

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Сумський державний університет**

Факультет електроніки та інформаційних технологій

Кафедра електроніки, загальної та прикладної фізики

«До захисту допущено»  
Завідувач кафедри

\_\_\_\_\_ Іван ПРОЦЕНКО  
\_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня бакалавр**

зі спеціальності 171 – Електроніка

освітньо-наукової програми «Електронні інформаційні системи»

на тему: **МІКРОКОНТРОЛЕРНІ ПЛАТФОРМИ STM ДЛЯ АВТОМАТИЗАЦІЇ  
ФІЗИЧНОГО ЕКСПЕРИМЕНТУ**

Здобувача групи ЕП-91 Ткаченко Юлія Романівна

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Юлія ТКАЧЕНКО

Керівник,

канд. фіз.-мат. наук, асистент кафедри ЕЗПФ

Андрій ЛОГВИНОВ

**Суми – 2023**

## ЗМІСТ

<b>РОЗДІЛ 1. STM МІКРОКОНТРОЛЕРИ.....</b>	<b>6</b>
1.1. Загальні відомості про мікроконтролери.....	6
1.2. Історія створення мікроконтролерів STM .....	10
1.3. Огляд мікроконтролерних платформ STM і їх використання.....	12
<b>РОЗДІЛ 2. ОПИС ХАРАКТЕРИСТИК STM</b>	
<b>КОНТРОЛЕРІВ.....</b>	<b>16</b>
2.1. Технологія STM та його особливості.....	16
2.2. Характеристики STM мікроконтролерів.....	18
2.3. Роль мікроконтролерів у фізичних експериментах .....	21
<b>РОЗДІЛ 3. СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ</b>	<b>25</b>
<b>ВИМІРЮВАННЯ ТЕНЗОЕФЕКТУ.....</b>	
3.1. Методика вимірювання тензоефекту.....	25
3.2. Написання коду під систему вимірювання тензоефекту для STM32....	27
<b>ВИСНОВКИ.....</b>	<b>31</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>32</b>

## СКОРОЧЕННЯ І УМОВНІ ПОЗНАЧЕННЯ

ЦП – Центральний процесор.

ШІМ – Широтно-імпульсної модуляції.

## АНОТАЦІЯ

Кваліфікаційна робота викладена на 33 сторінках, зокрема, містить 18 рисунків, 1 таблицю, список використаних джерел із 18 найменувань.

Актуальність роботи полягає у тому, що мікроконтролерні платформи STM є широко використовуваними в промисловості та наукових дослідженнях. Висока швидкодія та простота використання дозволяють використовувати їх для автоматизації різних експериментів, включаючи фізичні. На сьогоднішній день зростає кількість наукових досліджень та розробок, які пов'язані з фізикою, призводить до зростання потреби в швидкій та ефективній обробці даних.

Мета кваліфікаційної роботи бакалавра полягає у вивченні теми мікроконтролерів, характеристик та можливості використання технології STM для збору та обробки даних в реальному часі з різних датчиків та інших пристроїв. Також розгляд розробки програмного забезпечення для мікроконтролерних платформ STM з використанням мов програмування, таких як C++. Тестування розробленого програмного забезпечення у проведенні фізичних експериментів.

Під час виконання роботи була використана методика визначення тензоефекту і програмування мікроконтролерів на основі ядра ARM.

У результаті кваліфікаційної роботи було вивчено загальні відомості мікроконтролерів та проведено огляд мікроконтролерних платформ. Продемонстровано принципи дії покладені у роботу мікроконтролерів. Розглянуто характеристики STM контролерів, технологія використання такого типу мікроконтролерів. Характеристики STM32 показали, що дані мікроконтролери мають високу продуктивність, низьку вартість та енергоспоживання. Розроблена система автоматичного виміру тензоефекту на основі мікроконтролера STM32.

Рекомендації щодо використання: розроблена система автоматичного визначення і виміру параметрів для розрахунку тензоефекту може бути використана у лабораторних і навчальних цілях.

**Ключові слова:** АВТОМАТИЗОВАНА СИСТЕМА ПРОВЕДЕННЯ ФІЗИЧНОГО ЕКСПЕРИМЕНТУ, МІКРОКОНТРОЛЕР, ТЕНЗОЕФЕКТ, STM32.

## РОЗДІЛ 1

### STM МІКРОКОНТРОЛЕРИ

#### 1.1. Загальні відомості про мікроконтролери

Мікроконтролер – це програмований цифровий процесор з необхідною периферією. Це складні послідовні цифрові схеми, що призначені для виконання певної роботи до програми. Мікроконтролер у певному сенсі є щось середнє між мікропроцесором і мікрокомп'ютером. Як і мікропроцесори, термін мікроконтролер відноситься до окремого пристрою, однак він містить увесь мікрокомп'ютер на одному чіпі. Тому мікроконтролер матиме процесор, вбудовану пам'ять, а також різноманітність пристроїв вводу-виводу. На рисунку 1.1 показана схема контролера. Всі компоненти з'єднані через внутрішній чіп та об'єднані в єдину мікросхему. Модулі підключаються через контакти вводу-виводу [1].

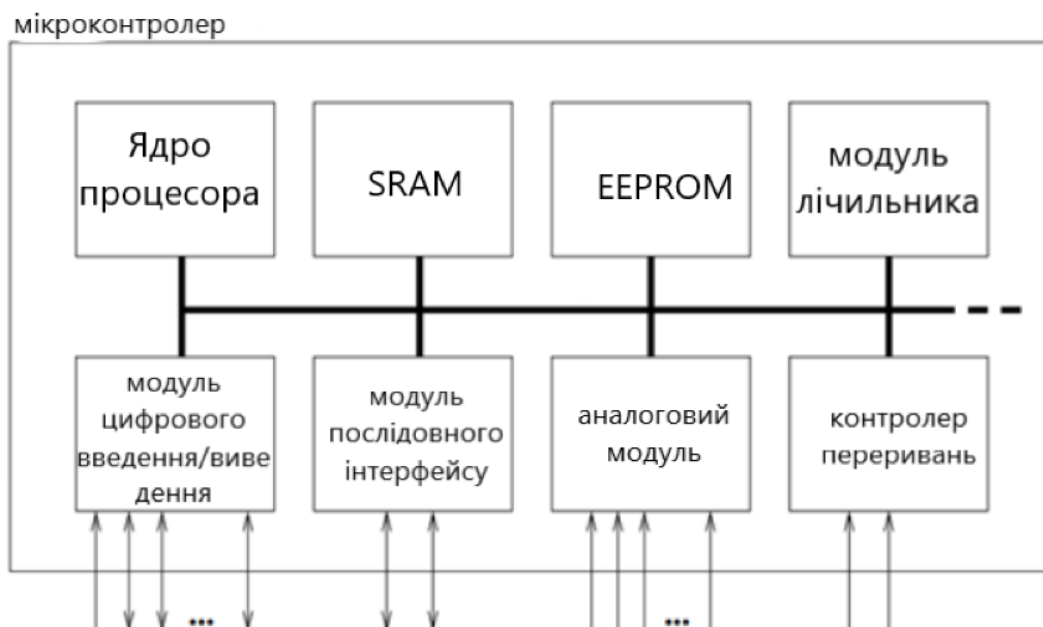


Рисунок 1.1 – Принципова схема мікроконтролера [2]

Ядро процесора – це основна частина мікроконтролера. Базова архітектура центрального процесора зображена на малюнку 1.2. Він складається з шляху даних,

який виконує інструкції, і блоку керування, який в основному повідомляє шляху даних, що робити.

