

Міністерство освіти і науки України  
Державний заклад  
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

**Гирич Наталія Аркадіївна**

**РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ ДОМАШНЬОЇ МЛИННИЦІ НА  
БАЗІ STM32**

**кваліфікаційна робота  
здобувача вищої освіти другого (магістерського) рівня  
освітньої програми «Комп'ютерні мережі»  
за спеціальністю 123 Комп'ютерна інженерія**

Особистий підпис \_\_\_\_\_ Наталія ГИРИЧ

Науковий керівник \_\_\_\_\_ Геннадій МОГИЛЬНИЙ,  
кандидат технічних наук, доцент  
кафедри інформаційних технологій  
та систем

Завідувач кафедри \_\_\_\_\_ Микола СЕМЕНОВ,  
кандидат педагогічних наук, доцент  
кафедри інформаційних технологій  
та систем

Полтава – 2025

## АНОТАЦІЯ

**Гирич Н. А.**

**Тема:** Розробка системи управління домашньої млинниці на базі STM32.

**Спеціальність:** 123 «Комп'ютерна інженерія».

**Установа:** ЛНУ імені Тараса Шевченка, 2025 р.

**Магістерська робота містить:** 64 с., 42 рис., 2 табл., 2 додат., 27 джерел.

**Об'єкт дослідження** – системи автоматичного управління побудовані на мікроконтролерах STM32.

**Предмет дослідження** – системи управління домашньої млинниці на базі STM32.

**Мета дослідження** – розробка системи автоматичного управління для побутової млинниці на основі сучасних мікроконтролерів сімейства STM32.

**Результати роботи.** В результаті роботи було проведено вивчення існуючих пристроїв класу який розробляється визначено вигляд пристрою і визначені його основні частини, розроблена конструктивна принципова схема і принципова схема управління.

**Висновок.** Розроблено принципову конструктивну схема пристрою і принципова схема системи управління млинниці, структурно-функціональна схема, обрані комплектуючі та створено програму системи управління.

**Ключові слова.** СИСТЕМА УПРАВЛІННЯ, ЕЛЕМЕНТНА БАЗА, МІКРОКОНТРОЛЕР.

## ABSTRACT

**Hyrych Nataliia**

**Theme:** Development of a control system for a home mill based on STM32.

**Speciality:** 123 "Computer Engineering"

**Institution:** Luhansk Taras Shevchenko National University (LTSNU), 2025.

**Master's work of:** 64 pages, 42 Fig., 2 Table, 2 adj., 27 source.

**A research object is** automatic control systems built on STM32 microcontrollers.

**The article of research is** control systems for a home mill based on STM32.

**An aim of work is** development of an automatic control system for a household flour mill based on modern microcontrollers of the STM32 family.

**Job performances.** As a result of the work, a study of existing devices of the class being developed was conducted, the appearance of the device was determined and its main parts were identified, a structural schematic diagram and a schematic control diagram were developed.

**Conclusions.** A schematic diagram of the device and a schematic diagram of the mill control system have been developed, a structural and functional diagram, components have been selected, and a control system program has been created..

**Keywords.** CONTROL SYSTEM, ELEMENT BASE, MICROCONTROLLER.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>5</b>
<b>РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ПРИСТРОЇВ ДЛЯ ВИГОТОВЛЕННЯ МЛИНЦІВ. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ УПРАВЛІННЯ.....</b>	<b>8</b>
1.1. Пристрої для виготовлення млинців.....	8
1.2. Теоретичні основи автоматичного керування .....	13
Висновки до розділу .....	24
<b>РОЗДІЛ 2. РОЗРОБКА ПРИНЦИПОВОЇ КОНСТРУКТИВНОЇ СХЕМИ ПРИЛАДУ ТА ПРИНЦИПОВОЇ СХЕМИ СИСТЕМИ КЕРУВАННЯ МЛИННИЦІ .....</b>	<b>25</b>
2.1. Принципова конструктивна схема та принцип функціонування приладу .....	25
2.2. Структурно-функціональна схема приладу .....	28
Висновки до розділу .....	35
<b>РОЗДІЛ 3. СИНТЕЗ ПРИНЦИПОВОЇ СХЕМИ ПРИСТРОЮ .....</b>	<b>36</b>
3.1. Опис плати STM32F4 Discovery.....	36
3.2. Принципова схема пристрою.....	41
3.3. Підготовка проєкту у Keil mVision 5.20.....	53
Висновки до розділу .....	60
<b>ВИСНОВКИ .....</b>	<b>61</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>62</b>
<b>ДОДАТОК А .....</b>	<b>65</b>
<b>ДОДАТОК Б .....</b>	<b>75</b>

## ВСТУП

Швидкий розвиток технологій суттєво трансформує сферу кулінарії, надаючи споживачам інноваційні пристрої, які спрощують процес приготування їжі та підвищують його якість. Автоматизація та інтеграція цифрових технологій у кухонне обладнання відкривають нові можливості для кулінарних експериментів як для професійних кухарів, так і для аматорів.

Сучасні кухонні прилади обладнані різноманітними сенсорами та підключені до інтернету, що дозволяє їм виконувати складні кулінарні операції з високою точністю. Розумні кухонні гаджети, такі як ваги, термометри та пательні, забезпечують автоматичний контроль над процесом приготування їжі, виключаючи людський фактор і гарантуючи досягнення ідеального результату. Наприклад, інтелектуальні пательні здатні автоматично регулювати температуру нагрівання та сповіщати користувача про готовність страви, а розумні термометри для м'яса дозволяють точно контролювати ступінь прожарювання.

Інтеграція кухонних приладів з мобільними додатками відкриває нові можливості для навчання кулінарному мистецтву. Мобільні додатки надають користувачам доступ до великої бази рецептів, детальних інструкцій та порад професійних кухарів. Крім того, такі додатки дозволяють створювати персоналізовані кулінарні плани та відстежувати прогрес у приготуванні їжі.

Таким чином, інноваційні технології перетворюють кулінарію на інтерактивну та інтуїтивно зрозумілу діяльність, доступну для широкого кола споживачів. Автоматизація та цифровізація кухонного обладнання не тільки спрощують процес приготування їжі, але й відкривають нові можливості для творчості та експериментів на кухні.

Автоматизація процесів приготування їжі є одним з найактуальніших напрямків розвитку побутової техніки. Сучасні кухонні прилади, такі як пароварки, мультиварки та хлібопічки, демонструють високий рівень інтеграції мікропроцесорної техніки та сенсорних систем, що забезпечує точне регулювання технологічних параметрів і автоматизацію багатьох операцій.

Особливий інтерес представляє розробка автоматизованих пристроїв для приготування традиційних страв, які мають важливе культурне значення. Одним з таких прикладів є млинці, що є невід'ємною частиною кулінарних традицій багатьох народів. Незважаючи на популярність млинців, на ринку побутової техніки відсутні доступні за ціною та зручні в експлуатації автоматичні млинниці.

**Об'єкт дослідження** – системи автоматичного управління побудовані на мікроконтролерах STM32.

**Предмет дослідження** – системи управління домашньої млинниці на базі STM32.

**Мета дослідження** – розробка системи автоматичного управління для побутової млинниці на основі сучасних мікроконтролерів сімейства STM32.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- детальне вивчення конструктивних особливостей та функціональних можливостей сучасних автоматизованих кухонних приладів з метою визначення перспективних напрямків розвитку;
- створення деталізованої структурної схеми автоматичної млинниці, що включає механічні, електронні та програмні компоненти;
- вибір оптимальних типів сенсорів, виконавчих механізмів та мікроконтролера, розробка алгоритмів управління процесом приготування млинців;
- створення програмного коду для мікроконтролера, що забезпечує реалізацію алгоритмів управління та взаємодію з користувачем.

**Методи дослідження** – основи мікроелектроніки, цифрова схемотехніка, проведення експерименту.

Результатом виконання даного дослідження стане розробка прототипу автоматичної побутової млинниці, що відрізнятиметься високою точністю приготування, простотою використання та доступною вартістю. Такий пристрій зможе значно полегшити процес приготування млинців та підвищити якість готової страви.

Актуальність цього дослідження зумовлена зростаючим попитом сучасних споживачів на зручні та швидкі у використанні кухонні прилади, а також відсутністю на ринку доступних автоматичних млинниць. Розробка такої системи не лише задовольнить потреби споживачів, а й відкриє нові перспективи для розвитку вітчизняної електроніки та підвищення конкурентоспроможності українських виробників побутової техніки на світовому ринку. Крім того, розроблена система управління може стати основою для створення інших інноваційних кухонних приладів.

# РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ПРИСТРОЇВ ДЛЯ ВИГОТОВЛЕННЯ МЛИНЦІВ. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ УПРАВЛІННЯ

## 1.1. Пристрої для виготовлення млинців

Млинцевий апарат АПП-1Т призначений для випічки млинців з заздалегідь приготованого тесту. Після випічки з одного боку, блін заповнюється начинкою, після чого автоматично згортається в трубочку.

Продуктивність до 120 шт / год

Вага одного млинця- 50 гр.

Розмір одного млинчика-200х300мм



Рис. 1.1. Млинцевий апарат АПП-1Т

Процес випікання млинців полягає в наступному: тісто рівномірно розподіляється по поверхні обертового нагрітого барабана. Після кожного повного оберту готовий млинець зрізається спеціальним ножем, а на очищену поверхню автоматично подається нова порція тіста за допомогою дозувального валика. Час випікання та температура нагріву барабана встановлюються заздалегідь і підтримуються автоматично, що забезпечує високу якість готових млинців.

Автоматична машина для випікання млинців Monferrina C3 – це високопродуктивний обладнання, призначене для випікання млинців різного діаметра (від 100 до 550 мм) у промислових масштабах. Машина оснащена насосом, який автоматично подає рідке тісто на жарочний барабан, при цьому витрату тіста можна регулювати. Швидкість обертання барабана також

регулюється, що дозволяє оптимізувати процес випікання. Температура поверхні барабана підтримується за допомогою термостата. Загальні габарити машини становлять 1300x1180x1371 мм, а вага – 378 кг. Ширина жарочної поверхні барабана становить 600 мм. Продуктивність машини може досягати 120-150 метрів стрічки млинців на годину, залежно від рецептури тіста.



Рис. 1.2. Автоматична машина для випічки млинців Monferrina C3

Ідея використання 3D-принтерів для приготування їжі колись здавалась фантастикою. Спочатку ця технологія викликала великий інтерес як у широкої публіки, так і у наукових кіл, зокрема NASA. Проте, незважаючи на перспективні плани, галас навколо 3D-друку їжі поступово стих. Однак професіонали галузі продовжують активно працювати над розробкою нових моделей 3D-принтерів для харчової промисловості, що свідчить про великий потенціал цієї технології.

Розглянемо кілька цікавих моделей.



Рис. 1.3. Chocola3D

Харчовий принтер Chocola3D White - перший професійний харчовий 3D-принтер виробництва України, спільне дітище 3divoprint і Chocola3d. Принтер друкує шоколадом, кондитерськими сумішами і глазур'ю, хумусом, сиром, паштетом, овочевими та фруктовими пастами, а також іншими пастоподібних харчовими матеріалами.

Принтер має вбудований мікрокомп'ютер на платформі Raspberry Pi3, з оригінальним ПО прошивки і семидюймовим кольоровим HD-дисплеєм, що дозволяє йому працювати автономно, без підключення до комп'ютера, з SD-картки або флеш-накопичувача.

Зв'язок з комп'ютером або мобільним технікою - планшетом або смартфоном, з якими принтер також працює, здійснюється за допомогою WiFi-модуля. Передбачено і управління через WEB.

Корпус апарату зроблений зі сталі і пофарбований фарбою, матеріал всіх контактують з їжею деталей сертифікований для харчового виробництва. Робоча область принтера - 25x21x18 сантиметрів.

Зовнішні розміри пристрою - 51x48x55 сантиметрів, а вага - 17 кілограм. Екструдер, також власної розробки, друкує харчовим матеріалом з точністю в 1,0 - 0,7 мм і зі швидкістю до 50 мм / сек, він оснащений комплектом змінних сопел для різних продуктів, а контейнери для матеріалу мають обсяг 60 кубічних сантиметрів.

У комплект поставки входять навчальні матеріали, ПО, набір 3D-моделей, доступ до онлайн-консультації і річна гарантія, а також бізнес-кейси застосування 3D-друку їжею.

Опціонально можна придбати також змінні екструдери для друку спеціальними матеріалами (тісто, карамель), закритий корпус з додатковим охолодженням і web-камеру.

Принтер оснащений автокалібруванням, що дозволяє друкувати відразу на сторонніх поверхнях - на тарілці, тості або крекері, наприклад.

Оснащений принтер і системою терморегулювання і обдування друкованої деталі, що охолоджує шоколад для якнайшвидшого застигання.

Основна робоча поверхня в комплекті зроблена із загартованого скла. У конструкції передбачений датчик наповнення контейнера, який сигналізує про те, що матеріал для друку закінчується.

Для роботи використовуються Open Source-слайсери, такі як Cura, Simplify і подібні.

Краща сфера застосування такого 3D-принтера: кондитерське виробництво, створення сувенірної харчової продукції, ресторани і кафе.



Рис. 1.4. Pancakebot

Pancakebot - дуже спеціалізований 3D-принтер. Конструктивно він схожий на будь-який харчовий 3D-принтер, але друкує виключно млинці та оладки будь-якої заданої форми і, в процесі друку, відразу смажить їх.

Розглянемо принтер ближче: відразу помітно, що представляє він собою 2 осьової ЧПУ верстат, з формальної 3 віссю - екструдер вручну можна розташувати на одному з 3 рівнів висоти, це може допомогти, якщо ви хочете не тільки приготувати млинець, а й намалювати на ньому що-небудь.

Даний 3D-принтер зазнав ряд конструктивних змін в порівнянні з попередніми версіями.

Область друку приблизно 430x210 мм. Можна помістити цілий пиріг!

Екструдер являє собою звичайну пляшку, в яку заливається тісто або інший наповнювач, до неї підключається трубка, яка створює в пляшці вакуум, щоб тісто не витікало, або повітряний тиск, щоб впливало з потрібною силою. Тиск повітря в пляшці контролюється на передній панелі за допомогою спеціального реле.

Принтер складається з 2 частин - корпусу з направляючими і плити з антипригарним покриттям. Живляться плита і принтер окремо, так що вам буде потрібно 2 розетки, оскільки розетки американські вам так само будуть потрібні перехідники під них. У плити є свій регулятор температури, так що принтер цілком можна використовувати як велику сковороду, яєчню на ньому можна приготувати без проблем.

Тепер розглянемо те без чого цей пристрій не більш ніж дорога плитка - програмне забезпечення, називається воно PancakePainter і викачується з офіційного сайту PancakeBot.

Функціонал програми, звичайно, не дотягує і до Paint, втім, це ж програма для малювання млинчиків, навіщо всі ці надмірності?

По суті, у вас є 2 інструменти - кисть, або вільна, або малює прямі лінії і заливка. Кольори 4 - вони визначають, в якому порядку буде друкуватися картина, і слідчо яка частина малюнка сильніше підсмажиться.

Для окреслення можна завантажити картинку на робоче поле. Ну і після того як ви намалювали лінію, ви можете відредагувати її по точках.

Функціонал програми на цьому закінчується. Для більшої зручності малювання можна підключити до комп'ютера графічний планшет, тоді малювати стає значно приємніше, хоча відсутність будь-якого гумки періодично дратує.

Принцип роботи принтера вкрай простий, ви включаєте плиту на потрібну температуру, заливаєте тісто в пляшку, ставите її на рухому рамку, і включаєте вашу програму з SD-карти або через USB-кабель безпосередньо з комп'ютера.

Залишається тільки налаштувати силу подачі тесту і можна йти, друкує принтер нешвидко, гріє плита не сильно, тому на друк середнього млинчика йде 5-10 хвилин часу.

Тільки далеко краще не йти, так як після того як млинець буде закінчений і принтер постоїть без завдання хвилин 10 - він відключиться і тісто з відкритого екструдера почне литися прямо на принтер.

Як показала практика тісто краще робити більш густим, так як сильно рідке тісто навіть на мінімальній подачі ллється великим потоком і заливає малюнок. Так само бажано пропускати тісто через сито, щоб позбутися від грудочок, які можуть застрягти в соплі в самий незручний момент, коли ви пішли намалювати ще більш божевільний млинець, наприклад.

## **1.2. Теоретичні основи автоматичного керування**

Автоматика – це міждисциплінарна галузь науки та техніки, що вивчає принципи створення, аналізу та синтезу систем автоматичного керування. Її метою є розробка систем, здатних самостійно виконувати задані функції без постійного втручання людини [1].

Теорія автоматичного керування – фундаментальна складова автоматики, що займається розробкою математичних моделей та алгоритмів, які описують поведінку систем автоматичного керування. Ця дисципліна дозволяє аналізувати стійкість, точність, швидкодію та інші характеристики систем, а також синтезувати оптимальні структури керування з урахуванням зовнішніх збурень та обмежень [1].

Технічні засоби автоматики – це сукупність приладів, механізмів та електронних компонентів, які реалізують алгоритми керування, розроблені в рамках теорії автоматичного керування. Вони забезпечують виконання конкретних функцій системи, таких як вимірювання, перетворення сигналів, прийняття рішень та управління виконавчими механізмами.

Сучасна автоматика активно використовує досягнення таких наук, як математика, фізика, інформатика та електротехніка. Вона знаходить широке застосування в різних галузях, включаючи промисловість, енергетику, транспорт, медицину та побутову техніку. Завдяки розвитку комп'ютерних технологій та мікроелектроніки, сучасні системи автоматичного керування стають все більш складними, інтелектуальними та надійними.

Система автоматичного керування (САК) – це сукупність технічних засобів, призначена для автоматичного підтримання заданого режиму роботи об'єкта керування. САК здійснює збір інформації про стан об'єкта, обробку цієї інформації та вироблення керуючих впливів, які подаються на виконавчі механізми [2, 3].

Основні завдання автоматики як науки:

1. Дослідження об'єктів керування. Вивчення властивостей та характеристик об'єктів, що підлягають автоматизації, з метою розробки ефективних алгоритмів керування.
2. Розробка алгоритмів керування. Створення математичних моделей і алгоритмів, які описують поведінку САК та забезпечують досягнення заданих цілей керування.
3. Аналіз та синтез САК. Дослідження стійкості, точності, швидкодії та інших характеристик САК, а також синтез оптимальних структур керування.

Типова САК складається з таких основних елементів:

- датчики: перетворюють фізичні величини, що характеризують стан об'єкта керування, в електричні сигнали, зручні для подальшої обробки.

- пристрій порівняння: порівнює фактичне значення регульованої величини з заданим і формує сигнал розходження.
- керуючий пристрій: обробляє сигнал розходження і виробляє керуючий сигнал.
- виконавчий механізм: перетворює керуючий сигнал у фізичну дію на об'єкт керування.

Сигнали в САК – це фізичні величини, що несуть інформацію про стан системи. Вони можуть бути аналоговими (неперервні за часом і амплітудою) або дискретними (переривчасті за часом або амплітудою). Основними сигналами в САК є вхідні сигнали (збурення) та вихідні сигнали (відгуки системи).

Сучасна автоматика активно використовує досягнення мікроелектроніки, комп'ютерних технологій та теорії керування. Завдяки цьому з'явилися нові можливості для створення складних і ефективних систем автоматичного керування, які знаходять широке застосування в різних галузях науки і техніки.

Автоматичне керування – це галузь науки та техніки, що вивчає принципи побудови систем, здатних самостійно виконувати задані функції без безпосередньої участі людини. Сутність автоматичного керування полягає у впливі на об'єкт з метою досягнення бажаного стану або процесу.

Система автоматичного керування (САК) – це сукупність технічних засобів, що забезпечують автоматичний контроль і регулювання технологічних процесів. Основними елементами САК є датчики, пристрої порівняння, керуючі пристрої та виконавчі механізми. Датчики збирають інформацію про стан об'єкта, пристрій порівняння порівнює фактичний стан з заданим, керуючий пристрій обробляє відхилення і формує керуючий сигнал, а виконавчий механізм безпосередньо впливає на об'єкт [2, 3, 4, 5, 6, 7].

Приклади САК:

Системи автоматичного регулювання: підтримують задані параметри процесу (температура, тиск, рівень тощо).

Системи автоматичного управління: керують рухом об'єктів (автопілоти, роботи).

Системи автоматичного контролю: здійснюють контроль за параметрами процесу і виявляють відхилення від заданих значень.

Автоматика відіграє все більш важливу роль у сучасному суспільстві. Вона використовується у виробництві, енергетиці, транспорті, медицині та багатьох інших галузях. Завдяки автоматизації підвищується продуктивність, якість продукції, безпека виробництва та знижуються витрати.

Основні поняття автоматики.

*Об'єкт керування* - фізичний процес або пристрій, на який спрямований вплив системи керування.

*Алгоритм керування* - сукупність правил, за якими здійснюється керування об'єктом.

*Зворотний зв'язок* - механізм, що забезпечує порівняння фактичного стану об'єкта з заданим і формування сигналу помилки.

Алгоритм в контексті систем автоматичного керування (САК) – це чітко визначена послідовність математичних операцій, що описує процес перетворення вхідних сигналів у вихідні з метою досягнення бажаного стану системи. Математичні моделі, що описують алгоритми керування, можуть бути як лінійними, так і нелінійними, дискретними або неперервними. Вибір типу моделі залежить від складності об'єкта керування та вимог до точності керування.

