

Міністерство освіти і науки України  
Рівненський державний гуманітарний університет  
Кафедра інформаційних технологій та моделювання

**Кваліфікаційна робота**  
за освітнім ступенем «бакалавр»  
на тему:  
**ТЕМА ДОСЛІДЖЕННЯ**  
**«Інформаційно-навчальні матеріали з програмованої електроніки»**

**Виконав:**

здобувач ІV курсу  
групи ІІЗ-41  
спеціальності 121  
«Інженерія програмного забезпечення»  
Жеребило Василь Федорович

**Науковий керівник:**

к.ф.-м.н. Ляшук Тарас Григорович

## Зміст

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....  | 3  |
| ВСТУП.....  | 4  |
| РОЗДІЛ 1. ОГЛЯД ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ.....  | 6  |
| 1.1. Визначення програмованої електроніки.....  | 6  |
| 1.2. Історія та розвиток програмованої електроніки.....   | 6  |
| 1.3. Важливість програмованої електроніки в сучасному світі.....  | 11 |
| РОЗДІЛ 2. ОСНОВНІ КОНЦЕПЦІЇ ТА ТЕХНОЛОГІЇ.....  | 17 |
| 2.1. Мікроконтролери та мікропроцесори: основні принципи роботи.....  | 17 |
| 2.2. Мови програмування для вбудованих систем.....  | 23 |
| 2.3. Середовища розробки для програмованої електроніки.....   | 24 |
| 2.4. Основні електронні компоненти та їх функції.....   | 31 |
| 2.5. Схемотехніка та дизайн печатних плат.....  | 33 |
| РОЗДІЛ 3: ІНФОРМАЦІЙНО-НАВЧАЛЬНІ РЕСУРСИ.....   | 40 |
| 3.1. Огляд навчальних платформ з програмованої електроніки.....   | 40 |
| 3.2. Розгляд популярних онлайн-ресурсів, відео-уроків, підручників та форумів, присвячених програмованій електроніці..... | 46 |
| РОЗДІЛ 4: РОЗРОБКА НАВЧАЛЬНИХ МАТЕРІАЛІВ.....   | 49 |
| 4.1. Визначення вимог до інформаційно-навчальних матеріалів.....  | 49 |
| 4.2. Вибір мети та аудиторії для навчальних матеріалів.....   | 50 |
| 4.3 Розробка інформаційно навчальних стендів.....   | 50 |
| ВИСНОВКИ.....   | 53 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....   | 54 |
| ДОДАТКИ.....  | 55 |

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

**МК** – Мікроконтролер.

**МП** – Мікропроцесор.

**ЕП** – Електронні пристрої.

**ПЕ** – Програмована електроніка.

**АЛП** – Арифметично-логічний пристрій.

**ОП** – Оперативна пам'ять.

**ІНМ** – Інформаційно-навчальні матеріали.

**ПП** – Печатна плата.

**ІР** – Інтернет речей.

**ЦАП** – Цифро-аналогового перетворювача.

**ВС** – Вбудова система.

**ІСР** – Інтегроване середовище розробки

**ЕОМ** – Електронно-обчислювальна машина.

**СКВ** – Система контролю версій.

**ІС** – Інтегральні мікросхеми.

## ВСТУП

У світі швидкої технологічної революції програмована електроніка стає основним фактором у розвитку сучасного суспільства. Ця галузь не лише сприяє автоматизації та оптимізації процесів, але й відкриває безмежні можливості для реалізації творчих ідей. В даній бакалаврській роботі детально розглянуто галузь програмованої електроніки в усіх її аспектах, зокрема: загальні положення, історію, розвиток та її значення у сучасному світі. Було також проаналізовано основні концепції та технології, які є фундаментом цієї галузі, а також створено інформаційно-навчальні ресурси для полегшення засвоєння концепцій програмованої електроніки (ПЕ). Все це сприяє формуванню повного розуміння та навичок у цій захоплюючій сфері, яка має досить таки важливе значення як в промисловому виробництві так і в народному господарстві.

**Актуальність теми.** Впровадження мікроконтролерів(МК) та мінікомп'ютерів (одноплатних комп'ютерів, мініЕОМ) у масове виробництво та їх широке використання в програмованій електроніці мають значний вплив на розвиток та трансформацію сучасного технологічного світу. Важливість цієї теми можна обґрунтувати з різних точок зору:

- **зниження вартості електронних пристроїв:** впровадження МК дозволило здешевити електронно-обчислювальні машини (ЕОМ) та інші пристрої. Це зробило їх доступнішими для широкого кола споживачів. Тепер електронні пристрої, які раніше були недосяжні з точки зору вартості, стали доступними для багатьох, що сприяє поширенню та використанню сучасних технологій.

- **мініатюризація, енергоефективність та великий обсяг застосувань:** МК та мініЕОМ значно менші за стандартні персональні комп'ютери та споживають значно менше електроенергії. Це дозволяє використовувати їх в різних пристроях, включаючи мобільні телефони, датчики, робототехнічні комплекси, технології Інтернету речей (IP) та інші пристрої з обмеженими ресурсами;

- **розвиток освіти та науки:** з ростом популярності ПЕ, з'явилася потреба у хороших навчальних матеріалах для різних груп людей. Знання про програмування МК стали важливими для молодих інженерів та науковців.. Розробка інформаційно навчальних матеріалів(ІНМ) стає ключовим елементом розвитку освіти та науки в цій галузі.

Загалом, тема "Інформаційно-навчальні матеріали з програмованої електроніки" є надзвичайно важливою та актуальною, оскільки демонструє внесок МК та мініЕОМ у сучасну технологічну еволюцію, розвиток електронної індустрії, прогрес в освіті та науці. Вона також сприяє поширенню цієї інформації серед ширшої аудиторії, роблячи її доступнішою для більшої кількості людей.

**Мета роботи** полягає в розробці (ІНМ) в сфері ПЕ, що визначається необхідністю забезпечення якісної та доступної освіти у галузі ПЕ для різних аудиторій, включаючи студентів, професіоналів, та всіх, хто цікавиться цією технологічною областю.

**Об'єкт дослідження:** програмована електроніка.

**Предмет дослідження:** МК, мініЕОМ, радіодеталі та інша суміжна техніка.

Досягнення мети передбачає виконання наступних **завдань:**

- дослідження історії електронних технологій та їх основні концепції;
- здійснення опису та порівняльної характеристики різноманітних електронних технологій;
- розробка макетів ІНМ.

При цьому, **методами дослідження** виступали:

- пошуково-теоретичний рівень дослідження;
- метод аналізу та порівняння;
- графічна розробка макетів ІНМ.

**Практичне значення дослідження.** Створення навчальних стендів дозволяє учням та студентам отримувати практичні навички на реальних електронних пристроях, що сприяє глибшому розумінню та зацікавленості в

даній темі. Актуальні та інноваційні матеріали сприятимуть не тільки зростанню рівня обізнаності, але і підготовці кваліфікованих фахівців, готових до викликів, які ставлять перед ними сучасні технології.

- апробація та впровадження результатів дослідження;

**Апробація та впровадження результатів дослідження.** Основні положення та результати дослідження обговорювалися та були схвалені на: звітній Міжнародній науковій конференції студентів і молодих науковців з програмування та електроніки. “ПЕРСПЕКТИВИ ВИКОРИСТАННЯ РОБОТОТЕХНІКИ В СУЧАСНОМУ СВІТІ” матеріали доп. III Всеукраїнській науково-практичній конференції, 6-9 грудня 2023 р., – Київ. – с. 196. та на конференції з “ОСОБЛИВОСТІ ВИКОРИСТАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ В ОСВІТНЬОМУ ПРОЦЕСІ” матеріали доп. XVI Всеукраїнській науково-практичній конференції, 1 листопада 2023 р., – Рівне. – с. 25.

- структура роботи.

**Структура та обсяг дисертації.** Дисертація складається із вступу, чотирьох розділів, загальних висновків, списку використаних джерел і додатку; викладена на 71 сторінках друкованого тексту, містить 25 рисунків.

# РОЗДІЛ 1. ОГЛЯД ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ.

## 1.1 ВИЗНАЧЕННЯ ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ

Програмована електроніка – важлива галузь сучасної техніки, яка вивчає розробку та застосування електронних систем, спроможних виконувати різноманітні завдання за допомогою програмного забезпечення. У своєму визначенні, програмована електроніка використовує вбудовані програми та логіку для контролю різних пристроїв, що дозволяє їм пристосовуватися до змінних (внутрішніх/зовнішніх) умов та виконувати різні функції.

У сучасному світі програмована електроніка знаходить своє застосування в найрізноманітніших сферах, починаючи від побутових пристроїв та закінчуючи високотехнологічними промисловими системами. Вона визначає принципово новий підхід до створення ЕП, де програмне забезпечення відіграє вирішальну роль у функціонуванні та управлінні обладнанням.

Розглядаючи визначення ПЕ, необхідно звернутися до ключових аспектів, таких як вбудовані системи, МК та мікропроцесори, що визначають її основні складові та функціональні можливості. В такому контексті важливо також враховувати постійний розвиток технологій, що надає програмованій електроніці безмежні перспективи для подальшого вдосконалення та інновацій[1].

## 1.2 ІСТОРІЯ ТА РОЗВИТОК ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ.

Історія ПЕ є захопливим шляхом від початкових етапів використання електроніки до сучасних високотехнологічних досягнень. Розглядаючи етапи її розвитку, можна виявити ключові події та технологічні вибори, що визначили її сучасний статус.

**Етап 1: перший поклик електроніки.** Період від народження електроніки до початку 20-го століття визначався експериментами та пошуками нових можливостей в галузі електрики та магнетизму. Під час цього етапу

електроніка формувалася як наука і розвивалася в рамках фундаментальних досліджень.

**Початки використання електроніки.** На початку 19 століття, вчені почали вивчати властивості електрики та магнетизму, що відкрило шлях до створення перших електричних пристроїв. Експерименти з електродинамікою та гальванічними явищами привели до створення перших елементів акумуляторів та лампових вакуумних трубок.

**Лампові прилади.** У другій половині 19 століття вчені почали вивчати явище термоемісії та створювати лампові електронні пристрої. Виникнення перших вакуумних ламп дозволило контролювати потік електронів та використовувати їх для підсилення сигналів.

**Розвиток телеграфії та телефонії.** Завдяки розвитку телеграфії та телефонії в середині 19 століття, електроніка стала важливою галуззю для забезпечення довідкових та комунікаційних послуг. Виникнення телеграфних та телефонних мереж вимагало створення нових пристроїв для передачі та підсилення сигналів.

**Виникнення електронних компонентів.** З появою електронних компонентів, таких як діоди та тріоди, в кінці 19 століття, стали можливими нові застосування для регулювання та підсилення електричних сигналів. Це відкрило дорогу до подальших досягнень у створенні ЕП.

На першому етапі розвитку електроніки фокус був спрямований на розуміння фізичних явищ та створення перших ЕП, що визначило фундаментальні засади для подальшого розвитку цієї науки.

**Етап 2: зародження транзисторної епохи.** Другий етап у розвитку ЕП, який охоплює період 1950-60-х років, визначався революційним відкриттям транзисторів та переходом від лампових технологій до напівпровідникових. Цей період був ключовим у зменшенні розмірів та вдосконаленні характеристик ЕП.

**Транзистор – ключовий елемент.** У 1947 році вчені Бардін, Бреттейн та Шоклі представили транзистор, що став ключовим компонентом для



подальшого розвитку електроніки. Транзистори, які використовують властивості напівпровідників, надали можливість контролювати електричний струм, визначаючи напрямок та підсилюючи сигнали.

**Перехід до напівпровідникової технології.** Транзистори відкрили двері для переходу від лампових технологій до напівпровідникових. Це значно зменшило розміри ЕП, збільшивши їхню надійність та знизивши витрати енергії. Виникла можливість створювати менші, легші та більш потужні пристрої.

**Розвиток технології інтегральних мікросхем.** З початку 1960-х років почалася ера інтегральних мікросхем. Це технологічне досягнення дозволило об'єднувати тисячі транзисторів та інших компонентів на одному чіпі, що робило пристрої значно меншими та дешевшими.

**Роль транзисторної технології в розвитку обчислювальної техніки.** Транзисторна технологія стала ключовою для розвитку обчислювальної техніки. Комп'ютери стали більш доступними, потужними та зручними у використанні. Це відкрило нові можливості для застосування ПЕ в різних галузях.

Другий етап розвитку ПЕ визначався переходом до напівпровідникових технологій, що відкрило шлях до створення більш потужних та компактних електронічних пристроїв.

**Етап 3: інтеграція на мікрошкалу.** Третій етап у розвитку ПЕ, який визначався в період від середини 20-го століття і до 19(70-80)-х років, характеризувався переходом до інтегральних мікросхем. Це важливе досягнення дозволило об'єднати значну кількість елементів на одному чіпі, що веде до експоненційного зростання обчислювальної потужності та зменшення розмірів пристроїв.

**Інтегральні мікросхеми: новий рівень інтеграції.** На початку 1960-их років вчені вперше запропонували інтеграцію кількох транзисторів на одному чіпі. Це стало можливим завдяки новим технологіям виготовлення, що

дозволили створювати інтегральні мікросхеми – компактні пристрої, на яких розташовано тисячі транзисторів та інших компонентів.

**Переваги інтегральних мікросхем.** Інтегральні мікросхеми відкрили нові перспективи для розробки електронних пристроїв. Зменшення розмірів та ваги, підвищення надійності та продуктивності, а також зниження витрат на виробництво дозволило виробникам створювати більш потужні, швидкі та доступні пристрої.

**Розширення застосувань мікросхем.** Інтегральні мікросхеми швидко знайшли застосування в різних галузях, включаючи телекомунікації, обчислювальну техніку, медичні пристрої та автоматизацію виробництва. Завдяки швидкому зростанню функціональності та зменшенню вартості виробництва, використання мікросхем стало все більш поширеним.

**Роль у розвитку комп'ютерів.** Розвиток інтегральних мікросхем суттєво вплинув на еволюцію комп'ютерів. Можливість вміщувати тисячі транзисторів на кристалі дала змогу створювати потужні обчислювальні машини зі значно зменшеними розмірами та енергоспоживанням.

Третій етап розвитку ПЕ визначав перехід до інтегральних мікросхем, що стало ключовим фактором у зменшенні розмірів та вдосконаленні функціональності ЕП.

#### **Етап 4: визначення вбудованих систем та мікроконтролерів.**

Четвертий етап у розвитку ПЕ, охоплюючи період 19(70-80)-х років, характеризувався появою вбудованих систем(ВС) та МК. Цей період визначався зростанням компактності, ефективності та універсальності ЕП завдяки використанню МК.

**Вбудовані системи: відділення від загальних комп'ютерів.** У цей період електроніка стала менш залежною від централізованих комп'ютерів, а замість цього, активно використовувала вбудовані системи. Вбудовані системи відзначаються вбудованою програмою та обладнанням, спеціально призначеним для конкретного застосування, що дозволяє ефективно та оптимально вирішувати завдання.

**Мікроконтролери: інтегровані обчислювальні пристрої.** МК, що виникли в цей період, представляли собою інтегровані схеми, які включали в себе обчислювальний блок, пам'ять та пристрої введення-виведення. Ці пристрої були спроектовані для автономної роботи ВС, забезпечуючи їм високий рівень функціональності та гнучкість.

**Застосування мікроконтролерів в пристроях.** МК стали ключовими компонентами в електроніці для різноманітних пристроїв, включаючи побутову техніку, автомобільну електроніку, медичні пристрої, телекомунікаційні системи та багато інших. Їхнє застосування визначалося їхньою здатністю виконувати програмні інструкції, що дозволяло надавати різноманітні функції пристроям без значного збільшення їхніх розмірів.

**Зростання ринку вбудованих систем.** Вбудовані системи та МК активно використовувались в електроніці, забезпечуючи високу продуктивність та надійність. Зростання їхнього застосування призвело до збільшення ринку ВС, що стало ключовим фактором у розвитку ПЕ в цьому періоді.

Четвертий етап розвитку ПЕ був періодом значного зростання використання ВС та МК, що привело до створення більш ефективних та компактних електронних пристроїв.

**Етап 5: ера програмованої електроніки.** П'ятий етап у розвитку ПЕ, який налічує останні десятиліття, характеризується високим рівнем доступності програмування та широким застосуванням в різних галузях. Цей період визначається поширеним використанням МК, розвитком інтернет речей (IP), та зростанням інтеграції програмованих елементів у всі аспекти життя людини.

**Використання в науці та дослідженнях.** Програмована електроніка знайшла широке використання в сфері науки та досліджень, де вона використовується для створення різноманітних дослідницьких пристроїв, сучасних лабораторних установок та автоматизованих систем збору і обробки даних.

П'ятий етап розвитку ПЕ – це ера широкого використання програмованих елементів та МК в різних сферах життя, що суттєво змінює наше оточення та підкреслює важливість програмування в електроніці[1].