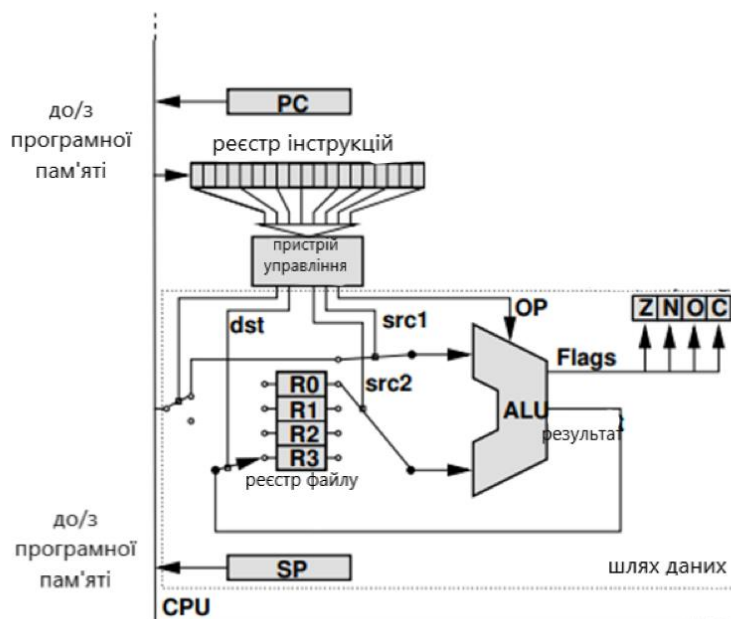


Рисунок 1.2 – Архітектура ядра процесора [2]

Пам'ять використовується для зберігання програмного коду та даних. Зазвичай мікроконтролер має вбудовану програмну пам'ять (ROM або FLASH), а також оперативну пам'ять (RAM). Існують два види моделі пам'яті, їх відрізняють по тому чи є пам'ять програми та даних інтегрованою або окремою [3]. Архітектура фон Неймана, названа на честь вченого, який долучився до Манхеттенського проекту та, через обчислювальні вимоги цього проекту, долучився до розробки комп'ютера із збереженою програмою EDVAC [4]. У цій архітектурі програма та дані зберігаються разом і доступ до них здійснюється через один чіп. На жаль, це означає, що доступ до програм і даних може конфліктувати. Необхідність обміну даними з зовнішньою по відношенню до ЦП пам'яттю призводить до обмеження пропускної здатності, відомого як вузьке місце фон Неймана [5]. Це вузьке місце є особливо серйозним у цій архітектурі, порівняно з іншими, до блоку керування та арифметично-логічного пристрою, що потребують читання та запису в пам'ять, отже, спільне використання обмеженого ресурсу в

системі (час доступу до пам'яті). Гарвардська архітектура – архітектура, що передбачає розділення програм та даних у окремих частинах пам'яті, які доступні через окремі чіпи. Це дозволяє уникнути конфліктів при одночасному доступі до коду та даних, що позитивно впливає на продуктивність системи. Недоліком такої архітектури є необхідність використання більшого обсягу апаратного забезпечення, так як потрібні два чіпи та або дві мікросхеми пам'яті, або двопортова пам'ять, що забезпечує два незалежних доступи одночасно.

Функції цифрового введення-виведення є одними з головних функцій мікроконтролерів. Кількість контактів введення/виведення коливається від 3-4 до понад 90, залежно від сімейства контролера та типу контролера (рис. 1.3).

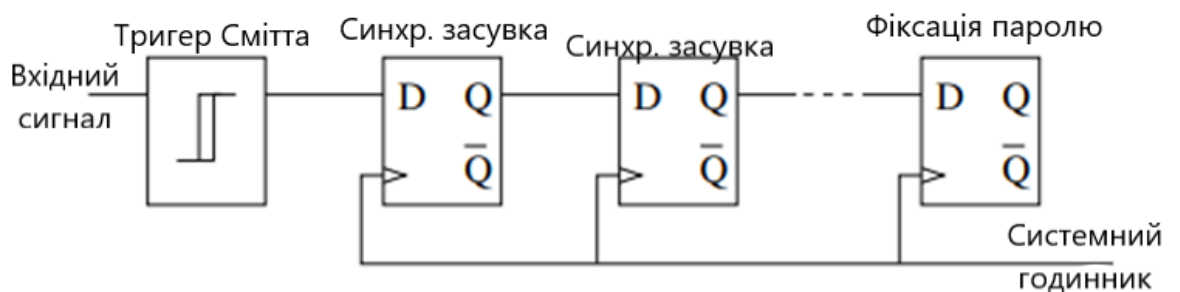


Рисунок 1.3 – Структурна схема принципової входної схеми контролера [2]

Schmitt-тригер – це елемент, який дозволяє підсилювати та стабілізувати вхідний сигнал, що дозволяє уникнути спотворення сигналу від шумів та інших перешкод. Він має два порогових значення: верхній та нижній. Якщо вхідний сигнал перевищує верхній поріг, він вважається логічним 1, а якщо він менше, ніж нижній поріг, то він вважається логічним 0. Synchr. Latch – це елемент, який дозволяє зберігати вхідний сигнал на момент вхідного імпульсу синхронізації. Він має вхідний сигнал, який потрапляє до нього через Schmitt-тригер, та вхідний імпульс синхронізації, який визначає момент захоплення вхідного сигналу. Pin Latch – це елемент, який дозволяє зберігати вхідний сигнал на деякий час. Його вхідний сигнал є вихідним сигналом Synchr. Latch, а вихідний сигнал Pin Latch є вхідним сигналом для додаткових блоків контролера [2].

Контроль переривання – це повідомлення процесору про подію, що сталася та потребує уваги. Переривання може викликано подіями, включаючи вхідні повідомлення, помилки в апаратному забезпеченні сповіщення про те, що завдання виконано, зміни значень вхідних контактів або таймерів, тощо. З популярних повідомлень про переривання, це переривання, яке виникає на комп'ютері при натисканні клавіші на клавіатурі. У більшості випадків натискання клавіш буферизуються та відображаються пізніше, без перерви, якщо комп'ютер активно не звертав увагу на натискання клавіш будуть пропущені [3].

Таймер дозволяє мікроконтролеру здійснювати вимірювання часу, розраховувати періоди та частоти сигналів. Найчастіше мікроконтролери мають 2 – 3 таймера. Основним завданням таймера – використання функцій лічильника, але таймери, як правило, також дозволяють користувачеві відмічати зовнішні події, ініціювати переривання після певної кількості тактів і навіть генерувати сигнали з широтно-імпульсною модуляцією для керування двигуном.

Аналоговий вхід/вихід: крім кількох невеликих контролерів, більшість мікроконтролерів мають інтегровані аналого-цифрові перетворювачі, які відрізняються кількістю каналів і їх роздільною здатністю (8-12 біт). У деяких випадках мікроконтролер містить цифро-аналогові перетворювачі.

Контролери, як правило, мають принаймні один послідовний інтерфейс, який можна використовувати для завантаження програми та зв'язку з ПК розробки в цілому. Оскільки послідовні інтерфейси також можна використовувати для зв'язку із зовнішніми периферійними пристроями, більшість контролерів пропонують кілька різноманітних інтерфейсів.

Оскільки критично важливі для безпеки системи є основною областю застосування мікроконтролерів, важливо запобігти помилкам у програмі та/або апаратному забезпеченні. Сторожовий таймер використовується для скидання контролера у разі «збоїв» програмного забезпечення.

Блок налагодження: деякі контролери оснащені додатковим апаратним забезпеченням, що дозволяє дистанційно налагоджувати мікросхему з ПК. Тому немає необхідності завантажувати спеціальне програмне забезпечення для



налагодження, яке має явну перевагу в тому, що помилковий код програми не може перезаписати налагоджувач [2].

## 1.2. Історія створення мікроконтролерів STM

STM32 — це мікроконтролери, побудовані на ядрі процесора Arm. Вони можуть мати від кількох Кб до кількох Мб вбудованої флеш-пам'яті для коду. Їх пам'ять на чіпі може відрізнитися залежно від чіпа. Всі вони мають велику кількість вбудованих периферійних пристроїв. На рисунку 1.4 показана схема плати з виходами і входами підключення на основі STM32 F446RE.

