 2 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

### Лабораторна робота 1

#### Основи роботи плати *Arduino*

**Мета роботи:** ознайомитись з апаратно-програмною платформою, її основними компонентами: мікроконтролерна плата вводу / виводу і середовище розробки на мові *Processing / Wiring*.

#### *Короткі теоретичні відомості*

*Arduino* – це інструмент для проектування електронних пристроїв (конструктор), який щільно взаємодіє з навколишнім фізичним середовищем. Це невелика плата з власним процесором та пам'яттю, до якої можна підключити різні компоненти: лампочки, датчики, мотори, чайники, роутери, магнітні дверні замки та взагалі все, що працює від електрики. Принцип роботи такий: «якщо щось трапилось – потрібно щось зробити». Як зробити, що зробити, на що

реагувати – вирішує сам програміст, включаючи різні компоненти до плати та закачуючи до неї алгоритм роботи.

Плати можна розділити на *контролери, шилди й аксесуари*. *Контролери* – це найважливіша частина – плата, яка містить мікроконтролер, і в яку записується виконувана програма. *Шилди* – це плати розширення, які містять ту чи іншу периферію, керовану контролером. Шилд надягається зверху на контролер, утворюючи своєрідний «бутерброд».

На рисунку 1.1 наведено мікросхему *Arduino Uno* – це пристрій на основі мікроконтролера ATmega328. У його склад входить все необхідне для зручної роботи з мікроконтролером: 14 цифрових входів / виходів (з них 6 можуть використовуватися як ШІМ виходи), 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для внутрішньосхемного програмування (ICSP) і кнопка скидання. Для початку роботи з пристроєм достатньо просто подати живлення від АС / DC-адаптера або батарейки, або підключити його до комп'ютера за допомогою USB-кабелю.

На відміну від всіх попередніх плат *Arduino, Uno* як перетворювач інтерфейсів USB–UART використовує мікроконтролер ATmega16U2 (ATmega8U2 до версії R2) замість мікросхеми FTDI.

Як зовнішнє джерело живлення (не USB) може використовуватися мережевий АС / DC-адаптер або акумулятор / батарея. Штекер адаптера (діаметр – 2,1 мм, центральний контакт – позитивний) необхідно вставити у відповідний роз'єм живлення на платі. У разі живлення від акумулятора / батареї, її провід необхідно під'єднати до виводів Gnd і Vin роз'єму POWER. Напруга зовнішнього джерела живлення може бути в межах від 6 В до 20 В. Проте зменшення напруги живлення нижче 7 В призводить до зменшення напруги на виводі 5 В, що може стати причиною нестабільної роботи пристрою. Використання напруги більше 12 В може призводити до перегріву стабілізатора напруги і виходу плати з ладу. З огляду на це рекомендується використовувати джерело живлення з напругою в діапазоні від 7 В до 12 В.

Інтегроване середовище розробки «*Arduino IDE*» – це багатоплатформий додаток на Java, що містить редактор коду, компілятор і модуль передачі прошивки в плату. Щоб почати використовувати «*Arduino IDE*», потрібно зайти на сайт <https://www.arduino.cc> (рисунок 1.2), перейти на вкладку SOFTWARE > DOWNLOADS та завантажити середовище розробки «*Arduino*» (рис.1.2).

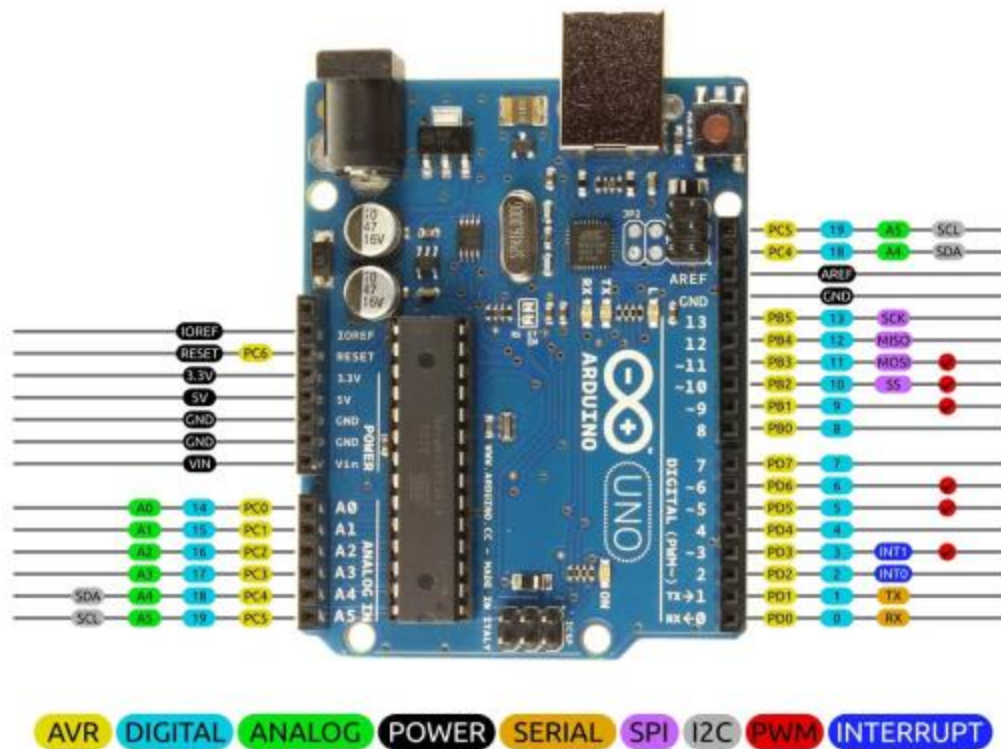


Рисунок 1.1 – Вигляд плати Arduino UNO

У налаштуваннях програми є вибір мови інтерфейсу. Програма, написана в середовищі «*Arduino IDE*», називається «*скетч*». Після того як скетч завантажено і в статусному вікні внизу з'явиться напис «*Завантаження завершено*», можна перевіряти результат. На платі з'явиться миготливий вогник. Паузи між увімкненням і вимиканням мають бути 1 с. Якщо відімкнути контролер від комп'ютера і просто подати на нього живлення (наприклад, від батарейок або блока живлення), то програма все одно буде виконуватися. Програма записана в пам'ять мікроконтролера і тепер буде в ньому до тих пір, поки не буде повторно переписана іншою програмою.




 <p><b>Arduino IDE 2.0.3</b></p> <p>The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.</p> <p>For more details, please refer to the <a href="#">Arduino IDE 2.0 documentation</a>.</p> <p>Nightly builds with the latest bugfixes are available through the section below.</p> <p><small>SOURCE CODE</small> The Arduino IDE 2.0 is open source and its source code is hosted on <a href="#">GitHub</a>.</p>	<p><b>DOWNLOAD OPTIONS</b></p> <p><b>Windows</b> Win 10 and newer, 64 bits Windows MSI installer Windows ZIP file</p> <p><b>Linux</b> AppImage 64 bits (x86-64) Linux ZIP file 64 bits (x86-64)</p> <p><b>macOS</b> Intel, 10.14: "Mojave" or newer, 64 bits macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits</p> <p><small>Release Notes</small></p>	 <p><b>Arduino IDE 1.8.19</b></p> <p>The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.</p> <p>Refer to the <a href="#">Getting Started</a> page for installation instructions.</p> <p><small>SOURCE CODE</small> Active development of the Arduino software is <a href="#">hosted by GitHub</a>. See the instructions for <a href="#">building the code</a>. Latest release source code archives are available <a href="#">here</a>. The archives are PGP-signed so they can be verified using <a href="#">this</a> gpg key.</p>	<p><b>DOWNLOAD OPTIONS</b></p> <p><b>Windows</b> Win 7 and newer Windows ZIP file</p> <p><b>Windows app</b> Win 8.1 or 10 </p> <p><b>Linux</b> 32 bits Linux 64 bits Linux ARM 32 bits Linux ARM 64 bits</p> <p><b>Mac OS X</b> 10.10 or newer</p> <p><small>Release Notes</small> <small>Checksums (sha512)</small></p>
--	---	--	---

Рисунок 1.2 – Програмне забезпечення для роботи з програмування мікроконтролера



Мова *Arduino* має чотири складових: оператори, дані, функції, бібліотеки. *Оператори це:* `setup ()`, `loop ()`, оператори мови C. *Дані:* типи даних з мови C. *Функції:* цифрове введення / виведення, аналогове введення / виведення, час, математичні обчислення, тригонометрія, випадкові числа, біти і байти, зовнішні переривання, переривання. *Бібліотеки:* EEPROM, SD, SPI, SoftwareSerial, Wire, допоміжні класи, клас Serial, клас Stream.

**ПРИМІТКА!** Необхідно вимикати плату від джерела живлення перед тим, як з'єднувати або роз'єднувати елементи на платі прототипування або «*Arduino!*» В іншому випадку плата «*Arduino*» або елементи можуть вийти з ладу!

### Опис установки

Як зазначалося в теоретичній частині, алгоритм роботи пишеться у середовищі програмування «*Arduino IDE*». Це програма, яка завантажується з сайту `arduino.cc` (розділ `software`) та встановлюється на комп'ютер користувача з відповідною операційною системою. Приклад «скетчу», який наведено на рисунку 1.3 є основним.

Це текстовий редактор: у ньому пишеться алгоритм роботи та закачується в Arduino (Sketch -> Upload). Якщо в алгоритмі роботи є синтаксичні помилки – Arduino IDE про це повідомить у нижній частині програми на чорному фоні.

Для зручності використання підгружаємо текстовий редактор для написання скетчу. У коді програми має бути декілька основних команд, які відповідають завданню лабораторної роботи. Вони наведені нижче.

Команди в скетчі:

– *digitalWrite* – це назва функції, яка відповідає за подачу напруги на пін;



Рисунок 1.3 – Вид текстового редактора, скетчу

– *LED\_BUILDIN* – це назва внутрішнього світлодіода. У більшості плат за цією закріплена цифра 13. Для плат Uno та Nano можна писати «13» замість *LED\_BUILDIN*;

- *HIGH* – умовна назва високого рівня сигналу, що включає світлодіод. Можна замінити цифрою «1»;
- *LOW* – умовне позначення низького рівня сигналу, що вимикає світлодіод. Можна замінити цифрою «0»;
- *delay* – функція, яка зупиняє виконання скетчу на певний час. Вкрай небажано використовувати її в реальних проєктах. У дужках ми вказуємо цифру – це кількість мілісекунд, що потрібні на очікування. Як тільки програма дійде до кінця, контролер перейде в початок блоку *loop* і буде виконувати всі команди заново. І так безкінечно. Для складання потрібних компонентів ланцюга використовувати схему, яка показана на рисунку 1.4 [2]. Додати можна світлодіоди, які будуть мигати по черзі. Принцип підключення при цьому не змінюється. Ми використовуємо два або більше пінів плати контролера для з'єднання зі світлодіодами.

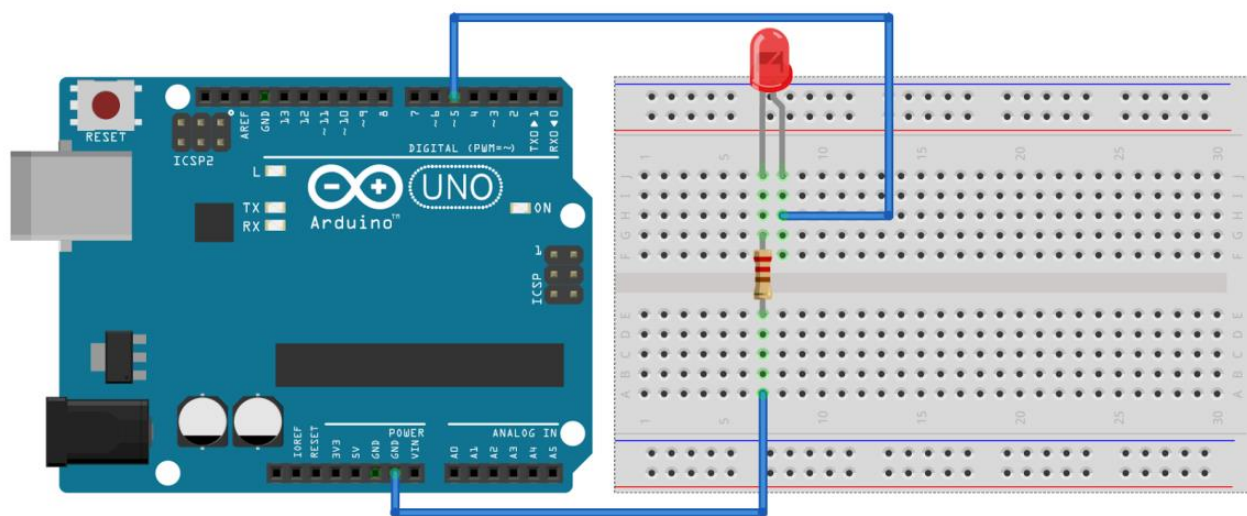


Рисунок 1.4 – Схема «Arduino» UNO

Далі обрати з таблиці 1.1 свій варіант за списком та реалізувати його програмно та апаратно для керуванням світлодіодом від мікроконтролера.