### **1.3 ВАЖЛИВІСТЬ ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ В СУЧАСНОМУ СВІТІ.**

В сучасному світі роль електроніки стала неоціненною, охоплюючи майже всі сфери нашого життя. Розповсюдження та використання прогресивних технологій в електроніці визначають не лише рівень комфорту, а й розвиток суспільства в цілому.

З появою та поширенням ЕП, таких як смартфони, комп'ютери, планшети, побутова та промислова електроніка, наше життя стало більш зручним та ефективним. Велика частина економіки тепер залежить від розробки та виробництва електроніки, що сприяє створенню нових робочих місць та економічному зростанню.

Електроніка грає ключову роль у розвитку науки та медицини, дозволяючи вченим та лікарям здійснювати більш точні дослідження та діагностику. В сфері зв'язку електроніка забезпечує надійний та швидкий обмін інформацією, що є невід'ємною частиною сучасного глобального спілкування.

Проте, разом із вагомими перевагами, експансія електроніки породжує виклики, пов'язані із справжньою потребою у сталому використанні ресурсів, управлінні відходами та збереженні екологічної рівноваги. Таким чином, важливо забезпечити баланс між технологічними досягненнями та екологічною стійкістю для сталого розвитку сучасного суспільства.

#### **Забезпечення технологічного прогресу.**

Програмована електроніка виступає не тільки як інструмент для створення нових технічних засобів, але й як важливий стимул для технологічного прогресу в широкому сенсі. Програмована електроніка є імпульсом для безперервного вдосконалення ЕП. Вона стимулює появу нових інновацій, включаючи більш швидкі та продуктивні процесори, нові форми

введення та виведення даних, а також покращені алгоритми обробки інформації.

Розвиток комп'ютерних технологій в програмована електроніці є ключовим елементом у розвитку комп'ютерних технологій. Завдяки постійній оптимізації та вдосконаленню програмованих компонентів, ми спостерігаємо зростання обчислювальної потужності, зменшення розмірів пристроїв та розширення їхніх функціональних можливостей.

Сприяння індустріальному розвитку. Програмована електроніка активно використовується в промисловому виробництві для автоматизації та контролю над процесами. Це призводить до підвищення ефективності, якості продукції та зниження витрат на виробництво.

У сфері медицини програмована електроніка грає важливу роль у розвитку нових медичних технологій. Вона сприяє створенню точних медичних приладів, систем моніторингу та діагностики, що підвищують рівень надання медичної допомоги та збільшують шанси на успішне лікування.

Програмована електроніка постійно розширює межі технологічних можливостей, від вирішення складних завдань штучного інтелекту до розробки передових систем зв'язку та мережі IP.

Економічний розвиток через технології. Використання ПЕ сприяє створенню нових галузей економіки, що базуються на технологіях. Вона не лише стимулює створення нових робочих місць, але і формує інноваційні моделі бізнесу та сприяє економічному розвитку.

### **Полегшення досліджень в різних галузях.**

Однією з основних сфер впливу ПЕ є полегшення проведення досліджень у різних галузях науки та інженерії. Завдяки високій обчислювальній потужності та гнучкості ПЕ, науковці можуть виконувати складні обчислення, моделювання та аналіз даних. Це надає можливість дослідникам здійснювати глибокі та високоточні дослідження у різноманітних областях, включаючи фізику, біологію, хімію та інші наукові галузі.

Програмована електроніка розширює можливості наукового співтовариства, дозволяючи швидше та ефективніше впроваджувати нові методи досліджень. У сучасному світі, де інновації швидко розвиваються, це надзвичайно важливий аспект. Наукові проекти, що використовують програмовану електроніку, можуть включати в себе створення складних моделей, симуляцій та експериментів, що полегшує розкриття нових знань та здобуття дослідницьких відкриттів.

Безумовно, впровадження ПЕ у дослідження створює умови для інноваційних підходів та дозволяє вченим виходити за межі традиційних методів. Це сприяє науковому прогресу та розширює горизонти можливостей для розв'язання великих наукових та технологічних викликів. Програмована електроніка стає важливим інструментом для реалізації амбітних наукових проектів, що сприяє подальшій еволюції сучасної науки.

#### **Участь у створенні передових технологій.**

Однією з ключових сфер впливу ПЕ є її активна участь у створенні передових технологій, що визначають напрямок технічного розвитку. Програмована електроніка є необхідною для розробки та імплементації таких інновацій, як штучний інтелект (ШІ), розпізнавання образів, автономні системи та робототехніка.

Штучний інтелект, що включає в себе навчання машин, аналіз даних та прийняття рішень на основі алгоритмів, стає реальністю завдяки потужності ПЕ. Високоякісні сенсори, потужні обчислювальні системи та ефективне програмне забезпечення є ключовими елементами, що дозволяють створювати інтелектуальні системи, здатні вчитися та вдосконалювати свою продуктивність з часом.

Розпізнавання образів, що базується на алгоритмах машинного зору, використовується в різноманітних сферах, включаючи медицину, виробництво та транспорт. Програмована електроніка дозволяє створювати ефективні системи розпізнавання образів, що забезпечують точність та швидкість у конкретних завдань.

Автономні системи та робототехніка також залежать від ПЕ для реалізації своїх функцій. Високоточні сенсори, системи навігації та управління, що базуються на програмованій електроніці, дозволяють створювати роботів, здатних працювати в різноманітних умовах та виконувати завдання, які раніше вважалися складними чи неможливими.

Узагальнюючи, участь програмованої електроніки у створенні передових технологій формує основу для технологічного прогресу, революціонізуючи різні галузі і покладаючи основи для подальших інновацій та покращень у сучасному технологічному ландшафті.

### **Сприяння міждисциплінарним дослідженням.**

ПЕ відіграє ключову роль у сприянні міждисциплінарним дослідженням, де різні галузі знань об'єднуються для вирішення складних проблем та досягнення нових наукових відкриттів. Це можливо завдяки властивостям програмованої електроніки, яка забезпечує інтеграцію різноманітних технологій та методів з різних галузей науки та техніки.

Міждисциплінарні дослідження, підтримувані програмованою електронікою, відкривають нові можливості для взаємодії між фахівцями з різних сфер. Наприклад, інженери, програмісти, біологи та фізики можуть співпрацювати для вирішення складних біологічних проблем або створення інноваційних медичних технологій. Висока обчислювальна потужність та гнучкість програмованої електроніки дозволяють ефективно обробляти та аналізувати дані з різних джерел, що сприяє комплексному розгляду проблеми.

Інтеграція ПЕ у міждисциплінарні дослідження створює сприятливе середовище для обміну ідеями та співпраці. Такий підхід може призводити до виникнення нових підходів, методів та технологій, які стають результатом синергії знань і ресурсів різних наукових галузей.

Узагальнюючи, сприяння міждисциплінарним дослідженням завдяки програмованій електроніці сприяє створенню інноваційних рішень та сприяє розвитку сучасної науки, визначаючи нові напрямки для розвитку технологій та наукових досліджень

## **Створення інноваційного середовища.**

Вплив ПЕ на створення інноваційного середовища є значущим, оскільки цей аспект визначає не лише технічний прогрес, але й сприяє розвитку новаторських підходів та партнерств в сучасному бізнес-середовищі.

Програмована електроніка допомагає сприяти інноваціям, створюючи сприятливий ґрунт для співпраці між малими та великими підприємствами. Це особливо важливо в контексті розвитку стартапів, де ідеї та технології можуть бути ефективно втілені завдяки доступу до передових розробок у галузі програмованої електроніки.

Створення інноваційного середовища також передбачає активну участь у вищій освіті та наукових дослідженнях. Використання ПЕ в університетах та наукових лабораторіях розширює можливості молодих дослідників для здійснення проєктів та створення нових технологій. Це важливий стимул для розвитку талановитих фахівців у галузі ПЕ та розширення меж наукових досягнень.

Ще однією важливою складовою створення інноваційного середовища є сприяння міжнародному співробітництву. Програмована електроніка дозволяє компаніям та дослідницьким групам співпрацювати в реальному часі, навіть перебуваючи на різних континентах. Це сприяє обміну ідеями, передовими технологіями та розробками, що може призводити до виникнення інноваційних рішень та відкриттів.

Узагальнюючи, програмована електроніка не лише технічно вдосконалює пристрої, але і формує сприятливе середовище для розвитку інновацій, сприяючи співпраці між компаніями, науковими установами та освітніми інституціями. Це сприяє створенню екосистеми, що стимулює творчий та технічний прогрес у сучасному бізнесі та науці.

Усі ці аспекти підкреслюють загальний високий вплив та значення ПЕ в різних сферах життя, роблячи її невід'ємною частиною нашого сучасного світу[1;3;4;5].



## РОЗДІЛ 2. ОСНОВНІ КОЦЕПЦІЇ ТА ТЕХНОЛОГІЇ

### 2.1. МІКРОКОНТРОЛЕРИ ТА МІКРОПРОЦЕСОРИ: ОСНОВНІ ПРИНЦИПИ РОБОТИ.

В соціальній сфері, сферах комунікації (спілкування) та зв'язку. Перше, що спадає на думку в даному контексті, – це комп'ютери, Інтернет і стільникові телефони. Людство вільне у пошуках необхідної інформації, має можливість вийти на зв'язок з бажаним абонентом, незважаючи на місцезнаходження. Також є можливість отримувати дистанційну освіту і об'єднуватися у групи за професійними, соціальними або культурними інтересами. Все це стало можливим у значній мірі завдяки винаходу мікропроцесорів(МП) і створенню мікропроцесорних систем.

**Мікропроцесор** – це мікроелектронний програмований пристрій, що призначений для обробки інформації та керування процесами обміну цією інформацією у складі мікропроцесорної системи (комп'ютера)[10].



*Рис.1.Мікропроцесор.*

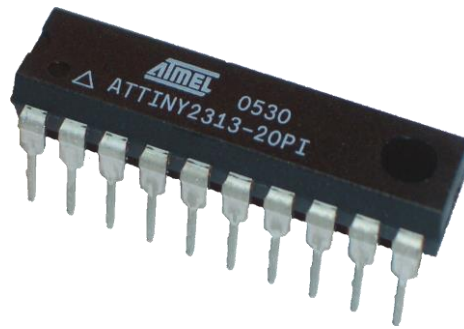
Мікропроцесори виготовляються за допомогою технологій сучасної мікроелектроніки на основі напівпровідникового кристалу. Інформація в

мікропроцесорній системі передається електричними імпульсами. Конструктивно МП виконується у вигляді однієї мікросхеми (інколи – декількох). Мікросхема складається з пластикового або керамічного корпусу, всередині якого міститься мініатюрна напівпровідникова. На цій підкладці лазером “накреслені” усі електронні схеми мікропроцесора. Входи та виходи схеми на підкладці з’єднані з металевими виводами, що розташовані по боках або знизу корпусу мікросхеми[15].

Обов’язковими компонентами мікропроцесора є реєстри, арифметико-логічний пристрій(АЛП) та блок керування. Реєстри призначені для тимчасового зберігання даних, арифметико-логічний пристрій – для виконання арифметичних і логічних операцій Блок керування відповідає за послідовне виконання команд програми та правильне спрямування потоків даних.

МП не може працювати сам по собі. Він є центральною ланкою мікропроцесорної системи, до якої також входять пристрої постійної та оперативної пам’яті, пристрої введення та виведення інформації тощо. Такі мікропроцесорні системи називають комп’ютерами.

Персональний комп’ютер може мати багато застосувань, проте це досить дорогий і громіздкий пристрій. А як же наділити елементами інтелекту побутову техніку, автомобілі, медичні прилади? Як зробити їх “розумними”? Зрозуміло, що у побутовий кондиціонер не можна вмонтувати системний блок звичайного персонального комп’ютера. Це підвищить його вартість у два – три рази. І у складі так званого смарт-телевізора ми не знайдемо окремого персонального комп’ютера у його звичайному вигляді. Для автоматизації такого роду техніки розроблені та виготовляються спеціальні процесорні пристрої – однокристальні МК. Таким чином, *мікроконтролер* – це спеціальний мікропроцесор, що призначений для автоматизації різноманітних пристроїв і керування їх роботою[11].



*Рис.2.Мікроконтролер.*

МК використовують у побутовій техніці, медичних приладах, системах керування ліфтами, телефонах, раціях та інших засобах зв'язку, електронних музичних інструментах та автомагнітолах, комп'ютерній периферії (клавіатурах, джойстиках, принтерах, тощо), світлофорах, автоматичних воротах та шлагбаумах, інтерактивних дитячих іграшках, автомобілях, локомотивах. МК також широко застосовуються в автомобільній електроніці. Наприклад, автомобіль "Peugeot 206" має на борту 27 МК, а в автомобілях високого класу, як наприклад "BMW" сьомої серії, використовується понад 60 МК.

МК, на відміну від мікропроцесора, зазвичай має порівняно невелику розрядність ((8 – 32) бітів) та багатий набір команд маніпулювання окремими бітами. Бітові команди дають змогу керувати дискретним обладнанням (підняти/опустити шлагбаум, увімкнути/вимкнути лампу, нагрівач, запустити/зупинити двигун, відкрити/закрити клапан, тощо). Наявність можливості оперувати окремими бітами, вводити та виводити дискретні сигнали називають "бітовим процесором".

Ще одна з основних відмінностей МК від мікропроцесора полягає у тому, що у складі мікросхеми контролера є усі елементи для побудови простої (а інколи – і досить складної) системи керування. Так, всередині МК є пам'ять даних (ОП), пам'ять програм (постійна пам'ять), генератор тактових імпульсів, таймери, лічильники, паралельні та послідовні порти. Тож система мінімальної конфігурації на основі МК може складатися з блока живлення, безпосередньо мікросхеми контролера та кількох пасивних елементів (резисторів, конденсаторів та кварцового резонатора). І це фактично є ніщо інше, як

одноплатний мінікомп'ютер на основі однієї мікросхеми, придатний для вбудовування до об'єкта керування. Середня вартість системи мінімальної конфігурації складає кількадесят доларів (порівняйте із середньою вартістю персонального комп'ютера).

Слід додати, що кожного року у світі продаються мільярди МК, а середній мешканець розвинутої країни протягом кожного дня десятки разів має справу з МК, які є невід'ємною частиною сучасного технологічного навколишнього середовища.

Окрім мікропроцесорів загального призначення та МК на ринку пропонуються так звані сигнальні процесори, що спеціально призначені для обробки сигналів у режимі реального часу. Вони використовуються у вимірювальних приладах, засобах зв'язку, передачі та відтворення аудіо- і відеопотоків, системах локації, космічній та військовій техніці.

**Сигнальні процесори** характеризуються високою розрядністю і швидкістю, мають в системі команд спеціальні інструкції для реалізації типових алгоритмів обробки сигналів. Також на одному кристалі, окрім власне процесорної частини, реалізуються аналого-цифрові та цифро-аналогові перетворювачі. Цифро-аналоговий перетворювач (ЦАП) замінює неперервний вхідний сигнал відповідним потоком цифрових даних. Далі ці дані обробляються процесорною частиною, після чого за допомогою ЦАП оброблені цифрові дані знову відтворюються в аналоговий сигнал. В такий спосіб сигнальний процесор може поглиблювати чіткість зображення, або, навпаки розмивати його, шифрувати та дешифрувати аудіо- та відеопотоки, відтворювати на екрані віртуальну або доповнену реальність, відслідковувати об'єкти, що рухаються, навіть в умовах великих завад і неповної вхідної інформації.

## **Типи мікропроцесорів**

### **Мікропроцесори загального призначення.**

**Використовуються:** для побудови персональних комп'ютерів, серверів та багатопроцесорних систем.

### **Особливості:**

- висока обчислювальна продуктивність;
- висока розрядність;
- універсальна архітектура.

### **Мікроконтролери.**

**Використовуються:** для реалізації нескладних функцій автоматизації.

### **Особливості:**

- вбудована пам'ять програм та пам'ять даних;
- бітовий процесор;
- таймери, лічильники, порти, інтерфейси;

### **Сигнальні процесори.**

**Використовуються:** для реалізації складних алгоритмів потокової обробки даних у режимі реального часу

### **Особливості:**

- висока обчислювальна продуктивність;
- команди для реалізації типових алгоритмів обробки сигналів;
- вбудовані АЦП, ЦАП або медіа-інтерфейси.

**Інші** (нейрочіпи, секціоновані та гібридні процесори).

**Використовуються:** для реалізації унікальних експериментальних або специфічних систем.

### **Особливості:**

- побудова одного процесора на кількох мікросхемах;
- комбінація кількох видів процесорів в одному виробі;
- специфічна архітектура.

**Програмований логічний контролер (ПЛК)** – це спеціалізована мікропроцесорна система, що використовується для автоматизації технологічних процесів та загальнопромислових установок і комплексів (конвеєрів, рольгангів, підйомних кранів, подрібнювачів, млинів, класифікаторів, змішувачів, пакувальників, робототехнічних та гнучких виробничих комплексів тощо).

