

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з дисципліни
«Обчислювальна техніка та програмування» частина 2
для студентів спеціальності 141 – Електроенергетика,
електротехніка та електромеханіка всіх форм навчання

Методичні вказівки до виконання лабораторних робіт з дисципліни «Обчислювальна техніка та програмування» частина 2 для студентів спеціальності 141 – Електроенергетика, електротехніка та електромеханіка всіх форм навчання. /Укл.: Л.С. Скрупська, С.І. Шило – Запоріжжя: НУЗП, 2025.– 29с.

Укладачі: Л.С. Скрупська, ст. викл.
С.І. Шило, ст. викл., к.т.н.

Рецензент: В.В.Василевський, доцент, к. т. н.

Відповідальний
за випуск: Л.С. Скрупська, ст. викл.

Затверджено
на засіданні кафедри
«Електричні та
електронні апарати»
Протокол № 1
від « 12 » 08 2025 р.

Затверджено НМК ЕТФ
Протокол № 1
від « 21 » 08 2025 р.

ЗМІСТ

Вступ.....	4
Лабораторна робота № 1. Основи роботи в редакторі Microsoft Excel	6
Лабораторна робота № 2. Основи алгоритмізації та структурного програмування.....	9
Лабораторна робота № 3. Програмне забезпечення для технічних розрахунків	14
Лабораторна робота № 4 Моделювання електронних схем у середовищі Electronics Workbench.....	16
Лабораторна робота № 5. Сучасне програмування та його особливості	19
Лабораторна робота № 6. Вбудовані системи в електронних апаратах	22
Лабораторна робота № 7. Особливості роботи вбудованих систем	25
Список використаних джерел посилання	28

ВСТУП

Обчислювальна техніка відіграє вирішальну роль у розвитку сучасної науки та техніки, особливо в таких галузях, як електроенергетика, електротехніка та електромеханіка. Комп'ютерні системи, програмне забезпечення та знання з алгоритмізації та програмування є незамінними інструментами для вирішення складних інженерних завдань, що виникають під час проектування, моделювання та оптимізації електронних схем, систем управління та багатьох інших процесів.

Лабораторні роботи з обчислювальної техніки та програмування мають на меті надати студентам фундаментальні знання про комп'ютерні системи, їх архітектуру, програмні засоби, а також практичні навички роботи з програмними пакетами, які використовуються для технічних розрахунків і моделювання в електроенергетиці та електротехніці.

Програма лабораторної роботи охоплює такі важливі аспекти:

1. Основи роботи в редакторі Microsoft Excel: Excel є потужним інструментом для обробки даних, виконання технічних розрахунків та візуалізації результатів. Студенти навчатимуться використовувати Excel для створення таблиць, графіків, виконання математичних операцій та статистичного аналізу даних.

2. Основи алгоритмізації та структурного програмування: Ці знання є базовими для створення програмного забезпечення, що автоматизує інженерні обчислення та оптимізацію процесів. Студенти вивчатимуть принципи побудови алгоритмів, використання структур керування (цикли, розгалуження), а також основи структурного програмування.

3. Програмне забезпечення для технічних розрахунків: Вивчення специфічних програмних пакетів, таких як MATLAB, є важливим для математичного моделювання та технічних розрахунків. MATLAB надає широкий спектр інструментів для виконання складних обчислень, аналізу даних та візуалізації результатів.

4. Моделювання електронних схем у середовищі Electronics Workbench: Моделювання електронних схем є важливим етапом проектування електронних пристроїв. Electronics Workbench

дозволяє студентам отримати практичні навички моделювання та аналізу електронних схем, що допомагає перевірити їх функціональність до побудови реальних пристроїв.

5. Сучасне програмування та його особливості: Розгляд основних принципів програмування для вирішення інженерних завдань, включаючи використання мов програмування та сучасних підходів до розробки програм. Студенти ознайомляться з різними мовами програмування (наприклад, C++, Python) та їх застосуванням для вирішення конкретних інженерних задач.

6. Вбудовані системи в електронних апаратах: Знайомство з особливостями роботи вбудованих систем, їх застосуванням в електронних пристроях та автоматизованих системах управління. Вбудовані системи є невід'ємною частиною багатьох сучасних електронних пристроїв, тому їх вивчення є важливим для майбутніх інженерів.

Метою лабораторних занять є формування у студентів вмінь та навичок ефективного використання сучасних обчислювальних та програмних засобів для розв'язання інженерних завдань у галузі електроенергетики, електротехніки та електромеханіки.

Ця лабораторна робота передбачає як теоретичну підготовку, так і практичні вправи, що дозволять студентам:

- Освоїти основи обчислювальної техніки та програмування;
- Зрозуміти принципи роботи комп'ютерних систем і програмного забезпечення;
- Навчитися використовувати програмні інструменти для технічних розрахунків та моделювання електронних схем;
- Зрозуміти роль вбудованих систем в електронних апаратах та їх практичне застосування.

Завдання лабораторної роботи допоможуть студентам закріпити теоретичні знання та отримати практичні навички для ефективного використання обчислювальної техніки в інженерній діяльності.

ЛАБОРАТОРНА РОБОТА № 1

Основи роботи в редакторі Microsoft Excel

Мета роботи:

Метою лабораторної роботи є освоєння основ роботи з редактором Microsoft Excel. Студенти повинні навчитися створювати прості таблиці, виконувати базові розрахунки, обробляти дані та використовувати функції Excel для вирішення технічних задач.