*Зауваження. Опис команд у кодї можна знайти на сайті компанії «Arduino». Не полїнуйтеся вводити код вручну – це допоможе швидше його зрозуміти!*

Таблиця 1.1 – Варіанти завдань до лабораторної роботи № 1

Варіанти	Кількість світлодіодів	Час горіння світлодіодів, с	Час не горіння світлодіодів, с	Піни підключення світлодіодів
1	2	3	4	5
1	4	1,7	1,0	3, 5, 12, 13
2	4	1,0	1,7	4, 5, 6
3	4	2,5	2,6	8, 9, 10, 11

Продовження таблиці 1.1

1	2	3	4	5
4	3	2,6	2,5	9, 10, 11
5	2	1,5	1,3	6, 7
6	3	1,3	1,5	9, 10, 11
7	3	2,0	2,9	7, 8, 9
8	4	1,6	1,6	3, 5, 7, 9
9	2	3,0	2,2	11, 3
10	3	0,9	2,3	3, 4, 5
11	4	2,2	3,0	5, 6, 7, 8
12	2	1,8	1,0	8, 11
13	3	1,0	1,8	4, 7, 10
14	2	2,3	0,9	12, 13
15	2	2,9	2,0	7, 10

Обладнання

Необхідні компоненти наведені у таблиці 1.2.

Таблиця 1.2 – Компоненти схеми для виконання лабораторної роботи

Назва	Фото
Мікроконтролер « <i>Arduino</i> » та USB – кабель	
З'єднувальні проводи	
Резистори (220 Ом)	
Світлодіоди	
Плата прототипування	

Завдання до роботи

1. Ознайомитись із базовою структурою програми для «*Arduino*».
2. Вивчити особливості мови «*Arduino*» (функції, оператори, типи даних).

3. Зібрати схему за рисунком 1.3. Варіанти підключення компонентів до входів-виходів взяти з таблиці 1.1. Написати програму увімкнення-вимкнення для керуванням світлодіодом від мікроконтролера на виходах мікроконтролера залежно від кількості світлодіодів та часу світіння.

4. Налагодити програму в середовищі «*Arduino*» і перевірити на макеті.

5. Відповісти на контрольні питання до лабораторної роботи.

6. Оформити звіт та надати висновки, завантажити у форматі pdf-файл у дистанційний курс.

### *Контрольні запитання*

1. Як вибрати порт на комп'ютері, до якого підключений мікроконтролер?

2. Як вибрати приклад зі бібліотеки стандартних прикладів «*Arduino*» IDE?

3. Як виконати компілювання скетчу?

5. Призначення частин програми розділених командами «*setup*» і «*loop*»?

6. Яка команда відповідає за подачу напруги на пін?

7. Яка команда зупиняє виконання скетчу на певний час?

8. Як програмно встановити пін мікроконтролера «на вхід» або «на вихід»?

## **Лабораторна робота 2**

### **Написання програми: мерехтіння світлодіода**

**Мета роботи:** вивчити світлотехнічні, електричні характеристики світлодіодів, під'єднаних до мікроконтролера, під час регулювання плавного увімкнення / вимкнення.

### *Короткі теоретичні відомості*

У цій лабораторній роботі ми продовжимо говорити про апаратні компоненти вбудованих систем, а саме про мікроконтролери. Мікроконтролер є центром вбудованої системи, тому ми приділимо багато уваги саме роботі з ним. На рисунку 2.1 наведено одну з плат «*Arduino*». Це не мікроконтролер, це друкована плата з безліччю елементів, серед яких, придивившись, ви зможете побачити велику чорну прямокутну мікросхему. Ось це якраз і є мікроконтролер, який виконує програми, що керують пристроєм. Отже, робота цієї мікросхеми полягає в тому, щоб виконувати код, і це центр всієї системи. Він зчитує дані з інших компонентів, і він також керує іншими компонентами.

Чим відрізняється мікроконтролер від мікропроцесора, який використовується в звичайних комп'ютерах? Мікроконтролер зазвичай менший і слабший, ніж мікропроцесор, тому, коли говорять про останній, то мають на увазі мікросхему від компанії «*Intel*» або «*AMD*», що встановлюється в настільний комп'ютер або ноутбук, і яка працює на дуже великій швидкості.



Рисунок 2.1 – Вигляд макетної плати

Мікропроцесори і мікроконтролери широко використовуються у побутовій техніці, автомобільній електроніці, аерокосмічній та військовій галузях і, звичайно ж, у промисловому виробництві.

Обов'язковими компонентами мікропроцесора є регістри, арифметико-логічний пристрій (АЛП) та блок керування. Регістри призначені для тимчасового зберігання даних, арифметико-логічний пристрій – для виконання арифметичних і логічних операцій (тобто, для обробки даних). Блок керування відповідає за послідовне виконання команд програми та правильне спрямування потоків даних. Мікропроцесор не може працювати сам по собі. Він є центральною ланкою мікропроцесорної системи, до якої також входять пристрої постійної та оперативної пам'яті, пристрої введення та виводу інформації, накопичувачі на жорстких магнітних дисках (так звані вінчестери), тощо. Такі мікропроцесорні системи називають комп'ютерами.

Отже, **мікроконтролер** – це спеціалізований мікроелектронний програмований прилад, що призначений для використання у керуючих пристроях, системах передачі даних та системах керування технологічними процесами.

Мікроконтролери використовують у побутовій техніці, медичних приладах, системах керування ліфтами, телефонах, раціях та інших засобах зв'язку, електронних музичних інструментах та автомагнітолах, комп'ютерній периферії (клавіатурах, джойстиках, принтерах, тощо), світлофорах, автоматичних воротах та шлагбаумах, інтерактивних дитячих іграшках, автомобілях, локомотивах та літаках, роботах та промислових верстатах.

Мікроконтролери також широко застосовуються в автомобільній електроніці. Наприклад, автомобіль «Peugeot 206» має на борту 27 мікроконтролерів, а в автомобілях високого класу, як наприклад «BMW» сьомої серії, використовується понад 60 мікроконтролерів. Вони керують вприском палива, жорсткістю адаптивної підвіски, світлотехнікою, двигунами двірників, склопідіймачів та дзеркал заднього вигляду тощо.



Світлодіод можна не тільки вмикати та вимикати, а й можна зробити стробоскопічний ефект, плавно регулювати його яскравість світіння, а також змінювати послідовність вмикання, тобто задавати режими роботи.

*Зовнішній світлодіод.* Світлодіод – це маленька «лампочка», яка світиться під час проходження струму через неї. Струм виникає на ділянці ланцюга при подачі напруги (+5v та GND). Світлодіод буде працювати тільки якщо +5v підключити до довшої ніжки, а GND підключити до більш короткої ніжки. Необхідно згадати, де анод і катод. *Резистор* – це обмежувач струму. Струм, що проходить через світлодіод, потрібно обмежувати резистором (220 Ом), інакше світлодіод перегріється та згорить!

Всі використовувані у лабораторних роботах функції є бібліотечними. У комплекті «*Arduino*» IDE є велика кількість прикладів програм. Під час виконання з реальною платою «*Arduino*» скетчу (див. рис. 1) спостерігається миготіння вбудованого світлодіода. Зовнішній світлодіод та послідовний з ним резистор приєднати з одного боку до пін 13, а з іншого – до виводу GND (Загальний). Опір резистора розрахуємо за формулою 2.1:

$$R = (E - U_d) / I_d \quad (2.1)$$

де  $E$  – напруга живлення;

$U_d$  – напруга на діоді;

$I_d$  – струм через діод.

Якщо відсутні реальна плата «*Arduino*», зовнішні електрорадіоелементи та пристрої, використовується програма-симулятор. Наприклад, UnoArduSim [2].

### Опис установки

Передусім необхідно зібрати на платі прототипування схему, наведену на рисунку 2.2. Принцип з'єднань у платі прототипування подано на рисунку 2.3.

