

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили

Методична серія
Випуск № 484

Заснована в 2016 році

Є. С. Дарнапук

СЕНСОРИ, ПЕРЕТВОРЮВАЧІ

Методичні рекомендації

*Для практичних занять
з дисципліни для здобувачів вищої освіти
першого (бакалаврського) рівня вищої освіти
спеціальності F7 Комп'ютерна інженерія*



Миколаїв – 2026

УДК 681.586 (076)

Д 20

*Рекомендовано до друку засіданням ради факультету
комп'ютерних наук ЧНУ ім. Петра Могили
(протокол № 6 від 15 січня 2026 р.)*

Рецензенти:

Трунов О. М., д-р тех. наук, професор, професор кафедри
АКІТ ЧНУ ім. Петра Могили.

Бойко А. П., канд. техн. наук, доцент, доцент кафедри
дизайну, декан факультету комп'ютерних наук
ЧНУ ім. Петра Могили.

Дарнапук Є. С.

Д 20 Сенсори, перетворювачі : метод. рек. для практичних занять з
дисципліни для здобувачів вищої освіти першого (бакалаврсь-
кого) рівня вищої освіти спеціальності F7 Комп'ютерна інжене-
рія. Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2026. 48 с.
(Методична серія ; вип. 484).

Методичні рекомендації містять опис освітньої компоненти, план лекційних та семінарських занять, перелік питань до підсумкового контролю, методи навчання та контролю, критерії оцінювання знань, умінь та навичок здобувачів вищої освіти, а також рекомендовані джерела інформації. Методичні рекомендації розроблено з метою активізації самостійної роботи студентів та полегшення підготовки до аудиторних занять. Методичні рекомендації розроблено відповідно до робочої програми освітньої компоненти «Сенсори, перетворювачі» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Комп'ютерна інженерія», спеціальності F7 Комп'ютерна інженерія.

УДК 681.586 (076)

ISSN 1811-492X

© Дарнапук Є. С.,
© ЧНУ ім. Петра Могили, 2026

ЗМІСТ

ОПИС ОСВІТНЬОЇ КОМПОНЕНТИ	4
МЕТА, ЗАВДАННЯ ТА РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ	5
ПРОГРАМНІ КОМПЕТЕНЦІЇ	6
ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	7
ПЛАН ПІВГРУПОВИХ ЗАНЯТЬ	11
Лабораторна робота № 1. Ознайомлення з роботою сенсорів і підключенням їх до Arduino	11
Лабораторна робота № 2. Використання п'єзовипромінювача в системах сповіщення.....	18
Лабораторна робота № 3. Ознайомлення з роботою позиційних та MEMS-сенсорів	20
Лабораторна робота № 4. Використання температурних сенсорів у системах сповіщення.....	22
Лабораторна робота № 5. Робота зі світловими сенсорами	30
Лабораторна робота № 6. Ознайомлення з роботою сервомоторів	33
Лабораторна робота № 7. Використання SPDT реле	35
Лабораторна робота № 8. Використання перемикачів.....	38
КРИТЕРІЇ ОЦІНЮВАННЯ ТА ЗАСОБИ ДІАГНОСТИКИ РЕЗУЛЬТАТІВ НАВЧАННЯ.....	44
РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ	46

ОПИС ОСВІТНЬОЇ КОМПОНЕНТИ

Найменування показника	Характеристика дисципліни		
Найменування дисципліни	Сенсори, перетворювачі		
Галузь знань	12/F Інформаційні технології		
Спеціальність	123/F7 Комп'ютерна інженерія		
Спеціалізація	–		
Освітня програма	Комп'ютерна інженерія		
Рівень вищої освіти	Бакалавр		
Статус дисципліни	Вибіркова		
Курс навчання	3		
Навчальний рік	2025–2026 н. р.		
Номер(и) семестрів (триместрів):	Денна форма	Заочна форма	
	6	10	
Загальна кількість кредитів ЄКТС/годин	4 кредитів / 120 годин		
Структура курсу:	Денна форма	Заочна форма	
	– лекції	18	4
	– практичні (групові) заняття	-	-
	– лабораторні (півгрупові) заняття	36	2
	– годин самостійної роботи студентів	66	114
	Відсоток аудиторного навантаження	45 %	5 %
	Мова викладання	Українська	
Форма проміжного контролю			
Форма підсумкового контролю	Залік		

МЕТА, ЗАВДАННЯ ТА РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ

Метою вивчення дисципліни є формування системи знань та навичок роботи з апаратно-програмними модулями сенсорів для отримання даних, первинної їх обробки, а також з апаратно-програмними перетворювачами для конвертації даних в енергію та навпаки, перетворення аналогового сигналу в цифровий і т. д.

Завдання дисципліни:

- формування базових знань роботи з поширеними типами сенсорів та перетворювачів;
- проєктування та реалізація програмної частини апаратно-програмних комплексів з використанням наявної елементної бази;
- ознайомлення з існуючими комплексами, які використовують в своїй роботі сенсори, двигуни та реле;

Передумови вивчення дисципліни:

- Комп'ютерна логіка;
- Алгоритми та методи обчислень;
- Програмування;
- Архітектура комп'ютерів.

У результаті вивчення дисципліни студент

має знати:

- основні типи сенсорів, їх принцип роботи та використання;
- основні типи двигунів, їх структуру та принцип керування;
- методи пошуку документації по наявних компонентах;
- принцип програмування мікроконтролерів, використовуючи середовища розробки Arduino IDE або подібні;
- проєктування апаратно-програмних комплексів залежно від поставленої задачі.

має вміти:

- використовувати наявну компонентну базу для реалізації поставленого завдання;
- налаштовувати середовища розробки для реалізації апаратно-програмних комплексів;
- проєктувати і реалізовувати програмну частину комплексу, використовуючи наявні бібліотеки або створюючи свої;

ПРОГРАМНІ КОМПЕТЕНЦІЇ

Інтегральна компетентність

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов.

Загальні компетентності:

Z2 – Здатність вчитися і оволодівати сучасними знаннями.

Z3 – Здатність застосовувати знання у практичних ситуаціях

Спеціальні (фахові) компетентності:

P5 – Здатність використовувати засоби і системи автоматизованого проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо

P7 – Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності

P8 – Готовність брати участь у роботах з впровадження комп'ютерних систем та мереж, введення їх до експлуатації на об'єктах різного призначення.

Програмні результати навчання:

N3 – Знати новітні технології в галузі комп'ютерної інженерії.

N4 – Знати та розуміти вплив технічних рішень в суспільному, економічному, соціальному і екологічному контексті.

N8 – Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

N11 – Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

N12 – Вміти ефективно працювати як індивідуально, так і у складі команди.

N13 – Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.

N19 – Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Сенсори та перетворювачі

Розглядається поняття сенсорів і перетворювачів як засобів перетворення фізичних величин у сигнал, зручний для подальшої обробки, та наводяться їхні базові визначення й роль у системах вводу/виводу. Висвітлюються типові приклади сенсорів за вимірюваними параметрами (світло, температура, тиск/сила, позиція, швидкість, звук) і відповідні пристрої виводу, а також пояснюються відмінності між активними та пасивними сенсорами. Демонструються приклади формування аналогових і цифрових сигналів на основі термопари та світлового сенсора, вводяться базові уявлення про фільтрацію сигналів (фільтри низьких і високих частот) і детально розглядається ультразвуковий далекомір HC-SR04: його характеристики, принцип роботи та схема підключення до Arduino Uno.

Тема 2. Позиційні сенсори

Розглядаються позиційні сенсори як засоби виявлення руху та визначення відносного положення об'єкта відносно контрольної точки, а також їх використання для фіксації наявності/відсутності предмета. Детально пояснюється принцип роботи потенціометра як змінного резистора з рухомим контактом (дільника напруги), наводяться типові приклади застосування та демонструється схема підключення і приклад зчитування аналогового сигналу мікроконтролером. Далі аналізуються індуктивні позиційні сенсори (зміна повного опору котушки при переміщенні елементів магнітного кола) та описується метод LVDT для вимірювання лінійних переміщень із акцентом на безконтактність, надійність і роботу у складних умовах. Окремо узагальнюються підходи до індуктивних датчиків відстані та принципи роботи обертових кодерів як засоби реєстрації кутового положення/швидкості, що застосовуються в системах керування, робототехніці та автоматичі.

Тема 3. Температурні сенсори

Розглядаються температурні сенсори як засоби вимірювання температури середовища, що поділяються на контактні (потребують фізичного контакту з об'єктом) і безконтактні (використовують конвекцію та випромінювання для моніторингу змін температури). Наводяться принципи роботи та приклади застосування біметалевого термостата як простого механічного перетворювача для підтримання заданих температурних режимів. Далі пояснюється будова й властивості терморезисторів/термісторів, у яких опір змінюється залежно від температури, та демонструються типові варіанти їх використання в електронних системах. Окремий блок присвячено термопарам: розкривається

термоелектричний ефект, структура чутливого елемента, подаються поширені типи термопар і їх робочі діапазони, а також обґрунтовується потреба у використанні підсилювача термопари для коректного зчитування сигналу вимірювальною системою.

Тема 4. Світлові сенсори

Досліджуються та розглядаються світлові сенсори як фотоелектричні пристрої, що перетворюють енергію світла (фотони) в електричний сигнал, та наводиться їх поділ на дві базові групи: фотовольтаїчні (генерують електроенергію при освітленні) і фотозалежні (змінюють електричні властивості залежно від рівня освітленості). Пояснюється класифікація світлових сенсорів (фотоємісійних, фотопровідних, фотовольтаїчних, фотоз'єднувальних) і детально розкривається принцип фотопровідних комірок та фоторезистора на основі ефекту фотопровідності зі зміною опору. Окремо висвітлюються фотоз'єднувальні пристрої: принцип роботи фотодіода, його конструктивні особливості, характеристики та приклад побудови фотодіодного підсилювача, а також фототранзистор як елемент перетворення світлових сигналів в електричні з одночасним підсиленням. Також розглядаються фотовольтаїчні комірки (сонячні елементи) як перетворювачі світлової енергії в електричну та аналізуються їхні основні характеристики.

Тема 5. Електричне реле

Розкривається призначення електричних реле як комутаційних пристроїв, що забезпечують керування навантаженням за допомогою малопотужного керувального сигналу та реалізують гальванічну розв'язку між колами керування і силовою частиною. Детально пояснюється принцип роботи електромеханічного реле: будова котушки й магнітної системи, перемикання контактних груп (NO/NC/COM), типові схеми вмикання та особливості експлуатації (контактний знос, іскріння, час спрацювання, потреба в захисному діоді для котушки постійного струму). Далі розглядається твердотільне реле (SSR) як безконтактна альтернатива, що виконує комутацію напівпровідниковими елементами; аналізуються його переваги (швидкодія, відсутність механічного зносу, безшумність) і обмеження (витік струму, тепловиділення, вимоги до радіатора, особливості роботи з AC/DC навантаженнями). На прикладах порівнюються критерії вибору типу реле для практичних задач автоматизації, робототехніки та IoT-систем із мікроконтролерами.

