

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Електронної та біомедичної інженерії

Кафедра \_\_ Мікроелектроніки, електронних приладів та пристроїв

## **КВАЛІФІКАЦІЙНА РОБОТА** **Пояснювальна записка**

рівень вищої освіти \_\_\_\_\_ другий (магістерський)

Дослідження властивостей різних типів мікроконтролерів та використання їх  
в електронних пристроях  
(тема)

Виконав:  
студент 2-го курсу, групи \_\_\_\_\_ ЕЕПм-22-1  
\_\_\_\_\_ Тимченко Ю.В.

Спеціальність 171 «Електроніка»  
(код і повна назва спеціальності)

Тип програми \_\_\_\_\_ освітньо-професійна  
(освітньо-професійна або освітньо -наукова)

Освітня програма «Електроні прилади та  
пристрої»  
(повна назва освітньої програми)

Керівник доц. каф. МЕЕПП Глухов О.В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_ Бондаренко І.М.  
(підпис) (прізвище, ініціали)

2023 р.



## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Електронної та біомедичної інженерії \_\_\_\_\_

Кафедра \_ Мікроелектроніки, електронних приладів та пристроїв \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 171 «Електроніка» \_\_\_\_\_

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_

Освітня програма \_\_\_\_\_ «Електронні пристрої та системи» \_\_\_\_\_

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

«\_\_\_\_» \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Тимченку Юрію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження властивостей різних типів мікроконтролерів та використання їх в електронних пристрояхзатверджена наказом університету від 20 11 2023 р. № 1369Ст2. Термін подання студентом роботи до екзаменаційної комісії 01 2024 р.3. Вихідні дані до роботи моделювання мікроконтролерів, PROTEUS PROFESSIONAL

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Провести аналіз мікроконтролерів2) Провести моделювання мікроконтролерів Texas Instruments, Microchip Technology та ST Microelectronics3) Провести вивчення та порівняння властивостей різних типів мікроконтролерів4) Проаналізувати отримані результати5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри): схеми моделювання роботи мікроконтролерів, слайд-презентація( 6 слайдів )

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	15.09.23	
2	Аналітичний огляд джерел	28.09.23	
3	Оформлення пояснювальної записки	05.10.23	
4	Вибір мікроконтролерів для моделювання	10.10.23	
5	Розробка міні проекту та її аналіз	19.10.23	
6	Оформлення креслеників	01.11.23	
7	Підготовка презентації	16.11.23	
8	Рецензування, нормоконтроль	05.01.24	
9	Захист роботи	11.01.24	

Дата видачі завдання \_\_\_\_\_ 20\_\_ р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. каф. МЕЕПП Глухов О.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить: 68 с., 31 рис., 1 табл., 30 джерел.

МІКРОКОНТРОЛЕР, МОДЕЛЮВАННЯ, PROTEUS PROFESSIONAL,  
ПЛАТА

Об'єкт дослідження – дослідження властивостей різних типів мікроконтролерів та використання їх в електронних пристроях.

Мета роботи – провести системний аналіз та порівняння властивостей різних типів мікроконтролерів з метою визначення їхньої ефективності та придатності для застосування в електронних пристроях.

Метод дослідження – вивчення та порівняння властивостей різних типів мікроконтролерів для їхнього ефективного використання у сучасних електронних пристроях.

У зв'язку з розвитком науки і техніки все частіше використовуються мікропроцесорні системи. Такі системи відрізняються повною функціональністю, високою технологічністю, адаптивністю. Часто, щоб адаптувати систему до нового завдання, досить просто змінити програму мікропроцесора. Високий ступінь адаптивності дозволяє використовувати такі системи для виконання різноманітних завдань обробки інформації.

Широке використання таких систем вимагає підготовки великої кількості фахівців, здатних розробляти та експлуатувати ці системи. Тому метою цієї роботи є надання огляду базових знань мікропроцесорної техніки, необхідних для розробки та експлуатації мікропроцесорних систем.

## ABSTRACT

The explanatory note contains: 68 pages, 31 figures, 1 table, 30 sources.

### MICROCONTROLLER, SIMULATION, PROTEUS PROFESSIONAL, BOARD

The object of the research is to study the properties of various types of microcontrollers and their use in electronic devices.

The purpose of the work is to carry out a system analysis and comparison of the properties of different types of microcontrollers in order to determine their effectiveness and suitability for use in electronic devices.

The research method is to study and compare the properties of different types of microcontrollers for their effective use in modern electronic devices.

In connection with the development of science and technology, microprocessor systems are increasingly used. Such systems are distinguished by full functionality, high technology, and adaptability. Often, to adapt the system to a new task, it is enough to simply change the microprocessor program. A high degree of adaptability allows such systems to be used to perform a variety of information processing tasks.

The widespread use of such systems requires the training of a large number of specialists capable of developing and operating these systems. Therefore, the purpose of this work is to provide an overview of the basic knowledge of microprocessor technology necessary for the development and operation of microprocessor systems.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
1 ПОНЯТТЯ ТА КЛАСИФІКАЦІЯ МІКРОКОНТРОЛЕРІВ.....	10
1.1 Визначення та поняття.....	10
1.2 Класифікація мікроконтролерів.....	14
2 ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ МІКРОКОНТРОЛЕРІВ ТА ВИКОРИСТАННЯ В ЕЛЕКТРОННИХ ПРИСТРОЯХ.....	18
2.1 Властивості та використання мікроконтролерів компанії Texas Instruments.....	18
2.2 Властивості та використання мікроконтролерів компанії Microchip Technology.....	20
2.3 Властивості та використання мікроконтролерів компанії ST Microelectronics.....	23
3 МОДЕЛЮВАННЯ РОБОТИ МІКРОКОНТРОЛЕРІВ В ПРОГРАМІ PROTEUS PROFESSIONAL.....	25
3.1 Знайомство з програмним пакетом Proteus.....	25
3.2 Моделювання MSP430G2 Texas Instruments.....	33
3.3 Моделювання PIC16F877A Microchip Technology.....	34
3.4 Моделювання STM32F103C6T6 ST Microelectronics.....	35
4 ДОСЛІДЖЕННЯ МІКРОКОНТРОЛЕРІВ В МІНІ ПРОЕКТАХ.....	37
4.1 Дослідження плати із мікроконтролером MSP430G2 Texas Instruments.....	37
4.2 Дослідження плати із мікроконтролером PIC16F877A Microchip Technology.....	44
4.3 Дослідження плати із мікроконтролером TM32F103C6T6 ST Microelectronics.....	57
ВИСНОВКИ.....	64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	66

## ПЕРЕЛІК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

- АЛП – арифметично-логічний пристрій.  
КП – керуючий пристрій.  
ЗП – запам’ятовуючий пристрій.  
ПВВ – пристрій введення/виведення.  
МПС – мікропроцесорна система.  
МП – мікропроцесор.  
ВІС – велика інтегральна схема.  
ОЗП – оперативний запам’ятовуючий пристрій.  
ПЗП – постійний запам’ятовуючий пристрій.  
ТГ – тактовий генератор.  
МК – мікроконтролер.  
РЗП – регістр загального призначення.  
РП – регістровий пристрій.  
ПЛ – програмний лічильник.  
РСП – регістр стану процесора.  
КР – керуючий регістр.  
КП – керуючий пристрій.  
ОП – операційний пристрій.  
ВКМ – внутрішній контроль магістралі.  
ШД – шина даних.  
ША – шина адреси.  
ARM – Advanced RISC Machine.  
CISC – Complex Instruction Set Computing.  
RISC – Reduced Instruction Set Computing.  
PIC – Programmable Interrupt Controller.  
ЦП – центральний процесор.  
TI – Texas Instruments.

## ВСТУП

Нинішня ситуація характеризується тим, що ринок переповнений автоматизованими пристроями різного призначення та складності, від пластикових платіжних карток до розумних кросівок, автомобілів і найскладніших пристроїв. Причиною цього явища є швидкий розвиток і автоматизоване виробництво інтегральних схем, однією з яких є компактні та повноцінні мікрокомп'ютери розміром з мікроконтролер. Сьогодні мікросхемні технології поширені у всіх сферах життя людини, електроніка в нашому середовищі зростає з року в рік.

Не так давно розробникам електронних схем доводилося мати справу з великою кількістю електронних компонентів, і їх було важко розмістити на друкованій платі. Найпоширенішим способом створення макета (прототипу) майбутнього пристрою було «на місці»: після вибору електронних компонентів розробник за допомогою паяльника збирає окремі компоненти або весь пристрій на макетній платі. Потім починається процес налагодження: виправлення помилок у схемі, встановлення режиму роботи, уточнення параметрів використовуваних компонентів тощо.

Сьогодні ми маємо багато різних автоматизованих систем проектування та розробки, які дозволяють нам значно спростити роботу розробників електронних пристроїв, одночасно зменшуючи матеріальні витрати та час випуску готової продукції, а також ризики під час процесу розробки. Використання систем автоматизованого проектування та інших інструментів може забезпечити швидку, легку та досить надійну розробку.

Найбільший набір засобів розробки призначений для використання з мікроконтролерами, створення нових пристроїв з ними тепер є необхідністю. При розробці радіоелектронної або обчислювальної техніки та засобів автоматизації використання мікроконтролерів, мікропроцесорів у поєднанні з програмованими інтегральними схемами дозволяє істотно спростити схемотехніку проекту; уніфікація елементної бази - один і той же контролер

можна використовувати для створення багатьох різних пристроїв, уніфікація, у свою чергу, призводить до зниження витрат виробництва, підвищення гнучкості розробки.

# 1 ПОНЯТТЯ ТА КЛАСИФІКАЦІЯ МІКРОКОНТРОЛЕРІВ

## 1.1 Визначення та поняття

У найпростішому випадку, згідно з принципом фон Неймана, будь-яка система обробки інформації повинна мати таке обладнання (рисунок 1.1):

- арифметико-логічний пристрій (АЛП), який виконує арифметичні та логічні операції;
- керування пристроєм (КП), організовує процес виконання програми;
- запам'ятовуючий пристрій (ЗП), або пам'ять, яка використовується для зберігання програм і даних;
- зовнішні пристрої (ПВВ) введення та виведення інформації.

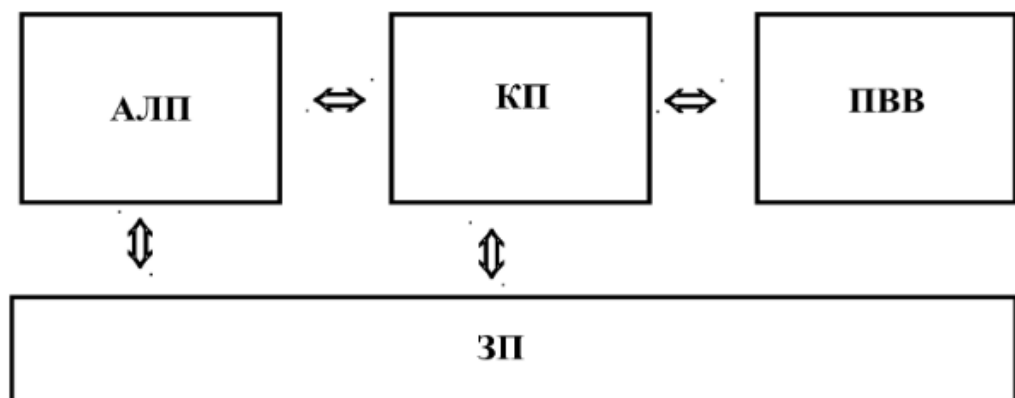


Рисунок 1.1 – Структура системи обробки інформації за принципом фон Неймана

Ці принципи частково складають основу будь-якої сучасної системи обробки інформації. Що таке сучасна мікропроцесорна система?

Мікропроцесорна система (МПС) – це функціонально-конструктивний закінчений виріб, до складу якого входить мікропроцесор. МПС призначений для прийому, аналізу, передачі та перетворення інформації.

Мікропроцесор (МП) – це програмно-керований пристрій, зазвичай побудований на великій інтегральній схемі (ІС), який керує обробкою інформації.

У сучасній техніці мікропроцесори часто об'єднують в систему з різними додатковими пристроями: операційними та енергонезалежними запам'ятовуючими пристроями (ОЗП і ПЗП), інтерфейсами, службовими пристроями (тактові генератори (ТГ), таймери, регістри) очікування. Ця асоціація називається мікроконтролером (МК). Мікропроцесори та мікроконтролери є основними компонентами МПС.

Інтерфейс – це уніфікований набір технологій і програмних засобів, необхідних для підключення зовнішніх пристроїв. Він забезпечує перетворення сигналів МП в сигнали, що сприймаються зовнішніми пристроями, і навпаки, посилення сигналу, є апаратними засобами та набором процедур передачі даних.

В даний час існує багато типів мікроконтролерів, що відрізняються архітектурою процесорного модуля, розміром і типом вбудованої пам'яті, набором периферійних пристроїв, типом корпусу і т.д.

Основою мікропроцесорних систем є мікропроцесори та мікроконтролери.

У мікропроцесорній техніці часто використовується термін «контролер». Він має різні значення залежно від характеру програми. В одному з варіантів воно використовується як назва пристрою керування периферійними пристроями мікропроцесорної системи (контролер клавіатури, контролер переривань тощо). Саме в цьому значенні цей термін буде використовуватися даній роботі.

Архітектура процесора – це наданий користувачеві комплекс апаратних і програмних засобів, а саме: набір доступних програмних регістрів і пристроїв виконання (операції), система команд і методи адресації, ємність і структура пам'яті, типи обробки переривань і методи.

Регістр процесора – це блок комірок пам'яті, які утворюють надшвидку оперативну пам'ять усередині процесора; використовується самим процесором і здебільшого недоступний для програміста: наприклад, коли наступна команда вибирається з пам'яті, вона розміщується в місці, яке недоступне для програміста, до якого звертається регістр команд.

При описі архітектури та функціональних можливостей мікропроцесора його представлення часто використовується в термінах набору доступних програмі регістрів, які утворюють регістр або модель програми. Ці регістри містять оброблені дані (операнди) та керуючу інформацію. Відповідно, регістрова модель включає набір регістрів загального призначення (РЗП) для зберігання операндів і набір службових регістрів, які забезпечують контроль за виконанням програми, режимами роботи процесора та управлінням пам'яттю.

Регістри загального призначення складають регістровий пристрій (RP), який є внутрішньою пам'яттю регістрів процесора. Склад службового реєстру визначається конкретною архітектурою мікропроцесора і зазвичай включає:

- програма лічильника (ПЛ);
- регістр стану процесора (РСП);
- регістр керування (КР).

Може бути кілька регістрів РСП і КР. Це залежить від складності процесора.

Як працює мікропроцесорна система?

