

**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 магістр**

зі спеціальності 171 Електроніка освітньо-наукової програми «Електронні інформаційні системи»

на тему: **АРХІТЕКТУРА ПРИСТРОЇВ ЗБОРУ ДАНИХ НА БАЗІ МІКРОКОНТРОЛЕРІВ ARDUINO**

Здобувача (ки) групи ЕП-91 Монастирєва Кирила Сергійовича

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

Кирило МОНАСТИРЬОВ

Керівник канд. фіз.-мат. наук, асистент

Андрій ЛОГВИНОВ

Консультант<sup>1)</sup> \_\_\_\_\_  
(посада, науковий ступінь, вчене звання Ім'я та ПРІЗВИЩЕ) (підпис)

**Суми – 2023**

## ЗМІСТ

|  | С. |
|--|----|
| <b>АНОТАЦІЯ</b> .....  | 3  |
| <b>РОЗДІЛ 1 ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ARDUINO І СИСТЕМИ РЕАЛІЗОВАНІ НА ЇЇ ОСНОВІ</b> .....                             | 4  |
| 1.1. Загальні відомості про Arduino.....   | 4  |
| 1.2. Конструкція та архітектура Arduino.....   | 5  |
| 1.3. Літературний огляд .....  | 8  |
| <b>РОЗДІЛ 2 МЕТОДИКА І ТЕХНІКА ЕКСПЕРИМЕНТУ</b> .....  | 12 |
| 2.1. Вплив температури на опір. Вимірювання температурного коефіцієнту опору .....                                   | 12 |
| 2.2. Постановка задачі і використані елементи у роботі.....  | 15 |
| 2.3. Методологія проведення вимірювання температурного коефіцієнту опору і середовище віртуального моделювання ..... | 18 |
| <b>РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА</b> .....   | 21 |
| 3.1. Принцип дії установки.....  | 21 |
| 3.2. Написання програмного коду для системи вимірювання ТКО.....   | 26 |
| 3.3. Аналіз і тестування програми.....   | 31 |
| <b>ВИСНОВКИ</b> .....  | 33 |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....  | 34 |

## АНОТАЦІЯ

Кваліфікаційна робота викладена на 36 сторінках, зокрема, містить 20 рисунків, 5 таблиць, список використаних джерел із 24 найменувань.

Актуальність теми полягає в тому, що пристрої збору даних в сучасному світі знаходять все більше застосування в різних сферах діяльності. Вони є необхідним інструментом для збору, аналізу та зберігання даних, що дозволяє зробити більш точний аналіз певних ситуацій або факторів. Збір даних стає надзвичайно важливим елементом у таких сферах як медицина, промисловість, енергетика, автоматизація та багато інших. Архітектура пристроїв збору даних на базі мікроконтролерів Arduino базується на інтеграції мікроконтролера та інших компонентів, таких як сенсори, модулі комунікації, дисплеї та інші електронні компоненти.

Мета кваліфікаційної роботи бакалавра полягає дослідженні архітектури пристроїв для збору даних, основою яких є мікроконтролери Arduino, їх можливості та сфери використання. Створення автоматичної системи збору даних для розрахунку коефіцієнту термічного опору на основі мікроконтролера Arduino.

Під час виконання роботи використовували методи визначення температурного коефіцієнту опору матеріалу та прилади необхідні для його розрахунку. Був використаний мікроконтролер Arduino Uno R3, датчик температури, мультиметр.

У результаті було створено систему для автоматичного збору та розрахунку параметру термічного коефіцієнту опору матеріалу. Проведена симуляція даної системи у середовищі Tinkercad, отримані результати симуляції співпадають із реальними показниками. Отриману автоматизовану систему можна використати в лабораторіях для досліджень і в навчальних цілях.

**Ключові слова:** АВТОМАТИЗОВАНА СИСТЕМА ЗБОРУ ДАНИХ, ARDUINO, ДАТЧИКИ, ТЕРМІЧНИЙ КОЕФІЦІЄНТ ОПОРУ

# РОЗДІЛ 1

## ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ARDUINO І СИСТЕМИ РЕАЛІЗОВАНІ НА ЇЇ ОСНОВІ

### 1.1. Загальні відомості про Arduino

Вперше Arduino була створена у 2005 році як проект з відкритим кодом в межах Італійського інституту дизайну. З того часу, ця платформа стала дуже популярною серед ентузіастів, студентів, інженерів та розробників по всьому світу. Однією з особливостей Arduino є його простота використання та доступність. Вона має вбудовану підтримку мови програмування C++, має відкрите програмне забезпечення та може працювати на різних платформах, таких як Windows, MacOS та Linux. Більшість плат Arduino мають схему підключення, що дозволяє підключати до них електронні компоненти без необхідності в розробці власної схеми [1].



Рисунок 1.1 – Приклад мікроконтролера Arduino Due [2]

За останні роки мікроконтролери стали доступнішими та зручнішими. У той же час комп'ютери стали швидшими та потужнішими, дозволяючи створювати

кращі і простіші засоби розробки. Основною ідеєю таких систем як Arduino є ідея швидкої практичної розробки прототипів. Створення кращих прототипів вимагає постійного пошуку швидших і потужніших методів [3].

Тож Arduino - це компактна мікроконтролерна плата з універсальними способами підключення у вигляді USB і виходів, що дозволяє під'єднувати до неї різноманітні зовнішні електронні пристрої: реле, датчики світла, гучномовці, мотори, світлодіоди, дисплеї та інше, за допомогою портів з'єднання на платі. Плата живиться від батареї 9В, джерела живлення або USB-з'єднання з комп'ютером. Мікроконтролер може контролюватися напряду від комп'ютера, або запрограмований і працювати автономно. Arduino дозволяє програмувати та керувати зовнішніми пристроями, і є широко використовується платформою для розробки прототипів інтерактивних електронних пристроїв. Проект Arduino був створений для освітнього процесу та є дуже популярним навчальним ресурсом. Використання відкритого коду породило спільноту, що вільно поширює знання, рішення та проекти, також надихнуло на обмін навчальними програмами, стратегіями викладання та іншими матеріалами. Широке використання електроніки з відкритим кодом і мікроконтролерів, таких як Arduino або Raspberry Pi, принесло вченим нові можливості створювати свої інструменти на основі відкритого коду. Таким чином Arduino також є універсальним контролером, який можна підлаштувати для будь-яких задач і перетворити на повноцінний електронний пристрій для будь-яких цілей, від побутового до науково-дослідницького пристрою [1-3].

## **1.2. Конструкція та архітектура Arduino**

Існує багато різних моделей Arduino, які можуть відрізнятися за конструкцією, функціональністю, розмірами, кількістю входів-виходів та іншими параметрами. На сьогоднішній день представлені різні види плат Arduino, які мають різну кількість входів/виходів та інші особливості, які можуть відрізнятися в залежності від потреб користувача. Такими прикладами є Arduino Uno, Arduino

Mega, Arduino Nano, Arduino Due, Arduino Pro Mini та Arduino Leonardo. Кожен з цих видів має свої власні особливості та можливості, які дозволяють користувачам виконувати різноманітні завдання та проекти.

Надалі будемо розглядати використання платформи останньої версії Arduino Uno. Основною частиною Arduino є мікроконтролер, який керує всіма операціями на платі. Найбільш популярним мікроконтролером на платі Arduino є ATmega328P, який має 32 кілобайти програмної пам'яті, 2 кілобайти оперативної пам'яті та 14 цифрових входів/виходів. Плата також містить кварцовий резонатор, який відповідає за точний розрахунок часу та дати, світлодіод та динамік, які забезпечують світлові та звукові сигнали. Вона має декілька типів входів/виходів, включаючи аналогові входи, цифрові входи та виходи, а також PWM виходи, які можуть керувати яскравістю світла на світлодіодах або швидкістю обертання моторів. Крім того, на платі є інтерфейси для зв'язку з зовнішніми пристроями, такі як USB, Ethernet, Bluetooth або Wi-Fi, які дозволяють взаємодіяти з Arduino через комп'ютер або мобільний пристрій [4-5].