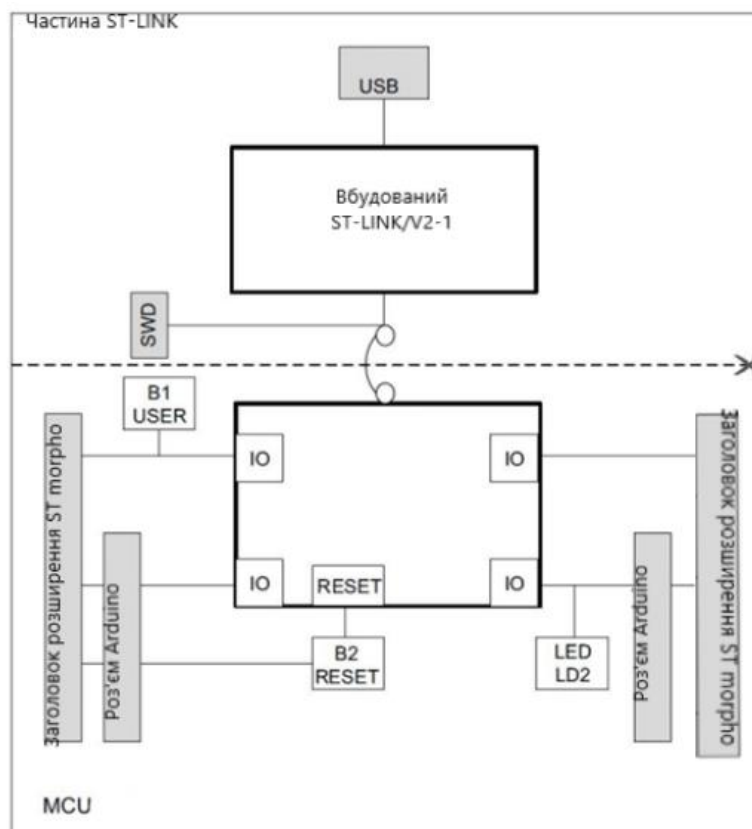


Рисунок 1.4 – Схема плати з виходами і входами підключення на основі STM32 F446RE [6]

Сімейство мікроконтролерів виробництва STMicroelectronics, які мають 32-бітну архітектуру, об'єднані в серії, кожна з яких містить одне й те саме 32-бітне

ядро ARM, таке як Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+ або Cortex-M0. Кожен мікроконтролер містить ядро процесора, яке може бути настроєне з безлічі опцій, що компанія вибирає індивідуально для кожного мікроконтролера, додаючи свої власні периферійні пристрої до ядра мікроконтролера. Крім того, кожен мікроконтролер містить статичну RAM-пам'ять, флеш-пам'ять, налагоджувальні та інші периферійні інтерфейси.

Перший процесор ARM, ARM1 (рис. 1.5), був розроблений в Acorn Computers Limited командою під керівництвом Софі Вілсон і Стіва Фербера, а перший кремній був виготовлений у квітні 1985 року. ARM1 швидко замінили ARM2, який додав апаратне забезпечення множника та використовувався в реальних системах, включаючи персональний комп'ютер Acorn Archimedes [7].

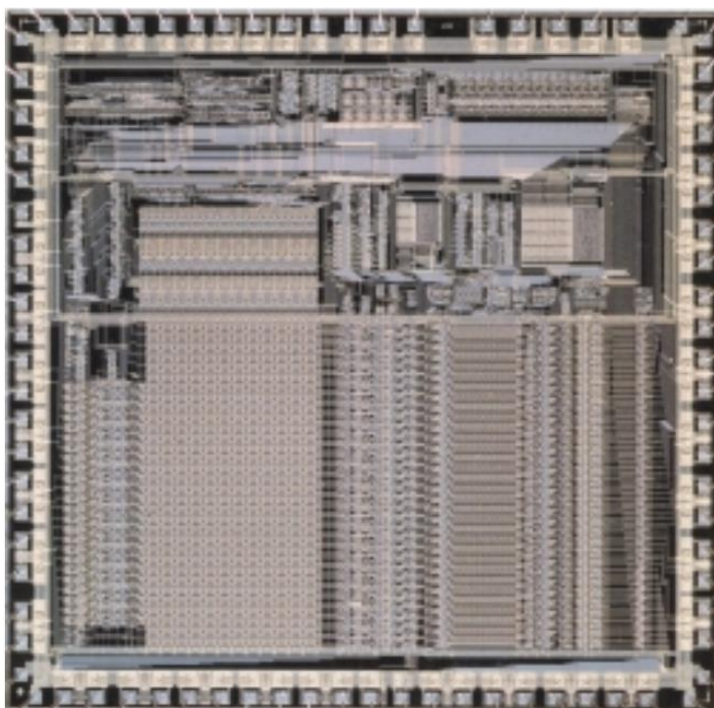


Рисунок 1.5 – Конструкція ARM1 [8]

У 1986 році було представлено проект ARM2, що працював на частоті 8 МГц і мав множник Бута, який дозволяв швидке виконання множення. У 1987 році випущено версію зі зниженою швидкістю від 10 до 12 МГц. ARM2 мав вражаючу продуктивність, набагато перевищуючи типову систему на базі 7 МГц 68000, і був

удвічі швидшим за Intel 80386, що працював на частоті 16 МГц. ARM2 мав низьке енергоспоживання і простий дизайн, який дозволяв досягати продуктивності на рівні дорогих робочих станцій за ціною, подібної до сучасних настільних ПК. ARM2 мав 32-розрядну шину даних, 26-розрядний адресний простір і 27 32-розрядних регістрів, 16 з яких були доступні в будь-який момент часу [8].

Наступник, ARM3, був створений з кеш-пам'яттю 4 КБ, що підвищило продуктивність. Шина адреси була розширена до 32 бітів в ARM6, але програмний код все ще повинен був лежати в межах перших 64 Мбайт пам'яті в 26-бітовому режимі сумісності через зарезервовані біти для прапорів стану [9].

### 1.3. Огляд мікроконтролерних платформ STM і їх використання

Розглянемо використання мікроконтролерних платформ STM на прикладі в розробці інтелектуальної системи контролю паркування [10]. Інтелектуальна система контролю паркування – це система, яка використовує різні технології для збору та обробки інформації про паркових місць і дозволяє ефективно контролювати процес паркування. Як зазначає автор, мета цього дослідження є завершення інтелектуального управління автостоянкою, щоб користувачі могли легко спостерігати за ситуацією на парковці. На рисунку 1.6 зображено основні функції, які виконує інтелектуальна система контролю паркування.



Рисунок 1.6 – Функції розумної системи контролю паркування [10]

Ця конструкція використовує чіп STC89C52RC, що виявляє паркування та транспортний рух. Вона має бездротові, сенсорні та модулі відображення. Для більшої інтуїції відображення місця для паркування, на екрані хоста показують розташування паркувальних місць. Конструкція містить три світлодіодні лампи для відображення трьох паркувальних місць. Коли автомобіль рухається, датчик, що знаходиться на дверях фіксує рух і передає інформацію на головний комп'ютер по бездротовій мережі та додає до трафіку. Коли автомобіль заїжджає на місце для паркування, відповідні записи про паркувальні місця, що були зайняті, передаються на головний комп'ютер. У той же час, світлодіодні ліхтарі відповідають паркуванню простір, що у хості освітлюється одночасно.

Програмне забезпечення, яке використовується в цій схемі – Altium Designer. Апаратне забезпечення складається з мікроконтролера STC89C52, РК-екрану LCD1602, бездротового модулю NRF24L01, датчику TCRT5000 та інфрачервоного датчику (рис. 1.7). Два підлеглих 89C52 виявляють паркувальне місце та виявлення транспортного потоку відповідно. Коли вимірюваний сигнал ефективно приймається датчиком, інформація від датчика отримується від мікроконтролера. Після обробки інформації інформація надходить до головного контролера через бездротовий модуль, а потім обробляється головним контролером для реалізації відповідної функції.