Об'єкт керування – це динамічна система, на яку впливають зовнішні збурення і на яку необхідно впливати з метою досягнення заданих характеристик. Математичний опис об'єкта керування зазвичай представляється у вигляді диференціальних або різницевих рівнянь.

Система автоматичного керування (САК) – це замкнута система, що складається з об'єкта керування, датчиків, пристрою порівняння, керуючого пристрою та виконавчих механізмів. Головним завданням САК є підтримання вихідної величини об'єкта керування в заданому режимі, незважаючи на вплив зовнішніх збурень.

Принцип роботи САК.

Сигнали з датчиків, що характеризують стан об'єкта, порівнюються із заданим значенням. Різниця між цими сигналами (помилка) використовується для обчислення керуючого впливу, який подається на виконавчий механізм. Керуючий пристрій реалізує алгоритм керування, який визначає характер і величину цього впливу.

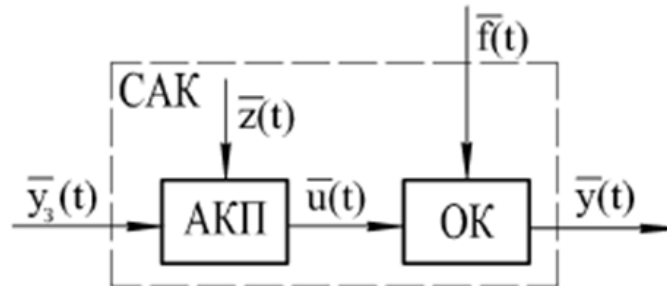


Рис. 1.5. Структура САК

У системах автоматичного керування (САК) керовані величини є вихідними параметрами об'єкта керування (ОК), які підлягають регулюванню. Це можуть бути фізичні величини, такі як температура, тиск, швидкість, або більш абстрактні параметри, які визначаються за допомогою математичних моделей. Мета керування полягає в тому, щоб підтримувати керовані величини на заданому рівні або змінювати їх за заданим законом, незважаючи на вплив зовнішніх збурень.

На керовані величини впливають різні чинники. Їх можна поділити на:

*Керуючі впливи:* це цілеспрямовані дії, які здійснюються на ОК з метою зміни його стану. Керуючі впливи формуються на основі інформації про бажаний стан системи та фактичному стані ОК.

*Збурюючі впливи:* це неконтрольовані зовнішні дії, які можуть виводити ОК з нормального режиму роботи. Збурення можуть бути випадковими або детермінованими.

Принцип зворотного зв'язку є одним з основних принципів побудови САК. Суть цього принципу полягає в тому, що для формування керуючого впливу використовується інформація про фактичний стан ОК. Тобто, порівнюється бажане значення керованої величини з її фактичним значенням, і на основі цієї

різниці (помилки) обчислюється керуючий вплив, який спрямований на зменшення помилки.

Способи формування керуючих впливів в системах автоматичного керування (САК) називають принципами керування. Сутність процесу керування полягає в оптимізації функціоналу якості, який характеризує відхилення фактичного стану системи від заданого. Для досягнення цієї мети використовується принцип зворотного зв'язку: керуючі дії формуються на основі постійного порівняння бажаного та фактичного станів системи.

Процес керування можна розглядати як цикл, що складається з таких етапів:

1. Сформулювання завдання керування: визначення бажаного стану системи та критеріїв якості.
2. Збір інформації: отримання даних про фактичний стан об'єкта керування за допомогою датчиків.
3. Обробка інформації: порівняння фактичного стану з бажаним та обчислення сигналу помилки.
4. Прийняття рішення: обчислення необхідних керуючих дій на основі алгоритму керування.
5. Виконання рішення: вплив на об'єкт керування з метою зменшення помилки.

Замкнуті системи керування, в яких широко використовується принцип зворотного зв'язку, забезпечують високу точність, стійкість та адаптивність до змін зовнішніх умов. Це досягається завдяки постійному моніторингу стану системи та оперативній корекції керуючих впливів. Її найбільш загальний вигляд наведений на рис. 1.6.



Рис. 1.6. Структура САК при керуванні за відхиленням

У багатьох системах автоматичного керування (САК) керуючий вплив формується не прямо пропорційно до значення керованої величини, а визначається сигналом помилки. Сигнал помилки – це різниця між заданим (бажаним) значенням керованої величини та її фактичним значенням. Математично це можна записати як:

$$d(t) = uz(t) - y(t),$$

де:

$d(t)$  – сигнал помилки в момент часу  $t$ ;

$uz(t)$  – задане значення керованої величини;

$y(t)$  – фактичне значення керованої величини.

Керуючий вплив  $u(t)$  обчислюється за формулою:

$$u(t) = A_p * d(t),$$

де  $A_p$  – оператор перетворення (алгоритм керування), який визначає характер і величину керуючого впливу в залежності від сигналу помилки.

Переваги принципу керування за відхиленням:

- Універсальність. Цей принцип дозволяє враховувати вплив будь-яких збурень, що впливають на керовану величину.
- Стійкість. Системи з таким принципом керування, як правило, більш стійкі до зовнішніх впливів.

Недоліки принципу керування за відхиленням:

- Затримка реакції. Система починає реагувати на збурення лише після того, як воно призвело до зміни керованої величини.
- Можливість коливань. Через наявність зворотного зв'язку в системі можуть виникати коливання, які можуть погіршити якість керування.
- Похибка керування. У перехідних процесах може спостерігатися відхилення фактичного значення керованої величини від заданого.

На відміну від систем з зворотним зв'язком, в розімкнених системах автоматичного керування відсутній прямий зв'язок між фактичним значенням керованої величини і керуючим впливом.

Існує два основних типи розімкнутих систем:

- САК, що здійснюють керування за збурюючим впливом (рис.1.7). В таких системах керуючий вплив формується виходячи з відомої інформації про збурення, що діє на систему.
- САК, що здійснюють керування за задаючим впливом. В таких системах керуючий вплив формується виходячи лише з заданого значення керованої величини, без урахування фактичного стану системи.

Розімкнуті системи зазвичай мають простішу структуру і меншу вартість, але вони менш точні і менш стійкі до збурень, ніж системи з зворотним зв'язком.

Вибір принципу керування залежить від конкретних вимог до системи, таких як точність, швидкість реакції, стійкість, вартість тощо.

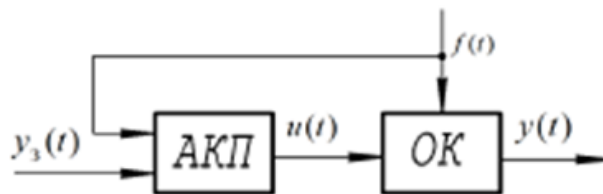


Рис. 1.7. Схема САК при керуванні за збурюючим впливом

З рисунка видно, що в системах, де керування здійснюється за збурюючим впливом, керуючий вплив  $u(t)$  обчислюється безпосередньо на основі інформації про величину збурюючого впливу  $f(t)$ . Тобто, система "передбачає" необхідну корекцію ще до того, як збурення встигне вплинути на керовану величину.

Переваги такого підходу - висока швидкість реакції. Система починає діяти проактивно, ще до появи значних відхилень керованої величини від заданого значення.

Недоліки такого підходу:

- Обмежена чутливість. Система реагує лише на той вид збурення, який було закладено в модель. Інші, непередбачені збурення можуть призвести до значних похибок.
- Високі вимоги до точності моделі. Для ефективного керування необхідно мати точну математичну модель зв'язку між збуренням і

керованою величиною. Будь-які неточності в моделі можуть знизити якість керування.

У системах, де керування здійснюється за задаючим впливом (рис.1.8), керуючий сигнал формується виключно на основі заданого значення керованої величини. Фактичний стан системи при цьому не враховується.

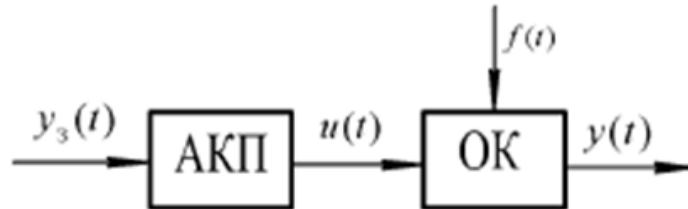


Рис. 1.8. Схема САК при керуванні за задаючим впливом

Сутність цього принципу полягає в тому, що керуючий сигнал формується виключно на основі заданого (бажаного) значення вихідної величини. Іншими словами, система "не знає" про фактичний стан об'єкта керування та не реагує на зовнішні збурення.

Переваги: такі системи мають просту структуру і не потребують складних алгоритмів керування.

Недоліки:

- Низька точність: системи чутливі до зовнішніх збурень, які можуть призводити до значних відхилень вихідної величини від заданого значення.
- Обмежена область застосування: ефективні лише для систем, де зовнішні збурення незначні або відсутні.

Для підвищення якості керування часто використовують комбінований принцип, що поєднує в собі елементи керування за відхиленням і за заданим впливом. Такий підхід дозволяє:

- швидко реагувати на зміни основних збурень, як у системах з розімкнутим керуванням;
- забезпечити високу точність керування, незалежно від причини відхилення вихідної величини, як у системах з зворотним зв'язком.

Робота системи автоматичного керування може відбуватися в двох основних режимах:

1. Статичний режим:

Вихідна величина системи є постійною в часі.

Система перебуває в рівновазі, тобто всі процеси в ній відбуваються стабільно.

Цей режим можливий лише за умови постійних зовнішніх впливів.

2. Динамічний режим:

Вихідна величина змінюється в часі.

Такий режим спостерігається під час перехідних процесів, коли система переходить з одного стану в інший після зміни зовнішніх впливів.

Динамічні режими можуть бути:

- Несталі: відбуваються відразу після зміни зовнішніх впливів.
- Сталі: встановлюються після закінчення перехідних процесів, коли вихідна величина змінюється в часі за тим самим законом, що і задаючий вплив.

Вибір принципу керування залежить від конкретних умов роботи системи, вимог до точності, швидкодії та інших характеристик. Для простих систем, де зовнішні збурення незначні, може бути достатнім принцип керування за заданим впливом. Для більш складних систем, де важлива висока точність і стійкість, доцільно використовувати комбіноване керування.

Системи автоматичного керування (САК) класифікують за різними ознаками, що відображають їхню структуру, функціональні можливості та сферу застосування.

***За призначенням:***

Системи стабілізації: підтримують заданий режим роботи об'єкта, нівелюючи вплив збурень.

Системи програмного керування: забезпечують зміну режимів роботи об'єкта за заданим законом у часі.

Системи спостереження: визначають значення певних параметрів об'єкта.

### ***За принципом дії:***

Розімкнуті САК: керуючий вплив формується без урахування фактичного стану об'єкта.

Замкнуті САК: керуючий вплив формується на основі порівняння заданого і фактичного значень вихідної величини.

Комбіновані САК: поєднують елементи розімкнутих і замкнутих систем.

### ***За здатністю до адаптації:***

Неадаптивні САК: параметри системи залишаються незмінними під час роботи.

Адаптивні САК: автоматично змінюють свої параметри для підтримання заданого режиму роботи в умовах змін зовнішніх впливів та властивостей об'єкта. Адаптивні системи поділяються на:

Екстремальні: змінюють лише керуючі впливи.

Самонастроювальні: змінюють керуючі впливи і параметри системи.

Системи, що самоорганізуються: змінюють керуючі впливи, параметри і структуру системи.

Системи, що навчаються: здатні змінювати свої характеристики на основі отриманої інформації.

### ***За характером сигналів:***

Безперервні САК: всі сигнали в системі є неперервними функціями часу.

Дискретні САК: сигнали квантуються за часом або рівнем.

### ***За видом математичного опису:***

Лінійні САК: описуються лінійними диференціальними рівняннями.

Нелінійні САК: описуються нелінійними диференціальними рівняннями.

### ***За величиною похибки в сталому режимі:***

Статичні САК: мають ненульову постійну похибку в сталому режимі.

Астатичні САК: похибка в сталому режимі дорівнює нулю.

### ***За характером параметрів:***

Стаціонарні САК: параметри системи не змінюються з часом.

Нестаціонарні САК: параметри системи змінюються з часом.

### ***За кількістю керованих величин:***

Одномірні САК: мають одну керовану величину.

Багатомірні САК: мають декілька керованих величин.

### ***За джерелом енергії:***

Системи прямої дії: використовують енергію керованого об'єкта.

Системи непрямої дії: використовують додаткове джерело енергії для створення керуючого впливу.

### **Висновки до розділу**

Проведене дослідження відкриває нові можливості для використання 3D-друку в кулінарії. Найбільш підходящим рішення для кухні в приватному будинку, квартирі є рішення яке повинне вибудовуватися на 3Д принтері Pancakebot. Але потрібно значно його модернізувати, а саме зменшити розмір рухомого стола, що дозволить зробити його рухомих, а портал з розміщеним на ньому контейнером з розчином для млинців зафіксувати. Це дозволить зменшити габарити пристрою так як його розміри вельми актуальні в разі зберігання на кухні.

Також зі складу пристрою необхідно виключити привод вертикального переміщення змінив його на настроюваний в ручну щілинний зазор та додати пристрій автоматичного знімання готових млинців з пристрою що дасть змогу забезпечити безперервне виготовлення млинців.

Модернізований 3D-принтер PancakeBot може стати основою для створення інноваційного кухонного гаджета, який не тільки автоматизує процес приготування млинців, але й дозволяє експериментувати з різноманітними рецептами.

Комбінований підхід до керування, що поєднує програмне керування та замкнений контур, забезпечить високу точність та стабільність роботи пристрою. Перспективним напрямком подальших досліджень є розробка алгоритмів адаптивного керування, а також інтеграція пристрою з іншими елементами "розумної кухні".

## РОЗДІЛ 2. РОЗРОБКА ПРИНЦИПОВОЇ КОНСТРУКТИВНОЇ СХЕМИ ПРИБАДУ ТА ПРИНЦИПОВОЇ СХЕМИ СИСТЕМИ КЕРУВАННЯ МЛИННИЦІ

### 2.1. Принципова конструктивна схема та принцип функціонування приладу

Схема приладу розміщена на рис.2.1.

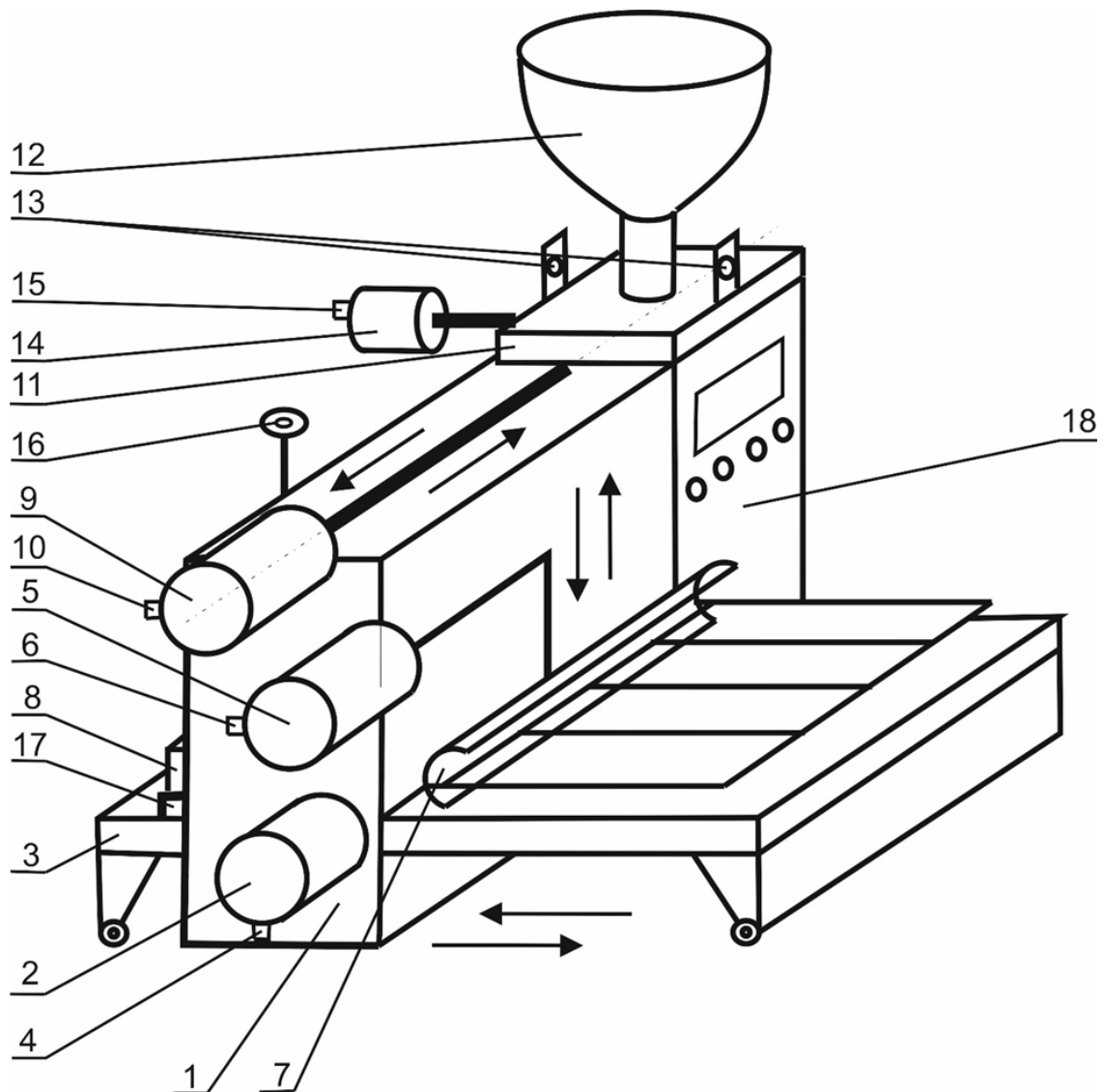


Рис. 2.1. Принципова конструктивна схема приладу млинниці

Млинниця складається з корпусу 1 (рис. 2.1.) На якому розміщені привід 2 переміщення поверхні для смаження 3, оснащений датчиком зворотного зв'язку 4, привід 5, оснащений датчиком зворотного зв'язку 6, переміщення скребка 7 механізму знімання готового млинця який за допомогою пружного зв'язку також

переміщує ніж 8 який формує необхідний зазор для забезпечення заданої товщини напівфабрикату на поверхні для смаження.

Також на корпусі розміщений привід 9 оснащений датчиком зворотного зв'язку 10, що забезпечує переміщення візка 11 на якому розміщений контейнер для млинцевого напівфабрикату 12, датчик наявності напівфабрикату 13, привід 14 відкриття сопла подачі млинцевого напівфабрикату оснащеного датчиком зворотного зв'язку 15. На корпусі розміщений ручний регулятор 16 положення упорів 17 ножа який регулює зазор і пульт управління 18 для включення пристрою і завдання програми управління.

Млинниця працює наступним чином.

Після установки (передбачається зберігати в розібраному вигляді для економії місця і збирати безпосередньо перед використанням, встановлювати поверхню для смаження на рівну строго горизонтальну плоску поверхню в крайнє праве положення) і підключення поверхні для смаження необхідно виставити рекомендований зазор, використовуючи ручний регулятор 16 і включити напругу.

За включення живлення треба вибрати програму необхідну для реалізації (ми розглянемо одну для випічки квадратного млинця) після цього перевіряється наявність млинцевого напівфабрикату в контейнері і при його наявності починає нагріватися панель для смаження, в якій встановлений інфрачервоний нагрівач і датчик температури на основі хромель-алюмелеві термопари (з аналізу, проведеного в першому розділі відомо що при температурі поверхні для смаження 220° С час приготування млинця 1 хв). Також по включенню всі приводи встановлюються в початкове положення (в якості приводів використовуються крокові двигуни) і потім по нагріванню поверхні для смаження до необхідної температури візок з контейнером для млинцевого напівфабрикату 12 починає переміщатися із заданою швидкістю у напрямку протилежного початкового стану при цьому за допомогою приводу 14 відкривається сопло для подачі напівфабрикату який розтікається смугою по поверхні для смаження до досягнення візком і відповідно соплом крайньої точки

при підході до якої сопло закривається і відбувається переміщення поверхні для смаження за допомогою приводу 2 в напрямку калібрувальної щілини сформованої площиною поверхні для смаження і ножа 8 після переміщення починається переміщення візка з відкриттям сопла в протилежному напрямку. Даний цикл повторюється до формування всієї поверхні млинця. Після чого відпрацьовує привід 5 при цьому ніж який згладжував нерівномірну викладку напівфабрикату за допомогою пружної зв'язку піднімається вгору забезпечуючи безперешкодний прохід поверхні для смаження з частково спеченого млинця в протилежному початковому напрямку. При цьому скребок 7 механізму знімання готового млинця також знаходиться в верхньому положенні.

Після досягнення вкрай правого положення поверхні для смаження з готовим млинцем скребок 7 опускається в крайнє нижнє положення при цьому ніж використовуючи пружну ланку повертається в початкове положення і лягає на упори 17.

Потім після перевірки наявності млинцевого напівфабрикату в контейнері повторюється цикл але при цьому в процесі руху поверхні для смаження до вкрай лівє положення відбувається з'їм готового млинця і укладання його на лоток з'єднаний з поверхні для смаження (на рис 2.1. Не вказано).

Робота пристрою триває поки є млинцевий напівфабрикат в контейнері.