У загальному вигляді роботу мікропроцесорної системи можна описати наступним чином. Спочатку введіть програму в пам'ять системи за допомогою будь-якого зовнішнього пристрою. Керуючий пристрій зчитує вміст запам'ятовуючого пристрою, де знаходиться перша інструкція програми, і організовує її виконання. Команда може визначати виконання арифметичних або логічних операцій, читання даних із пам'яті для виконання цих операцій або запис їх результатів у пам'ять, введення даних у

пам'ять із зовнішнього пристрою або виведення даних із пам'яті на зовнішній пристрій.

Зазвичай після виконання команди пристрій керування починає виконувати команду з місця зберігання, розташованого відразу після щойно виконаної команди. Однак цей порядок можна змінити за допомогою спеціальних команд передачі керування, які вказують місце пам'яті, з якого має продовжуватися виконання програми. Це перетворення може бути організоване таким чином, що дозволяє створювати складні програми.

Результати роботи процесора можна представити як зміни стану регістра, які відбуваються внаслідок виконання операторів програми або обробки сигналів із зовнішніх пристроїв. Зміни в стані реєстру спричинять зміни в усьому стані системи. Зміна стану регістра викликає зміну потенціалу на зовнішньому виводі мікроконтролера, пов'язаного з ним. Це дозволяє керувати зовнішніми пристроями (вмикати, вимикати, змінювати режим їх роботи).

Цей опис є дещо спрощеним. Реальні системи на основі мікропроцесорів і мікроконтролерів мають більш складну архітектуру і тому дещо інші алгоритми роботи. Приклад блок-схеми сучасного мікроконтролера показаний на рис. 1.2.

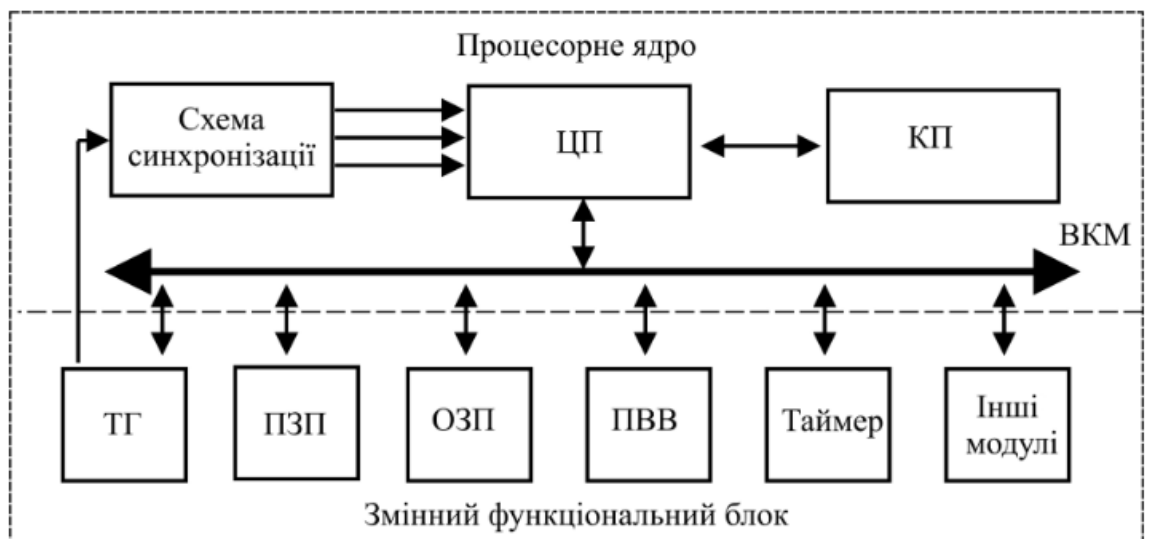


Рисунок 1.2 – Структурна схема сучасного мікроконтролера

Навіть мікроконтролери з однаковою архітектурою можуть відрізнитися набором функціональних вузлів і характером зв'язку між ними.

Однак будь-який мікроконтролер містить наступні функціональні блоки:

- центральний процесор або ядро процесора, що складається з пристроєм керування (КП) і пристроєм керування (ОП);
- внутрішня пам'ять (РП, кеш, ОЗП і блоки ПЗП);
- блок інтерфейсу, що забезпечує доступ до системної шини та обмін даними із зовнішніми пристроями через паралельні або послідовні порти введення/виведення;
- периферійне обладнання (таймерний модуль, аналого-цифровий перетворювач, спеціальний контролер);
- різні допоміжні схеми (генератори тактових імпульсів, схеми налагодження та тестування, сторожові таймери тощо).

Зв'язок між різними компонентами мікропроцесорної системи здійснюється за допомогою внутрішньої шини керування (ВШК), яка складається з шини даних (ШД), шини адреси (ША) і шини керування (ШК). Синхронізація роботи компонентів мікропроцесорної системи досягається за допомогою тактових генераторів (ТГ) і спеціальних схем синхронізації [1, 2].

## 1.2 Класифікація мікроконтролерів

Існуючі мікроконтролери можна класифікувати за різними характеристиками. За розрядністю шини даних мікроконтролери можна розділити на наступні категорії:

- 1-розрядний;
- 4-розрядний;
- 8-розрядний;
- 16-розрядний;
- 32-розрядний.

З часом можуть стати доступними мікроконтролери з більшими шинами даних, оскільки існують 64-розрядні мікропроцесори.

За своїм функціональним призначенням мікроконтролери можна розділити на мікроконтролери загального призначення та мікроконтролери спеціального призначення. Спеціалізації можна розділити на мікроконтролери для цифрової обробки сигналів, мікроконтролери для керування двигунами та інші (рис. 1.3).

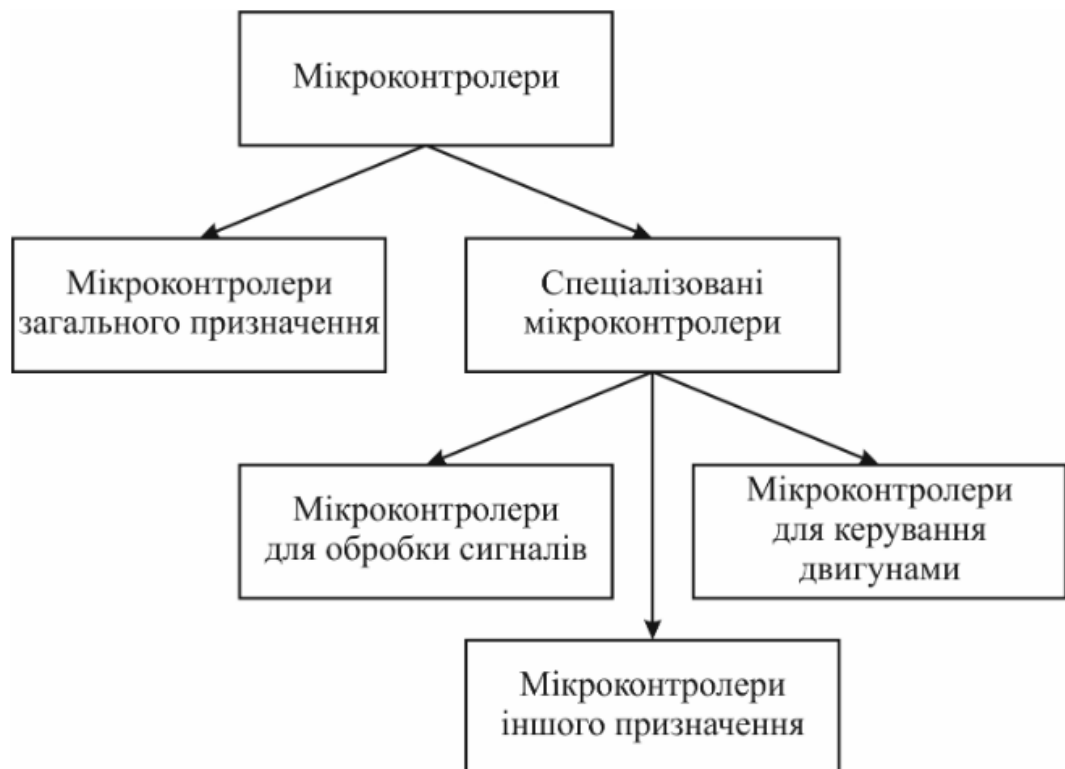


Рисунок 1.3 – Класифікація МК за призначенням

За типом архітектури МК можна розділити на архітектури CISC (Complex Instruction Set Computing) і RISC ((Reduced Instruction Set Computing)). Особливістю архітектури CISC є те, що команди виконуються послідовно та мають різну довжину та структуру.

Вибір команд з пам'яті відбувається побайтно, а самі команди виконуються протягом декількох циклів. Архітектури CISC включають: MC

серії Motorola HC05/HC08, MC з ядром MCS-51, MC серії Infineon C500 та інші.

Основна ідея RISC-архітектури полягає в заміні складних інструкцій простими однотипними і виконанням їх в одному потоці в паралельному конвеєрі. Усі команди мають фіксовану довжину та в ідеалі повинні виконуватися за один такт замість кількох, що покращує продуктивність. Саме ця архітектура вважається більш перспективною і в даний час є більш поширеною.

Контролер PIC 16C54 від Microchip був одним із перших мікросхем, які прийняли архітектуру RISC. Контролери PIC швидко стали популярними в усьому світі завдяки високій продуктивності та трьом десяткам простих команд. Невдовзі розробники з таких компаній, як Atmel, Scenix та інших, наслідували цей приклад.

Іноді при класифікації за типом архітектури окремо виділяють МК на базі ARM. Архітектура ARM (походить від англ. Advanced RISC Machine – вдосконалена RISC-машина) – це RISC-архітектура, заснована на ліцензованих 32-розрядних і 64-розрядних ядрах мікропроцесорів, розроблених ARM Limited.

За характером роботи з пам'яттю мікроконтролери можна розділити на два типи: принстонський і гарвардський. Принстонський тип був розроблений незалежно Джоном фон Нейманом і академіком С. О. Лебедевим. він використовує

Спільна пам'ять для зберігання програм і даних (рис. 1.4). Основними перевагами є спрощення схеми центрального процесора (CPU) і гнучкість розподілу ресурсів між областями зберігання.

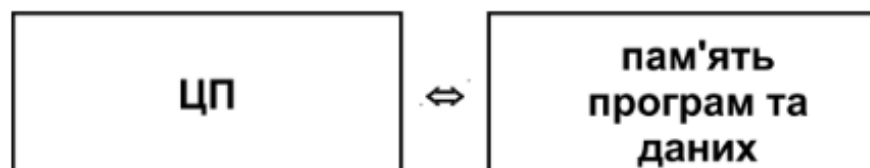


Рисунок 1.4 – Робота із пам'яттю по пристанському типу

Характерною рисою гарвардського типу є наявність окремих адресних просторів для зберігання команд і даних (рис. 1.5). Цей тип майже не використовувався до кінця 1970-х років, коли розробники МК нарешті зрозуміли, що це дає їм певну перевагу. Зокрема, аналіз реальних програм показує, що обсяг пам'яті даних МК, який використовується для зберігання проміжних результатів, приблизно на порядок менший, ніж необхідний обсяг пам'яті програм. В результаті можна знизити розрядність шини даних, зменшивши кількість транзисторів у мікросхемі при одночасному прискоренні доступу до інформації. Тому зараз більшість сучасних МК використовують RISC-архітектуру гарвардського типу.



Рисунок 1.5 – Робота із пам'яттю по гарвардському типу

Мікроконтролери діляться на кілька сімейств. Продукти з однаковим ядром належать до сімейства – набору понять, таких як система команд, схема циклу для операцій центрального процесора, організація пам'яті програм і даних, система переривань і базовий набір периферійних пристроїв. Відмінності між різними представниками одного сімейства полягають в основному в складі периферійних пристроїв і обсязі зберігання програм або даних. Найважливішою особливістю цієї серії є програмна сумісність на рівні двійкового коду для всіх МК серії. Як правило, фірми-виробники дають назву своїй серії МК. Наприклад, PIC16 – це сімейство 16-розрядних мікроконтролерів Microchip, а AVR – сімейство мікроконтролерів Atmel.

## 2 ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ МІКРОКОНТРОЛЕРІВ ТА ВИКОРИСТАННЯ В ЕЛЕКТРОННИХ ПРИСТРОЯХ

### 2.1 Властивості та використання мікроконтролерів компанії Texas Instruments

Компанія Texas Instruments (TI) була в авангарді виробництва мікроконтролерів, які зробили революцію у світі електроніки. Ці невеликі пристрої знайшли свій шлях до різних застосувань, від керування побутовою технікою до автоматизації промислових процесів.

Однією з ключових особливостей мікроконтролерів TI є їх універсальність і сумісність із широким спектром периферійних пристроїв і програмного забезпечення. TI пропонує мікроконтролери з різними архітектурами, включаючи ARM, MSP430 і C2000, кожен зі своїми унікальними функціями та можливостями. Мікроконтролери TI також пропонують ряд комунікаційних інтерфейсів, включаючи Ethernet, USB і CAN, що робить їх придатними для використання в мережевих і комунікаційних системах. Крім того, мікроконтролери TI відомі своїм низьким енергоспоживанням, що робить їх ідеальними для використання в пристроях з батарейним живленням.

Сімейство мікроконтролерів F2837xD є одним із сімейств мікроконтролерів, що підтримується Energia IDE. Це використовується для розробки основи для реалізації стратегій цифрового контролю та управління ресурсами. Крім того, сімейство MSP430 x5xx є мікроконтролером TI, який використовується завдяки хорошим енергетичним властивостям у різних сферах. Він має наднизьке енергоспоживання і тому підходить для пристроїв, які вимагають тривалого часу автономної роботи. Мікроконтролер також має можливість генерувати випадкові числа для криптографічних систем, і компанія Texas Instruments пропонує рішення для генерації випадкових чисел у MSP430. Крім того, мікроконтролери мають доступ до налагодження та

апаратний співпроцесор AES, який може стати мішенню для атак із бічного каналу та атак диференціального аналізу помилок. Мікроконтролери платформи SimpleLink від Texas Instruments також були досліджені на сприйнятливність до неінвазивних фізичних атак. Ці мікроконтролери мають функціональні можливості, що залишилися від процесу виробництва інтегральних схем, якими можна зловживати під час атак, а також завантажувач ПЗУ, який можна отримати та проаналізувати за допомогою статичного аналізу та емуляції. Загалом, мікроконтролери Texas Instruments можна використовувати для реалізації стратегій цифрового керування, управління ресурсами та безпеки.