Тема 6. Двигуни постійного струму

Пояснюється призначення двигунів постійного струму як електромеханічних перетворювачів, що трансформують електричну енергію у механічне обертання або поступальний рух, і наводяться основні залежності між напругою, струмом, швидкістю та обертовим моментом.

Розглядаються лінійні соленоїдні приводи як різновид актуаторів для короткохідного лінійного переміщення, після чого подаються базова модель DC-двигуна та її ключові вузли (статор, ротор/якір, магнітне поле). Деталізується принцип роботи щіткового двигуна з комутатором і щітками, аналізуються його конструктивні варіанти (з постійними магнітами та з електромагнітами) і типові властивості у прикладних системах. Далі порівнюється безщітковий DC-двигун (з внутрішнім/зовнішнім ротором), пояснюється роль електронного контролера в комутації та наводяться приклади застосувань у побутових пристроях, дронах і промисловості. Окремий блок присвячений сервомоторам як системам позиційного керування (двигун + редуктор + потенціометр + плата керування) із поясненням керування за допомогою ШІМ-імпульсів і залежності кута повороту від тривалості імпульсу. Завершується лекція оглядом принципів керування моторами постійного струму (зміна швидкості/напрямку, узгодження з контролером) та розглядом крокового двигуна як актуатора, що перетворює послідовність цифрових імпульсів у дискретні «кроки» обертання, із поясненням будови та механізму формування кроку.

Тема 7. Загальна інформація про перетворювачі. Перетворювачі звуку.

Подано базові поняття про перетворювачі як пристрої, що здійснюють перетворення енергії та сигналів між різними фізичними формами, з акцентом на акустичні перетворювачі, які працюють у двох напрямках: перетворення звуку в електричний сигнал та електричного сигналу в звук. Пояснюється, що таке звук і акустичні хвилі, їхні ключові параметри (частота, довжина хвилі, швидкість поширення), а також вводиться визначення акустичного перетворювача як елемента, що генерує акустичні коливання або формує електричний вихідний сигнал під дією звукового тиску. Далі розглядаються вхідні акустичні перетворювачі на прикладі мікрофона, аналізуються принципи роботи та основні типи мікрофонів (вугільні, п'єзоелектричні, магнітоелектричні, конденсаторні) і наводиться приклад конструкції магнітоелектричного мікрофона. Завершується лекція оглядом вихідних акустичних перетворювачів – гучномовців, їх типів (електродинамічних, електростатичних, п'єзоелектричних, електромагнітних, плазменних) та структурних схем електро-механічного й електростатичного гучномовця.

Тема 8. Вхідна інтерфейсизація

Розкривається сутність вхідної інтерфейсизації (англ. interfacing) як способу підключення комп'ютера або мікроконтролера до зовнішніх пристроїв із узгодженням напруги та струму між різними електронними вузлами. Пояснюються принципи роботи найпоширеніших елементів

введення – механічних перемикачів (кнопки, геркони, поворотні та ключові перемикачі), зокрема логіка станів «розімкнено/замкнено» та приклади схем підключення. Окремо аналізується проблема брязкоту контактів (англ. contact bounce) і демонструються типові методи її усунення: RC-ланцюг та апаратна «debounce»-обробка на логічних елементах NAND/NOR. Далі вводиться поняття оптрона (оптоперемикача) як засобу оптичного розділення та безпечного узгодження сигналів, а також розглядаються приклади оптичних рішень, зокрема рефлексивний оптичний перемикач. Завершується лекція прикладами практичної інтерфейсації світлочутливих елементів – фотодіодів і LDR-фоторезисторів – із поясненням типових схем включення для формування вимірюваного електричного сигналу.

Тема 9. Терморезистори

Висвітлюється принцип роботи терморезистора як температурного сенсора, у якому електричний опір змінюється залежно від температури, та пояснюється, чому такі елементи широко застосовують у вимірювальних і керуючих системах. Детально розглядаються NTC-терморезистори (англ. Negative Temperature Coefficient) – їхня характерна залежність «температура–опір», типові діапазони використання та особливості підключення у схемах (наприклад, як плеча подільника напруги для зчитування АЦП мікроконтролера). Окремий блок присвячено прикладам застосування терморезисторів: контроль температури в побутовій техніці й електроніці, термозахист і обмеження струму при пускових режимах, температурна компенсація параметрів схем, а також використання у системах моніторингу та автоматики, де важливі простота, низька вартість і достатня точність вимірювання.

ПЛАН ПІВГРУПОВИХ ЗАНЯТЬ

Лабораторна робота № 1. Ознайомлення з роботою сенсорів і підключенням їх до Arduino

Мета: здобуття навичок роботи з платою мікроконтролерів Arduino та підключення сенсорів до неї.

Завдання:

1. встановіть середовище розробки Arduino IDE;
2. підключіть плату мікроконтролерів Arduino Uno до комп'ютера за допомогою USB кабелю;
3. переконайтеся в коректності підключення пристроїв та їх розпізнавання комп'ютером. У випадку, якщо Arduino IDE не може знайти вашу плату – встановіть драйвери до неї;
4. налаштуйте Arduino IDE для роботи з вашою платою, обравши потрібний COM-порт;
5. виконайте завдання згідно з варіантом і сформуєте звіт, щодо виконаної роботи, що буде містити:
 - завдання згідно варіанту;
 - зображення роботи модулю (фото та скріншоти);
 - опис алгоритму роботи комплексу;
 - код програми.

Варіанти:

№	Завдання згідно варіанту:
1	<ol style="list-style-type: none">1. Підключити до плати Arduino Uno ультразвуковий далекомір HC-SR04.2. Під'єднати до плати управління RGB-світлодіод.3. Написати програму, яка змінює колір RGB-світлодіода залежно від зміни відстані між далекоміром та предметом. Шкала зміни кольорів в залежності від відстані: 0–0,5 м - #6E6EFF 0,5–1 м - # 0000FF 1–1,5 м - #00FF66 1,5–2 м - # 00FF00 2–2,5 м - # C8FF00 2,5–3 м - # FF6400 3–3,5 м - # FF3200 3,5–4 м - # FF0000

2	<ol style="list-style-type: none"> 1. Підключити до плати Arduino Uno ультразвуковий далекомір HC-SR04. 2. Під'єднати до плати управління RGB-світлодіод. 3. Написати програму, яка змінює колір RGB-світлодіода з зеленого на червоний, якщо далекомір зафіксував різку зміну відстані в сторону зменшення, яка відрізняється від відстані, вимірної на початку роботи пристрою, більше ніж на 50 см. У випадку, якщо перешкода прибрана та/або відстань від сенсора HC-SR04 до певного предмета стає приблизно такою ж, як і на початку роботи програми, то колір світлодіода повинен стати зеленим.
3	<ol style="list-style-type: none"> 1. Підключити до плати Arduino Uno ультразвуковий далекомір HC-SR04. 2. Підключити до плати управління зуммер. 3. Написати програму, яка видає звук на зуммер, якщо далекомір зафіксував різку зміну відстані в сторону зменшення, яка відрізняється від відстані, вимірної на початку роботи пристрою, більше ніж на 50 см. В консоль повинно виводитися повідомлення про кількість таких змін відстані та теперішню відстань від далекоміра до певного предмета. У випадку, якщо перешкода прибрана та/або відстань від сенсора HC-SR04 до певного предмета стає приблизно такою ж, як і на початку роботи програми – зумер повинен перестати пищати.

Інформація щодо виконання лабораторної роботи:

Дана практична робота (як і всі наступні) може бути виконана на базі абсолютно різних контролерів, які сумісні з Arduino IDE (рис. 1) або Arduino App Lab (рис. 2). Звісно, в варіантах завдання вказано саме використання Arduino Uno, але використання ESP-8266/ESP-12/ESP-32/Arduino UNO Q та архітектурно подібних плат як плат управління не заборонене. Використання Raspberry PI або інших одноплатних комп'ютерів також не заборонене, але лише за умови використання АЦП та ЦАП – не всі сенсори, які використовуються під час виконання лабораторних робіт, є цифровими. Під час виконання роботи необхідно забезпечити правильне живлення модулів та спільну «землю» (GND) між усіма компонентами.

Найбільш популярним та простим середовищем розробки є Arduino IDE, яке, незважаючи на мінімалістичний дизайн, є достатньо потужним інструментом для розробки проектів з використанням плат сімейства

Arduino/Arduino-сумісних, а також додаткових мікроконтролерів, підтримку яких можна додати, інсталивавши потрібні бібліотеки.

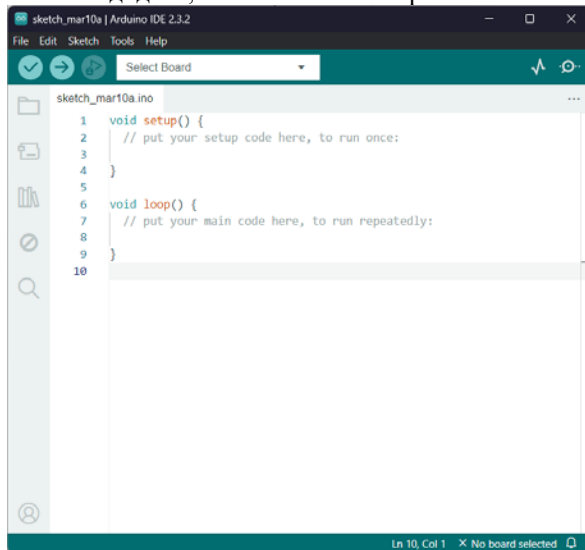


Рисунок 1 – Головне вікно Arduino IDE

Arduino Lab IDE (або Arduino Cloud Editor) – середовище, яке дозволяє працювати зі скетчами у форматі «проєкту» з кількома файлами (наприклад, .ino, .h, .cpp, README), компілювати та завантажувати прошивки на плату з відображенням повідомлень у вбудованій консолі. Це зручно для робіт, де потрібні бібліотеки, приклади, або структурований код, а також для студентів, які виконують завдання на різних комп'ютерах.

Короткий порядок роботи з Arduino LAB IDE:

1. відкрити Arduino Lab IDE/Editor та створити або відкрити проєкт;
2. обрати плату (наприклад, Arduino UNO) і порт/підключення;
3. додати потрібні бібліотеки (якщо проєкт їх використовує);
4. натиснути **Run/Upload** для компіляції та прошивання;
5. перейти в **Console/Serial** (якщо є) для перегляду повідомлень і результатів вимірювання.