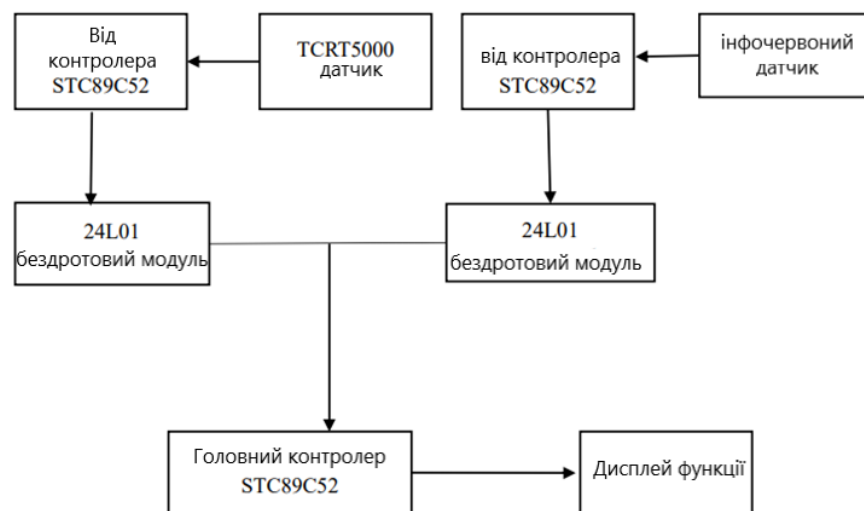


Рисунок 1.7 – Апаратне забезпечення [10]

У цій конструкції використовується чіп-процесор STC89C52, який широко використовується в електроніці. Він має 8-розрядний центральний процесор (CPU) і 128 байт внутрішньої оперативної пам'яті. РК-екран LCD1602 використовується для розробки частини дисплея, що покращує чіткість шрифту дисплея. Робоча напруга мікросхеми 4,5 В ~ 5,5 В, робочий струм 2,0 мА (5,0 В). Основний порт даних D0-D7. На рисунку 1.8 показана NRF24L01 у цій конструкції, бездротовий модуль має діапазон 2,4G ISM. У цій конструкції 24L01 керується мікроконтролером для бездротового зв'язку між головним і підлеглим комп'ютерами. На рисунку 1.9 показана внутрішня схема інфрачервоного датчика.

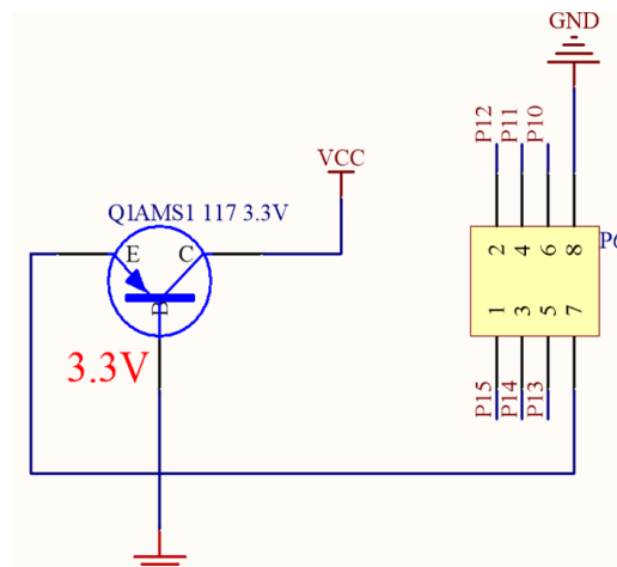


Рисунок 1.8 – Схема підключення NRF24L01 [10]

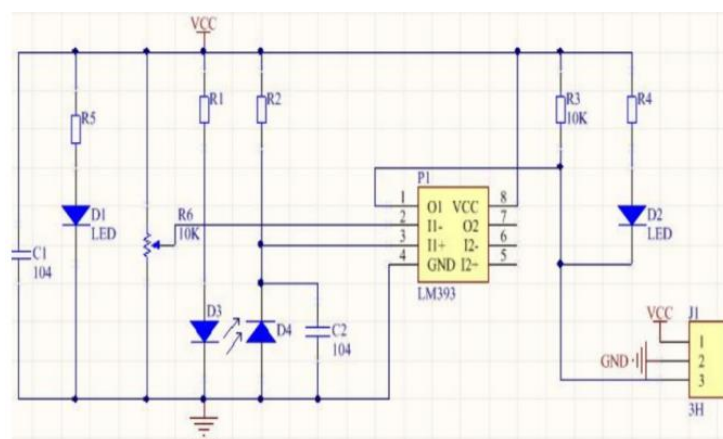


Рисунок 1.9 – Внутрішня схема інфрачервоного датчика [10]



Рисунок 1.10 – Блок-схема програмного забезпечення [10]

Для програмування використовується мова С. Програмне забезпечення – Keil u Vision5. Програмне забезпечення забезпечує оптимізацію конфігурації багатьох типів і рівнів. Блок-схема показана на рисунку 1.10.

## РОЗДІЛ 2

### ОПИС ХАРАКТЕРИСТИК STM КОНТРОЛЕРІВ

#### 2.1. Технологія STM та його особливості

STM32 є версією ядра Arm, яка є набагато досконалішою, ніж традиційний мікроконтролер. STM32 має багато ресурсів, яких немає, таких як контролер USB. Програми STM32 MCU є модульними, а інтерфейс відносно простий, оскільки він має багато функцій і працює швидко. Натомість, традиційний мікроконтролер має кілька власних функцій, вимагає багато периферійних компонентів і вимагає знайомства з електронікою. До появи мікроконтролерів з'єднаних серій STM32F105 і STM32F107 компанія STMicroelectronics випустила базову серію STM32, розширену серію, базову серію USB і додаткові серії. Нова серія продуктів наслідує частоту обробки 72 МГц у розширеній серії. Пам'ять включає флеш-пам'ять від 64 КБ до 256 КБ і вбудовану SRAM від 20 КБ до 64 КБ. Нова серія поставляється в трьох упаковках: LQFP64, LQFP100 і LFBGA100 [11].

Для більш детального розгляду технології STM32 порівняємо з мікроконтролерами з серії ESP32 (рис. 2.1). ESP32 — це єдиний комбінований чіп Wi-Fi і Bluetooth на частоті 2,4 ГГц, розроблений за 40-нм технологією ультранизького споживання TSMC. Він розроблений для досягнення найкращої потужності та радіочастотних характеристик, демонструючи міцність, універсальність і надійність у різноманітних додатках і сценаріях живлення. Серія мікросхем ESP32 включає ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, ESP32-D0WDQ6, ESP32-D2WD, ESP32-S0WD і ESP32-U4WDH, серед яких ESP32-D0WD-V3, ESP32-D0WDQ6-V3 і ESP32-U4WDH, засновані на пластині ECO V3 [12].



Рисунок 2.1 – Фізичний вигляд мікроконтролера з сімейства ESP32 (ESP32-DevKitC-32UE) [12]

Мікроконтролер ESP32 має інтегровані багато периферійних інтерфейсів, таких як SPI, IIC, IIS, AD, DA, PWM, IR, UART, CAN, що дозволяє легко підключатись до різноманітних пристроїв. Однак, через обмежену кількість ІО, основний функціонал кожного ІО залежить від конкретного застосування. Мікросхема має велику флеш-пам'ять 448 КБ та оперативну пам'ять 520 КБ, а також прямо навішену флешку на 4 МБ, що забезпечує достатній обсяг пам'яті для зберігання програмного забезпечення та даних. Швидкість мікроконтролера є дуже високою, хоча частота зовнішнього кристала становить лише 40 МГц, внутрішня основна частота може підтримувати швидкості до 80 МГц, 160 МГц, 240 МГц, а обчислювальна потужність досягає 600 MIPS. ESP32 підтримує wifi та bluetooth, але наразі не може використовуватись одночасно. Це робить його зручним для розробки пристроїв з бездротовими зв'язками. Узагалі, мікроконтролер є потужним



та зручним для розробки різноманітних пристроїв, забезпечує великий обсяг пам'яті, високу швидкість та підтримку бездротових зв'язків [11].

Для порівняння взято STM32F407VET6. Сімейство STM32F407 містить високопродуктивне 32-розрядне RISC-ядро, що працює на частоті до 168 МГц. Це ядро має блок з плаваючою комою одинарної точності та повний набір інструкцій DSP. Крім того, має вбудовану флеш-пам'ять до 1 Мбайт, до 192 Кбайт SRAM, широкий спектр розширених входів/виходів і периферійних пристроїв, 12-розрядні АЦП, ЦАП, RTC з низьким енергоспоживанням, таймери загального призначення і генератор випадкових чисел. Вони також мають стандартні та розширені комунікаційні інтерфейси [14].

Порівняння показало, що вони мають різні характеристики та призначення. ESP32 є компактним, дуже швидким та потужним мікроконтролером, який підтримує Wi-Fi та призначений для використання в Інтернеті речей. У той же час, STM32 має більшу кількість контактів та складні функції, хоча він не має підтримки Wi-Fi та Bluetooth та меншої швидкості, ніж ESP32. STM32 може забезпечити керування більшою кількістю периферійних пристроїв та застосовується в побутовій електроніці та промисловому керуванні [11].