Мікроконтролери Texas Instruments мають ряд переваг і недоліків, які необхідно зважити при розгляді проекту. Перевагою мікроконтролерів ТІ є простота використання GUI для програмування TMS320F28379xD, що дозволяє користувачам швидко та інтуїтивно зрозуміло вносити зміни. Крім того, компенсація холодного спаю є простою та зрозумілою, з тим самим TDC (TMP121, Texas Instruments), що використовується для всіх моделей. Крім того, таймерну систему мікроконтролера можна використовувати для вимірювання вхідних сигналів і підрахунку кількості зовнішніх подій, що дозволяє ефективно керувати часом. Однак відсутній повний аналіз генератора та його властивостей, а також можливості використовувати MSP430 у повній мірі. Крім того, розмір блоку та свобода використання користувача з будь-яким мікроконтролером, крім ТІ MSP430, обмежені. Нарешті, деякі реалізації з недорогим мікроконтролером/DSP не використовують оптимізацію MPPT, що знижує ефективність системи [3].

У порівнянні з іншими мікроконтролерами на ринку, мікроконтролери ТІ виділяються своєю високою продуктивністю, можливостями керування в реальному часі та універсальністю. Мікроконтролери ТІ пропонують широкий спектр функцій і можливостей, які недоступні в інших мікроконтролерах. Наприклад, 32-розрядні контролери цифрових сигналів (DSC) ТІ розроблені для забезпечення високопродуктивних можливостей

обробки сигналів, що робить їх придатними для використання в таких додатках, як керування двигунами та силова електроніка. Мікроконтролери TI також відомі своєю простотою використання та надійними інструментами розробки, які полегшують розробникам проектування та тестування своїх додатків.

Мікроконтролери Texas Instruments знаходять широке застосування в автомобільній промисловості, де вони використовуються для керування двигуном, силовим агрегатом і системами безпеки. Ці мікроконтролери розроблені для роботи в суворих умовах навколишнього середовища та високих вимог до продуктивності сучасних автомобілів. Мікроконтролери забезпечують високий рівень точності, дозволяючи розробляти передові системи, які можуть підвищити економію палива, зменшити викиди та підвищити загальну продуктивність автомобіля. Вони також використовуються для розробки передових систем допомоги водієві (ADAS), які включають такі функції, як запобігання зіткненням, попередження про виїзд зі смуги руху та адаптивний круїз-контроль [4, 5].

Іншою галуззю, яка значною мірою покладається на мікроконтролери Texas Instruments, є виробництво медичного обладнання. Ці мікроконтролери використовуються в широкому спектрі медичних пристроїв, включаючи кардіостимулятори, монітори рівня глюкози та інсулінові помпи. Вони розроблені для забезпечення високого рівня точності та надійності, а також низького енергоспоживання, що робить їх ідеальними для використання в портативних медичних пристроях [6-8].

## 2.2 Властивості та використання мікроконтролерів компанії Microchip Technology

Microchip Technology є провідним постачальником рішень для мікроконтролерів, що пропонує широкий спектр продуктів, розроблених для задоволення потреб різних галузей. Портфоліо компанії включає

мікроконтролери, аналогові продукти та рішення Flash IP, усі вони підкріплені винятковою технічною підтримкою. Технологія Microchip має довгу історію інновацій і була рушійною силою в розробці нових технологій мікроконтролерів, таких як мікроконтролер PIC.

Важливість мікроконтролерів у сучасній техніці неможливо переоцінити. Вони використовуються у всьому, від простих побутових приладів до складних промислових машин, і вони відіграють вирішальну роль у роботі багатьох сучасних систем. Мікроконтролери є важливими компонентами Інтернету речей (IoT), які дозволяють пристроям безперервно підключатися та спілкуватися один з одним. Вони також використовуються в автомобільній промисловості, медичних пристроях і багатьох інших сферах, де потрібне точне керування та контроль. Оскільки технологія продовжує розвиватися, мікроконтролери й надалі залишатимуться важливою частиною цифрового ландшафту.

Можливості пам'яті та обробки є важливими характеристиками мікроконтролерів, і Microchip Technology пропонує широкий спектр можливостей для цього. Пам'ять і можливості обробки мікроконтролерів Microchip оптимізовані для високої продуктивності та низького енергоспоживання, що дозволяє ефективно виконувати складні програми. Ці мікроконтролери мають флеш-пам'ять, EEPROM і SRAM, що забезпечує достатньо місця для зберігання програм і даних. Крім того, мікроконтролери Microchip оснащені розширеними можливостями обробки, включаючи апаратні операції множення та ділення, що забезпечує швидку та ефективну обробку даних.

Мікроконтролери Microchip також пропонують широкий спектр інтерфейсів вводу/виводу та протоколів зв'язку [1]. Ці інтерфейси та протоколи забезпечують бездоганну інтеграцію з іншими пристроями та системами, що робить мікроконтролери Microchip ідеальними для використання в широкому діапазоні програм. Деякі з інтерфейсів

вводу/виводу та протоколів зв'язку, які підтримуються мікроконтролерами Microchip, включають:

- USB;
- Ethernet;
- SPI;
- I2C.

Ці інтерфейси та протоколи призначені для забезпечення надійного та ефективного зв'язку між мікроконтролером та іншими пристроями чи системами. Простота використання цих мікроконтролерів додатково підвищується завдяки їх сумісності з широким спектром інструментів розробки та програмного забезпечення.

Ще одне важливе застосування мікроконтролерів Microchip - промислова автоматизація та системи керування. Ці мікроконтролери використовуються для моніторингу та керування різними промисловими процесами, включаючи виробництво, складання та пакування. Завдяки високошвидкісним можливостям обробки та надійним протоколам зв'язку мікроконтролери Microchip дозволяють системам промислової автоматизації працювати з точністю та ефективністю. Ці мікроконтролери також пропонують розширені функції безпеки, забезпечуючи безпеку та цілісність промислових систем керування.

Мікроконтролери Microchip також широко використовуються в автомобільних і аерокосмічних системах. Ці мікроконтролери використовуються в різних додатках, включаючи керування двигуном, керування трансмісією та системи безпеки. Завдяки високій надійності, низькому енергоспоживанню та міцній конструкції мікроконтролери Microchip ідеально підходять для використання в суворих умовах, наприклад, в автомобільній та аерокосмічній промисловості. Крім того, ці мікроконтролери пропонують розширені функції, такі як обробка даних у реальному часі та можливості зв'язку, що забезпечує ефективну та ефективну роботу складних автомобільних і аерокосмічних систем.

Мікроконтролери стали невід'ємною частиною сучасних технологій, і Microchip Technology є провідним виробником мікроконтролерів. Оскільки попит на меншу, швидшу та ефективнішу технологію зростає, розуміння властивостей і використання мікроконтролерів від Microchip Technology має вирішальне значення для інженерів, розробників і дослідників. Таким чином, це дослідження дасть цінну інформацію про світ мікроконтролерів та їх практичне застосування.

Microchip Technology Inc. пропонує мікроконтролери сімейства PIC, які можна використовувати в однофазних лічильниках електроенергії [9-10].

### 2.3 Властивості та використання мікроконтролерів компанії ST Microelectronics

ST Microelectronics є провідним постачальником мікроконтролерів, які широко використовуються в різних додатках. У цій дослідницькій роботі ми дослідимо властивості та використання мікроконтролерів ST Microelectronics. Спочатку ми обговоримо властивості цих мікроконтролерів, включаючи їх продуктивність, енергоспоживання та варіанти підключення. Потім ми розглянемо особливості мікроконтролерів ST Microelectronics, такі як розмір пам'яті та швидкість обробки. Нарешті, ми дослідимо застосування цих мікроконтролерів у різних галузях, включаючи автомобільну, промислову та споживчу електроніку. До кінця цього підрозділу читачі краще зрозуміють можливості мікроконтролерів ST Microelectronics та їх потенційне використання в різних галузях промисловості.

Мікроконтролери ST Microelectronics засновані на ядрах Arm Cortex-M, що пропонує широкий спектр пристроїв. Вони є ідеальним вибором для невеликих проектів, а також для наскрізних платформ, оскільки коштують менше 0,10 доларів за одиницю та є ефективними. MCU оптимізовано за вартістю та енергочутливістю, а середня ціна 32-розрядного мікроконтролера становить менше 1 долара. Сімейство STM32 пропонує широкий спектр

пристроїв із ядром Cortex-M0, що знижує рівень шуму. Крім того, вони пропонують значні переваги в додатках початкового рівня та відповідають вимогам до обчислювальної продуктивності за рахунок оптимізації тактової частоти. Набір інструкцій RISC Arm дозволяє виконувати прості інструкції, а базова архітектура забезпечує наднизьке енергоспоживання. Крім того, вони мають невеликий розмір і можуть бути легко перепрограмовані, якщо вони не працюють належним чином. Комунікаційні периферійні пристрої, які підтримуються MCU, включають USB, I2C, SPI та UART, тоді як ST Microelectronics пропонує велику різноманітність STM32 MCU. Вибір MCU залежить від функціональних вимог програми, таких як наднизьке енергоспоживання, висока продуктивність або бездротове підключення. Інші фактори, які слід враховувати, включають архітектуру, вимоги до пам'яті та живлення, периферійні пристрої та вартість. Мікроконтролери STM32 базуються на процесорі Arm Cortex-M, і ліцензійна угода про оціночний продукт ST Microelectronics забороняє на використанні цих наборів у будь-якій виробничій системі або будь-якому продукті, який пропонується для продажу [11-19].

## 3 МОДЕЛЮВАННЯ РОБОТИ МІКРОКОНТРОЛЕРІВ В ПРОГРАМІ PROTEUS PROFESSIONAL

### 3.1 Знайомство з програмним пакетом Proteus

Можливо, багато радіоаматорів стикаються з ситуацією, коли через відсутність досвіду, або через помилки в плануванні, або в силу інших обставин, вирішивши зібрати вподобаний і безсумнівно потрібний пристрій, вони просто спалюють здобуте дорогі електронні компоненти. Швидше за все, багато людей розчарувалися першою невдачею і назавжди кинули радіоелектроніку.

У наше століття повсюдної комп'ютеризації люди знайшли вихід із цього глухого кута. З'явилася велика кількість програм-симуляторів, які замінюють реальні радіодеталі та обладнання віртуальними моделями. Симулятор дозволяє налаштовувати роботу схеми без збирання реального пристрою, знаходити помилки, отримані на етапі проектування, видаляти непотрібні функції та інше.

PROTEUS VSM є однією з таких програм. Але симуляція радіокомпонентів – не єдина можливість програми. Proteus VSM, створений компанією Labcenter Electronics на базі ядра SPICE3F5 Університету Берклі, є так званим наскрізним середовищем проектування, що означає створення пристрою, починаючи з його графічного зображення (схеми) і закінчуючи друкованою версією пристрою. Виготовлення друкованої плати завершено, і її можна контролювати на кожному етапі виробництва.

Однак, незважаючи на те, що ця програма виглядає складною, нею можуть користуватися не тільки професіонали в галузі радіоелектроніки, а й новачки, які навчилися або не навчилися відрізнити резистори від транзисторів.

PROTEUS VSM складається з двох незалежних програм ISIS і ARES. ARES – програма для трасування друкованих плат, яка дозволяє створювати

власну бібліотеку корпусів і не розглядатиметься в цій роботі. Основною програмою є ISIS, яка забезпечує гаряче зв'язування ARES для передачі дизайну на плату.

«Сфера впливу» PROTEUS VSM коливається від найпростіших аналогових пристроїв до складних систем, створених на популярних сьогодні мікроконтролерах. Він має величезну бібліотеку моделей елементів, які користувачі можуть додавати самостійно. Звичайно, для цього потрібно досконально розбиратися в роботі компонента і вміти його програмувати. Здатність оживляти складні схеми робить цю програму чудовим навчальним посібником на шкільних та університетських курсах. Багаті інструменти та функції, включаючи вольтметри, амперметри, осцилографи, різні генератори та можливості налагодження програмного забезпечення мікроконтролерів, роблять PROTEUS VSM хорошим помічником для розробників електронних пристроїв.

PROTEUS VSM – це середовище для проектування та моделювання роботи електронних схем. Пакет Proteus VSM відрізняється здатністю імітувати роботу програмованих пристроїв:

Мікроконтролер (МК), мікропроцесор, DSP та ін. PROTEUS VSM дозволяє дуже надійно моделювати та налагоджувати досить складні пристрої, які можуть містити кілька МК одночасно або навіть різні родини в одному пристрої.

PROTEUS складається з двох основних модулів:

– ISIS – графічний редактор схем для введення розроблених проектів і подальшого моделювання та передачі для розробки друкованих плат в ARES. Крім того, після завершення налагодження обладнання друковану плату можна відразу розмістити в ARES, а він підтримує автоматичне розміщення та відстеження відповідно до існуючих рішень.

– ARES – графічний редактор друкованих плат із вбудованим менеджером бібліотек і автомаршрутизатором ELECTRA для автоматичного розміщення компонентів на друкованих платах.

На рисунку 3.1 показано головне вікно програми.

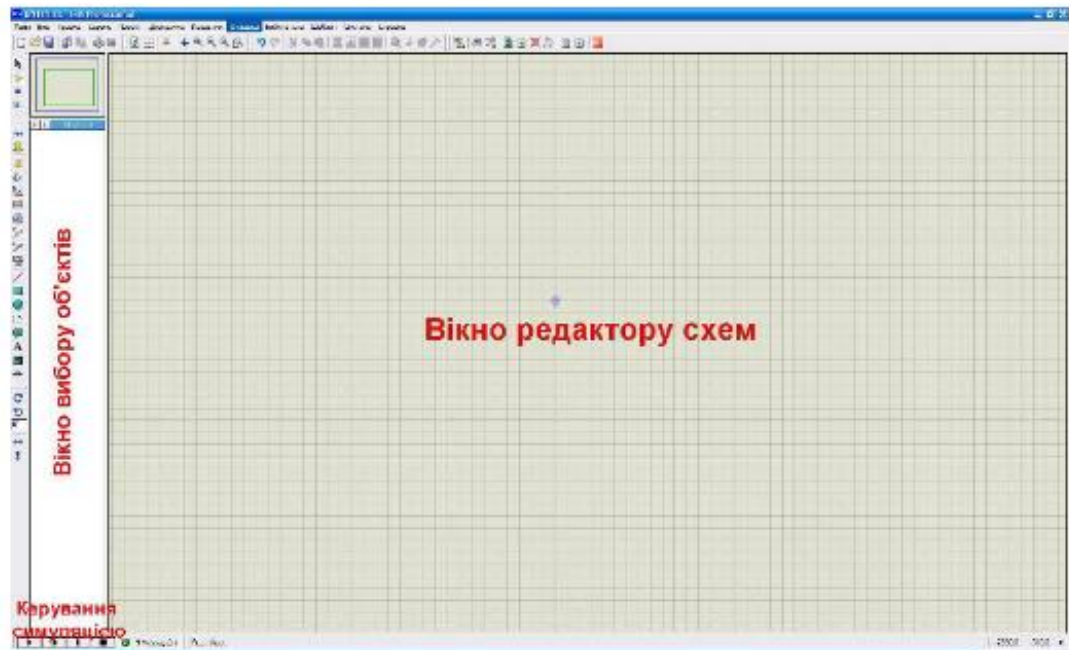


Рисунок 3.1 – Головне вікно Proteus

Вся робоча область програми ділиться на кілька основних частин:

- вікно редактора рішення (здійснює синтез кожного компонента);
- вікно вибору об'єкта (доступні різні елементи в залежності від обраного режиму);
- панель керування симуляцією (розташована в нижньому лівому куті, містить наступні команди: Почати; Виконати захід, що включає час окремого кроку симуляції, який встановлюється в розділі головного меню «System» Set Animation Options; Пауза; СТОП).