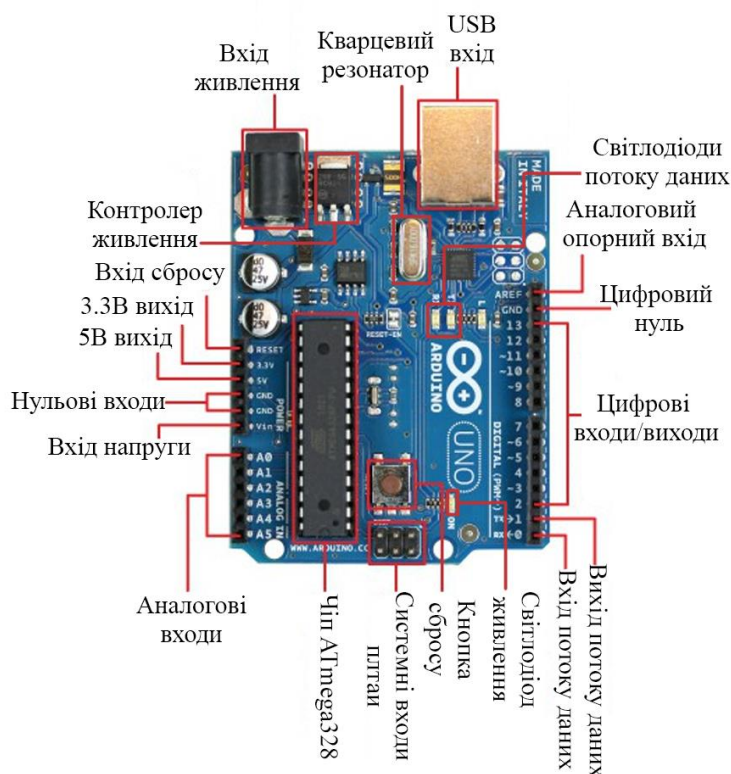


Рисунок 1.2 – Конструкція мікроконтролера Arduino Uno R3. Адаптовано з праці [4]

Arduino використовує архітектуру типу Гарварда, яка передбачає використання окремих блоків пам'яті для зберігання програмних інструкцій та даних. Ключовою особливістю цієї архітектури є використання двох окремих блоків пам'яті, одного для зберігання програмних інструкцій і іншого для зберігання програмних даних. Дані зберігаються в пам'яті даних, а код – у флеш-пам'яті. Центральний процесор отримує доступ до обох блоків пам'яті за допомогою різних комунікаційних шин. Така архітектура забезпечує ефективну роботу платформи Arduino та її можливості взаємодії з різними елементами системи та іншими зовнішніми пристроями.

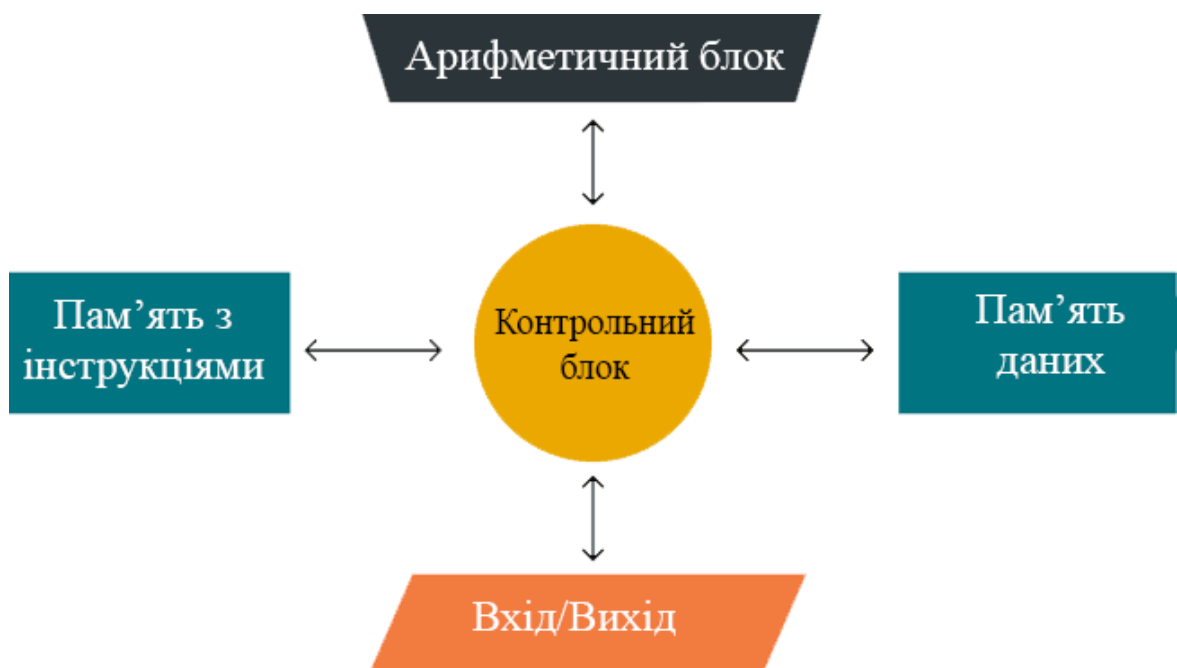


Рисунок 1.3 – Блок-схема архітектури Гарвард. Адаптовано з праці [6]

В основі Arduino Uno R3 є мікроконтролер ATmega328P. Він підтримує 8-бітну обробку даних. Має вбудовану флеш-пам'ять об'ємом 32 КБ і 1 КБ електрично незалежної програмованої постійної пам'яті (EEPROM), що надає можливості навіть при відсутності електричного живлення, що подається на мікроконтролер можливе зберігання даних та надавання результатів після повторного підключення до електричного живлення. Крім того, ATmega-328 має 2 КБ статичної оперативної пам'яті (SRAM). Особливостям мікроконтролера є вдосконалена архітектура RISC, висока продуктивність, низьке енергоспоживання,

лічильник реального часу з окремим генератором, 6 контактів ШІМ, програмований послідовний USART, блокування програмування для безпеки програмного забезпечення, пропускну здатність до 20 MIPS.

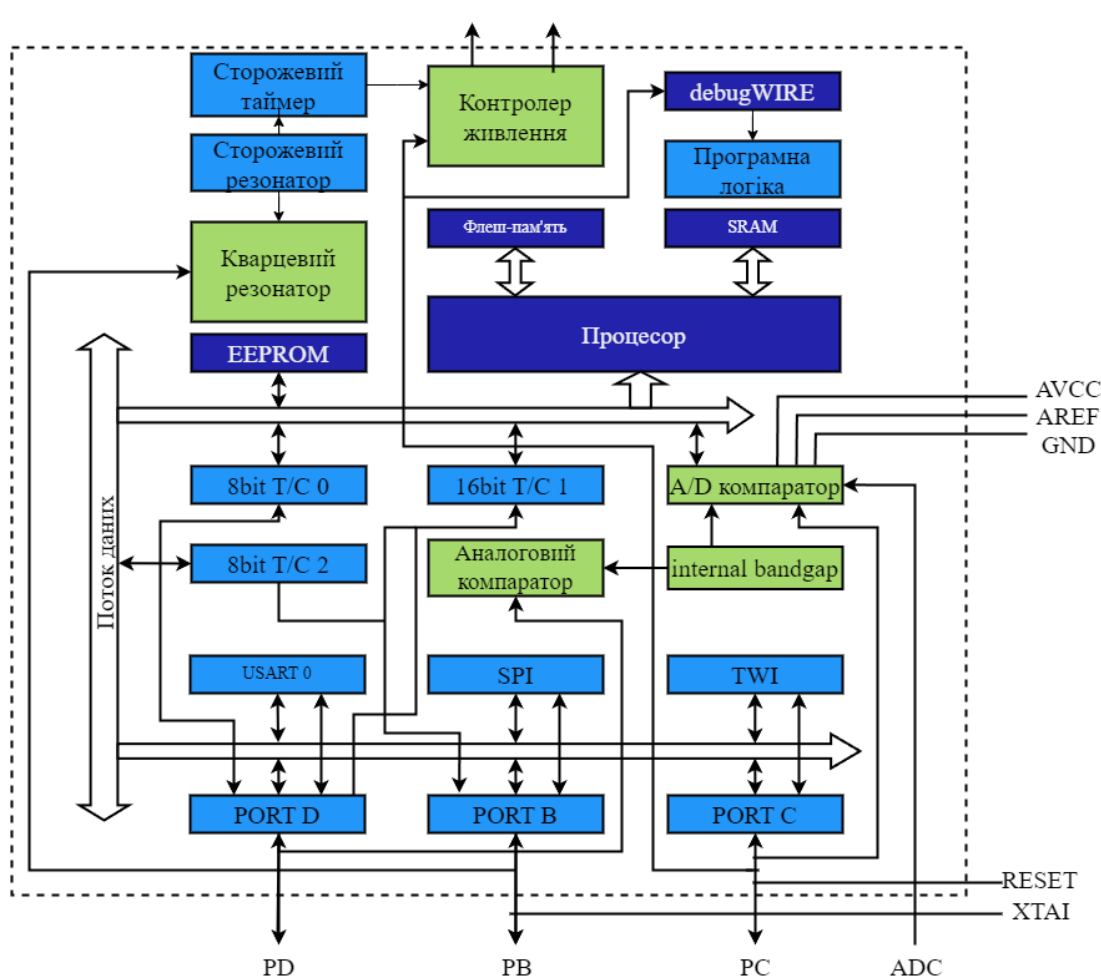


Рисунок 1.4 – Блок-схема мікроконтролера ATmega328P. Адаптовано з праці [4]

### 1.3. Принципи функціонування пристрою

Arduino може бути використана як основа для розробки різноманітних пристроїв збору даних. Завдяки своїм можливостям з підключення зовнішньої електроніки, Arduino дозволяє зчитувати різноманітні дані з датчиків, реле, гучномовців, мікрофонів та інших пристроїв. Наприклад, можна створити пристрій збору даних для моніторингу температури та вологості в приміщенні. Дані з цих датчиків можна зчитувати за допомогою програми, написаної для Arduino, і



виводити на дисплей, зберігати в пам'яті або надсилати через Wi-Fi або Bluetooth. Перейра П. займався дослідженням дешевих датчиків температури, вологості і CO<sub>2</sub>. Об'єктами дослідження протягом 24 місяців були такі датчики як SHT31, SHT85, SHT75, DHT11, DHT22 які показали прийнятні показники з допустимою похибкою для повсякденного використання, наприклад у системах розумних будинків. Але всі виміри проходили всередині приміщення, тому вплив зовнішнього середовища може сильніше впливати на результат [7]. Іншим прикладом використання є реалізація Лі Ж. розумної системи освітлення у офісі, за допомогою мікроконтролера Arduino Uno, wi-fi модулів і сенсорів освітлення та руху. В праці було описано загальний принцип дії системи, але не наведений програмний код, схеми підключення [8].