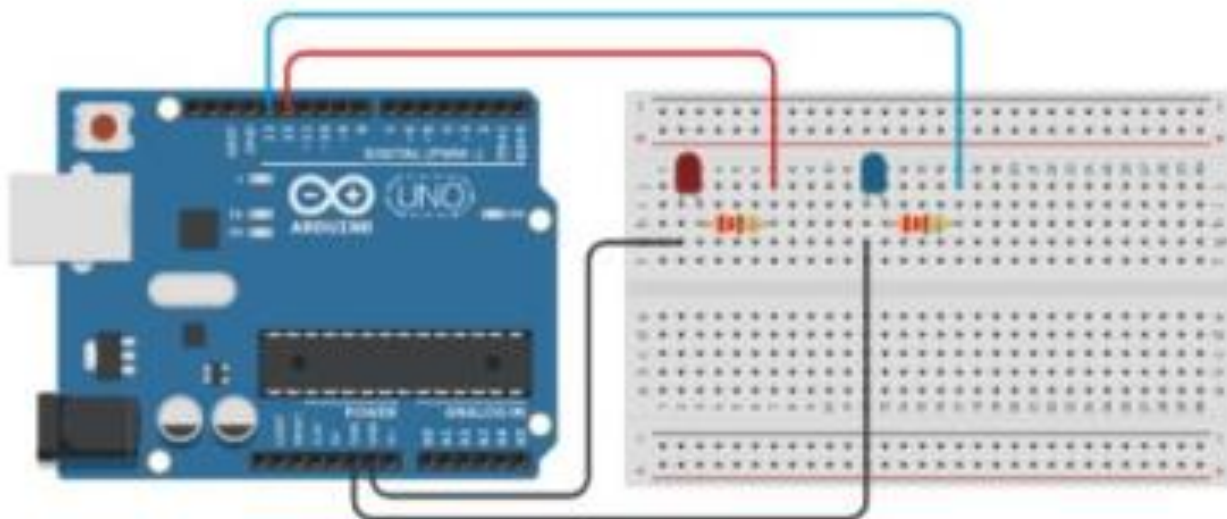


Рисунок 2.2 – Схема підключення світлодіодів до «*Arduino*» UNO

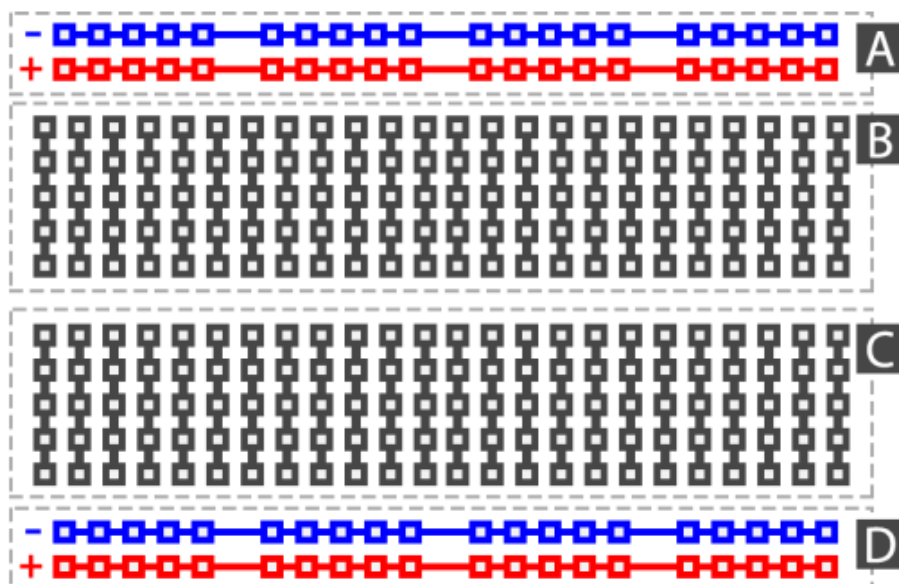


Рисунок 2.3 – Принцип з'єднань у платі прототипування

Збираємо схему, але необхідно змінити програмний код, який наведено на рисунку 2.4. Світлодіод буде плавно вмикатися / вимикатися на цифрових пінах 3, 5, 6, 9, 10, 11, які підтримують сигнали PWM (піни позначені знаком тильда ~ на платі «Arduino»). Опис роботи наведено для під'єданого єдиного світлодіода, для двох і більше світлодіодів необхідно переписати програмний код. У першому циклі змінна  $i$  змінює значення від 0 до 255. Це значення записується в пін 5. Нуль – означає, що світлодіод не горить, 255 – світлодіод горить на повну потужність. Світлодіод плавно спалахує. У другий цикл змінна  $i$  змінює значення від 255 до 0, звідси виходить, що світлодіод плавно гасне. На скільки швидко світлодіод загоряється / гасне – залежить від затримки `delay`.

```

void setup()
{
  pinMode(5, OUTPUT); // определяем пин как выход
}
void loop()
{
  // цикл для переменной i от 0 до 255
  for (int i=0; i<=255; i++)
  {
    analogWrite(5, i);
    delay(3); // пауза 3 мс
  }
  // цикл для переменной i от 255 до 0
  for (int i=255; i>=0; i--)
  {
    analogWrite(5, i);
    delay(3); // пауза 3 мс
  }
}

```

Рисунок 2.4 – Програмний код

## Обладнання

Необхідні компоненти наведені у таблиці 2.1.

Таблиця 2.1 – Необхідні компоненти для виконання лабораторної роботи

Назва	Фото
Мікроконтролер « <i>Arduino</i> » та USB – кабель	
З'єднувальні проводи	
Резистори (220 Ом)	
Світлодіоди	
Плата прототипування	

### Завдання до роботи

1. Ознайомитись із теоретичним матеріалом.
2. Ознайомитись із схемою підключення компонентів до мікроконтролера.
3. Зібрати схему на платі прототипування за рисунком 2.2. Варіанти підключення компонентів до входів-виходів взяти з таблиці 2.2.
4. Написати програму плавного увімкнення-вимкнення для керуванням світлодіодом від мікроконтролера.
5. Прописаний скетч протестувати в середовищі «*Arduino*» і перевірити на макеті.
5. Відповісти на контрольні запитання до лабораторної роботи.
6. Оформити звіт та надати висновки, завантажити у форматі pdf файл у дистанційний курс.



Таблиця 2.2 – Варіанти завдань до лабораторної роботи 2

Варіанти	Кількість світлодіодів	Плавне вмикання	Плавне вимикання	Тривалість паузи (мс) та кількість циклів
1	2	$i = 0, i \leq 255$	$I = 255, i \geq 0$	1, $\infty$
2	3	$i = 0, i \leq 500$	$i = 500, i \geq 0$	5, $\infty$
3	2	$i = 0, i \leq 100$	$i = 100, i \geq 0$	3, $\infty$
4	2	$i = 0, i \leq 455$	$i = 455, i \geq 0$	2, $\infty$
5	3	$i = 0, i \leq 105$	$i = 105, i \geq 0$	0,5, $\infty$
6	2	$i = 0, i \leq 255$	$i = 455, i \geq 0$	3, $\infty$
7	2	$i = 0, i \leq 455$	$i = 100, i \geq 0$	6, $\infty$
8	3	$i = 0, i \leq 200$	$i = 200, i \geq 0$	2, $\infty$
9	2	$i = 0, i \leq 300$	$i = 300, i \geq 0$	1, $\infty$
10	2	$i = 0, i \leq 255$	$i = 455, i \geq 0$	3, $\infty$
11	3	$i = 0, i \leq 455$	$i = 100, i \geq 0$	5, $\infty$
12	2	$i = 0, i \leq 100$	$i = 100, i \geq 0$	1, $\infty$
13	2	$i = 0, i \leq 300$	$i = 455, i \geq 0$	4, $\infty$
14	3	$i = 0, i \leq 455$	$i = 100, i \geq 0$	2, $\infty$
15	2	$i = 0, i \leq 300$	$i = 255, i \geq 0$	3, $\infty$

### Контрольні запитання

1. У чому відмінність мікроконтролера та мікропроцесор?
2. Як впливають на термін роботи світлодіодів часті вмикання та вимикання їх?
3. Яка команда відповідає за подачу напруги на пін?
4. Як називається пристрій, за допомогою якого остаточну версію програми переносять з комп'ютера на мікроконтролер?
5. Виберіть правильні твердження:
  - а) робота мікроконтролера полягає в тому, щоб виконувати код;
  - б) мікропроцесор менший і слабший за мікроконтролер;
  - в) процесор і пам'ять мікроконтролера початкового рівня знаходяться на різних мікросхемах;
  - г) для підвищення продуктивності мікроконтролера використовують зовнішню пам'ять.
6. Що працює швидше: програма, що виконується на звичайному процесорі, або ППВМ (вентильні матриці), що реалізує ту ж логіку, що і програма?
7. Оберіть датчики, з яких може надходити аналогова інформація:
  - а) кнопка;
  - б) датчик, що вимірює рівень шуму;
  - в) датчик освітленості;
  - г) трипозиційний перемикач.

## Лабораторна робота 3

### Програма світлофор

**Мета роботи:** познайомитись докладніше з алгоритмом роботи пристрою, мікроконтролера, а також розібрати кілька варіантів написання програми реалізації світлофору на «*Arduino*».

#### *Короткі теоретичні відомості*

У цій лабораторній ми поговоримо про програмування мікроконтролерів. Точніше, як обробляється програма перед тим, як її виконає мікроконтролер. Ключовим моментом тут є те, що мікроконтролер насправді не виконує буквально той код, який пишуть програмісти. Спочатку він повинен бути оброблений транслятором, а потім вже він виконується мікроконтролером. Навіщо потрібен цей додатковий етап? Він потрібен тому, що мікроконтролер, як і будь-який інший процесор, не розуміє мов Сі, або С ++, Java, Паскаль, Пітон і будь-які інші мови, якими зазвичай пишуться програми. Він розуміє і може виконувати тільки свою власну машинну мову. Візьмемо процесор Intel. Він розуміє машинну мову X86. Інші процесори, наприклад, мікроконтролери Atmel, які стоять на платах «*Arduino*», розуміють свою машинну мову. Різні сімейства процесорів використовують свою власну мову. Що вона собою становить? Це набір простих інструкцій, закодованих у двійковому форматі за допомогою нулів і одиниць. Тому, якщо ви подивитесь безпосередньо на машинний код, то все, що ви побачите, – це нулі і одиниці. Це, звичайно, рідко хто робить, але якщо і робить, то зазвичай код записується в шістнадцятковому форматі – так він займає менше місця і трохи більше зручний для сприйняття. Але в будь-якому випадку машинна мова не виглядає читаною для людини, хоча це те, що машина насправді виконує. Для того щоб зробити програми більш легкими для читання, їх переводять мовою асемблера, які складаються з коротких і зрозумілих мнемонічних команд, на кшталт ADD, SUB, MUL, MOV, JMP і т. д. Ніякої виразної потужності асемблер не додає, оскільки, по суті, це однозначне відображення з машинних кодів в слова. Він існує тільки заради зручності читання машинних кодів людиною.

Хоч мову асемблера і придумали для полегшення читання і запису програм, але все-таки це дуже низький рівень. Асемблерна мова називається мовою з мітками, бо навпроти будь-якої інструкції можна поставити мітку, а потім перейти до неї за допомогою команд умовного і безумовного переходу (наприклад goto у високорівневих мовах). Причому переходи по мітках – це єдиний спосіб організувати розгалуження і цикли, оскільки в подібних мовах немає ні циклів for або while, ні можливості об'єднати інструкції в блоки коду, як у структурованих мовах програмування. Можна писати код мовою асемблера, але це важко, і ми не робитимемо це в цій роботі. Іноді це необхідно, щоб досягти максимальної продуктивності. Ми працюватимемо з мовами вищого рівня: Сі,

C++, Python, Lua. Таким чином, програма, написана мовою високого рівня, має бути переведена на машинну мову, перш ніж її зможе виконати мікроконтролер.

Існує два способи, якими це можна зробити. Якщо використовується компільована мова, то програма цією мовою відразу цілком перекладається в машинний код. Після такого перекладу ви отримуєте те, що називається виконуваним файлом, що містить саме той код, який виконується щоразу під час запуску програми. Приклади компільованих мов – Сі, С ++, Паскаль й інші. Коли ми використовуватимемо плати «*Arduino*», ми працюватимемо із спрощеною мовою С ++. З іншого боку, крім компіляції, є також інтерпретація. В інтерпретованому середовищі інструкції мовою високого рівня перетворюються в машинний код щоразу під час виконання за допомогою інтерпретатора. Прикладами таких мов є Пітон, Java, Lua та ін. Недолік інтерпретованих програм у тому, що вони працюють суттєво повільніше, адже процесор виконує не їх, а код інтерпретатора, який, зі свого боку, виконує вашу програму! Скомпільована ж програма, як вже було сказано раніше, займає центральний процесор цілком і повністю без будь-яких посередників. Але інтерпретація має і переваги, оскільки вона звільняє програміста від певних завдань. Наприклад, якщо в Сі ви хочете використовувати динамічний масив, вам потрібно викликати спеціальну функцію для отримання пам'яті, а потім не забути звільнити цю пам'ять. Це дуже складно відстежити і зробити коректно у великих програмах, тому часто виникають помилки. З іншого боку, в інтерпретованих мовах ви просто використовуєте готовий динамічний масив, а інтерпретатор вирішує всі завдання з керуванням пам'яттю. Тому яку мову використовувати, як завжди, залежить від розв'язуваної задачі.

У цій лабораторній роботі ми продовжимо говорити про апаратні компоненти вбудованих систем, а саме про мікроконтролери, і отже розглянемо далі на практиці підключення світлодіодів і написання програми «світлофор» з використанням плати «*Arduino*». Мікроконтролер є центром вбудованої системи, тому ми приділимо багато уваги саме роботі з ним. На рисунку 3.1 наведено вигляд світлодіодного модуля для виконання лабораторної роботи. Такий варіант може використовуватись, але на практиці ми будемо проводити роботи з світлодіодами одиничними. Варіантів використання таких приладів дуже багато.



Рисунок 3.1 – Вигляд світлодіодного модуля світлофора 5 мм

Повсюдно поширені основні сигнали світлофорів:

- червоний сигнал світлофора забороняє проїзд за стоп-лінію (за її відсутності за світлофор);
- жовтий зобов'язує скинути швидкість і бути готовим до того, що світлофор через 0,5–1 секунд перемкнеться на червоний;
- зелений – дозволяє рух зі швидкістю, що не перевищує максимальний рівень для конкретної автотраси.

Візьмемо за основу алгоритму цю поширену роботу сигналів світлофора, та пропишемо його у вигляді коду для завантаження у мікроконтролер.

### Опис установки

Насамперед необхідно зібрати на платі прототипування схему, наведену на рисунку 3.2. Кожен світлодіод підключений через резистор до окремого виводу Arduino. Червоний – виводу 2, жовтий – виводу 3, зелений – виводу 4. Якщо подати одиничний сигнал на виводі 2, то загориться червоний світлодіод, якщо на 3 – жовтий, якщо на 4 – зелений.

Запустіть на комп'ютері програму *Arduino IDE*, підключіть мікроконтролер через кабель USB до комп'ютера. Виберіть у програмі потрібну плату та порт. Інструменти – плата – *Arduino uno*. Інструменти – порт – COM 4 (цифра може бути іншою).