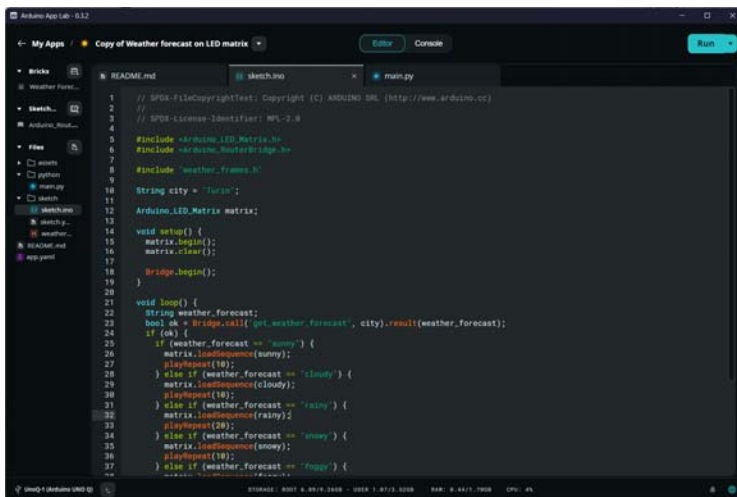


Рисунок 2 – Вікно проєкту в Arduino App Lab

Arduino IDE за замовчуванням містить підтримку базових плат Arduino, однак для роботи з ESP8266 (ESP-12), ESP32 та деякими іншими контролерами необхідно встановити додаткові «ядра» плат через **Boards Manager**:

1. Відкрийте меню **File** → **Preferences**.

2. У полі **Additional Boards Manager URLs** додайте посилання на пакети плат (можна вставити одне або кілька, розділяючи їх комами або з нового рядка) і натисніть **OK**.

Для ESP8266 використовуйте URL:

https://arduino.esp8266.com/stable/package_esp8266com_index.json

Для ESP32 використовуйте URL:

https://espressif.github.io/arduino-esp32/package_esp32_index.json

3. Далі перейдіть до **Tools** → **Board** → **Boards Manager...**

4. У пошуку введіть назву платформи й встановіть відповідний пакет:

- для ESP8266: «**esp8266 by ESP8266 Community**» → Install;
- для ESP32: «**esp32 by Espressif Systems**» → Install.

5. Після встановлення виберіть потрібну плату в **Tools** → **Board**, наприклад:

- ESP8266: *NodeMCU 1.0 (ESP-12E Module)* або *Generic ESP8266 Module*;

- ESP32: *ESP32 Dev Module* (або модель вашої плати).

6. Підключіть плату до ПК, у **Tools** → **Port** оберіть відповідний COM-порт, після чого можна компілювати та завантажувати скетч.

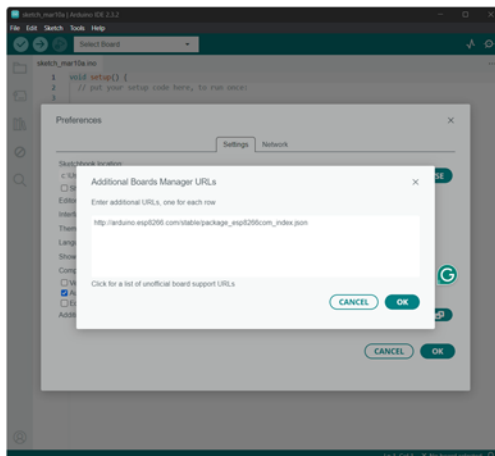


Рисунок 3 – Додавання бібліотек для роботи з платами, що не входять до стандартного списку підтримуваних Arduino IDE

Для перевірки роботи плати та підготовки робочого середовища до роботи з платою Arduino виконайте наступні дії:

1. Встановіть Arduino IDE.
2. Під'єднайте плату Arduino Uno до комп'ютера через USB-кабель.
3. У меню **Tools** → **Board** виберіть модель плати, а в **Tools** → **Port** – відповідний COM-порт.

4. Для перевірки зв'язку завантажте тестовий приклад **File** → **Examples** → **01.Basics** → **Blink**. Якщо скетч не завантажується:

- перевірте кабель (потрібен «data-cable», а не лише для заряджання);
- перевірте COM-порт;
- встановіть драйвер USB-UART (найчастіше CH340/CP210x – залежить від плати).

Примітка (рівні напруги): Arduino Uno працює з логікою **5 В**, а ESP8266/ESP32/Raspberry Pi – **3,3 В**. Не подавайте **5 В** на входи **3,3 В**-логіки. Для сумісної роботи інколи потрібні дільники напруги/level shifter (особливо для ліній ЕСНО в ультразвукових датчиках, якщо вони видають **5 В**).

Етапи з вибором порту та плати, а також процес компіляції та вивантаження скетчу на плату наведено на рис. 4 та рис. 5 відповідно.

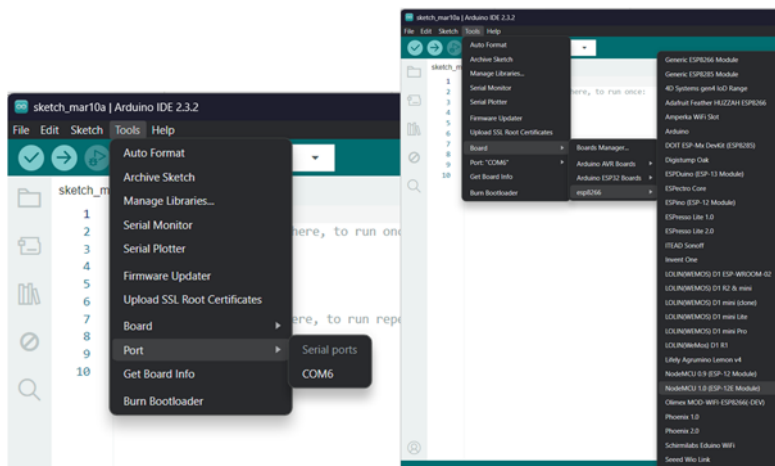


Рисунок 4 – Процес налаштування порту та вибору плати для компіляції та вивантаження скетчу проекту на плату

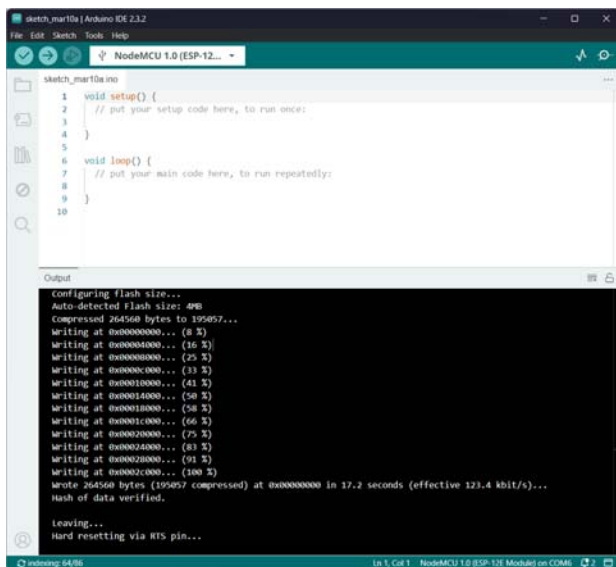


Рисунок 5 – Процес компіляції та вивантаження скетчу до плати

HC-SR04 – ультразвуковий датчик відстані, що вимірює дистанцію до об'єкта за принципом ехолокації. Після подачі короткого імпульсу на вхід TRIG модуль випромінює ультразвук (близько 40 кГц) і формує на виході ECHO імпульс, тривалість якого відповідає часу проходження хвилі до перешкоди та назад; за цією тривалістю обчислюють відстань. Датчик має 4 виводи (VCC, TRIG, ECHO, GND), простий у підключенні до мікроконтролерів і часто використовується в навчальних роботах, робототехніці та системах контролю наближення/перешкод. Типовий робочий діапазон – приблизно 2–400 см, а точність за сприятливих умов становить близько кількох міліметрів (залежить від поверхні, кута та шумів). На рис. 6 показано схему підключення сенсора HR-SR04.

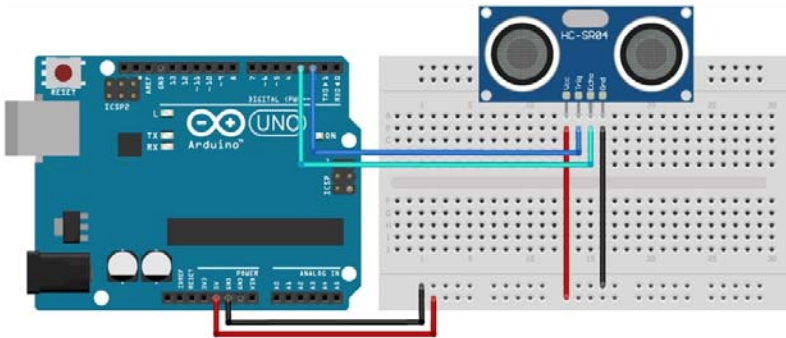


Рисунок 6 – Схема підключення сенсора HR-SR04 до Arduino UNO
Приклад коду програми для роботи з сенсором наведено нижче. Елементи цього коду ви можете використовувати під час виконання цієї лабораторної роботи.

```
#define trigPin 2
#define echoPin 3

// Define variables:
long duration;
int distance;

void setup() {
  // Define inputs and outputs:
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  //Begin Serial communication at a baudrate of 9600:
  Serial.begin(9600);
}

void loop() {
  // Clear the trigPin by setting it LOW:
```

```
digitalWrite(trigPin, LOW);
delayMicroseconds(5);

// Trigger the sensor by setting the trigPin high for 10
microseconds:
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Read the echoPin, pulseIn() returns the duration (length
of the pulse) in microseconds:
duration = pulseIn(echoPin, HIGH);
// Calculate the distance:
distance = duration * 0.034 / 2;

// Print the distance on the Serial Monitor (Ctrl+Shift+M):
Serial.print("Distance = ");
Serial.print(distance);
Serial.println(" cm");

delay(50);
}
```

Лабораторна робота № 2. Використання п'єзовипромінювача в системах сповіщення

Мета: здобуття навичок моделювання та створення проєктів на базі п'єзовипромінювача.

Завдання:

1. побудуйте схему пристрою залежно від вашого варіанту;
2. напишіть код програми для вашої схеми та завантажте прошивку

на плату.

У звіті має бути наведено фото схеми, код та результат виводу роботи плати в Serial Monitor. У випадку відсутності можливості зібрати схему використовуйте симулятор Autodesk Tinkercad.

Варіанти:

1. Пристрій, який буде оповіщати з використанням зумера про рух, який виявив PIR-сенсор. В Serial Monitor повинні виводитися сповіщення про початок руху та значення відстані, отримані сенсором HR-SR04. Частота звуку, який подається на зумер, – довільна. Приклад схеми наведено на рис. 7.

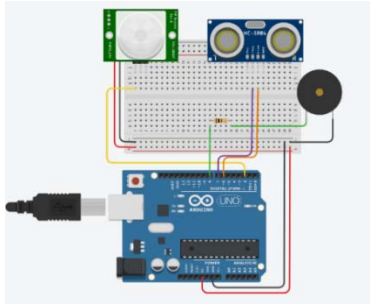


Рисунок 7 – Схема пристрою для варіанту 1. Вона включає в себе Arduino UNO, макетну плату, PIR-сенсор, ультразвуковий далекомір HR-SR04, резистор на 100 Ом та зумер

2. Пристрій, який буде оповіщати з використанням зумера про загазованість приміщення або про перевищення температури. У Serial Monitor повинні постійно виводитися значення рівня загазованості та температури. Якщо значення температури, отриманого сенсором TMP36, буде перевищувати 100 градусів – на зумер має подаватися сигнал. Якщо значення на аналоговому піні, до якого під'єднано газовий сенсор, буде перевищувати 500, то на зумер так само має подаватися сигнал. Частота звуку, який подається на зумер, – довільна. Приклад схеми наведено на рис. 8.