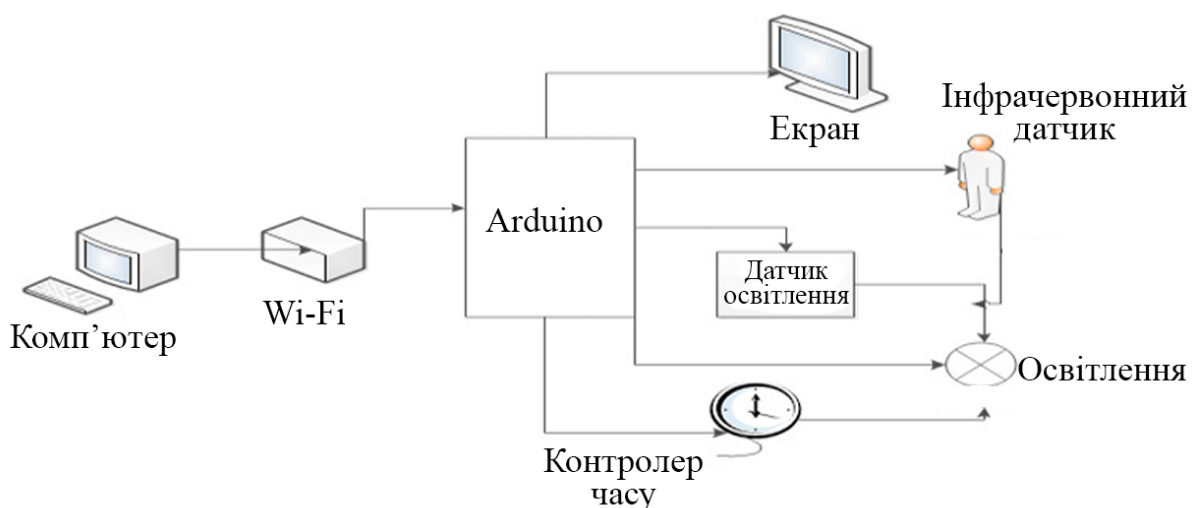


Рисунок 1.5 – Схема розумного освітлення з використанням Arduino. Адаптовано з праці [8]

Окрім того, можна використовувати Arduino для створення пристроїв збору даних у навчальних процесах, наукових експериментах, медичних обстеженнях, екологічних дослідженнях та інших областях. Завдяки своїй простоті та доступності, Arduino стала дуже популярною платформою для розробки пристроїв збору даних у різних галузях. Валеїо У. разом із своєю командою описав методологію і створив калориметр на основі Arduino для бюджетних хімічних лабораторій. У цій праці головною перевагою є простота конструкції і швидкість її

збірки, але похибка вимірювань у 2% не підходить для наукових досліджень. Також проблемою може бути пластиковий корпус транзистора, який використовується як датчик температури [9].

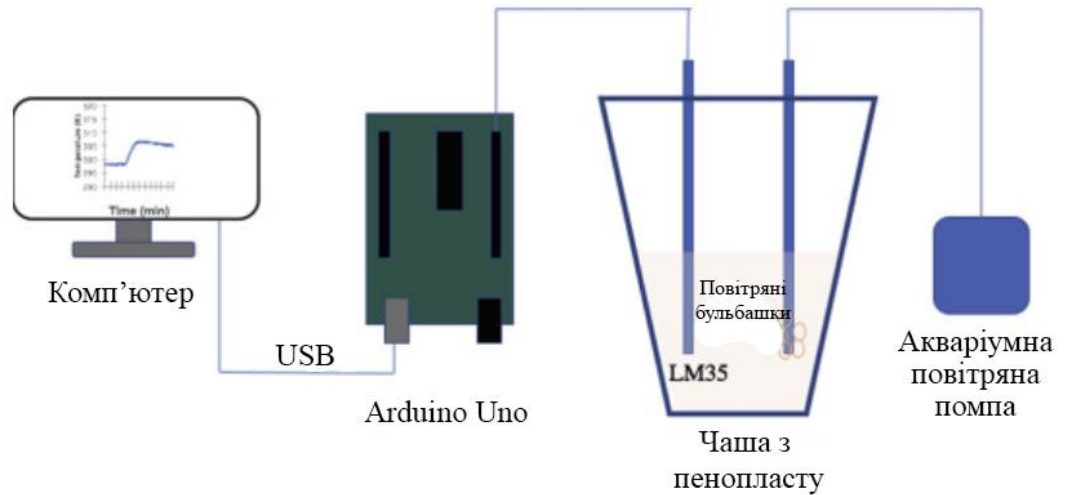


Рисунок 1.6 – Схема калориметру на основі транзистора LM35 і Arduino Uno. Адаптовано з праці [9]

Лей К. продемонстрував використання мікроконтролера Arduino у CubeSat – невеликому автономному апараті запущеному на висоту у 16.3 км над рівнем моря, для зняття показників температури, тиску, положення і орієнтації, передачі даних і фото/відео зйомки. Результатом дослідження є можливість використання Arduino, як платформи для запуску і зняття показників у межах тропосфери.

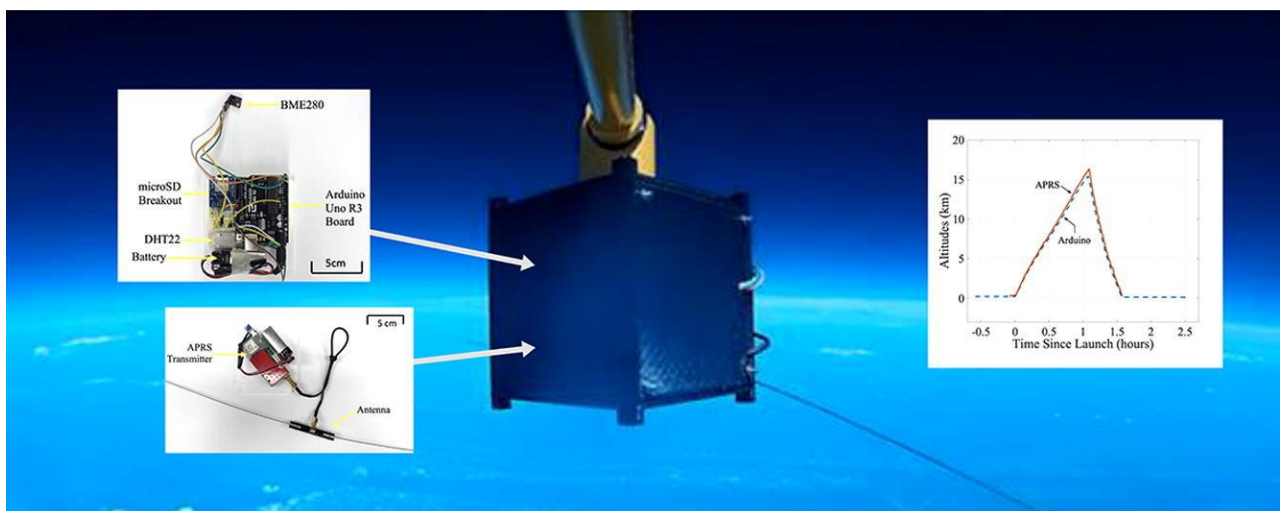


Рисунок 1.7 – CubeSat з використанням Arduino як системи для контролю польотом і зняття показників із датчиків [10]

У медичній сфері Бора П. займався створенням системи контролю біологічних показників, таких як серцебиття, температури, серцевого ритму пристарілих людей у домашніх умовах. Як зазначає автор публікації, такі системи є актуальними на сьогоднішній день через необхідність контролю стану здоров'я людей, за якими неможливо цілодобово приглядати. Така система також має відеокамеру і GPS модуль для того, щоб родичі/медичні працівники знали про місцезнаходження та його стан у реальному часі. Але у публікації не були продемонстровані реальні схеми для створення такої системи і не наведені переліки реально існуючих датчиків, також не зрозуміло, який час система може працювати автономно і необхідність підключення до інтернету [11].