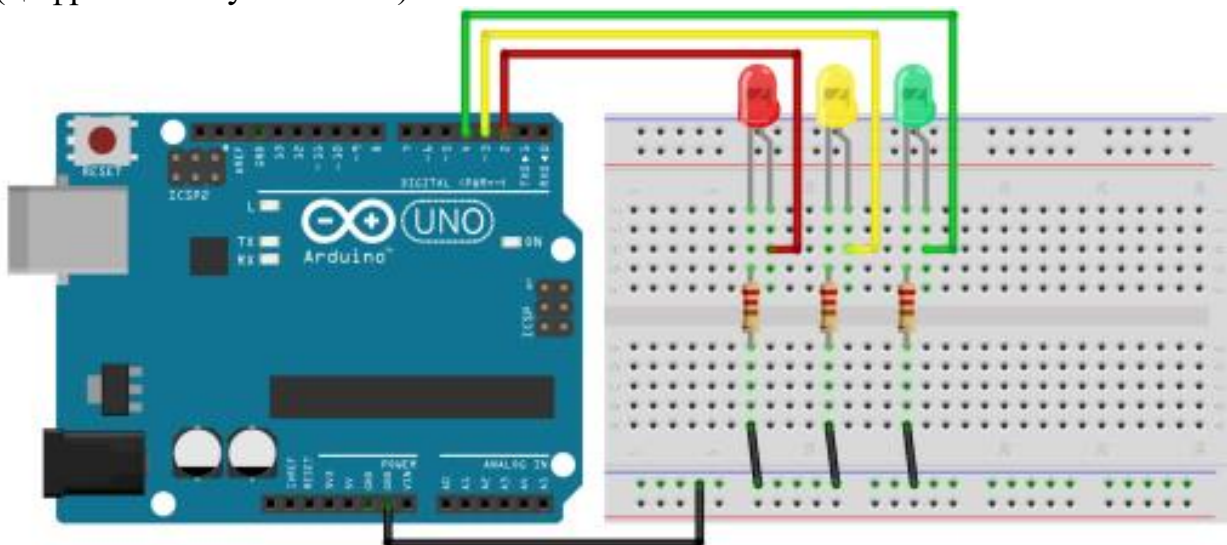
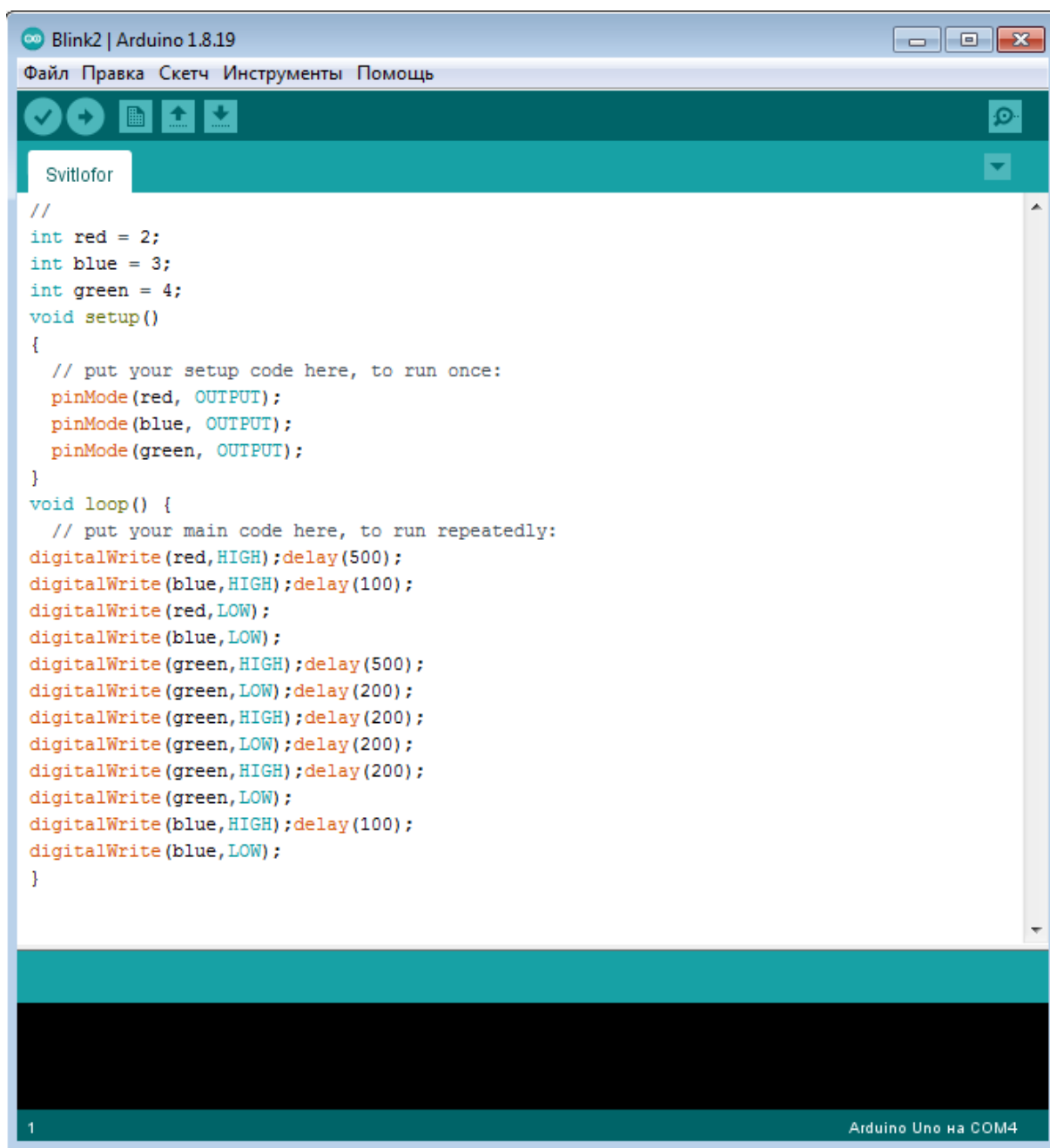


Рисунок 3.2 – Схема підключення світлодіодів до *Arduino UNO*

Збираємо схему, але необхідно змінити програмний код, який наведено на рисунку 3.3. Складання аналогічно минулим лабораторним, але в цій роботі ми використаємо 3 світлодіоди, і ми задіємо 3 цифрових порти. Кожен світлодіод має бути захищений своїм резистором 220 Ом.

Зверніть увагу на полярність світлодіода: довга ніжка – це плюс, який підключається до цифрового порту, а коротка ніжка з'єднується із масою (GND). Опір ставиться у розрив ланцюга, однаково з якого боку світлодіода – плюса чи землі. На схемі опір поставлено з боку землі (GND). Насправді можна

використовувати будь-які цифрові піни «Arduino» від 2 до 12. Зробіть це, як вам зручніше, але не забудьте скоригувати номери пінів у скетчі.



```
//
int red = 2;
int blue = 3;
int green = 4;
void setup()
{
  // put your setup code here, to run once:
  pinMode(red, OUTPUT);
  pinMode(blue, OUTPUT);
  pinMode(green, OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(red,HIGH);delay(500);
  digitalWrite(blue,HIGH);delay(100);
  digitalWrite(red,LOW);
  digitalWrite(blue,LOW);
  digitalWrite(green,HIGH);delay(500);
  digitalWrite(green,LOW);delay(200);
  digitalWrite(green,HIGH);delay(200);
  digitalWrite(green,LOW);delay(200);
  digitalWrite(green,HIGH);delay(200);
  digitalWrite(green,LOW);
  digitalWrite(blue,HIGH);delay(100);
  digitalWrite(blue,LOW);
}
```

Рисунок 3.3 – Програмний код

### *Обладнання*

Необхідні компоненти наведені у таблиці 3.1. Для самостійного виконання роботи за одним варіантом необхідно скористуватись додатковими джерелами, які наведено в кінці роботи для написання алгоритму роботи та програмного коду. На рисунку 3.3 наведено код програми «світлофор», а для під'єднання кнопки потрібно додатково прописати команди, яким буде присвячено теоретичний матеріал.

Таблиця 3.1 – Необхідні компоненти для виконання лабораторної роботи

Назва	Фото
Мікроконтролер « <i>Arduino</i> » та USB – кабель	
З'єднувальні проводи	
Резистори (220 Ом)	
Світлодіоди	
Плата прототипування	
Кнопка	

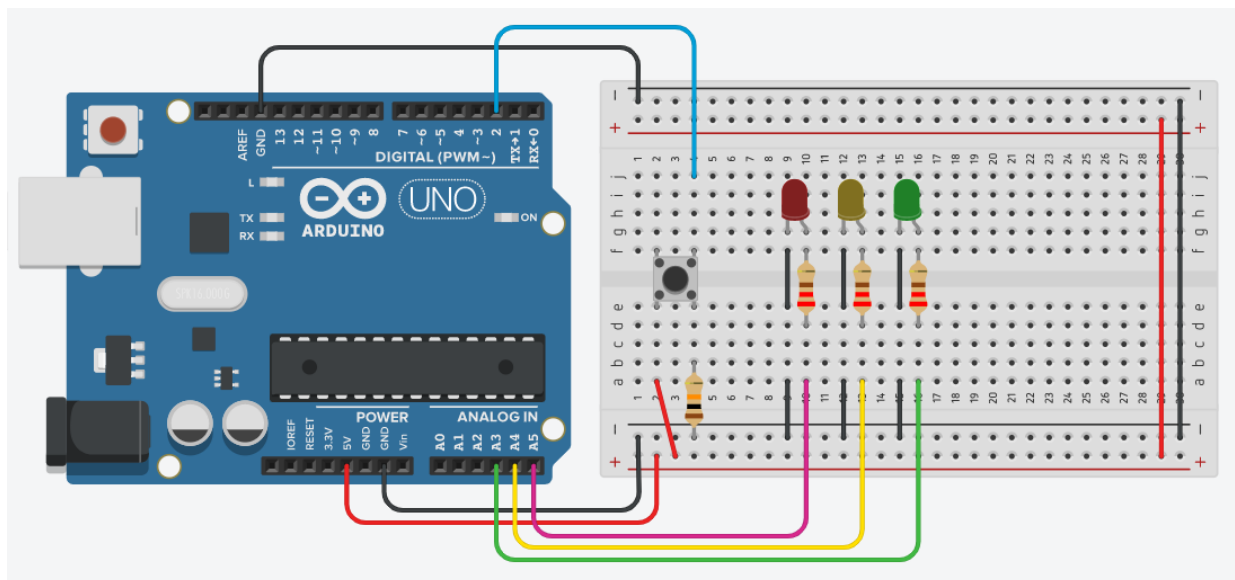


Рисунок 3.4 – Схема підключення світлодіодів до «*Arduino*» UNO (завдання до самостійної роботи)

### *Завдання до роботи*

1. Ознайомитись із теоретичним матеріалом.
2. Ознайомитись із схемою підключення компонентів до мікроконтролера.
3. Зібрати схему на платі прототипування за рисунком 3.4 з використанням кнопки перемикання.
4. Написати програму роботи світлофора, який може керуватись кнопкою.
5. Прописаний скетч протестувати в середовищі «*Arduino*» і перевірити на макеті.
5. Відповісти на контрольні питання до лабораторної роботи.
6. Оформити звіт та надати висновки.

### *Контрольні запитання*

1. Які саме пристрої належать до спеціалізованих професійних інструментальних засобів налагоджування?
2. Що є найважливішим інструментом будь-якого розробника програмного забезпечення?
3. Наведіть структуру типового схемного емулятора.
4. Розмістіть в порядку від низького до високого рівня такі способи написання коду:
  - а) код на Cі, Java, Python;
  - б) машинний код;
  - в) мова асемблера.
5. Виберіть правильні твердження:
  - а) Python є компільованою мовою;
  - б) інструкції компільованої мови цілком переводяться в машинний код тільки один раз;
  - в) приклад компільованої мови C ++;
  - г) код, написаний компільованою мовою, виконується повільніше, ніж код, написаний інтерпретуючою мовою.