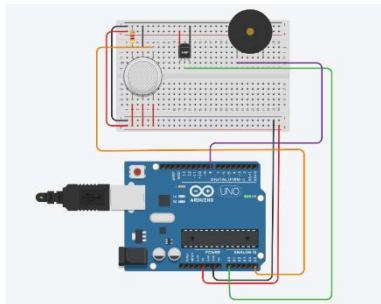


Рисунок 8 – Схема пристрою для варіанту 2. Вона включає в себе Arduino UNO, макетну плату, газовий сенсор MQ-2, температурний сенсор TMP36, резистор на 4,7 кОм та зумер

Інформація щодо виконання лабораторної роботи:

На даному етапі вам потрібно створити два скрипти – один скрипт буде виконувати певну задачу на стороні Arduino і надсилати на серійний порт потрібну «сиру» інформацію, другий скрипт повинен бути запущений на стороні комп'ютера, до якого підключена плата Arduino.

Дані повинні передаватися та прийматися за весь період часу підключення плати до комп'ютера.

Як варіант ви можете використовувати модуль *pyserial* в *python*, встановивши його через *pip*:

```
pip install pyserial --user
```

Нижче наведено приклад як отримати дані і вивести в консоль з використанням скрипту на *python*:

```
import serial
ser_input = serial.Serial('COM4', 9600)
b = ser_input.readline()
print(b)
ser_input.close()
```

Не забувайте, що основне тіло скрипту повинно бути додано в нескінченний цикл і не забувайте перевіряти наявність з'єднання між вашою платою та комп'ютером.

Лабораторна робота № 3. Ознайомлення з роботою позиційних та MEMS-сенсорів

Мета: здобуття навичок отримання первинної інформації з плат управління Arduino та їх обробки на стороні комп'ютера.

Завдання:

1. підключіть плату мікроконтролерів Arduino Uno до комп'ютера за допомогою USB-кабелю;
2. переконайтеся в коректності підключення пристроїв та їх розпізнавання комп'ютером. У випадку, якщо Arduino IDE не може знайти вашу плату – встановіть драйвери до неї;
3. налаштуйте Arduino IDE для роботи з вашою платою, обравши потрібний COM-порт;
4. виконайте завдання згідно з варіантом.

Варіанти:

№	Завдання згідно варіанту:
1	<ol style="list-style-type: none"> 1. Підключити до плати Arduino Uno трьохпозиційний акселерометр GY-61; 2. Під'єднати до плати управління RGB-світлодіод; 3. Написати програму, яка змінює колір RGB-світлодіода залежно від зміни координат акселерометра. Кожна координата (x, y, z) відповідає одному кольору світлодіода (RGB);

	4. Написати скрипт, який буде отримувати з COM-порту дані X, Y, Z з акселерометра і записувати дані координат в окремий CSV-файл.
2	<ol style="list-style-type: none"> 1. Підключити до плати Arduino Uno трьохпозиційний акселерометр GY-521; 2. Під'єднати до плати управління RGB-модуль CJMCU-2812-8; 3. Написати програму, яка змінює колір RGB-світлодіодів залежно від зміни координат (x,y,z). Кількість працюючих світлодіодів змінюється кожні 5 секунд. Початкова кількість працюючих світлодіодів – 1. Якщо кількість працюючих світлодіодів буде 8, лічильник скидається до 1; 4. Написати скрипт, який буде отримувати з COM-порту дані (x,y,z) з акселерометра і записувати дані кожної з координат в окремий текстовий файл.

Інформація щодо виконання лабораторної роботи:

На даному етапі вам потрібно створити два скрипти – один скрипт буде виконувати певну задачу на стороні Arduino і надсилати на серійний порт потрібну «сиру» інформацію, другий скрипт повинен бути запущений на стороні комп'ютера, до якого підключена плата Arduino. Дані повинні передаватися та прийматися за весь період часу підключення плати до комп'ютера.

Як варіант ви можете використовувати модуль *pyserial* в *python*, встановивши його через *pip*:

```
pip install pyserial --user
```

Нижче наведено приклад як отримати дані і вивести в консоль з використанням скрипту на *python*:

```
import serial
ser_input = serial.Serial('COM4', 9600)
b = ser_input.readline()
print(b)
ser_input.close()
```

Не забувайте, що основне тіло скрипту повинне бути додане в нескінченний цикл і не забувайте перевіряти наявність з'єднання між вашою платою та комп'ютером.

Схеми підключення акселерометра GY-61 до Arduino Uno наведено на рис. 9.

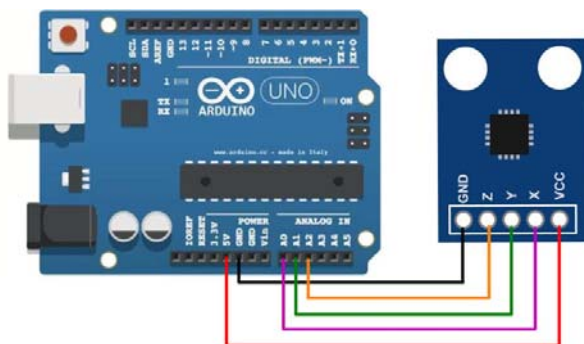


Рисунок 9 – Схема підключення акселерометру GY-61 до Arduino UNO

Схеми підключення акселерометру GY-521 до Arduino Uno наведено на рис. 10.

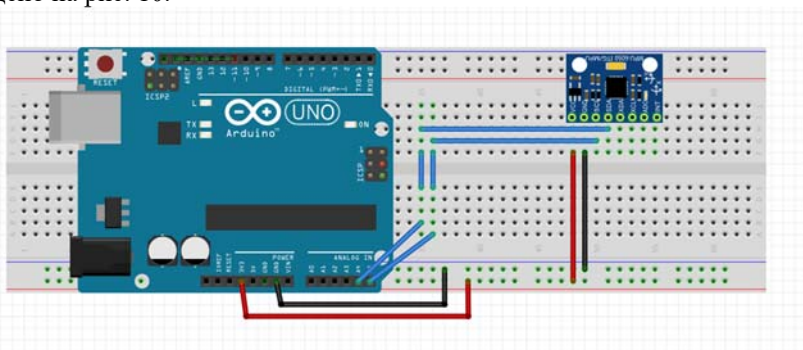


Рисунок 10 – Схема підключення акселерометру GY-521 до Arduino UNO

У звіті зафіксуйте приклад щонайменше **100 послідовних вимірів** (виведення у консоль або збереження у файл) та коротко поясніть, які саме поля/формат «сирих» даних передаються з Arduino (наприклад, три осі X/Y/Z або X/Y/Z + температура). За потреби додайте просту перевірку коректності прийому (лічильник пакетів або контрольний роздільник у рядку), щоб переконатися, що під час передачі дані не втрачаються.

Лабораторна робота № 4. Використання температурних сенсорів у системах сповіщення

Мета: здобуття навичок використання даних, отриманих з сенсорів, в системах сповіщень.

Завдання:

1. підключіть плату мікроконтролерів Arduino Uno до комп'ютера за допомогою USB кабелю;
2. переконайтеся в коректності підключення пристроїв та їх розпізнавання комп'ютером. У випадку, якщо Arduino IDE не може знайти вашу плату – встановіть драйвери до неї;
3. налаштуйте Arduino IDE для роботи з вашою платою, обравши потрібний COM-порт;
4. підключіть температурний сенсор згідно вашого варіанту до Arduino UNO;
5. підключіть сенсор MQ-9 до Arduino UNO;
6. реалізуйте програму на стороні Arduino, яка буде виводити на Serial Monitor показники з сенсорів, які перевищують задане пікове значення;
7. реалізуйте програму на стороні комп'ютера, до якого підключено плату Arduino, яка буде записувати в log файл значення з сенсорів і у випадку перевищення робити специфічний запис в цей файл з приміткою «WARNING».

Структура запису має бути приблизно така:

<дата> <час> Температура: <значення температури> Рівень CO: <значення концентрації угарного газу>

8. у випадку перевищень значень, вказаних у варіанті, зумер повинен подати звуковий сигнал.

Варіанти:

№	Завдання згідно варіанту:
1	Сенсор: DHT22 Пікова температура: 33 °C Пікове значення CO: 3 %
2	Сенсор: DS18B20 Пікова температура: 40 °C Пікове значення CO: 2,5 %
3	Сенсор: KY-028 Пікова температура: 20 °C Пікове значення CO: 3,1 %

Інформація щодо виконання лабораторної роботи:

В даній роботі вам доведеться використати ще раз напрацювання з попередньої роботи, а саме взаємодію Arduino з комп'ютером через

СОМ-порт для отримання даних з сенсорів. Головною задачею є реалізація логування даних, отриманих з сенсорів. В даному випадку будуть використані температурні сенсори різного виду та сенсори вуглекислого газу MQ-9. Сповіщення повинні надсилатися у випадку перевищення певного значення температури або газу в навколишньому середовищі. Пікові значення температури, тип температурного сенсора та газу залежать від варіанту завдання.

Для отримання даних через СОМ порт в python ви можете використати *pyserial* модуль:

```
pip install pyserial --user
```

Не забувайте, що основне тіло скрипту повинне бути закинуто в нескінченний цикл і не забувайте перевіряти наявність з'єднання між вашою платою та комп'ютером.