Автоматизовані та автономні системи збору даних є важливими елементами для спостереження за екологічною обстановкою. Саї К. представив приклад дешевої системи для контролю показників якості повітря на основі датчиків М-серії, які можуть розпізнавати та фіксувати концентрацію вуглекислого газу, амоніаку і диму. Головним недоліком є способи виміру, які базуються на розрахунках одразу з декількох датчиків і потенційна похибка набуває суттєвих значень [12]. Схожий проект розробляв Кенон Дж., він створив робочий прототип системи для дослідження стану навколишнього середовища і випробовував його протягом 180 днів [13]. Arduino може бути використана і для створення систем автоматизації, наприклад, системи контролю доступу до приміщення або автомобіля. Вірареспаті А. створив робочий прототип для автомобільної системи безпеки, в якій користувач проходить перевірку перед тим, як отримати можливість завести авто. Також система збирає показники місцезнаходження і дані користувача, який намагався пройти верифікацію [14].

## РОЗДІЛ 2

### МЕТОДИКА І ТЕХНІКА ЕКСПЕРИМЕНТУ

#### 2.1. Вплив температури на опір. Вимірювання температурного коефіцієнту опору

Електричний опір матеріалу залежить від різних факторів, включаючи хімічний склад матеріалу, температуру, ступінь забруднення та розміри матеріалу. Для багатьох матеріалів, електричний опір змінюється зі зміною температури, що може бути важливим фактором при розробці електронних пристроїв та датчиків.

$$R = \rho \frac{l}{S} \quad (2.1.)$$

де  $R$  – опір провідника,  $\rho$  – питомий опір матеріалу,  $l$  – довжина провідника,  $S$  – площа перерізу провідника.

Первинні дослідження демонструють, що залежність питомого опору матеріалу від температури може бути наближено лінійною. Це означає, що збільшення температури призводить до збільшення опору матеріалу в однаковій пропорції. Однак, при подальшому вивченні можна виявити нелінійні залежності, особливо в екстремальних умовах. Наприклад, при дуже низьких температурах можуть виникнути фазові переходи, які значно змінюють опір матеріалу, а при дуже високих температурах можуть виникати термоелектричні ефекти, які також впливають на значення опору. Тому, для точного опису залежності опору від температури, необхідно проводити детальні дослідження в широкому діапазоні температур [15].

$$\rho = \rho_0 [1 + \alpha(T - T_0)] \quad (2.2.)$$

де  $\rho_0$  – питомий опір за температури  $T_0$ ,  $\alpha$  – температурний коефіцієнт опору

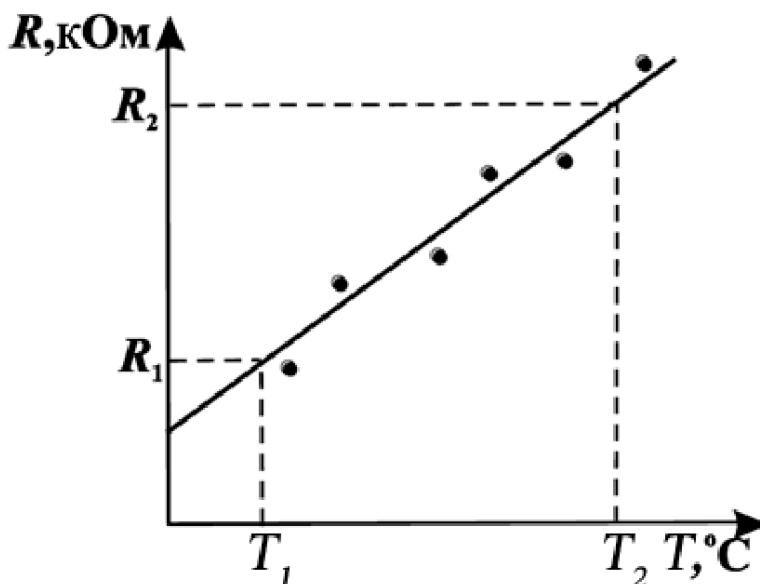


Рисунок 2.1 – Температурна залежність опору металів [2]

Питомий опір провідників залежить від домішок та способу виготовлення, а також від стану речовини, зокрема від температури, що характеризується температурним коефіцієнтом опору. Температурний коефіцієнт опору відображає числову залежність питомого опору однорідної речовини від зміни температури на 1К, і залежить не лише від хімічного складу, але і від стану речовини. Температурний коефіцієнт опору - це величина, яка характеризує зміну електричного опору провідника при зміні температури. Ця величина вимірюється у одиницях Ом/градус Цельсія ( $\Omega/^\circ\text{C}$ ). Температурний коефіцієнт опору базується на законі Ома, який стверджує, що електричний струм, що протікає через провідник, пропорційний напрузі та обернено пропорційний опору провідника. Оскільки температура впливає на кількість тепла, що генерується в провіднику, то це призводить до зміни його опору. І чим більший температурний коефіцієнт опору матеріалу, тим більше змінюється його опір при зміні температури.

$$\alpha = \frac{R-R_0}{R_0(T-T_0)} \quad (2.3.)$$

де  $R$  – опір за температури  $T$ ,  $R_0$  – опір за попередньої температури  $T_0$

Згідно з квантовою теорією, проходження струму через метал можна пояснити за допомогою розповсюдження електронних хвиль атомними коридорами (хвилеводи). Якщо атоми металу розташовані на площині в ідеальному порядку, на однакових відстанях один від одного, то ця площина повністю відбиває електронні хвилі. Розсіювання та поглинання електронних хвиль відбувається тільки тоді, коли порушується строгий порядок, це може бути спричинено тепловим рухом атомів і іонів [16].



Рисунок 2.2 – Тепловий рух атомів і іонів у провіднику, що порушують порядок хвилеводу [16]

Одним з методів визначення температурного коефіцієнту опору провідника є метод електричного нагрівання. Для цього провідник з'єднують з джерелом струму та вимірювальним приладом для вимірювання напруги та сили струму. Провідник поступово нагрівають, збільшуючи силу струму, що протікає через нього. Під час нагрівання вимірюють опір провідника та температуру, яку вимірюють за допомогою терморпарі або термометра. Після цього обчислюють температурний коефіцієнт опору за формулою 2.3. Методологія проведення виміру значення температурного коефіцієнту опору матеріалу зазвичай включає декілька етапів, що можна поділити на дві частини: експериментальну і аналітичну. У експериментальній частині перш за все необхідно підготувати матеріал, який буде використовуватися для вимірювання, повинен бути приготований до

експерименту. Це може включати чищення поверхні матеріалу та використання спеціальних засобів для забезпечення належного контакту з датчиком температури. Саме вимірювання температури повинно проводитися за допомогою точного датчика температури, який може бути підключений до платформи Arduino, або іншого мікроконтролерного пристрою. Датчик температури повинен бути розміщений в безпосередньої близькості до матеріалу, щоб отримати максимально точні результати. Для вимірювання опору матеріалу потрібен високоточний мультиметр або інший прилад для вимірювання опору. Мультиметр повинен бути підключений до платформи Arduino або іншого мікроконтролерного пристрою, щоб забезпечити автоматичне зчитування даних.

Аналітична частина полягає у зборі даних про температуру та опору матеріалу, ці дані повинні бути оброблені, щоб визначити температурний коефіцієнт опору матеріалу. Це може включати використання спеціальних формул та алгоритмів, щоб перетворити вимірювання опору на температурний коефіцієнт. Результати вимірювання повинні бути аналізовані, щоб забезпечити належну точність та достовірність вимірювань. Якщо необхідно, можуть бути проведені [15,17].

## **2.2. Постановка задачі і використані елементи у роботі**

Завданням цієї праці є створення системи автоматичного збору даних на основі мікроконтролера Arduino, що дозволить вимірювати значення опору провідника при різних температурах та розраховувати його температурний коефіцієнт опору. Система повинна бути здатна зчитувати значення температури з датчика температури, а також забезпечувати точне вимірювання опору провідника з використанням вбудованого аналого-цифрового перетворювача. Результати вимірювань повинні зберігатися в пам'яті мікроконтролера та передаватися на зовнішній пристрій для подальшого аналізу. Така система може бути використана для визначення температурного коефіцієнту опору різних матеріалів, що є

важливим для багатьох наукових досліджень та промислових застосувань, значно спрощує цей процес і економить час.

Основою системи є використання Arduino Uno R3 для створення системи автоматичного збору даних для розрахунку температурного коефіцієнту опору є доцільним і практичним рішенням, оскільки ця платформа має достатньо ресурсів та можливостей для реалізації такого проекту. Arduino Uno R3 має вбудований USB-інтерфейс, що дозволяє легко передавати дані на комп'ютер або в Інтернет, що є необхідною функцією для автоматичного збору даних і їх подальшого аналізу і передачі.