Для того щоб зібрати схему будь-якого пристрою, необхідно підготувати набір елементів, з яких буде складатися схема. Для цього перейдіть у режим Компонент (рисунок 3.2a) і натисніть клавішу P, яка розташована поруч із клавішею L під вікном перегляду (рисунок 3.2b).



Рисунок 3.2 – Панель інструментів

Перед нами з'являється Менеджер компонентів, який пропонує нам вибрати всі елементи, що містяться в бібліотеці (рисунок 3.3).

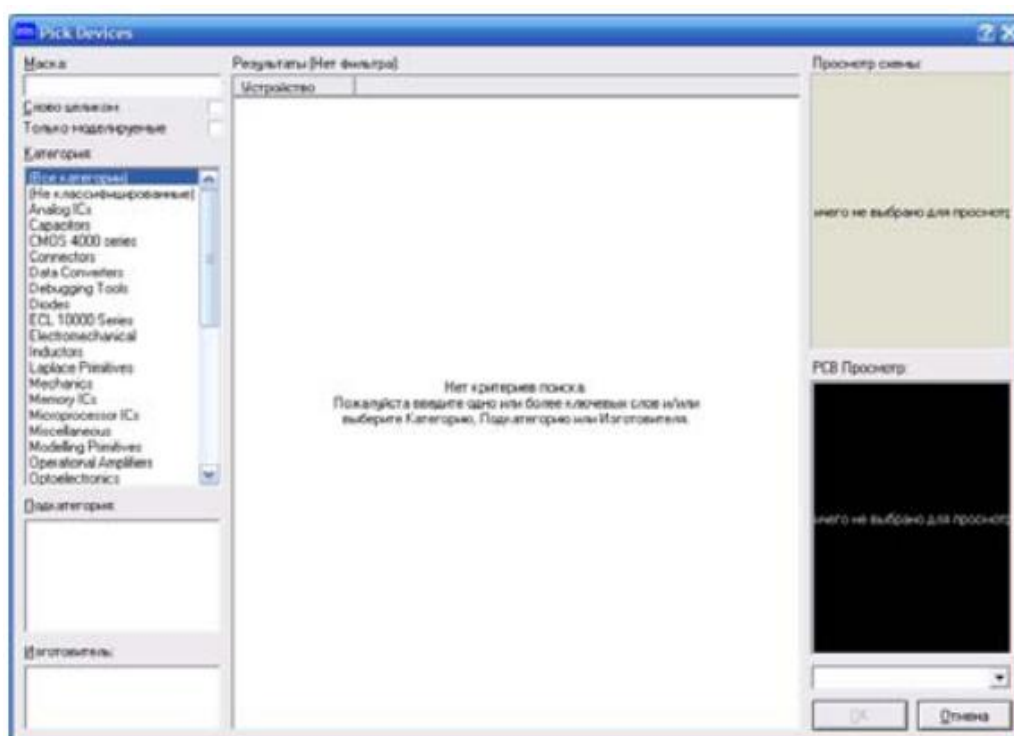


Рисунок 3.3 – Вікно менеджера компонентів

Вам потрібно скористатися рядком пошуку, розташованим у верхньому лівому куті. Коли потрібний компонент знайдено, подвійне клацання лівою кнопкою миші на його назві додають його до списку використовуваних компонентів. Для прикладу створимо електронну схему, до складу якої

входять: світлодіод, резистор і блок живлення (акумулятор). У рядок пошуку (маска) введіть перший елемент, який ви хочете додати до схеми: led-green. У списку елементів (рисунок 3.4) з'явиться світлодіод з такою назвою, на ньому двічі клацніть лівою кнопкою миші.

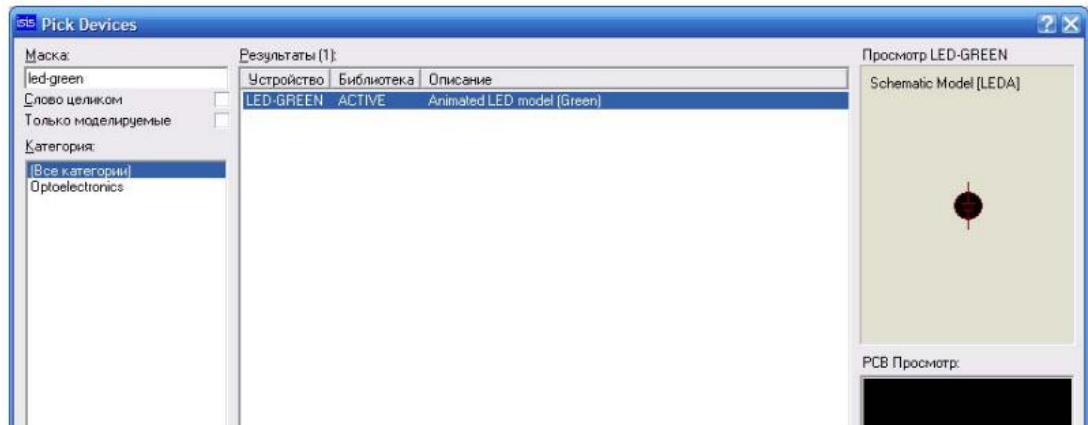


Рисунок 3.4 – Компонент пошуку

Далі таким же чином шукаємо такі назви: Resistor, Battery (Library - Active) і двічі клацаємо ці елементи в списку, щоб додати їх до нашого проекту. Вибравши всі компоненти, натисніть кнопку ОК. Наше вікно вибору об'єкта тепер міститиме всі чотири компоненти, які ми вибрали, як показано на рисунку 3.5.

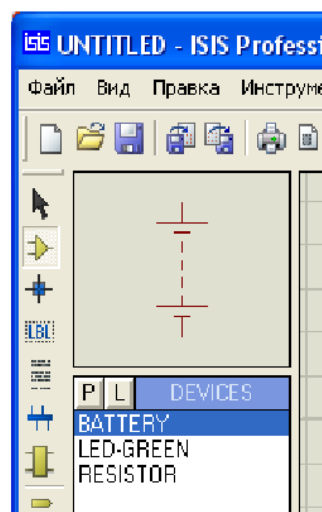


Рисунок 3.5 – Вікно вибору об'єкта

Для встановлення компонента у вікні редактора сценаріїв необхідно вибрати його зі списку та подвійним клацанням лівої кнопки миші встановити в потрібне місце. Перед монтажем компонентів на схемі їх можна попередньо повернути в потрібне положення, яким можна керувати в оглядовому вікні.

Встановіть усі компоненти, як показано на рисунку 3.6 (а), і з'єднайте їх (рисунок 3.6 б). Щоб з'єднати два компоненти разом, спочатку виберіть інструмент зі стрілкою, потім на самій діаграмі підведіть курсор миші до кінця елемента, повинен з'явитися квадрат червоного кольору, натисніть і відпустіть ліву кнопку миші, потім перемістіть курсор до іншої точки контакту одиниці, на обох кінцях іншої одиниці, ми також клацаємо лівою кнопкою миші, щоб завершити побудову зв'язку між двома одиницями.

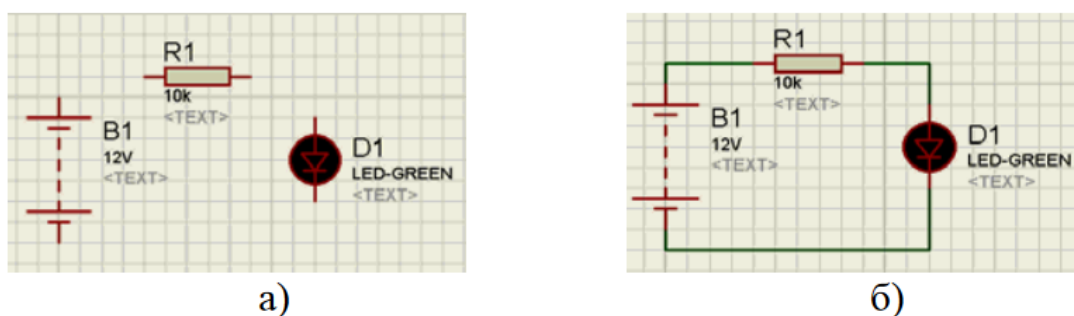


Рисунок 3.6 – Структура схеми

Після створення схеми ми змінюємо властивості компонентів. Тепер, коли батарея живить схему на 12 В, нам потрібно змінити джерело живлення схеми на 3 В. Для цього двічі клацніть по акумулятору і в полі Напруга встановіть 3 В.

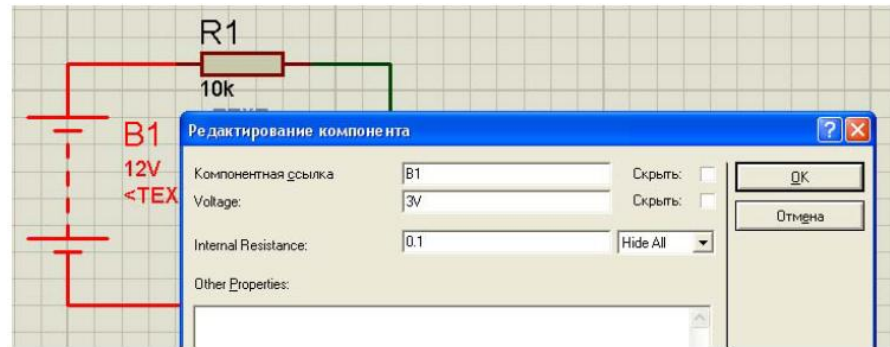


Рисунок 3.7 – Зміна параметрів живлення

Також ми змінили параметри світлодіодів. У нашому випадку ми маємо зелений світлодіод, тому у вікні параметрів світлодіода виставляємо падіння напруги 2В і робочий струм 20мА.

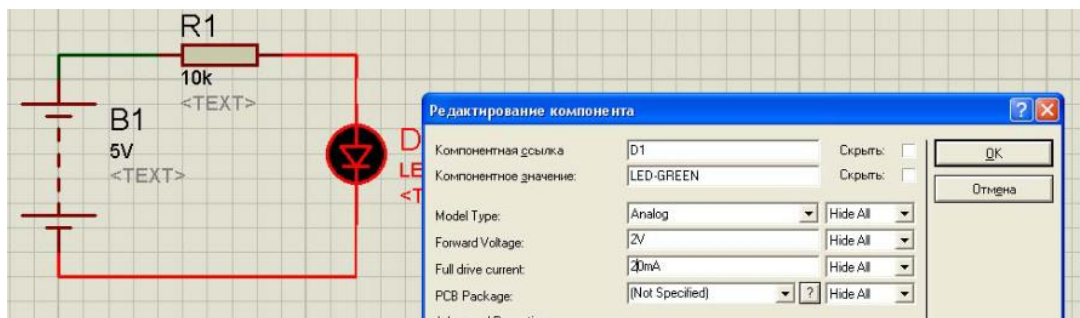


Рисунок 3.8 – Зміна параметрів опору

Далі розрахуємо опір, необхідний для нормальної роботи світлодіода. Для цього розраховуємо, яка напруга вийде на світлодіоді:

$$U_{\text{Extremeer}} = 3\text{V} - 2\text{V} = 1\text{V}. \quad (3.1)$$

Відповідно до закону Ома необхідне значення опору резистора:

$$R = \frac{U}{I} = \frac{1}{0,02} = 50 \text{ Ом}. \quad (3.2)$$

У симуляторі ми можемо записати резистор 50 Ом, але коли нам потрібно спроектувати реальну схему, оскільки такого резистора немає в стандартній серії резисторів, ми візьмемо той, який має найближчий напрямок зростання - 51 Ом.

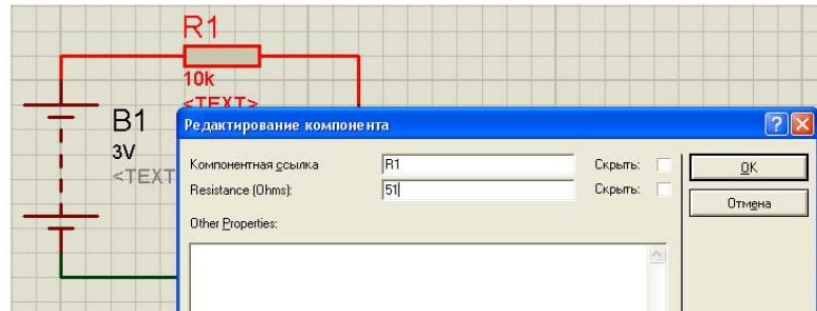


Рисунок 3.9 – Зміна параметрів опору

Потім натисніть кнопку Play на панелі керування моделюванням.



Рисунок 3.10 – Аналогова панель керування

Тепер наше рішення може віртуально відтворити ввімкнення світлодіода (рис. 3.11) [20].

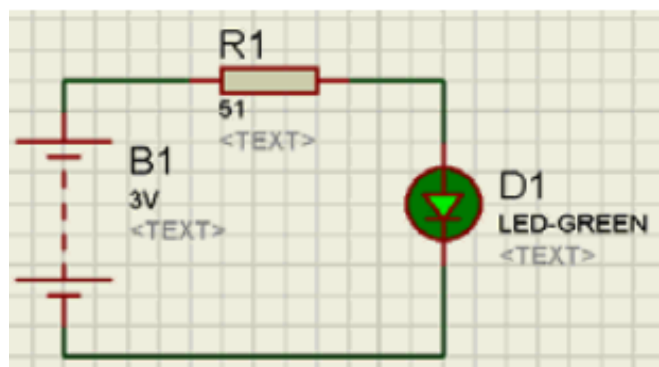


Рисунок 3.11 – Моделювання роботи світлодіода

### 3.2 Моделювання MSP430G2 Texas Instruments

Сімейство мікроконтролерів із наднизьким енергоспоживанням Texas Instruments MSP430 складається з кількох пристроїв із різними наборами периферійних пристроїв, призначених для різних програм. Архітектура в поєднанні з п'ятьма режимами низького енергоспоживання оптимізована для досягнення подовженого терміну служби акумулятора в портативних вимірювальних програмах. Пристрій має потужний 16-розрядний процесор RISC, 16-розрядні регістри та генератори констант, які сприяють максимальній ефективності коду. Генератор з цифровим керуванням (DCO) дозволяє прокидатися з режимів низького енергоспоживання в активний режим менш ніж за 1 мкс.