MQ-9 – напівпровідниковий газовий сенсор (серія MQ) на основі SnO₂ із вбудованим нагрівачем, який використовують для виявлення монооксиду вуглецю (CO) та горючих газів (зокрема метану/пропану/LPG). Принцип роботи такий: нагрітий чутливий елемент змінює свій електричний опір при взаємодії з газами; ці зміни перетворюються на аналоговий вихідний сигнал (зчитується АЦП), а у багатьох модулів також є цифровий вихід через компаратор із регульованим порогом (підлаштування потенціометром на платі модуля). Схема підключення MQ-9 до Arduino наведено на рис. 11:

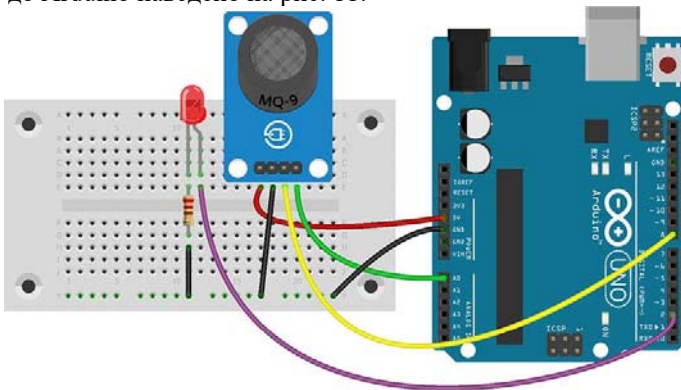


Рисунок 11 – Підключення сенсору MQ-9 до Arduino UNO

Приклад коду для взаємодії з сенсором MQ-9 наведено нижче:

```
const int analogPin = A0; // Connect AOUT pin of MQ-9 to  
Arduino analog pin A0  
const int digitalPin = 2; // Connect DOUT pin of MQ-9 to  
Arduino digital pin 2
```

```
void setup() {
  Serial.begin(9600); // Initialize serial communication at
  9600 baud
  pinMode(digitalPin, INPUT); // Set DOUT pin as input
}

void loop() {
  int analogValue = analogRead(analogPin); // Read analog
  value from AOUT
  int digitalValue = digitalRead(digitalPin); // Read digital
  value from DOUT

  // Convert analog value to voltage (assuming 5V reference)
  float voltage = analogValue * (5.0 / 1023.0);

  // Print the readings to the Serial Monitor
  Serial.print("Analog Value: ");
  Serial.print(analogValue);
  Serial.print(" | Voltage: ");
  Serial.print(voltage);
  Serial.print("V | Digital Value: ");
  Serial.println(digitalValue);

  delay(1000); // Wait for 1 second before the next reading
}
```

DHT22 (AM2302) – цифровий сенсор температури та відносної вологості, який передає дані по однопровідному інтерфейсу (власний протокол, не I²C). Усередині містить датчик вологості, термодатчик і мікроконтролер, який формує готові цифрові значення, тому на відміну від аналогових сенсорів не потребує АЦП. Зазвичай має 3–4 виводи (VCC, DATA, GND, інколи ще NC), працює від 3,3 В до 5 В, а для стабільної роботи лінії DATA потрібен підтягувальний резистор ~4,7–10 кΩ до VCC (часто вже встановлений на модулі). Сенсор зручний для навчальних робіт та IoT, але має невисоку швидкість оновлення (типово ~1 вимір/2 с) і чутливий до довжини проводів та наводок. Для Arduino/ESP найчастіше використовують бібліотеки на кшталт DHT Sensor Library (Adafruit), а результат зчитування (температура °C, вологість %) виводять у Serial Monitor або передають мережею. Схема підключення DHT22 наведена на рис. 12:

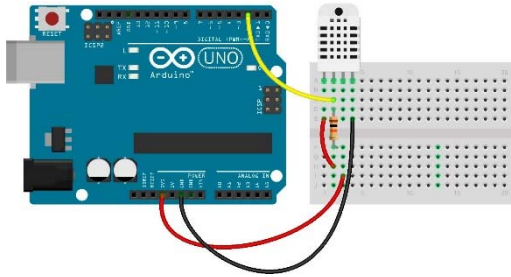


Рисунок 12 – Підключення сенсора DHT22 до Arduino UNO

Приклад коду для взаємодії з сенсором DHT22 наведено нижче:

```
#include "DHT.h"

#define DHTPIN 2 // what pin we're connected to
#define DHTTYPE DHT22 // DHT 22 (AM2302)
#define fan 4

int maxHum = 60;
int maxTemp = 40;

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  pinMode(fan, OUTPUT);
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250
  // milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a
  // very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
}
```

```
if(h > maxHum || t > maxTemp) {
    digitalWrite(fan, HIGH);
} else {
    digitalWrite(fan, LOW);
}

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.println(" *C ");
}
}
```

Для роботи з сенсором DHT22 не забувайте підключати додаткові бібліотеки для роботи.

DS18B20 – цифровий термодатчик, що вимірює температуру та передає дані по шині 1-Wire, завдяки чому кілька датчиків можна підключити паралельно до одного піну мікроконтролера (кожен має унікальну 64-бітну адресу). Працює зазвичай від 3,0 до 5,5 В, має виводи VDD, GND, DQ (DATA) і потребує підтягувального резистора $\sim 4,7$ к Ω між DQ та VDD. Датчик підтримує роздільну здатність 9–12 бітів (чим вища – тим довший час перетворення), часто застосовується для вимірювання температури в системах моніторингу, «розумному домі» та лабораторних роботах. Для Arduino/ESP його зазвичай використовують разом із бібліотеками OneWire та DallasTemperature, що спрощують пошук датчиків на шині та зчитування температури у $^{\circ}\text{C}$. Схема підключення DS18B20 наведена на рис. 13.

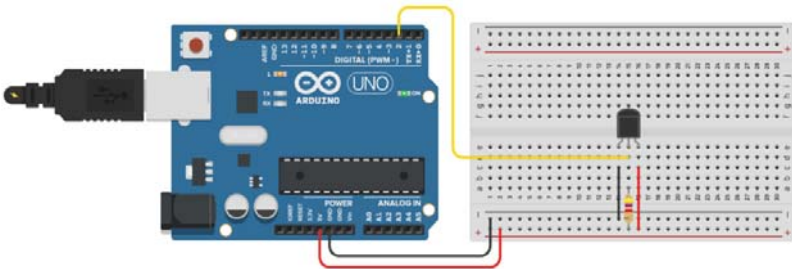


Рисунок 13 – Підключення сенсору DS18B20 до Arduino UNO

Для роботи з цим сенсором в Arduino IDE потрібно встановити додаткову бібліотеку DallasTemperature за допомогою **Tools** → **Manage Libraries**. Приклад коду для взаємодії з сенсором DS18B20 наведено нижче:

```
// Include the required Arduino libraries:
#include "OneWire.h"
```

```
#include "DallasTemperature.h"

// Define to which pin of the Arduino the 1-Wire bus is
connected:
#define ONE_WIRE_BUS 2

// Create a new instance of the oneWire class to communicate
with any OneWire device:
OneWire oneWire(ONE_WIRE_BUS);

// Pass the oneWire reference to DallasTemperature library:
DallasTemperature sensors(&oneWire);

void setup() {
  // Begin serial communication at a baud rate of 9600:
  Serial.begin(9600);
  // Start up the library:
  sensors.begin();
}

void loop() {
  // Send the command for all devices on the bus to perform a
  temperature conversion:
  sensors.requestTemperatures();

  // Fetch the temperature in degrees Celsius for device index:
  float tempC = sensors.getTempCByIndex(0); // the index 0
  refers to the first device
  // Fetch the temperature in degrees Fahrenheit for device
  index:
  float tempF = sensors.getTempFByIndex(0);

  // Print the temperature in Celsius in the Serial Monitor:
  Serial.print("Temperature: ");
  Serial.print(tempC);
  Serial.print(" \xC2\xB0"); // shows degree symbol
  Serial.print("C | ");

  // Print the temperature in Fahrenheit
  Serial.print(tempF);
  Serial.print(" \xC2\xB0"); // shows degree symbol
  Serial.println("F");

  // Wait 1 second:
  delay(1000);
}
```

КУ-028 – навчальний модуль для вимірювання температури, який зазвичай містить NTC-терморезистор (термістор) і компаратор LM393, тому може працювати у двох режимах: як аналоговий датчик (плавна

зміна напруги залежно від температури) та як цифровий «пороговий» датчик (спрацювання при досягненні встановленої температури).

– Виводи модуля: зазвичай VCC, GND, AO (англ. Analog Out), DO (англ. Digital Out).

– AO: напруга від подільника з терморезистором → зчитується АЦП (Arduino: A0; ESP32: ADC; для ESP8266 з AO обережно через обмеження ADC і рівнів напруги).

– DO: вихід компаратора LM393; поріг налаштовується потенціометром на платі (зазвичай є світлодіоди індикації живлення та спрацювання).

– Живлення: часто 3,3–5 В (залежить від конкретного модуля), але на практиці модулі часто орієнтовані на 5 В.

Важливо зазначити, що KY-028 дає відносний аналоговий рівень, а не готову температуру в °C «з коробки». Щоб отримати температуру в градусах, потрібна калібровка або використання рівнянь/таблиці для NTC (наприклад, модель Beta або Steinhart-Hart), тоді як цифровий вихід DO зручний для задач типу «перевищено поріг / не перевищено».

Схема підключення KY-028 наведена на рис. 14.

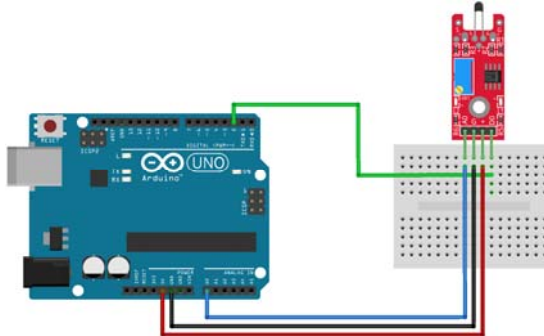


Рисунок 14 – Підключення сенсору KY-028 до Arduino UNO
Приклад коду для взаємодії з сенсором KY-028 наведено нижче:

```
const int PIN_AO = A0;
const int PIN_DO = 2;

const float BETA = 3950.0;           // Beta (K)
const float R0 = 10000.0;           // опір при 25°C (Ом)
const float T0 = 298.15;            // 25°C у Кельвінах

const float R_FIXED = 10000.0;      // Ом

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

```
    pinMode(PIN_DO, INPUT);
}

float analogToCelsius(int adc) {
    // Захист від крайніх значень
    if (adc <= 0) adc = 1;
    if (adc >= 1023) adc = 1022;

    // Припускаємо схему подільника: (VCC) -- R_FIXED --
    (вузол=AO)-- NTC -- GND
    // Тоді: Vout = Vcc * (Rntc / (R_FIXED + Rntc))
    // => Rntc = R_FIXED * Vout / (Vcc - Vout)
    float vRatio = adc / 1023.0; // Vout/Vcc
    float rNtc = R_FIXED * vRatio / (1.0 - vRatio);

    // Модель Beta: 1/T = 1/T0 + (1/B)*ln(R/R0)
    float invT = (1.0 / T0) + (1.0 / BETA) * log(rNtc / R0);
    float tempK = 1.0 / invT;
    return tempK - 273.15;
}

void loop() {
    int ao = analogRead(PIN_AO);
    int d0 = digitalRead(PIN_DO);

    float tC = analogToCelsius(ao);