## **2.2. Характеристики STM мікроконтролерів**

STM32 в основному має наступні характеристики. Нова серія 32-розрядних MCU базується на ядрі ARM Cortex-M, яка спеціально розроблено для вбудованих програм з високою продуктивністю, низькою вартістю та енергоспоживанням. Ця серія пропонує розширену архітектуру, що забезпечує високу продуктивність, низький рівень напруги та енергоспоживання, інноваційне ядро та периферійні пристрої. Крім того, вона є простою у використанні, безкоштовною та має низький ризик. Cortex-M3 є однією з архітектур ARMv7. З серії архітектури ARMv7 мають чітку розподіл ролей: серія "A" призначена для операційних систем та програм користувача, які використовують віртуальну пам'ять; серія "R" - для систем реального часу, а серія "M" - для мікроконтролерів [14].

Розглянемо для опису чіп STM32F103RBT6 (рис. 2.2). STM32 є мікроконтролером з ядром ARM Cortex-M3, який має 32 розряди. F є підсерією мікросхем. 103 є розширеним рядом. R позначає кількість контактів, де T - 36, C - 48, R - 64, V - 100 та Z - 144. B позначає ємність вбудованої флеш-пам'яті, де 6 - 32 Кбайт, 8 - 64 Кбайт, B - 128 Кбайт, C - 256 Кбайт, D - 384 Кбайт, E - 512 Кбайт. T позначає тип пакету, де H - BGA, T - LQFP та U - VFQFPN. 6 позначає діапазон робочих температур, де 6 - -40—85°C, а 7 - -40—105°C.

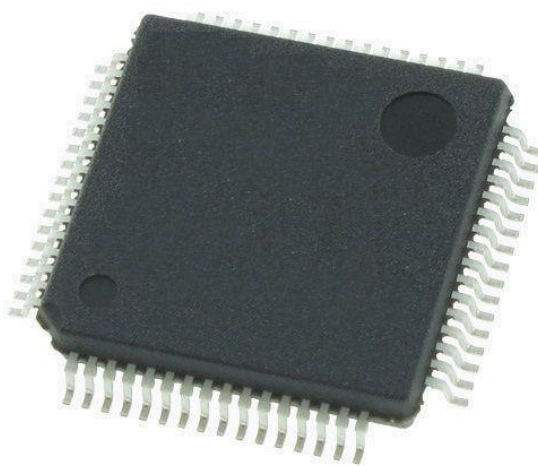


Рисунок 2.2 – Вигляд STM32F103RBT6 [15]

Загалом, мікроконтролерна платформа STM32 має наступні переваги: Висока продуктивність завдяки основному ядру Cortex. Розумні та багаті периферійні пристрої, які використовують енергію ефективно та доступні за розумною ціною. Підтримка програмного забезпечення, що включає багаті пакети програмного забезпечення. Широка технічна документація та детальне описання моделей чіпів. Наявність багатьох типів моделей чіпів, що охоплюють широкий діапазон потреб користувачів. Міцна база користувачів, оскільки компанія успішно випробувала чіп CM3 та накопичила велику кількість груп користувачів для його використання. На рисунку 2.3 зображена внутрішня блок-схема типового STM32 на основі ядра Cortex M-4.

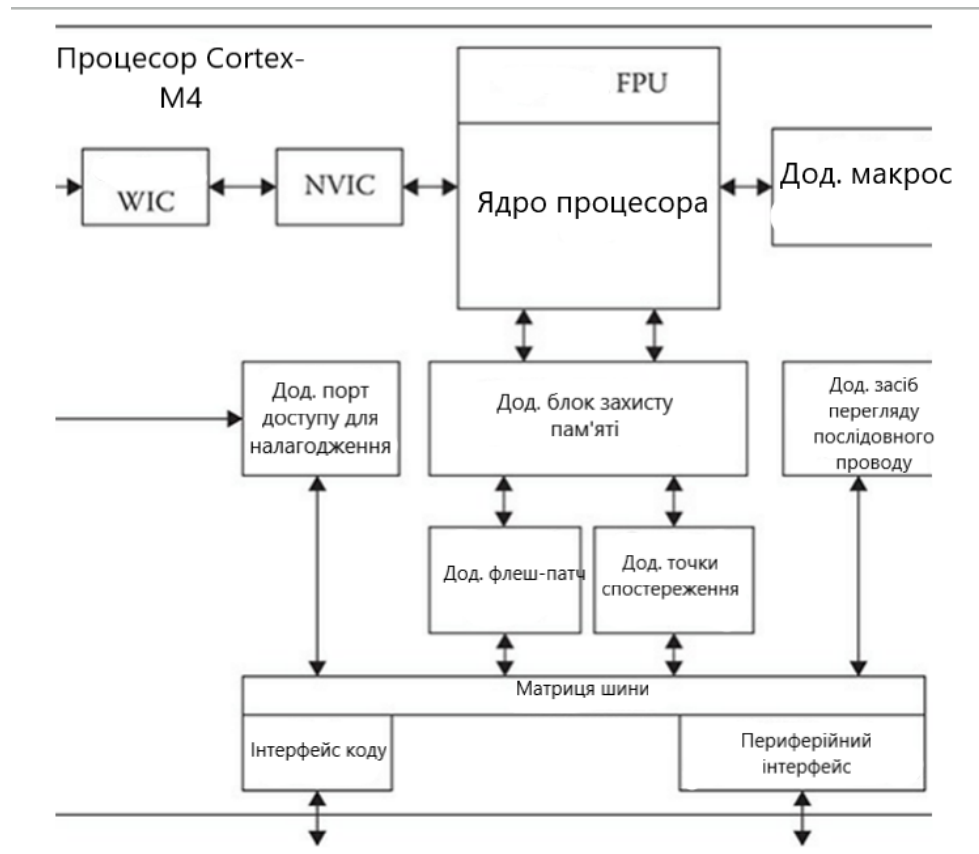


Рисунок 2.3 – Внутрішня блок-схема типового STM32 на основі ядра Cortex M0 [13]

Незважаючи на всі ці ускладнення, мікроконтролери STM32 мають деякі серйозні апаратні переваги перед іншими мікроконтролерами. Розглянемо основні характеристики STM32F030K6T6. (табл. 2.1).

Нові мікроконтролери STM32 мають ряд периферійних пристроїв, включаючи таймери, аналого-цифрові та цифро-аналогові перетворювачі, інтерфейси I2C, USART та SPI. Вони також мають 12 каналів DMA (прямий доступ до пам'яті) та блок обчислення. Нова серія мікроконтролерів продовжує переваги низької напруги та енергозбереження, має діапазон робочої напруги від 2,0 В до 3,6 В, споживаючи всього 27 мА. Чотири режими низької потужності дозволяють знизити споживаний струм до двох мА. STM32 має швидкий старт з режиму низького енергоспоживання, і для виведення мікроконтролера з режиму зупинки потрібно менше 6 мікросекунд.

Таблиця 2.1 – Основні характеристики STM32F030K6T6 [13]

Тактова частота (МГц)	48
Флешпам'ять (КБ)	32
Статична оперативна пам'ять з довільним доступом (байт) (SRAM)	4096
EEPROM	Відсутній
USART (Універсальний асинхронний приймач/передавач)	1
SPI (Послідовний периферійний інтерфейс)	1
Інтерфейсна шина I2C	1
Таймери	4 * 16 біт
Аналого-цифровий перетворювач (ADC)	10 * 12 біт
Інтерфейс для зв'язку між компонентами комп'ютерної системи (GPIO)	26
Внутрішнє посилення на АЦП	Так

### 2.3. Роль мікроконтролерів у фізичних експериментах

Мікроконтролери є важливою складовою фізичних експериментів. У фізичних експериментах мікроконтролери використовуються для керування різними пристроями та сенсорами. Це дозволяє збирати дані в реальному часі та здійснювати контроль над експериментом, щоб забезпечити його стабільність та точність.

У фізичному експерименті [18], де була розроблена швидка та високоточна система контролю температури з використанням пропорційно-інтегрально-

похідної (PID) алгоритму керування з STM32 як блоком мікроконтролера (MCU) (рис. 2.4 – 2.5).