1. Обладнання та програмне забезпечення

Обладнання:

- Комп'ютер із встановленим програмним забезпеченням Microsoft Excel (версія 2016, 2019 або Office 365).

Програмне забезпечення:

- Microsoft Excel.

2. Завдання лабораторної роботи

2.1. Створити таблицю даних в Microsoft Excel.

2.2. Виконати розрахунки за допомогою простих математичних операцій.

2.3. Навчитися працювати з листами робочої книги.

2.4. Використовувати основні функції Excel для обробки даних.

2.5. Виконати обчислення функції в Excel та зберегти результат.

3. Теоретичні відомості

3.1. Створення таблиці в Microsoft Excel:

- Excel — це потужний інструмент для обробки даних, який дозволяє створювати таблиці, що складаються з рядків і стовпців.

- Кожна клітинка таблиці може містити текст, числа, формули або результати функцій.

- Структура таблиці в Excel визначається набором рядків та стовпців, де кожен стовпець має своє ім'я (наприклад, А, В, С), а кожен рядок — номер (1, 2, 3 і так далі).

3.2. Робота з листами робочої книги:

- У Microsoft Excel кожен документ складається з робочої книги, яка містить один або кілька листів.

- Кожен лист є окремою таблицею, з якою можна працювати індивідуально.

- Листи можуть бути додані або видалені, а також перейменовані для зручності користувача.

3.3. Виконання розрахунків у таблиці:

- Для виконання розрахунків Excel використовує формули. Формула починається з символу "=", після якого вказуються операції та функції.

- Наприклад, для обчислення суми значень у клітинках використовується функція SUM(A1:A5).

- Excel підтримує операції додавання, віднімання, множення та ділення.

3.4. Обробка даних таблиці:

- Excel має інструменти для фільтрації, сортування та форматування даних.

- Дані можуть бути відсортовані за зростанням чи спаданням, що спрощує їх аналіз.

- Можна застосовувати умовне форматування для виділення важливих значень.

3.5. Обчислення функцій у Excel:

- Excel має велику кількість вбудованих функцій для математичних, статистичних, логічних та фінансових розрахунків.

- Функції можуть використовувати діапазони клітинок і повертати результати обчислень.

4. Хід виконання роботи

Крок 1: Створення простої таблиці

- Відкрийте Microsoft Excel і створіть новий порожній документ.

- Заповніть перший стовпець (A) даними (наприклад, перерахувати 5 елементів).

- У клітинку A1 введіть "Назва".

- У клітинки A2-A6 введіть назви предметів чи інші дані.

- Заповніть другий стовпець (B) числами або значеннями, з якими потрібно працювати (наприклад, кількість одиниць продукції чи ціни).

- У клітинку В1 введіть "Кількість".
- У клітинки В2-В6 введіть значення.

Крок 2: Виконання розрахунків у таблиці

- У клітинку С1 введіть "Вартість".
- В клітинки С2-С6 введіть формулу для обчислення вартості (наприклад, якщо клітинки А2 і В2 мають відповідно назву товару та кількість, а С2 — це ціна за одиницю):
 - У клітинку С2 введіть формулу: =В2*Ціна_за_одиницю.
 - Скопіюйте формулу на інші клітинки стовпця С.
- Виконайте підсумок вартості. У клітинку С7 введіть формулу:
 - =SUM(С2:С6) для підсумку всіх вартостей.

Крок 3: Робота з листами робочої книги

- Додайте новий лист до робочої книги, натиснувши кнопку "Додати лист" у нижній частині вікна Excel.
- Переіменуйте новий лист, подвійно клацнувши на його назву (наприклад, з "Лист2" на "Підсумки").
- Додати до неї стовпці Загальна вартість, грн., Загальна вартість, у.о.
- Виконати розрахунки Загальної вартості, грн. для першого рядку. Для розрахунку загальної вартості товару в у.о. використати абсолютну адресацію, курс у.о. взяти з таблиці, яку розмістити понад заголовком таблиці..

Курс	\$
Курс	\$

- Скопіювати у всі комірки відповідних стовпців формулу, яка введена у перший рядок, методом протягування.
- Знайти загальну кількість товару та його середню ціну за допомогою відповідних функцій.
- Зберегти результати роботи у поточній книзі.

Крок 4: Обробка даних таблиці

- Виберіть стовпець з даними, наприклад, "Вартість", і застосуйте умовне форматування для виділення значень, що перевищують певний поріг (наприклад, більше 1000).
- Перейдіть на вкладку *Головна* → *Умовне форматування* → *Правила для виділення клітинок* → *Більше ніж*.

- Використовуйте фільтри для сортування даних. Виберіть таблицю і натисніть Дані → Фільтр. Застосуйте фільтрацію для відображення лише тих значень, які вас цікавлять.

Крок 5: Виконання обчислення функції

- Виберіть клітинку, де потрібно виконати обчислення функції (наприклад, для обчислення середнього значення вартості).

- Введіть функцію:

- У клітинку C8 введіть формулу =AVERAGE(C2:C6), щоб обчислити середнє значення.

- Для знаходження максимального значення використовуйте функцію =MAX(C2:C6).

- Додати у таблицю записи за допомогою форми, виконати розрахунки Загальної вартості грн. та Загальної вартості, у.о. для доданих рядків.

- Створити Звіт надходження товару за останні 2 дні, для чого встановити відповідний фільтр.

Крок 6: Збереження і закриття документа

- Збережіть документ, натиснувши Файл → Зберегти як і вибравши формат Excel (.xlsx).