## 2.2. Структурно-функціональна схема приладу

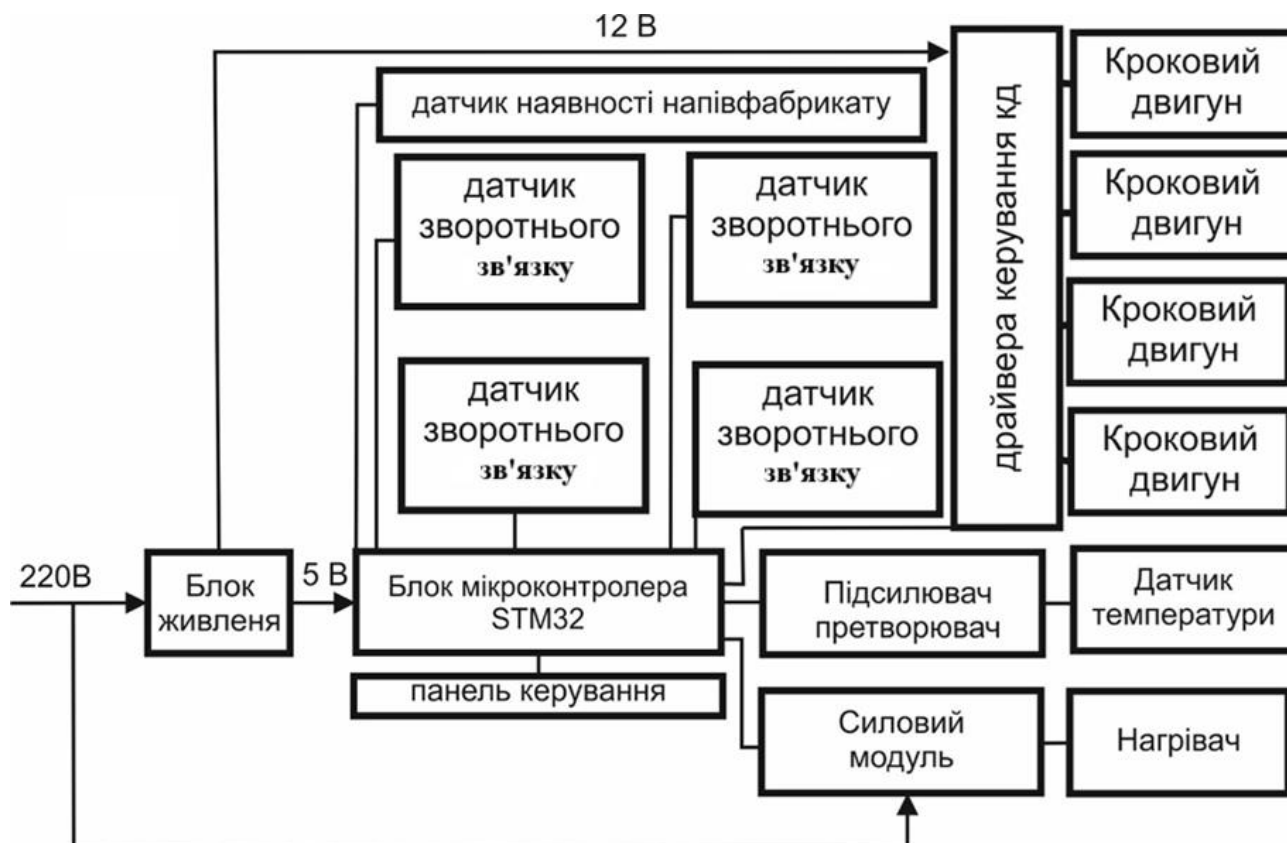


Рис. 2.2. Структурно-функціональна схема приладу млинниці

З рисунка 2.2 видно, що блок живлення повинен бути представлений електротехнічним пристроєм, що забезпечує перетворення вхідної змінної напруги 220 В в постійну стабілізовану напругу 5 В, необхідну для живлення мікросхем які входять до складу системи і в 12 В напруги для живлення крокових двигунів.

Блок живлення пристрою має забезпечувати перетворення змінної напруги мережі 220 В у постійні стабілізовані напруги 5 В та 12 В для живлення мікросхем. Для цього пропонується використовувати готовий трансформатор, який зменшить напругу до необхідних значень. Далі, для отримання стабілізованої напруги 5 В можна використати готовий імпульсний блок живлення на 9 В, а для отримання 12 В - лабораторний блок живлення. Такий підхід дозволить спростити конструкцію та знизити вартість пристрою.



Рис. 2.3. Трансформатор

Підсилювач перетворювач являє собою пристрій на основі операційного підсилювача OP07 і забезпечує коефіцієнт посилення близько 100-200, що дозволяє використовувати вбудований АЦП мікроконтролера.

Як датчик температури використовується термопара ТР-09 яка представляє собою мініатюрний температурний зонд типу К (хромель-алюмелеві) загального призначення, відкритий контакт (рис. 2.4).

Термопари широко застосовують для вимірювання температури різних об'єктів, а також в автоматизованих системах управління і контролю (таблиця 2.1). Вимірювання температур за допомогою термопар набуло широкого поширення через надійної конструкції датчика, можливості працювати в широкому діапазоні температур і дешевизни. Широкому застосуванню термопари зобов'язані в першу чергу своїй простоті, зручності монтажу, можливості вимірювання локальної температури. Вони набагато більш лінійні, ніж багато інших датчики, а їх нелінійність на сьогоднішній день добре вивчена і описана в спеціальній літературі. До достоїнств термопар відносяться також мала інерційність, можливість вимірювання малих різниць температур.

Термопари незамінні при вимірюванні високих температур (аж до 220<sup>0</sup>С) в агресивних середовищах. Термопари можуть забезпечувати високу точність вимірювання температури на рівні  $\pm 0,01$  ° С. Вони виробляють на виході термо ЕРС в діапазоні від мікрвольт до мілівольт, однак вимагають стабільного посилення для подальшої обробки.

Термопары відносяться до класу термоелектричних перетворювачів, принцип дії яких заснований на явищі Зеебека: якщо спаї двох різнорідних металів, що утворюють замкнутий електричний ланцюг, мають неоднакову температуру ( $T \neq T_2$ ), то в ланцюзі протікає електричний струм. Зміна знака у різниці температур спаїв супроводжується зміною напрямку струму. Під термоелектричним ефектом розуміється генерування термоелектрорухоючої сили (термоЕРС), що виникає через різницю температур між двома з'єднаннями різних металів і сплавів.






















Рис. 2.4. Термопара TP-09

Таблиця 2.1

**Основні типи термопар та їх параметри відповідно з міжнародною специфікацією**

Позначення ANSI	Матеріал термоелектродів		Діапазон робочих температур	Позначення за ГОСТ	Максимальна похибка	Колірне кодування
	Позитивний	Негативний				
<b>J</b>	Залізо (Fe)	Константан	-210 до +1200	-	$\pm 2,2$ °C або 0,75%	ANSI (США): Чорний
<b>K</b>	Хромель (Cr-Ni)	Алюмель (Ni-Al)	-270 до +1372	ТХА	$\pm 2,2$ °C або 0,75% вище 0°C	ANSI (США): Жовтий
<b>T</b>	Мідь (Cu)	Константан	-270 до +400	ТХК	$\pm 1$ °C або 1,5% нижче 0°C	ANSI (США): Синій
<b>E</b>	Хромель (Cr-Ni)	Константан	-270 до +1000	ТХК	$\pm 1$ °C або 1,5%	ANSI (США): Фіолетовий

Позначення ANSI	Матеріал термоелектродів		Діапазон робочих температур	Позначення за ГОСТ	Максимальна похибка	Колірне кодування
	Позитивний	Негативний				
N	Нікросил	Нісил (Ni-Si)	-270 до +1300	-	$\pm 2,2$ °C або 0,75%	ANSI (США): Оранжевий
R	Платина-родій (13% Rh)	Платина	-50 до +1768	ТПП	$\pm 1,5$ °C вище 800°C	ANSI (США): Білий
S	Платина-родій (10% Rh)	Платина	-50 до +1768	ТПП	$\pm 1,5$ °C вище 800°C	ANSI (США): Білий
B	Платина-родій (30% Rh)	Платина-родій (6% Rh)	0 до +1820	ТПР	$\pm 1,5$ °C вище 800°C	ANSI (США): Сірий
C	Вольфрам-рений (5% Re)	Вольфрам-рений (26% Re)	0 до +2320	ТВР	$\pm 5$ °C до +425°C; до +2320°C	ANSI (США): Жовтий

Позначення, ANSI	Тип за ГОСТ *	Максимальна похибка	Колірне кодування		
			ANSI США	МЭК 584-3	DIN 43710
J	—	2,2 °C або 0,75%			
K	ТХА	2,2 °C або 0,75% вище 0°C, 2,2 °C або 2% нижче			
T	—	1 °C або 0,75% вище 0°C, 1 °C або 1,5% нижче			
E	—	1,7 °C або 0,5% вище 0°C, 1,7 °C або 1% нижче			
N	—	2,2 °C або 0,75% вище 0°C, 2,2 °C або 2% нижче			—
R	—	1,5 °C або 0,25%	—		
S	ТПП	1,5 °C або 0,25%	—		
B	ТПР	0,5% понад +800 °C	—	—	
C	ТВР	4,5 °C до +425 °C, 1% до +2320 °C	—	—	—

Отже, термопара може бути пристроєм (або його складовою), що використовує термоелектричний ефект для вимірювання температури (рис. 2.45). У поєднанні з електровимірювальним приладом вона утворює термоелектричний термометр. Вимірювальний прилад або електронна вимірювальна система підключаються до кінців термоелектродів або в розрив одного з них. У місцях з'єднання провідників термопари з вимірювальною системою виникають додаткові термоЕРС. Внаслідок цього на вхід вимірювальної системи надходить сума сигналів від робочої термопари та від «термопар», що утворюються в місцях з'єднання. Основні характеристики термопар, що виробляються промисловістю, наведені в табл. 1 (ГОСТ 6616-94 «Перетворювачі термоелектричні»).

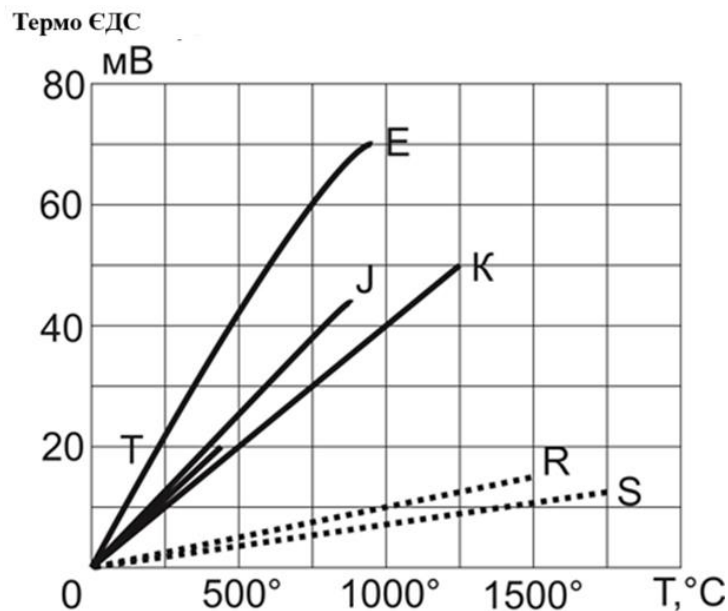


Рис. 2.5. Залежності ЕРС від температури за типами термопар

На рис. 2.5. представлені залежності ЕРС від температури найбільш поширених типів термопар, у яких температура холодного спаю підтримується рівною  $0^{\circ}\text{C}$ . З нього видно, що термопари типу E найбільш чутливі і розвивають найбільшу вихідну напругу при одному і тому ж зміні температури, ніж інші. З іншого боку, термопари типу S є найменш чутливими. На жаль, у більшості термопар ці залежності в деяких діапазонах температур носить нелінійний характер.

В якості приводу використаємо кроковий двигун SY42STH47-0406MB (Рис. 2.6)



Рис. 2.6. Кроковий двигун SY42STH47-0406MB

Характеристики двигуна представлені в таблиці 2.2

Таблиця 2.2

### Характеристики двигуна

Model No.		Rated Voltage	Current /Phase	Resistance /Phase	Inductance /Phase	Holding Torque	# of Leads	Rotor Inertia	Weight	Detent Torque	Operating Curve	Length
Single Shaft	Double Shaft	V	A	Ω	mH	Kg-cm		g-cm <sup>2</sup>	kg	g-cm		mm
SY42STH33-0956MA	SY42STH33-0956MB	4	0.95	4.2	4	1.58	6	35	0.22	200	a	33
SY42STH33-0606MA	SY42STH33-0606MB	6	0.6	10	9.5						b	
SY42STH33-0316MA	SY42STH33-0316MB	12	0.31	38.5	33						c	
SY42STH33-1334MA	SY42STH33-1334MB	2.8	1.33	2.1	4.2	2.2	4	54	0.28	220	d	38
SY42STH38-1206MA	SY42STH38-1206MB	4	1.2	3.3	3.4	2.59	6				e	
SY42STH38-0806MA	SY42STH38-0806MB	6	0.8	7.5	6.7		6				f	
SY42STH38-0406MA	SY42STH38-0406MB	12	0.6	30	30						g	
SY42STH38-1684MA	SY42STH38-1684MB	2.8	1.68	1.65	3.2	3.3	4				h	
SY42STH47-1206MA	SY42STH47-1206MB	4	1.2	3.3	4	3.17	6	i				
SY42STH47-0806MA	SY42STH47-0806MB	6	0.8	7.5	10			j				
SY42STH47-0406MA	SY42STH47-0406MB	12	0.4	30	38			k				
SY42STH47-1684MA	SY42STH47-1684MB	2.8	1.68	1.65	4.1	4.4	4	l				

В якості драйверу пропонується використати драйве на базі L298N двійного мосту (рис. 2.7).

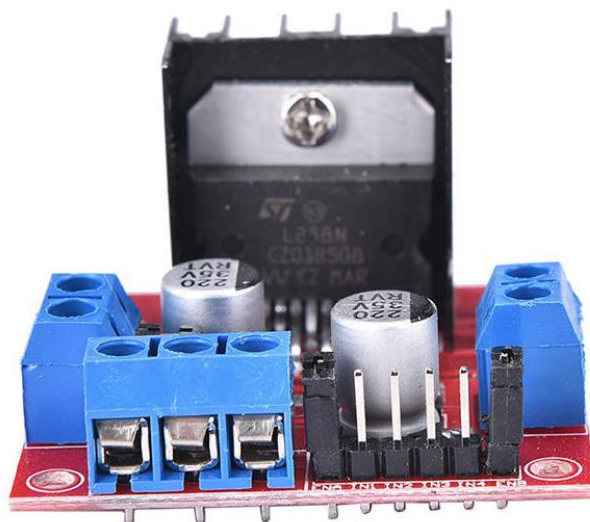


Рис. 2.7. Драйвер крокового двигуна на базі L298N двійного мосту

В якості блоку керування будемо використовувати RGB LCD екран 16x2 з клавіатурою для Raspberry Pi від Adafruit

Плата Adafruit Pi Plate призначена для роботи з RGB-екраном 16x2.

Платою можна управляти за допомогою всього двох I2C-пінів, і паралельно управляти 3-ма пінами підсвічування і 5-ма пінами клавіатури. Два I2C-Піна, про які йде мова, залишаються доступними, і до них можна підключити сенсор з I2C-інтерфейсом, модуль RTC і т.п. Це кращий спосіб позбутися від зайвих проводів.

Розробка плати стане в нагоді, якщо ви плануєте створити автономний проєкт із власним інтерфейсом користувача. Чотири кнопки у вигляді "стрілок" і кнопка "Вибір" забезпечують можливість управління пристроєм без використання комп'ютера.

Плата сумісна з Raspberry Pi Revision 1 і Revision 2, використовуючи I2C-піни SDA та SCL. У комплекті додається довгий 26-піновий перехідник, що підходить до роз'ємів USB і Ethernet.

Продукт поставляється у вигляді набору, який включає якісну друковану плату, компоненти для її складання (зокрема кнопки, перехідник тощо), а також LCD-матрицю RGB Positive 16x2. Процес складання настільки простий, що навіть новачок у пайці зможе впоратися з цим завданням за 30 хвилин.



Рис. 2.8. Набір для пайки

### Висновки до розділу

В даному розділі було проведено вивчення існуючих пристроїв класу який розробляється визначено вигляд пристрою і визначені його основні частини, розроблена конструктивна принципова схема і принципова схема управління.

## РОЗДІЛ 3. СИНТЕЗ ПРИНЦИПОВОЇ СХЕМИ ПРИСТРОЮ

### 3.1. Опис плати STM32F4 Discovery

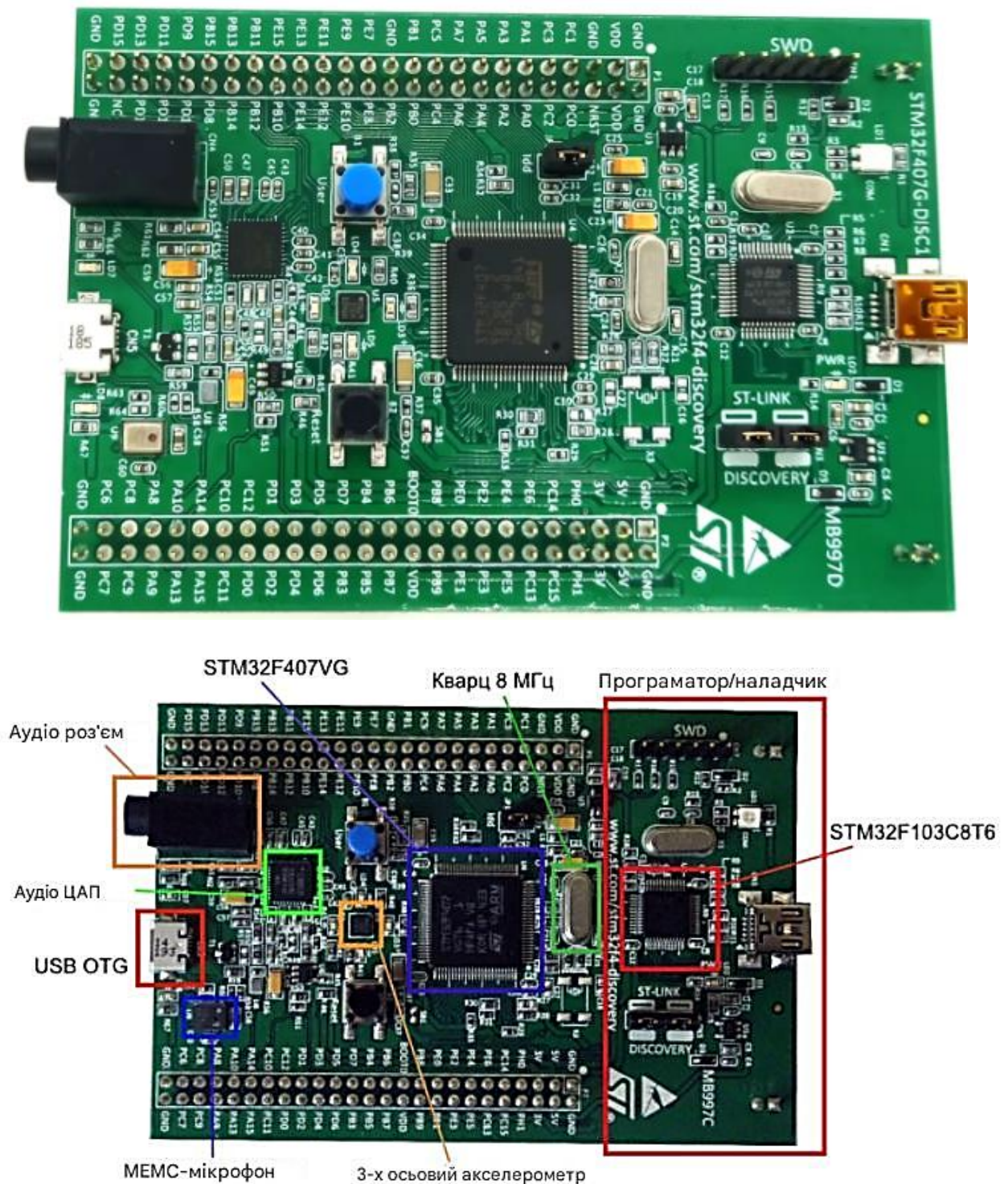


Рис. 3.1. Зовнішній вигляд плати STM32F4 Discovery

Плата STM32F4 Discovery є популярною платформою для розробки та прототипування електронних пристроїв на базі мікроконтролерів STM32F4. Вона пропонує широкий спектр можливостей для розробників різного рівня кваліфікації. Існує дві основні модифікації цієї плати: MB997C та MB997B,

причому перша є більш сучасною і замінила попередню версію завдяки використанню більш досконалого акселерометра LIS3DSH [3].