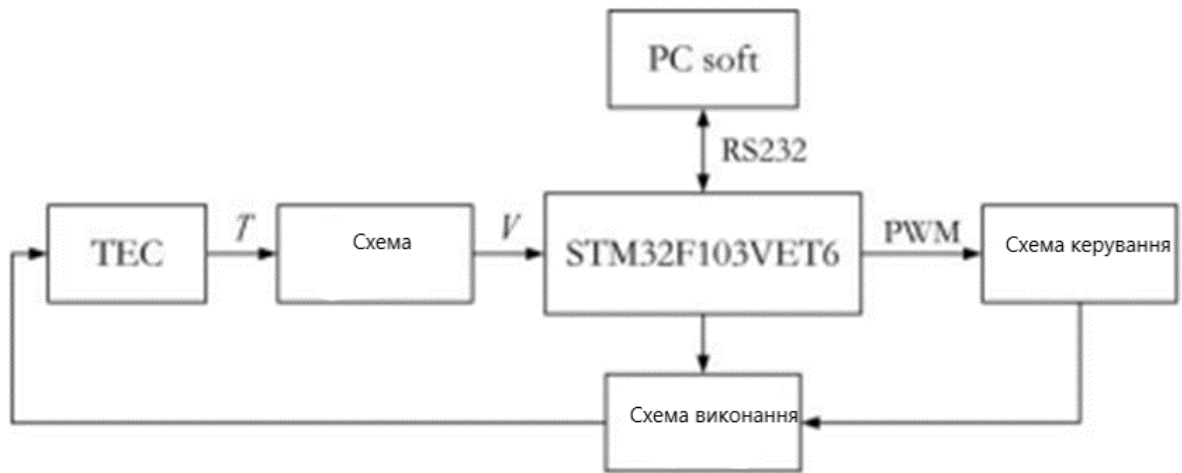


Рисунок 2.4 – Схема проекту системи контролю температури [16].

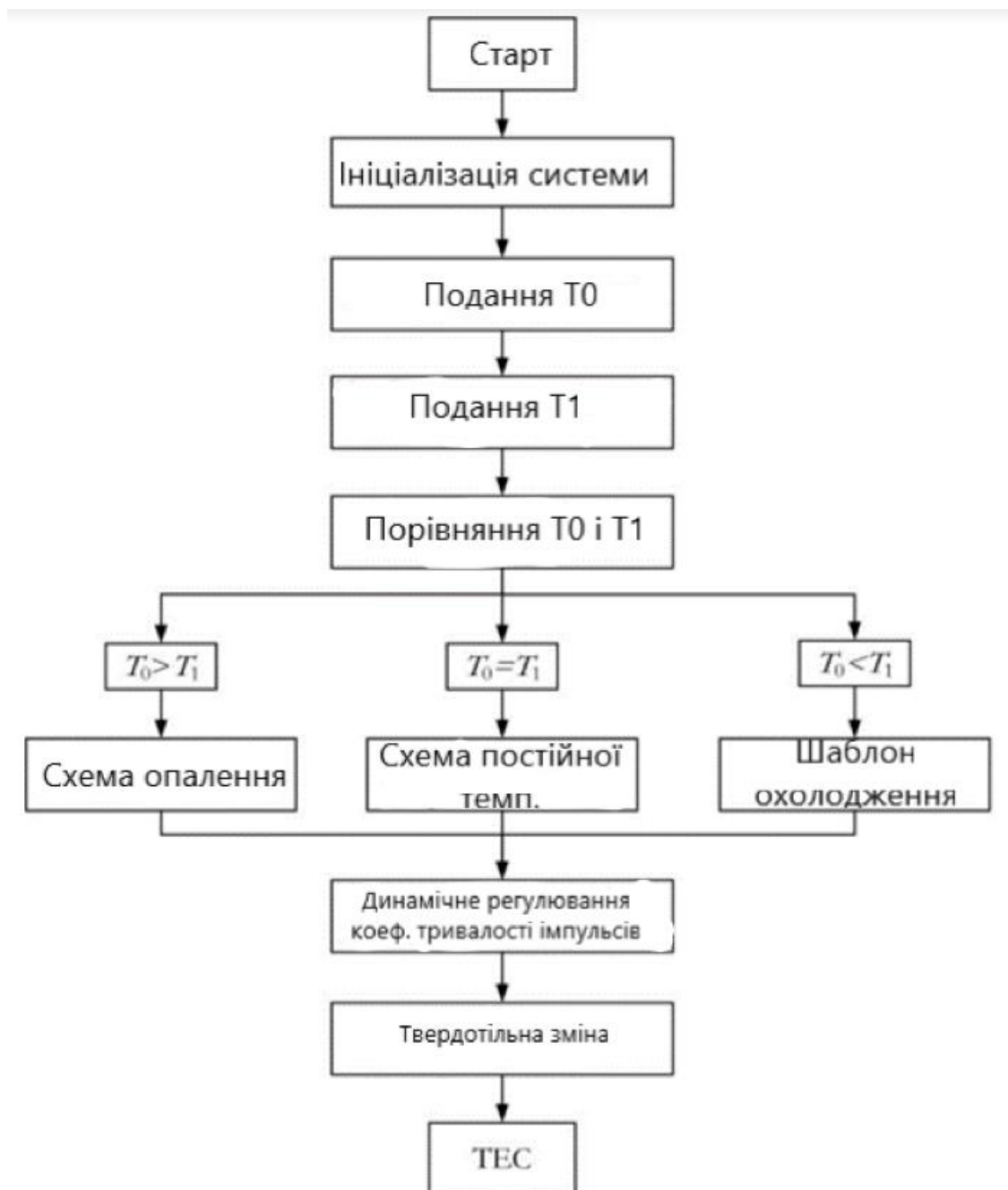


Рисунок 2.5 – Блок-схема системного програмного забезпечення [16]

Така система контролю температури може бути застосована в галузях, що пред'являються високі вимоги до швидкодії і точності регулювання температури. Схема збору температури в системі використовує термометр опору Pt1000 як датчик температури. За допомогою цієї схеми збору сигнал моніторингу фактичної температури може бути перетворений у сигнал напруги та переданий у MCU. Фотоелектричний сполучник TLP521-1 був узгоджений з силовим транзистором

BD237 для керування термоелектричним охолоджувачем (ТЕС) у модулі FTA951 (рис. 2.6).

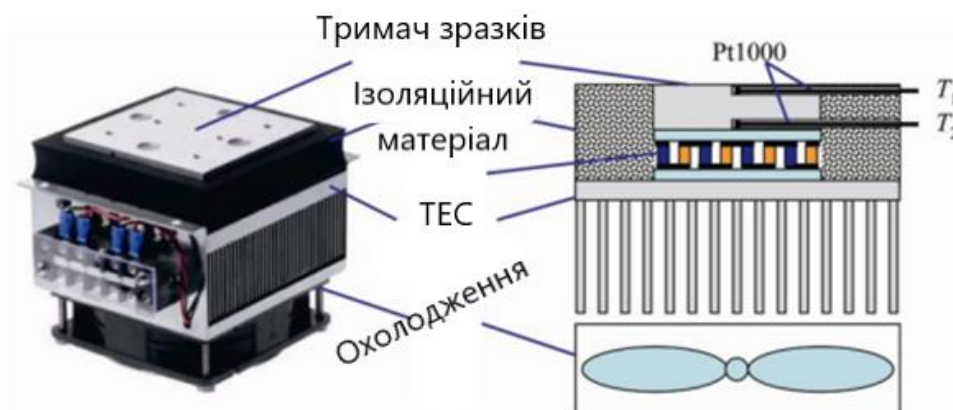


Рисунок 2.6 – Структура модуля опалення та охолодження FTA951 [17]

Ефективна електрична потужність ТЕС контролювалася за допомогою сигналів ШІМ. Параметри сигналу ШІМ можна своєчасно регулювати за допомогою алгоритму відповідно до різниці між фактичною температурою моніторингу та заданою температурою. Верхній комп'ютер використовувався для введення заданої температури та моніторингу стану роботи системи через послідовний порт. Результати прикладного експерименту показують, що система контролю температури (рис. 2.3) відрізняється простою конструкцією, швидкою реакцією, хорошою стабільністю та високою точністю регулювання температури з похибкою менше  $\pm 0,5^\circ\text{C}$ .