Тобто, основна сфера застосування ПЛК – це сфера промислового виробництва. Проте вони також використовуються для автоматизації будівель (контроль доступу до приміщення, керування освітленням, обігрівом, вентиляцією та кондиціонуванням повітря, керування ліфтами, ескалаторами тощо). Також ПЛК можуть бути застосовані для створення мікроклімату в тепличному господарстві, на птахофабриках, тваринницьких фермах.

Апаратна та програмна уніфікація ПЛК дає можливість легко переходити на контролери іншого виробника, переносити програми з однієї системи на іншу. Це підвищує гнучкість систем автоматизації, сприяє конкурентному інноваційному розвитку ринку.

МК та мікропроцесори є ключовими елементами ПЕ. МП центральний обчислювальний блок, що відповідає за виконання операцій та керування системою. МК, у свою чергу, включає в себе не лише процесор, а й різні периферійні пристрої, такі як таймери, порти вводу-виводу, інтерфейси зв'язку, що робить його ідеальним для ВС.

**Основні принципи** роботи полягають у виконанні інструкцій, які подаються від програми. МК реагують на вхідні сигнали, обробляють їх та виконують відповідні дії. Мови програмування, такі як С, С++, або асемблери, використовуються для написання програм, які керують роботою цих пристроїв.

**Принципи МП** включають в себе:

- **Інструкційний цикл:** МП виконує інструкції, які подаються йому програмою, в інструкційному циклі. Це включає отримання інструкції, декодування, виконання та запис результату.
- **Регістри:** МП має різні регістри для зберігання тимчасових даних та адрес. Наприклад, реєстр інструкцій (IR) зберігає поточну виконувану інструкцію.
- **Арифметично-логічний блок (АЛБ):** Виконує операції додавання, віднімання, множення, ділення та логічні операції над даними.
- **Оперативна пам'ять:** Тимчасово зберігає дані та інструкції для швидкого доступу МП.

Отже МК це спеціалізований вид мікросхеми, яка об'єднує в собі процесор, пам'ять та периферійні пристрої для ВС. **Принципи роботи МК** включають:

- **Вбудовані периферійні пристрої:** МК містить в собі різні периферійні пристрої, такі як таймери, порти вводу-виводу, АЦП (аналогово-цифровий перетворювач), що робить його придатним для керування різними сенсорами та пристроями.
- **Низьке споживання енергії:** Багато МК оптимізовані для роботи в енергоефективних ВС, що робить їх ідеальними для мобільних пристроїв та батарейних пристроїв.
- **Програмується на мовах високого рівня:** Для програмування МК можна використовувати мови високого рівня, такі як С або С++, що полегшує розробку ВС.

Загальна ідея полягає в тому, що МК та МП є ключовими компонентами програмованої електроніки, здатними виконувати інструкції програми та керувати різними аспектами ВС. Вони є основою для створення ефективних та потужних ЕП у сучасному технічному світі[].

## **2.2. МОВИ ПРОГРАМУВАННЯ ДЛЯ ВБУДОВАНИХ СИСТЕМ.**

Мови програмування для ВС відіграють визначальну роль у розробці програмного забезпечення, яке вбудовується безпосередньо в електронні пристрої. Вибір правильної мови програмування має критичне значення для ефективності та надійності ВС. Також вибір мови програмування для ВС є стратегічним рішенням, оскільки від цього вибору залежить ефективність та надійність розробки вбудованого програмного забезпечення. Вибір мови для ВС залежить від конкретних вимог проекту та обмежень, і не існує універсальної “кращої” чи “гіршої” мови. Замість цього, інженерам слід враховувати ряд властивостей та характеристик, які визначають ефективність мови для конкретного застосування. Розглянемо деякі ключові аспекти та питання, пов'язані із мовами програмування для ВС.

- **Властивості та Характеристики Мов**

Вибір мови програмування залежить від вимог конкретного проекту. Такі мови як C та C++, є популярними у вбудованому програмуванні завдяки своїй ефективності та близькому до апаратного рівня програмуванню. З іншого боку, мови високого рівня, такі як Python, можуть забезпечити більшу гнучкість на етапі розробки, але можуть мати обмеження в продуктивності.

- **Ефективність та Обмеження Ресурсів**

Вбудовані системи часто працюють з обмеженими ресурсами, такими як пам'ять та потужність обчислень. Мова програмування повинна бути обрана так, щоб забезпечити оптимальне використання ресурсів. Мови з низьким рівнем абстракції, наприклад, асемблери або C, можуть бути вибором для оптимізації роботи з обмеженими ресурсами.

- **Розробка та Спільнота Розробників**

Важливо враховувати наявність інструментів розробки та підтримку спільноти розробників для обраної мови програмування. Наявність надійних інструментів спрощує розробку та підтримку ВС.

- **Надійність та Безпека**

У ВС надійність та безпека є важливими аспектами. Мова програмування повинна дозволяти вирішувати завдання, пов'язані з обробкою великої кількості даних, забезпечуючи при цьому високий рівень безпеки.

- **Легкість Використання та Швидкість Розробки**

У випадку проектів з короткими строками та великою кількістю ітерацій важливо враховувати легкість використання мови та швидкість розробки. Вибір мови з великою кількістю готових бібліотек та фреймворків може прискорити процес створення ВС.

Отже вибір мови програмування для ВС повинен бути обдуманим та залежати від конкретних вимог проекту, ресурсів та відомостей розробників. Врахування цих факторів допомагає забезпечити оптимальну продуктивність та функціональність ВС.



## 2.3. СЕРЕДОВИЩА РОЗРОБКИ ДЛЯ ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ

Середовище розробки є ключовим елементом у створенні ПЕ, оскільки воно надає інженерам і розробникам інструменти для створення, налагодження та тестування вбудованого програмного та апаратного забезпечення. Розглянемо основні аспекти та характеристики середовища розробки ПЕ

**Інтегроване середовище розробки (ICP).** Інтегровані середовища розробки (ICP) є ключовими інструментами для розробки програмованих електронних продуктів. Він поєднує в собі набір інструментів, необхідних для написання, налагодження та тестування коду для МК і ВС. Розглянемо основні особливості та переваги інтегрованих середовищ розробки:

### **PlatformIO**

- **Основні функції:** PlatformIO – це відкрите Інтегроване Середовище Розробки, спеціально призначене для вбудованих систем, МК та МП.
- **Підтримка мікроконтролерів:** Підтримує широкий спектр МК від різних виробників, таких як Arduino, ESP8266, STM32, AVR та інші.
- **Інтеграція з платформами:** Взаємодіє з різними інтеграційними платформами, такими як Arduino, PlatformIO, і надає легкість у виборі платформи для розробки.

### **Arduino ICP**

- **Простота використання:** Розроблено для простоти використання, зокрема для початківців, маючи досить простий інтерфейс.
- **Підтримка багатьох платформ:** Підтримує широкий вибір платформ, включаючи оригінальні плати Arduino, ESP8266, ESP32, і інші.
- **Бібліотеки та зразки коду:** Має велику кількість бібліотек та зразків коду, що спрощує розробку.

### **MPLAB X ICP**

- **Для мікроконтролерів Microchip:** Розроблене спеціально для МК виробника Microchip, таких як PIC та dsPIC.

- **Відлагодження та симуляція:** Надає інструменти для відлагодження та симуляції коду на рівні апаратури.
- **Підтримка засобів розробки Microchip:** Інтегрується з іншими засобами розробки від Microchip, що полегшує роботу з їхнім обладнанням.

### **Keil**

- **Спеціалізоване середовище для ARM:** Орієнтоване на розробку для МК ARM, таких як STM32, NXP, та інші.
- **Підтримка іншого обладнання:** Надає можливості для роботи з різними МК та апаратурою, що базується на архітектурі ARM Cortex-M.

### **IAR Embedded Workbench**

- **Для професійної розробки:** Використовується для професійної розробки вбудованих систем та МК.
- **Висока швидкість компіляції:** Має високоефективний компілятор, що дозволяє швидко розробку та тестування.

Інтегровані середовища розробки надають розробникам зручні інструменти для розробки програмованих електронних продуктів, дозволяючи їм ефективно працювати над різними проектами та платформами. Кожен з них має свої особливості, і вибір залежить від конкретних потреб забудовника та вимог проекту[13].

**Мови програмування та фреймворки для вбудованих систем.** Як було вже зазначено, вибір мови програмування та фреймворку залежить від конкретних вимог проекту, типу вбудованої системи та можливостей МК чи МП. Розглянемо основні мови програмування та фреймворки, які використовуються для розробки програмованої електронік[13]:

### **Мови програмування.**

#### **Python:**

- *Простота та читабельність:* Python є високорівневою мовою з простим синтаксисом, що полегшує розробку та читання коду.

- *Інтерпретована мова:* Зручна для швидкого прототипування, але може бути менш ефективною в порівнянні з мовами, компільованими в машинний код[14].

#### **C та C++:**

- *Ефективність та продуктивність:* Мови C та C++ володіють високою продуктивністю та низьким рівнем абстракції, що дозволяє точно керувати ресурсами вбудованих систем.

- *Наявність компіляторів:* Для більшості МК доступні компілятори C/C++.

#### **Java:**

- *Платформонезалежність:* Можливість запуску на різних апаратних платформах за допомогою віртуальної машини.

- *Багатозадачність:* Вбудована підтримка багатопотоковості.

### **Фреймворки та середовища виконання.**

#### **PlatformIO:**

- *Універсальна розробка:* Підтримує багато МК та МП, надаючи єдиний інтерфейс розробки.

- *Інтеграція з багатьма іншими середовищами:* Забезпечує інтеграцію з різними ІСР та фреймворками.

#### **STM32CubeMX:**

- *Графічний конфігуратор та генератор коду:* Забезпечує зручний інтерфейс для налаштування параметрів МК та генерації початкового коду.

- *Підтримка Великої Кількості Платформ:* Дозволяє розробляти для МК STM32.

#### **MicroPython:**

- *Інтерпретована Python на мікроконтролерах:* Дозволяє використовувати Python для програмування вбудованих систем.

- *Високорівневі Операції:* Забезпечує високорівневий доступ до функцій МК через Python.

## **Arduino Framework:**

- *Спрощений розвиток:* забезпечує високорівневий інтерфейс та багато готових бібліотек, що спрощує розробку для платформ Arduino та сумісних.
- *Спрощене управління ресурсами:* Дозволяє легко керувати платформою без деталей низькорівневого програмування.

Вибір мови програмування та фреймворка залежить від конкретних завдань проекту, обраного МК, а також від власних навичок розробника. Мови C та C++ загальнопризнані своєю ефективністю на різних апаратних платформах, тоді як використання високорівневих мов, таких як Python, може спростити розробку для деяких застосувань. Фреймворки, такі як Arduino та STM32CubeMX, дозволяють швидше створювати прототипи та робити розробку менш трудомісткою.

## **Симуляція та відлагодження в розробці програмованої електроніки.**

Симуляція та відлагодження є ключовим етапом у розробці програмованої електроніки, де інженер може тестувати та відлагоджувати програмний код перед його впровадженням на реальну апаратуру. Симуляція дозволяє моделювати роботу вбудованих систем та МК, враховуючи їх взаємодію з електронними компонентами та зовнішнім середовищем. Це сприяє виявленню та виправленню помилок до фізичної реалізації.

Відлагодження дозволяє інженерам виявляти та виправляти помилки в програмному коді, а також відстежувати виконання програми для полегшення виявлення проблем. Деякі інтегровані середовища дозволяють відлагоджувати код на реальних МК або використовувати імітацію введення/виведення для тестування реакції системи на різні умови.

Інструменти, такі як **Simulide**, надають можливість симулювати електронні схеми перед їх фізичною реалізацією, що полегшує розробку та зменшує ризик помилок. Загальною метою цих процесів є забезпечення ефективного та надійного функціонування вбудованих систем в реальних умовах експлуатації.

**Підтримка мікроконтролерів та мікропроцесорів у розробці програмованої електроніки.** Підтримка МК та МП грає важливу роль у розробці програмованої електроніки, де інженерам необхідно ефективно взаємодіяти з вбудованими системами. Розглянемо основні аспекти цього процесу:

## **ІСР.**

### **Підтримка різних мікроконтролерів:**

- *Розширені Бібліотеки та Шаблони Коду:* Інтегровані середовища розробки надають багатофункціональні бібліотеки та шаблони коду, спеціально розроблені для різних МК. Це полегшує створення програмного забезпечення для конкретних пристроїв.

### **Симуляція на рівні апаратури:**

- *Можливість відлагодження коду.* Деякі ІСР дозволяють відлагоджувати програмний код безпосередньо на реальних МК, що полегшує виявлення та виправлення помилок.

## **Інструменти розробки та програмування.**

### **Інструменти прошивки та завантаження:**

- *Прошивка та завантаження коду.* Інструменти дозволяють ефективно прошивати програмний код на МК, забезпечуючи правильну роботу вбудованих систем.

### **Підтримка мов програмування:**

- *Можливість використання різних мов.* Інженери можуть використовувати різні мови програмування для роботи з МК.

## **Відкриті засоби розробки.**

### **Відкриті засоби для спільної розробки:**

- *Спільна робота та обмін кодом.* Використання відкритих інструментів сприяє спільній роботі команди розробників, обміну кодом та вирішенню спільних завдань.

### **Спільнота розробників:**

- *Підтримка та знання спільноти.* Багато відкритих інструментів мають активні спільноти розробників, що дозволяє отримати підтримку, обговорення та рекомендації від досвідчених фахівців.

### **Аналіз та тестування.**

#### **Інструменти для аналізу коду:**

- *Автоматизовані засоби аналізу.* Деякі ІСР включають в себе засоби автоматизованого аналізу коду, що допомагає виявити потенційні проблеми та оптимізувати код.

#### **Можливості тестування:**

- *Інтеграція засобів тестування.* Інструменти розробки часто інтегруються з засобами автоматизованого тестування, дозволяючи ефективно перевіряти функціональність та надійність програмного забезпечення.

В цілому, підтримка МК та МП у розробці програмованої електроніки включає широкий спектр інструментів, які сприяють ефективній розробці, відлагодженню, прошивці та тестуванню вбудованих систем.

**Системи контролю версій.** (СКВ) є необхідним інструментом для ефективного керування вихідним кодом та спільної роботи над проектом. Дві з найбільш популярних СКВ, які використовуються у сфері програмованої електроніки, - Git і SVN, пропонують розробникам зручні та потужні інструменти для керування версіями.

**Git** є розподіленою системою контролю версій, що дозволяє кожному розробнику мати повністю функціональну копію репозиторію. Він використовує концепцію гілок, що дозволяє створювати окремі лінії розвитку для різних функціональностей. Гібридна модель зберігання (індекс для підготовки змін і об'єктний каталог для зберігання даних) забезпечує ефективне керування проектами будь-якої складності. Однією з ключових переваг Git є можливість працювати офлайн і дозволяти розробникам фіксувати зміни локально, а потім синхронізувати їх з віддаленим репозиторієм.

**SVN** (Apache Subversion). SVN є централізованою системою контролю версій, де весь вихідний код та його історія зберігаються на центральному

сервері. Він використовує принцип "клієнт-сервер", де розробники взаємодіють із центральним сервером. У порівнянні з Git, SVN простіший у використанні, але може виникнути конфлікт при одночасних змінах у різних частинах коду. Він також використовує номери версій для ідентифікації змін.

Вибір між **Git** та **SVN**. Вибір між Git і SVN залежить від конкретних потреб проекту. Git частіше використовується в сучасних розробках завдяки своїй розподіленій моделі, гнучкості та швидкості. В той час як SVN залишається популярним в окремих випадках, особливо коли централізована модель керування версіями відповідає потребам проекту.

Загалом, обидві системи контролю версій допомагають розробникам ефективно співпрацювати, відстежувати зміни та забезпечувати стабільність проекту на будь-якому етапі його розвитку.

Середовища розробки для програмованої електроніки відіграють важливу роль у створенні та оптимізації вбудованого програмного забезпечення та апаратури. Різноманітні інтегровані середовища розробки, такі як PlatformIO, Arduino IDE, MPLAB X, Keil, IAR Embedded Workbench, забезпечують інженерам зручний інтерфейс та необхідні інструменти для роботи з різними МА та МП.

Мови програмування, такі як C, C++, та Python, разом із фреймворками, наприклад, Arduino Framework та MicroPython, надають ефективний спосіб реалізації функціональності для вбудованих систем. Симуляційні інструменти, такі як Proteus та LTspice, дозволяють моделювати та відлагоджувати електричні схеми, сприяючи уникненню помилок на етапі розробки.

Підтримка МК та МП здійснюється через різноманітні інструменти, такі як Arduino, Raspberry Pi, STM32CubeMX, що надають розробникам необхідні інструменти для роботи з конкретними пристроями.

Загальний інструментарій, що включає системи контролю версій, інструменти вимірювання та осцилографію, а також можливості співпраці та комунікації через онлайн-платформи, дозволяє ефективно ведення розробки та обмін знаннями в спільноті розробників[8].