Моделювання MSP430G2 в Proteus Professional є важливим кроком у процесі розробки для багатьох інженерів і дизайнерів. Створивши віртуальну модель MSP430G2, розробники можуть перевірити та вдосконалити свої конструкції, перш ніж приступити до фізичного виробництва. Це може заощадити час і гроші, а також зменшити ризик помилок або недоліків конструкції. Крім того, моделювання MSP430G2 у Proteus Professional дозволяє розробникам досліджувати нові можливості дизайну та експериментувати з різними конфігураціями та налаштуваннями. Загалом, можливість моделювання MSP430G2 у Proteus Professional є цінним інструментом для будь-кого, хто працює у сфері електронного проектування та розробки [21].

Моделювання якого представлено на рисунку 3.12.

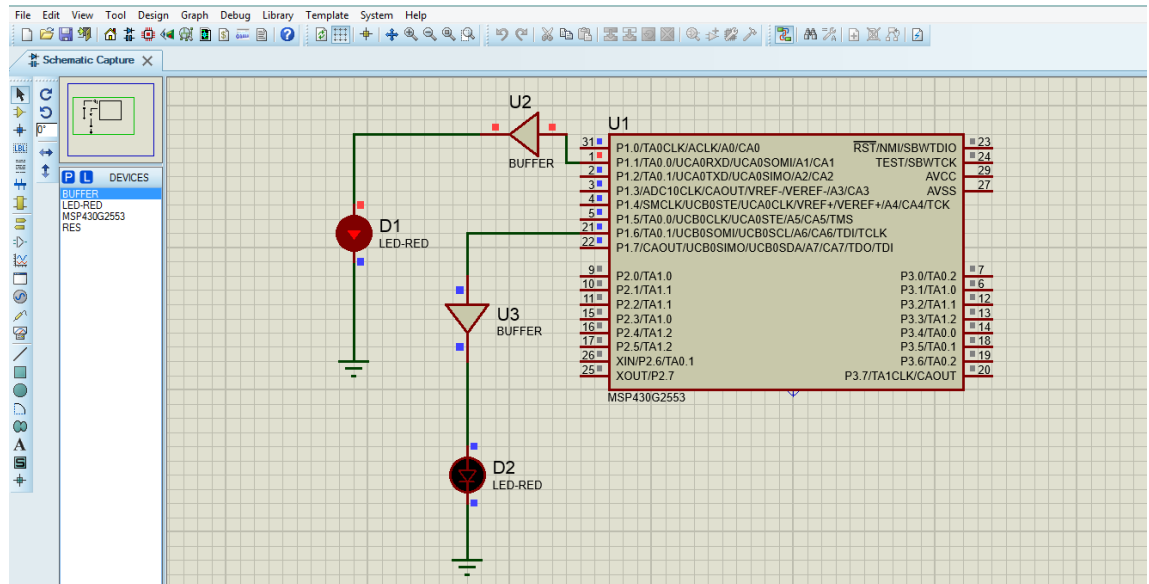


Рисунок 3.12 – Моделювання МК MSP430G2 Texas Instruments

### 3.3 Моделювання PIC16F877A Microchip Technology

Мікроконтролер PIC PIC16F877A є одним із найвідоміших мікроконтролерів у галузі. Цей мікроконтролер дуже зручний у використанні, кодування або програмування цього контролера також простіше. Одна з головних переваг полягає в тому, що його можна записувати-стирати стільки разів, скільки можливо, оскільки він використовує технологію FLASH-пам'яті. Він має загальну кількість 40 контактів і є 33 контакти для входу та виходу. PIC16F877A використовується в багатьох проектах мікроконтролерів. PIC16F877A також має широке застосування в схемах цифрової електроніки.

Proteus Professional є чудовим інструментом моделювання для проектів на основі мікросхеми PIC16F877A. Програмне забезпечення дозволяє користувачам імітувати роботу мікроконтролера, включаючи його входи, виходи та внутрішні регістри. Ця можливість дає змогу тестувати та налагоджувати проекти на основі мікроконтролерів до того, як вони будуть реалізовані в апаратному забезпеченні, заощаджуючи час і знижуючи ризик помилок. Proteus Professional також підтримує широкий спектр периферійних

пристроїв, включаючи РК-дисплеї, датчики та виконавчі механізми, що робить його ідеальним інструментом для моделювання складних систем на основі мікроконтролерів [22].

Моделювання якого представлено на рисунку 3.13.

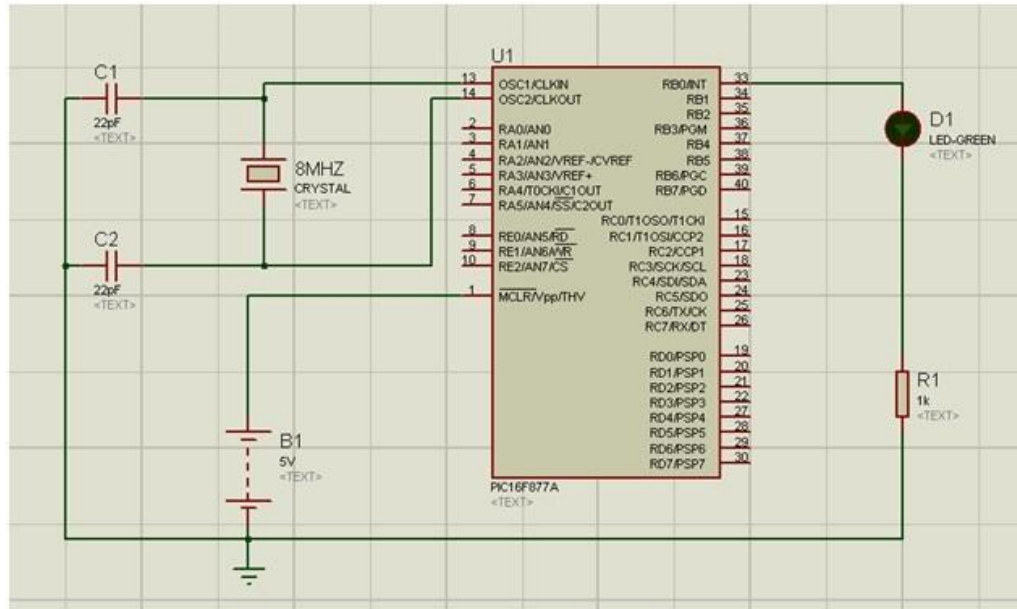


Рисунок 3.13 – Моделювання МК PIC16F877A

### 3.4 Моделювання STM32F103C6T6 ST Microelectronics

У сфері мікроконтролерів STM32F103C6T6 займає високу позицію, демонструючи свою присутність як потужна, універсальна та широко поширена технологія. Детально зануримося у внутрішню роботу, функції та області застосування цього популярного мікроконтролера, пропонуючи інформацію як початківцям, так і досвідченим користувачам.

STM32F103C6T6 – це мікроконтролер середньої щільності на основі ARM Cortex-M3, виготовлений компанією ST Microelectronics. Завдяки високій продуктивності, можливостям роботи в режимі реального часу та роботі з низьким енергоспоживанням не дивно, що цей пристрій викликає хвилю у світі технологій.



## 4 ДОСЛІДЖЕННЯ МІКРОКОНТРОЛЕРІВ В МІНІ ПРОЕКТАХ

### 4.1 Дослідження плати із мікроконтролером MSP430G2 Texas Instruments

Плата мікроконтролера Texas Instruments MSP430G2 є популярним і універсальним інструментом для розробки вбудованих систем. Плата пропонує низку функцій, включаючи низьке енергоспоживання, 16-розрядну RISC-архітектуру та різноманітні комунікаційні інтерфейси. Крім того, дошка відносно недорога, що робить її доступною для любителів, студентів і професіоналів. Ці особливості роблять плату MSP430G2 привабливим варіантом для широкого спектру додатків, від простих систем моніторингу датчиків до більш складних систем керування та автоматизації.

MSP430G2553 (дивись рисунок 4.1) належить до мікроконтролерів, які працюють на шістнадцяти бітах і містять флеш-пам'ять для зберігання даних. З флеш-пам'яттю на цій платі є оперативна пам'ять, і на цій платі також створено центральний процесор із частотою 16 МГц. На цій платі є десятирозрядний аналого-цифровий перетворювач десяти каналів, а також він має ємнісні сенсорні вхід і вихід, які існують на платах. Для зв'язку з іншими типами контролерів і деякими зовнішніми пристроями він використовує універсальний послідовний інтерфейс зв'язку. На цій платі є набір Launchpad, що підтримується подвійним вбудованим пакетом, який містить майже двадцять розводок, що допомагає підключати різні типи пристроїв. Він забезпечує зв'язок компонентів початкового рівня з платою.



Рисунок 4.1 – Плата із МК MSP430G2553 [24]

У цьому пункті розділу розглянемо інтерфейс крокового двигуна (дивись рисунок 4.2) за допомогою MSP430. MSP-EXP430G2 – це інструмент розробки, також відомий як LaunchPad, наданий компанією Texas Instruments для вивчення та практики використання їхніх мікроконтролерів. Ця плата відноситься до категорії MSP430 Value Line, де можна програмувати всі мікроконтролери серії MSP430.

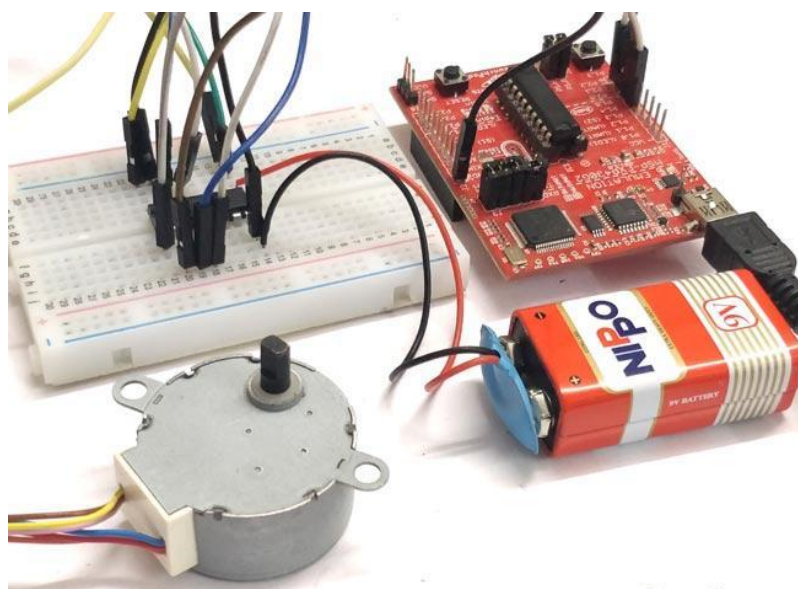


Рисунок 4.2 – Інтерфейс крокового двигуна з MSP430G2

Кроковий двигун (рисунок 4.3) – це тип безщіткового двигуна постійного струму, який перетворює електричні імпульси в чіткі механічні

рухи. Вал крокового двигуна обертається дискретними кроками. Можна отримати точні кроки та швидкість відповідно до наших потреб.



Рисунок 4.3 – Кроковий двигун

Будемо використовувати біполярний кроковий двигун 35BYJ46 (рисунок 4.4), який дешево доступний на ринку. Він має 6 проводів, але також поставляється з 5 проводами. У нашому кроковому двигуні є 2 котушки. З кожного виходить по 3 дроти. З 3 проводів 1 має відвод по центру, а решта 2 дроти з'єднані безпосередньо з котушкою. Загалом у нас є 4 сигнальні дроти та 2 центрованих дроти, які підключаються до джерела живлення 5-12 В.

У випадку, якщо від двигуна виходить 5 проводів, тоді 4 дроти є сигнальними, а 1 з'єднаний по центру з обома котушками.

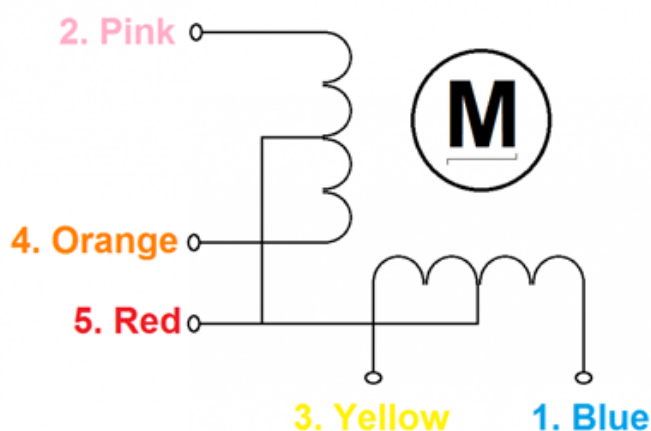


Рисунок 4.4 – Принципіальна схема біполярного крокового двигуна 35BYJ46

Щоб перевірити, який дріт має центральне відведення, а який є сигнальним, потрібно перевірити опір проводів, що виходять з двигуна. Таким чином, ті дроти, які з'єднані з тією ж котушкою, мають високе значення опору в порівнянні з опором центрованих відводів.

На рисунку 4.4, якщо перевірили значення опору синього та жовтого проводів і опір між ними більше, ніж значення між жовтим і червоним або синім і червоним. Отже, червоний – центрований провід з відводом.

Більшість крокових двигунів працюватимуть лише за допомогою драйверного модуля. Це пов'язано з тим, що модуль контролера (у нашому випадку MSP) не зможе забезпечити достатній струм від контактів введення/виведення для роботи двигуна. Тому будемо використовувати зовнішній модуль, наприклад ULN2003 модуль, як драйвер крокового двигуна. Існує багато типів модулів драйверів, і рейтинг одного змінюватиметься залежно від типу використовуваного двигуна. Основним принципом для всіх модулів драйверів буде джерело/споживання достатнього струму для роботи двигуна.

На наведеній схемі (рисунок 4.5) червоний дріт крокового кроку не підключений до PIN5 мікросхеми. Його потрібно підключити до 5 В. Кольоровий код вашого крокового двигуна може відрізнитися від кольорів, наведених на електричній схемі. Отже, підключіть дроти після перевірки правильних сигнальних проводів.