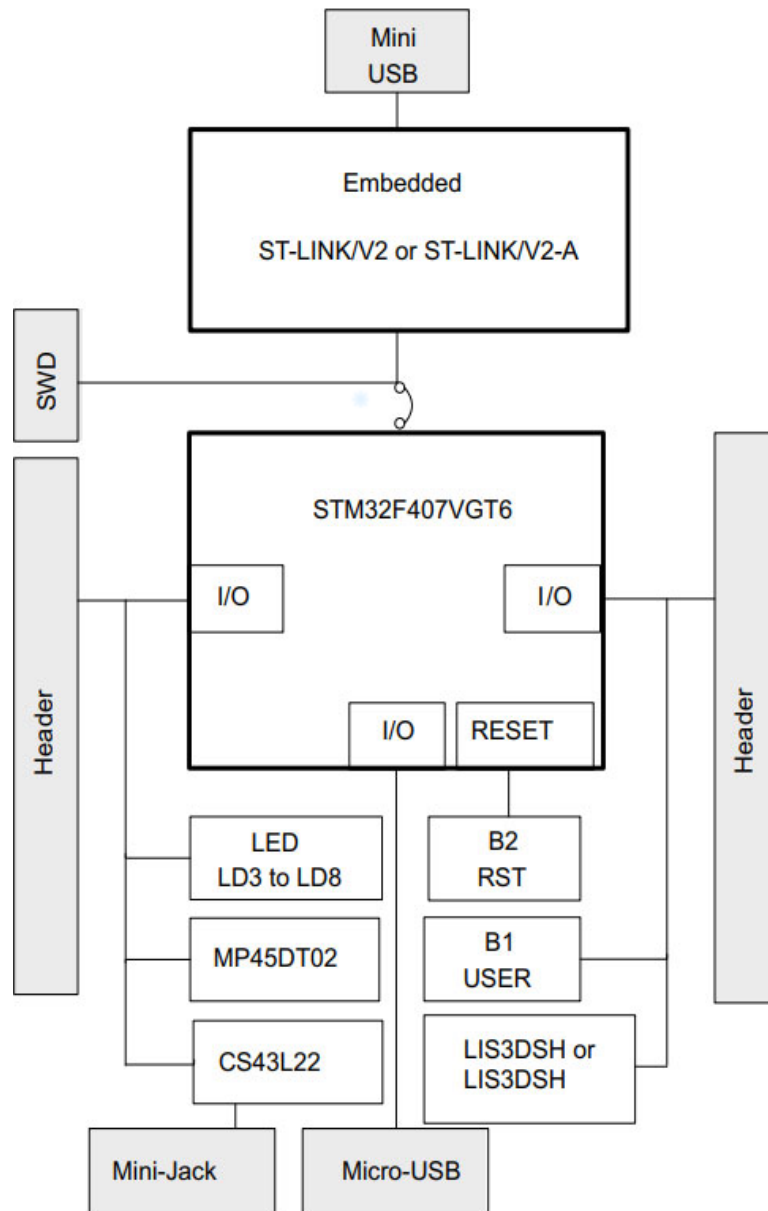


Рис. 3.2. Схема STM32F4DISCOVERY

Основні технічні характеристики плати:

- Мікроконтролер: 32-бітний STM32F407VGT6 з ядром ARM Cortex-M4F, який забезпечує високу продуктивність обчислень завдяки тактовій частоті 168 МГц та наявності вбудованого блоку для операцій з рухомою комою (FPU).
- Пам'ять: 1 МБ флеш-пам'яті для зберігання програмного коду та 193 КБ оперативної пам'яті для даних.

- Програматор: вбудований програматор ST-LINK/V2 дозволяє легко програмувати мікроконтролер та інші пристрої, підключені до плати.
- Периферія: плата оснащена різноманітними периферійними модулями, включаючи акселерометр, мікрофон, ЦАП, світлодіоди, кнопки та інтерфейс USB.
- Живлення: плата може житися від USB-порту комп'ютера або від зовнішнього джерела живлення.

Плата розробки STM32F4 Discovery оснащена потужним 32-бітним мікроконтролером STM32F407VGT6, що підтримує операції з числами з рухомою комою (FPU). Це суттєво прискорює обчислення, особливо в таких областях, як спектральний аналіз (ШПФ) та обробка сигналів в системах керування безпілотними літальними апаратами (БПЛА). Однак, відсутність роз'єму JTAG обмежує можливості низькорівневого тестування.

Структурна схема плати (рис. 3.3) демонструє, що напруга живлення +3 В не є ідеально стабілізованою через використання діода Шотткі. Максимальний струм споживання на цій лінії обмежений 150 мА. Живлення 5 В може подаватися як від USB-порту комп'ютера, так і від зовнішнього джерела. Завдяки діоду D1 забезпечується можливість одночасного використання обох джерел.

Програматор ST-Link, реалізований на мікроконтролері STM32F103C8T6, використовується для програмування як основного мікроконтролера STM32F407VGT6, так і зовнішніх мікросхем. Для перемикання між цими режимами використовуються джампери J2 і J3.

Для взаємодії з периферійними модулями, такими як акселерометр, мікрофон та ЦАП, мікроконтролер використовує шини SPI, I2C та I2S. Детальний розподіл портів наведено в таблиці 3.1.

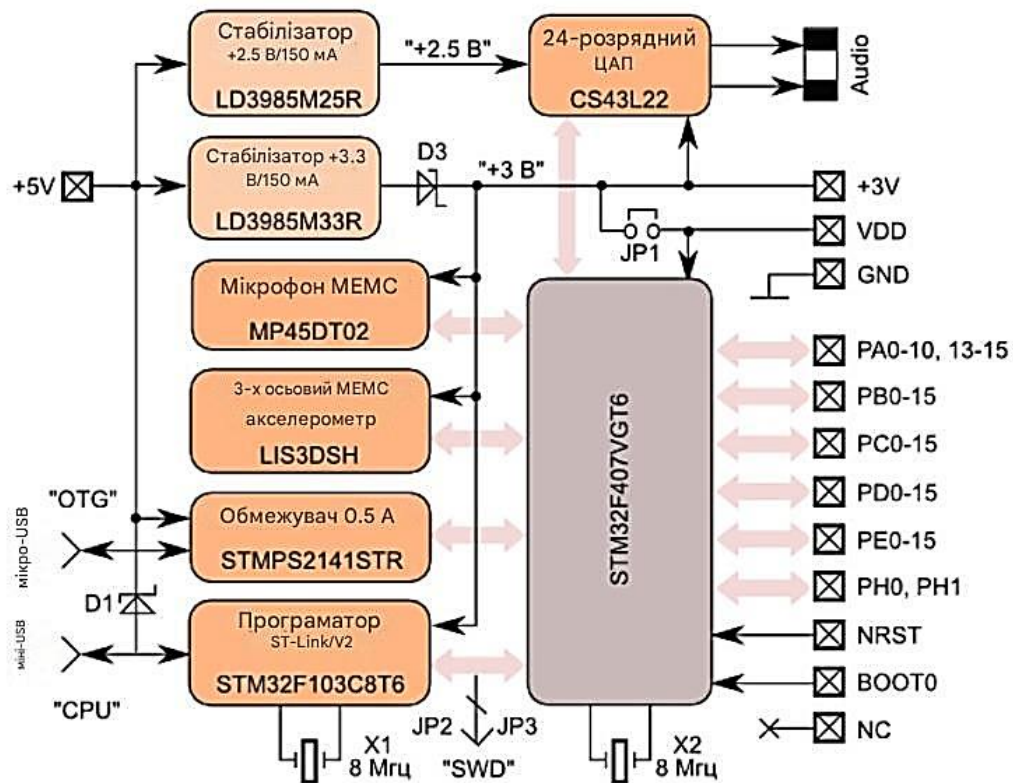


Рис. 3.3. Структурна схема оцінної плати STM32F4Discovery

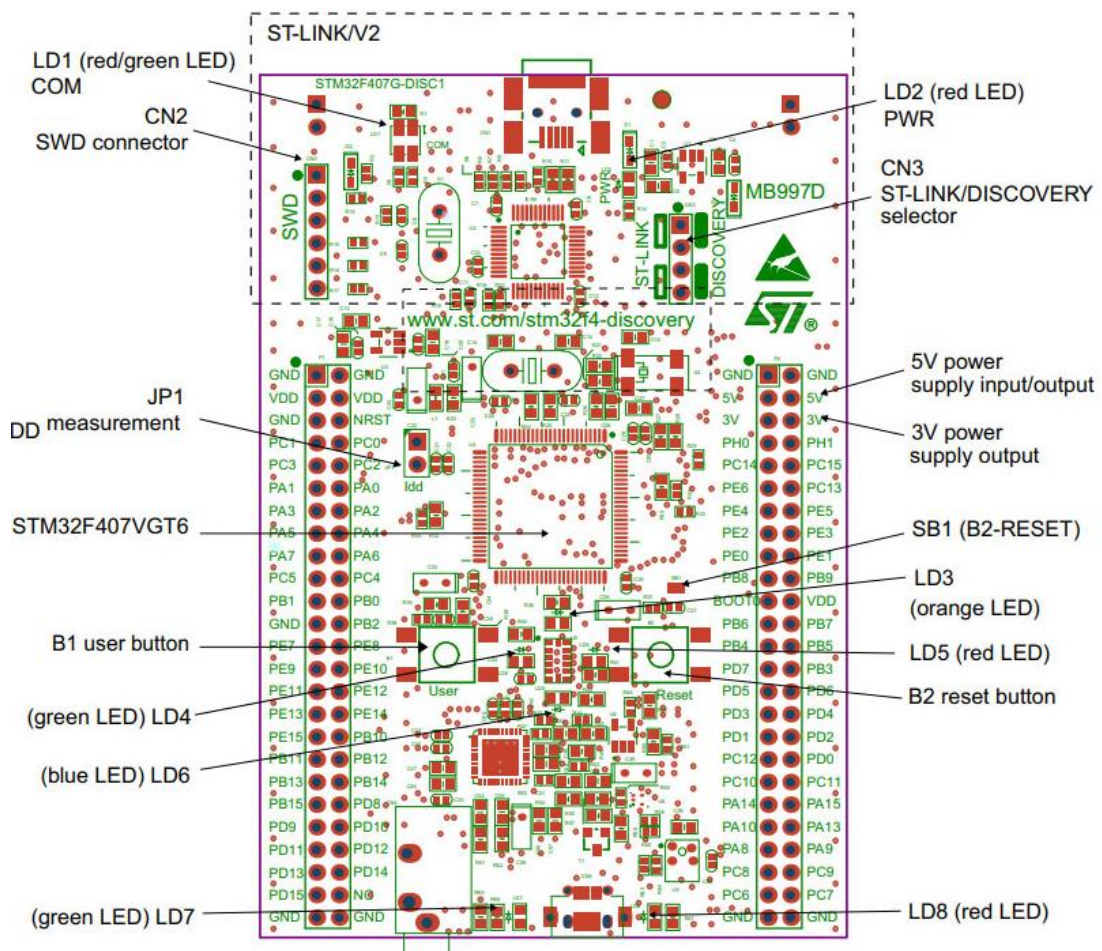


Рис. 3.4. Елементи плати та призначення контактів STM32F4DISCOVERY

## Опис портів мікроконтролера

Вивід порту	Функції порту
PA0	Кнопка «User»
BOOT0	Вхід бутлоудера, сигнал BOOT0
PB2	Вхід бутлоудера, сигнал BOOT1
PA1-PA3, PA8, PA15, PB0, PB1, PB4, PB5, PB7, PB8, PB11, PB13-PB15, PC1, PC2, PC4-PC6, PC8, PC9, PC11, PD0-PD3, PD6-PD11, PE2, PE4-PE15	Вільні лінії I / O, толерантні до 5 В, максимальне навантаження $\pm 25$ мА, pull-up / down резистори 30...50 кОм (всього 46 ліній)
PB12	Вільна лінія з pull-up/down резистором 8...15 кОм
PC13	Вільна лінія з навантаженням $\pm 3$ мА
PA3-PA6, PB6, PB9, PC7, PC10, PC12, PD4	Стереофонічний аудіо-ЦАП CS43L22
PA5-PA7, PE0, PE1, PE3	3-осьовий акселерометр LIS3DSH
PA9-PA12, PC0, PD5	Рознім мікро-USB (OTG)
PA13, PA14, PB3	Рознім програматора SWD
PB10, PC3	Вбудований цифровий мікрофон MP45DT02
PC14, PC15	Кварцовий резонатор 32 кГц (є місце)
PD12-PD15	Зелений, оранжевий, червоний, синій світлодіоди
PH0, PH1	Кварцовий резонатор 8 МГц для МК
NRST	Зовнішнє початкове скидання МК
+ 3 V, + 5 V, VDD, GND, NC	Ланцюги живлення 3 В, 5 В, МК, «земля», порожній контакт

Для перевірки працездатності плати STM32F4 Discovery підключіть її до комп'ютера за допомогою USB-кабелю. Переконайтесь, що засвітилися індикатори живлення та активності. Натисніть кнопку  $B_1$  для тестування акселерометра. Для програмування мікроконтролера використовуйте спеціалізоване програмне забезпечення, яке можна завантажити з офіційного сайту STMicroelectronics.

### 3.2. Принципова схема пристрою

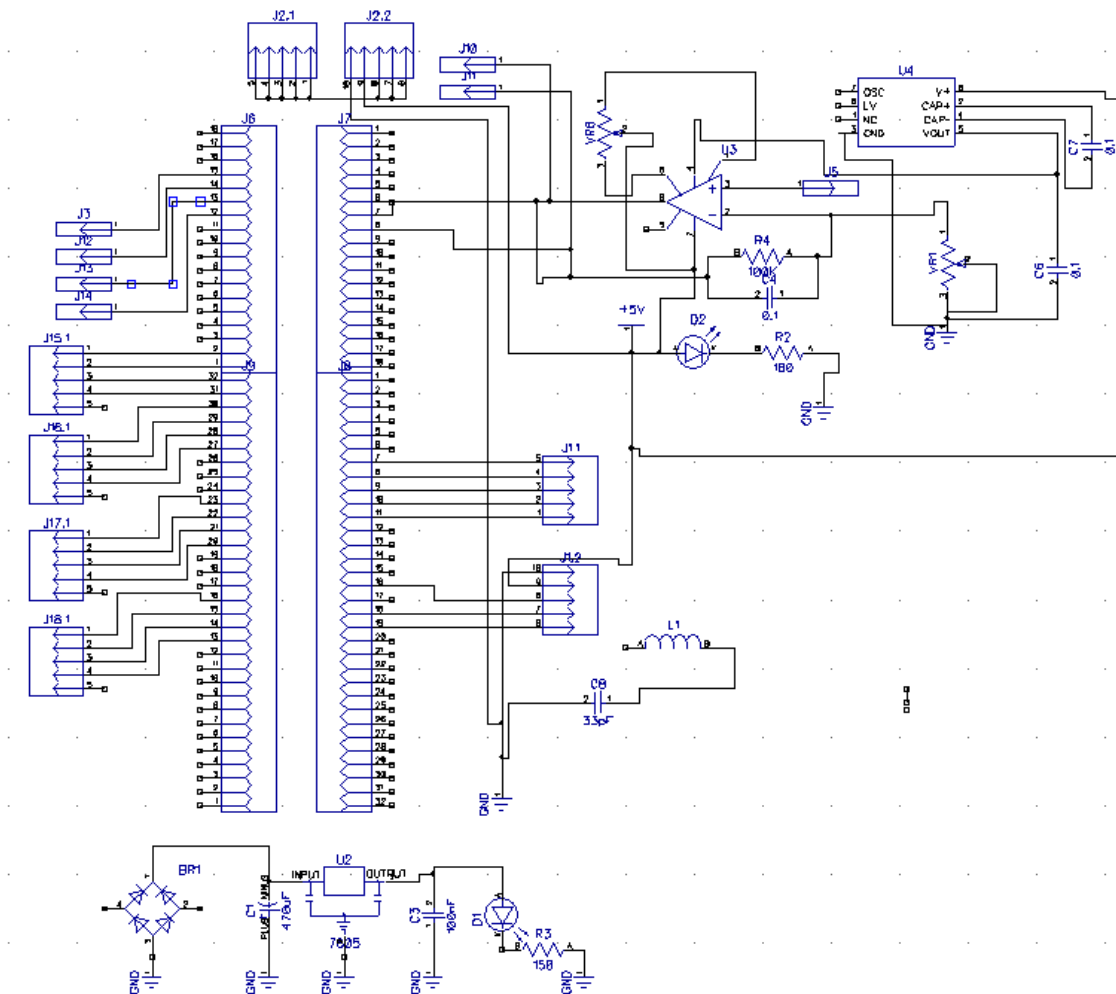


Рис. 3.5. Принципова схема пристрою

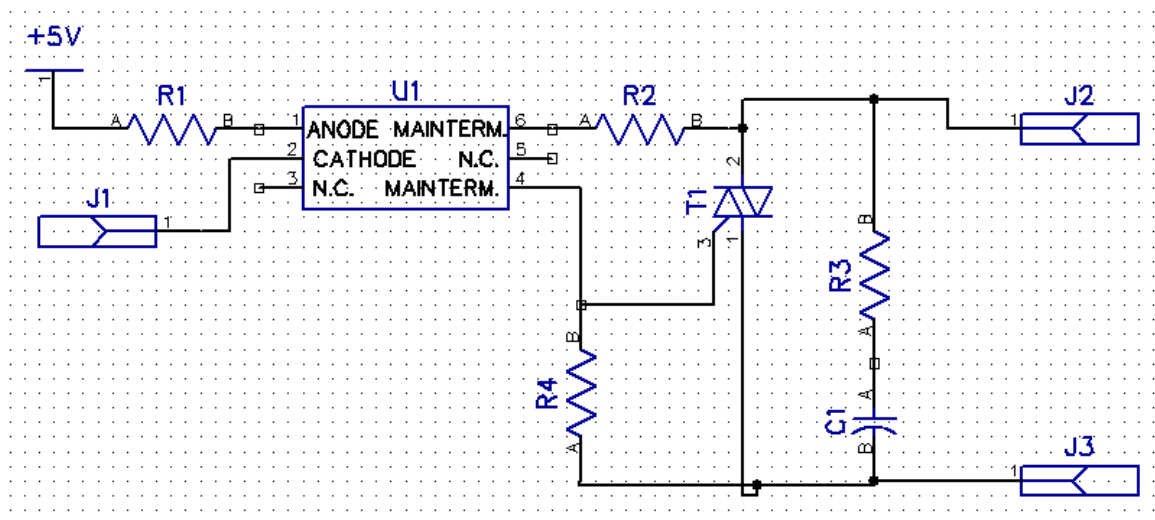


Рис. 3.6. Принципова схема силового модулю

Для конфігурації периферійних модулів та виводів мікроконтролера STM32 ми використаємо інтегроване середовище розробки STM32CubeMX.

STM32Cube – це безкоштовний набір програмних пакетів для операційної системи Windows, розроблений для роботи з мікроконтролерами та мікропроцесорами STM32 від компанії STMicroelectronics. Він підходить для користувачів, які вже працюють з такими середовищами розробки, як Keil або IAR, оскільки різні компоненти, наприклад, STM32CubeMX, STM32CubeProgrammer або STM32CubeMonitor, легко інтегруються в ці IDE.

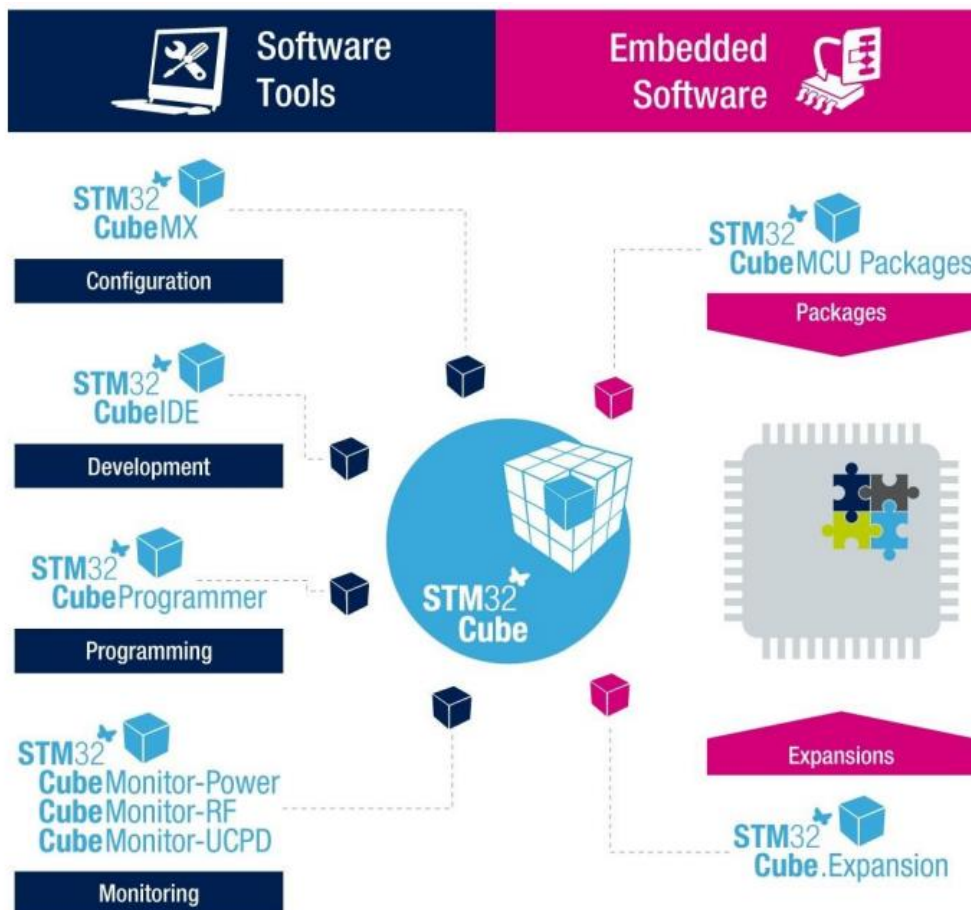


Рис. 3.7. Пакети STM32Cube

**STM32CubeMX** – програмний інструмент для конфігурації компонентів STM32, що дозволяє генерувати код мовою C для мікроконтролерів на основі ядра Cortex-M та пристроїв з ядром Cortex-A під Linux. Завдяки простому й функціональному графічному інтерфейсу його використання є зручним та інтуїтивним.