## РОЗДІЛ 3

### СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИМІРЮВАННЯ ТЕНЗОЕФЕКТУ

#### 3.1. Методика вимірювання тензоефекту

Тензоефект – це явище, що виникає при зміні форми або розміру об'єкта під дією механічних напруг. Він проявляється у зміні фізичних властивостей матеріалу, таких як електрична провідність, магнітна сприйнятливність, оптичні характеристики тощо. Основою для виникнення тензоефекту є тензорні взаємозв'язки між деформаціями та фізичними величинами.

Принцип виникнення тензоефекту полягає в тому, що при механічній деформації в кристалічних структурах матеріалу відбуваються зміни в просторовому розташуванні атомів. Ці зміни викликають зміну взаємодії між атомами і зміну електронної структури матеріалу. Як результат, фізичні властивості матеріалу змінюються. Такі зміни можуть бути різними залежно від типу матеріалу та специфіки деформації [18].

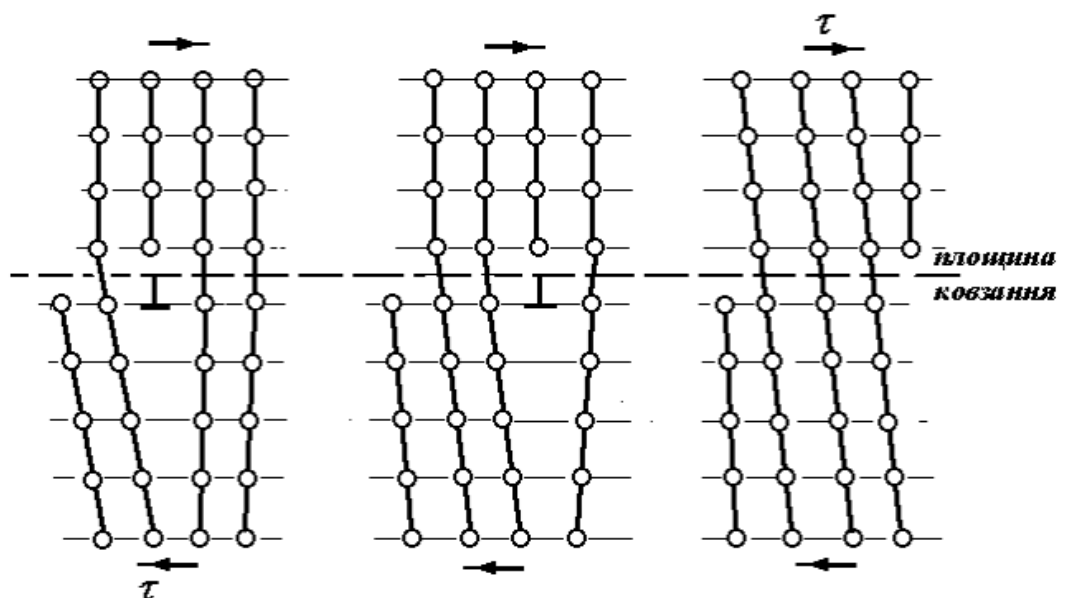


Рисунок 3.1 – Деформація атомних площин, що призводить до зміни фізичних параметрів [18]



Тензоефект характеризується чутливістю матеріалу до механічних деформацій. Основною характеристикою є коефіцієнт відносної тензочутливості, що залежить від відносного опору та довжини вимірюваного матеріалу:

$$k = \frac{\varepsilon_R}{\varepsilon_l} = \frac{\Delta R/R}{\Delta l/l} \quad (3.1)$$

де  $\varepsilon_R$  – відносний опір матеріалу;

$\varepsilon_l$  – відносна довжина матеріалу;

$R$  – опір матеріалу;

$l$  – довжина матеріалу.

Тензочутливість також залежить від типу деформації. В залежності від виду деформації, видовження матеріалу поділяються на повздовжні та поперечні. Повздовжній коефіцієнт питомого опору визначається за наступним співвідношенням:

$$\gamma = \frac{\Delta \rho / \rho}{\Delta l / l} \quad (3.2)$$

де  $\rho_{\Pi}$  – початкове значення питомого опору.

Слід зазначити що при розрахунку коефіцієнтів тензочутливості ускладнюється тим фактором, що початковий опір  $R$  збільшується. Цей фактор пояснюється залишковою деформацією. Тому для розрахунків краще використовувати миттєві коефіцієнти тензочутливості:

$$k = \frac{\delta \ln R}{\delta \ln l} \quad (3.3)$$

$$\gamma = \frac{\delta \ln \rho}{\delta \ln l} \quad (3.4)$$

Проведення експериментальної частини з визначення коефіцієнту тензоефекту від зміни фізичних властивостей матеріалу під дією мехнічних напруг буде проходити у декілька ключових етапів:

1. Виготовлення зразків матеріалу відповідно до вимог експерименту. Зразки можуть мати різні форми та розміри, залежно від конкретного тензоефекту, який досліджується.

2. Прикладання механічних навантажень до зразків, щоб створити деформацію. Це може бути розтягувальне, стискання, згинання, крутне навантаження або їх комбінація, залежно від конкретного дослідження. В нашому випадку використовуємо деформацію згинання.

3. Вимірювання змін фізичних властивостей під час навантаження зразка, пов'язаних з тензоефектом. В нашому випадку це буде зміна електричного опору за допомогою омметра.

Отримані дані про зміни фізичних властивостей під дією механічних напруг аналізуються та інтерпретуються.

### **3.2. Написання коду під систему вимірювання тензоефекту для STM32**

Нижче приведена програма, яка розраховує тензоефект самостійно (рис. 3.2). Є електродвигун, який кожну секунду на 1 градус повертається і тим самим розтягує пластинку металу. До металу підключений омметр, що вимірює опір. Потім за допомогою значень опору розраховується коефіцієнт тензочутливості. Розглянемо код програми.

Визначаються змінні `motorPin1`, `motorPin2`, що містять номер піна Arduino, до якого підключений перший пін мотора (PA6-PA15) та другий пін мотора (PA7-PA15) відповідно:

```
const int motorPin1 = PA6-PA15;
```

```
const int motorPin2 = PA7-PA15;
```

Визначаються змінні `encoderPinA`, яка містить номер піна Arduino, до якого підключений пін А енкодера (РА4-РА15) та `encoderPinA`, яка містить номер піна Arduino, до якого підключений пін А енкодера (РА4-РА15):

```
const int encoderPinA = PA4-PA15;
```

```
Arduino const int encoderPinB = PA5-PA15;
```

Після цього визначається змінна `analogPin`, яка містить номер аналогового піна Arduino для зчитування даних з датчика (РА1-РА15):

```
const int analogPin = PA1-PA15;
```

Змінні `encoderPos`, `encoderPinALast` та `n` потрібні для зберігання інформації про поточну позицію енкодера. Тому їх також прописуємо в кодї програми.

```
volatile int encoderPos = 0;
```

```
volatile int encoderPinALast = LOW;
```

```
volatile int n = LOW;
```

У програмному кодї "int" є ключовим словом мови програмування C++ і використовується для оголошення змінних цілочисельного типу даних. Він може займати певну кількість байтів в пам'яті, залежно від платформи та компілятора, але зазвичай використовується для представлення 32-бітних цілих чисел в діапазоні від -2,147,483,648 до 2,147,483,647. Для встановлення режиму пінів, що є вихідними `motorPin1` і `motorPin2`, для вхідних з внутрішнім підтягуванням до живлення `encoderPinA` і `encoderPinB` та `analogPin` - як вхідний прописуємо функцію `setup()`:

```
void setup() {
```

```
pinMode(motorPin1, OUTPUT);
```

```
pinMode(motorPin2, OUTPUT);
```

```
pinMode(encoderPinA, INPUT_PULLUP);
```

```
pinMode(encoderPinB, INPUT_PULLUP);
```

```
pinMode(analogPin, INPUT);
```

```
attachInterrupt(digitalPinToInterrupt(encoderPinA), updateEncoder, CHANGE);}

```

Функція `attachInterrupt()` відповідає за підключення переривання зміну стану піна А енкодера (`encoderPinA`). Це означає, що кожного разу, коли стан піна А змінюється, виконується наступна функція `updateEncoder()`:

```
void updateEncoder() {
  int MSB = digitalRead(encoderPinA);
  int LSB = digitalRead(encoderPinB);
  int encoded = (MSB << 1) | LSB;
  int sum = (encoderPinALast << 2) | encoded;
  if (sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011) {
    encoderPos++; }
  else if (sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000) {
encoderPos--; }
  encoderPinALast = encoded; }
```

Функція `updateEncoder()` викликається при зміні стану піна А енкодера. Вона зчитує стан пінів А і В, кодує їх значення і об'єднує зі значенням попереднього стану піна А, щоб визначити нову позицію енкодера. Залежно від комбінації станів пінів А і В збільшується або зменшується значення `encoderPos`.