Таблиця 2.1 – Характеристики Arduino Uno R3 [6].

|                                  |                                   |
|----------------------------------|-----------------------------------|
| Мікроконтролер                   | ATmega328                         |
| Робоча напруга                   | 5В                                |
| Напруга живлення (рекомендована) | 7-12В                             |
| Напруга живлення (граничне)      | 6-20В                             |
| Цифрові входи/виходи             | 14 (6 можуть використовувати ШІМ) |
| Аналогові входи                  | 6                                 |
| Макс. струм одного виходу        | 40 мА                             |
| Макс. струм виходу 3.3V          | 50 мА                             |
| Flash-пам'ять                    | 32 КБ (0.5 КБ для завантажувача)  |
| SRAM                             | 2 КБ                              |
| EEPROM                           | 1 КБ                              |
| Тактова частота                  | 16 МГц                            |

Для керування напругою на нитці розжарювання використовуємо МОН-транзистор, що дозволяє плавно керувати напругою. Головними параметрами при виборі для нас є максимальна напруга та струм витік-стік, порогова напруга на затворі для керування  $<5$  В, опір транзистора у відкритому стані, що відповідає за нагрів і втрати на ньому. Для нашої схеми підійдуть транзистори IRF3205.



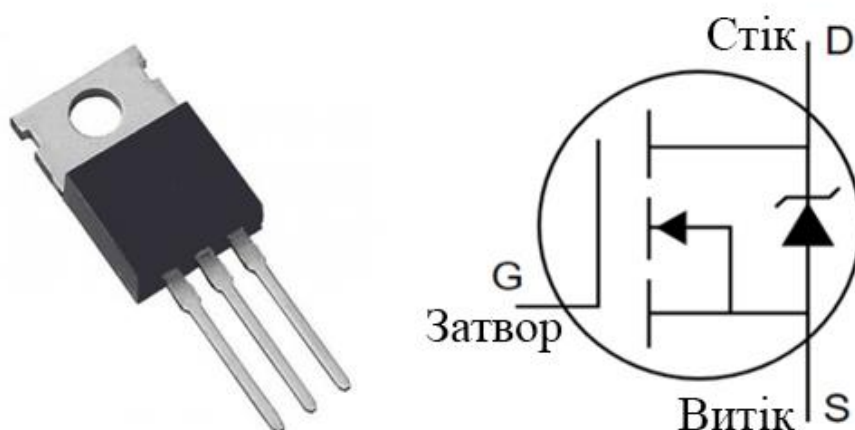


Рисунок 2.3 – Зовнішній вигляд транзистора IRF3205 і схематичне позначення [19]

Таблиця 2.2 – Основні характеристики транзистора IRF3205 [19].

|                                      |                      |
|--------------------------------------|----------------------|
| Максимальний струм витік-стік        | 110 А                |
| Максимальний напруга витік-стік      | 55 В                 |
| Порогова напруга на затворі          | 2...4 В              |
| Опір стік-виток у відкритому стані   | 8 мОм                |
| Мінімальна і максимальна температура | Від -55°C до +175 °C |

Для вимірювання температурного коефіцієнту опору необхідно використовувати термічні датчики. Для визначення температурного коефіцієнту опору металу необхідно мати досить точний температурний датчик з точністю вимірювання не менше  $\pm 0.5^{\circ}\text{C}$ . Також необхідна межа вимірювань до  $+120^{\circ}\text{C}$ . При вимірюванні в закритій камері нагрівання достатньо датчика температури TMP36.

Таблиця 2.3 – Основні характеристики датчика температури TMP36 [20].

|                                      |   |
|--------------------------------------|---|
| Діапазон вимірювань                  | Від -40°C до 150°C                      |
| Діапазон сигналу на вихід            | Від 0.1 В (-40°C) до 2.0 В (150°C)      |
| Точність вимірювання                 | $\pm 0.5^{\circ}\text{C}$               |
| Необхідне живлення                   | Від 2.7 В до 5.5 В, сила струму 0.05 мА |
| Мінімальна і максимальна температура | Від -55°C до +175 °C                    |



велика, необхідно час від часу проводити вимикання нагрівача. Це сповільнить швидкість його нагрівання і підвищить точність результатів, що одержуються.

5. Обробка результатів і розрахунок температурного коефіцієнту опору  $\alpha$  за формулою 2.3
6. Проведення аналізу та підведення висновків, порівняння результатів з літературними даними, аналіз похибок та факторів, що впливають на вимірювання, висновки щодо відповідності отриманих даних теоретичним підрахункам [16,18].

В якості середовища моделювання і проектування схеми для вимірювання температурного коефіцієнту використовуємо веб-сайт Autodesk Tinkercad. Tinkercad - це популярний пакет програмного забезпечення для 3D-моделювання, моделювання електричних схем і блокового кодування, який доступний безкоштовно через веб-браузер. Головною перевагою є його зручність і простота. Він складається з трьох розділів, кожен з яких можна вважати окремим середовищем та використовується для різних цілей. У нашій роботі буде використано середовище з моделювання електричної схеми з Arduino і написання коду.

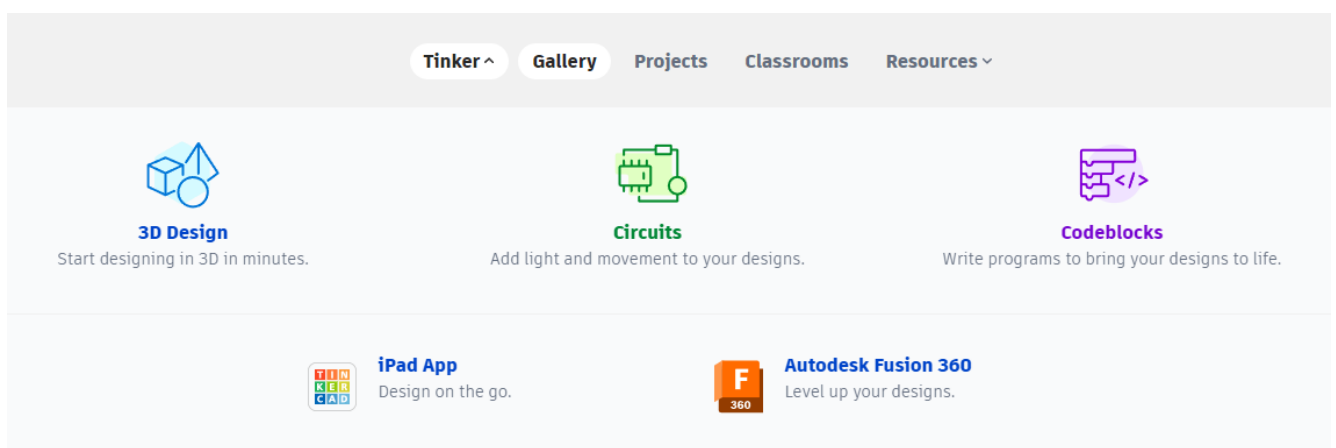


Рисунок 2.5 – Меню сайту Tinkercad з середовищами розробки [20]

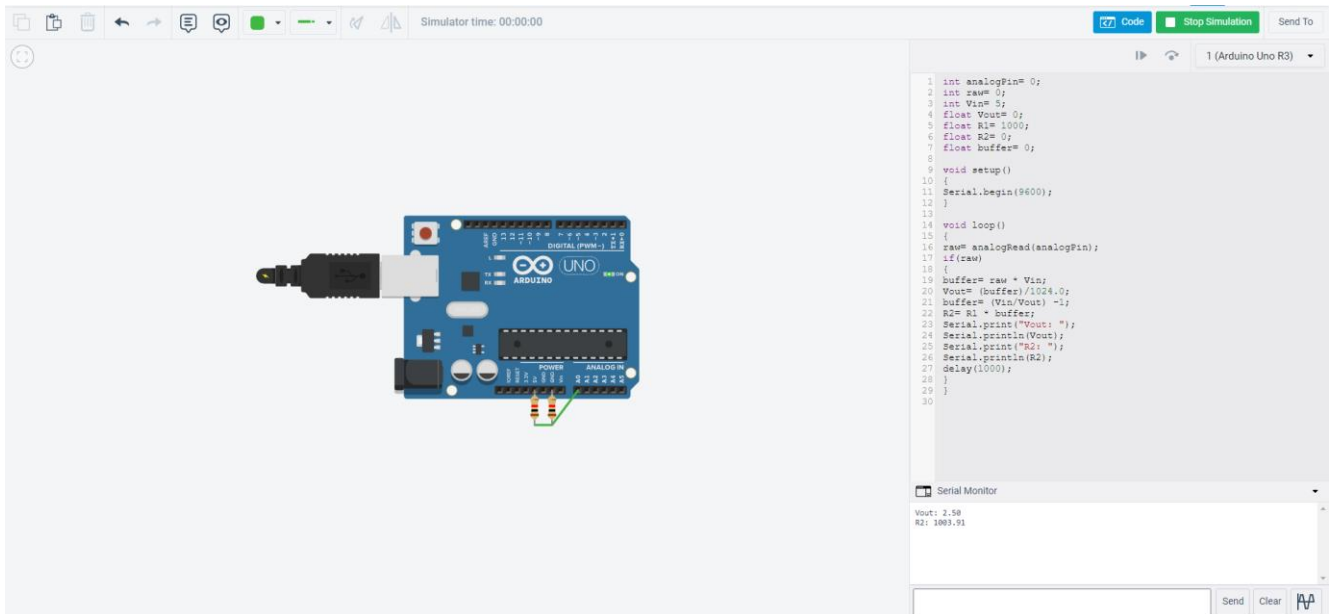


Рисунок 2.6 – Середовище розробки схеми і простір для написання програмного коду з консоллю [21]

Іншою важливою перевагою є можливість швидкого перегляду вже існуючих схем від інших користувачів. Також можлива публікація із захистом прав автора, та редагування схем, якщо це дозволив автор. Таким чином середовище Tinkercad є гарним способом попереднього моделювання нашої схеми для вимірювання температурного коефіцієнту опору.