**STM32CubeIDE** – інтегроване середовище розробки, побудоване на основі відкритих рішень, таких як Eclipse і GNU C/C++. Воно забезпечує можливості компіляції коду та пропонує розширені функції для налагодження.

**STM32CubeProgrammer** – програмний інструмент для запису машинного коду в пам'ять компонентів STM32. Він підтримує функції читання, запису та перевірки коду через різні інтерфейси, включаючи JTAG, SWD, UART, USB DFU, I2C, SPI, CAN та інші.

**STM32CubeMonitor** – набір інструментів для моніторингу в реальному часі, що допомагає розробникам оптимізувати роботу та продуктивність своїх застосунків.

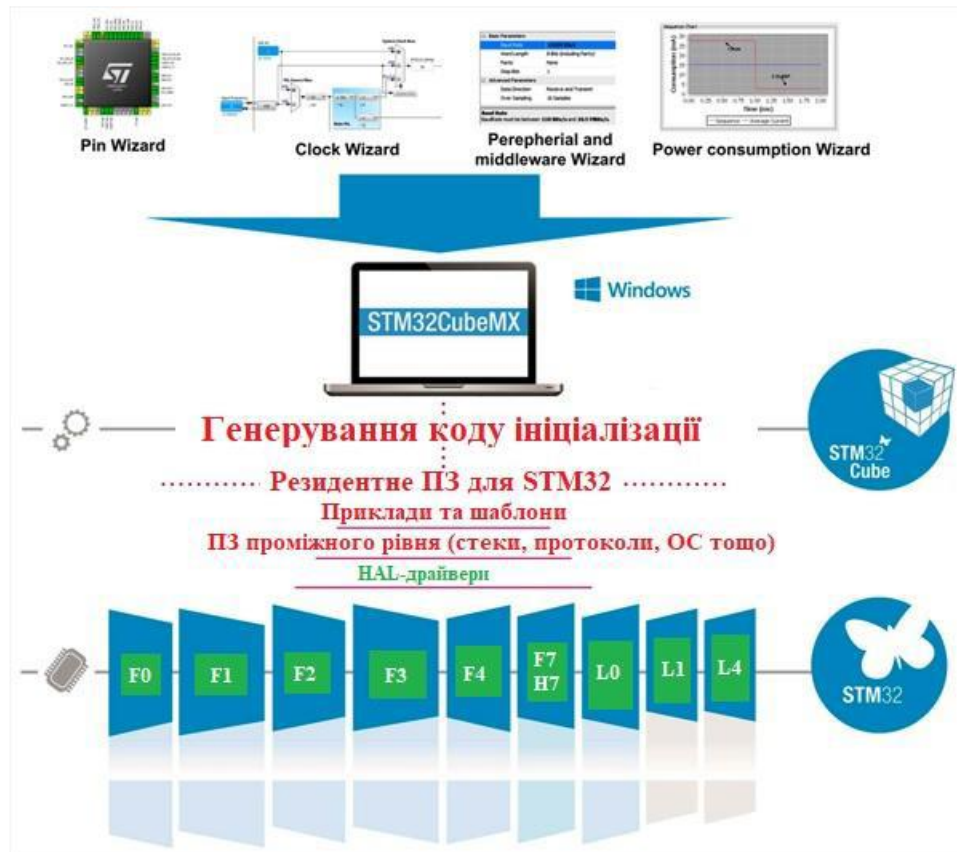


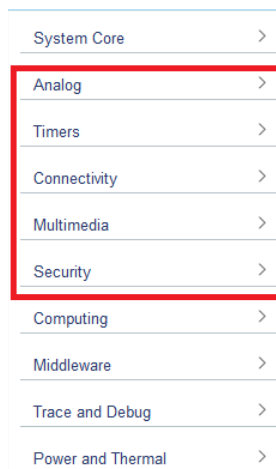
Рис. 3.8. Пакет *STM32CubeMX*

STM32CubeMX з графічним інтерфейсом надає можливість:

- обирати конкретний мікропроцесорний компонент STM32 із відповідним ядром, наприклад, серій F0–L4;
- визначати тип плати STBoard при використанні стандартних апаратних засобів налагодження:

- ☐ Discovery kit
- ☐ Evaluation Board
- ☐ Nucleo USB Dongle
- ☐ Nucleo-RF Kit
- ☐ Nucleo144
- ☐ Nucleo32
- ☐ Nucleo64

- налаштовувати та обирати інтегровані на кристалі модулі для роботи в мікропроцесорній системі відповідно до їхнього функціонального призначення (виділяються червоним кольором), а також інші вузли компонентів STM32:



- виконувати ініціалізацію системи синхронізації на основі заданої апаратної конфігурації (зовнішній сигнал синхронізації або тактовий генератор на базі кварцового резонатора потрібної частоти);
- конфігурувати основні лінії мікропроцесорного компонента STM32 у відповідності до обраних вузлів і їхнього функціонального призначення для мікропроцесорної системи;
- вибір середовища IDE, під яке STM32CubeMX згенерує код:



Рис. 3.9.Вибір середовища IDE

Після запуску програми користувачеві пропонується створити новий проєкт або імпортувати вже існуючий (див. рис. 3.10). Це дозволить нам

візуально налаштувати всі необхідні параметри мікроконтролера та згенерувати початковий код для подальшої розробки.

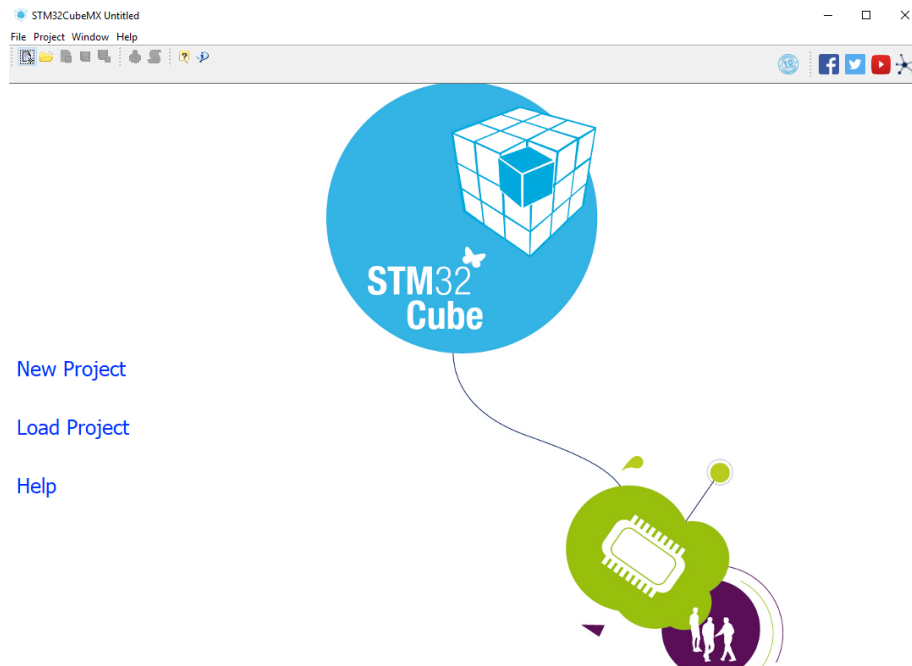


Рис.3.10. Вікно STM32CubeMX

Розпочнемо з створення нового проєкту в STM32CubeMX. Після цього відкриється вікно вибору мікроконтролера. За допомогою фільтрів знайдемо потрібний нам STM32F407VG (Рис. 3.11). У списку результатів буде відображено детальну інформацію про знайдені моделі. Вибравши потрібний мікроконтролер подвійним кліком, ми перейдемо до вікна конфігурації. Це вікно містить візуальне представлення мікроконтролера з усіма його виводами (Рис. 3.12).

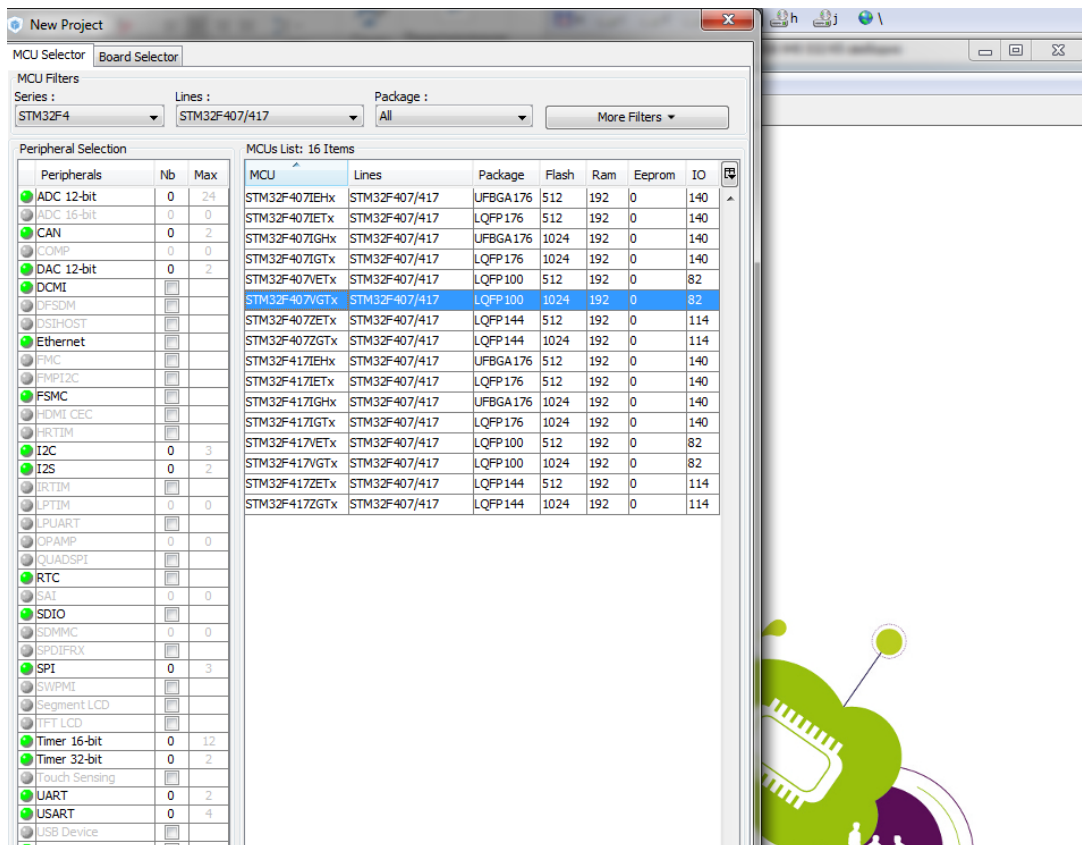


Рис. 3.11. Вибір МК у CubeMX

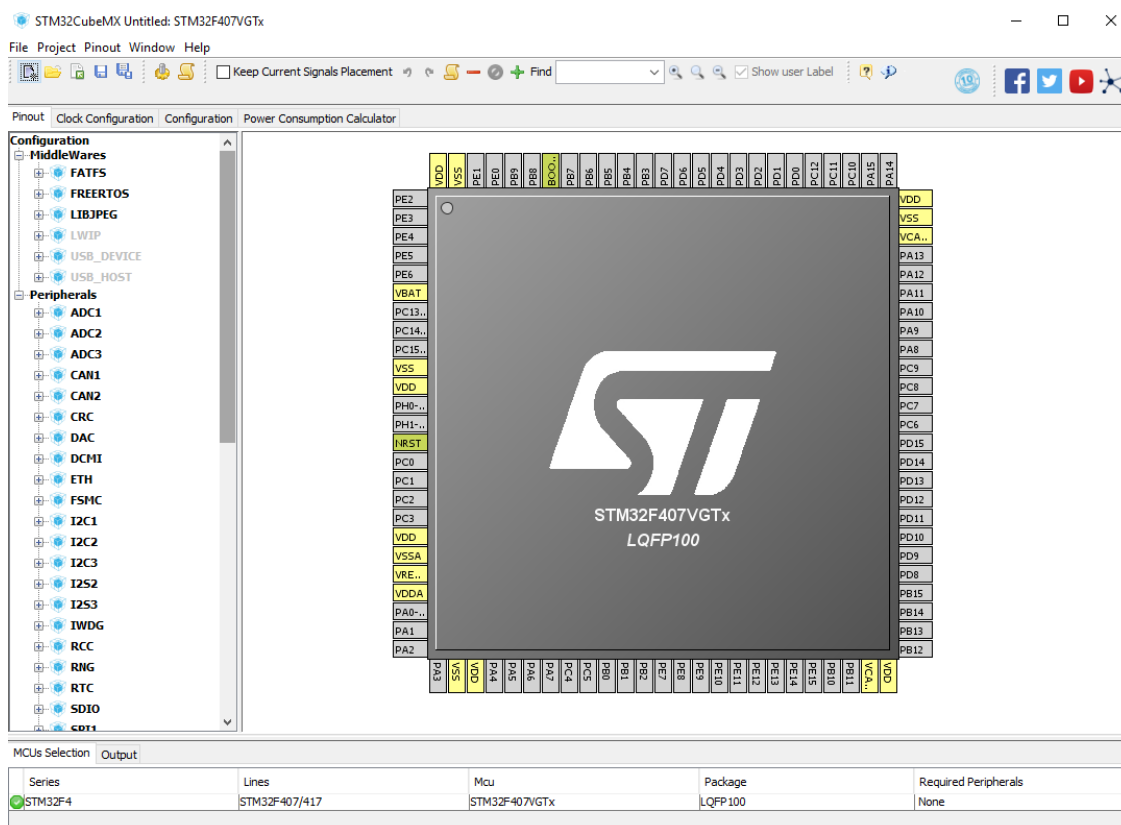


Рис. 3.12. Вікно конфігурації

Для збереження внесених змін до проєкту слід перейти у верхнє меню програми та обрати послідовно пункти "Проєкт" - "Налаштування" (див. рис.

3.13). Цей крок дозволить зберегти всі налаштування та конфігурації проєкту перед подальшою роботою.

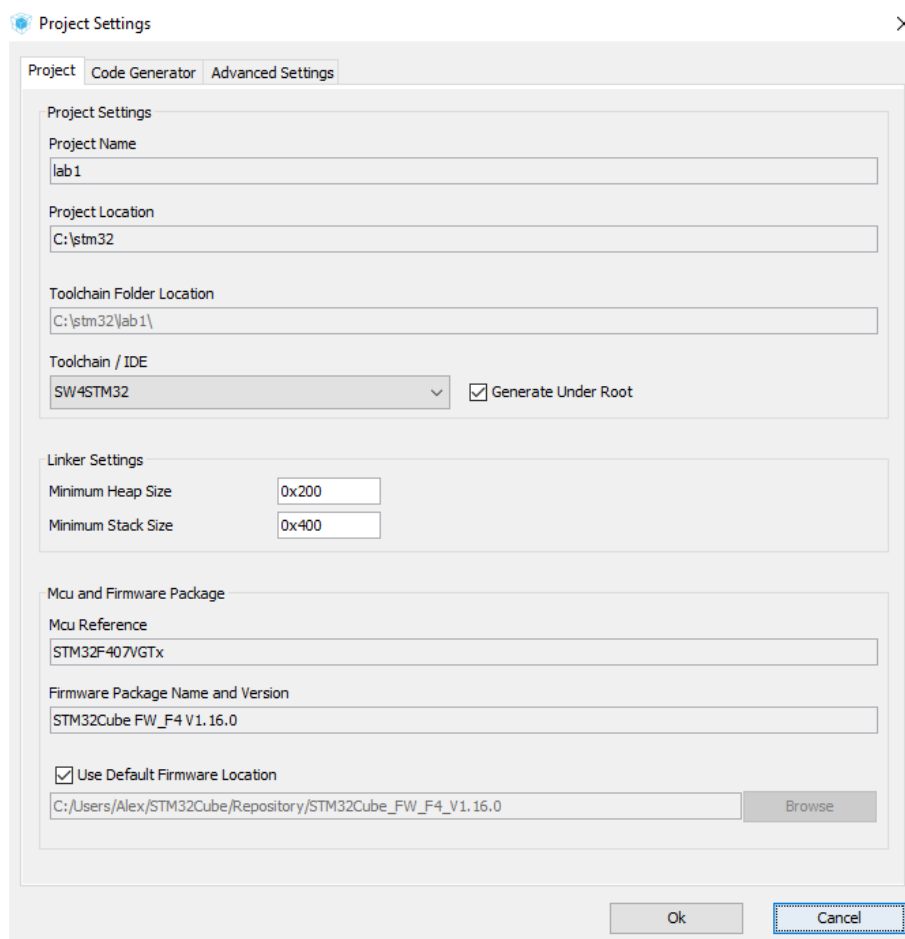


Рис. 3.13. Вікно налагодження проєкту

Спочатку збережемо наш проєкт. У верхньому меню обираємо "Проект" - > "Налаштування". У вікні, що з'явиться, вводимо назву проєкту "lab1" та вказуємо папку для збереження. Як середовище розробки вибираємо SW4STM32. Інші параметри залишаємо без змін. Після натискання "ОК" підтвердимо збереження у меню "Файл" -> "Зберегти проєкт". Далі перейдемо до налаштування тактового генератора. Відкриємо вкладку "Конфігурація годинника" (Clock Configuration). З'явиться діаграма, що відображає схему роботи тактового генератора. Тут ми налаштуємо необхідні ділильники та множники частоти, щоб отримати параметри, вказані на рис. 3.114.

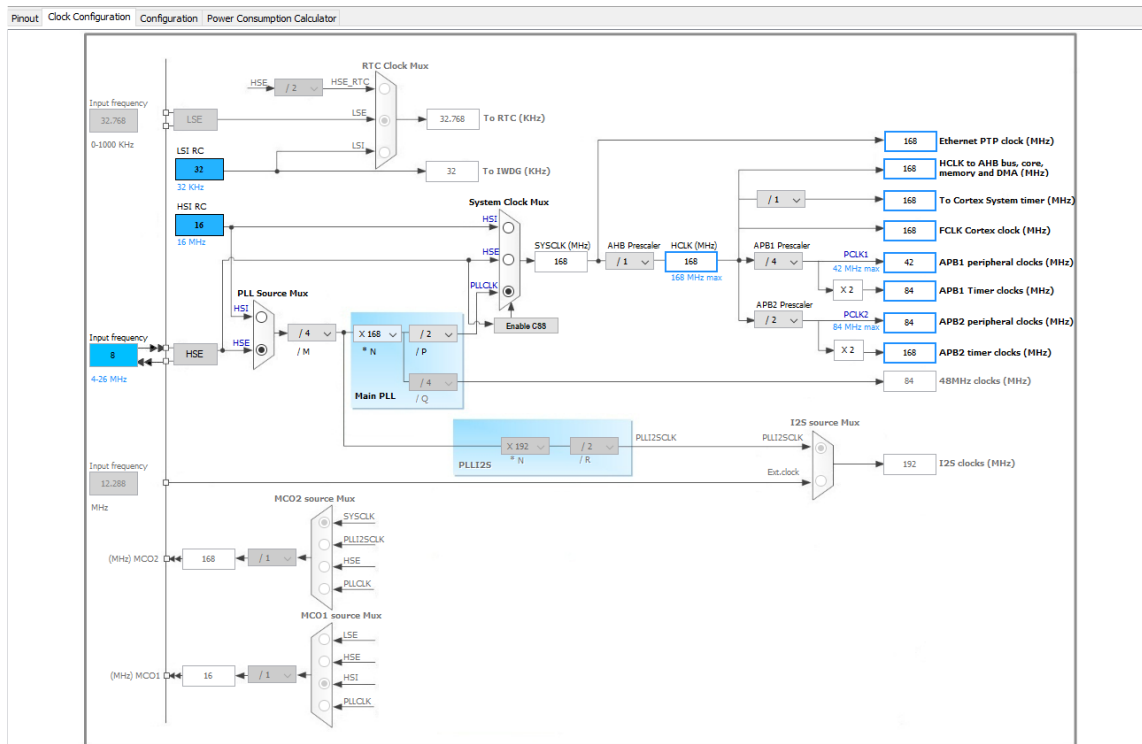


Рис. 3.14. Інтерактивна схема тактового генератора

Виводи PH-0 і PH-1 призначимо для підключення кварцового резонатора. Для цього у вікні конфігурації виберемо відповідні виводи та присвоїмо їм функції `RCC_OSC_IN` і `RCC_OSC_OUT` зі списку доступних функцій.

Таймер `TIM1` дозволить нам керувати шістьма каналами ШІМ. Виводи `PA8`, `PA9` і `PA10` налаштуємо як виходи каналів `TIM1_CH1`, `TIM1_CH2` і `TIM1_CH3` відповідно. Ці виводи керуватимуть верхніми ключами інвертора. Аналогічно, виводи `PB13`, `PB14` і `PB15` налаштуємо як інверсні виходи `TIM1_CH1N`, `TIM1_CH2N` і `TIM1_CH3N` для керування нижніми ключами.

Щоб активувати таймер `TIM1`, у дереві периферії знайдемо його і виберемо джерело тактування. Також активуємо канали 1, 2 і 3 (див. рис. 3.115). Для задання частоти вихідної напруги використаємо таймер `TIM3`.