## **Лабораторна робота 4**

### **Використання триколірного *RGB*-світлодіода**

**Мета роботи:** навчитись програмувати систему управління триколірним світлодіодом на базі мікроконтролерів «*Arduino*».

#### *Короткі теоретичні відомості*

Особливість цього модульного світлодіода у можливості змішування кольорів. Цей триколірний світлодіодний модуль (рис. 4.1) містить RGB-

світлодіод з трьома входами, що дозволяє отримувати потрібний колір випромінювання за допомогою подачі потрібної напруги на вході. Світлодіод має загальний катод. На платі встановлені обмежувальні резистори. Є можливість підключення до різних мікроконтролерів. Модуль RGB-світлодіод KY-016 призначений для використання в моделях і пристроях, які використовують платформу «Arduino». Світлодіод містить кристали трьох кольорів: червоний, зелений і синій. Вони можуть світитися одночасно або окремо. Також може світитися всіма кольорами веселки. Цього можна досягти, змінюючи інтенсивність світла на ніжках, що відповідають за три основні кольори. На платі модуля розташований штирьовий роз'єм, має чотири контакти з позначеннями.

Подібні RGB-світлодіоди застосовуються для індикації режимів роботи приладів, підсвічування РКІ індикаторів і клавіш. Використовується в іграшках і складній розважальній техніці. Ви можете подавати різне значення ШІМ-сигналу в діапазоні від 0 до 255, що дозволить отримати практично будь-який колір з 16 000 000 можливих. Можна підключити безпосередньо до материнської плати «Arduino», без лінії «Dupont». Стандартні характеристики наведено в таблиці 4.1.

Таблиця 4.1 – Характеристики триколірного світлодіода

Характеристика	Значення	Одиниця вимірювання
Робоча напруга	3,3/5	В
Номінальний струм	20	мА
Максимальний імпульсний струм	50	мА
Розмір	19 × 15	мм

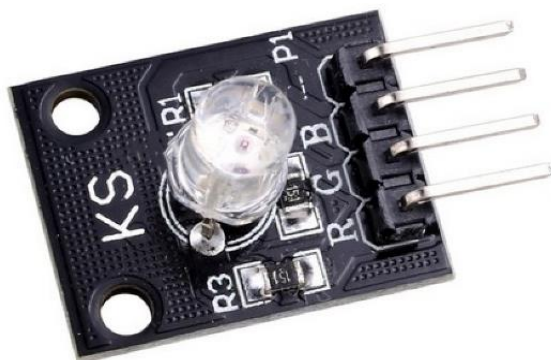


Рисунок 4.1 – Вигляд світлодіодного триколірного модуля



## Опис установки

**Зауваження.** На RGB-модулі помилково вказані літери кольорів. Правильні значення: G – червоний, B – синій, R – зелений.

Насамперед необхідно зібрати на платі прототипування схему, наведену на рисунку 4.2. На початку коду оголошуємо змінні, даючи їм імена кольорів, а значення – номер піна. У функції «setup» визначаємо піни як виходи. Створюємо свою функцію «show», яка прийматиме 3 параметри: значення червоного, зеленого та синього кольорів від нуля до 255. Змішуючи ці 3 кольори у різних пропорціях, отримуємо новий колір.

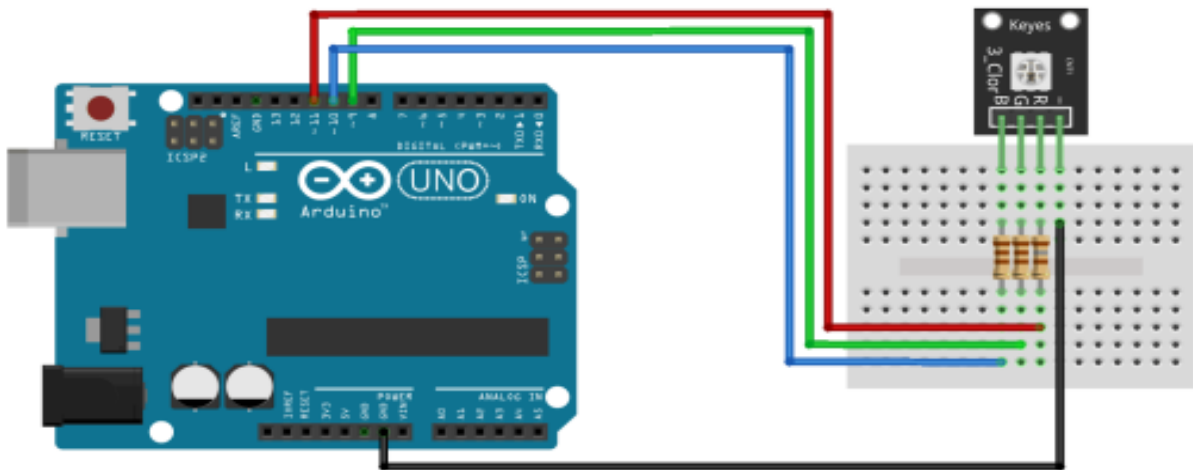


Рисунок 4.2 – Схема підключення триколірного світлодіода до *Arduino UNO*

```
RGB_LED | Arduino 1.8.19
Файл  Правка  Скетч  Инструменты  Помощь

RGB_LED

//
int red = 9;
int green = 10;
int blue = 11;
void setup()
{
  // put your setup code here, to run once:
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(blue, OUTPUT);
}
//
void show(int r, int g, int b)
{
  analogWrite(red, r);
  analogWrite(green, g);
  analogWrite(blue, b);
}
void loop()
{
  show(255,0,0); delay(1000);
  show(0,255,0); delay(1000);
  show(0,0,255); delay(1000);
}
```

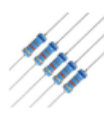
Рисунок 4.3 – Програмний код

Переглянути палітру кольорів можна в програмі MSPainte в операційній системі Windows. У функції «loop» перебираємо кольори з інтервалом 1 с. Запустіть на комп'ютері програму «Arduino» IDE, підключіть мікроконтролер через кабель USB до комп'ютера. Виберіть у програмі потрібну плату та порт. Інструменти – плата– «Arduino uno». Інструменти – порт – COM 4 (цифра може бути іншою).

### Обладнання

Необхідні компоненти наведені у таблиці 4.2. Для самостійного виконання роботи необхідно написати програму роботи цієї схеми, з урахуванням власних побажань перемикання та тривалості вмикання різних кольорів, користуючись додатковими джерелами, які наведено в кінці роботи.

Таблиця 4.2 – Необхідні компоненти для виконання лабораторної роботи

Назва	Фото
Мікроконтролер «Arduino» та USB – кабель	
З'єднувальні проводи	
Резистори (220 Ом)	
Світлодіоди	
Плата прототипування	
Триколірний світлодіодний модуль	

## Завдання до роботи

1. Ознайомитись із теоретичним матеріалом.
2. Ознайомитись із схемою підключення компонентів до мікроконтролера.
3. Зібрати схему на платі прототипування за рисунком з використанням триколірного світлодіода.
4. Написати програму роботи цієї схеми з урахуванням власних побажань перемикання та тривалості вмикання різних кольорів.
5. Прописаний скетч протестувати в середовищі «*Arduino*» і перевірити на макеті.
5. Відповісти на контрольні запитання до лабораторної роботи.
6. Оформити звіт та надати висновки, завантажити у форматі pdf файл у дистанційний курс.

## Контрольні запитання

1. Які функції обов'язково мають бути присутніми в скетчі для *Arduino*?
  - а) `setup ()`;
  - б) `main ()`;
  - в) `startup ()`;
  - г) `loop ()`;
  - д) `init ()`.
2. Виберіть правильні твердження:
  - а) «клас» – це тип даних, а «об'єкт» – це значення такого типу;
  - б) для використання будь-якої стандартної функції в середовищі *Arduino*, потрібно на початку програми написати директиву `#include` і вказати ім'я бібліотеки;
  - в) бібліотеки пишуть для спрощення роботи з різними пристроями;
  - г) щоб працювати з двома однаковими пристроями, для яких є бібліотека, потрібно буде двічі підключити цю бібліотеку за допомогою `#include`;
  - д) щоб працювати з двома однаковими пристроями, для яких є бібліотека, потрібно буде один раз підключити бібліотеку за допомогою `#include` і створити два об'єкти відповідного класу.
3. Які особливості необхідно враховувати при живленні плати «*Arduino*» від зовнішніх джерел живлення?
4. Що таке калібрування датчика? Як проводиться калібрування датчика та для чого воно потрібне?
5. Що таке світлодіод? Як правильно його підключити до плати «*Arduino*?»
6. Перерахуйте та поясніть основні функції та константи, що використовуються під час роботи зі світлодіодом та «*Arduino*».
7. Які існують способи підключення кнопки у схему з платою «*Arduino*?»
8. Що таке брязкіт кнопки та як його знешкодити?

## Лабораторна робота 5

### Фоторезистор

**Мета роботи:** дослідити роботу зібраної схеми з використанням фотоелемента та мікроконтролера «*Arduino*».

#### *Короткі теоретичні відомості*

У цій лабораторній роботі ми розглянемо підключення фоторезистора до мікроконтролера «*Arduino*». Що таке фоторезистор? Фоторезистор – це резистор, який змінює свій опір після зміни кількості світла, що на нього потрапляє. Іншими словами фоторезистор – це резистор, опір якого залежить від кількості світла, що падає на його поверхню, або це електронний компонент, чутливий до освітленості. Його опір змінюється залежно від кількості світла, яке на нього падає. За допомогою «*Arduino*» і фоторезистора можна вимірювати освітленість в навколишньому середовищі та використовувати ці дані для різних цілей. LDR – light dependent resistor. Підключається до «*Arduino*» за допомогою додаткового резистора 10 кОм (рис. 5.2). Приклад такого фоторезистора наведено на рисунку 5.1.

Для виготовлення фоторезисторів використовують напівпровідникові матеріали із шириною забороненої зони, оптимальною для розв'язуваного завдання. Зокрема, для реєстрації видимого світла використовуються фоторезистори з селеніду та сульфїду кадмію, Se. Для реєстрації інфрачервоного випромінювання використовуються Ge (чистий або легований домішками Au, Cu або Zn), Si, PbS, PbSe, PbTe, InSb, InAs, HgCdTe, що часто охолоджуються до низьких температур за допомогою термоелектричного охолоджувача, компактної судини Дьюара з рідким азотом за допомогою дроселювання азоту. Напівпровідник наносять у вигляді тонкого шару на скляну або кварцову підкладку або вирізають у вигляді тонкої пластинки монокристалу. Шар або пластинку напівпровідника забезпечують двома електродами і поміщають у захисний корпус.

Під час підключенні фоторезистора до мікроконтролера, наприклад *Arduino* або *Espruino*, виникне проблема: контролери не вмiють зчитувати опір компонентів. Для підключення фоторезистора до мікроконтролерів використовуйте схему дільника напруги. Резисторний дільник – це два послідовно з'єднані резистори між плюсом і мінусом, які називаються плечима. Сума напруги на плечах дорівнює вхідній напругі живлення. Плече між мінусом та середньою точкою називають нижнім, а друге плече – верхнім.

Фоторезистор у загальному випадку не призначений для точного вимірювання освітленості, скоріше придатний для визначення, наскільки світлішим чи темнішим стає навколишнє середовище.

У конструкції фоторезисторів немає р-n переходу, тим самим є можливість підключати компоненти у схему, не плутаючи плюс і мінус. Тут наявна інертність, тобто існує час затримки між зміною опору від освітлення. Тому для

значного падіння опору від впливу променя світла необхідно близько десяти мілісекунд. При зворотній дії знадобиться близько 1 секунди для відновлення значення опору. Далі через ці особливості фоторезистор поступово витісняється іншими компонентами, які швидко фіксують значні стрибки освітленості.