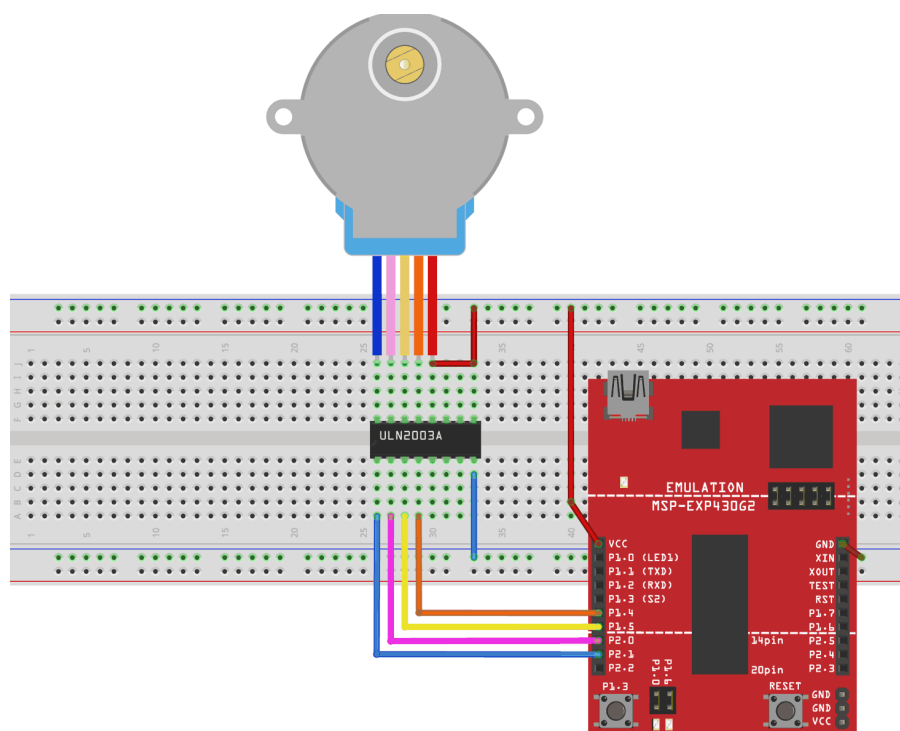


Рисунок 4.5 – Схема підключення та керування крокового двигуна за допомогою мікроконтролера MSP-EXP430G2

Напишемо код за допомогою Energia IDE. Він такий самий, як Arduino IDE, і простий у використанні. Зразок коду керування степером також можна знайти в прикладі меню Arduino IDE.

Перш ніж почнемо програмувати з нашим MSP430, давайте зрозуміємо, що насправді має відбуватися всередині програми. Будемо використовувати метод 4-етапної послідовності, тому у нас буде чотири кроки для виконання одного повного оберту. Розглянемо A, B, C і D як чотири котушки (таблиця 4.1).

Таблиця 4.1 – Метод чотирьох-етапної послідовності

Крок	Pin під напругою	Котушки під напругою
1	6 і 7	A і B
2	7 і 8	B і C
3	8 і 9	C і D
4	9 і 6	D і A

Напишемо код крокового двигуна. Перед написанням коду розберемо кілька важливих рядків.

Кількість кроків на оберт для нашого крокового двигуна була розрахована як 32; тому вводимо це, як показано в рядку нижче:

```
const int STEPS = 32.
```

Далі потрібно створити екземпляри, в яких ми вказуємо контакти, до яких ми підключили кроковий двигун:

```
Stepper myStepper (STEPS, 6, 7, 8, 9).
```

Оскільки використовуємо бібліотеку Stepper, ми можемо встановити швидкість двигуна за допомогою рядка нижче. Швидкість може коливатися від 0 до 200 для крокових двигунів 35BYJ46:

```
MyStepper.setSpeed(200).
```

Тепер, щоб змусити двигун рухатися на один крок, можемо використати наступний рядок:

```
myStepper.step(STEPS).
```

Оскільки маємо 32 кроки та 64 передавальне число, нам потрібно переміститися на 2048 ( $32 \cdot 64 = 2048$ ), щоб зробити один повний оберт. Тепер завантажте наведений нижче код і змініть номер кроків відповідно до ваших потреб. Тепер напишемо сам код:

```
#include <Stepper.h>
```

```

const int STEPS = 32; // змінити це відповідно до кількості кроків на
оберт
// для вашого двигуна
// ініціалізуйте степерну бібліотеку на контактах з 6 по 9 або ви можете
використовувати будь-які контакти на MSP430:
Stepper myStepper(STEPS, 6,7,8,9);
void setup() {
// встановіть швидкість 200 об/хв або як вам потрібно:
myStepper.setSpeed(200);
}
void loop() {
myStepper.step(STEPS);
}

```

Плата MSP-EXP430G2 має деякі переваги для проектів з кроковими двигунами:

а) низький рівень споживання енергії: крокові двигуни часто використовуються в застосунках, де важлива низька енергоспоживання. Плата MSP-EXP430G2 має низьке споживання енергії, що може бути корисним для довготривалих проектів або пристроїв, які працюють від батарей;

б) простота інтерфейсу: ця плата має простий інтерфейс, що дозволяє легко керувати кроковими двигунами. Вона може бути використана для створення контролера, який керує рухом крокового двигуна із зручним програмуванням;

в) розширювані можливості: MSP-EXP430G2 можна легко розширити, додавши додаткові модулі, сенсори чи інші пристрої, що може бути корисним у більш складних системах, які потребують більшого контролю або збирання даних.

У разі розробки більш складних речей, плата може стати основою для системи керування кроковими двигунами в більш великих пристроях або системах. Наприклад, вона може бути використана як частина промислового обладнання або пристрою автоматизації, де крокові двигуни використовуються для точного позиціонування.

Потенціал готового пристрою на основі цього проекту полягає в створенні компактного, низькоенергетичного пристрою з кроковим двигуном, що може знайти своє застосування в різноманітних областях, від робототехніки до автоматизованих систем виробництва чи у вбудованих системах для точного переміщення [28].

#### 4.2 Дослідження плати із мікроконтролером PIC16F877A Microchip Technology

Плата мікроконтролера Microchip Technology PIC16F877A (рисунок 4.2) – це популярна плата мікроконтролера, яка використовується в широкому діапазоні програм. Ця плата є продуктом компанії Microchip Technology Incorporated, провідного постачальника технології мікроконтролерів. Плата мікроконтролера PIC16F877A є членом сімейства мікроконтролерів PIC16 і має низку функцій, включаючи 14 КБ програмної пам'яті, 368 Б SRAM і 256 Б EEPROM [1]. Microchip надає безкоштовне інтегроване середовище розробки для плати мікроконтролера PIC16F877A, що робить його популярним вибором для розробників [27].

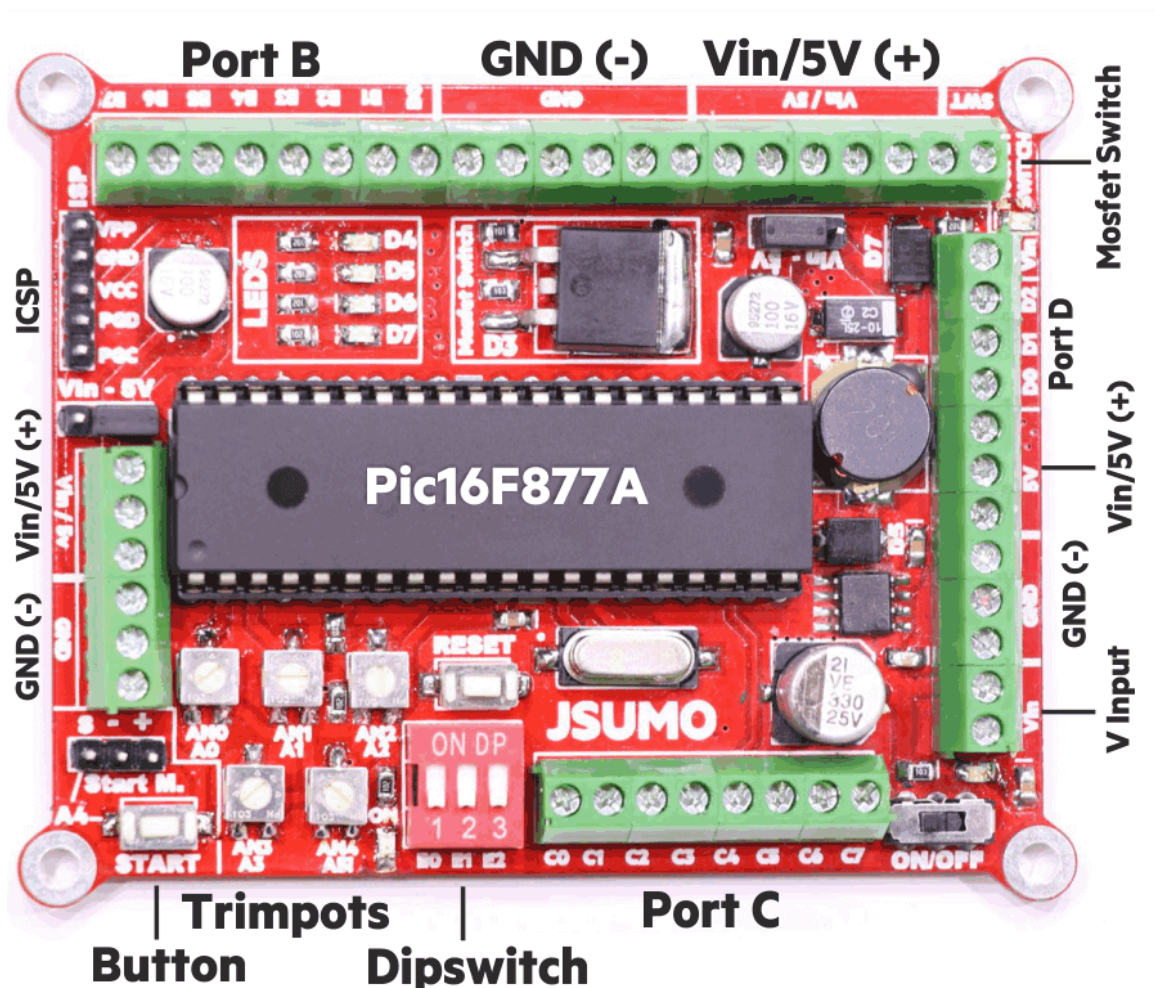


Рисунок 4.6 – Плата із МК PIC16F877A [25]

Поворотний енкодер – це пристрій введення, який допомагає користувачеві взаємодіяти з системою. Він більше схожий на радіопотенціометр, але він видає серію імпульсів, що робить його застосування унікальним. Коли ручка енкодера обертається, вона обертається у вигляді невеликих кроків, що допомагає використовувати її для керування кроковим/серводвигуном, навігації між пунктами меню, збільшення/зменшення значення числа та багато іншого.

У цьому пункті даного розділу дізнаємося про різні типи ротаційних енкодерів і про те, як вони працюють. Також будемо інтерфейсувати його з мікроконтролером PIC PIC16F877A та контролювати значення цілого числа шляхом обертання кодувальника та відображати його значення на РК-екрані 16\*2.

Поворотний енкодер часто називають валовим енкодером. Це електромеханічний перетворювач, тобто він перетворює механічні рухи в електронні імпульси або, іншими словами, перетворює кутові положення, рух або положення вала в цифровий або аналоговий сигнал. Він складається з ручки, яка при обертанні буде рухатися крок за кроком і створювати послідовність імпульсів із заздалегідь визначеною шириною для кожного кроку.

На ринку існує багато типів поворотних датчиків, дизайнер може вибрати один відповідно до своєї програми. Найпоширеніші типи перераховані нижче:

- інкрементальний кодер;
- абсолютний кодер;
- магнітний кодер;
- оптичний кодер;
- лазерний кодер.

Ці кодери класифікуються на основі вихідного сигналу та технології вимірювання, інкрементний кодер і абсолютні кодери класифікуються на основі вихідного сигналу, а магнітні, оптичні та лазерні кодери класифікуються на основі технології вимірювання. Кодер, який тут використовується, є кодувальником інкрементального типу.

Абсолютний кодер зберігає інформацію про положення навіть після відключення живлення, і інформація про положення буде доступною, коли знову подамо до нього живлення.

Інший базовий тип, інкрементний кодер, надає дані, коли кодер змінює своє положення. Не вдалося зберегти інформацію про позицію.

Схема контактів інкрементального енкодера КУ-040 показана нижче на рисунку 4.7. У цьому проекті ми будемо інтерфейсувати цей Rotary Encoder з популярним мікроконтролером PIC16F877A від microchip.



Рисунок 4.7 – Схема контактів інкрементального енкодера KY-040

Перші два контакти (заземлення та Vcc) використовуються для живлення кодера, зазвичай використовується джерело живлення +5 В. Окрім обертання ручки за годинниковою стрілкою та проти годинникової стрілки, енкодер також має перемикач (Active low), який можна натиснути, натиснувши ручку всередині. Сигнал від цього перемикача надходить через контакт SW. Нарешті, він має два вихідних контакти (DT і CLK), які створюють сигнали.

Вихід повністю залежить від внутрішніх мідних контактів, які забезпечують з'єднання GND і VCC з валом.

Є дві частини Rotary Encoder. Колесо вала, яке з'єднане з валом і обертається за або проти годинникової стрілки залежно від обертання вала, а також основа, де виконується електричне з'єднання. Основа має порти або точки, які з'єднані з DT або CLK таким чином, що, коли колесо валу обертається, воно з'єднує базові точки та створює прямокутну хвилю як на порту DT, так і на порту CLK.

Два порти забезпечують меандр, але є невелика різниця в синхронізації. Через це, якщо ми приймаємо вихідні дані як 1 і 0, може бути лише чотири

стани, 0 0, 1 0, 1 1, 0 1. Послідовність двійкового виведення визначає поворот за годинниковою стрілкою або проти годинникової стрілки. Наприклад, якщо поворотний енкодер надає 1 0 у стані неактивності та надає 1 1 після цього, це означає, що енкодер змінює своє положення на один крок у напрямку за годинниковою стрілкою, але якщо він надає 0 0 після неактивного 1 0, означає, що вал змінює своє положення проти годинникової стрілки за один крок (рисунок 4.8).