    Serial.print("AO=");
    Serial.print(ao);
    Serial.print(" DO=");
    Serial.print(d0);
    Serial.print(" Temp~");
    Serial.print(tC, 1);
    Serial.println(" C");

    delay(500);
}
```

Лабораторна робота № 5. Робота зі світловими сенсорами

Мета: Отримання досвіду роботи з компонентами, які змінюють свої характеристики залежно від інтенсивності світла.

Завдання:

Реалізувати схему пристрою та програму для нього в залежності від варіанту:

Перший варіант:

Компоненти: Arduino UNO, резистори, IR-ресівер, фототранзистор, RGB-світлодіод

Пристрій, який активується червоною кнопкою (On/Off) на пульті управління. Залежно від рівня освітленості, який потрапляє на фотодіод, змінюється колір RGB світлодіода. Шкала градації складається з 5 юнітів, кожен юніт включає в себе певний діапазон рівня освітленості (діапазони можуть бути нерівні; ви можете розставляти їх на свій розсуд). Кольори світлодіоду відносно номерів діапазонів освітленості:

1. #6E6EFF
2. #0000FF
3. #00FF66
4. #00FF00
5. #C8FF00

Також мають бути реалізовані два режими зміни кольорів при натисненні кнопок 1 та 2 на пульті управління. Перший режим – кольори змінюються в прямому порядку списку, який зазначено вище; другий режим – в оберненому порядку. Схема підключення наведена на рис. 15.

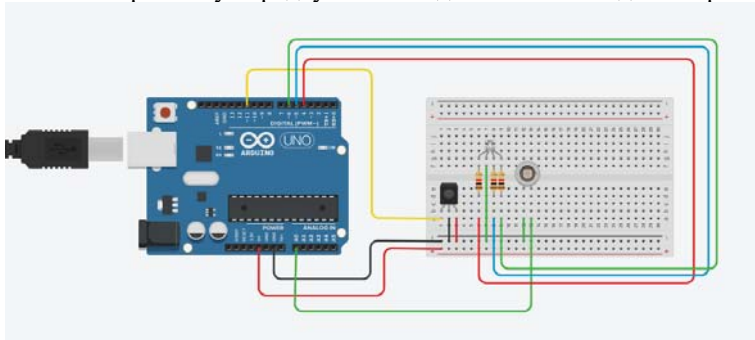


Рисунок 15 – Схема підключення компонентів до Arduino UNO

Другий варіант:

Компоненти: Arduino UNO, резистор, IR-ресівер, фоторезистор, зумер

Пристрій, який активується червоною кнопкою (On/Off) на пульті управління. Залежно від рівня освітленості, який потрапляє на фоторезистор, змінюється частота тону зумеру. Шкала градації складається з 5 юнітів, кожен юніт включає в себе певний діапазон рівня освітленості (діапазони можуть бути нерівні, ви можете розташовувати їх на свій розсуд). Частота звуку зумеру відносно номерів діапазонів освітленості:

1. 0
2. 900
3. 1000

4. 1200
5. 1500

Також має бути реалізовано три режими зміни частоти звуку при натисненні кнопок VOL+ та VOL- на пульті управління. Перший режим – частота звуку при будь-якому рівні освітленості дорівнює 0. Другий режим – частота звуку така ж, як і зазначена в списку. Третій режим – частота збільшена в два рази. Схема підключення наведена на рис. 16.

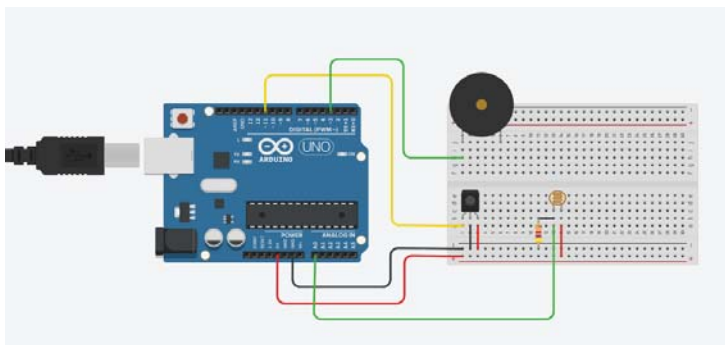


Рисунок 16 – Схема підключення компонентів до Arduino UNO

Інформація щодо виконання лабораторної роботи:

Завданням даної роботи є створення пристрою на базі IR-ресивера, який під час отримання певного сигналу з пульту управління буде виконувати певну задачу.

Для виконання даної задачі вам потрібно буде підключити бібліотеку <IRremote.h>, яка допоможе вам відправляти та отримувати сигнали в інфрачервоному діапазоні. Найбільшою проблемою в виконанні даної роботи на живих пристроях є те, що під час процесу прийому/передачі ресивер часто видає «сміття», яке викликано проблемою видимості поміж пристрою-трансмівтера та ресивера. У випадку виконання роботи в середовищі Autodesk Tinkercad дана проблема відсутня.

Також не забувайте, що бібліотека IRremote.h для трансмітера за замовчуванням вважає, що data pin підключено до 3-го пину Arduino. Для того, щоб замінити номер пину на інший, під час створення об'єкта IRrecv передайте значення номера вашого дата-пину.

Наприклад:

```
IRrecv irrecv(11);
```

Лабораторна робота № 6. Ознайомлення з роботою сервомоторів

Мета: здобуття навичок роботи з сервомотором SG90 та набуття навичок роботи з бібліотекою Servo.h.

Завдання:

Реалізуйте схему з використанням сервоприводу, яка буде включати в себе такі компоненти, як:

1. Arduino Uno;
2. IR receiver;
3. Сервопривід;
4. RGB світлодіод;
5. Пульт дистанційного керування.

При отриманні сигналу з пульта дистанційного керування кут повороту сервоприводу має змінюватися на 40 градусів, а також має змінюватися колір RGB світлодіоду. Кнопки на пульті, що задіяні в управлінні, – це «1» та кнопка вимкнення. У випадку натискання кнопки вимкнення – сервопривід повертається до початкового положення та вимикається світлодіод. У випадку натискання кнопки «1» – кут повороту змінюється на 40 градусів, а також вмикається RGB-світлодіод. Кольори світлодіоду в залежності від кута повороту сервопривода:

1. 40 - #00FF00
2. 80 - #80FF00
3. 120 - #FFFF00
4. 160 - #FF8000

Якщо після натискання кнопки «1» значення кута повороту стає більшим за 160 градусів – сервопривід повертається до початкового положення.

Інформація щодо виконання лабораторної роботи:

Servo.h – це бібліотека, яка має в собі набір функцій для роботи з сервоприводами. Вона дозволяє управляти одночасно великою кількістю сервоприводів, які підключені до плати Arduino.

Для того, щоб почати роботу з даною бібліотекою, треба її спочатку підключити в Arduino IDE і після цього створити зміну типу Servo. Виглядає це так:

```
#include <Servo.h>
Servo servo;
```

Основні методи даної бібліотеки:

```
servo.attach(pin, min, max);
```

– pin – обов'язковий параметр: цифровий PIN, до якого підключено сигнальний провід сервоприводу;

– `min` – необов'язковий параметр, ширина імпульсу в мікросекундах, яка відповідає початковому (0 градусам) положенню сервоприводу (за замовчуванням 544);

– `max` – необов'язковий параметр, ширина імпульсу в мікросекундах, яка відповідає максимальному (180 градусам) положенню сервоприводу.

```
servo.write(angle);
```

`angle` – обов'язковий параметр, кут повороту сервоприводу

```
servo.read()
```

Повертає `int` значення положення сервомотору на даний момент

```
servo.attached(pin)
```

`pin` – цифровий пін, до якого підключено сигнальний провід сервоприводу

Повертає `bool` значення – `true` якщо підключено, `false` – якщо ні