- Закрийте документ після виконання роботи.

Контрольні питання

1. Як створити таблицю в Excel і що таке клітинка?
2. Які операції можна виконувати з даними в таблиці Excel?
3. Як застосувати формулу для виконання математичних операцій в Excel?
4. Як додати новий лист в робочу книгу Excel?
5. Які функції Excel можуть бути використані для обчислень середнього значення та максимуму?
6. Як застосувати умовне форматування до даних у таблиці?

ЛАБОРАТОРНА РОБОТА № 2

Основи алгоритмізації та структурного програмування

Мета роботи: Метою лабораторної роботи є ознайомлення студентів з основами алгоритмізації, принципами побудови алгоритмів, а також з техніками структурного програмування.

Студенти навчатися будувати алгоритми та представляти їх за допомогою блок-схем. Окрім того, буде розглянуто типи алгоритмів та основні конструкції, що застосовуються в структурному програмуванні.

1. Обладнання та програмне забезпечення

1. Персональний комп'ютер.
2. Операційна система: Windows, Linux (залежно від комп'ютерного середовища).
3. Програмне забезпечення для розробки алгоритмів та програмування:
4. Мови програмування: C, Python, Pascal (залежно від вибору викладача).
5. Інструменти для побудови блок-схем: Microsoft Visio, Lucidchart, або інші інструменти для створення діаграм.

2. Завдання лабораторної роботи

- Ознайомитись з теоретичними основами алгоритмізації, алгоритмами та їх властивостями.
- Навчитись будувати алгоритми для простих задач та представляти їх у вигляді блок-схем.
- Ознайомитись з основними типами алгоритмів та принципами структурного програмування.
- Створити програму, яка реалізує алгоритм з використанням основних конструкцій структурного програмування.
- Зрозуміти і продемонструвати принципи інкапсуляції, модульності, та використання контрольних структур.

3. Теоретичні відомості

3.1. Алгоритм та його властивості:

- Алгоритм — це чітка та послідовна інструкція для виконання певної задачі.
- Основні властивості алгоритмів:
 - Коректність: алгоритм повинен завжди давати правильний результат.
 - Точність: алгоритм має бути без недоліків і не повинно бути двозначностей в його описі.

- **Завершеність:** алгоритм повинен завершувати виконання після певного числа кроків.

- **Ефективність:** алгоритм має бути оптимальним за витратами часу та пам'яті.

3.2. Способи подання алгоритмів:

- Текстове подання (послідовність команд у вигляді кроків).
- Блок-схеми — графічне подання алгоритмів.
- Програмний код (реалізація алгоритму в конкретній мові програмування).

3.3. Побудова алгоритмів:

- Аналіз задачі та визначення вхідних і вихідних даних.
- Пошук ефективного методу розв'язку.
- Опис алгоритму у вигляді тексту чи блок-схеми.

3.4. Блок-схеми як інструмент для побудови алгоритмів:

- Використовуються для візуалізації логіки алгоритмів.
- Блоки в схемі:
- Овал: початок/кінець.
- Прямокутник: обчислювальна дія.
- Ромб: умовне відгалуження.
- Паралелограм: введення/виведення даних.

3.5. Основні типи алгоритмів:

- Лінійні алгоритми — виконуються поетапно, без розгалужень та циклів.

- Алгоритми з розгалуженнями — використовують умовні оператори для вибору одного з кількох варіантів виконання.

- Циклічні алгоритми — виконують певні дії кілька разів до досягнення умови.

3.6. Принципи структурного програмування:

- Модульність — програму розбивають на підпрограми (функції, процедури), що виконують конкретні завдання.

- Ієрархічність — підпрограми можуть викликати інші підпрограми.

- Інкапсуляція — дані та функції приховані від зовнішнього доступу.

- Слідування принципу "одна функція — одна задача".

3.7. Основні конструкції структурного програмування:

- Послідовність — виконання операцій одна за одною.

- Умовні оператори (if, else) — виконання дій в залежності

від умов.

- Цикли (for, while, do-while) — повторення операцій до виконання певної умови.

- Функції — блоки коду, що виконують певну задачу і можуть бути викликані з основної програми.

4. Хід виконання роботи

4.1. Підготовка до роботи:

- Ознайомтесь із завданням лабораторної роботи.

- Переконайтесь, що на комп'ютері встановлено необхідне програмне забезпечення для виконання лабораторної роботи (наприклад, текстовий редактор для написання алгоритмів).

- Підготуйтеся до роботи з папером і ручкою для малювання блок-схем або використовуйте спеціальні програми для їх створення (наприклад, Microsoft Visio, Lucidchart).

4.2. Визначення задачі для алгоритму

- Виберіть задачу для алгоритмізації (наприклад, обчислення факторіалу числа, пошук максимального елемента в масиві, обчислення чисел Фібоначчі тощо).

- Сформулюйте чітко умови задачі та визначте необхідні вхідні й вихідні дані.

4.3. Побудова алгоритму

Почніть розробку алгоритму для вирішення обраної задачі. Для цього:

- а). Проаналізуйте умови задачі та визначте основні етапи її розв'язання.

- б). Використовуйте текстовий опис або псевдокод для запису послідовності дій.

- в). Пам'ятайте про основні принципи алгоритмізації: визначення вхідних/вихідних даних, побудова послідовності операцій, використання умовних та циклічних конструкцій.