Вибір того чи іншого середовища розробки залежить від конкретних вимог проекту, навичок розробників та специфікацій пристрою, проте в цілому, інструментарій, який пропонується, забезпечує необхідність для успішного впровадження програмованої електроніки.

## **2.4. ОСНОВНІ ЕЛЕКТРОННІ КОМПОНЕНТИ ТА ЇХ ФУНКЦІЇ.**

Основні електронні компоненти в програмованій електроніці грають важливу роль у створенні та функціонуванні ЕП. Електронні компоненти є елементарними електронними елементами або електронними деталями, як правило, упаковані в дискретній формі з двома або більше сполучними виводами або металевими накладками.

Електронні компоненти призначені для з'єднання між собою, як правило, шляхом пайки до друкованої плати для створення електронної схеми з певною функцією (наприклад, підсилювач, радіоприймач, генератор, бездротовий зв'язок)[2].

**Електронні компоненти бувають 2-ох типів : активні та пасивні електронні компоненти .**

- ***Пасивні електронні компоненти*** – це ті, які не мають підсилення або спрямованості. Наприклад, резистори, конденсатори, діоди, індуктори тощо.

- ***Активними компонентами*** є ті, що володіють належністю або спрямованістю. Наприклад: транзистори, інтегральні схеми, логічні елементи та інші .

Розглянемо ключові компоненти та їх функції:

- **Резистори .**

Резистори використовуються для обмеження потоку електричного струму в колі. Їх основна функція – створення певного опору, що дозволяє контролювати напругу та струм у схемі.

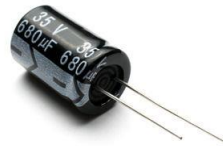




*Рис.3.Резистори.*

- **Конденсатори.**

Конденсатори є компонентами, які здатні зберігати електричний заряд. Вони застосовуються в якості накопичувачів енергії електричного поля, зокрема для вирівнювання напруги, фільтрації сигналів та реактивного опору в електричних колах.



*Рис.4.Конденсатори.*

- **Індуктивності.**

Індуктивності, відомі також як індуктивні котушки, використовуються для накопичення енергії у магнітному полі. Зазвичай застосовуються для фільтрації сигналів, стабілізації струму та в інших сценаріях, де важливий магнітний ефект.



*Рис.5.Котушка індуктивності*

- **Транзистори.**





*Рис.8.Кварцовий резонатор.*

- **Джерела живлення.**

Джерела живлення, такі як батареї або джерела живлення змінного струму, постачають енергію для ЕП. Вони є необхідним елементом для живлення різноманітних пристроїв від портативної електроніки до великих систем.

- **Роз'єми та конектори.**

Роз'єми та конектори використовуються для з'єднання електронних компонентів та плат. Це важливий аспект при розробці та монтажі ЕП, забезпечуючи надійне електричне з'єднання між різними частинами схеми.



*Рис.9. Перехідник з роз'єму PH2.0 на штирьові конектори (4 pin)*

Основні електронні компоненти, представлені у програмованій електроніці, є важливими будівельними блоками для створення та оптимізації різноманітних ЕП. Кожен компонент виконує унікальну функцію, сприяючи взаємодії та вирішенню специфічних завдань у схемах та платах.

Комбінація цих компонентів в схемі дає можливість інженерам створювати інноваційні та ефективні рішення в електроніці, розширюючи можливості від простих пристроїв до складних, від аналогових до цифрових. Розуміння функцій кожного компонента є ключовим для успішного проектування та оптимізації електронних систем.

## 2.5 СХЕМОТЕХНІКА ТА ДИЗАЙН ПЕЧАТНИХ ПЛАТ.

Схемотехніка та дизайн печатних плат(ПП) є ключовими етапами у розробці ЕП і систем. Обидві ці області знаходяться на стику інженерії та дизайну, і вони тісно пов'язані між собою у процесі створення функціональних ЕП.

**Схемотехніка** є першим і важливим етапом в розробці програмованої електроніки, де інженери створюють графічне представлення електричної схеми пристрою або системи. Інженери використовують спеціальні програми для визначення з'єднань між компонентами, стабільності схеми та електричних характеристик. Результатом цього етапу є схема, яка служить основою для подальшого дизайну печатної плати. Цей процес визначає взаємозв'язки між всіма компонентами та їх функціональність.

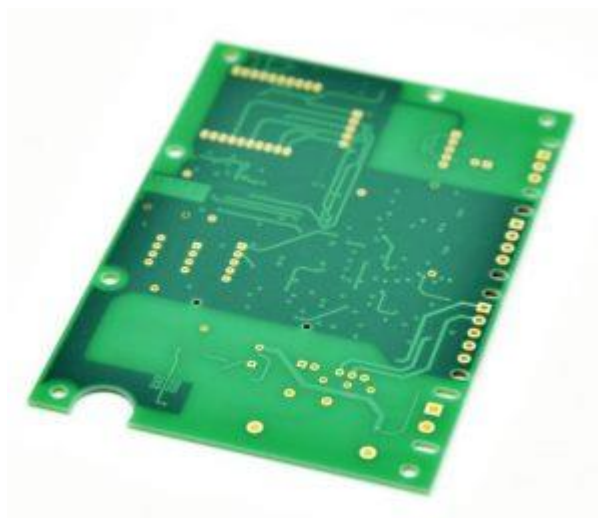
- **Визначення функцій та вимог.** Інженери аналізують завдання та вимоги до проекту, визначаючи функції, які повинні виконувати електронні компоненти.
- **Вибір компонентів.** Відбираються компоненти, які найкраще відповідають вимогам проекту. Це включає вибір МК, датчиків, інтегральних схем, резисторів, конденсаторів і т.д.
- **Побудова електричної схеми.** За допомогою спеціальних програм для проектування схем інженери розміщують та з'єднують компоненти, визначаючи напрямки потоку сигналу та взаємозв'язки між ними.
- **З'єднання та сигнали.** Визначаються шляхи для електричних з'єднань між компонентами. Враховуються електромагнітна сумісність, електричний потенціал і сигнальні параметри.
- **Створення блок-схем.** Велика схема може бути поділена на блоки, що спрощує аналіз та вивчення окремих частин схеми. Це полегшує співпрацю та розуміння проекту різними членами команди.
- **Аналіз та виправлення.** Проводиться аналіз схеми на предмет можливих проблем, таких як конфлікти сигналів, неправильні з'єднання,

електромагнітні перешкоди. Здійснюється виправлення помилок та оптимізація схеми.

- **Визначення характеристик.** Встановлюються електричні характеристики схеми, такі як напруги, струми, опори, для подальших розрахунків та аналізу.

- **Документація.** Завершується створенням технічної документації, яка включає в себе схему, опис компонентів та їх функцій, електричні параметри та інші важливі відомості.

Дизайн ПП включає в себе розміщення компонентів та визначення слідів для провідників на поверхні плати. Цей процес враховує електричні та теплові властивості компонентів, оптимізуючи розташування для максимальної ефективності та надійності. Інженери враховують електромагнітну сумісність, теплові втрати, імпедансні властивості, інші параметри для створення оптимального дизайну ПП.



*Рис.10.Печатна плата.*

Дизайн ПП є критичним етапом у розробці програмованої електроніки, де створюється конкретний фізичний об'єкт для розміщення та з'єднання всіх електронних компонентів. Процес дизайну ПП включає в себе кілька ключових етапів:

- **Вибір розмірів та форми (ПП).** Визначаються габаритні розміри та форма ПП відповідно до вимог проекту та обмежень просторових характеристик пристрою.

- **Розміщення компонентів.** Компоненти, розроблені на етапі схемотехніки, розміщуються на поверхні ПП з врахуванням електричних та термічних параметрів.
- **Виведення з'єднань.** Доріжки для електричних з'єднань розташовуються так, щоб мінімізувати електричний опір та електромагнітні перешкоди, а також забезпечити електричну і термічну стійкість.
- **Урахування шляхів сигналу.** Визначаються шляхи для критичних сигналів, забезпечуючи їх правильну імпедансну характеристику та уникнення перешкод.
- **Створення мультишарових печатних плат (якщо потрібно).** У складних проектах використовуються мультишарові ПП для поліпшення електричних та термічних характеристик.
- **Розміщення діодів та конденсаторів.** Деякі компоненти, такі як діоди та конденсатори, розміщуються так, щоб забезпечити стабільність електричних параметрів.
- **Розміщення зовнішніх з'єднань.** Розміщуються контактні площадки для зовнішніх з'єднань, таких як роз'єми, для спрощення з'єднання пристрою з іншими системами чи обладнанням.
- **Термічний дизайн.** Враховується теплова продукція електроніки, і розташування компонентів та вентиляційні отвори обираються для ефективного охолодження.
- **Визначення виділення печатної плати.** Визначаються місця для вирізання ПП або створення отворів для легшого використання в кінцевому виробі.
- **Проведення перевірок та виправлень.** Всі аспекти дизайну перевіряються за допомогою спеціалізованих програм та виправляються можливі помилки.
- **Створення технічної документації.** Створюється технічна документація, що містить гербарій, виготовлення, вимірювання та інші характеристики ПП для виробництва.

Дизайн ПП грає ключову роль у забезпеченні надійності, функціональності та виробничої ефективності ЕП.

До створення схемотехніка та печатних плат, можна віднести ще декілька аспектів:

**Вибір компонентів та розміщення.** Інженери враховують функціональні та електричні вимоги проекту, обираючи відповідні компоненти. Розміщення цих компонентів на ПП визначається оптимізацією простору, мінімізацією взаємних впливів та полегшенням виробництва.

**Провідникова система та сліди.** Визначають шляхи провідників для забезпечення ефективного електричного з'єднання між компонентами. Оптимізація шляхів провідників важлива для зниження імпедансу, мінімізації втрат сигналу та забезпечення електромагнітної сумісності.

**Перевірка та симуляція.** Перед виготовленням прототипу проводиться перевірка дизайну за допомогою спеціалізованих програм та симуляторів. Це дозволяє виявити та виправити можливі проблеми ще на етапі проектування

**Виготовлення Прототипу та Тестування.** Після завершення дизайну інженери створюють прототип ПП, який потім тестується для підтвердження його працездатності та відповідності вимогам проекту.

Схемотехніка та дизайн печатних плат представляють собою важливий етап у розробці ЕП, де інженери формують структуру схеми та геометрію ПП. Схемотехніка включає у себе створення графічного зображення всіх елементів схеми та їх взаємодій, що служить основою для наступного етапу – дизайну ПП.

Дизайн ПП, у свою чергу, охоплює розміщення компонентів та визначення слідів для провідників на поверхні плати. Цей процес враховує електричні та теплові властивості компонентів, оптимізуючи їхнє розташування для досягнення максимальної ефективності та надійності. Важливим етапом є вибір компонентів, їх розміщення та оптимізація провідникової системи для забезпечення ефективного електричного з'єднання.

Процес перевірки та симуляції перед виготовленням прототипу дозволяє виявити та виправити можливі проблеми ще на етапі проектування. Виготовлення прототипу та його тестування завершують цей етап, підтверджуючи працездатність та відповідність дизайну вимогам проекту.

Усі ці аспекти взаємодіють для створення оптимального та функціонального електронного пристрою, де якість схемотехніки та дизайну печатної плати є визначальними факторами для успішної реалізації проекту.



## РОЗДІЛ 3: ІНФОРМАЦІЙНО-НАВЧАЛЬНІ РЕСУРСИ

### 3.1. ОГЛЯД НАВЧАЛЬНИХ ПЛАТФОРМ З ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ

Існує ряд навчальних платформ, спрямованих на вивчення програмованої електроніки, що надають інженерам, студентам та ентузіастам доступ до засобів для розробки та тестування ЕП. Ось огляд деяких із них:

**Arduino** є однією з найпопулярніших навчальних платформ. Вона базується на МК AVR або ARM та використовує мову програмування на основі C/C++. Arduino надає широкий вибір модулів та датчиків, а також має активну спільноту користувачів.

Курси з Arduino в навчанні – це структуровані програми, які надають учням та ентузіастам систематичний набір знань та практичних навичок щодо роботи з платформою Arduino. Ці курси мають різні рівні складності, починаючи від основ та закінчуючи більш складними проектами. Ось деякі ключові аспекти, які можна знайти в курсах Arduino:

- **Основи Arduino.** Курси починаються з основ, включаючи вступ до платформи Arduino, огляд її складових, вивчення мови програмування Arduino (зазвичай, мова C/C++), та ICP.
- **Робота з різними сенсорами та периферійними пристроями.** Курси дозволяють учням ознайомитися з різними сенсорами, модулями та периферійними пристроями, які можна використовувати з Arduino. Це може включати роботу з датчиками світла, температури, руху, LCD-дисплеями, реле, та іншими компонентами.
- **Створення проектів.** Курси сприяють розвитку практичних навичок через створення конкретних проектів. Це може бути все від простих світлодіодних маяків до складніших систем автоматизації, які використовують різноманітні сенсори та актуатори.

- **Відладка та оптимізація.** Учні вивчають методи відладки програмного забезпечення для Arduino, а також стратегії оптимізації коду для забезпечення ефективної роботи на обмежених ресурсах МК.

- **Застосування в реальних проектах.** Курси дозволяють учням використовувати набуті знання у реальних сценаріях, що сприяє розвитку навичок роботи з Arduino в індустрії та реальних проектах.

Курси Arduino у навчанні сприяють не тільки освоєнню конкретних навичок, але і розвитку критичного мислення, проблемного підходу та творчості в інженерії та програмуванні.

**Raspberry Pi** – це невелика одноплатна система, яка працює на базі ARM. Хоча вона в основному використовується як міні-комп'ютер, Raspberry Pi також може взаємодіяти з електронікою через GPIO. Це відмінний інструмент для вивчення програмування та роботи з різними сенсорами.

Програма для навчання Raspberry Pi може бути надзвичайно корисною для вивчення та розуміння цього одноплатного комп'ютера та його можливостей. Ось деякі ключові аспекти та переваги цієї програми в навчанні Raspberry Pi:

- **Освоєння одноплатного комп'ютера.** Програма надає можливість зазирнути всередину роботи Raspberry Pi. Вона допомагає користувачам зрозуміти апаратні та програмні аспекти цього одноплатного комп'ютера, включаючи процесор, порти вводу/виводу (GPIO), модулі підключення, інтерфейси та інше.

- **Освіта в галузі програмування.** Програма для Raspberry Pi часто містить навчальні матеріали та завдання з програмування. Користувачі можуть вивчати мови програмування, такі як Python, які широко використовуються в екосистемі Raspberry Pi.

- **Робота з GPIO та електронікою:** Raspberry Pi має порти GPIO, які можна програмувати для взаємодії з електронними компонентами. Програми для навчання дозволяють користувачам вивчати основи електроніки та створювати власні проекти, використовуючи GPIO.

- **Створення проектів та застосувань.** Програма може включати в себе вправи та завдання для створення певних проектів, таких як веб-сервер, медіацентр, система моніторингу тощо. Це дозволяє студентам застосовувати знання у практичних сценаріях.

- **Навчання інтернету речей (IP).** З використанням Raspberry Pi можна навчати концепції Інтернету Речей, які стають все більш важливими в сучасному світі. Програми можуть включати завдання з підключення Raspberry Pi до мережі та взаємодії з іншими IP-пристроями.

- **Практичні лабораторії та симуляції:** Деякі програми можуть надавати можливість для віртуальних лабораторій та симуляцій, де користувачі можуть експериментувати з Raspberry Pi без необхідності фізичного обладнання.

- **Спільнота та ресурси:** Користувачі отримують доступ до активної спільноти, форумів, та додаткових освітніх ресурсів. Це дозволяє обмінюватися досвідом, отримувати допомогу вирішення проблем та знаходити натхнення для нових проектів.

Загалом, програми для навчання Raspberry Pi сприяють розвитку технічних навичок, знань в області програмування та робототехніки, а також створюють платформу для творчого застосування отриманих знань.

**Micro:bit** розроблений як навчальний інструмент для дітей. Він включає в себе вбудований дисплей, акселерометр, магнітний компас та інші сенсори. Micro:bit може програмуватися за допомогою блок-мови, Python та інших мов.

Програма для навчання Micro:bit є дуже корисною для студентів та учнів, оскільки ця платформа пропонує низку можливостей для навчання та розвитку різноманітних навичок. Ось кілька ключових аспектів, які роблять її корисною в навчанні:

- **Вивчення основ програмування.** Micro:bit пропонує простий та доступний спосіб вивчення основ програмування. З використанням блокових мов програмування, таких як Scratch, користувачі можуть створювати програми, не використовуючи складних текстових мов.

- **Освоєння основ електроніки.** Платформа має вбудовані датчики, світлодіоди та кнопки, що дозволяє студентам вивчати основи електроніки, працюючи над проектами та експериментами.

- **Розвиток креативності.** Micro:bit може бути використаний для створення різноманітних проектів, що сприяє розвитку творчості студентів. Вони можуть створювати свої ігри, анімації, інтерактивні історії та інше.