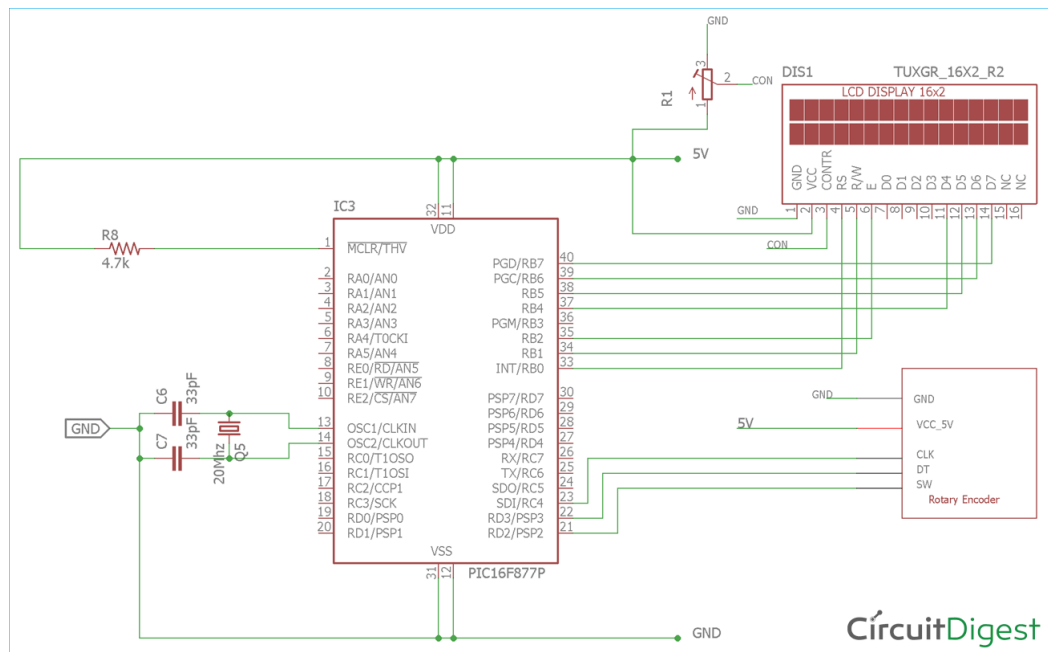


Рисунок 4.8 – Принципова схема інтерфейсу поворотного кодера PIC16F877A

Нижче наведено зображення остаточного налаштування після підключення компонентів відповідно до принципової схеми (рисунок 4.9):

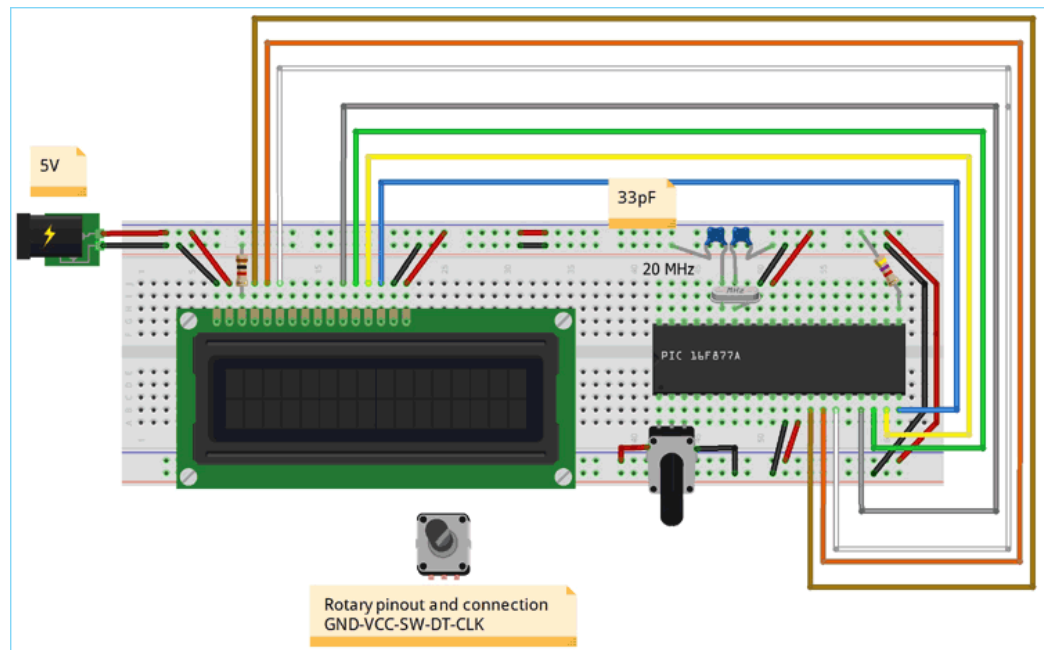


Рисунок 4.9 – Налаштування поворотного кодера

Повний код PIC наведено в кінці цього проекту з демонстраційним відео , тут ми пояснюємо кілька важливих частин коду. Якщо ви новачок у роботі з мікроконтролером PIC, дотримуйтеся наших посібників з PIC з самого початку.

Як ми обговорювали раніше, нам потрібно перевірити вихід і відрізнити двійковий вихід для DT і CLK , тому ми створили частину if-else для операції.

```

if (Encoder_CLK != позиція){
if (Encoder_DT != позиція){
// lcd_com (0x01);
лічильник++; // Збільшити лічильник, який буде надруковано на lcd
lcd_com (0xC0);
lcd_puts(" ");
lcd_com (0xC0);
lcd_bcd(1,лічильник);
}
}

```

```

else{
// lcd_com (0x01);
lcd_com (0xC0);
лічильник--; // зменшити лічильник
lcd_puts(" ");
lcd_com (0xC0);
lcd_bcd(1,лічильник);
//lcd_puts("Ліворуч");
}
}

```

Нам також потрібно зберегти позицію на кожному кроці . Для цього ми використали змінну «position» , яка зберігає поточну позицію.

position = Encoder\_CLK; // Зберігає позицію годинника кодера у змінній. Може бути 0 або 1.

Крім цього, передбачена опція сповіщення про натискання перемикача на РК-дисплеї.

```

if (Encoder_SW == 0){
sw_delayms(20); //затримка усунення дребезгу
if (Encoder_SW == 0){
//lcd_com(1);
//lcd_com(0xC0);
lcd_puts ("перемикач натиснутий");
// itoa(лічильник, значення, 10);
// lcd_puts(значення);

```

Функція `system_init` . використовується для ініціалізації операції вводу-виводу, РК-дисплея та для збереження положення поворотного енкодера:

```
void system_init(){
    TRISB = 0x00; // ПОРТ В як вихід, цей порт використовується для
LCD
    TRISDbits.TRISD2 = 1;
    TRISDbits.TRISD3 = 1;
    TRISCbits.TRISC4 = 1;
    lcd_init(); // Це призведе до ініціалізації позиції LCD
    = Encoder_CLK;// Сортування позиції CLK під час ініціалізації системи
перед початком циклу while.
}
```

Повний код наведено нижче та пояснення коментарями:

```
/*
 * Configuration Related settings. Specific for microcontroller unit.
 */
#pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)
#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = ON // Brown-out Reset Enable bit (BOR enabled)
#pragma config LVP = OFF // Low-Voltage (Single-Supply) In-Circuit Serial
Programming Enable bit (RB3/PGM pin has PGM function; low-voltage
programming enabled)
#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit
(Data EEPROM code protection off)
```

```
#pragma config WRT = OFF    // Flash Program Memory Write Enable bits
                             (Write protection off; all program memory may be written to by EECON control)

#pragma config CP = OFF     // Flash Program Memory Code Protection bit
                             (Code protection off)

#define _XTAL_FREQ 2000000

/*

 * System Header files inclusions

 */

#include <xc.h>

// #include <string.h>

#include <stdlib.h>

#include "supporting c files/lcd.h"

#define Encoder_SW PORTDbits.RD2

#define Encoder_DT PORTDbits.RD3

#define Encoder_CLK PORTCbits.RC4

/*

 * Program flow related functions

 */

int counter; // It will hold the count of rotary encoder.

int position; // It will store the rotary encoder position.
```

```
void sw_delayms(unsigned int d);
```

```
int value[7];
```

```
/*
```

```
 * System Init Function
```

```
*/
```

```
void system_init ();
```

```
/* Main function single Thread*/
```

```
void main(void) {
```

```
    system_init();
```

```
    lcd_puts ("Circuit Digest");
```

```
    lcd_com(0xC0);
```

```
    counter = 0;
```

```
    while(1){
```

```
        lcd_com(0xC0);
```

```
        if (Encoder_SW == 0){
```

```
            sw_delayms(20);
```

```
            if (Encoder_SW == 0){
```

```
                //lcd_com(1);
```

```
                //lcd_com(0xC0);
```

```
        lcd_puts ("switch pressed");  
//        itoa(counter, value, 10);  
//        lcd_puts(value);  
    }  
}
```

```
if (Encoder_CLK != position){  
    if (Encoder_DT != position){  
        // lcd_com (0x01);  
        counter++;  
        lcd_com (0xC0);  
        lcd_puts("      ");  
        lcd_com (0xC0);  
        lcd_bcd(1,counter);  
    }  
    else{  
        // lcd_com (0x01);  
        lcd_com (0xC0);  
        counter--;  
        lcd_puts("      ");  
        lcd_com (0xC0);  
        lcd_bcd(1,counter);  
        //lcd_puts("Left");  
    }  
}
```

```
    }  
    }  
    position = Encoder_CLK;  
}
```

```
    return;  
}
```

```
void sw_delayms(unsigned int d){  
    int x, y;  
    for(x=0;x<d;x++)  
        for(y=0;y<=1275;y++);  
}
```

```
void system_init(){  
    TRISB = 0x00; // PORT B as output, This port is used for LCD  
    TRISDbits.TRISD2 = 1;  
    TRISDbits.TRISD3 = 1;  
    TRISCbits.TRISC4 = 1;  
    lcd_init(); // This will Initialize the LCD  
    position = Encoder_CLK; // Set the CLK position on system init, before the  
    while loop start.  
}
```

Плата PIC16F877A є чудовим вибором для проектів поворотного кодера з кількома причинами:

а) інтегрованість та простота використання:

1) плата має вбудований аналогово-цифровий конвертер (ADC)

2) можливість обробки аналогових сигналів, що важливо для зчитування поворотних кодерів.

3) достатньо портів введення/виведення для обробки сигналів від кодерів;

б) низькі вимоги щодо електроживлення та вартості:

1) PIC16F877A споживає мало енергії, що робить його відмінним вибором для пристроїв, які працюють від батарей або живляться від обмеженого джерела енергії.

2) доступність за ціною, і за доступністю на ринку;

в) простота програмування:

1) PIC16F877A можна програмувати за допомогою різних IDE та мов програмування, включаючи мову асемблера та C. Це дозволяє вам вибрати оптимальний метод розробки в залежності від досвіду та вимог проекту.

У розробці більш складних систем, плата PIC16F877A може використовуватися як частина комплексних систем керування, які включають багато поворотних кодерів. Наприклад, вона може бути використана в системах автоматизованого управління, де потрібно багато каналів введення для відстеження різних параметрів або позицій.

Готовий пристрій на основі цього проекту може бути, наприклад, контролером для системи позиціонування, яка використовує поворотні кодери для точного визначення положення рухомого об'єкта. Це може бути застосовано у різних областях, таких як автоматизовані системи виробництва, робототехніка або точні вимірювальні пристрої [29].

### 4.3 Дослідження плати із мікроконтролером TM32F103C6T6 ST Microelectronics

ST Microelectronics є світовим лідером у галузі напівпровідників, що надає широкий спектр інноваційних рішень для електроніки та вбудованих систем. Продукція компанії використовується в різноманітних сферах застосування: від автомобільної та промислової до побутової електроніки та інтелектуальних пристроїв. Однією з ключових пропозицій від ST Microelectronics є плата мікроконтролера TM32F103C6T6, яка забезпечує потужну та гнучку платформу для розробки вбудованих систем і програм.

Плата мікроконтролера TM32F103C6T6 (рисунок 4.10) створена на основі процесора ARM Cortex-M3, що забезпечує високу продуктивність і низьке енергоспоживання. Плата має широкий спектр периферійних пристроїв та інтерфейсів, включаючи USB, CAN, SPI, I2C і UART, що робить її придатною для різноманітних застосувань. Плата також містить ряд вбудованих функцій, таких як 8-канальний 12-розрядний АЦП, 12-розрядний ЦАП і годинник реального часу, які можна використовувати для реалізації широкого діапазону функцій [26].

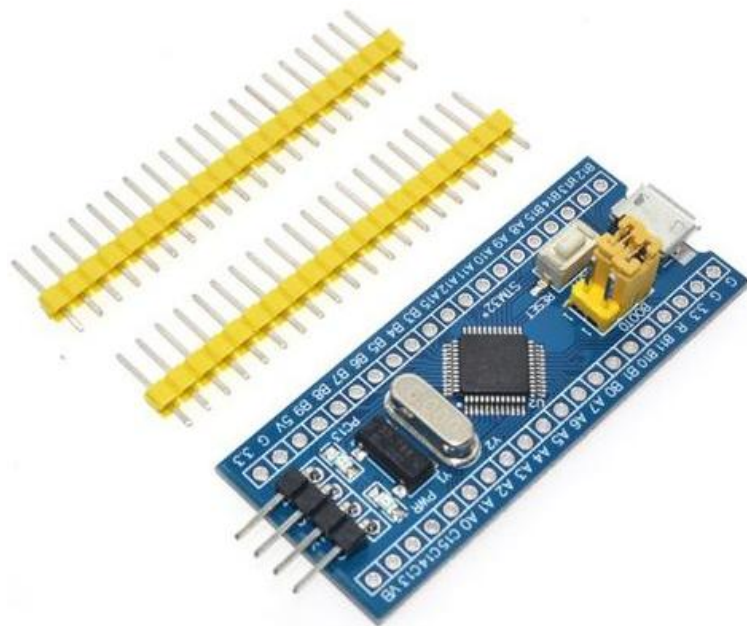


Рисунок 4.10 – Плата із МК TM32F103C6T6 [27]

В електроніці серводвигуни в основному використовуються в проектах робототехніки через їх точність і легкість у використанні. Серводвигуни меншого розміру, вони дуже ефективні та енергоефективні. Вони забезпечують високий крутний момент і можуть використовуватися для підйому або штовхання тягарів відповідно до специфікації двигунів. У цьому пункті розділу дізнаємося про серводвигун і як інтерфейс сервоприводу з платою STM32F103C8 (рисунок 4.11). Потенціометр також підключений для зміни положення вала серводвигуна та РК-дисплей для відображення значення кута.