## РОЗДІЛ 3

### ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

#### 3.1. Принцип дії установки

Для початку роботи автоматичної системи збору даних для розрахунку коефіцієнту термічного опору на Arduino, необхідно підключити датчики опору та датчика температури до Arduino. Далі, система повинна запускатись автоматично кожні 5 секунд і проводити вимірювання показників датчиків опору та температури. Отримані дані про опір та температуру використовуються для розрахунку коефіцієнта термічного опору, що є важливим показником в розрахунках електричних систем та пристроїв. Після отримання даних, система повинна підвищувати на 1 В напругу котушки розжарювання. Це необхідно для поступового збільшення температури і тим самим підвищення температури вимірюваному металі і зміни його опору.

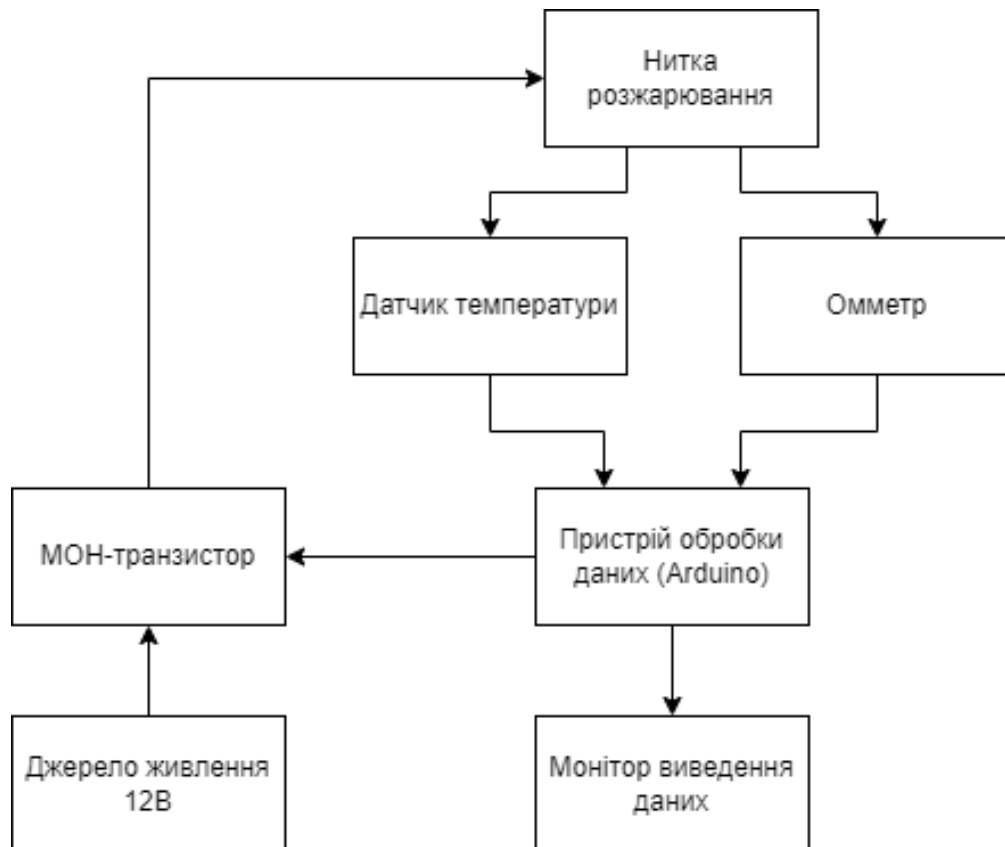


Рисунок 3.1 – Структурна схема конструкції системи для визначення ТКО

Для забезпечення стабільної та точної роботи системи, необхідно врахувати різноманітні фактори, такі як джерело живлення, температурні умови, точність датчиків та програмного забезпечення, а також виконати належне калібрування системи. Реалізація поступового підвищення температури реалізується через зовнішнє джерело живлення 12 В та МОН-транзистор. Транзистор підключений до 9го входу, це необхідно для використання ширини імпульсної модуляції (ШІМ). ШІМ - це техніка, яка використовується для управління аналоговими сигналами за допомогою цифрових сигналів. ШІМ часто використовується, як метод управління елементами що вимагають постійного струму, але при цьому дозволяють змінювати силу струму, що йде через прилад. ШІМ генерується шляхом швидкого перемикання високої напруги в певному діапазоні частот. Наприклад, при ШІМ на 50% напруга буде включена на половину часу, а вимкнена на іншу половину часу.

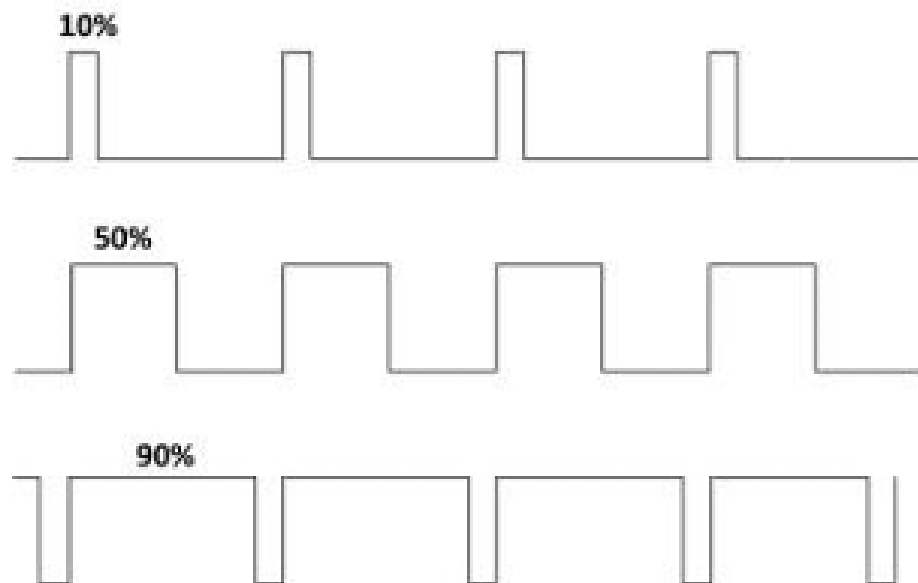


Рисунок 3.2 – Зображення ШІМ імпульсів із різними степенями заповненості [23]

Цей періодичний включено-вимкнений сигнал має меншу середню енергію порівняно з постійним сигналом, що дозволяє зменшити теплові втрати та підвищити ефективність роботи пристроїв. В Arduino ШІМ генерується на основі таймерів, які використовуються для генерації періодичного сигналу з визначеною частотою та шириною імпульсу. Під час переходу від увімкненого стану до

вимкненого значення струму і напругу не повністю нульові. Тому вони розсіюються на перемикачах. Контролер Arduino забезпечує зміну ширини імпульсу згідно з вхідним сигналом управління. Це дозволяє змінювати середнє значення напруги і керувати енергією, що постачається до підключених до плати пристроїв [23].

Arduino з мікроконтролером ATmega328 має так ШІМ 8-бітні виходи: 3,5,6,9,10,11. Таким чином ці виходи можуть видавати 256 значень напруги для керування. МОН-транзистор використовується для керування потужністю. У нашій системі використовується N-канальний транзистор через його більші можливості і запас потужності по керуванню навантаженням. Для поетапного збільшення навантаження необхідно використовувати ШІМ з мікроконтролера. Потрібно пам'ятати, що МОН має свою ємність і в момент перемикання на затворі є великі струми, що можуть пошкодити контролер. Резистор на 1 кОм або 100 Ом виконують захисну функцію від ємнісної дії транзистора, а резистор обмежує струм між виходом Arduino та затвором. Резистор на 10 кОм виконує захисну функцію від випадкового від'єднання від мікроконтролера, або обриву проводу. При такій ситуації затвор може відкритися від струмів наводки, або під'єданого джерела живлення [22-24].