- **Навчання інтернету речей .** Micro:bit може бути використаний для навчання концепцій Інтернету речей. Це дозволяє учням розуміти, як пристрої можуть взаємодіяти один з одним та зовнішнім середовищем.

- **Розвиток логічного мислення.** Використання блокових мов допомагає розвивати логічне мислення та алгоритмічні навички, оскільки учні створюють послідовності команд для досягнення конкретних результатів.

- **Сприяння проблемному вирішенню.** Розв'язання завдань та створення проектів на Micro:bit вимагає вирішення різних проблем та творчого підходу до досягнення мети.

- **Використання в освітніх програмах.** Micro:bit активно використовується в освітніх програмах для впровадження STEM (наука, технологія, інженерія, математика) концепцій та практичного навчання.

- **Залучення до інженерних завдань.** Учні можуть брати участь у різних конкурсах та проектах, використовуючи Micro:bit, що заохочує їх долучатися до інженерних викликів та розвивати свої навички.

Всі ці аспекти роблять Micro:bit прекрасним інструментом для вступу в світ програмування та електроніки, а також для стимулювання інтересу до STEM-освіти.

**PlatformIO** – це відкрите середовище розробки для МК, яке підтримує різні платформи, такі як Arduino, ESP8266, Raspberry Pi та багато інших. Його можна інтегрувати з різними ICP, забезпечуючи зручні умови для розробки програмного забезпечення для вбудованих систем.

PlatformIO – це інтегроване середовище розробки (ICP) та система управління проектами, спеціально спроектовані для роботи з МК та

вбудованими системами. Вона надає декілька значущих переваг для навчання та розробки проектів на основі МК. Ось деякі з корисних аспектів використання PlatformIO в навчанні:

- **Кроссплатформенність.** PlatformIO підтримує Windows, macOS та Linux, що дозволяє користувачам працювати на різних операційних системах без проблем.
- **Підтримка різних мікроконтролерів.** PlatformIO підтримує широкий спектр МК від різних виробників, що дозволяє використовувати його для роботи з різними платформами.
- **Простота установки та використання.** Встановлення та налаштування PlatformIO досить просте, що полегшує його використання навіть для початківців.
- **Інтеграція з різними ІСР.** PlatformIO може бути інтегровано в різні популярні ІСР, такі як Visual Studio Code, Atom, CLion тощо, що надає додаткові можливості для розробки.
- **Автоматизована установка бібліотек.** Легка установка та керування бібліотеками дозволяє легко використовувати готові рішення та розширення для роботи з різними пристроями та датчиками.
- **Управління залежностями проекту.** PlatformIO автоматично вирішує питання залежностей проекту, забезпечуючи зручну роботу зі сторонніми бібліотеками та іншими компонентами.
- **Вбудовані інструменти для роботи з проектами.** PlatformIO включає в себе вбудовані інструменти для роботи з кодом, відлагодження та перевірки налагодження.
- **Система збірки та завантаження.** Однією з ключових переваг є можливість автоматизованої збірки та завантаження програм в МК.
- **Спільнота та документація.** PlatformIO має активну спільноту користувачів та детальну документацію, що полегшує навчання та розвиток в Інтернеті.

Загалом, PlatformIO є потужним інструментом для навчання програмування МК, оскільки він надає зручний та ефективний спосіб розробки проектів на основі вбудованих систем.

Також можна виділити і ще кілька цікавих програм для навчання:

### **NI myRIO**

Вироблений компанією National Instruments, myRIO – це вбудована система з FPGA, ARM-процесором і великим набором введення/виведення. Вона використовується для вивчення та дослідження в галузі програмованої електроніки та системного проектування.

### **TI LaunchPad**

LaunchPad від Texas Instruments – це серія навчальних платформ, які використовують МК від самої TI. Вони призначені для розробки, вивчення та впровадження в реальні проекти.

Навчальні платформи з програмованої електроніки, такі як Arduino, Raspberry Pi, Micro:bit, PlatformIO, NI myRIO і TI LaunchPad, стали важливими інструментами для інженерів, студентів та ентузіастів. Кожна з цих платформ має свої особливості та переваги, відповідаючи різним потребам користувачів.

Загалом, ці навчальні платформи розширюють можливості користувачів у галузі програмованої електроніки, надаючи засоби для творчості, експериментів та вивчення нових технологій. Вибір конкретної платформи залежить від індивідуальних потреб та цілей кожного користувача[13].

## **3.2. РОЗГЛЯД ПОПУЛЯРНИХ ОНЛАЙН-РЕСУРСІВ, ВІДЕО-УРОКІВ, ПІДРУЧНИКІВ ТА ФОРУМІВ, ПРИСВЯЧЕНИХ ПРОГРАМОВАНІЙ ЕЛЕКТРОНІЦІ.**

### **Онлайн ресурси**

Існує безліч онлайн-ресурсів, що надають велику кількість матеріалів з програмованої електроніки. Деякі з найпопулярніших включають:

- **Adafruit Learning System:** Adafruit пропонує широкий вибір проектів та уроків, охоплюючи все від Arduino до Raspberry Pi та інших платформ.

- **SparkFun Tutorials:** SparkFun, відомий своїми електронними компонентами, має велику базу даних уроків і проектів для ентузіастів та інженерів.

- **Electronics Hub:** Цей ресурс пропонує навчальні матеріали та ідеї для робіт з електронікою та МК.

### **Відео-уроки:**

Відео-уроки часто є ефективним засобом вивчення, особливо для візуально орієнтованих користувачів. Популярні канали з програмованої електроніки на YouTube включають:

- **Arduino:** Офіційний канал Arduino надає відео-уроки для різних рівнів користувачів, починаючи від початківців до більш складних проектів[15].

- **The Raspberry Pi Guy:** Цей канал присвячений проектам та урокам з використання Raspberry Pi[15].

- **Jeremy Blum:** Інженер та автор, він надає детальні відео-уроки з програмованої електроніки та робототехніки.

### **Підручники:**

Підручники є важливими ресурсами для більш глибокого розуміння концепцій та практик програмованої електроніки. Кілька рекомендованих підручників:

- **"Getting Started with Arduino" by Massimo Banzi:** Одна з найпопулярніших книг для початківців в програмуванні Arduino.

- **"Raspberry Pi Cookbook" by Simon Monk:** Книга, яка пропонує практичні проекти для використання Raspberry Pi.

- **"Make: Electronics" by Charles Platt:** Цей підручник надає введення в основи електроніки для початківців.

### **Форуми:**

Форуми дозволяють обговорювати проблеми, ділитися ідеями та отримувати підтримку від спільноти. Деякі з популярних форумів:

- **Arduino Forum:** Офіційний форум Arduino, де можна задавати питання та обговорювати проекти.

- **Raspberry Pi Forums:** Форум для обговорення всіх аспектів використання Raspberry Pi.

-



## РОЗДІЛ 4: РОЗРОБКА НАВЧАЛЬНИХ МАТЕРІАЛІВ

### 4.1 ВИЗНАЧЕННЯ ВИМОГ ДО ІНФОРМАЦІЙНО-НАВЧАЛЬНИХ МАТЕРІАЛІВ.

Визначення вимог до ІНМ для програмованої електроніки включає кілька ключових аспектів, які спрямовані на забезпечення ефективного та зрозумілого навчання. Розглянемо кожен аспект більш детально:

**Специфікації мети навчання:** Перед тим як розпочати розробку навчальних матеріалів, важливо чітко визначити мету навчання. Наприклад, чи це вивчення основ програмування для вбудованих систем чи розробка ЕП? Це визначить зміст та глибину навчального матеріалу.

**Аналіз аудиторії:** Детальний аналіз цільової аудиторії є важливим етапом. Враховуючи рівень підготовки, попередні знання та особливості вивчення, ви зможете адаптувати матеріали для кращого засвоєння. Наприклад, якщо ваша аудиторія - початківці, то матеріали повинні бути пояснені простими та доступними термінами.

**Формати та методи навчання:** Враховуйте різні формати та методи навчання для забезпечення максимальної доступності та ефективності. Використання текстових матеріалів, відео-уроків, інтерактивних завдань та практичних лабораторій дозволить студентам вибирати той формат, який їм більше підходить.

**Рівень складності та прогресія:** Забезпечте плавну прогресію у складності матеріалів, розпочинаючи з основних концепцій та поступово переходячи до більш складних тем. Це дозволить студентам систематично засвоювати новий матеріал та розвивати свої навички.

**Доступність та мобільність:** Зробіть матеріали доступними онлайн та мобільними, щоб студенти мали можливість вивчати з будь-якого місця та пристрою. Це збільшить гнучкість та зручність навчання.

**Перевірка знань та зворотний зв'язок:** Введіть методи оцінки знань та систему зворотного зв'язку, щоб студенти могли оцінювати свій прогрес та

виправляти недоліки. Це стимулює активну участь та покращує розуміння матеріалу.

**Забезпечення актуальності:** Підтримуйте актуальність матеріалів, регулярно оновлюючи їх у відповідності до сучасних технологічних тенденцій та ринкових вимог. Це дозволить студентам вивчати сучасні методи та техніки програмованої електроніки.

Враховуючи ці аспекти, можна створити навчальні матеріали, які будуть максимально адаптованими до потреб цільової аудиторії.

## **4.2. ВИБІР МЕТИ ТА АУДИТОРІЇ ДЛЯ НАВЧАЛЬНИХ МАТЕРІАЛІВ**

Мета розділу полягає в створенні навчальних матеріалів для розробки інформаційних стендів для ознайомлення учнів і студентів з приладами програмованої електроніки. Основною метою є ознайомлення з інформаційними стендами, які допоможуть їм взаємодіяти з програмованою електронікою, зокрема з платами Arduino UNO та Raspberry Pi[6].

Аудиторія може включати учнів та студентів технічних спеціальностей, електроніки, комп'ютерних наук та інших суміжних галузей, а також вчителів, лаборантів та інших фахівців, які хочуть ознайомитись із програмованою електронікою.

## **4.3 РОЗРОБКА ІНФОРМАЦІЙНО НАВЧАЛЬНИХ СТЕНДІВ**

У світі ростучого інтересу до програмованої електроніки, створення ефективних навчальних матеріалів для ознайомлення з програмованою електронікою та платами Arduino та Raspberry Pi вимагає відданості та професійності. В даному розділі детально розглянуто процес розробки стендів, які будуть спрямовані на вивчення основ програмованої електроніки та роботи з Arduino та Raspberry Pi.

### **Пошук та ознайомлення із інформації для розробки стендів**

- **відео-уроки:** Розглянуто різноманітні відео-уроки, які надали візуальні приклади, висвітили ключові аспекти програмованої електроніки.
- **Підручники:** Вивчення різноманітних підручників дозволило систематизувати знання та отримати докладний огляд можливостей.
- **Форуми:** Участь у форумах дозволила отримати інсайти та поради від спільноти ентузіастів, що працюють в сфері програмованої електроніки з платами Arduino та Raspberry Pi.

Інформаційно-навчальні стенди по платах Arduino Uno та Raspberry Pi є надзвичайно корисними в освітньому процесі. Вони забезпечують візуалізацію та доступність інформації про живлення, розпінування та поради щодо використання плат, що сприяє глибшому розумінню та ефективнішому навчанню. Стенди допомагають структурувати інформацію, економлять час, підтримують самостійне навчання і забезпечують безпеку при роботі з електронними компонентами. Вони є незамінним ресурсом для студентів, викладачів та ентузіастів, сприяючи швидшому засвоєнню матеріалу та розвитку практичних навичок.

Такий підхід дозволив отримати широкий спектр інформації та поглибити розуміння плат та програмованої електроніки, що стало основою для подальшого впровадження цих знань у розробку ІНМ.

### **Вибір програми для розробки інформаційних стендів - Adobe Photoshop**

Вибір оптимальної програми для створення електронних стендів є важливим етапом в процесі досягненні мети роботи. В умовах максимальної ефективності та гнучкості обрано Adobe Photoshop – потужний графічний редактор, який дозволить нам втілити наші ідеї в життя[7].

Однією з ключових задач є ефективна інтеграція інформації про програмовану електроніку та плат у графічний дизайн. Зважаючи на це, було вивчено методи впровадження текстової інформації, зображень та можливо відео-матеріалів для максимальної зрозумілості та зацікавленості користувачів.

Під час розробки електронних стендів, відбулася активна взаємодія із створеними прототипами.

Такий розгорнутий підхід до вибору та освоєння Adobe Photoshop гарантує не лише високий рівень графічного дизайну, а й забезпечує оптимальну адаптацію програми для конкретних навчальних потреб.

## ВИСНОВОКИ

Під час проведення дослідження та розгляду ІНМ з програмованої електроніки було виявлено ряд значущих аспектів, які визначають неабияку вагомість цієї галузі в контексті сучасного технологічного прогресу. Визначено, що програмована електроніка має глибокі корені в історії та продовжує активно розвиватися, стаючи неодмінною складовою сучасної технологічної індустрії, що визначає та перетворює спосіб функціонування сучасного світу.

В ході огляду основних концепцій та технологій, таких як МК, мови програмування, середовища розробки, електронні компоненти та схемотехніка, відбулося поглиблене вивчення основ програмованої електроніки та її практичного використання. Отримані знання стають невід'ємними для тих, хто цікавиться розробкою вбудованих систем та електроніки загалом, що сприяє формуванню компетентностей у даному сегменті знань. Розгляд мов програмування для вбудованих систем підкреслив важливість вибору правильного інструменту для реалізації конкретних завдань. Середовища розробки виявилися ключовими для зручного та ефективного програмування в програмованій електроніці. Вивчення основних електронних компонентів та їх функцій в контексті програмованої електроніки розкрило необхідність глибокого розуміння цих складових для успішного створення функціональних пристроїв.

Аналіз існуючих навчальних платформ та інформаційно-навчальних ресурсів підкреслив різноманіття доступних можливостей для навчання програмованій електроніці. Ключовим етапом стало визначення вимог до ІНМ, а також обстеження популярних ресурсів та інструментів для ефективного засвоєння матеріалу. Зазначено, що в сучасному цифровому середовищі інформаційно-навчальні ресурси відіграють критичну роль у підготовці фахівців. Вивчення цих даних підкреслює важливість адаптації освітніх програм до стрімкого темпу змін у галузі програмованої електроніки та підкріплює необхідність постійного вдосконалення та розвитку інформаційно-навчальних ресурсів.

Розробка навчальних матеріалів виявилася складним процесом, вимагаючи чіткого визначення цілей та аудиторії, а також детального створення навчального плану та структури матеріалів. Важливим аспектом цього процесу стало обґрунтування вибору методів викладання та практичних завдань, спрямованих на максимальне засвоєння учнями навичок програмування електроніки.

Ураховуючи вищезазначене, можна зазначити, що інформаційно-навчальні матеріали з програмованої електроніки відіграють беззаперечну та ключову роль у процесі підготовки кваліфікованих фахівців, які виявляють високу ефективність у швидко змінюваному та високотехнологічному сегменті індустрії.

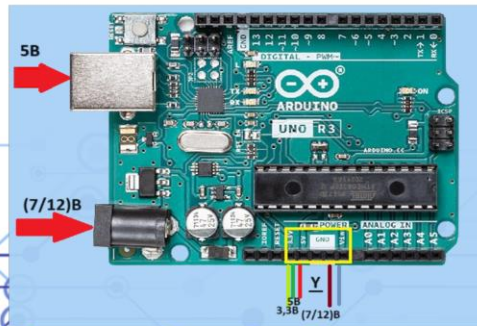
## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Смарт-технології в житті людини / С. О. Котов, Ю. В. Савченко, О. І. Шеремет та ін. - Київ: Видавництво "Нова Книга", 2017. - 256 с.
2. Sterling B. Shaping Things / Bruce Sterling. - Cambridge, MA: The MIT Press, 2005. - 144 p.
3. Greenberg A. Smart Home Hacks: Tips & Tools for Automating Your House / Andrew Greenberg. - Sebastopol, CA: O'Reilly Media, 2004. - 304 p.
4. Інтернет речей: від розумного дому до індустрії 4.0 / Л. І. Гришук, Р. В. Гришук, О. В. Коробов та ін. - Київ: Видавничий дім "Слово", 2018. - 384 с.
5. Жеребило В.Ф. Перспективи використання робототехніки в сучасному світі / В.Ф. Жеребило, Т.Г. Ляшук // Інноваційні практики наукової освіти: матеріали доп. III Всеукр. наук.-практ. конф., 6-9 грудня 2023 р., – Київ. – с. 196.
6. Жеребило В.Ф. Особливості використання інформаційно-комунікаційних технологій в освітньому процесі / В.Ф. Жеребило, Т.Г. Ляшук // Інформаційні технології в професійній діяльності: матеріали доп. XVI Всеукр. наук.-практ. конф., 1 листопада 2023 р., – Рівне. – с. 25.
7. Conrad Chavez Adobe Photoshop Classroom in a Book 1st Edition. Adobe Press, 2022. - 416 p.
8. Simon Monk 30 Arduino Projects for the Evil Genius, 2nd Edition McGraw Hill TAB, 2013. – 224p.
9. Simon Monk Programming Arduino: Getting Started with Sketches, 2nd Edition McGraw Hill TAB, 2016. - 192 p.
10. Болюх В. Ф., Данько В. Г. Основи електроніки і мікропроцесорної техніки: Навч. посібник / В.Ф. Болюх, В.Г. Данько. - Харків: НТУ"ХТ", 2011р. - 257 с.
11. Рябенкий В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки. 2016р. -397с
12. The Art of Electronics 3rd Edition by Paul Horowitz and Winfield Hill. 2015. - 1223p.
13. "Programming Arduino: Getting Started with Sketches" by Simon Monk. 2011. -175p.
14. Python Cookbook: Recipes for Mastering Python 3 / David Beazley, Brian K. Jones. - Sebastopol, CA: O'Reilly Media, 2013. - 706 p.
15. Бізін В. Raspberry Pi для початківців: Навчальний посібник з програмування та проектування / Володимир Бізін. – Київ: Самміт-книга, 2019. – 288 с.