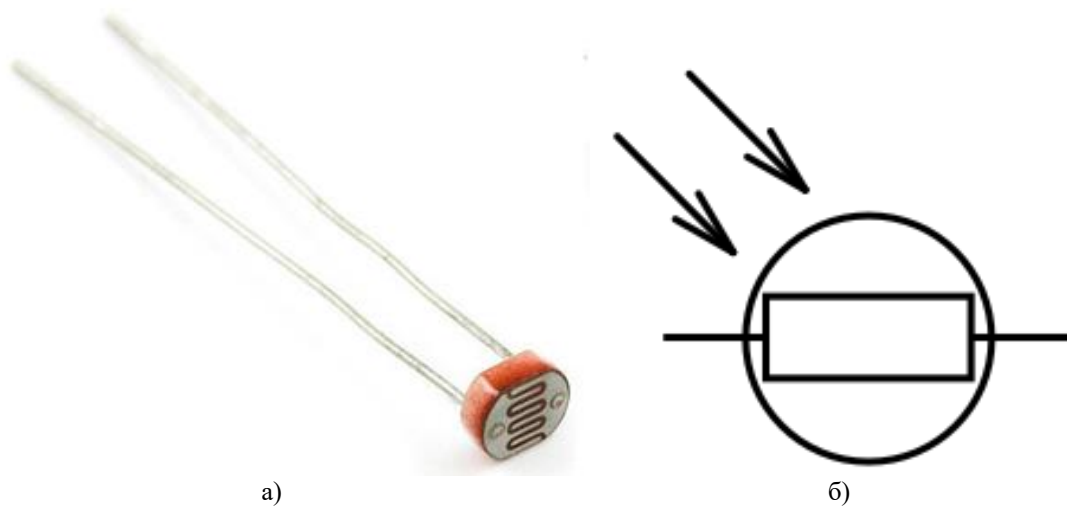


Рисунок 5.1 –Вигляд фотоелемента:  
а – реальний фоторезистор; б – умовне позначення фоторезистора на електричних принципових схемах

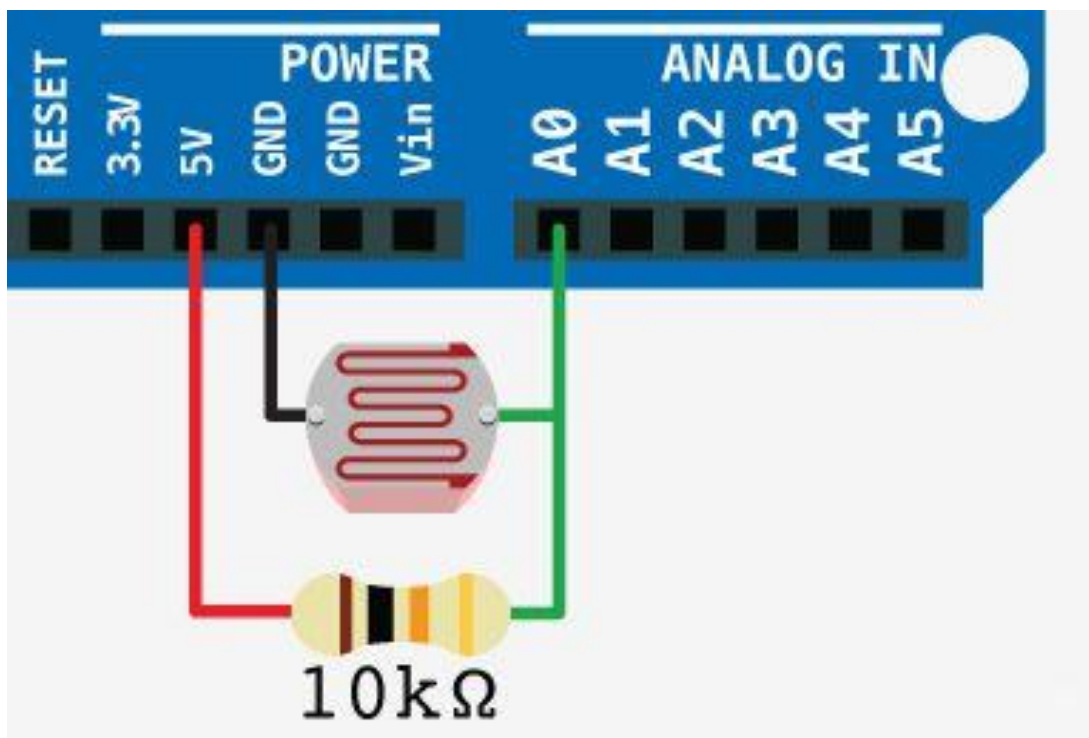
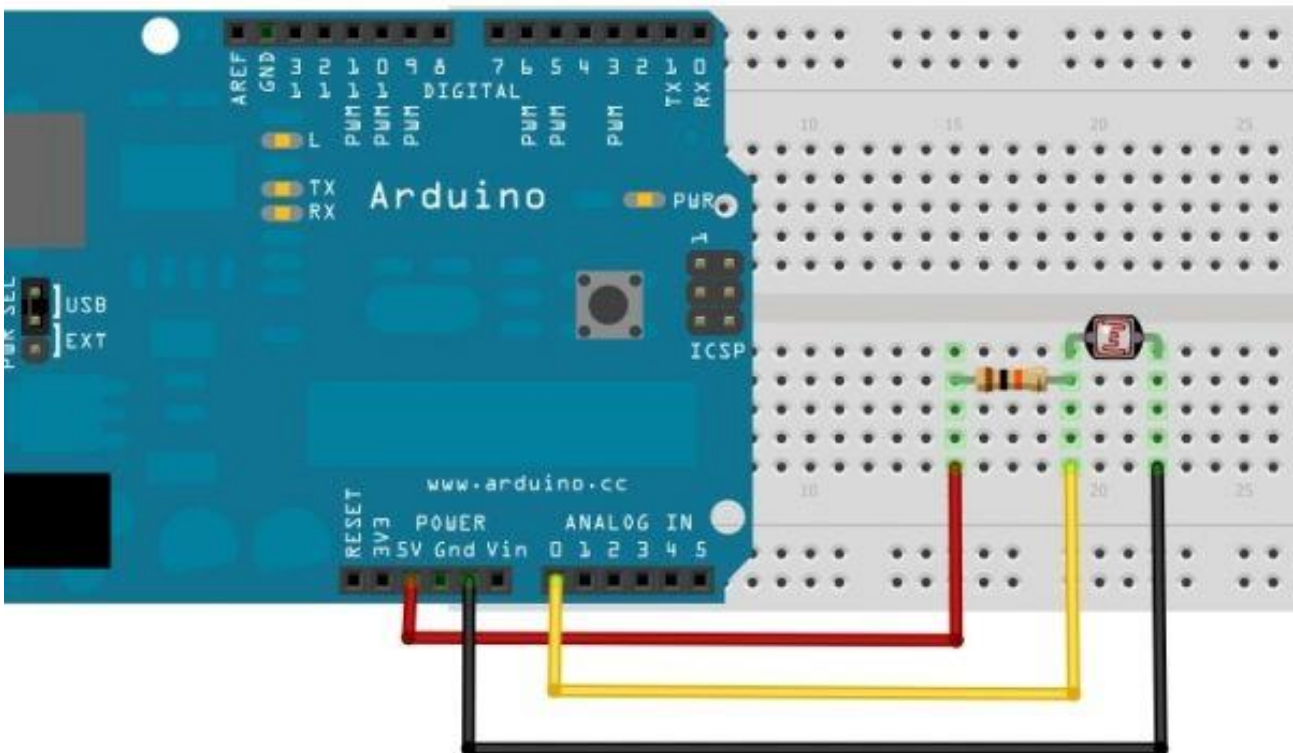


Рисунок 5.2 – Схема підключення фоторезистора до плати «Arduino»

## Опис установки

**ЗАУВАЖЕННЯ.** Аналоговий пін «*Arduino*» читає значення від нуля до 1023. Після відкриття монітора порту перевірте, щоб швидкість порту, вказана в коді (`Serial.begin(9600)`), збігалася зі швидкістю в моніторі (9600 baud).

Передусім необхідно зібрати на платі прототипування схему, наведену на рисунку 5.3. У функції «`setup`» запускаємо обмін даними по порту «*Serial*» – тепер наша «*Arduino*» зможе виводити будь-які значення в цей порт, а ми зможемо їх бачити через монітор порту. Порт у цьому випадку – це інтерфейс, у якому «*Arduino*» підключається до комп'ютера (USB кабель, цифрові піни 0 та 1). У функції «`loop`» ми створюємо змінну `ldr` і записуємо до неї значення, яке читаємо з аналогового піна A0 «*Arduino*». Кожні 100 мс виводимо значення `ldr` у монітор порту. Якщо прикрити фоторезистор рукою, то значення змінної `ldr` буде близько до 1023. Якщо підсвітити фоторезистор ліхтариком, то значення змінної `ldr` буде близьким до нуля.



Рисunek 5.3 – Схема підключення фоторезистора до «*Arduino*» UNO

У повній темряві він має максимальний опір у сотні кіло Ом, а в міру зростання освітленості опір зменшується до десятків кіло Ом. Підключимо фоторезистор, як показано в описі до роботи, підключимо до мікроконтролера та включимо програму. Закачаємо в «*Arduino*» програмний код, який наведено на рисунку 5.4. Оскільки фоторезистор не має полярності, можна підключити будь-якою стороною, до плати його можна припаяти, під'єднати проводами за допомогою монтажної плати або використовувати звичайні кліпси (крокодили)

для з'єднання. Джерелом живлення у схемі є сам «*Arduino*». Нижче подано основний приклад програми для вимірювання освітленості з фоторезистором:

```
void setup()
{
  // запускаем порт на скорости 9600
  Serial.begin(9600);
}
void loop()
{
  // читаем значение фоторезистора
  int ldr = analogRead(A0);
  // выводим значение в порт
  Serial.println(ldr);
  // пауза 100 мс
  delay(100);
}
```

Рисунок 5.4 – Програмний код

У цьому коді *analogRead* (photoresistorPin) зчитує значення аналогового сигналу з фоторезистора, і *Serial.println* (sensorValue) виводить його на *Serial Monitor* в «*Arduino*» IDE. Затримка 1 секунда дозволяє проводити вимірювання з інтервалами.

Під час розроблення проєкту важливо врахувати калібрування фоторезистора під конкретні умови освітлення. Ви можете вимірювати значення в певних умовах і використовувати ці дані для конвертації значень фоторезистора в реальні одиниці вимірювання освітленості, такі як люкс (Lux).

Після завантаження коду відкриваємо монітор порту (Tools -> Serial monitor) або натискаємо CTRL + SHIFT + M. У монітор порту надходять значення, отримані від фоторезистора.

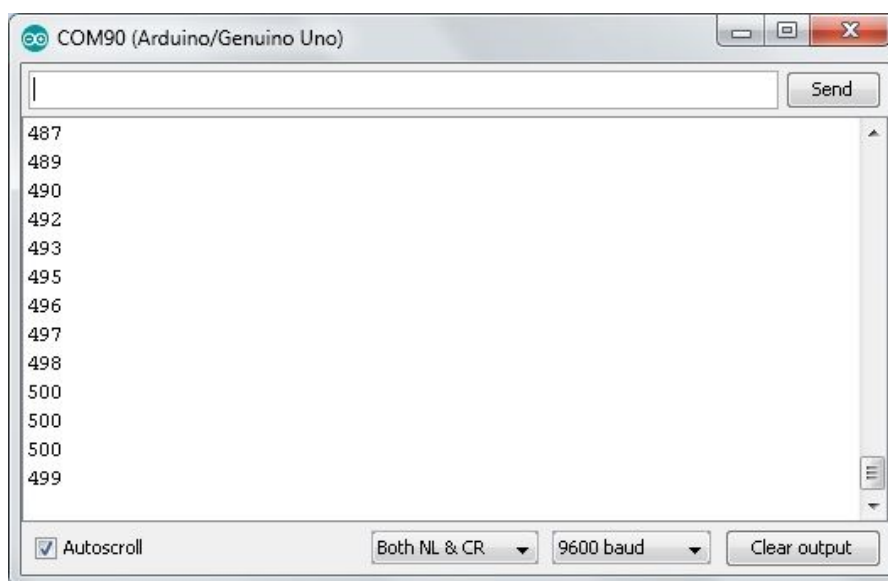


Рисунок 5.5 – Значення освітленості через програму спостереження

## Обладнання

Необхідні компоненти наведені у таблиці 5.1. Для самостійного виконання роботи написати програму дискретної роботи цієї схеми зі світлодіодами, з урахуванням методичних рекомендації та користуючись додатковими джерелами, які наведено в кінці роботи.