4.4 Побудова блок-схеми

Для опису алгоритму побудуйте блок-схему. Дотримуйтесь таких кроків:

- Нанесіть початкову точку (старт) на блок-схему.

- Розставте блоки для кожного етапу алгоритму (вхідні/вихідні дані, операції, умови).

- Для умовних конструкцій використовуйте ромбічні блоки,

для циклів – прямокутники зі спеціальними позначками.

- Перевірте правильність з'єднання блоків та логічну послідовність дій.

4.5. Перевірка алгоритму

- Перевірте побудований алгоритм на логічні помилки, переконайтесь, що алгоритм коректно вирішує задачу, і всі етапи виконуються правильно.

- Якщо алгоритм складається з умовних операторів або циклів, перевірте, чи працюють всі можливі варіанти виконання.

4.6. Перехід до програмування

- Використовуйте мову програмування (наприклад, Python, C++, Java або іншу) для реалізації побудованого алгоритму. Впишіть алгоритм у вигляді програми.

- Тестуйте програму, ввівши різні вхідні дані та перевірте правильність результатів.

- Переконайтесь, що програма коректно обробляє всі випадки, які можуть виникнути під час виконання.

4.7. Аналіз ефективності

- Оцініть ефективність вашого алгоритму. Як швидко він працює для великих обсягів даних? Чи є можливість його оптимізації (зменшення кількості операцій, покращення швидкості виконання)?

- Проаналізуйте використання пам'яті та час виконання програми для різних вхідних даних.

Контрольні питання

1. Що таке алгоритм? Назвіть його основні властивості.
2. Які існують способи подання алгоритмів? Який з них найзручніший для інженерів?
3. Як правильно будувати алгоритм для технічних задач?
4. Як використовувати блок-схеми для візуалізації алгоритмів?
5. Які основні типи алгоритмів існують? Наведіть приклади.
6. Що таке принципи структурного програмування?
7. Як організувати програму з використанням структурного програмування?
8. Назвіть основні конструкції структурного програмування та їх застосування.

ЛАБОРАТОРНА РОБОТА № 3

Програмне забезпечення для технічних розрахунків

Мета роботи:

- Ознайомлення студентів з основними можливостями математичного пакету MATLAB.
- Розвиток навичок використання MATLAB для технічних розрахунків, моделювання та вирішення інженерних задач.
- Вивчення основ використання MATLAB у галузі електроенергетики та електротехніки.

1. Обладнання та програмне забезпечення:

- Персональний комп'ютер.
- Програмне забезпечення MATLAB (версія не нижче 2020 року).
- Підключення до інтернету для доступу до ресурсів MATLAB, якщо необхідно.

2. Завдання лабораторної роботи:

- 2.1. Ознайомлення з інтерфейсом MATLAB.
- 2.2. Виконання простих обчислень і вирішення математичних задач у MATLAB.
- 2.3. Створення функцій та скриптів для вирішення інженерних задач.
- 2.4. Моделювання технічних процесів (наприклад, розрахунок електричних схем, аналіз навантаження на електричні елементи тощо).
- 2.5. Виведення результатів у графічному та текстовому вигляді.

3. Теоретичні відомості:

MATLAB (Matrix Laboratory) — це потужне програмне середовище для технічних обчислень, аналізу даних, математичного моделювання та візуалізації результатів. MATLAB є стандартом у багатьох галузях техніки, зокрема в електроенергетиці, електротехніці та інших інженерних напрямках, завдяки своїм потужним можливостям для обробки великих обсягів числових даних.

3.1. Основні компоненти MATLAB:

- Командний рядок — основне місце для введення команд.
- Скрипти та функції — для зберігання наборів команд, які можуть бути виконані безпосередньо або через виклик.
- Математичні можливості — підтримка лінійної алгебри, обчислення інтегралів, диференціація, статистика, оптимізація тощо.
- Графіка та візуалізація — вбудовані функції для побудови графіків, поверхонь, гістограм тощо.
- Simulink — додатковий інструмент для моделювання, який інтегрується з MATLAB і дозволяє проводити моделювання на основі блок-схем.

Використання MATLAB для технічних розрахунків в електроенергетиці включає такі завдання:

3.2. Розрахунок параметрів електричних схем.

- Моделювання та аналіз динаміки систем.
- Оптимізація роботи електричних мереж.
- Проведення статистичного аналізу та обробка сигналів.

4. Хід виконання роботи:

4.1. Підготовка середовища:

- Встановіть MATLAB на вашому комп'ютері.
- Запустіть програму та ознайомтесь з інтерфейсом (вікно командного рядка, вікно Workspace, вікно командного історії).

4.2. Основні операції в MATLAB:

- Введіть прості арифметичні вирази в командному рядку (наприклад, розрахунок $3 * (2 + 5)$).
- Створіть змінні та присвоюйте їм значення (наприклад, $A = 5$, $B = 10$).
- Використовуйте вбудовані функції для обчислень (наприклад, \sin , \cos , $\sqrt{}$).

4.3. Створення скриптів:

- Напишіть простий скрипт для розв'язку задачі (наприклад, розрахунок параметрів кола чи аналіз навантаження на трансформатор).

- Створіть функцію для виконання математичних обчислень та збережіть її у файлі.

4.4. Моделювання інженерних задач:

- Використовуйте MATLAB для створення математичних моделей технічних процесів.
- Побудуйте графіки для візуалізації результатів (наприклад, характеристика електричного кола).
- Створіть блок-схему в Simulink для моделювання електричних систем.