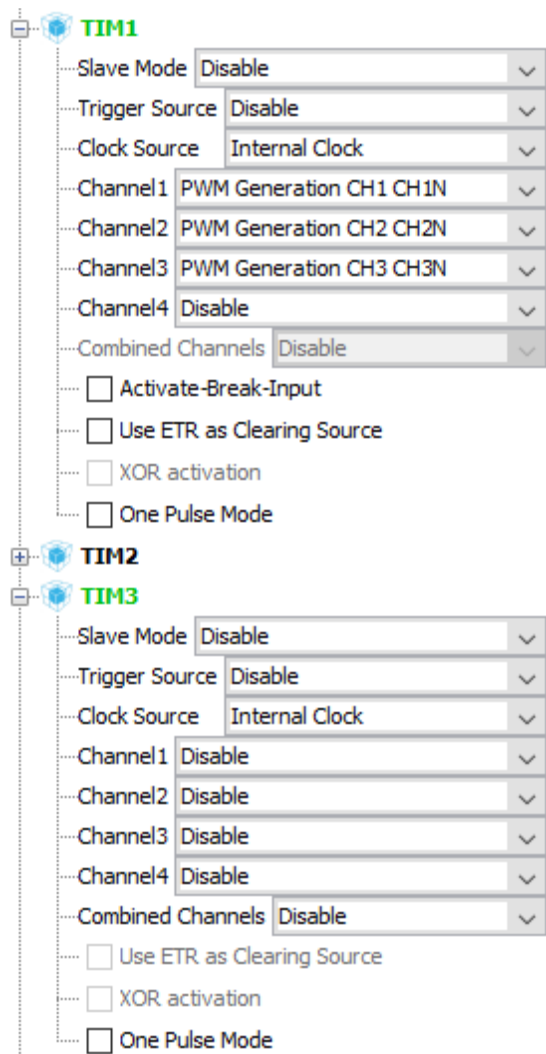


Рис. 3.15. Підключення таймеру

Вивід PA1 ми налаштуємо як вхід для аналого-цифрового перетворювача (АЦП), а саме ADC1\_IN1. До цього виводу під'єднаємо потенціометр, за допомогою якого будемо змінювати значення напруги, а отже, і частоту. Для більш детального налаштування таймерів та АЦП перейдемо на вкладку "Конфігурація" (рис. 3.16).

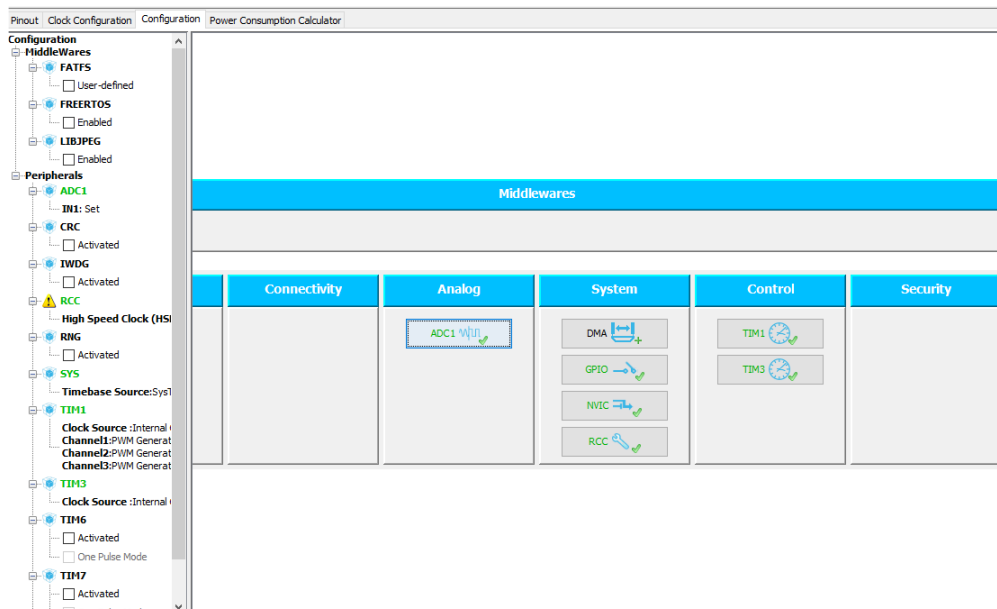


Рис. 3.16. Вкладка Configuration

Проведемо конфігурацію аналого-цифрового перетворювача (АЦП) відповідно до рис. 3.17. Встановимо такі параметри: режим роботи - незалежний, переддільник частоти - 4, розрядність - 12 біт, вирівнювання даних - вправо, а також увімкнемо безперервний режим перетворення. Решту параметрів залишимо без змін.

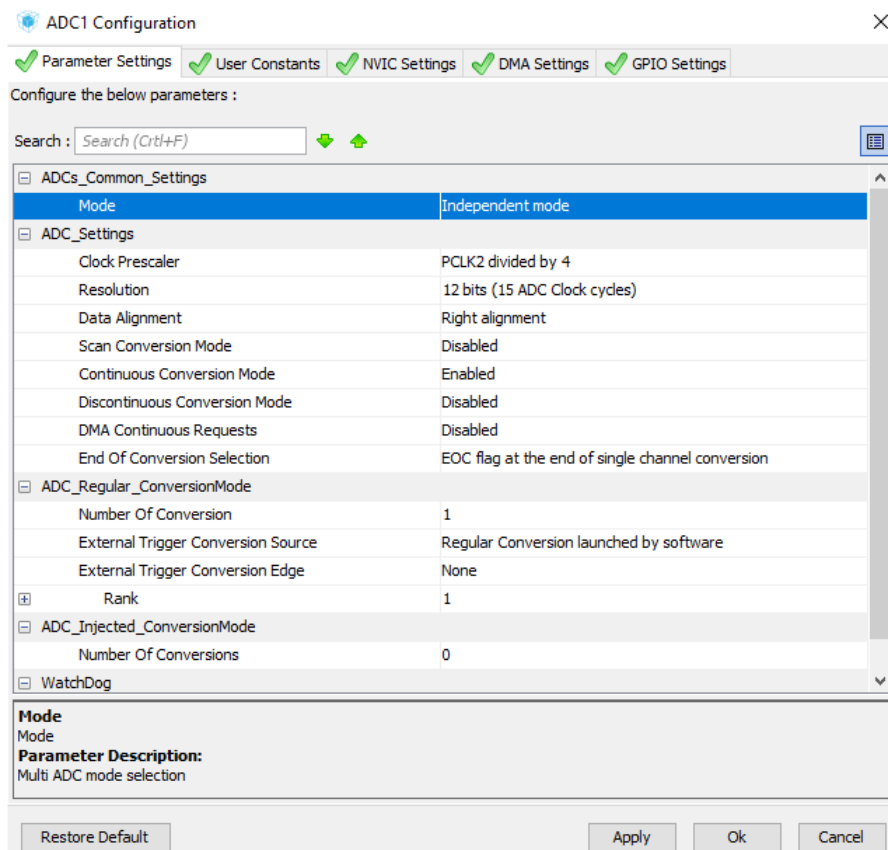


Рис. 3.17. Вікно налаштування АЦП

Наступним кроком налаштуємо таймер TIM3 відповідно до схеми на рис. 3.18. Встановимо такі параметри: переддільник частоти - 999, режим рахунку - вгору, а початкове значення лічильника (уставка) - максимальне, тобто 65535 (0xFFFF). Це значення ми будемо змінювати в процесі виконання програми. Для того, щоб отримувати переривання від таймера, на вкладці NVIC Settings активуємо відповідний прапорець.

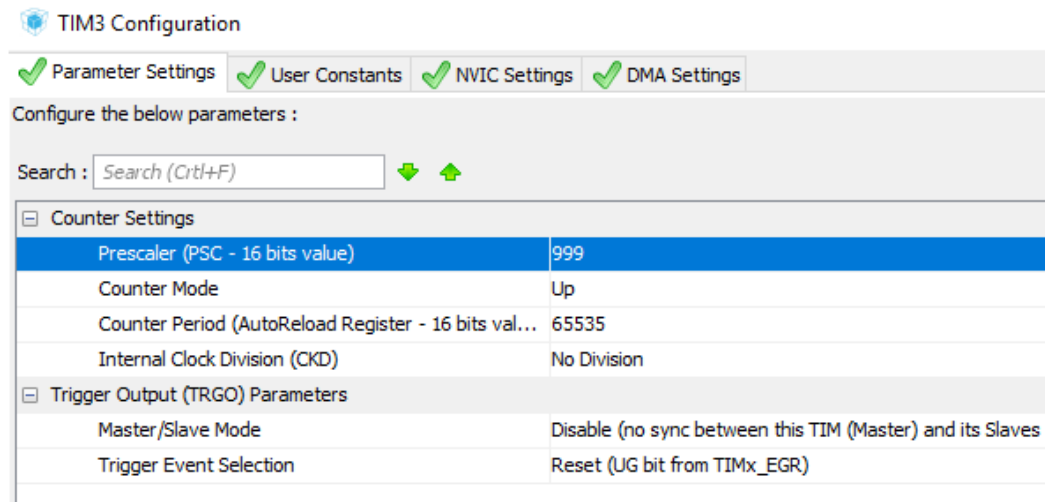


Рис. 3.18. Налаштування таймера TIM3

Перейдемо до налаштування таймера TIM1 відповідно до схеми на рис. 3.19. Встановимо такі параметри: переддільник - 0 (вимкнений), режим рахунку - вгору, максимальне значення лічильника (уставка) - 65535 (0xFFFF), тригер відключаємо, затримки перемикавання - 10 (120 нс). Виберемо режим ШІМ - прямий (PWM Mode 1), а параметр шпаруватості залишимо нульовим для подальшого змінення в програмі. Полярність сигналу встановимо як звичайну (логічна "1" відповідає високому рівню напруги). На вкладці налаштування GPIO для всіх виходів задамо максимальну частоту 50 МГц. Після завершення налаштування конфігурація в CubeMX повинна відповідати зображенню на рис. 3.20.



Щоб застосувати всі зміни, згенеруємо код проєкту та забриємо вікно налаштувань.

### 3.3. Підготовка проєкту у Keil mVision 5.20

Відкриємо проєкт у Keil mVision 5.20. Запустимо програму Keil і вкажемо шлях до папки, де ми зберегли проєкт, створений у CubeMX. Далі відкриємо файл проєкту за допомогою меню "Проєкт" -> "Відкрити проєкт". (див. рис. 3.21)

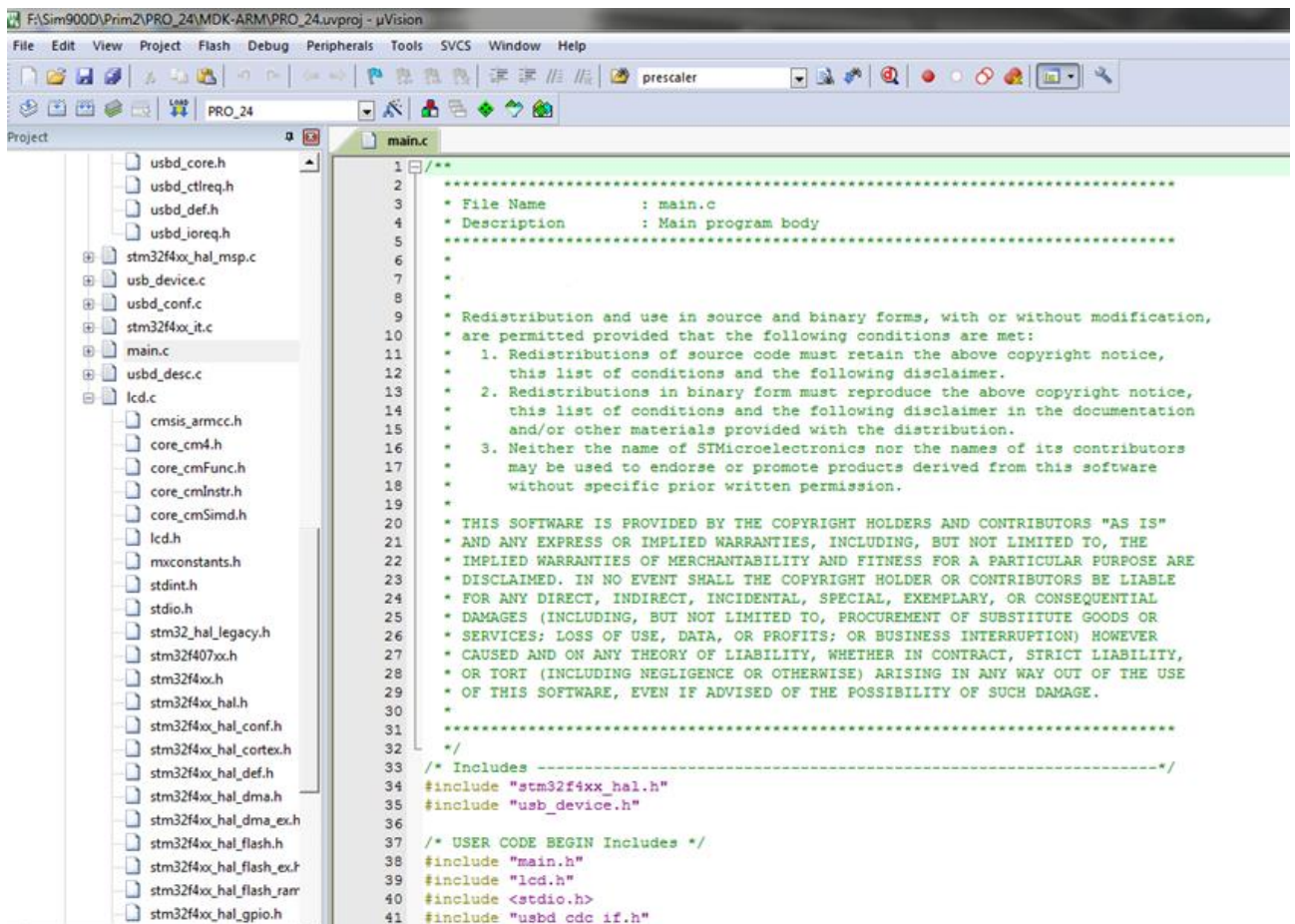


Рис. 3. 21. Вікно проєкту

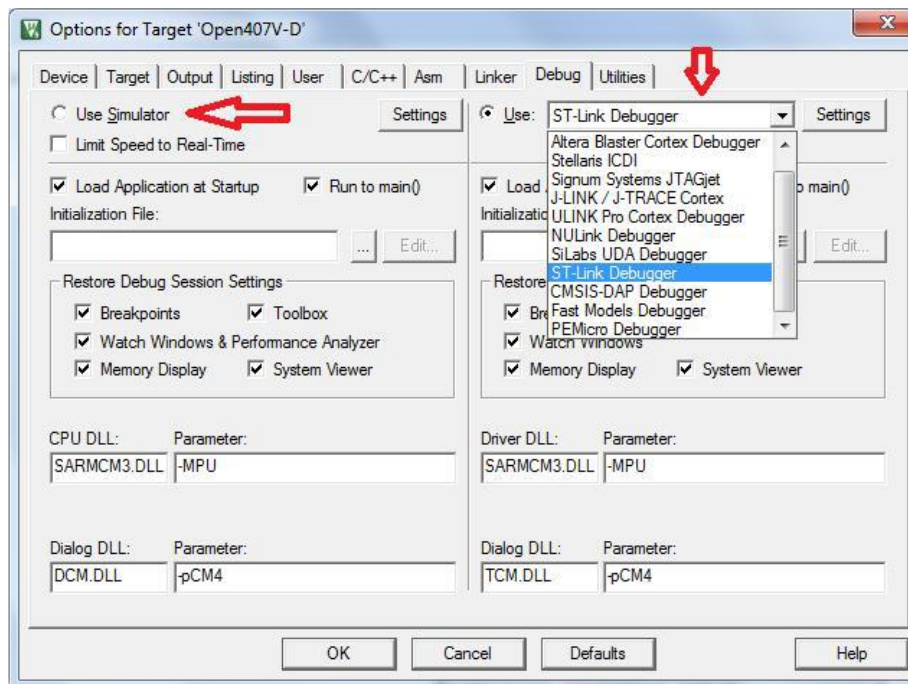


Рис. 3.22. Вікно вибору адаптера для програмування та відлагодження в Keil mVision

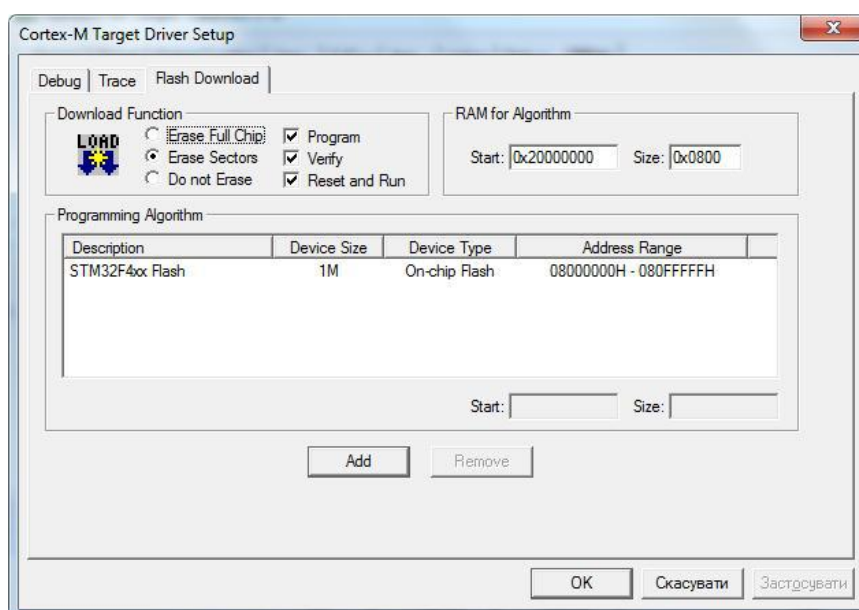


Рис. 3.23. Вікно конфігурації вбудованого програматора в Keil mVision

### Головна програма

Головна програма, представлена на блок-схемі (рис. 3.24), відповідає за керування всією системою. Вона написана мовою C і використовує бібліотеку HAL для взаємодії з периферією мікроконтролера STM32F4.

На початку програми підключаються необхідні заголовкові файли та оголошується глобальна змінна SINTABLE, яка містить попередньо розраховані значення синуса для трьох фаз.

Далі, у функції main виконується ініціалізація мікроконтролера, налаштовується тактовий генератор, АЦП, таймери TIM1 та TIM3. АЦП використовується для зчитування аналогового сигналу з потенціометра, а таймери - для генерації ШІМ сигналів, які керують інвертором.

У нескінченному циклі програма постійно зчитує значення з АЦП, масштабує його і встановлює нове значення уставки для таймера TIM3. Це призводить до зміни частоти вихідного сигналу інвертора залежно від положення потенціометра.

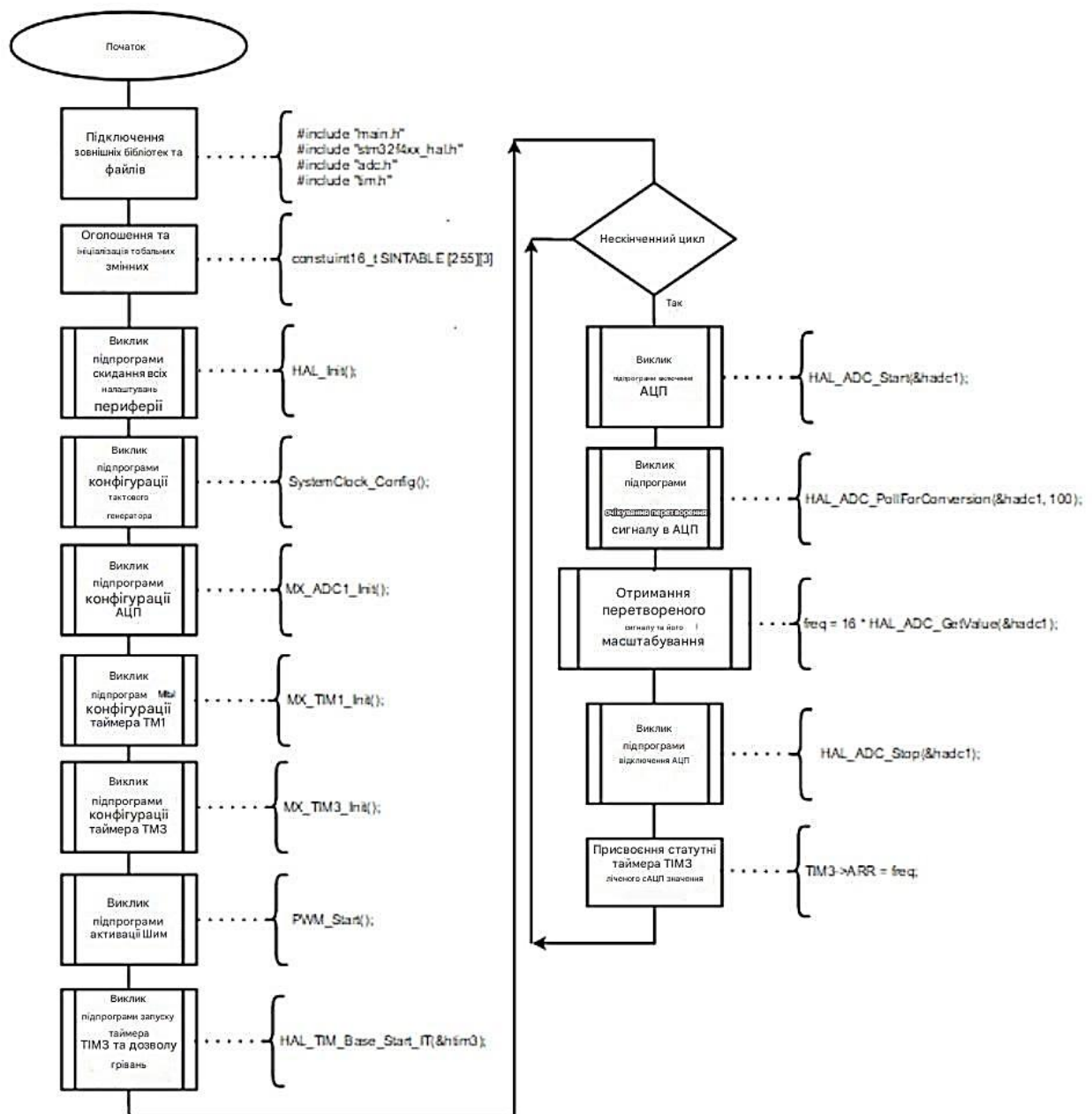


Рис. 3.24. Блок схема алгоритму програми

## Підпрограма конфігурації тактового генератора

На рисунку 3.25 зображено логічну схему функції `SystemClock_Config`, яка відповідає за налаштування тактового генератора мікроконтролера. Ця функція використовує дві структури даних: `RCC_OscInitTypeDef` та `RCC_ClkInitTypeDef`, визначені в бібліотеці HAL. Ці структури містять поля, які безпосередньо відповідають за конфігурацію регістрів мікроконтролера, що керують тактовими частотами.