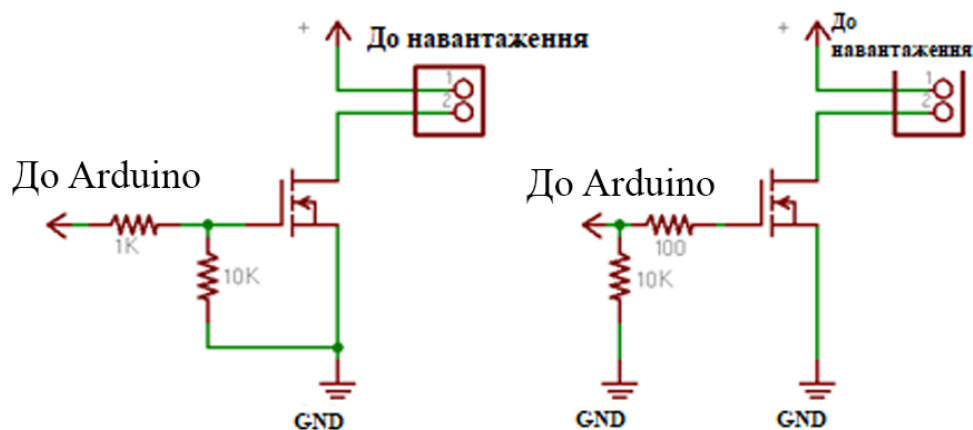


Рисунок 3.3 – Схеми підключення МОН у режимі ключа. Адаптовано з праці [23]

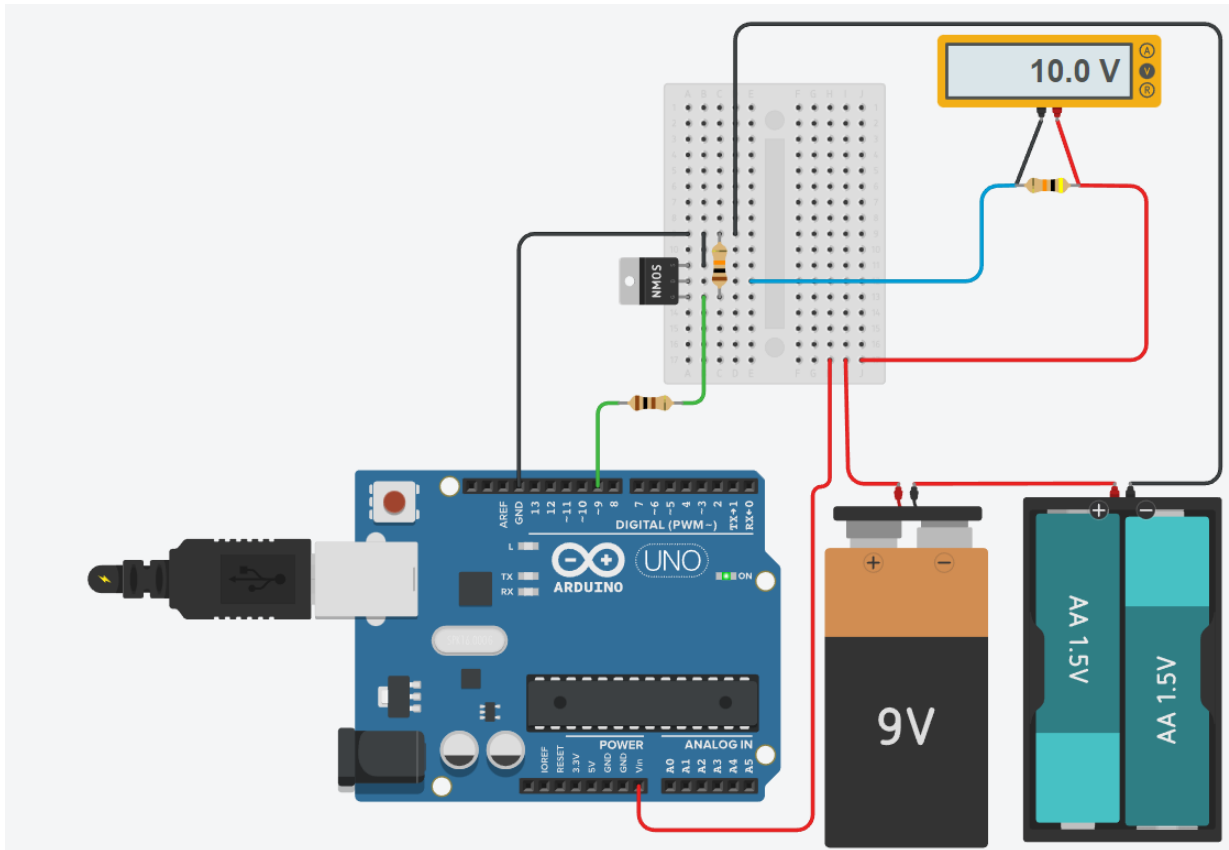


Рисунок 3.4 – Макет схеми для керування поступовим збільшенням напруги від 0 до 12 В на основі МОН-транзистору

Схема для збору даних і розрахунку ТКО металу на основі Arduino у зборі налічує такі компоненти:

- Arduino Uno R3
- Джерело живлення 12 В
- Макетна плата
- 3 резистори (100 Ом, 10 кОм, 15 кОм)
- Нитка розжарювання
- N-канальний МОН-транзистор IRF3205
- Датчик температури
- Омметр(мультиметр)

Через особливості середовища моделювання нитка розжарювання реалізується за допомогою резистора 40 кОм, також через неможливість прямого



підключення мультиметру до Arduino, вимірювання показників опору здійснено через потенціометр із змінним опором 40 кОм. Для проведення моделювання необхідно в процесі збільшувати показник потенціометру, що моделює збільшення опору внаслідок зростання температури на нитці розжарювання. Таким чином загальна схема системи автоматизованого вимірювання ТКО має такий вигляд:

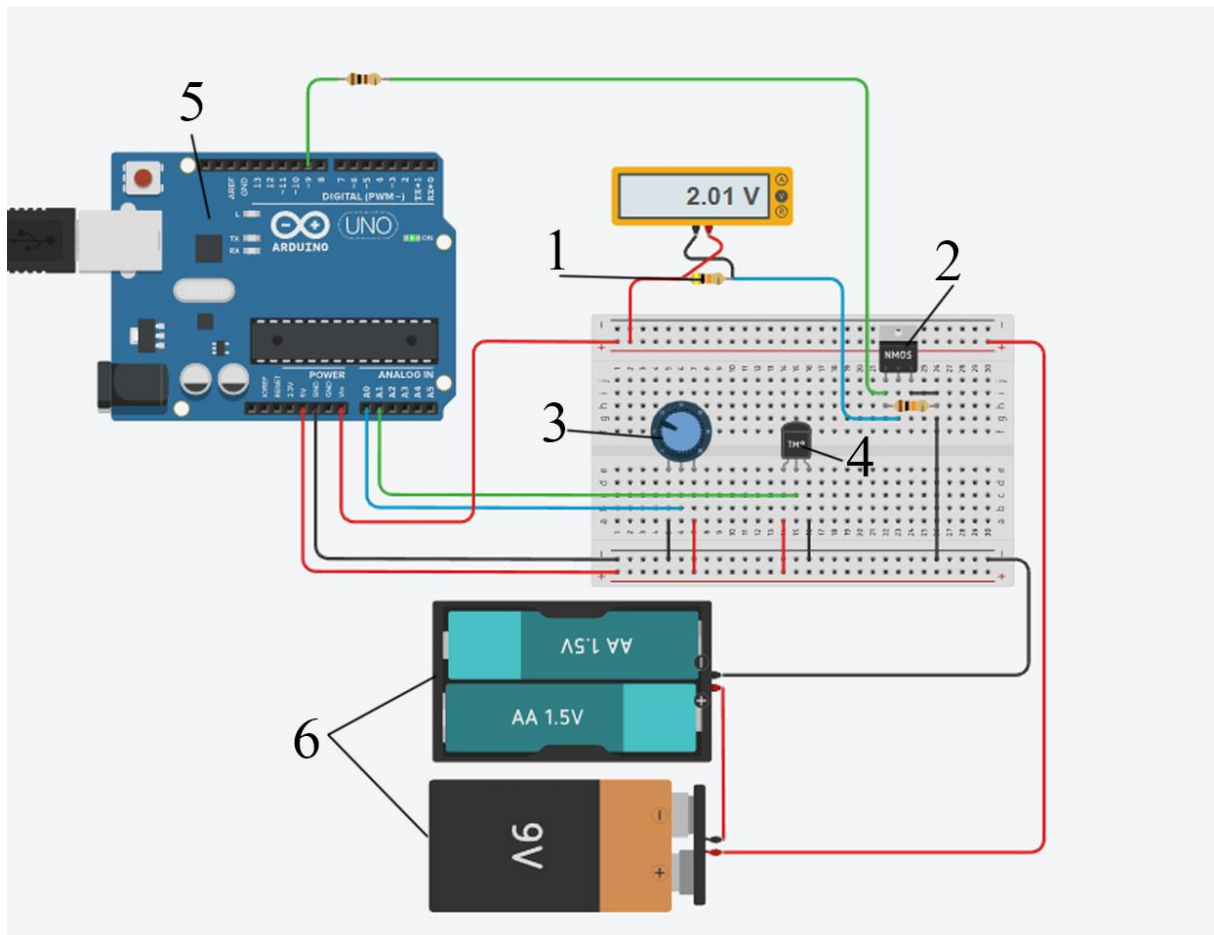


Рисунок 3.5 – Макет схеми для вимірювання ТКО: 1 – нитка розжарювання, представлена резистором 40 кОм; 2 – МОН-транзистор, що керує напругою ниткою розжарення; 3 – Омметр/мультиметр, що представлений потенціометром, 4 – датчик температури; 5 – Arduino UNO R3; 6 – джерело живлення 12 В

### 3.2. Написання програмного коду для системи вимірювання ТКО

Код для Arduino пишеться на мові програмування C++. Як вже було розглянуто, Arduino використовує мікроконтролери які мають архітектуру Harvard, тобто окремі шини для програмної та даних пам'яті. Це означає, що для програмування мікроконтролера на Arduino потрібен спеціальний компілятор, який здатен перетворити код, написаний на C++, в машинний код, який може бути виконаний мікроконтролером.