4.5. Аналіз результатів:

- Збережіть графіки та обчислені значення у файли.
- Оцініть точність розрахунків та можливості оптимізації за допомогою MATLAB.

Контрольні питання:

1. Що таке MATLAB і для яких задач він використовується?
2. Які типи даних підтримує MATLAB?
3. Як створити та використовувати функцію в MATLAB?
4. Як будувати графіки та візуалізувати дані в MATLAB?
5. Які основні математичні можливості MATLAB для розв'язку інженерних задач?
6. Як використовувати Simulink для моделювання технічних процесів?
7. У чому переваги використання MATLAB у порівнянні з іншими програмними продуктами для технічних обчислень?

ЛАБОРАТОРНА РОБОТА № 4

Моделювання електронних схем у середовищі Electronics Workbench

Мета роботи:

- Ознайомлення зі середовищем для моделювання електронних схем — Electronics Workbench.
- Навчитися аналізувати та вимірювати параметри електронних схем.
- Виконати оптимізацію електронних пристроїв за допомогою програмного забезпечення.

1. Обладнання та програмне забезпечення

- Персональний комп'ютер.

- Програмне забезпечення: Electronics Workbench (або аналогічне середовище для моделювання електронних схем).

2. Завдання лабораторної роботи

- Створити електронну схему в середовищі Electronics Workbench.

- Проаналізувати поведінку схеми за допомогою симуляційних можливостей.

- Виконати вимірювання електричних параметрів: напруги, струми, потужність.

- Оптимізувати електронну схему для досягнення необхідних характеристик.

- Порівняти теоретичні та симульовані результати.

3. Теоретичні відомості

Програмне середовище Electronics Workbench надає різноманітні інструменти для вимірювання, моніторингу, проектування та моделювання параметрів електронних схем:

- Моделювання електронних схем дозволяє перевіряти роботу пристроїв без фізичного зібрання, що економить час та ресурси.

- Вимірювання в схемах включають напруги, струми, потужність та інші електричні характеристики компонентів.

- Оптимізація електронних пристроїв передбачає налаштування параметрів елементів схеми для досягнення заданих характеристик.

4. Хід виконання роботи

4.1. Запуск програми Electronics Workbench

- Відкрийте програму Electronics Workbench. Ознайомтесь з основним інтерфейсом (панель інструментів, робоче середовище, меню).

- Створіть новий проект.

4.2. Створення електронної схеми

- Оберіть компоненти для вашої схеми з бібліотеки (резистори, конденсатори, транзистори тощо).

- Перетягніть їх на робочу площину і з'єднайте дротами відповідно до завдання.

- Для кожного компонента виберіть параметри, наприклад, для резистора — номінал, для джерела живлення — напругу.

4.3. Запуск симуляції та аналіз результатів

- Після завершення побудови схеми натисніть на кнопку запуску симуляції.

- Виберіть вид симуляції: постійний струм (DC), змінний струм (AC), транз'єнтний аналіз (для схем з динамічними елементами).

- Перевірте схему на наявність помилок (неправильне підключення, відсутні компоненти).

- Огляньте результат симуляції, зокрема параметри, які вас цікавлять, як-от струм, напруга на різних елементах.

4.4. Вимірювання параметрів схеми

- Використовуйте вбудовані інструменти програми для вимірювання параметрів: вольтметр, амперметр, осцилограф.

- Виміряйте напругу на різних точках схеми.

- Виміряйте струм, що проходить через компоненти схеми.

- Запишіть отримані значення для подальшого аналізу.

4.5. Оптимізація схеми

- Після вимірювання основних параметрів схеми спробуйте оптимізувати її для досягнення бажаних характеристик (зменшення споживаної потужності, зміна номіналів компонентів, покращення стабільності роботи).

- Змініть номінали резисторів, конденсаторів або інші параметри, щоб досягти найкращих результатів.

- Запустіть повторну симуляцію для нових параметрів.

4.6. Порівняння теоретичних та симульованих результатів

- Виконайте розрахунки за допомогою теоретичних методів (з використанням формул для аналізу схеми).

- Порівняйте отримані теоретичні значення з результатами симуляції.

4.7. Збереження результатів роботи

- Збережіть вашу схему та результати вимірювань для подальшого аналізу або для демонстрації викладачу.

- Зробіть знімки екрану з параметрами симуляцій та результатами вимірювань.

Контрольні питання

1. Які основні переваги моделювання електронних схем у середовищі Electronics Workbench?
2. Як правильно вибирати параметри компонентів при створенні схеми?
3. Що таке оптимізація електронних пристроїв і як її здійснити?
4. Як можна виміряти напругу та струм у схемі за допомогою Electronics Workbench?
5. Які типи симуляцій доступні в Electronics Workbench і як вибирати необхідний тип?
6. Чому важливо порівнювати теоретичні та симульовані результати?

ЛАБОРАТОРНА РОБОТА № 5

Сучасне програмування та його особливості

Мета роботи:

Ознайомити студентів з основами сучасного програмування, основними програмними засобами, конструкціями та тенденціями. Розвинути навички роботи з сучасними середовищами розробки та ознайомити з основами взаємодії програмного забезпечення у сфері електронних апаратів.

1. Обладнання та програмне забезпечення

- Комп'ютер або ноутбук з ОС Windows/Linux/macOS;
- Комп'ютер із встановленим середовищем Electronics Workbench.
- Мікроконтролер (Arduino Uno, STM32 або аналогічний);
- Додаткові електронні компоненти (світлодіоди, датчики температури, резистори, провідники);
- Програмне забезпечення для емуляції роботи мікроконтролерів (Proteus, Tinkercad або аналогічне).