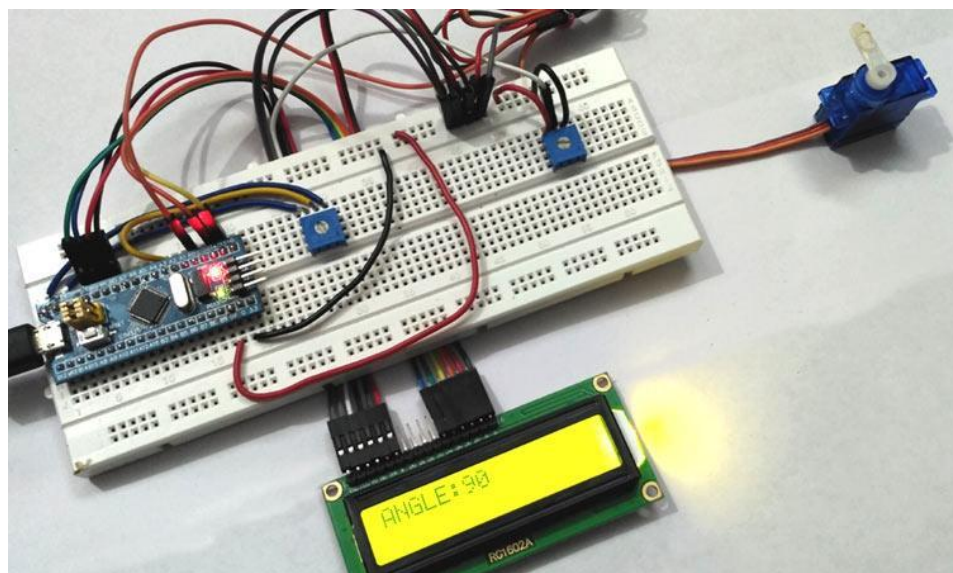


Рисунок 4.11 – Інтерфейс серводвигуна з STM32F103C8

У STM32F103C8 маємо 10 контактів АЦП (PA0-PB1), а тут використовуємо лише один контакт (PA3) для `analogread()` для встановлення положення вала двигуна за допомогою потенціометра. Крім того, серед 15 штифтів ШІМ STM32 (PA0, PA1, PA2, PA3, PA6, PA7, PA8, PA9, PA10, PB0, PB1, PB6, PB7, PB8, PB9) один вихід буде використовуватися для подачі імпульсів на сервопривід. штифт ШІМ двигуна (зазвичай він помаранчевого кольору), рисунок 4.12.

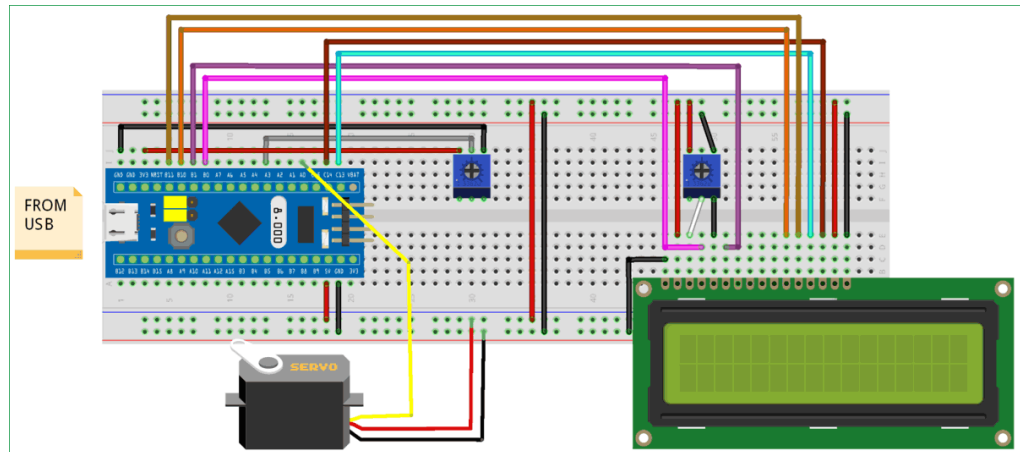


Рисунок 4.12 – Принципова схема та підключення МК SMT32F103C8

Використовуємо два потенціометри (дивись рисунок 4.7):

Потенціометр праворуч використовується для зміни контрастності РК-дисплея. Він має три контакти, лівий контакт для +5 В, а правий для GND, а центральний контакт підключений до V0 РК-дисплея.

Потенціометр ліворуч використовується для зміни положення вала серводвигуна шляхом керування аналоговою вхідною напругою, лівий штифт має вхід 3,3 В, а правий має GND, а центральний вихід підключений до (PA3) STM32.

Запрограмували STM32F103C8 за допомогою Arduino IDE через порт USB без використання програматора FTDI. Ми можемо продовжити програмування, як Arduino. Повний код наведено нижче в кінці проекту.

Спочатку включили бібліотечні файли для функцій сервоприводу та LCD:

```
#include<Servo.h>
#include<LiquidCrystal.h>
```

Потім оголосив контакти для РК-дисплея та ініціалізував його. Також оголошено кілька інших змінних для ШІМ і потенціометра:

```

const int rs = PB0, en = PB1, d4 = PB10 , d5 = PB11 , d6 = PC13, d7 =
PC14;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
int servoPin = PA0;
int potPin = PA3.

```

Тут створили змінний сервопривід із типом даних Servo та приєднали його до попередньо оголошеного ШІМ-контакту.

```

Servo servo;
servo.attach(servoPin).

```

Потім прочитали аналогове значення з контакту PA3, оскільки це контакт АЦП, він перетворює аналогову напругу (0-3,3) у цифрову форму (0-4095).

```

analogRead(potPin).

```

Оскільки цифровий вихід має 12-бітну роздільну здатність, нам потрібно отримати значення в діапазоні градусів (0-170), він ділить значення АЦП (0-4096) відповідно до максимального кута 170 градусів, тому ми ділимо на 24.

```

angle = (reading/24).

```

Наведена нижче функція змушує серводвигун обертати вал на заданий кут.

```

servo.write(angle).

```

Повний код наведено нижче та добре пояснено коментарями.

```
//ІНТЕРФЕЙС СЕРВО З STM32
//ДАЙДЖЕСТ
#include<Servo.h> //включаючи бібліотеку сервосистем
#include<LiquidCrystal.h> //включаючи бібліотеку РК-дисплеїв
const int rs = PB0, en = PB1, d4 = PB10, d5 = PB11, d6 = PC13, d7 = PC14;
//оголошення імен і номерів пінів lcd
LiquidCrystal lcd (rs, en, d4, d5, d6, d7); //налаштування РК та його
параметрів
int servoPin = PA0; //оголосити та ініціалізувати контакт для виходу
ШИМ сервоприводу
int potPin = PA3; //потенціометр вхід АЦП
Servo servo; // створення сервоприводу змінної з типом даних Servo
void setup()
{
  lcd.begin(16,2); //налаштування РК-дисплея як 16x2
  lcd.setCursor(0,0); // встановлення курсору на перший рядок і перший
стовпець
  lcd.print("CIRCUIT DIGEST"); //поміщає CIRCUIT DIGEST на LCD
  lcd.setCursor(0,1); // встановлення курсору у другому рядку та першому
стовпці
  lcd.print("SERVO WITH STM32"); //поміщає SERVO WITH STM32 на
LCD
  delay(3000); // затримка на 3 секунди
  lcd.clear(); // очищає РК-дисплей
  servo.attach(servoPin); //він з'єднує висновок PA0 з двигуном як
зворотний зв'язок управління шляхом надання імпульсів
}
void loop()
```

```

{
  lcd.clear(); // очищає РК-дисплей
  int angle;; //оголосити змінний кут як int
  int reading; //оголосити читання змінної як int
  reading = analogRead(potPin); //читання аналогового значення з виводу
  PA3
  angle = (reading/24); //він ділить значення ADC відповідно до
максимального кута 170 градусів
  servo.write(angle); //поміщає значення кута в серво
  lcd.setCursor(0,0); // встановлення курсору на перший рядок і перший
стовпець
  lcd.print("ANGLE:"); //поміщає ANGLE в LCD
  lcd.print(angle); //розміщує значення під кутом
  delay(100); //затримка в часі
}

```

Плата STM32F103C8 має кілька ключових переваг для проектів з серводвигунами:

а) ARM Cortex-M3 процесор з достатньою продуктивністю та потужністю для керування серводвигунами. Це дозволяє вам здійснювати точне управління і регулювання руху серводвигунів.

б) Багатофункціональність:

1) мікроконтролер має різноманітні периферійні пристрої, такі як таймери, пристрої зберігання даних та різні інтерфейси, що дозволяє використовувати їх для точного управління серводвигунами та обробки зворотного зв'язку.

в) Широкий спектр підтримки:

1) STM32 має велику спільноту розробників, багато документації та різні

програмні засоби, що полегшують розробку. Це робить його привабливим вибором для проектів, де потрібна підтримка та можливості розширення.

У розробці більш складних систем, STM32F103C8 може використовуватися як основа для управління не лише одним, але й багатьма серводвигунами. Він може бути частиною складних мехатронних систем, роботів або автоматизованих ліній виробництва, де потрібне точне керування рухом.

Потенціал готового пристрою на основі даного проекту може бути в розробці пристрою з точним позиціонуванням, такого як CNC-машини, 3D-принтери або робототехніка. Завдяки потужності цього мікроконтролера та його можливостям управління рухом, можна створити надійний та точний пристрій для вирізання, друку чи руху в роботах, які потребують точного позиціонування [30].

## ВИСНОВКИ

В даній магістерській роботі було проведено аналіз мікроконтролерів:

- MSP430G2, PIC16F877A та TM32F103C6T6 були предметом ретельного дослідження;

- проведено порівняльний аналіз їхніх характеристик, включаючи архітектуру, швидкодію, енергоефективність та можливості периферійних пристроїв.

Оцінка плат на основі мікроконтролерів:

- проведено аналіз плат, що базуються на кожному з мікроконтролерів, з урахуванням їхньої ефективності, можливостей розширення та придатності для конкретних застосувань.

а) MSP430G2:

- переваги: низьке енергоспоживання, висока енергоефективність;  
- недоліки: обмежені можливості для складних завдань, менший набір периферійних пристроїв.

б) PIC16F877A:

- переваги: різноманітність периферійних пристроїв, зручний для початківців;  
- недоліки: обмежена продуктивність та можливості для вимогливих застосувань.

в) TM32F103C6T6:

- переваги: висока продуктивність, розширені можливості, багато периферійних пристроїв;  
- недоліки: збільшене енергоспоживання порівняно з іншими.

Серед аналогічних програм обрано PROTEUS PROFESSIONAL через його ряд переваг:

- багатий функціонал: забезпечує широкий спектр можливостей для симуляційних проектів;

- широка підтримка: має велику бібліотеку моделей електронних компонентів та мікроконтролерів;

- інтуїтивний інтерфейс: забезпечує зручність використання для користувачів будь-якого рівня.

Характеристикою Proteus Professional є потужний пакет програмного забезпечення для симуляції електронних систем. Він має інтуїтивний інтерфейс та широкий спектр функціоналу для розробки, тестування та відлагодження електронних пристроїв і систем. Його бібліотека моделей включає широкий вибір мікроконтролерів та електронних компонентів, що робить його потужним інструментом для імітації та верифікації роботи електронних схем.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до лабораторних робіт з дисципліни "Мікропроцесорні системи контролю та керування" для студентів усіх форм навчання спеціальності 171 - Електроніка, освітньо-професійна програма "Електронні пристрої та системи" / упоряд.: І. М. Бондаренко, О. В. Бородін, В. П. Карнаушенко ; М-во освіти і науки України, ХНУРЕ. – Харків : ХНУРЕ, 2019.
2. Jimenez, D., Parrado, A. Overview of a framework for Implementation of digital controllers in Energia IDE using Texas Instruments microcontrollers.
3. Fujdiak, R., Misurec, J., Mlynek, P. Analysis of random number generator from texas instrument in MSP430 x5xx families.
4. Wouters, L., Gierlichs, B., Preneel, B. On the susceptibility of Texas Instruments SimpleLink platform microcontrollers to non-invasive physical attacks.
5. Mahesh, M., Sivraj, P. DrawCode: Visual tool for programming microcontrollers.
6. Elgandy, M. Photovoltaic pumping systems with microcontroller-based MPPT control.
7. Barrett, S., Pack, D. Microcontroller Programming and Interfacing Texas Instruments MSP430: Part II.
8. Sarma, U., Boruah, P. Design and development of a high precision thermocouple based smart industrial thermometer with on line linearisation and data logging feature.
9. Loss, P., Lamego, M., Sousa, G. A single phase microcontroller based energy meter.
10. Hamrita, T., Hoffacker, E. Development of a “smart” wireless soil monitoring sensor prototype using RFID technology.
11. Abrial, A., Bouvier, J., Renaudin, M. A new contactless smart card IC using an on-chip antenna and an asynchronous microcontroller.

12. Van Herrewege, A., van der Leest, V., Schaller, A. Secure PRNG seeding on commercial off-the-shelf microcontrollers.
13. Shao, J., Nolan, D., Teissier, M. A novel microcontroller-based sensorless brushless DC (BLDC) motor drive for automotive fuel pumps.
14. Zappa, F. Microcontrollers. Hardware and Firmware for 8-bit and 32-bit devices.
15. Li, W., Shi, J., Li, F., Lin, J., Wang, W., Guan, L.  $\mu$ AFL: non-intrusive feedback-driven fuzzing for microcontroller firmware.
16. De Vita, F., Nocera, G., Bruneo, D. On-device training of deep learning models on edge microcontrollers.
17. Joly, Y., Lopez, L., Portal, J., Aziza, H. Impact of hump effect on MOSFET mismatch in the sub-threshold area for low power analog applications.
18. Kulaksız, A., Gökkuş, G., Alhajomar, F. Rapid control prototyping based on 32-Bit ARM Cortex-M3 microcontroller for photovoltaic MPPT algorithms.
19. Billè, S. Two or three shunt resistor based current sensing circuit design in 3-phase inverters.
20. Бондаренко І.М. Електронні системи: навчальний посібник /І.М. Бондаренко, О.В. Глухов, О.О, Кравчук - Харків, ХНУРЕ, 2019, - 219 с.
21. <https://www.ti.com/lit/ds/symlink/msp430g2253.pdf>.
22. <https://microcontrollerslab.com/pic16f877a-introduction-features/>.
23. <https://www.quarktwin.com/blogs/integrated%20circuit/unraveling-the-world-of-stm32f103c6t6-a-comprehensive-guide?id=293>.
24. <https://www.ti.com/product/MSP430G2553>.
25. <https://www.jsumo.com/propic40-ultimate-pic16f877a-mainboard>.
26. <https://www.watelectronics.com/pic16f877a-microcontroller/>.
27. <https://nl.mouser.com/new/stmicroelectronics/stm32/>.
28. <https://circuitdigest.com/microcontroller-projects/interfacing-stepper-motor-with-msp430>.
29. <https://circuitdigest.com/microcontroller-projects/pic16f877a-rotary-encoder-interfacing>.

30. <https://circuitdigest.com/microcontroller-projects/servo-motor-interfacing-with-stm32f103c8-stm32-development-board>.