Таблиця 5.1 – Необхідні компоненти для виконання лабораторної роботи

Назва	Фото
Мікроконтролер « <i>Arduino</i> » та USB – кабель	
З'єднувальні проводи	
Фоторезистори	
Опори	
Плата прототипування	

## Завдання до роботи

1. Ознайомитись із теоретичним матеріалом.
2. Ознайомитись із схемою підключення компонентів до мікроконтролера.
3. Зібрати схему на платі прототипування за рисунком з використанням фоторезистора.



4. Обрати з таблиці 5.1 свій варіант за списком та реалізувати його програмно та апаратно для дискретного керування яскравістю світлодіода залежно від рівня зовнішньої освітленості.

5. Прописаний скетч протестувати в середовищі «*Arduino*» і перевірити на макеті.

6. Відповісти на контрольні питання до лабораторної роботи.

7. Оформити звіт та надати висновки, завантажити у форматі pdf файл у дистанційний курс.

### *Додаткові джерела інформації*

1. Інформаційні матеріали за темою: «Резистор». URL: <https://uk.wikipedia.org/wiki/Резистор> (дата звернення: 14.02.2024).

### *Контрольні запитання*

1. Дайте визначення поняттю «фоторезистор»?
2. Фоторезистор та резистор, які між ними відмінності?
3. Яке схематичне позначення фоторезистора?
4. На які піни мікроконтролера можна підключати фоторезистор?
5. Резистивний дільник напруги, які основні функції?
6. Відмінності дискретного від пропорційного керування?
7. Для чого потрібна функція «*map* ()» у скетчі?

## **Лабораторна робота 6**

### **Програма: менше світла – яскравіше світлодіод**

**Мета роботи:** дослідження вмикання світлодіода від роботи приймача світла. У разі зміни кількості світла, що падає на фоторезистор, буде змінюватися яскравість світлодіода.

### *Короткі теоретичні відомості*

У цій лабораторній роботі ми розглянемо підключення фоторезистора до плати, але так, що у разі зміни кількості світла, яке падає на фоторезистор, буде змінюватися яскравість світлодіода. Для цього нам знадобляться: фоторезистор (див. детальніше лабораторну роботу 5), *Arduino Uno* (Mega, Leonardo, Nano тощо), з'єднувальні дроти, резистори на 10 КОм, 220 (300) Ом, світлодіод. Приклад використаного резистора наведено на рисунку 6.1. Нижче поговоримо про існуючі опори різних номіналів.

Резистори для пайки доступні в різних номіналах опору, що вимірюється в омах ( $\Omega$ ). Вони використовуються в електронних схемах для обмеження потоку

струму або для поділу напруги. Ці резистори мають різні характеристики, такі як потужність, точність, температурний коефіцієнт опору тощо, що може впливати на їхнє використання в різних застосунках.

Резистори для використання з платформою «*Arduino*» мають ті ж характеристики, що й звичайні резистори, але їхні параметри можуть змінюватися залежно від конкретної задачі чи проєкту. Ось деякі загальні характеристики резисторів, які можуть бути використані з «*Arduino*»:

1. Опір (Resistance): резистори з різними значеннями опору (в омах) використовуються для обмеження потоку струму в електричних ланцюгах. Номінальне значення опору може бути від кількох омів до кількох мегаомів.

2. Точність (Accuracy): деякі застосунки можуть вимагати дуже точних резисторів, тоді як у інших випадках допустима менша точність.

3. Потужність (Power rating): це максимальна потужність, яку резистор може розігрівати без ризику пошкодження. Зазвичай вимірюється у ватах.

4. Температурний коефіцієнт опору (Temperature coefficient of resistance): цей параметр вказує на те, наскільки сильно змінюється опір резистора зі зміною температури.

5. Тип монтажу (Mounting type): резистори можуть бути випускатися в обох типах: для монтажу на поверхню (SMD) або для монтажу в отвір на печатній платі (THT).

При виборі резисторів для проєкту з «*Arduino*» важливо врахувати потрібне значення опору, потужність, а також можливі вимоги до точності та температурних умов. Для більшості проєктів важливо також врахувати сумісність з іншими компонентами та відповідність стандартам, що встановлені для конкретного застосування. Останнім часом імпортовані резистори часто маркуються за допомогою кольорових смуг на корпусі, які вказують на їхній номінальний опір і допуск. Кожна компанія-виробник може мати свою систему маркування, що іноді призводить до плутанини. Проте, загалом, існує певна стандартна система маркування, якої дотримуються більшість виробників [6].

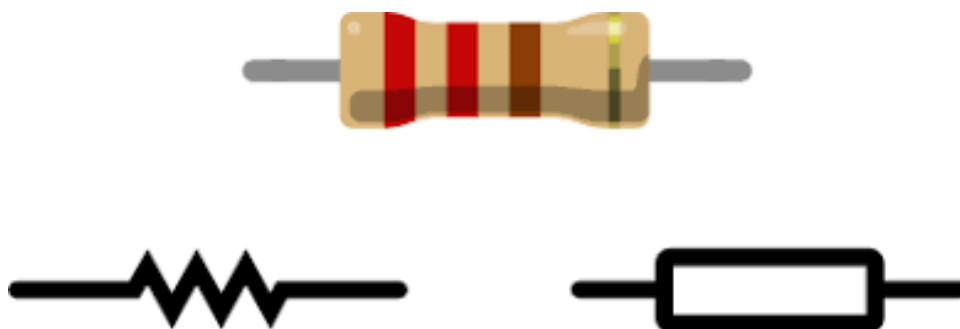


Рисунок 6.1 – Вигляд та умовне позначення резистора на електричних принципових схемах

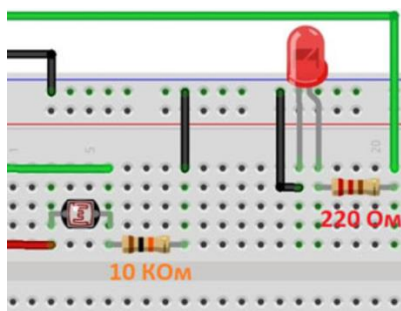


Рисунок 6.2 – Схема підключення резисторів

### Опис установки

**ЗАУВАЖЕННЯ.** Головна умова – не переплутати + і – у світлодіода (до чорного дроту приєднуємо – (катод) світлодіода!

Перш за все необхідно зібрати на платі прототипування схему, наведену на рисунку 6.3. У функції «setup» запускаємо обмін даними по порту «Serial» – тепер наша «Arduino» зможе виводити будь-які значення в цей порт, а ми зможемо їх бачити через монітор порту. У функції «loop» ми створюємо код, світлодіод зчитується з цифрового піна 13 «Arduino». Якщо прикрити фоторезистор рукою, то значення змінної не сильно суттєво зміниться. Однак ми бачимо, що світлодіод не реагує на значення сигналу з фоторезистора, якщо ми просто закриваємо його пальцем. Тільки якщо ми закриємо його чимось більш світлонепропускним (наприклад, ковпачком від ручки), тільки тоді він реагує на це, і він не змінює яскравість, а просто вмикається – вмикається.

Це відбувається через те, що ми підключили світлодіод до цифрового піна, що не дозволяє нам регулювати його яскравість. Давайте спробуємо зробити все те ж саме, проте підключивши світлодіод вже до ШІМ-піну (на «Arduino» ШІМ-піни позначені такою штучкою ~, їх всього 6: це піни № 3, 5, 6, 9, 10, 11). Для досягнення мети необхідно обрати відповідний пін, наприклад 11 (рис. 6.6), далі підправимо скетч, тоді отримаємо те, що було на меті (рис. 6.5).

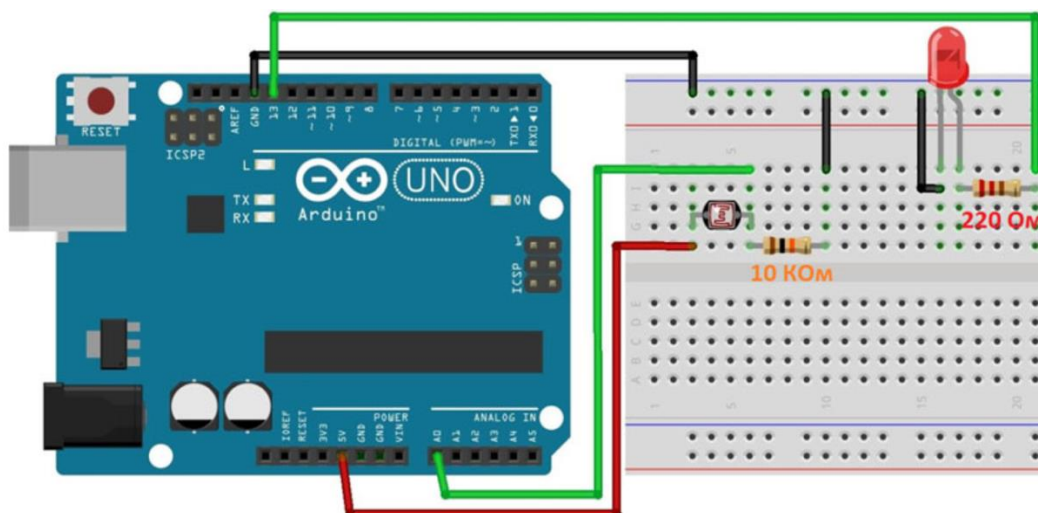


Рисунок 6.3 – Схема підключення фоторезистора до Arduino UNO

```
sketch_feb13a.ino
1  int led = 13;
2  int val;
3  int potpin = 0;
4  void setup()
5  {
6  | pinMode(led, OUTPUT) // put your setup code here, to run once:
7  }
8
9  void loop()
10 {
11 | val = analogRead(potpin);
12 | val = map(val, 0, 1023, 0, 255);
13 | analogWrite(led, val);
14 | delay(15); // put your main code here, to run repeatedly:
15 }
16
```

Ln 14, Col 58 × No board selected

Рисунок 6.4 – Програмний код варіант 1

```
sketch_feb13a.ino
1  int led = 11;
2  int val;
3  int potpin = 0;
4  void setup()
5  {
6  | pinMode(led, OUTPUT) // put your setup code here, to run once:
7  }
8
9  void loop()
10 {
11 | val = analogRead(potpin);
12 | val = map(val, 0, 1023, 0, 255);
13 | analogWrite(led, val);
14 | delay(15); // put your main code here, to run repeatedly:
15 }
16
```

Ln 1, Col 13 × No board selected

Рисунок 6.5 – Програмний код варіант 2

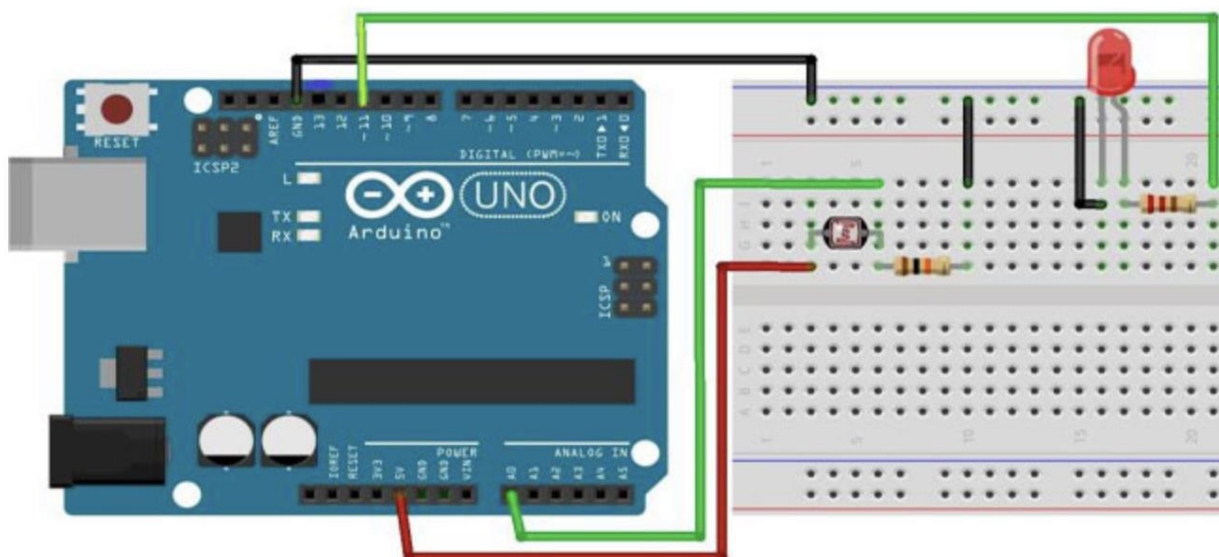


Рисунок 6.6 – Схема підключення фоторезистора до «*Arduino*» UNO змінена


*Обладнання*

Необхідні компоненти наведені у таблиці 6.2. Для самостійного виконання роботи написати програму дискретної роботи цієї схеми із трьома світлодіодами, з урахуванням методичних рекомендації та користуючись додатковими джерелами, які наведено в кінці роботи.