## ДОДАТКИ

### 1. Живлення плати Arduino UNO:

#### Живлення схеми від плати Arduino UNO:



Піни та роз'єми живлення плати Arduino UNO

| Пін/роз'єм    | Напряв     | Опис   |
|---------------|------------|--|
| DC Power Jack | вхід       | подача на плату напруги (7 ÷ 12) В                 |
| USB Type B    | вхід       | подача на плату напруги 5 В                        |
| 5V            | вхід/вихід | подача/отримання на/з плат(у/и) напруги 5 В        |
| 3.3V          | вихід      | отримання з плати напруги 3,3 В                    |
| Vin           | вхід/вихід | подача/отримання на/з плат(у/и) напруги (7 ÷ 12) В |
| GND           | вихід      | земля  |
| AREF          | вхід       | подача на плату опорної напруги для АЦП            |
| IOREF         | вихід      | напруга входу/виходу (HIGHТ) плати                 |

#### Спосіб № 1) Пін 5V (5 В; 500 мА):

Електричні компоненти можуть бути заживлені від піна 5V. В такому разі, на піні буде присутня напруга 5 В, яка надходить від стабілізатора напруги, незалежно від способу живлення Arduino (DC Power Jack, USB або Vin).

Обмеження по струмові:

• живлення Arduino від DC Power Jack або Vin. Максимально допустимий струм, який можна отримати з даного піна складає 800 мА, проте рекомендується використання 500 мА через проблеми з розсіюваною потужністю. Тут можна зробити деяку ремарку. По-правильному, щоб розрахувати допустимий струм через стабілізатор, необхідно враховувати напругу на його вході. Так, в багатьох платах Arduino UNO r3 використовується стабілізатор NCP1117ST50T3G у корпусі SOT-223. Згідно його технічних характеристик, максимальна робоча температура стабілізатора становить 150°C, а тому необхідно слідкувати, щоб вона була менша хоча б на 20 % (120°C). В даташиті також сказано, що регулятор досягає температури 67°C для кожного розсіюваного вата. Це означає, що максимальна розсіювана потужність регулятора становить близько 2 Вт. Зважаючи на такі міркування, можна розрахувати допустиму силу струму через регулятор, при різних напругах на його вході ( $I=W/(U_{in}-U_{out})$ ).

Наприклад:

- о джерело живлення 12 В:  $I = 2 / (12-5) = 2 / 7 = 285$  (мА);
- о джерело живлення 9 В:  $I = 2 / (9-5) = 2 / 4 = 500$  (мА);
- о джерело живлення 7 В:  $I = 2 / (7-5) = 2 / 2 = 1$  (А).

Дані розрахунки можна брати за основу, хоча вони є дещо неточними, оскільки не враховують падіння напруги на захисному діоді.

• живлення Arduino від USB-кабеля. Максимальний струм контакту 5V обмежується 500 мА.

#### Спосіб № 2) Пін 3.3V (3,3 В; 50 мА):

Живити датчики та модулі можна від виводу 3.3V. Даний пін видає напругу 3,3 В, за рахунок ще одного регулятора напруги Arduino (який підключається до виходу регулятора 5V).

Максимальний струм, який можна отримати з піна 3.3V становить 150 мА, хоча бажаним являється 50 мА. Використання піна 3.3V в якості живлення також обмежує максимальний струм на виході піна 5V, оскільки стабілізатор 3,3 В під'єднується до виходу стабілізатора 5 В, що при конструюванні схем необхідно обов'язково враховувати.

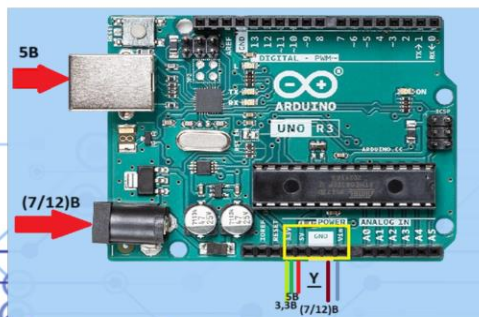
#### Спосіб № 3) Пін Vin ((7 ÷ 12) В; 1 А):

Подача живлення можлива і при використанні піна Vin. Даний вивід напруги з'єднаний із входом зовнішнього живлення Arduino через роз'єм DC Power Jack, минаючи внутрішній стабілізатор. Внаслідок цього, напруга на такому піні буде такою ж як і на гнізді DC Power Jack ((7 ÷ 12) В), що досить таки зручно при живленні модулів з відповідною робочою напругою (напр. заживлювати 12 В реле). При цьому, Vin не буде задіяний, якщо плата Arduino живиться від USB або піна 5V.

Слід зауважити, що даний вивід міг би витримати такий же струм як і джерело живлення Arduino, проте, зважаючи на наявність вбудованого захисного діода, обмеження по струму такого піна складає 1 А.



## 2. Живлення від плати Arduino UNO:



### Живлення схеми від плати Arduino UNO:

Піни та роз'єми живлення плати Arduino UNO

| Пін/роз'єм    | Напряв     | Опис   |
|---------------|------------|--|
| DC Power Jack | вхід       | подача на плату напруги (7 ÷ 12) В                 |
| USB Type B    | вхід       | подача на плату напруги 5 В                        |
| 5V            | вхід/вихід | подача/отримання на/з плат(у/и) напруги 5 В        |
| 3.3V          | вихід      | отримання з плати напруги 3,3 В                    |
| Vin           | вхід/вихід | подача/отримання на/з плат(у/и) напруги (7 ÷ 12) В |
| GND           | вихід      | земля  |
| AREF          | вхід       | подача на плату опорної напруги для АЦП            |
| IOREF         | вихід      | напруга входу/виходу (HIGH) плати                  |

#### Спосіб № 1) Пін 5V (5 В; 500 мА):

Електричні компоненти можуть бути заживлені від піна 5V. В такому разі, на піні буде присутня напруга 5 В, яка надходить від стабілізатора напруги, незалежно від способу живлення Arduino (DC Power Jack, USB або Vin).

Обмеження по струмові:

- живлення Arduino від DC Power Jack або Vin. Максимально допустимий струм, який можна отримати з даного піна складає 800 мА, проте рекомендується використання 500 мА через проблеми з розсіюваною потужністю. Тут можна зробити деяку ремарку. По-правильному, щоб розрахувати допустимий струм через стабілізатор, необхідно враховувати напругу на його вході. Так, в багатьох платах Arduino UNO r3 використовується стабілізатор NCP1117ST50T3G у корпусі SOT-223. Згідно його технічних характеристик, максимальна робоча температура стабілізатора становить 150°C, а тому необхідно слідкувати, щоб вона була менша хоча б на 20 % (120°C). В даташиті також сказано, що регулятор досягає температури 67°C для кожного розсіюваного вата. Це означає, що максимальна розсіювана потужність регулятора становить близько 2 Вт. Зважаючи на такі міркування, можна розрахувати допустиму силу струму через регулятор, при різних напругах на його вході ( $I=W/(U_{in}-U_{out})$ ). Наприклад:

- о джерело живлення 12 В:  $I = 2 / (12-5) = 2 / 7 = 285$  (мА);
- о джерело живлення 9 В:  $I = 2 / (9-5) = 2 / 4 = 500$  (мА);
- о джерело живлення 7 В:  $I = 2 / (7-5) = 2 / 2 = 1$  (А).

Дані розрахунки можна брати за основу, хоча вони є дещо неточними, оскільки не враховують падіння напруги на захисному діоді.

- живлення Arduino від USB-кабеля. Максимальний струм контакту 5V обмежується 500 мА.

#### Спосіб № 2) Пін 3.3V (3,3 В; 50 мА):

Живити датчики та модулі можна від виводу 3.3V. Даний пін видає напругу 3,3 В, за рахунок ще одного регулятора напруги Arduino (який підключається до виходу регулятора 5V).

Максимальний струм, який можна отримати з піна 3.3V становить 150 мА, хоча бажаним являється 50 мА. Використання піна 3.3V в якості живлення також обмежує максимальний струм на виході піна 5V, оскільки стабілізатор 3,3 В під'єднується до виходу стабілізатора 5 В, що при конструюванні схем необхідно обов'язково враховувати.

#### Спосіб № 3) Пін Vin ((7 ÷ 12) В; 1 А):

Подача живлення можлива і при використанні піна Vin. Даний вивід напряму з'єднаний із входом зовнішнього живлення Arduino через роз'єм DC Power Jack, минаючи внутрішній стабілізатор. Внаслідок цього, напруга на такому піні буде такою ж як і на гнізді DC Power Jack ((7 ÷ 12) В), що досить таки зручно при живленні модулів з відповідною робочою напругою (напр. заживлювати 12 В реле). При цьому, Vin не буде задіяний, якщо плата Arduino живиться від USB або піна 5V.

Слід зауважити, що даний вивід міг би витримати такий же струм як і джерело живлення Arduino, проте, зважаючи на наявність вбудованого захисного діода, обмеження по струму такого піна складає 1 А.



### 3.Стенд по історії програмованої електроніки.

#### ІСТОРІЯ ТА РОЗВИТОК ПРОГРАМОВАНОЇ ЕЛЕКТРОНІКИ

Програмована електроніка - є захопливим шляхом від початкових етапів використання електроніки до сучасних високотехнологічних досягнень. Розглядаючи етапи її розвитку, можна виявити ключові події та технологічні вибори, що визначили її сучасний статус.

**Етап 1: Перший поклик електроніки.** Період від народження електроніки до початку 20-го століття визначався експериментами та пошуками нових можливостей в галузі електрики та магнетизму. Під час цього етапу, електроніка формувалася як наука і розвивалася в рамках фундаментальних досліджень.

**Етап 2: Зародження транзисторної епохи.** Другий етап у розвитку програмованої електроніки, який охоплює період 19(50-60)-х років, визначався революційним відкриттям транзисторів та переходом від лампових технологій до напівпровідникових. Цей період був ключовим у зменшенні розмірів та вдосконаленні характеристик електронних пристроїв.

**Етап 3: Інтеграція на мікрошкалу.** Третій етап у розвитку програмованої електроніки, який визначався в період від середини 20-го століття і до 19(70-80)-х років, характеризувався переходом до інтегральних мікросхем. Це важливе досягнення дозволило об'єднати значну кількість елементів на одному чіпі, що веде до експоненційного зростання обчислювальної потужності та зменшення розмірів пристроїв.

**Етап 4: Визначення в будованих систем та мікроконтролерів.** Четвертий етап у розвитку програмованої електроніки, охоплюючи період 19(70-80)-х років, характеризувався появою вбудованих систем та мікроконтролерів. Цей період визначався зростанням компактності, ефективності та універсальності електронічних пристроїв, завдяки використанню мікроконтролерів.

**Етап 5: Ера програмованої електроніки.** П'ятий етап у розвитку програмованої електроніки, який налічує останні десятиліття, характеризується високим рівнем доступності програмування та широким застосуванням в різних галузях. Цей період визначається поширенням використання мікроконтролерів, розвитком інтернет речей, та зростанням інтеграції програмованих елементів у всі аспекти життя людини.





## 4. Плати Arduino UNO

### ТИПИ ПЛАТ ARDUINO

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

**Arduino Leonardo** - це пристрій на базі мікроконтролера ATmega32U4 . До його складу входить все необхідне для роботи з цим мікроконтролером: 20 цифрових входів/виходів (7 з яких можуть працювати як ШІМ-виходи, 12 - як аналогові входи), кварцовий резонатор на 16 МГц, роз'єм мікро-USB, роз'єм живлення , роз'єм для внутрішньосхемного програмування ICSP (In-Circuit Serial Programming) та кнопка скидання. Для початку роботи з Leonardo досить просто подати живлення від AC/DC-адаптера або батареї або підключити його до комп'ютера за допомогою USB-кабелю.

**Arduino Due** - це пристрій на основі мікропроцесора Atmel SAM3X8E ARM Cortex-M3 . Це перша плата Ардуіно на базі 32-розрядного мікроконтролера ARM. До її складу входять 54 цифрових виводів (з яких 12 можуть працювати як аналогові входи), 4 UART (апаратних приймачів, що здійснюють послідовну передачу даних), генератор тактової частоти на 84 МГц, USB з підтримкою технології OTG, 2 ЦАП (цифро-аналоговий перетворювач), 2 TWI, роз'єм живлення, роз'єм SPI, роз'єм JTAG, кнопка скидання і кнопка очищення пам'яті.

**Arduino Micro** - це пристрій на основі мікроконтролера ATmega32u4 , розроблений спільно з Adafruit . До його складу входить все необхідне для зручної роботи з мікроконтролером: 20 цифрових входів/виходів (з них 7 можуть використовуватися як ШІМ-виходи, 12 - як аналогові входи), кварцовий резонатор на 16 МГц, роз'єм micro-USB, роз'єм ICSP для внутрішньосхемного програмування та кнопка скидання. Для початку роботи з пристроєм досить просто підключити його до комп'ютера за допомогою кабелю USB. Пристрій розроблений таким чином, щоб його можна було зручно розміщувати на макетній платі.

**Arduino TRE**, по суті, об'єднує два пристрої: Arduino на базі мікропроцесора Sitara з ядром Linux і Arduino на базі AVR-мікроконтролера. Вбудоване AVR-ядро дозволяє не тільки зберегти простоту та традиції програмування Arduino , а й забезпечує повну сумісність Arduino TRE з існуючими платами розширення. Завдяки цьому, розробники можуть розширювати функціональність Arduino TRE, що дозволяє використовувати його для вирішення широкого спектру завдань, таких, як управління 3D-принтерами, автоматизація будівель та освітлення, збір та обробка даних від бездротових датчиків та інших операцій у режимі реального часу.





## 5. Arduino UNO поради щодо живлення .

### Arduino UNO: поради при роботі з живленням

- підключення живлення Arduino або його модулів повинно проводитися строго згідно полярності провідників:

- о «+» («5V», «3.3V», «Vin», «Vcc») до «+» («5V», «3.3V», «Vin», «Vcc»);
- о «GND» («-», «COM») до «GND» («-», «COM»);

- не можна перевищувати живлення електронного компонента, будь то Arduino або його модулі. В протилежному випадку, це може призвести до виводу з ладу відповідних електронних компонентів;

- не можна перевищувати логічні рівні. Якщо високий рівень МК складає 5 В, то і датчики, кнопки або інші МК повинні посилати сигнал з такою ж напругою;

- загальні виводи («GND», «-», «COM») електронних компонентів схеми повинні бути з'єднані разом, навіть якщо вони заживлені від різних джерел. Це потрібно для того, щоб «опорна точка відліку» була однаковою для всіх модулів;

- максимальний допустимий струм піна вводу/виводу Arduino складає 40 мА, при бажаних 20 мА. Загальний ж струм на тарні пінах не повинен перевищувати 200 мА. Так, підключення незнайомого навантаження до GPIO-піна передбачає використання резистора ~ 250 Ом, що не дозволить протіканню струму вищому від  $5 / 250 = 20$  (мА). Якщо по такому принципу підключити світлодіоди, то їх кількість не повинна перевищувати  $200 \text{ мА} / 20 \text{ мА} = 10$ ;

- піни живлення («5V», «3.3V», «Vin») та GPIO плати Arduino не можна напряму сполучати із GND, оскільки це призведе до короткого замикання!

- якщо в проекті присутні потужні пристрої (напр. двигуни), їх необхідно живити не від стабілізатора плати Arduino (піни «5V» і «3.3V»), а від зовнішнього джерела або піни «Vin», задля того, щоб убезпечити від виходу з ладу плату. Фактично пристрій, що споживає струм більший від 500 мА повинен бути заживлений від зовнішнього джерела. В свою чергу, серед модулів, які можна живити засобами Arduino виступають наприклад малопотужні датчики, дисплеї, реле, світлодіоди тощо;

- звертайте увагу на параметри джерел живлення:

- о напруга – відповідає напрузі, яка подається на споживача електроенергії;

- о струм – являє собою максимально допустимий струм, який може видати дане джерело. Але це не означає, що саме такий струм буде на його виході – все залежить від потрібного струму для споживача. Який струм споживатиме електрична схема, такий струм і буде на виході джерела живлення. Тому необхідно слідувати, щоб максимальний струм джерела живлення перевищував (хоча б на 20 %) струм, яким живляться радіокомпоненти, підключені до даного джерела. В протилежному випадку, джерело перегріватиметься і може вийти з ладу!