2. Завдання лабораторної роботи

- 2.1. Ознайомитися з основними програмними засобами, що

використовуються у сучасному програмуванні.

2.2. Вивчити основні програмні конструкції та їх застосування у технічних системах.

2.3. Дослідити основні тенденції сучасного програмування.

2.4. Виконати практичне моделювання роботи програмних конструкцій у Electronics Workbench..

2.5. Проаналізувати отримані результати та зробити висновки.

3. Теоретичні відомості

3.1. Програмні засоби

Сучасне програмування передбачає використання широкого спектру програмних засобів, таких як компілятори, інтерпретатори, бібліотеки, середовища розробки. Популярними є:

- Arduino IDE – для програмування мікроконтролерів Arduino;
- STM32CubeIDE – для роботи з STM32;
- Python – високорівнева мова для автоматизації, аналізу даних та взаємодії з електронними пристроями;
- Visual Studio Code – зручне середовище для програмування різними мовами.

3.2. Основні програмні конструкції

Програмні конструкції поділяються на:

- Секвенційні операції – виконуються послідовно одна за одною.
- Умовні оператори – реалізують прийняття рішень у програмі.
- Циклічні оператори – дозволяють повторювати певні дії кілька разів.
- Функції та процедури – використовуються для структуризації програмного коду.

3.3. Основні тенденції сучасного програмування

- Використання високорівневих мов програмування (Python, Java, C#).
- Автоматизація та оптимізація коду (штучний інтелект, нейромережі).
- Об'єктно-орієнтоване програмування (структурування даних у вигляді об'єктів).
- Інтернет речей (IoT) – впровадження програмованих пристроїв у побутові та промислові системи.

3.4. Особливості сучасного програмування

- Оптимізація програм для мікроконтролерів;
- Використання протоколів зв'язку (UART, I2C, SPI);
- Реалізація енергоефективних алгоритмів.

4. Хід виконання роботи

Крок 1: Підготовка середовища Electronics Workbench

- Запустіть Electronics Workbench на комп'ютері.
- Створіть новий проєкт.
- Відкрийте бібліотеку доступних компонентів.

Крок 2: Ознайомлення з програмуванням логічних схем

- Виберіть мікроконтролер або логічний модуль із бібліотеки.
- Додайте необхідні елементи на робочий простір (логічні вентилі AND, OR, NOT, таймери).
- Ознайомтесь із принципами роботи цих елементів.

Крок 3: Створення базової логічної схеми

- Використовуючи логічні вентилі, змодельуйте базову систему керування, наприклад, світлодіодний індикатор.
- З'єднайте входи та виходи компонентів.
- Перевірте коректність схеми.

Крок 4: Аналіз поведінки схеми

- Запустіть симуляцію у Electronics Workbench.
- Переконайтеся, що логічні оператори працюють відповідно до заданих умов.
- Внесіть зміни у конфігурацію схеми та повторно перевірте результат.

Крок 5: Дослідження впливу змін на роботу системи

- Додайте таймери або тригери до схеми.
- Запрограмуйте логіку перемикання станів.
- Запустіть симуляцію та спостерігайте, як змінюються вихідні сигнали.

Крок 6: Завершення роботи

- Проаналізуйте отримані результати.
- Запишіть висновки про особливості програмування логічних схем у Electronics Workbench.
- Збережіть створений проєкт.

Контрольні питання

1. Які основні програмні засоби використовуються в сучасному програмуванні?
2. Які основні програмні конструкції існують?
3. Які тенденції сучасного програмування?
4. Як впроваджується програмування вбудованих систем?
5. Як логічні оператори впливають на роботу електронних апаратів?
6. Як змодельовати цифрову схему у Electronics Workbench?

ЛАБОРАТОРНА РОБОТА № 6

Вбудовані системи в електронних апаратах

Мета роботи:

Ознайомлення студентів з основами вбудованих систем, їх класифікацією, основними характеристиками та особливостями розробки. Вивчення принципів використання вбудованих систем у сучасних електронних апаратах.

1. Обладнання та програмне забезпечення

1. Комп'ютер з операційною системою Windows/Linux.
2. Програмне середовище для роботи з вбудованими системами (наприклад, Arduino IDE, Electronics Workbench).
3. Вбудований контролер (наприклад, Arduino Uno, STM32, ESP32).
4. Модулі та датчики (температурний датчик, світлодіоди, кнопки).
5. Блок живлення або USB-кабель для підключення контролера.

2. Завдання лабораторної роботи

- Ознайомитися з основними компонентами вбудованих систем.
- Вивчити класифікацію вбудованих систем та їх характеристики.
- Розглянути особливості розробки вбудованих систем.

- Виконати практичне завдання з аналізу роботи вбудованої системи (наприклад, запрограмувати мікроконтролер на керування світлодіодом або зчитування даних з датчика).

3. Теоретичні відомості

3.1. Основні компоненти вбудованих систем

Вбудовані системи складаються з таких основних елементів:

- *Процесор (мікроконтролер, мікропроцесор)* – керує роботою пристрою.

- *Оперативна пам'ять (ОЗП)* – тимчасово зберігає дані.

- *Постійна пам'ять (ROM, флеш-пам'ять)* – містить прошивку пристрою.

- *Датчики та модулі* – отримують дані про навколишнє середовище.