Таблиця 6.2 – Необхідні компоненти для виконання лабораторної роботи

Назва	Фото
1	2
Мікроконтролер « <i>Arduino</i> » та USB – кабель	
З'єднувальні проводи	
Фоторезистори	
Світлодіоди	
Опори	

## Продовження таблиці 6.2

1	2
Плата прототипування	

### *Завдання до роботи*

1. Ознайомитись із теоретичним матеріалом.
2. Ознайомитись із схемою підключення компонентів до мікроконтролера.
3. Зібрати схему на платі прототипування за рисунком з використанням фоторезистора за варіантом 2.
4. Реалізувати програмно та апаратно код для дискретного керування яскравістю трьох світлодіодів залежно від рівня зовнішньої освітленості.
5. Прописаний скетч протестувати в середовищі «*Arduino*» і перевірити на макеті.
6. Відповісти на контрольні запитання до лабораторної роботи.
7. Оформити звіт та надати висновки, завантажити у форматі pdf файл у дистанційний курс.

### *Контрольні запитання*

1. Плата прототипування. Дайте визначення.
2. Які переваги у використанні фоторезистора в освітлювальних установках?
3. Як під'єднується схематично фоторезистор до мікроконтролера?
4. Різниця між мікроконтролером та контролером?
5. Резистивний дільник напруги. Дайте визначення.
6. Відмінності цифрового та аналогового керування?
7. Для чого потрібна функція «*loop ()*» у скетчі?

## **3 РЕКОМЕНДАЦІЇ ЩОДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ**

Під час самостійного вивчення курсу студент повинен опрацювати такі теми згідно з робочою програмою, а також виконати тестові завдання в дистанційному курсі: **Модуль 1 Програмування мікроконтролерів.**

**Змістовий модуль 1** Методи оцінки перешкод в електромережі від роботи систем освітлення.

**Тема 1** Чинники виникнення перешкод в електромережі.

1. Показники якості електроенергії в електричних мережах. 2. Види перешкод в електромережі. 3. Захист від перешкод.

**Тема 2** Імпульсні та високочастотні перешкоди.

1. Основні типи спотворень якості електричної енергії за роботи освітлювальної установки. 2. Класифікація перешкод. 3. Відхилення напруги і частоти.

**Тема 3** Наявності спотворень якості електроенергії. Архітектура і характеристика мікроконтролерів.

1. Засоби вимірювання та покращення якості електричної енергії. 2. Спеціалізований мікроелектронний програмований прилад. 3. Історія розвитку мікроконтролерної техніки.

**Тема 4** Програмні засоби підтримки проектування інтелектуальної системи освітлення.

1. Вбудовані системи. 2. Середовища розробки програмного забезпечення для мікроконтролерів AVR. 3. Інтегроване середовище розробки програм AVR Studio. 4. Програмне забезпечення для програмування мікроконтролерів.

**Змістовий модуль 2** Основи програмування на C, C++ на ПК.

**Тема 5** Апаратні засоби підтримки мікроконтролерів AVR. 1. Внутрішньосхемні відлагоджувачі. 2. Засоби розробки для мікроконтролерів різних фірм. 3. Апаратні засоби підтримки проектування та відлагодження систем.

**Тема 6** Стандарти мов програмування C/C++1. Елементи мови C. Алфавіт мови. Константи. Ідентифікатори. Ключові слова. Коментарі. Оператори. 2. Типи даних мови програмування C. Функції. Директиви препроцесора. 3. Структура програми на C. 4. Засоби C++ не пов'язані безпосередньо з ООП. 5. Об'єктно-орієнтоване програмування (ООП).

**Тема 7** Програмування мікроконтролерів AVR1. Загальна інформація. 2. Програмування по послідовному каналу. 3. Паралельне програмування. 4. Програмування по інтерфейсу JTAG. 5. Самопрограмування мікроконтролерів набору «Mega».

**Тема 8** Ведення в систему вводу-виводу C/C++. 1. Файлові структури даних. 2. Базові положення системи вводу-виводу C/C++. 3. Стандартні об'єкти вводу-виводу. 4. Ієрархія класів вводу-виводу C++.

**Тема 9** Особливості мікроконтролера «Arduino». 1. «Arduino» – контролер системи освітлення. 2. Основні характеристики. 3. Периферійні пристрої.

4. Програмування мікроконтролера. 5. Цифровий ввід / вивід за допомогою плати «*Arduino*». 6. Алгоритм роботи пристрою. 7. Сучасні проєкти.

**Змістовий модуль 3** Програмування мікроконтролерів для керування світловими приладами.

**Тема 10** Алгоритм системи віддаленого управління освітленням. 1. Периферія, яка може бути присутня у мікроконтролерах. 2. Особливості системи. 3. Переваги. 4. Технічні рішення.

**Тема 11** Програмування мікроконтролера для підключення системи освітлення. 1. Програмований пристрій та схеми підключення освітлювальних приладів. 2. Розробка програмного забезпечення для пристрою.

**Тема 12** Тестування системи. 1. Симуляція та верифікація роботи пристрою. 2. Варіанти реалізованих проєктів.



## ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Дайте визначення поняттю «мікроконтролер».
2. Як відбувається програмування мікроконтролера?
3. Мова програмування C.
4. Відмінності мови C++ та мови асемблера.
5. Підтримуване обладнання для програмування.
6. Програмактор, плата розробки, електронні компоненти, що це?
7. Інтегроване середовище розробки (IDE).
8. Види сучасних мікроконтролерів.
9. «*Arduino*», дайте коротку характеристику.
10. Функції та специфікації різних видів мікроконтролерів.
11. Структурна схема системи управління мікроконтролером.
12. Периферійні пристрої.
13. Програмне забезпечення для програмування.
14. Давачі освітленості та фоторезистор.
15. Ефективний метод програмування мікроконтролера.
16. Електричний шум та запис даних на мікроконтролер.
17. Електромагнітні перешкоди (EMI).
18. Апаратне забезпечення для програмування.
19. Проблеми синхронізації.
20. Проблеми з пам'яттю.
21. Платформи розробки програмного коду.
22. Архітектура сучасних мікроконтролерів.
23. Мови програмування мікроконтролерів.
24. Система команд програмування та відлагодження.
25. Ефективні та функціональні електронні проекти.
26. Режими роботи освітлювальних установок з використанням мікроконтролерної техніки.
27. Шляхи зниження витрат електроенергії під час експлуатації освітлювальних установок різного функціонального призначення.

## СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Фізика і техніка світлодіодів : навч. посіб. / Л. А. Назаренко, А. І. Колесник ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2019. – 244 с.
2. Іоффе К. І. Системи керування світлотехнічними пристроями : конспект лекцій для магістрів денної і заочної форм навчання спеціальності 141 – Електроенергетика, електротехніка та електромеханіка (спеціалізація «Світлотехніка і джерела світла»)) / К. І. Іоффе, О. Л. Черкашина ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2018. – 57 с.
3. Програмування мікроконтролерів AVR : навч. посіб. / С. М. Цирульник, О. Д. Азаров, Л. В. Крупельницький, Т. І. Трояновська. – Вінниця : ВНТУ, 2018. – 111 с.
4. Назаренко Л. А. Світлотехнічні розрахунки : навч. посіб. / Л. А. Назаренко, Т. В. Можаровська, В. С. Чернець ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 142 с.
5. Карась В. І. Світлодіоди: фізика, технологія виготовлення, застосування : навч. посіб. / В. І. Карась, Л. А. Назаренко, І. В. Карась ; Харків. нац. академ. міськ. госп-ва. – Харків : ХНАМГ, 2012. – 323 с.
6. Методичні вказівки до виконання лабораторних робіт з дисципліни «Мікроелектронні та мікропроцесорні пристрої та системи» освітніх програм «Електричні та електронні апарати, Електромеханічне обладнання енергоємних виробництв» для студентів спеціальності 141 – Електроенергетика, електротехніка та електромеханіка всіх форм навчання / Укл. : М. О. Поляков. – Запоріжжя : Національний університет «Запорізька політехніка», 2022. – 35 с.
7. Blum J. Exploring Arduino: Tools and Techniques for Engineering Wizardry / J. Blum. – England: John Wiley & Sons, Inc., 2019. – 512 p.
8. Мікропроцесорні пристрої : навч. посіб. для студентів зі спец-ті «Електроніка» / Т. О. Терещенко, В. А. Годоренко, Л. М. Батрак, Ю. С. Ямненко. – Київ : Кафедра, 2017. – 244 с.
9. Маркування резисторів [Електрон. ресурс] : інформаційний сайт. – Електрон. текст. дані. – Режим доступу: <https://arduino.com.ua/ua/p1862979384-rezistor-330-025w.html>, вільний (дата звернення: 13.02.2024). – Назва з екрана.
10. Штучне зовнішнє освітлення : навч. посіб. / Л. А. Назаренко, К. І. Іоффе ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 88 с.
11. Назаренко Л. А. Фізичні основи джерел світла : навч. посіб. / Л. А. Назаренко. – Харків : ХНАМГ, 2009. – 206 с.

12. Суворова К. І. Джерела світла : навч. посіб. / К. І. Суворова, Л. Д. Гуракова ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2021. – 110 с.

13. Теорія автоматичного керування : конспект лекцій з курсу / П. П. Говоров та ін. – Харків : ХНУМГ, 2012. – 221 с.

#### Програмне забезпечення та Інтернет-ресурси:

14. Дистанційний курс ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ [Електрон. ресурс]. – Електрон. текст. дані. – Режим доступу: <https://dl.kname.edu.ua/course/index.php?categoryid=51>, вільний (дата звернення: 20.01.2024). – Назва з екрана.

15. Arduino – Software [Електрон. ресурс]. – Електрон. текст. дані. – Режим доступу: <http://arduino.cc/en/main/software>, вільний (дата звернення: 13.01.2024). – Назва з екрана.

16. Сайт фірми Atmel [Електрон. ресурс]. – Електрон. текст. дані. – Режим доступу: [www.atmel.com](http://www.atmel.com), вільний (дата звернення: 10.02.2024). – Назва з екрана.

*Електронне навчальне видання*

Методичні рекомендації  
до проведення лабораторних робіт, організації самостійної роботи  
з навчальної дисципліни

**«ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ»**

*(для здобувачів першого (бакалаврського) рівня вищої освіти  
денної форми навчання зі спеціальності 141 – Електроенергетика,  
електротехніка та електромеханіка,  
освітньо-професійна програма «Світлотехніка та дизайн світлового  
середовища»)*

Укладачі: **КОЛЕСНИК** Анастасія Ігорівна,  
**ПОЛЩУК** Валентина Миколаївна

Відповідальний за випуск *А. І. Колесник*  
Редактор *О. В. Михаленко*  
Комп'ютерне верстання *А. І. Колесник*

План 2023, поз. 204М

---

Підп. до друку 19.04.2024. Формат 60 × 84/16.  
Ум. друк. арк. 2,5

Видавець і виготовлювач:  
Харківський національний університет  
міського господарства імені О. М. Бекетова,  
вул. Маршала Бажанова, 17, Харків, 61002.  
Електронна адреса: office@kname.edu.ua  
Свідоцтво суб'єкта видавничої справи:  
ДК № 5328 від 11.04.2017.