- при підключенні до схеми декількох компонентів, їх струми споживання сумуються.

Так наприклад, підключення до Arduino 3-ох датчиків, кожен з яких споживає струм 5 мА, призведе до того, що загальний струм їх споживання становитиме  $3 \cdot 5 = 15$  (мА). При цьому, слід також враховувати струм, необхідний для самої плати Arduino, який залежить від декількох факторів (напр. наявність тих чи інших електронних компонентів на платі, режим роботи, задіяна внутрішня периферія (GPIO, SPI тощо)). Тому, необхідно залишати запас принаймні в 200 мА, для однієї лише Arduino. В такому разі, мінімальна вимога до зовнішнього джерела – 250 мА, що дозволить запустити лише одну плату Arduino з мінімальною для неї обв'язковою (Рис.1);

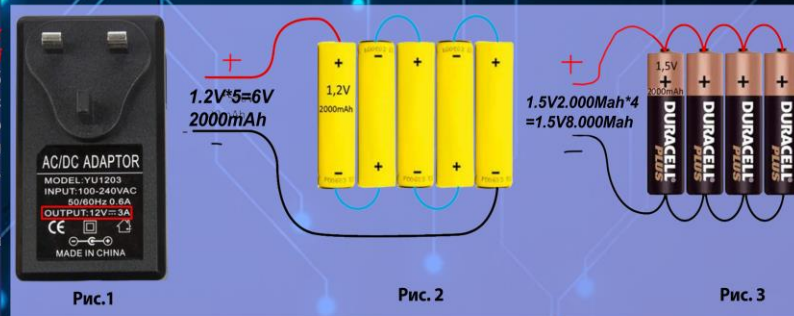
- у якості зовнішнього джерела можна використовувати декілька з'єднаних між собою джерел (батареї, акумулятори тощо). Такі джерела повинні бути абсолютно ідентичними:

- о по-можливості походити з однієї партії або упаковки;

- о використовувати однакову технологію (напр. AGM, GEL, MG);

- о мати однаковий вольтаж та ємність (у випадку лужних батарей – бути новими; у випадку акумуляторів – повністю бути або зарядженими або розрядженими); В протилежному разі, робота такої групи елементів буде непередбачуваною, а також призводить до передчасного виходу з ладу елементів живлення;

- послідовне з'єднання джерел живлення переважає сумування їх напруг. При цьому, з'єднуються різноміснї полярності, так що «+» однієї батареї під'єднується до «-» іншої. Наприклад, якщо з'єднати 5 АКБ по 1,2 В і 2000 мАг, то сумарна їх напруга на виході становитиме  $1,2 \cdot 4 = 6$  (В) при загальній ємності 2000 мАг (Рис. 2);



- паралельне з'єднання джерел живлення переважає сумування їх ємностей та максимально допустимого вихідного струму. При цьому, «+» всіх батарей з'єднуються разом. Це ж саме стосується і «-». Наприклад, якщо з'єднати 4 батареї по 1,5 В, то сумарна їх ємність становитиме  $4 \cdot 2000 = 8000$  (мАг) при загальній напрузі 1,5 В (Рис.3).



## 6. Розпізнавання плати Arduino Uno.

### ПЕРЕФІРІЯ

**GPIO:** 20 пінів: 0- 19. Цифрові порти загального призначення. Логічний рівень одиниці - 5 В, нуля - 0 В. До контактів підключені підтягаючі резистори, які за замовчуванням вимкнені, проте можуть бути включені програмно.

**ADC:** 6 пінів: (14 – 19) (A0 – A5) Аналого-цифровий перетворювач. Діапазон вхідної напруги (0÷5)В. Розрядність ADC не змінюється та встановлена в 10 біт.

**PWM:** 6 пінів: 3, 5, 6 і 9 – 11 Широтно-імпульсна модуляція. Розрядність PWM не змінюється та встановлена в 8 біт.

**I2C:** 2 пини: SDA(18/A4) та SCL(19/A5). Послідовний інтерфейс передачі даних I2C. Для зв'язку контролера з платами розширення та сенсорами.

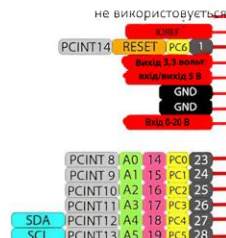
**SPI:** 4 пини: SCK(13), MISO(12), MOSI(11) та SS(10)

Послідовний інтерфейс передачі даних SPI. Для зв'язку контролера з платами розширення та сенсорами.

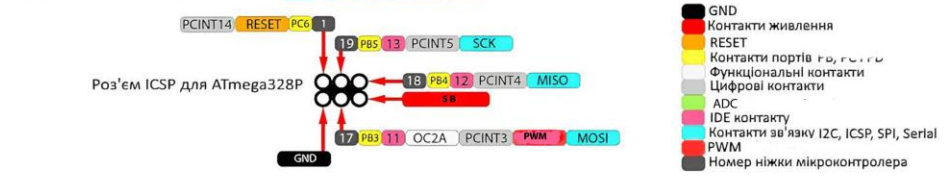
**UART:** 2 пини: TX(1) та RX(0). Інтерфейс передавання даних для зв'язку контролера з платами розширення та сенсорами. Контакти також пов'язані з відповідними виводами ATmega16U2 для зв'язку плати по USB. Під час прошивки та налагодження програми через ПК, не слід використовувати дані пини у проекті.

## Arduino UNO Pinout

Роз'єм для живлення від зовнішнього джерела



Роз'єм ICSP для ATmega16U2



Роз'єм ICSP для ATmega328P



- GND
- Контакти живлення
- RESET
- Контакти портів I2C, SPI, UART
- Функціональні контакти
- Цифрові контакти
- ADC
- IDE контакту
- Контакти зв'язку I2C, ICSP, SPI, Serial
- PWM
- Номер ніжки мікроконтролера

### ПІНИ ЖИВЛЕННЯ

**VIN:** Живлення (7 ÷ 12В). Пін може працювати як на вхід так на вихід.

**5V:** Живлення (5В). Пін може працювати як на вхід так на вихід.

**3V3:** Вихідний пін для отримання напруги з плати (3,3 В).

**IOREF:** Вихідний пін виского логічного рівня плати. Пін дозволяє отримати шилдам (попередньо підключеними до Arduino) інформацію щодо значення напруги, на якій працює МК плати Arduino (у випадку Arduino UNO це 5 В).

**AREF:** Вхідний пін для подачі на ADC плати опорної напруги.

**GND:** земля.

## 7. Raspberry Pi поради щодо живлення.

### ПОРАДИ ЩОДО ВИКОРИСТАННЯ ПЛАТ RASPBERRY PI

#### **Підключення живлення Raspberry Pi повинно відповідати певним вимогам:**

Встановіть блок живлення відповідно до вимог конкретної моделі Raspberry Pi. Наприклад, для Raspberry Pi 4 рекомендується блок живлення 5V 3A. Не використовуйте блоки живлення з низькою напругою або струмом, оскільки це може призвести до нестабільної роботи системи.

#### **Не перевищуйте допустимий рівень живлення:**

Звертайте увагу на максимальну потужність пристроїв, які підключаються до Raspberry Pi через USB порти та GPIO піни. Використовуйте стабілізатори напруги, якщо підключаєте додаткові компоненти, щоб уникнути перенапруги.

#### **Забезпечте правильну полярність:**

При підключенні компонентів через GPIO піни або інші інтерфейси дотримуйтесь правильної полярності (плюс до плюса, мінус до мінуса).

Неправильне підключення може спричинити пошкодження як Raspberry Pi, так і підключених компонентів.

#### **Використання зовнішніх джерел живлення:**

Для проектів, що потребують більшої потужності, розгляньте можливість використання зовнішніх джерел живлення. Наприклад, можна підключити додатковий блок живлення через GPIO піни або використовувати спеціальні плати розширення з власними джерелами живлення.

#### **Уникайте перевантаження системи:**

Не підключайте одночасно занадто багато пристроїв до USB портів Raspberry Pi, оскільки це може перевантажити систему.

Використовуйте активні USB хаби для підключення великої кількості USB пристроїв

#### **Поради щодо живлення від акумуляторів:**

Використовуйте якісні акумулятори та переконайтеся, що їх напруга відповідає вимогам Raspberry Pi. При паралельному з'єднанні акумуляторів сумарна ємність збільшується, але напруга залишається тією ж (наприклад, паралельне з'єднання 4 акумуляторів 1.5V 2000mAh дає 1.5V 8000mAh).

#### **Живлення через Power over Ethernet (PoE):**

Якщо ваша модель Raspberry Pi підтримує PoE, розгляньте можливість використання цього методу для зручного підключення живлення через Ethernet кабелі.

#### **Рекомендації щодо забезпечення безпеки:**

Переконайтеся, що всі підключення надійні та безпечні, щоб уникнути короткого замикання або інших електричних проблем.

Використовуйте якісні кабелі та конектори для підключення компонентів.

#### **Моніторинг напруги та струму:**

Використовуйте інструменти для моніторингу напруги та струму, щоб вчасно виявляти та усувати можливі проблеми з живленням.

Інтегруйте в ваші проекти модулі захисту від перевантаження та перенапруги.

#### **Підключення додаткових компонентів:**

Для підключення додаткових компонентів, таких як дисплеї, датчики, актуатори, використовуйте захищені схеми живлення.

Дотримуйтесь інструкцій виробників компонентів щодо правильного підключення та живлення.



## 8. Типи плат Raspberry Pi 1.

### ТИПИ ПЛАТ RASPBERRY PI

#### Raspberry Pi 4 Model B

**Процесор:** 1.5GHz 64-bit quad-core ARM Cortex-A72

**Оперативна пам'ять:** Від 2 Гб до 8 Гб LPDDR4-3200 SDRAM

**Порти:** 2 x USB 3.0, 2 x USB 2.0, 2 x micro HDMI (підтримка 4К), Gigabit Ethernet, Wi-Fi, Bluetooth 5.0

**Особливості:** Більша продуктивність та можливість підключення до двох моніторів.



#### Raspberry Pi 400

**Процесор:** 1.8GHz quad-core ARM Cortex-A72

**Оперативна пам'ять:** 4 Гб LPDDR4-3200 SDRAM

**Порти:** 2 x micro HDMI, 3 x USB 2.0, 1 x USB 3.0, Gigabit Ethernet

**Особливості:** Інтегрований у корпус клавіатури, що робить його ідеальним для навчання та простого використання.

#### Raspberry Pi 3 Model A+

**Процесор:** 1.4GHz 64-bit quad-core ARM Cortex-A53

**Оперативна пам'ять:** 512 Мб LPDDR2 SDRAM

**Порти:** 1 x USB 2.0, HDMI, 10/100 Ethernet, Wi-Fi, Bluetooth 4.2

**Особливості:** Менша плата з основними функціями моделі B+, що ідеально підходить для проектів з обмеженим простором



#### Raspberry Pi Pico

**Процесор:** Dual-core ARM Cortex-M0+ @ 133MHz

**Оперативна пам'ять:** 264 Кб SRAM, 2 Мб флеш-пам'яті

**Порти:** 26 multi-function GPIO pins, USB 1.1

**Особливості:** Створений для використання в проектах, що потребують невеликого контролера, з низьким енергоспоживанням.

#### Raspberry Pi 3 Model B+

**Процесор:** 1.4GHz 64-bit quad-core ARM Cortex-A53

**Оперативна пам'ять:** 1 Гб LPDDR2 SDRAM

**Порти:** 4 x USB 2.0, HDMI, 10/100 Ethernet, Wi-Fi, Bluetooth 4.2

**Особливості:** Покращена продуктивність та бездротовий зв'язок у порівнянні з попередніми моделями.



## 9. Типи плат Raspberry Pi 2.

### ТИПИ ПЛАТ RASPBERRY PI

#### RASPBERRY PI COMPUTE MODULE 4

**Процесор:** QUAD-CORE CORTEX-A72 (ARM v8) 64-біт SoC @ 1.5GHz  
**Оперативна пам'ять:** 1GB, 2GB, 4GB, АБО 8GB LPDDR4-3200 SDRAM  
**Зберігання:** 8GB, 16GB, АБО 32GB eMMC ФЛЕШ-ПАМ'ЯТІ  
**Мережа:** GIGABIT ETHERNET, Wi-Fi 802.11ac, BLUETOOTH 5.0 (ОПЦІОНАЛЬНО)  
**Особливості:** Модульний формат, ВИКОРИСТАННЯ В ВБУДОВАНИХ СИСТЕМАХ



#### RASPBERRY PI 1 MODEL B+

**Процесор:** Single-core ARM1176JZF-S @ 700MHz  
**Оперативна пам'ять:** 512MB LPDDR2 SDRAM  
**Порти:** 4 x USB 2.0, HDMI  
**Мережа:** 10/100 Ethernet  
**Особливості:** Класична модель, доступна ціна

#### RASPBERRY PI 2 MODEL B

**Процесор:** Quad-core Cortex-A7 (ARM v7) 32-bit SoC @ 900MHz  
**Оперативна пам'ять:** 1GB LPDDR2 SDRAM  
**Порти:** 4 x USB 2.0, HDMI  
**Мережа:** 10/100 Ethernet  
**Особливості:** Поліпшена продуктивність порівняно з першим Raspberry Pi, доступна ціна



#### Raspberry Pi Zero W

**Процесор:** Single-core ARM1176JZF-S @ 1GHz  
**Оперативна пам'ять:** 512MB LPDDR2 SDRAM  
**Порти:** mini-HDMI, 2 x micro-USB (OTG), 40-pin GPIO header  
**Мережа:** Wi-Fi 802.11n, Bluetooth 4.1  
**Особливості:** Компактний розмір, низька ціна, вбудований Wi-Fi та Bluetooth





## 10. Основи розпінування плат Raspberry Pi 1.

### PINOUT RASPBERRY PI

**ПЕРИФЕРІЯ**

**GPIO:** 40 пінів: 1 - 40. Цифрові порти загального призначення. Логічний рівень одиниці - 3.3 В, нуля - 0 В. До контактів підключені підтягаючі резистори, які за замовчуванням вимкнені, проте можуть бути включені програмно.

**W1-GPIO:** 1 пін: (GPIO4). Однопровідний інтерфейс.

**JTAG:** 11 пінів: alt 5(7,29,31,32,33) alt 4 (15,16,18,22,37,13). Послідовний інтерфейс відлагодження плати

**PWM:** 4 пини: 12,32,33 та 35. Широтно-імпульсна модуляція. Розрядність PWM не змінюється та встановлена в 8 біт.

**I2C:** 4 пини: 3,5,27 та 28. Послідовний інтерфейс передачі даних I2C. Для зв'язку контролера з платами розширення та сенсорами.

**SPI:** 11 пінів: SPI0(26,24,21,19 та 23), SPI1(36,11,12,35,38,40). Послідовний інтерфейс передачі даних SPI. Для зв'язку контролера з платами розширення та сенсорами.

**GPCLK:** 3 пини: 7,29 та 3. Тактування загального призначення, дальні пини дозволяють генерувати сигнали різної частоти.

**UART:** 18 пінів 3,5,7,8,10,11,19,21,23,24,26,27,28,29,31,32,33 та 36. Інтерфейс передавання даних для зв'язку контролера з платами розширення та сенсорами. Контакти також пов'язані з відповідними виводами ATmega16U2 для зв'язку плати по USB. Під час прошивки та налагодження програми через ПК, не слід використовувати дані пини у проєкті.

**DPI:** 28 пінів: 3,5,7,8,10,11,12,13,15,16,18,19,21,22,23,24,26,27,28,29,31,32,33,35,36,37,38 та 40. Паралельний інтерфейс для підключення різноманітних RGB-дисплеїв.