- *Актори (приводи, реле, двигуни)* – виконують дії на основі команд контролера.

- *Комунікаційні модулі (Wi-Fi, Bluetooth, CAN, UART, SPI, I2C)* – забезпечують обмін даними.

3.2. Класифікація вбудованих систем

- За рівнем продуктивності:

- Прості (мікроконтролери з мінімальним функціоналом).

- Середньої складності (з можливістю підключення модулів).

- Високопродуктивні (багатоядерні процесори, операційні системи).

- За призначенням:

- Критичні до часу (реального часу – RTS).

- Некритичні (звичайні контролери для автоматизації побутових пристроїв).

3.3. Особливості розробки вбудованих систем

- Оптимізація використання пам'яті та енергоспоживання.

- Вибір відповідної архітектури мікроконтролера.

- Використання ефективних алгоритмів програмування.

- Захист від зовнішніх перешкод (екранування, фільтрація сигналів).

3.4. Основні характеристики вбудованих систем

- Робота в реальному часі.

- Обмежені ресурси (пам'ять, продуктивність).
- Надійність і безперервність роботи.
- Низьке енергоспоживання.

4. Хід виконання роботи

Крок 1: Підготовка середовища Electronics Workbench

- Запустіть Electronics Workbench на комп'ютері.
- Створіть новий проект.
- Відкрийте бібліотеку компонентів та ознайомтесь із доступними мікроконтролерами та іншими електронними елементами.

Крок 2: Створення схеми вбудованої системи

- Додайте до робочого простору наступні компоненти:
- Мікроконтролер (наприклад, 8051 або AVR, якщо доступний).
- Джерело живлення.
- Світлодіод (LED) для індикації роботи.
- Резистори для обмеження струму.
- З'єднайте компоненти відповідно до принципової схеми вбудованої системи.

Крок 3: Налаштування параметрів компонентів

- Відкрийте вікно властивостей мікроконтролера.
- Виберіть частоту тактування (наприклад, 8 МГц).
- Налаштуйте параметри входів/виходів (наприклад, встановіть один із портів як вихід для керування світлодіодом).

Крок 4: Симуляція роботи вбудованої системи

- Запустіть симуляцію в Electronics Workbench.
- Спостерігайте за поведінкою системи (чи загоряється світлодіод?).
- Внесіть зміни у параметри мікроконтролера або схему та знову запустіть симуляцію.

Крок 5: Аналіз результатів

- Проаналізуйте, як мікроконтролер виконує команди.
- Досліджуйте, як змінюється поведінка системи при зміні параметрів схеми.

Контрольні питання

1. Що таке вбудовані системи?

2. Які основні компоненти вбудованих систем?
3. Як класифікуються вбудовані системи?
4. Які характеристики мають вбудовані системи?
5. Які особливості роботи мікроконтролера?
6. Як працюють датчики у вбудованих системах?

ЛАБОРАТОРНА РОБОТА № 7

Особливості роботи вбудованих систем

Мета роботи:

- Ознайомитися з основними принципами роботи вбудованих систем.
- Дослідити процеси керування вбудованими системами.
- Вивчити управління енергоспоживанням у вбудованих системах.
- Ознайомитися з особливостями функціонування систем реального часу..

1. Обладнання та програмне забезпечення

- Комп'ютер із встановленим середовищем Electronics Workbench (тепер відомий як Multisim).
- Мікроконтролери та мікросхеми для створення схем вбудованих систем.
- Операційна система Windows (або інша сумісна).
- Додаткові інструменти для тестування схем (віртуальні осцилоскопи, джерела живлення, мультиметри).

2. Завдання лабораторної роботи

- Ознайомлення з принципами роботи вбудованих систем.
- Розробка схем для керування процесами за допомогою мікроконтролерів.
- Розробка схеми для управління енергоспоживанням на прикладі вбудованої системи.
- Створення та симуляція реальних процесів, що працюють в реальному часі.

3. Теоретичні відомості

Вбудовані системи — це пристрої, що виконують конкретні функції в рамках більшої системи. Вони часто використовуються в автоматизації, побутовій техніці, автомобільних системах і робототехніці. Основні особливості вбудованих систем:

- Управління процесами: вбудовані системи керують певними технологічними процесами, використовуючи датчики та виконавчі механізми.

- Управління енергоспоживанням: оптимізація енергетичних ресурсів для продовження роботи при мінімальних витратах.

- Реальний час: вбудовані системи мають здатність виконувати операції в реальному часі, що критично для багатьох застосувань, таких як автомобільна електроніка або медичні пристрої.

4. Хід виконання роботи

Крок 1. Запуск програми Electronics Workbench (Multisim)

- Відкрийте програму Electronics Workbench на вашому комп'ютері.

- Створіть новий проект для роботи вбудованої системи.

Крок 2. Проектування схеми для керування процесами

- У панелі компонентів знайдіть мікроконтролер (наприклад, PIC16F877A або ATmega).

- Додайте мікроконтролер до робочої області.

- Підключіть датчики (наприклад, термістор для вимірювання температури) та виконавчі пристрої (наприклад, реле або сервопривод) до пінів мікроконтролера.

- Програмуйте мікроконтролер для виконання заданого процесу (наприклад, включення вентилятора при перевищенні температури).

Крок 3. Проектування схеми для управління енергоспоживанням

- Додайте до попередньої схеми резистори та світлодіоди, що імітують навантаження вбудованої системи.