У функції `loop()` викликається функція `rotateMotor()` для повороту мотора на 1 градус:

```
void loop() {
  rotateMotor(1);
  int sensorValue = analogRead(analogPin);
  float k = 5 * sensorValue / 1023.0;
  float fsrR = 3230 * k / 5;
  Serial.println(fsrR);
  Serial.println("ohm");
  Serial.println(sensorValue);
  delay(1000); }
```

Після цього зчитуються дані з аналогового піна `analogPin` за допомогою функції `analogRead()`. Отримане значення конвертується у величину `sensorValue`

і використовується для обчислення значення опору (`fsrR``) та його виведення через послідовний порт (функція `Serial.println()`).

```
void rotateMotor(int degrees) {
  int targetPos = encoderPos + degrees; int direction = degrees > 0 ? HIGH : LOW;
  while (encoderPos != targetPos) {
    digitalWrite(motorPin1, direction);
    digitalWrite(motorPin2, !direction); }
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, LOW); }
```

Функція, що приведена вище, `rotateMotor()` обертає мотор на вказану кількість градусів. Вона визначає цільову позицію мотора шляхом додавання к поточній позиції значення `degrees`. В залежності від знаку `degrees` встановлюється напрямок обертання. Запускається цикл і він продовжується, поки поточна позиція енкодера не досягне цільової, встановлюється напрямок обертання мотора шляхом встановлення відповідних значень на пінах `motorPin1` і `motorPin2`. Після досягнення цільової позиції мотор зупиняється шляхом встановлення LOW на пінах `motorPin1` і `motorPin2`. Цикл `loop()` виконується постійно, повторюючи описані вище дії, поки Arduino не буде вимкнено або перезавантажено.

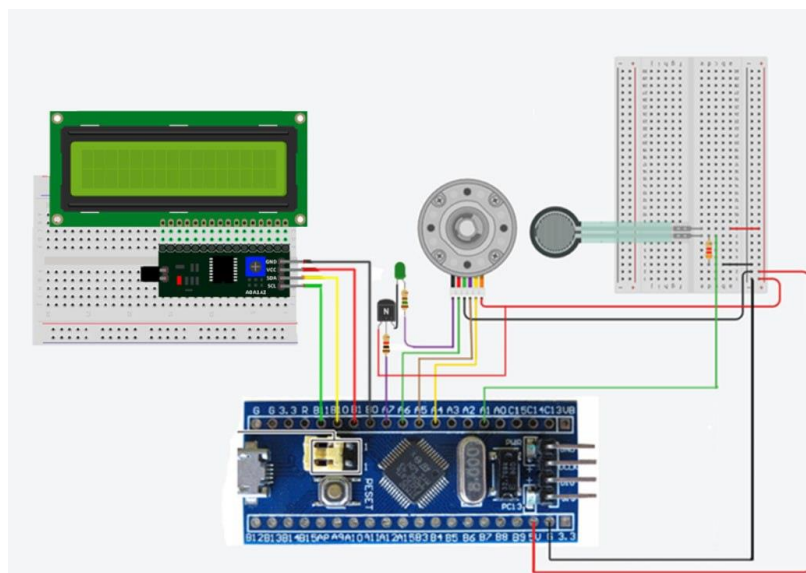


Рисунок 3.2 – Зібрана схема системи автоматичного виміру тензоефекту

## ВИСНОВКИ

1. Вивчено загальні відомості мікроконтролерів та проведено огляд мікроконтролерних платформ. Продемонстровано принципи дії покладені у роботу мікроконтролерів, розглянута його архітектура та наведені основні блоки, що взаємодіють у процесі розрахункових дій.
2. Розглянуто характеристики STM контролерів, технологія використання такого типу мікроконтролерів. Характеристики STM32 показали, що дані мікроконтролери мають високу продуктивність, низьку вартість та енергоспоживання. Також нові версії мікроконтролерів мають ряд периферійних пристроїв, включаючи таймери, аналого-цифрові та цифро-аналогові перетворювачі, інтерфейси I2C, USART та SPI.
3. Продемонстровано роль мікроконтролерів у фізичних експериментах та на прикладі представлена система контролю температури з використанням пропорційно-інтегрально-похідної алгоритму керування з STM32 як блоком мікроконтролера.
4. Розроблена система автоматичного виміру тензоефекту на основі мікроконтролера STM32. За допомогою електродвигуну реалізовано поступовий розтяг пластини досліджуваного матеріалу, що призводить виникнення тензоефекту, а омметром зчитується опір. Автоматично розраховується і виводиться коефіцієнт тензочутливості.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hur B. Learning Embedded Systems with MSP430 FRAM Microcontrollers: MSP430FR5994 with Code Composer Studio. Byul Hur, 2022. 234 p.
2. Harish Kondoor. Kick-Start to MicroPython using ESP32 / ESP8266. 2021. 61 p.
3. Miguel Angel Garcia-Ruiz. DIY Microcontroller Projects for Hobbyists: The ultimate project-based guide to building real-world embedded applications in C and C++ programming, 2021. 320 p.
4. Amos B. Hands-On RTOS with Microcontrollers: Building Real-Time Embedded Systems Using FreeRTOS, STM32 MCUs, and SEGGER Debug Tools. Packt Publishing, Limited, 2020. 496 p.
5. Mazidi M. A., Naimi S., Naimi S. STM32F103 Arm Microcontroller and Embedded Systems: Using Assembly and C. MicroDigitalEd, 2020. 544 p.
6. Mazidi M. A., Chen S., Ghaemi E. STM32 Arm Programming for Embedded Systems. MicroDigitalEd, 2018. 378 p.
7. ARM Cortex-A Series Programmer's Guide for ARMv7-A. Arm Developer.
8. Markus Levy. The History of The ARM Architecture: From Inception to IPO. Convergence Promotions. 6 p.
9. He C., Li H., Ren B. Design of Intelligent Parking Control System. // Journal of Physics: Conference Series. 2021. Vol. 2113, no. 1. P. 012019.
10. Moyes K. ESP32 vs STM32, Which is Better and How to Choose 2022 Embedded Technology Information EmbedIc. EmbedIc - Embedded Technology Full-Service Platform. 2022. 8 p.
11. ESP32 Series Datasheet. China: Espressif. 2023. 64 p.
12. Donald J. Norris. Programming with STM32: Getting Started with the Nucleo Board and C/C++.e United States: McGraw-Hill Education. 2018. 41 p.
13. STM32 Microcontroller detailed interpretation Embedded Technology Information EmbedIc. EmbedIc - Embedded Technology Full-Service Platform. 2020. 5 p.

14. John Teel. ATmega versus STM32: Which Microcontroller is Best for Your Application. USA. Harvard University. 2022. 12 p.
15. STM32F103x8, STM32F103xB datasheet. Switzerland: STMicroelectronics. 2022. 116 p.
16. J. Zhang. Design of PID temperature control system based on STM32 // IOP Conference Series: Materials Science and Engineering. 2018. Vol. 322. P. 072020.
17. Anbazhagan K. Programming STM32 Microcontroller circuit: STM32 Microcontroller, Keil uVision and STM32CubeMX, ESP8266 with STM32F103C8, Stepper & Servo Motor with STM32F103C8, Heartbeat Modulation. 2020. 205 p.
18. Layers with the tensoresistive properties and their possible applications in medicine / L.P. Ichkitidze, A.Yu. Gerasimenko, V.M. Podgaetsky, S.V. Selishchev. // Materials Physics and Mechanics Materials Physics and Mechanics. 2018. V. 37. P. 153-158 .