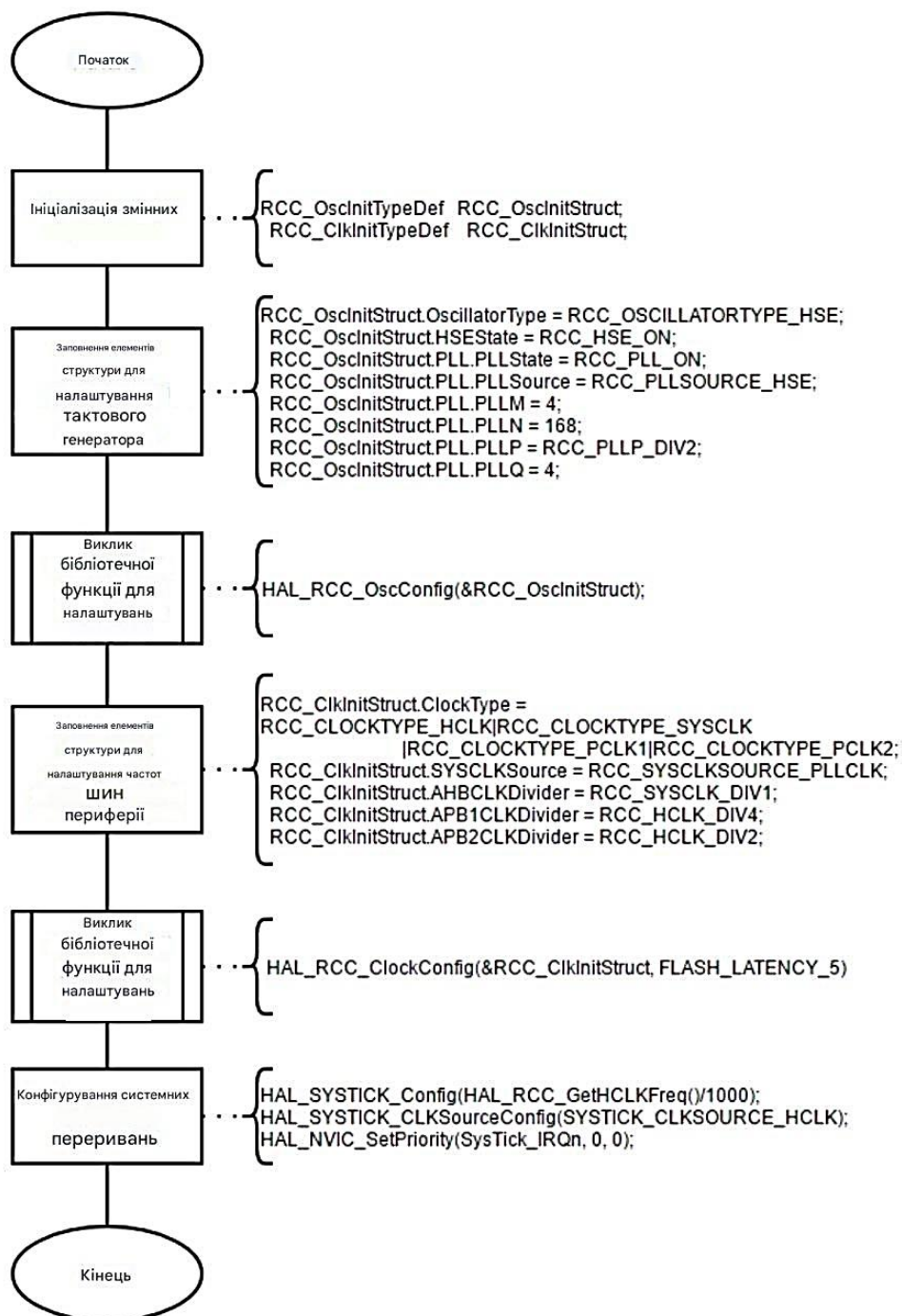


Рис. 3.25. Блок-схема підпрограми конфігурації тактового генератора

Після того, як ми задамо необхідні параметри в структурах, бібліотека автоматично оновить відповідні регістри мікроконтролера, звільнивши нас від ручного конфігурування регістрів.

### Підпрограма конфігурації АЦП

Мікроконтролери серії STM32F4 оснащені трьома 12-бітними аналого-цифровими перетворювачами (АЦП). Кожен з них здатний приймати вхідну напругу через спеціалізовані мультиплексовані входи та перетворювати її в цифровий код. Вхідна напруга порівнюється з опорною напругою (VREF), яка може надходити від аналогового живлення VDDA або використовуватися з інтегрованого чи зовнішнього джерела.

Додатково до зовнішніх входів, АЦП підтримує кілька внутрішніх каналів, таких як:

- сенсор температури (Vsense),
- внутрішня опорна напруга (Vrefint),
- моніторинг зовнішньої напруги живлення батареї (Vbat).

АЦП може працювати в різних режимах перетворення: одноканальному (single), багатоканальному (scan), безперервному (continuous) або переривчастому (discontinuous). Результати перетворення зберігаються у 16-розрядному регістрі з можливістю лівого або правого вирівнювання.

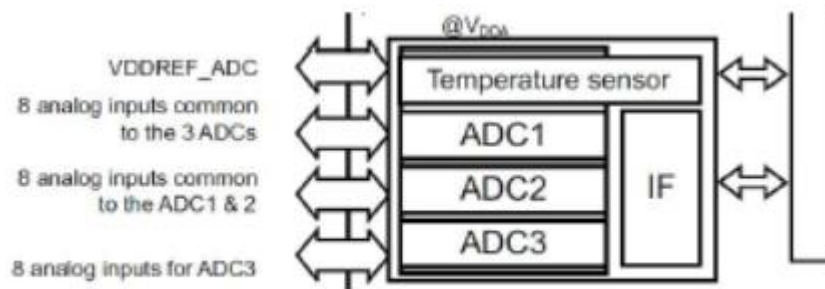


Рис. 3.26. АЦП мікроконтролера STM32F4

Також АЦП може виконувати функції аналогового сторожового таймера, генеруючи переривання, якщо вхідна напруга виходить за межі встановленого діапазону.

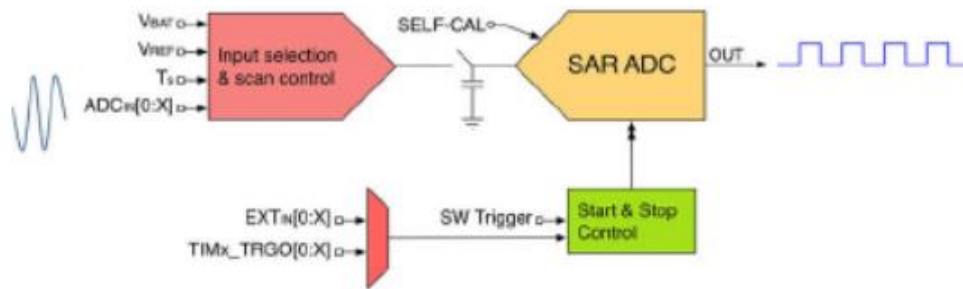


Рис. 3.27. Структура АЦП

Блок вибору каналу та контролю сканування (Input Selection & Scan Control) автоматично перемикає входні канали відповідно до обраного режиму роботи, забезпечуючи зчитування кожного з них у заданій послідовності. Виходи цього блоку підключаються до АЦП (SAR ADC).

Контроль запуску та зупинки перетворення здійснюється блоком Start & Stop Control. Цей блок може керуватися програмно або через події від зовнішніх джерел, а також через внутрішню лінію TRGO, яка пов'язана з таймерами. Це дозволяє реалізувати перетворення за таймером у режимі DMA.

Вхідні канали мікроконтролера закріплені за конкретними виводами (наприклад, IN0, IN1 тощо), однак їх послідовність вибірок може бути змінена логічно.

Перепризначення каналів здійснюється за допомогою присвоєння кожному каналу індексу в діапазоні від 1 до 16. Цей індекс, відомий як Rank у CubeHAL, визначає порядок перетворення каналів у послідовності регулярних каналів.

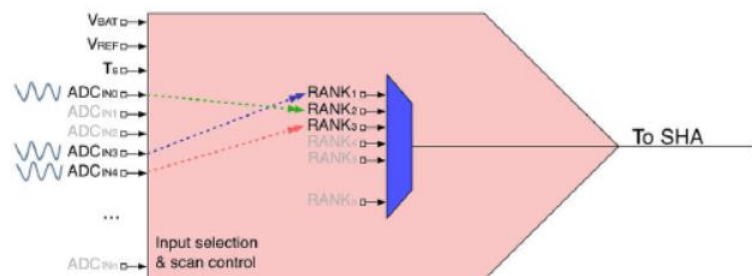


Рис. 3.28. Переназначення каналів через індекс RANK

На рисунку 3.29 зображено логіку роботи функції MX\_ADC1\_Init, яка відповідає за налаштування аналого-цифрового перетворювача (АЦП). Для цього використовуються три структури даних, визначені в бібліотеці HAL:

ADC\_HandleTypeDef, ADC\_ChannelConfTypeDef та GPIO\_InitTypeDef. Ці структури містять поля, які відповідають за різноманітні параметри конфігурації АЦП та його підключення до виводів мікроконтролера. Після того, як всі необхідні параметри будуть встановлені в цих структурах, викликаються спеціальні функції бібліотеки HAL, які записують ці значення безпосередньо в регістри мікроконтролера, таким чином конфігуруючи АЦП.

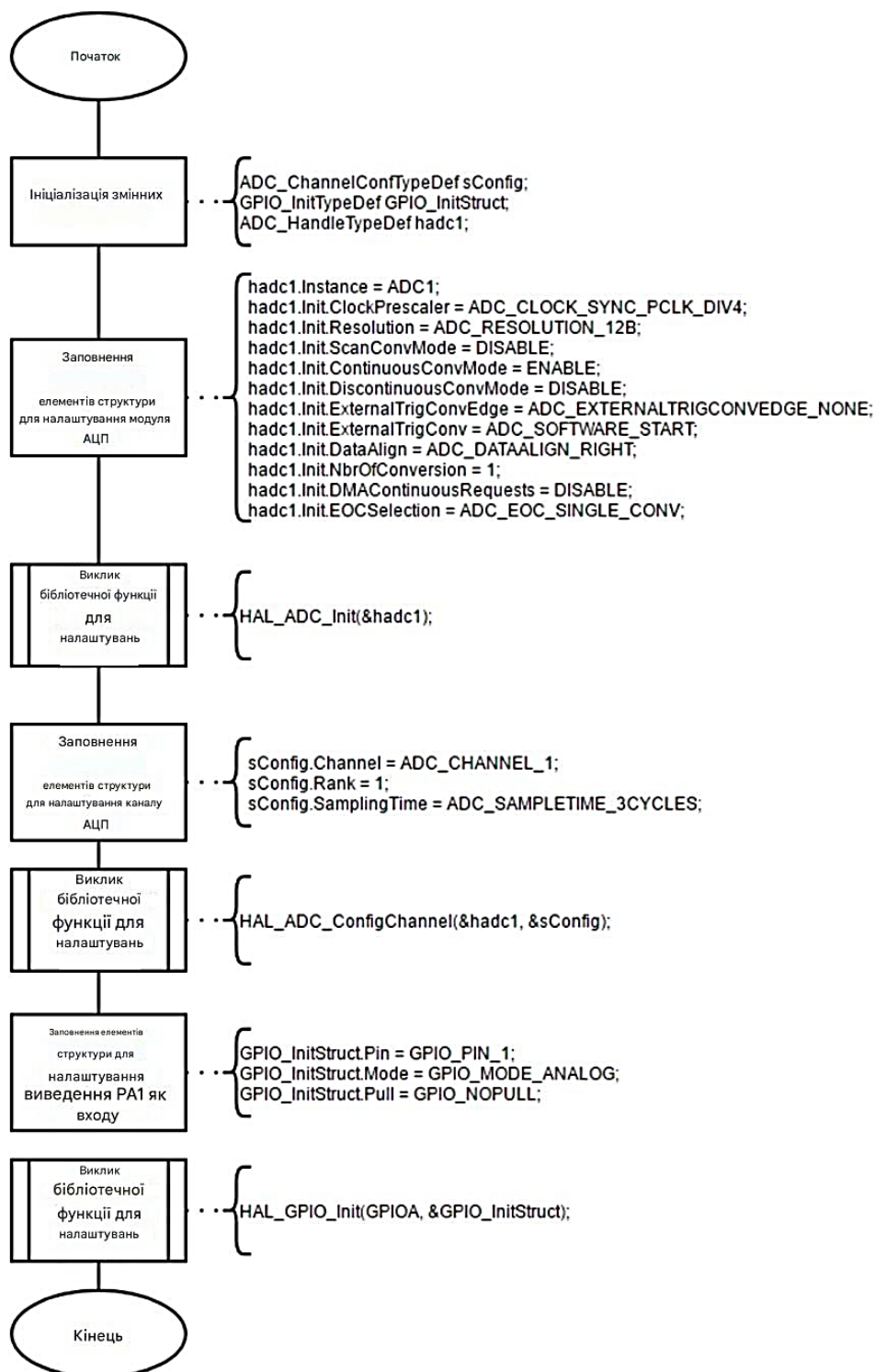


Рис. 3.26. Блок-схема підпрограми конфігурації АЦП

## **Висновки до розділу**

В даному розділі розроблено принципову конструктивну схему пристрою і принципова схема системи управління млинниці, структурно-функціональна схема, обрані комплектуючі та створено програму системи управління

## ВИСНОВКИ

В результаті виконання магістерського дослідження було успішно розроблено автоматизовану систему для приготування млинців, що базується на мікроконтролері STM32. Завдяки використанню 3D-друку та модульної конструкції, пристрій має компактні розміри та може бути легко адаптований під різні потреби користувача. Розроблена система керування забезпечує точний контроль температури нагрівальної поверхні за допомогою термодатчика та ПІД-регулятора. Для досягнення оптимального смаку млинців, передбачено можливість регулювання часу випікання та товщини тіста. Функціональний прототип демонструє високу ефективність та надійність у роботі, що дозволяє розглядати його як перспективний продукт для домашнього використання.

Розроблена система для автоматичного приготування млинців є інноваційним рішенням, яке поєднує в собі передові технології та зручність використання. Завдяки використанню відкритого програмного забезпечення та модульної конструкції, пристрій має високий потенціал для подальшого розвитку. Додаткові функції, такі як автоматичне дозування інгредієнтів, підключення до смартфона та розширення асортименту рецептів, можуть бути легко реалізовані в майбутніх версіях. Цей проєкт демонструє можливості сучасних технологій в сфері домашньої автоматизації та відкриває нові перспективи для розробки розумних кухонь.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Белов А.В. Самоучитель по микропроцессорной технике/А.В. Белов. - СПб.: Наука и техника, 2003. - 224с.
2. Бесекерский В.А. Теория систем автоматического управления / В.А. Бесекерский, Е.П. Попов. М.: "Профессия", 2004. – 747 с.
3. Власов К.П. Теория автоматического управления. – Харьков.: Изд-во "Гуманитарный центр", 2007. – 526 с.
4. Водовозов А. М. Микроконтроллеры для систем автоматики: учебное пособие/ А. М. Водовозов : Инфра-Инженерия , 2016. - 164 с.
5. Воронов А.А. Основы теории автоматического управления. - М.–Л.: Энергия, 1965, ч.1, - 423 с.
6. Гайдук А.Р. Теория автоматического управления в примерах и задачах с решениями в MATLAB : учеб. пособие для вузов / А. Р. Гайдук, В. Е. Беляев, Т. А. Пьявченко. - 2-е изд., испр. - Электрон. дан. - СПб. : Лань, 2011. - 464 с.
7. Голубцов М.С. Микроконтроллеры AVR: от простого к сложному/ М.С. Голубцов - М.: Изд -во СОЛОН-Пресс, 2003. - 292 с.
8. Гусев В.Г.Электроника и микропроцессорная техника/ В.Г.Гусев, Ю.М. Гусев, 2013.
9. Джон Мортон. Микроконтроллеры AVR. Вводный курс. / Пер. с англ./Мортон Джон - М.: Издательский дом "Додэка XXI", 2006. - 272 с.
- 10.Зайцев Г.Ф. Теория автоматического управления и регулирования. 2-е издание перераб. и дополн. – К.: Высшая школа, 1988. – 430 с.
- 11.Иващенко Н.Н. Автоматическое регулирование. Теория и элементы систем. - М.: Машиностроение, 1973. – 606 с.
- 12.Калабеков Б.А. Микропроцессоры и их применение в системах передачи и обработки сигналов, Учебное пособие./ Б.А. Калабеков. М.: Радио и связь, 368 с.
- 13.Колонтаєвський Ю.П. Промислова електроніка та мікросхемотехніка./ Ю.П. Колонтаєвський, А.Г. Сосков К.: Каравела, 2004

14. Колонтаєвський Ю.П. Електроніка і мікросхемотехніка / Ю.П. Колонтаєвський, А.Г. Сосков, К.: Каравела, 2006
15. Лукас В.А. Теория автоматического управления. - М.: Недра, 1990. - 416 с.
16. Математические основы теории автоматического регулирования / под ред. Б.К. Чемоданова. - М.: Высшая школа, 1971. - 807 с.
17. Мирошник И.В. Теория автоматического управления. Линейные системы. - СПб: "Питер", 2005. - 333 с.
18. Михайлов В.С. Теория управления. Учебное пособие для ВУЗов. - К.: Высшая школа, 1988. - 309 с.
19. Н.М. Щупляк. Основи електроніки і мікроелектроніки (книга 2), Дрогобич, 2012 13. Головки В.М. Теоретичні основи автоматики / В.М. Головки, Ніжин, 2004.
20. Пантелеев А.В. Теория управления в примерах и задачах / А.В. Пантелеев, А.С. Бортакровский. - М., Высшая школа, 2003. - 583 с.
21. Петров М.Н. Моделирование компонентов и элементов интегральных схем: учебное пособие для студентов вузов / М. Н. Петров, Г. В. Гудков. - Электрон. дан. - СПб. : Лань, 2011. - 464 с.
22. Попов Е.П. Теория линейных систем автоматического регулирования и управления. - М.: Наука, 1989. - 496 с.
23. Решмин Б.И. Имитационное моделирование и системы управления: учебно-практическое пособие / Б.И. Решмин: Инфра-Инженерия, 2016
24. Смирнов Ю.А. Основы микроэлектроники и микропроцессорной техники: учеб. пособие для студентов вузов / Ю. А. Смирнов, С. В. Соколов, Е. В. Титов. - 2-е изд., испр. - Электрон. дан. - СПб. : Лань, 2013. - 496 с.
25. Солодовников В.В. Основы теории и элементы систем автоматического регулирования / В.В. Солодовников, В.Н. Плотников, А.В. Яковлев. - М.: Машиностроение, 1985. - 536 с.
26. Теория автоматического управления / под ред. А.В. Нетушила. - М.: Высшая школа, 1972. - 432 с.

27. Филипс Ч. Системы управления с обратной связью / Ч. Филипс, Р. Харбор.  
- М.: Лаборатория базовых знаний, 2001. – 616 с.