- Виміряйте споживаний струм та напругу у різних вузлах схеми за допомогою віртуального мультиметра в Electronics Workbench.

- Проведіть експеримент із відключенням частини логічних вузлів і проаналізуйте зміну споживаної потужності.
- Зробіть висновки про способи зниження енергоспоживання у вбудованих системах.

Крок 4. Створення симуляції для реального часу

- Розробіть просту систему для симуляції реального часу. Наприклад, додайте таймери для вимірювання часу і синхронізації процесів.
- Використовуйте генератор імпульсів для моделювання процесів, що мають виконуватися у певні проміжки часу.
- Проаналізуйте затримки сигналів та реакцію схеми на зміну вхідних параметрів.
- Змініть частоту генератора тактових імпульсів і оцініть вплив на час виконання операцій.
- Підготуйте короткий звіт про отримані результати.

Крок 5. Тестування та аналіз роботи системи

- Після розробки схем та програмування мікроконтролера, перевірте роботу схеми в режимі симуляції.
- Перевірте правильність керування процесами, енергоспоживанням та роботою в реальному часі. У разі потреби внесіть корективи в схему чи програму.

Контрольні питання

1. Що таке вбудована система і де вона застосовується?
2. Які основні елементи входять до складу вбудованих систем?
3. Що таке управління процесами в контексті вбудованих систем?
4. Як можна оптимізувати енергоспоживання в вбудованій системі?
5. У чому особливості роботи в реальному часі для вбудованих систем?
6. Як впливає симуляція в реальному часі на ефективність тестування вбудованих систем?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ПОСИЛАННЯ

1. Ласкун В.І. "Microsoft Excel: підручник для початківців". — Київ: "Освіта", 2020. — 350 с.
2. Павленко О.Ю. "Microsoft Excel: основи та практичні завдання". — Харків: "Фоліо", 2019. — 280 с.
3. Захарова В.Г. "Практикум з Excel для студентів технічних спеціальностей". — Харків: "Літера", 2021. — 220 с.
4. Кузьміна Н.В. "Обробка даних за допомогою Microsoft Excel". — Львів: "Альфа", 2021. — 180 с.
5. Піткін, О. О. "Основи алгоритмізації та програмування: підручник". — Київ: Вища школа, 2017. — 320 с.
6. Грок, С. Б. "Алгоритми та структури даних: підручник". — Львів: ЛНУ ім. І. Франка, 2016. — 450 с.
7. Шилов, І. В. "Вступ до програмування та алгоритмізація: навчальний посібник". — Харків: ХНУРЕ, 2019. — 250 с.
8. Дейкстра, Е. "Алгоритми: основи та теорія". — Мюнхен: Видавництво комп'ютерних наук, 2018. — 280 с.
9. Wirth, N. "Algorithms + Data Structures = Programs". — Prentice Hall, 2020. — 400 с.
10. Matlab for Engineers. Holly Moore, Pearson, 2019. — 728 стор.
11. MATLAB: A Practical Introduction to Programming and Problem Solving. Stormy Attaway, Elsevier, 2020. — 624 стор.
12. MATLAB for Electrical Engineers. S. M. Ali, McGraw-Hill, 2020. — 312 стор.
13. Simulink: Simulation and Model-Based Design. MathWorks, 2021. — 550 стор.
14. Introduction to MATLAB for Engineers. William J. Palm III, McGraw-Hill, 2019. — 408 стор.
15. Numerical Methods for Engineers. Steven C. Chapra, Raymond P. Canale, McGraw-Hill, 2020. — 896 стор.
16. MATLAB for Engineers: Applications in Control, Electrical Engineering, and Digital Signal Processing. Muhammad Younus, Springer, 2021. — 500 стор.
17. Поляков Ю. А. "Основи електроніки та мікросхемотехніки". — К.: Лібра, 2015.

18. Стороженко І. В., Стороженко О. В. "Електронні пристрої: теорія і практичні аспекти". — Х.: НТУ «ХПІ», 2018.
19. Hayward, D., Smith, R. "Simulation and Modeling of Electronic Circuits". — McGraw-Hill, 2008.
20. Петрук І. Б. "Моделювання електронних схем у середовищі Electronics Workbench". — К.: Основа, 2017.
21. Electronics Workbench. Документація та допомога.
22. Таненбаум Е. "Структури операційних систем". — К.: Видавничий дім "Києво-Могилянська академія", 2008. — 832 с.
23. Шилдт Г. "C++ для початківців". — М.: Вільямс, 2019. — 752 с.
24. Electronics Workbench. "Документація користувача". — Офіційний сайт Electronics Workbench, 2023.
25. Д.І. Кузнєцов. "Вбудовані системи: принципи та застосування".
26. П. Вольф. "Архітектура мікроконтролерів".
27. Документація Arduino, STM32, ESP32.
28. Таненбаум Е. "Архітектура комп'ютерів: пер. з англ. / Ендрю Таненбаум". — К.: Наукова думка, 2020. — 864 с.
29. Крисюк Ю. С. "Мікропроцесорні системи керування: навчальний посібник" / Ю. С. Крисюк. — Харків: Харківський національний університет радіоелектроніки, 2019. — 312 с.
30. Hennessy J. L., Patterson D. A. "Computer Organization and Design: The Hardware/Software Interface" / John L. Hennessy, David A. Patterson. — 5th ed. — Burlington: Morgan Kaufmann, 2021. — 800 p.
31. Electronics Workbench User Guide. — National Instruments, 2021. — 245 p.