```
servo.detach(pin)
```

`pin` – цифровий пін, до якого підключено сигнальний провід сервоприводу

Від'єднує змінну `servo` від вказаного виходу.

Завданням даної роботи є створення пристрою-ресивера, який буде оброблювати сигнал з пульта дистанційного керування.

Для виконання даної задачі вам потрібно буде підключити бібліотеку `<IRremote.h>`, яка допоможе вам відправляти та отримувати сигнали в інфрачервоному діапазоні. Найбільшою проблемою в виконанні даної роботи є те, що під час процесу прийому/передачі ресивер часто видає «сміття», яке викликано проблемою видимості поміж трансмітером та ресивером.

Схема підключення IR-ресивера наведено на рис. 17:

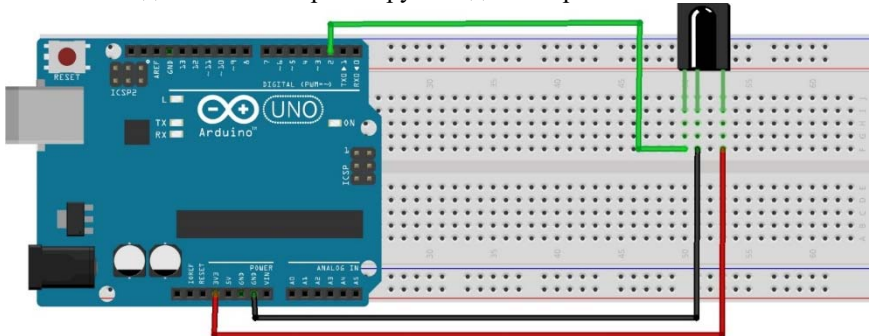


Рисунок 17 – Схема підключення IR-ресивера до Arduino UNO

Майте на увазі, що остання схема валідна, якщо ви підключаєте тільки IR-ресівер, без плати. У випадку, якщо ресівер є частиною плати – схема підключення віддзеркалюється.

Схема підключення сервоприводу наведена на рис. 18:

Більшість сервоприводів має 3 контакти підключення – 5 V, GND та пін управління. Не є винятком і сервопривід SG90. Характеристики його такі:

- швидкість без навантаження: 0,12 сек/60 °. при напрузі 4,8 В;
- обертовий момент: 2 кг / см;
- температурний діапазон: від 0 до + 50°C;
- ширина мертвої зони: 4 мікросекунди;
- робоча напруга живлення: 3,5–5 В;
- Струм, що споживається під час руху: 50–80 мА;
- Струм, що споживається під час утримання: 5–10 мА;
- кут повороту 180 °;
- розміри: 3,3 см × 3 см × 1,3 см;
- вага: 9 г.

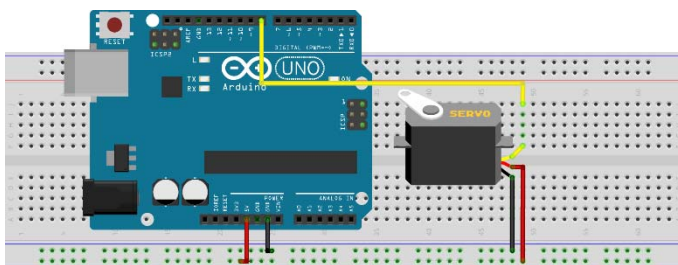


Рисунок 18 – Схема підключення сервоприводу до Arduino UNO

Лабораторна робота № 7. Використання SPDT реле

Мета: здобуття навичок використання SPDT-реле в проєктах на базі плат керування Arduino UNO або подібних за функціоналом.

Завдання:

1. побудуйте схему залежно від вашого варіанту;
2. напишіть код програми для вашої схеми, скопіюйте та завантажте його на плату / запустіть симуляцію з ним в Tinkercad;

3. звіт з даної роботи повинен включати в себе схему та моменти роботи/симуляції з відкритим Serial Monitor, а також в звіті має бути наявний код програми.

Варіанти:

1. Пристрій, який буде включати лампочку у випадку, якщо температура буде вища за поставлене обмеження. В Serial Monitor повинні виводитися сповіщення про зміну стану реле та зміни значення температури. Приклад схеми наведено на рис. 19.

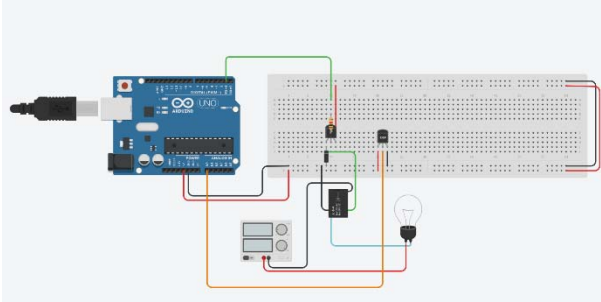


Рисунок 19 – Схема пристрою. Вона включає в себе Arduino UNO, макетну плату, джерело живлення, температурний сенсор TMP36, SPDT реле, PNP транзистор, діод, резистор на 1 кОм.

2. Пристрій, який буде включати лампочку у випадку, якщо PIR-сенсор зреагує на рух, і відключатиметься, якщо момент руху завершиться. У Serial Monitor повинні виводитися сповіщення про зміну стану реле та моменти початку та завершення руху. Приклад схеми наведено на рис. 20.

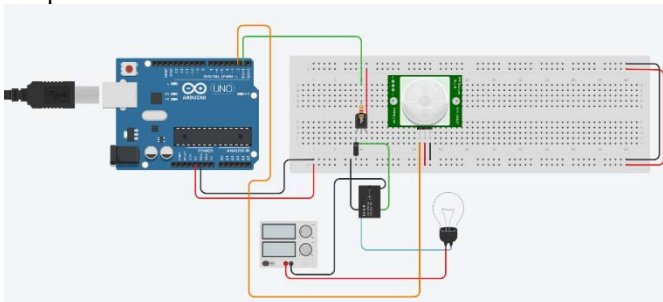


Рисунок 20 – Схема пристрою. Вона включає в себе Arduino UNO, макетну плату, джерело живлення, PIR сенсор, SPDT реле, PNP транзистор, діод, резистор на 1 кОм

Інформація щодо виконання лабораторної роботи:

Реле – це електричний пристрій, призначений для виконання певних перемикачів під час зміни певних електричних або неелектричних входних взаємодій.

Реле бувають різних типів, але зазвичай вони складаються з котушки, якоря (який замикає контакти під час включення реле) і контактів. В залежності від кількості контактів, які замикаються, та полюсів, вони поділяються на такі типи:

- SPST – однополюсні однопозиційні;
- DPST – двополюсні однопозиційні;
- SPDT – однополюсні двопозиційні;
- DPDT – двополюсні двопозиційні;

Схема SPDT реле наведена на рис. 21. Це реле працює таким чином – в верхній частині знаходяться контакти котушки (1, 2), яка дозволяє перемикати контакти 5 та 3, подаючи напругу на неї. В відключеному стані замикаються контакти 5 та 4.

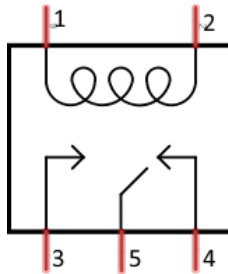


Рисунок 21 – Схема SPDT реле

В Autodesk Tinkercad SPDT-реле виглядає як на рис. 22. На даному елементі вже є наведена схема реле, яка пояснює, для чого слугує кожен з контактів даного реле.

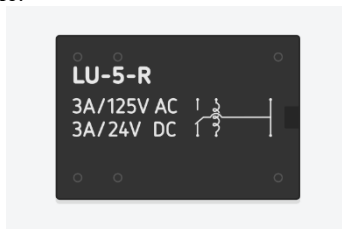


Рисунок 22 – SPDT-реле в Autodesk Tinkercad

Під час виконання лабораторної роботи необхідно пам'ятати, що реле дозволяє керувати навантаженням більшої потужності малим

керувальним сигналом (наприклад, з мікроконтролера), забезпечуючи розділення кола керування і силового кола. Для коректної роботи у схемах постійного струму рекомендовано передбачати захист від зворотної ЕРС котушки (захисний діод, підключений паралельно котушці у зворотному напрямку), а також перевіряти допустимі значення напруги та струму контактів реле відповідно до обраного навантаження.

Лабораторна робота № 8. Використання перемикачів

Мета: здобуття навичок створення проєктів на базі Arduino UNO з застосуванням різних типів перемикачів.

Завдання:

1. побудуйте схему пристрою (приклад наведено на рис. 23), який буде працювати за певним алгоритмом дій;

2. перша кнопка відповідає за номер сервоприводу, яким планується оперувати. Залежно від кількості натискань – змінюється номер сервоприводу;

3. друга кнопка відповідає за кут повороту сервоприводу. Кожне натискання на другу кнопку змінює кут повороту обраного сервоприводу на 40 градусів. У випадку, коли кнопка натиснута в п'ятий раз – кут повороту має дорівнювати 0 градусам;

4. третя кнопка відповідає за увімкнення/вимкнення RGB-світлодіоду. Колір світлодіоду змінюється від кута повороту сервоприводу. Список кольорів в залежності від кута нахилу:

- 0 градусів - #FFFFFF
- 40 градусів - #CCFF33
- 80 градусів - #00CCFF
- 120 градусів - #9933FF
- 160 градусів - #FF0000

5. напишіть код програми для вашої схеми, завантажте та запустіть його на пристрої.

Звіт з даної роботи повинен включати в себе фото/скріншоти схеми та моменту симуляції в Tinkercad, а також в звіті має бути наявний код програми.

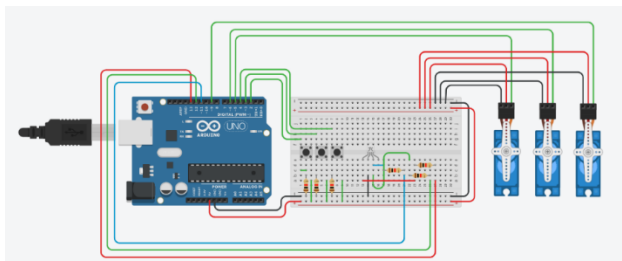


Рисунок 23 – Приблизна схема пристрою. Вона включає в себе Arduino UNO, макетну плату, три push-button, RGB світлодіод, три сервоприводи та три резистори на 1 кОм

Інформація щодо виконання лабораторної роботи:

Перемикачі – це механічні пристрої, які мають два або більше електричних контактів. Коли перемикач відкритий або відключений – то електричне коло роз’єднане, а коли перемикач закритий або підключений – то контакти з’єднані. Приклад коду з використанням перемикача PushButton (який під’єднано до піну 4) наведено далі.