## ДОДАТОК А

Лістинг коду

```
int main(void)
{
    uint32_t freq;
    /* MCU Configuration-----*/
    /* Reset of all peripherals, Initializes the Flash interface and the SysTick. */
    HAL_Init();
    /* Configure the system clock */
    SystemClock_Config();
    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_TIM1_Init();
    MX_TIM3_Init();
    HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);
    HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_2);
    HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_3);
    64
    HAL_TIMEx_PWMN_Start(&htim1,TIM_CHANNEL_1);
    HAL_TIMEx_PWMN_Start(&htim1,TIM_CHANNEL_2);
    HAL_TIMEx_PWMN_Start(&htim1,TIM_CHANNEL_3);
    HAL_TIM_Base_Start_IT(&htim3);
    /* Infinite loop */
    while (1)
    {
        HAL_ADC_Start(&hadc1);
        HAL_ADC_PollForConversion(&hadc1, 100);
        freq = 16 * HAL_ADC_GetValue(&hadc1);
        HAL_ADC_Stop(&hadc1);
        if (freq > 12000)
        {
            TIM3->ARR = freq;
        }
        else
```

```

{
TIM3->ARR = 500;
}
}
}

/** System Clock Configuration
*/

void SystemClock_Config(void)
{
RCC_OscInitTypeDef RCC_OscInitStruct;
RCC_ClkInitTypeDef RCC_ClkInitStruct;

/**Configure the main internal regulator output voltage
*/

__HAL_RCC_PWR_CLK_ENABLE();

__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCAL
E1);

/**Initializes the CPU, AHB and APB busses clocks
*/

RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
RCC_OscInitStruct.HSEState = RCC_HSE_ON;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
RCC_OscInitStruct.PLL.PLLM = 4;
RCC_OscInitStruct.PLL.PLLN = 168;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 4;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
__Error_Handler(__FILE__, __LINE__);
}

/**Initializes the CPU, AHB and APB busses clocks
*/

RCC_ClkInitStruct.ClockType
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;

```

```

RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
65
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
/**Configure the SysTick interrupt time
*/
HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);
/**Configure the SysTick
*/
HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);
/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

```

Листинг файла tim.c

```

#include "tim.h"
TIM_HandleTypeDef htim1;
TIM_HandleTypeDef htim3;
/* TIM1 init function */
void MX_TIM1_Init(void)
{
    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;
    TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig;
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 0;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 65535;
    //htim1.Init.Period = 10000;
}

```

```

htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
htim1.Init.RepetitionCounter = 0;
if (HAL_TIM_PWM_Init(&htim1) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
sConfigOC.OCMode = TIM_OCMODE_PWM1;
sConfigOC.Pulse = 0;
sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCNPolarity = TIM_OCNPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
sConfigOC.OCIdleState = TIM_OCIDLESTATE_RESET;
sConfigOC.OCNIdleState = TIM_OCNIDLESTATE_RESET;
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_1) !=
HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_2) !=
HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_3) !=
HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
sBreakDeadTimeConfig.OffStateRunMode = TIM_OSSR_DISABLE;

```

```

sBreakDeadTimeConfig.OffStateIDLEMode = TIM_OSSI_DISABLE;
sBreakDeadTimeConfig.LockLevel = TIM_LOCKLEVEL_OFF;
sBreakDeadTimeConfig.DeadTime = 0;
sBreakDeadTimeConfig.BreakState = TIM_BREAK_DISABLE;
sBreakDeadTimeConfig.BreakPolarity = TIM_BREAKPOLARITY_HIGH;
sBreakDeadTimeConfigAutomaticOutput = TIM_AUTOMATICOUTPUT_DISABLE;
if (HAL_TIMEx_ConfigBreakDeadTime(&htim1, &sBreakDeadTimeConfig) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
67
}
HAL_TIM_MspPostInit(&htim1);
}
void MX_TIM1_Init2(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;
    TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig;
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 0;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 65535;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    if (HAL_TIM_Base_Init(&htim1) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
    if (HAL_TIM_PWM_Init(&htim1) != HAL_OK)

```

```

{
    _Error_Handler(__FILE__, __LINE__);
}

sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

sConfigOC.OCMode = TIM_OCMode_PWM1;
sConfigOC.Pulse = 0;
sConfigOC.OCpolarity = TIM_OCPolarity_HIGH;
sConfigOC.OCNPolarity = TIM_OCNPolarity_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
sConfigOC.OCIdleState = TIM_OCIDLESTATE_RESET;
sConfigOC.OCNIdleState = TIM_OCNIDLESTATE_RESET;
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_1) !=
HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_2) !=
HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_3) !=
HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

sBreakDeadTimeConfig.OffStateRunMode = TIM_OSSR_ENABLE;
sBreakDeadTimeConfig.OffStateIDLEMode = TIM_OSSI_ENABLE;
sBreakDeadTimeConfig.LockLevel = TIM_LOCKLEVEL_OFF;
sBreakDeadTimeConfig.DeadTime = 10;

```

```

sBreakDeadTimeConfig.BreakState = TIM_BREAK_ENABLE;
68
sBreakDeadTimeConfig.BreakPolarity = TIM_BREAKPOLARITY_HIGH;
sBreakDeadTimeConfigAutomaticOutput = TIM_AUTOMATICOUTPUT_DISABLE;
if (HAL_TIMEx_ConfigBreakDeadTime(&htim1, &sBreakDeadTimeConfig) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
HAL_TIM_MspPostInit(&htim1);
}
/* TIM3 init function */
void MX_TIM3_Init(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;
    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 0;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 65535;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

```

```

}

void HAL_TIM_PWM_MspInit(TIM_HandleTypeDef* tim_pwmHandle)
{
    if(tim_pwmHandle->Instance==TIM1)
    {
        /* Peripheral clock enable */
        __HAL_RCC_TIM1_CLK_ENABLE();
    }
}

void HAL_TIM_Base_MspInit(TIM_HandleTypeDef* tim_baseHandle)
{
    if(tim_baseHandle->Instance==TIM3)
    {
        /* Peripheral clock enable */
        __HAL_RCC_TIM3_CLK_ENABLE();
        /* TIM3 interrupt Init */
        HAL_NVIC_SetPriority(TIM3_IRQn, 0, 0);
        HAL_NVIC_EnableIRQ(TIM3_IRQn);
    }
}

void HAL_TIM_MspPostInit(TIM_HandleTypeDef* timHandle)
{
    GPIO_InitTypeDef GPIO_InitStruct;
69
    if(timHandle->Instance==TIM1)
    {
        /**TIM1 GPIO Configuration
        PB13 -----> TIM1_CH1N
        PB14 -----> TIM1_CH2N
        PB15 -----> TIM1_CH3N
        PA8 -----> TIM1_CH1
        PA9 -----> TIM1_CH2
        PA10 -----> TIM1_CH3
        */
        GPIO_InitStruct.Pin = GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15;
    }
}

```

```

GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF1_TIM1;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF1_TIM1;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}
}
void HAL_TIM_PWM_MspDeInit(TIM_HandleTypeDef* tim_pwmHandle)
{
if(tim_pwmHandle->Instance==TIM1)
{
/* Peripheral clock disable */
__HAL_RCC_TIM1_CLK_DISABLE();
}
}
void HAL_TIM_Base_MspDeInit(TIM_HandleTypeDef* tim_baseHandle)
{
if(tim_baseHandle->Instance==TIM3)
{
/* Peripheral clock disable */
__HAL_RCC_TIM3_CLK_DISABLE();
/* TIM3 interrupt Deinit */
HAL_NVIC_DisableIRQ(TIM3_IRQn);
}
}
70

```

Лістинг файла adc.c

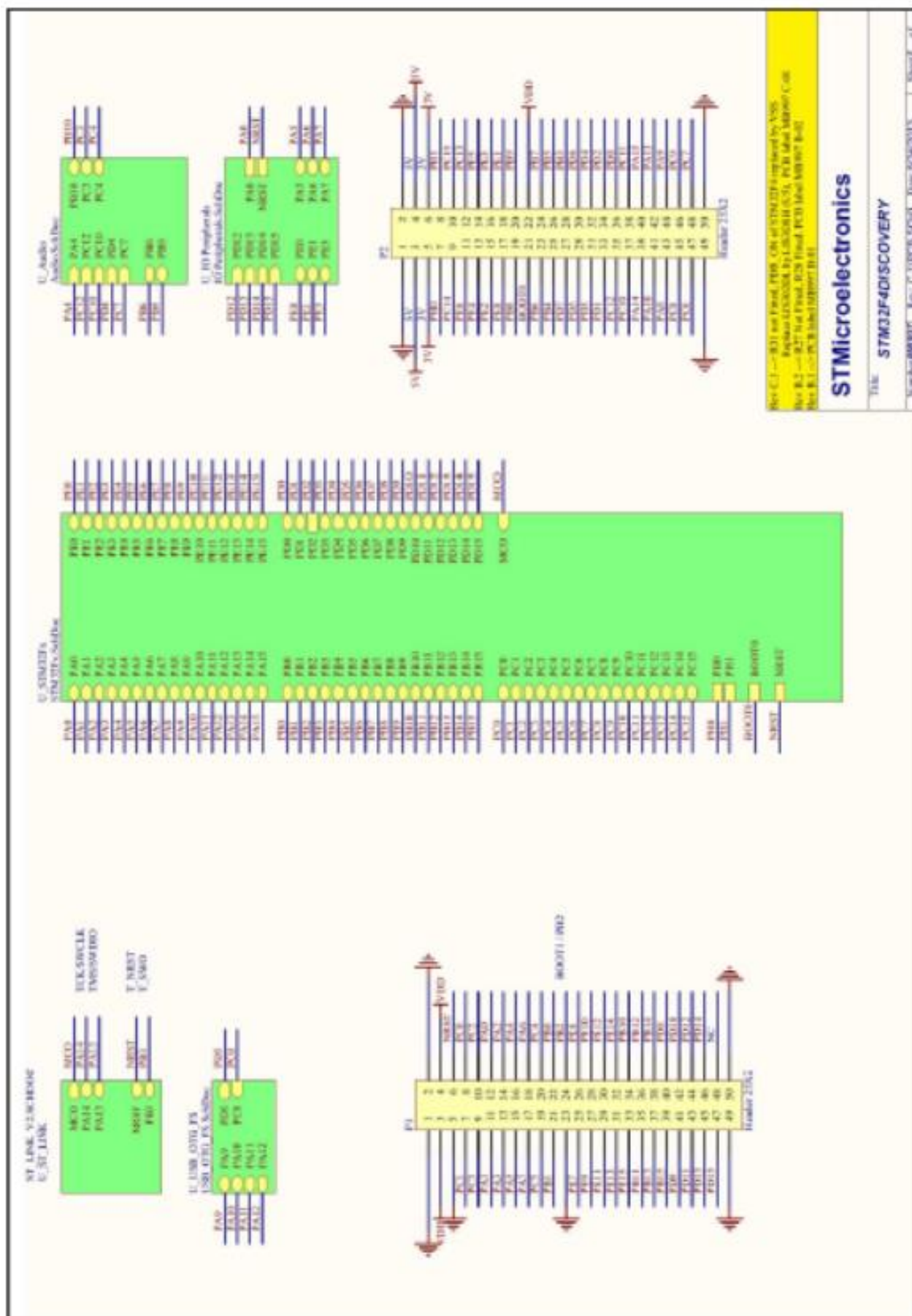
```

#include "adc.h"
ADC_HandleTypeDef hadc1;

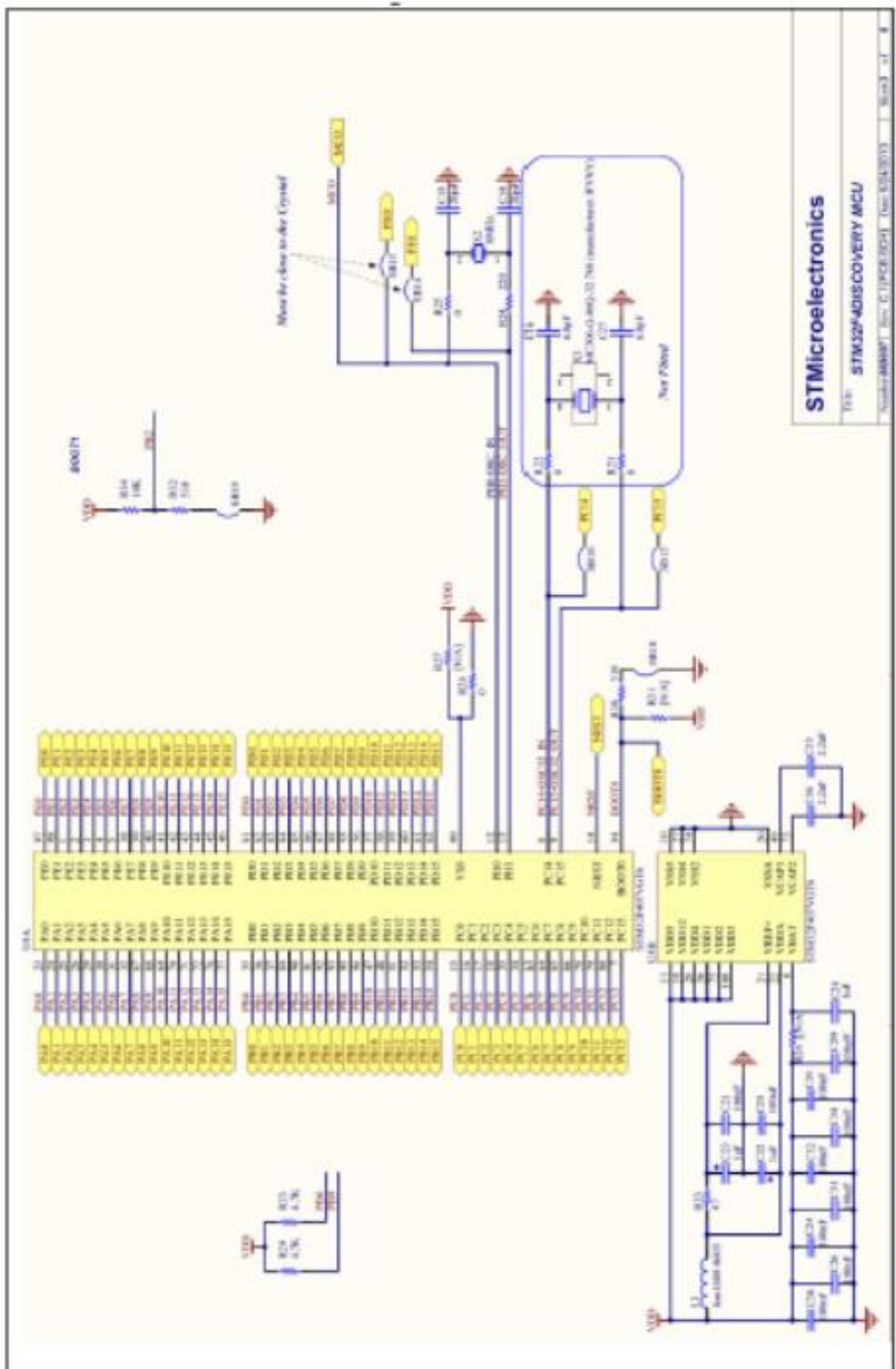
```

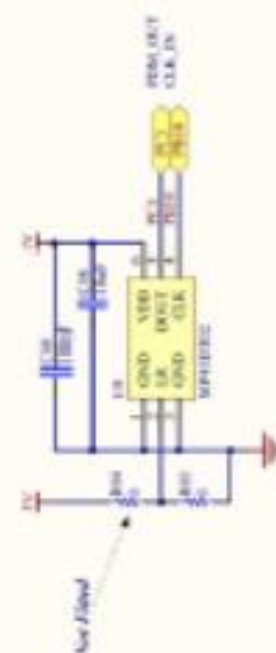
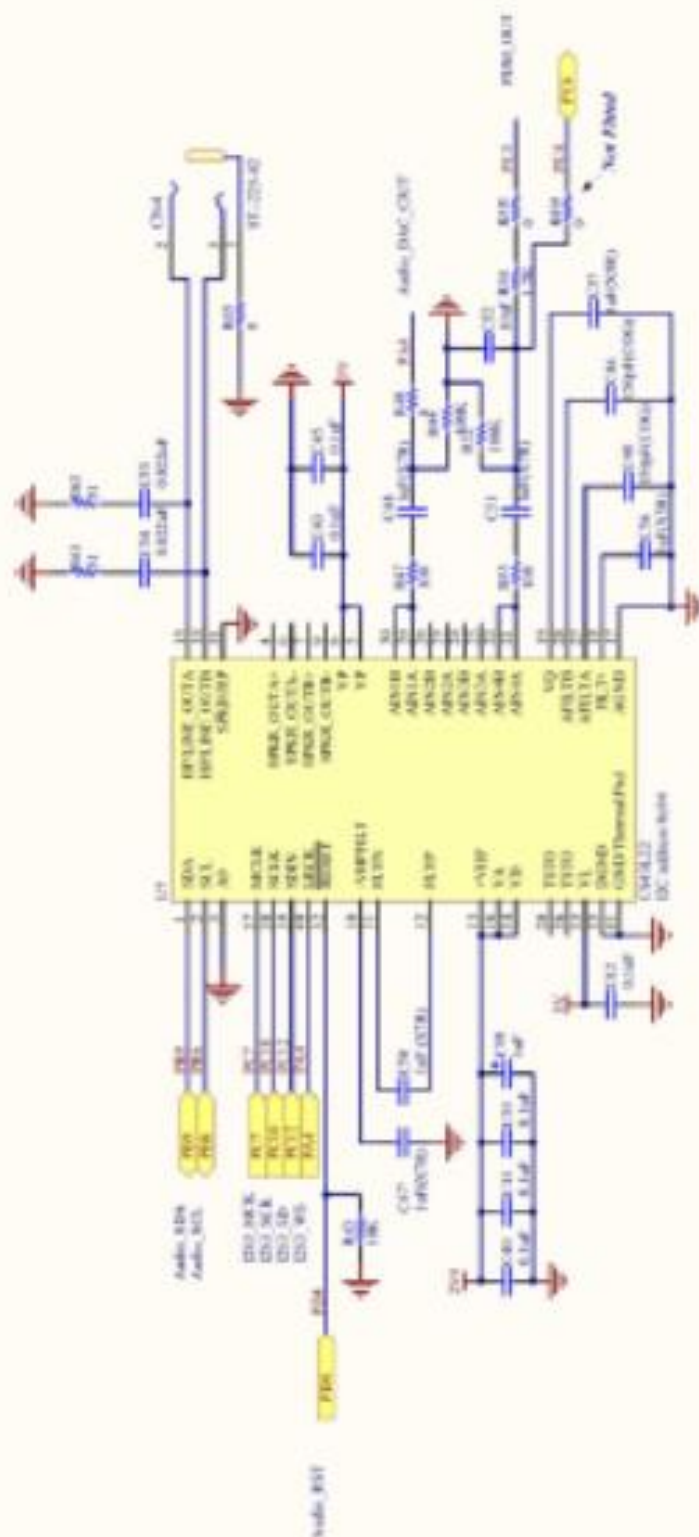
```
/* ADC1 init function */  
void MX_ADC1_Init(void)  
{  
    ADC_ChannelConfTypeDef sConfig;  
    hadc1.Instance = ADC1;  
    hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;  
    hadc1.Init.Resolution = ADC_RESOLUTION_12B;  
    hadc1.Init.ScanConvMode = DISABLE;  
    hadc1.Init.ContinuousConvMode = ENABLE;  
    hadc1.Init.DiscontinuousConvMode = DISABLE;  
    hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;  
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;  
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;  
    hadc1.Init.NbrOfConversion = 1;
```

## Схеми модуля STM32F4DISCOVERY









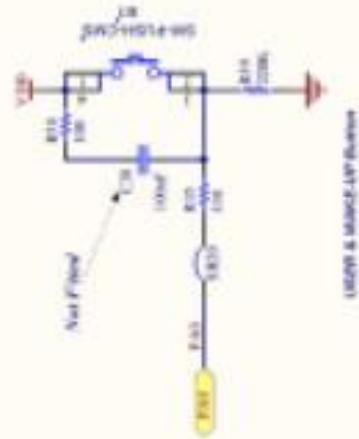
**STMicroelectronics**

STIMULUS CONVEY AUDIO

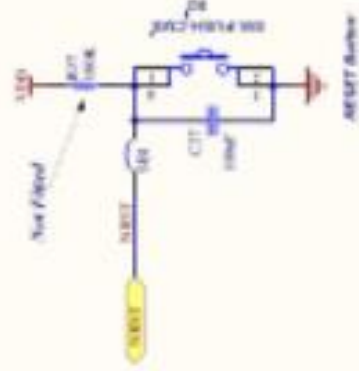
STATISTICS IN TRANSITION: New Zealand 2007

**Journal of Management Inquiry**

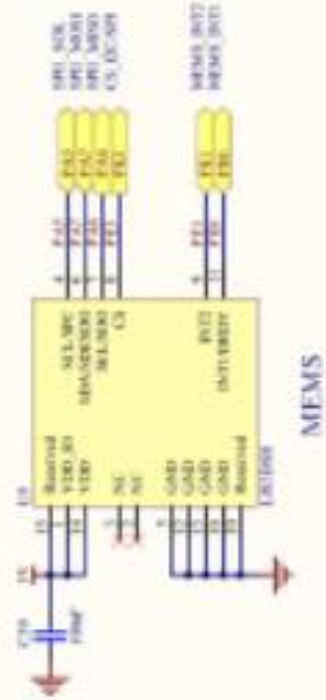




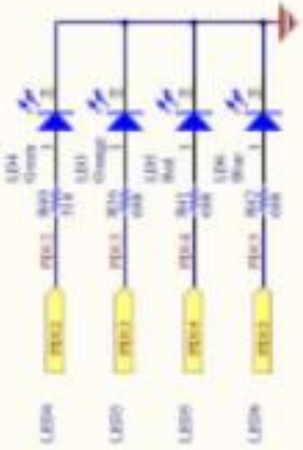
USB to UART Bridge



USB to UART Bridge



MEMS



LEDs