**SDIO:** 6 пінів: 13,15,16,18,22 та 37. Інтерфейс SD-карти.

**PCM:** 4 пини: 12,35,38 та 40. Імпульсно-кодова модуляція.

|                     |    |   |   |    |                     |
|---------------------|----|---|---|----|---------------------|
| 3v3 Power           | 1  | ● | ● | 2  | 5v Power            |
| GPIO 2 (I2C1 SDA)   | 3  | ● | ● | 4  | 5v Power            |
| GPIO 3 (I2C1 SCL)   | 5  | ● | ● | 6  | Ground              |
| GPIO 4 (GPCLK0)     | 7  | ● | ● | 8  | GPIO 14 (UART TX)   |
| Ground              | 9  | ● | ● | 10 | GPIO 15 (UART RX)   |
| GPIO 17             | 11 | ● | ● | 12 | GPIO 18 (PCM CLK)   |
| GPIO 27             | 13 | ● | ● | 14 | Ground              |
| GPIO 22             | 15 | ● | ● | 16 | GPIO 23             |
| 3v3 Power           | 17 | ● | ● | 18 | GPIO 24             |
| GPIO 10 (SPI0 MOSI) | 19 | ● | ● | 20 | Ground              |
| GPIO 9 (SPI0 MISO)  | 21 | ● | ● | 22 | GPIO 25             |
| GPIO 11 (SPI0 SCLK) | 23 | ● | ● | 24 | GPIO 8 (SPI0 CE0)   |
| Ground              | 25 | ● | ● | 26 | GPIO 7 (SPI0 CE1)   |
| GPIO 0 (EEPROM SDA) | 27 | ● | ● | 28 | GPIO 1 (EEPROM SCL) |
| GPIO 5              | 29 | ● | ● | 30 | Ground              |
| GPIO 6              | 31 | ● | ● | 32 | GPIO 12 (PWM0)      |
| GPIO 13 (PWM1)      | 33 | ● | ● | 34 | Ground              |
| GPIO 19 (PCM FS)    | 35 | ● | ● | 36 | GPIO 16             |
| GPIO 26             | 37 | ● | ● | 38 | GPIO 20 (PCM DIN)   |
| Ground              | 39 | ● | ● | 40 | GPIO 21 (PCM DOUT)  |

**ПІНИ ЖИВЛЕННЯ**

**5V:** 2 пини: 2 та 4. Живлення (5В). Пін може працювати як на вхід так на вихід.

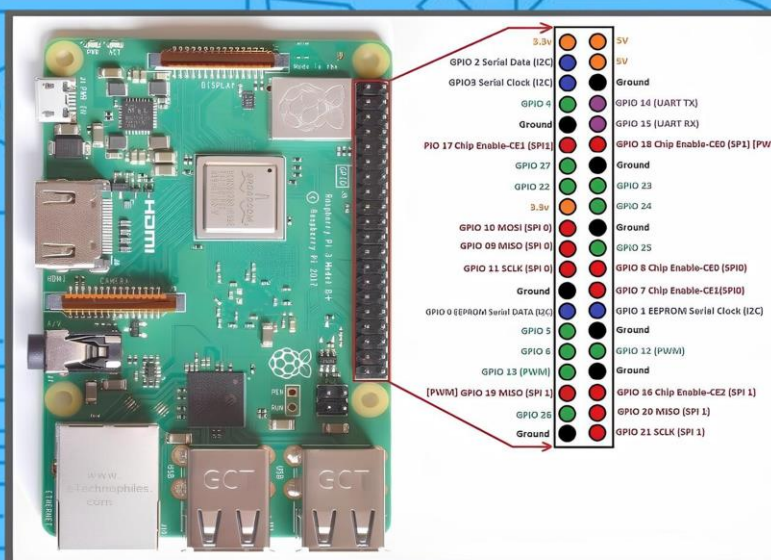
**3V3:** 2 пини: 1 та 17. Вихідний пін для отримання напруги з плати (3,3 В).

**GND:** 8 пінів : 6,9,14,20,25,30,34 та 39. Земля.

## 11. Розпінування плати Raspberry Pi 3B+.

### РОЗПІНУВАННЯ ПЛАТИ RASPBERRY PI 3 B+

Розпінування GPIO (GENERAL PURPOSE INPUT/OUTPUT) для RASPBERRY PI 3 B+ НАДАЄ МОЖЛИВІСТЬ ПІДКЛЮЧЕННЯ РІЗНОМАНІТНИХ ЗОВНІШНІХ ПРИСТРОЇВ, ДАТЧИКІВ ТА МОДУЛІВ. РОЗ'ЄМИ НА ПЛАТІ ПРИЗНАЧЕНІ ДЛЯ РІЗНИХ ЦІЛЕЙ: ЖИВЛЕННЯ, ЗАЗЕМЛЕННЯ, ПЕРЕДАЧІ ДАНИХ ТОЩО. НИЖЧЕ ПРЕДСТАВЛЕНО ДЕТАЛЬНЕ РОЗПІНУВАННЯ РОЗ'ЄМУ GPIO.



### ОСНОВНІ КОНТАКТИ ЖИВЛЕННЯ ТА ЗАЗЕМЛЕННЯ:

**3.3V** - КОНТАКТИ ДЛЯ ПОДАЧІ НАПРУГИ 3.3 В.

**5V** - КОНТАКТИ ДЛЯ ПОДАЧІ НАПРУГИ 5 В.

**GROUND** - КОНТАКТИ ЗАЗЕМЛЕННЯ.

Розпінування GPIO складається з 40 контактів, розташованих у два ряди по 20 контактів. Кожен контакт має свою функцію, що дозволяє підключати до RASPBERRY PI різноманітні пристрої для виконання різних завдань.

### КОНТАКТИ ВВОДУ-ВИВОДУ:

**GPIO 2** (SDA1, I2C) - Послідовні дані.

**GPIO 3** (SCL1, I2C) - Послідовний таймер.

**GPIO 4** - Загальне призначення.

**GPIO 17** - Загальне призначення.

**GPIO 27** - Загальне призначення.

**GPIO 22** - Загальне призначення.

**GPIO 10** (MOSI, SPI0) - Головний вихід, послідовний вхід.

**GPIO 9** (MISO, SPI0) - Головний вхід, послідовний вихід.

**GPIO 11** (SCLK, SPI0) - Послідовний таймер.

**GPIO 5** - Загальне призначення.

**GPIO 6** - Загальне призначення.

**GPIO 13** (PWM1) - Широтно-імпульсна модуляція.

**GPIO 19** (MISO, SPI1, PWM1) - Головний вхід, серійний вихід.

**GPIO 26** - Загальне призначення.

**GPIO 14** (TXD0) - Передавач UART.

**GPIO 15** (RXD0) - Приймач UART.

**GPIO 18** (SPI1, PWM1) - Вибір чіпа.

**GPIO 23** - Загальне призначення.

**GPIO 24** - Загальне призначення.

**GPIO 25** - Загальне призначення.

**GPIO 8** (SPI0, CE0) - Вибір чіпа.

**GPIO 7** (SPI0, CE1) - Вибір чіпа.

**GPIO 1** (EEPROM, I2C) - Послідовний таймер.

**GPIO 12** (PWM0) - Широтно-імпульсна модуляція.

**GPIO 16** (SPI1, CE2) - Вибір чіпа.



## 12. Живлення зовнішніх пристроїв від плат Raspberry Pi 1.

### Живлення зовнішніх пристроїв від плат Raspberry Pi

#### Порти живлення Raspberry Pi

##### USB-порти:

Raspberry Pi має кілька USB-портів (2.0 та 3.0 на новіших моделях), які можна використовувати для живлення зовнішніх пристроїв, таких як клавіатури, миші, флеш-накопичувачі та інші периферійні пристрої. Максимальний струм, який може подавати один USB-порт, зазвичай обмежений 500 мА.

##### GPIO-піни:

Raspberry Pi має 40-контактний GPIO (General Purpose Input/Output) роз'єм, який може використовуватися для живлення невеликих пристроїв.

##### На GPIO доступні пін живлення 5V і 3.3V:

Піни 2 і 4: 5V.

Пін 1: 3.3V.

GPIO також включає пін заземлення (GND), необхідні для завершення електричного кола.

##### Живлення через GPIO:

GPIO-піни надають можливість живлення зовнішніх пристроїв, таких як сенсори, модулі та невеликі електронні компоненти.

##### Обмеження по струму:

5V пін може надавати струм до 2.5A в залежності від моделі Raspberry Pi і поточного споживання інших компонентів.

3.3V пін може забезпечувати до 50 мА.

##### Приклад підключення світлодіода

Підключення світлодіода до GPIO:

Підключіть анод світлодіода до піна 1 (3.3V).

Підключіть катод світлодіода через резистор (220 Ом) до піна заземлення

##### Запобіжні заходи

**Перевірка споживання струму:** Перед підключенням перевірте споживання струму всіх зовнішніх пристроїв, щоб уникнути перевантаження.

**Використання стабілізаторів:** Для чутливих пристроїв можна використовувати стабілізатори напруги для запобігання пошкодженню від коливань напруги.

#### Використання HATs і модулів розширення

**HAT** (Hardware Attached on Top) — це модулі розширення, які встановлюються на GPIO та часто мають власні джерела живлення або використовують живлення від Raspberry Pi.

##### Приклади HATs

*Motor HAT:* Для керування двигунами постійного струму.

*Sensor HAT:* Для підключення різноманітних сенсорів (температури, вологості, руху тощо).

*Power Management HAT:* Для забезпечення стабільного живлення Raspberry Pi та підключених пристроїв.



#### Зовнішні джерела живлення

Якщо підключені пристрої вимагають більше потужності, ніж може забезпечити Raspberry Pi, можна використовувати зовнішні джерела живлення.

##### Живлення через USB-хаб з власним живленням:

Використання USB-хабу з власним живленням дозволяє підключати більше потужних пристроїв без перевантаження Raspberry Pi.

##### Адаптери живлення:

Можна використовувати окремі адаптери живлення для більш потужних компонентів, таких як дисплеї або мотори.



## 13.Програмування електроніка.

### Програмована електроніка

**Програмована електроніка** — це поєднання електронних компонентів і мікроконтролерів або мікропроцесорів, які можуть бути запрограмовані для виконання певних завдань. Вона дозволяє створювати складні електронні пристрої та системи, які можуть реагувати на зовнішні сигнали, обробляти інформацію та взаємодіяти з іншими пристроями.

#### Основні компоненти програмованої електроніки

##### 1. Мікроконтролери та мікропроцесори

**Мікроконтролери** (наприклад, Arduino): Одночипові комп'ютери, які містять процесор, пам'ять та периферійні інтерфейси на одному кристалі. Призначені для виконання простих і середніх завдань у режимі реального часу.

**Мікропроцесори** (наприклад, Raspberry Pi): Повноцінні процесори, які потребують зовнішньої пам'яті та периферійних пристроїв. Використовуються для виконання складних обчислювальних завдань і підтримки операційних систем.

##### 2. Сенсори

Вимірюють фізичні параметри (температура, вологість, світло, рух і т.д.) і перетворюють їх в електричні сигнали, які можна обробити за допомогою мікроконтролера або мікропроцесора.

##### 3. Актуатори

Пристрої, що виконують дії у відповідь на команди від мікроконтролера (наприклад, двигуни, реле, світлодіоди).

#### Популярні платформи для програмованої електроніки

##### Arduino

**Призначення:** Простота у використанні для початківців і швидка розробка прототипів.

**Мова програмування:** C/C++ (Arduino IDE).

**Переваги:** Легка інтеграція з різними сенсорами та актуаторами, велика спільнота користувачів.

**Приклади застосування:** Робототехніка, домашня автоматизація, інтернет речей (IoT).

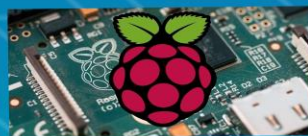
##### Raspberry Pi

**Призначення:** Більша обчислювальна потужність та функціональність для складних проектів.

**Мова програмування:** Python, C/C++, Java та інші.

**Переваги:** Підтримка операційної системи Linux, можливість підключення до мережі та роботи з мультимедіа.

**Приклади застосування:** Медіацентри, сервери, навчальні комп'ютери, проекти IoT.



#### Застосування програмованої електроніки

**Інтернет речей (IoT):** Підключення різних пристроїв до інтернету для збору даних та віддаленого керування.

**Робототехніка:** Створення автономних роботів для промислових, дослідницьких та побутових застосувань.

**Домашня автоматизація:** Інтелектуальне керування освітленням, опаленням, безпекою та іншими системами будинку.

**Медицина:** Розробка медичних пристроїв для моніторингу стану пацієнтів, діагностики та лікування.

**Освіта:** Використання мікроконтролерів і мікропроцесорів для навчання основ програмування та електроніки.



## 14. Порівняння плат Raspberry Pi та Arduino UNO.

### Порівняння Arduino UNO і Raspberry Pi

#### ARDUINO UNO

**Процесор:** ATmega328P (8-бітний мікроконтролер)

**Тактова частота:** 16 МГц

**Оперативна пам'ять (RAM):** 2 КБ

**Постійна пам'ять (Flash):** 32 КБ

**Цифрові входи/виходи:** 14 (з яких 6 можуть використовуватись як PWM виходи)

**Аналогові входи:** 6

**Живлення:** 7÷12 В постійного струму

**Порти комунікації:** UART, SPI, I2C

**Програмування:** Arduino IDE (мова програмування C/C++)

**Операційна система:** Відсутня

**Призначення:** Робота з сенсорами, керування пристроями, прості автоматизації



#### RASPBERRY PI

**Процесор:** BCM2835 (32-бітний ARM процесор), BCM2711 (64-бітний ARM процесор на новіших моделях)

**Тактова частота:** 700 МГц - 1.5 ГГц залежно від моделі

**Оперативна пам'ять (RAM):** 512 МБ - 8 ГБ залежно від моделі

**Постійна пам'ять:** MicroSD карта

**Цифрові входи/виходи:** 26-40 GPIO пінів залежно від моделі

**Живлення:** 5 В через microUSB або USB-C залежно від моделі

**Порти комунікації:** UART, SPI, I2C, HDMI, USB, Ethernet (залежно від моделі)

**Програмування:** Python, C/C++, Scratch, інші

**Операційна система:** Linux (Raspbian, Ubuntu, інші)

**Призначення:** Високопродуктивні обчислення, мультимедійні проекти, мережеві додатки, повноцінні ПК-проекти

#### Основні відмінності

**Продуктивність:** Raspberry Pi має набагато потужніший процесор та більшу кількість оперативної пам'яті, що дозволяє використовувати його для більш складних завдань.

**Операційна система:** Raspberry Pi працює на Linux, що надає доступ до широкого спектру програмного забезпечення, тоді як Arduino UNO не має операційної системи.

**Програмування:** Arduino UNO зазвичай програмується через Arduino IDE з використанням C/C++, тоді як Raspberry Pi підтримує багато мов програмування, включаючи Python.

**Інтерфейси комунікації:** Raspberry Pi має більше комунікаційних можливостей, включаючи HDMI, USB та Ethernet порти, що робить його більш універсальним для підключення до інших пристроїв.

**Живлення:** Arduino UNO живиться від 7-12 В, тоді як Raspberry Pi живиться від 5 В.



## 15. Середовища розробки.

### Порівняння Arduino UNO і Raspberry Pi

#### ARDUINO UNO

**Процесор:** ATmega328P (8-бітний мікроконтролер)  
**Тактова частота:** 16 МГц  
**Оперативна пам'ять (RAM):** 2 КБ  
**Постійна пам'ять (Flash):** 32 КБ  
**Цифрові входи/виходи:** 14 (з яких 6 можуть використовуватись як PWM виходи)  
**Аналогові входи:** 6  
**Живлення:** 7÷12 В постійного струму  
**Порти комунікації:** UART, SPI, I2C  
**Програмування:** Arduino IDE (мова програмування C/C++)  
**Операційна система:** Відсутня  
**Призначення:** Робота з сенсорами, керування пристроями, прості автоматизації



#### RASPBERRY PI

**Процесор:** BCM2835 (32-бітний ARM процесор), BCM2711 (64-бітний ARM процесор на новіших моделях)  
**Тактова частота:** 700 МГц - 1.5 ГГц залежно від моделі  
**Оперативна пам'ять (RAM):** 512 МБ - 8 ГБ залежно від моделі  
**Постійна пам'ять:** MicroSD карта  
**Цифрові входи/виходи:** 26-40 GPIO пінів залежно від моделі  
**Живлення:** 5 В через microUSB або USB-C залежно від моделі  
**Порти комунікації:** UART, SPI, I2C, HDMI, USB, Ethernet (залежно від моделі)  
**Програмування:** Python, C/C++, Scratch, інші  
**Операційна система:** Linux (Raspbian, Ubuntu, інші)  
**Призначення:** Високопродуктивні обчислення, мультимедійні проекти, мережеві додатки, повноцінні ПК-проекти

#### Основні відмінності

**Продуктивність:** Raspberry Pi має набагато потужніший процесор та більшу кількість оперативної пам'яті, що дозволяє використовувати його для більш складних завдань.  
**Операційна система:** Raspberry Pi працює на Linux, що надає доступ до широкого спектру програмного забезпечення, тоді як Arduino UNO не має операційної системи.  
**Програмування:** Arduino UNO зазвичай програмується через Arduino IDE з використанням C/C++, тоді як Raspberry Pi підтримує багато мов програмування, включаючи Python.  
**Інтерфейси комунікації:** Raspberry Pi має більше комунікаційних можливостей, включаючи HDMI, USB та Ethernet порти, що робить його більш універсальним для підключення до інших пристроїв.  
**Живлення:** Arduino UNO живиться від 7-12 В, тоді як Raspberry Pi живиться від 5 В.