```
#define BUTTON_PIN 4
void setup()
{
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT);
}
void loop()
{
  byte buttonState = digitalRead(BUTTON_PIN);
  if (buttonState == LOW) {
    Serial.println("Button is pressed");
  }
  else {
    Serial.println("Button is not pressed");
  }
  delay(100);
}
```

Для стабільної роботи перемикачів у практичних схемах необхідно враховувати явище брязкоту контактів (англ. *debounce*), коли під час натискання/відпускання сигнал короткочасно «стрибає» між станами HIGH/LOW і може спричиняти хибні спрацювання. Тому рекомендується використовувати підтягувальний резистор (англ. *pull-up* або *pull-down*) та програмну фільтрацію: наприклад, додати коротку затримку

після зміни стану або перевіряти повторно значення через кілька мілісекунд; у разі потреби можна застосувати і апаратне RC-узгодження.

У межах даної лабораторної роботи доцільно організувати опрацювання натискань кнопок не через безперервне утримання, а через фіксацію окремої події натискання. Це означає, що зміна номера сервоприводу, кута повороту або стану RGB-світлодіода має виконуватися лише один раз у момент переходу кнопки зі стану «не натиснуто» у стан «натиснуто». Такий підхід дозволяє уникнути багаторазового спрацювання однієї і тієї ж команди під час утримання кнопки, робить поведінку пристрою передбачуваною та значно спрощує подальше налагодження програми.

Окрему увагу слід приділити правильному підключенню сервоприводів і RGB-світлодіода до плати Arduino UNO. Сервоприводи повинні отримувати коректний сигнал керування з цифрових виводів, а також стабільне живлення, оскільки під час зміни кута повороту можливі короточасні стрибки споживаного струму. Для RGB-світлодіода необхідно використовувати струмообмежувальні резистори, а в програмі бажано реалізувати окрему функцію, яка буде встановлювати потрібний колір залежно від поточного кута обраного сервоприводу, що зробить код більш зрозумілим і зручним для супроводу.

ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

У рамках самостійної роботи з дисципліни «Сенсори, перетворювачі», студенти мають реалізувати свій власний Proof-of-Concept проєкт. Для цього повинна бути обрана тематика проєкту, яка буде узгоджена з керівником курсу. Передбачається, що студент буде використовувати вільне програмне забезпечення Arduino IDE для розробки програмної частини, а також плати Arduino Uno або їх апаратні клони, до яких будуть підключені потрібні апаратні модулі.

Студент має продемонструвати працюючий проєкт відповідно до обраної теми, а також розуміння принципів реалізації, що використовувалися під час виконання завдання.

№	Теми проєктів (орієнтовний перелік)
1	Створення еквалайзера на базі RGB-світлодіодної панелі, мікрофону, плеєра DFPlayer YX5200 та Arduino Uno.
2	Створення системи підрахунку автомобільного трафіку на базі інфрачервоних сенсорів руху HC-SR501.
3	Станція мікроклімату аудиторії на базі DHT22/DS18B20 + LCD1602/OLED + зумер (пороги «занадто жарко/волого»).
4	Сигналізація витоку газу/CO на базі MQ-2/MQ-9 + зумер + світлова індикація (RGB/LED) + журнал подій у Serial Monitor.
5	Система контролю рівня води в баку: HC-SR04 (над водою) або поплавковий датчик + реле для насоса + індикація рівня (LED/LCD).
6	Парктронік/антизіткнення: HC-SR04 + зумер (частота залежить від відстані) + світлова шкала (LED bar / RGB).
7	Розумне освітлення: фоторезистор/LDR або BH1750 + PWM-керування яскравістю світлодіода/стрічки + ручний режим (потенціометр).
8	Підрахунок відвідувачів/перетинів проходу: два ІЧ-бар'єри (або дві пари IR-Тх/Rx) → визначення напрямку (вхід/вихід) → лічильник на LCD.
9	Система контролю доступу: RFID RC522 + сервопривід (замок) + звукова індикація + журнал спроб доступу.
10	Метео-індикатор тиску: BMP280/BME280 + OLED/LCD + тренд (стрілка «росте/падає») + попередження про різкі зміни.
11	Антикрадіжка/охоронний датчик: PIR HC-SR501 + магнітний геркон на двері + сирена/зумер + затримка постановки/зняття.

ПІДСУМКОВИЙ КОНТРОЛЬ

В кінці шостого семестру як підсумковий контроль передбачено залік. Кожен заліковий білет складається з трьох питань, кожне з яких оцінюється в 10 балів. Максимальна можлива кількість балів за залік – 30.

Перелік питань до заліку:

1. Дайте визначення терміну «сенсор».
2. У чому різниця між сенсором та актуатором?
3. Поясніть значення терміну «Електричний перетворювач».
4. Розподіліть за типами вимірюваних даних сенсори.
5. Наведіть приклади температурних сенсорів.
6. Принцип роботи аналогових сенсорів.
7. Що таке цифрові сенсори?
8. Що таке кондиціонування сигналу?
9. Дайте визначення поняттю «позиційний сенсор».
10. Принцип роботи потенціометра.
11. Що таке диференційний трансформатор для вимірювання лінійних переміщень? Наведіть приклад його використання.
12. Що таке індуктивний безконтактний датчик? Наведіть приклад його структури.
13. Що таке датчик кута повороту? Наведіть приклади таких датчиків.
14. Наведіть приклад використання інкрементного енкодера.
15. Дайте визначення терміну «температурний сенсор».
16. Які існують два типи температурних сенсорів? Дайте їх визначення.
17. Що таке термостат? Наведіть його структуру.
18. Дайте визначення терміну «терморезистор».
19. Наведіть структуру терморезистора.
20. Що мається на увазі під аббревіатурою RTD?
21. Наведіть принцип роботи RTD.
22. Що таке термопара? Наведіть приклади застосування термопари в побуті.
23. Які існують типи термопари залежно від кольору?
24. Що таке світловий сенсор?
25. Які існують типи світлових сенсорів?
26. Принцип роботи фотоемісійних комірок.
27. Принцип роботи фотокондуктивних комірок.
28. Принцип роботи фотоелектричних комірок.
29. Принцип роботи фотоз'єднувальних комірок.
30. Що таке фоторезистор? Поясніть його структуру.

31. Що таке фотодіод? Наведіть приклад його застосування.
32. Наведіть структуру фототранзистора.
33. Поясніть принцип роботи сонячних батарей.
34. Що таке електромеханічне реле? Наведіть його приклад.
35. Яка структура електромеханічного реле?
36. Які матеріали використовуються в контактних наконечниках електричного реле?
 37. Які типи контактних електричних реле існують?
 38. Що таке твердотільне реле?
 39. Який принцип роботи твердотільних реле?
 40. Які характеристики твердотільного реле існують?
 41. Що таке інтерфейс модулю вводу/виводу?
 42. Що таке лінійний соленоїдний перетворювач?
 43. Як працюють поворотні соленоїди?
 44. За яким принципом і за рахунок чого відбувається перемикання соленоїдів?
 45. Що таке двигун постійного струму?
 46. Що таке двигун змінного струму?
 47. Які існують базові типи двигунів постійного струму?
 48. Наведіть приклад використання щіткового мотора змінного струму.
 49. Як працює сервомотор? Наведіть структуру простого сервомотора.
 50. Як працює контролер регулювання швидкості мотора?
 51. Що таке кроковий двигун постійного струму?
 52. Яка структура крокового мотора?
 53. Що таке звуковий перетворювач?
 54. Що таке мікрофон? Принцип роботи мікрофону?
 55. Які існують плеєри для відтворення звуку, сумісні з Arduino?
 56. За яким принципом працює звуковий вивід на динаміки?
 57. Які існують частоти для певних типів звуків?
 58. HiFi-структура аудіовиводу.
 59. Які існують датчики-детектори руху сумісні з Arduino?
 60. Які існують акселерометри, сумісні з Arduino?

КРИТЕРІЇ ОЦІНЮВАННЯ ТА ЗАСОБИ ДІАГНОСТИКИ РЕ- ЗУЛЬТАТІВ НАВЧАННЯ

№ з/п	Вид діяльності (завдання)	Максимальна кількість балів
1.	Лабораторна робота № 1	5
2.	Лабораторна робота № 2	5
3.	Лабораторна робота № 3	5
4.	Лабораторна робота № 4	5
5.	Лабораторна робота № 5	5
6.	Лабораторна робота № 6	5
7.	Лабораторна робота № 7	5
8.	Лабораторна робота № 8	5
9.	Самостійна робота	30
10.	Залік	30
	Разом	100

За одну практичну роботу студент може отримати максимально 5 балів. Залежно від правильності виконання студентом роботи, оцінка може варіюватися від 0 до 5. Градація оцінки за практичну роботу наведена в таблиці нижче.

Ступінь виконання роботи	Максимальна кількість балів
Завдання повністю виконане у відповідності до поставленої задачі	5
Завдання виконане частково	3
Завдання виконане, але результат не відповідає поставленій задачі	1
Завдання не виконано	0

Максимальною кількістю балів за самостійну роботу є 30. Залежно від коректності результатів роботи студентського проекту студент може отримати від 0 до 30 балів. Бали за кожну частину виконаної роботи наведено нижче.

Елемент виконання самостійної роботи	Кількість балів
Проектування алгоритму роботи комплексу та блок-схем	6
Підбір обладнання для виконання самостійної роботи та його компонування в один комплекс	10
Створення програмної частини проекту	10

«Сенсори, перетворювачі»

Елемент виконання самостійної роботи	Кількість балів
Формування звіту щодо виконання самостійної роботи	4
Разом	30

До заліку допускаються лише ті студенти, які накопичили не менше 20 балів за виконання завдань протягом семестру. Заліковий білет складається з 3 питань, кожне з яких оцінюється в 10, 10 та 20 балів відповідно у випадку вірної відповіді.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основні:

1. Arduino IDE. *Arduino Documentation*. URL: <https://docs.arduino.cc/software/ide/> (Last accessed: 13.02.2026).
2. Arduino-Based Experiments: Leveraging Engineering Design and Scientific Inquiry in STEM Lessons / N. D. Dat et al. *International Journal of STEM Education for Sustainability*. 2024. Vol. 4, no. 1. P. 38–53. DOI: 10.53889/ijses.v4i1.317.
3. Arduino IDE. *Arduino Documentation*. URL: <https://docs.arduino.cc/software/ide/> (Last accessed: 13.02.2026).
4. Design and Implementation of ESP32-Based IoT Devices / D. Hercog et al. *Sensors*. 2023. Vol. 23, no. 15. P. 6739. DOI: 10.3390/s23156739.
5. Exploring Students' Perception of Virtual Laboratory Adoption on an IoT Course / M. N. A. Wahyudi et al. *TEM Journal*. 2025. P. 1531–1547. DOI: 10.18421/tem142-52.
6. Grossi M., Omaña M. Accuracy of NTC Thermistor Measurements Using the Sensor to Microcontroller Direct Interface. ICSEE 2024. Basel Switzerland, 2024. P. 12. DOI: 10.3390/ecsa-11-20527.
7. He Y., Jiao M. A Mini-Review on Metal Oxide Semiconductor Gas Sensors for Carbon Monoxide Detection at Room Temperature. *Chemosensors*. 2024. Vol. 12, no. 4. P. 55. DOI: 10.3390/chemosensors12040055.
8. Low-Cost Ultrasonic Range Improvements for an Assistive Device / D. Abreu et al. *Sensors*. 2021. Vol. 21, no. 12. P. 4250. DOI: 10.3390/s21124250.
9. Petkovšek M., Nemec M., Zajec P. Algorithm Execution Time and Accuracy of NTC Thermistor-Based Temperature Measurements in Time-Critical Applications. *Mathematics*. 2021. Vol. 9, no. 18. P. 2266. DOI: 10.3390/math9182266.
10. Remote IoT Education Laboratory for Microcontrollers Based on the STM32 Chips / P. Jacko et al. *Sensors*. 2022. Vol. 22, no. 4. P. 1440. DOI: 10.3390/s22041440.
11. Sreenivasappa B. V., Shashikala A. R. A review on recent developments in materials and methodologies for metal oxide semiconductor liquefied petroleum gas sensors operating at room temperature. *Sensors International*. 2026. Vol. 7. P. 100354. DOI: 10.1016/j.sintl.2025.100354.
12. Rakshitha Kiran. Integration of MQ-5 Gas Sensors with Arduino Board. *Journal of Electrical Systems*. 2024. Vol. 20, no. 10s. P. 2154–2165. DOI: 10.52783/jes.5540.

13. Babu D. A. V. Smart Blind Stick Using ESP32. *International Journal for Research in Applied Science and Engineering Technology*. 2024. Vol. 12, no. 3. P. 3404–3409. DOI: 10.22214/ijraset.2024.59614.

Додаткові:

14. HC-SR04 Ultrasonic Module User Guide. *ELECFREAKS*. URL: <https://www.electfreaks.com/blog/post/hc-sr04-ultrasonic-module-user-guide.html> (Last accessed: 16.03.2026).

15. HC-SR04. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/hc-sr04/> (Last accessed: 16.03.2026).

16. DS18B20 Datasheet and Product Info | Analog Devices. Mixed-signal and digital signal processing ICs | *Analog Devices*. URL: <https://www.analog.com/en/products/ds18b20.html> (Last accessed: 16.03.2026).

17. DallasTemperature. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/dallastemperature/> (Last accessed: 16.03.2026).

18. OneWire. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/onewire/> (Last accessed: 16.03.2026).

19. DHT sensor library. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/dht-sensor-library/> (Last accessed: 16.03.2026).

20. DHT sensor library for ESPx. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/dht-sensor-library-for-esp/> (Last accessed: 16.03.2026).

21. DHT11, DHT22 and AM2302 Sensors. *Adafruit Learning System*. URL: <https://learn.adafruit.com/dht/overview> (Last accessed: 16.03.2026).

22. MQ-9B CO & CH4 Detection Sensor--Winsen. *Winsen Gas Sensor_CO2 Sensor_Air Quality Sensor_Dust Sensor_CO Sensor-Winsen Electronics*. URL: <https://www.winsen-sensor.com/sensors/co-sensor/mq-9b.html> (Last accessed: 16.03.2026).

23. MQ-9B WINSSEN - Sensor: gas | methane (CH4), carbon monoxide (CO); MQ-9B | Transfer Multisort Elektronik. *TME*. URL: <https://www.tme.eu/en/details/mq-9b/gas-sensors/winsen/> (Last accessed: 16.03.2026).

24. LM393. *Analog | Embedded processing | Semiconductor company | TI.com*. URL: <https://www.ti.com/product/LM393> (Last accessed: 16.03.2026).

Євген Сергійович Дарнапук

Методична серія
Випуск № 484

СЕНСОРИ, ПЕРЕТВОРЮВАЧІ

Методичні рекомендації

Для практичних занять
з дисципліни для здобувачів вищої освіти
першого (бакалаврського) рівня вищої освіти
спеціальності F7 Комп'ютерна інженерія

Друкується в авторській редакції

Друк С. Волинець. Фальцювально-палітурні роботи О. Мішалкіна.

Підп. до друку 22.04.2026.
Формат 60x84 ¹/₁₆. Папір офсет.
Гарнітура «Times New Roman». Друк різнограф.
Ум. друк. арк. 2,79. Обл.-вид. арк. 1,63.
Тираж 10 пр. Зам. № 7209.

Видавець і виготовлювач: ЧНУ ім. Петра Могили.
54003 м. Миколаїв, вул. 68 Десантників, 10.
Тел.: +380 (512) 50-03-32, +380 (512) 76-55-81,
e-mail: rector@chmnu.edu.ua

Свідоцтво суб'єкта видавничої справи ДК №6124 від 05.04.2018