Рисунок 3.6 – Блок-схема алгоритму роботи програми

Код пишеться в Arduino IDE, інтегроване середовище розробки, яке дозволяє легко створювати, збирати та завантажувати програми на Arduino. В Arduino IDE є вбудований текстовий редактор, консоль для виводу повідомлень та інструменти для компіляції та завантаження програми на мікроконтролер. Особливістю є те, що мікроконтролери мають обмежену кількість оперативної та програмної пам'яті, то програми для Arduino мають бути оптимізовані та написані з урахуванням обмежень мікроконтролера. Для зменшення використання пам'яті зазвичай використовуються бібліотеки, які містять готові функції для певних операцій, таких як взаємодія з датчиками або керування сервоприводами. Програма має декілька структурних блоків через особливості роботи Arduino

- директиви препроцесора
- оголошення змінних, констант, масивів
- ініціалізація пристроїв
- цикл виконання основної програми

У нашому середовищі вже включені `#include` препроцесори необхідні для роботи з Arduino, директива `#define` не використовується, бо не потрібна текстова підміна. Наступним є оголошення глобальних змінних. Змінні типу `const` використовуються для призначення виходів датчиків і кількості ітерацій вимірювання, так як їх не потрібно змінювати впродовж програми. Вид змінних `int` для цілочисельних змінних, таких як лічильники, кількість рядків масиву, крок напруги(за необхідності може бути змінений на `float`). `Float` використовується для зняття показників з плаваючою комою, таких як дані датчиків, клітинки масиву необхідні для запису вимірянних і розрахованих даних.

```
const float SENSOR1_PIN = A0; // Перший датчик
const int SENSOR2_PIN = A1; // Другий датчик
const int NUM_READINGS = 10; // Кількість записів
float readings[NUM_READINGS][4]; // Масив із 4х рядків
int index = 0;
int transistorPin = 9; // вихід підключення бази транзистора
int voltageStep = 1; // шаг зміни напруги на нитці розжарювання
```

```
unsigned long lastTime = 0;
```

Таблиця 3.1 - Таблиця змінних і пояснення

| Символьні імена                             | Пояснення                                     |
|---|---|
| NUM_READINGS                                | Кількість ітерацій зчитування даних           |
| readings[NUM_READINGS][4]                   | Масив запису даних із n рядків і 4 стовпчиків |
| index                                       | Лічильних ітерації                            |
| voltage                                     | Величина напруги                              |
| voltageStep                                 | Шаг зміни напруги                             |
| lastTime                                    | Останній фіксований час у ітерації            |
| SENSOR1_PIN<br>SENSOR2_PIN<br>transistorPin | Надання змінним значення виходу на платі      |

Блок ініціалізації пристроїв необхідний для налаштування виходів Arduino. Таким чином він набуває двох значень: INPUT – на прийом даних із датчиків, OUTPUT – на відсилання даних через ШІМ до транзистору для збільшення напруги на нитці розжарювання.

```
void setup() {
  Serial.begin(9600);
  pinMode(SENSOR1_PIN, INPUT); // Налаштування пінів на прийом даних
  pinMode(SENSOR2_PIN, INPUT);
  pinMode(transistorPin, OUTPUT); // Встановлення піна на базу транзистора
}
```

Головним циклом є void loop, що повторювано виконує основну програму системи. Його особливістю є можливість для входу і не повертає ніякі дані та не має точки виходу, а також не закінчується до примусового закінчення програми. Код будемо розглядати по окремим блокам для простоти аналізу

```

void loop() {
    static int voltage = 0; // зміна, що зберігає значення напруги
    // Перевірка на запис даних кожні 5 секунд
    if (millis() - lastTime >= 5000) {
        // Оновлення часу
        lastTime = millis();
        // Переназначаємо напругу виходу на базу транзистора
        analogWrite(transistorPin, map(voltage, 0, 12, 0, 255));
        // збільшуємо напругу на заданий шаг
        voltage += voltageStep;
    }
}

```

Використовуємо статичну змінну `static` для того, щоб змінна зберігалася, навіть, при виході із блоку, у нашому коді використовується для зберігання параметру напруги котушки розжарювання. Оператор `if` використаний для того, щоб подальший цикл виконання програми запускався тільки кожні 5 секунд, за бажанням, цей параметр може бути змінений. Формує задану аналогову напругу на виведенні у вигляді ШІМ-сигналу. Останнім є збільшення значення напруги на нитці на заданий шаг.

```

// Зняття даних
float Vout = (5.0 / 1023.0)* analogRead(SENSOR1_PIN);
float sensor1 = (15000*Vout)/5;
int sensor2 = ((5.0 / 1024.0)* analogRead(SENSOR2_PIN)-0.5)*100+273;
// Запис часу даних і суми в масив
readings[index][0] = millis()/1000;
readings[index][1] = sensor1;
readings[index][2] = sensor2;
float sum = 100*(readings[index][1]-readings[index-1][1])/(readings[index-1][1]*(readings[index-1][2]-readings[index][2]));
readings[index][3] = sum;

```

Даний блок відповідає за розрахунок і переведення отриманої напруги від датчиків до фізичного числового значення. Також призначаються значення по

рядку масива, перший стовпчик – час, другий – дані опору, третій – температура, четвертий – розрахований температурний коефіцієнт опору.

```
// Збільшення індексу для переходу на іншу строку масива
index++;
if (index >= NUM_READINGS) {
    index = 0;
}
// Збільшення напруги на нитці розжарювання до 12 і зменшення до 0
if (voltage >= 12) {
    voltageStep = -1;
}
if (voltage <= 0) {
    voltageStep = 1;
}
```

Після завершення запису відбувається збільшення `index`, що відповідає переходу на новий рядок масива. При виконанні умови, що значення напруги ще не більше 12, відбувається збільшення значення на задану кількість. Останній блок відповідає за вивід тексту позначень стовпчиків і потім значень масиву, під кінець оновлюється час у змінній.

```
// Вивід масиву
Serial.println("Time(s)   R(kOm)   T(K)   Temp.koef(10^-3)");
for (int i = 0; i < NUM_READINGS; i++) {
    Serial.print(readings[i][0]);
    Serial.print("   ");
    Serial.print(readings[i][1]);
    Serial.print("   ");
    Serial.print(readings[i][2]);
    Serial.print("   ");
    Serial.println(readings[i][3]);
}
```

```

    }
    Serial.println();
  }
}

```

### 3.3. Аналіз і тестування програми

Написаний код для визначення термічного коефіцієнту опору металу на мікроконтролері Arduino використовує схему з аналоговим датчиком температури LM35 і омметром. Схема є модульною і точність вимірювань залежать напряду від датчиків, тому за необхідністю їх можна замінити. Також можливе під'єднання інших модулів, таких, як LCD дисплей. Код складається з основного циклу, який проводить замір опору при зміні температури. Програма використовує масив для зберігання прочитаних даних, а також використовує транзистор для контролю напруги на нитці розжарювання. На виводі на моніторі порту серійного з'єднання виводяться дані у форматі час (секунди), опір (кОм), температура (кельвіни), температурний коефіцієнт ( $10^{-3}$ ). Дана програма може бути корисною для вимірювання температурного коефіцієнту опору різних матеріалів, які змінюють свій опір залежно від температури, наприклад, термісторів. Проведемо симуляцію експерименту з визначення ТКО за лабораторними даними:

Таблиця 3.2 – дані для визначення ТКО

|        |     |     |     |     |     |     |     |
|--------|-----|-----|-----|-----|-----|-----|-----|
| T, К   | 298 | 303 | 308 | 313 | 318 | 323 | 328 |
| R, кОм | 59  | 60  | 61  | 62  | 63  | 64  | 65  |

Для даної симуляції було змінено параметри потенціометра на максимальне значення 70 кОм, для більш точного встановлення опору на досліджуваному металі. Також необхідно зробити зміни у кодї, для того, щоб програма правильно працювала з даними параметрами схеми:

```
float sensor1 = (70*Vout)/5;
```

де 70 – значення максимального опору на потенціометрі в кОм.

Провівши експеримент ми отримали такі дані:

| Time(s) | R(kOm) | T(K)   | Temp.koef(10 <sup>-3</sup> ) |
|---------|--------|--------|------------------------------|
| 5.00    | 58.44  | 297.00 | inf                          |
| 10.00   | 59.87  | 304.00 | 0.35                         |
| 15.00   | 61.24  | 308.00 | 0.57                         |
| 20.00   | 62.61  | 312.00 | 0.56                         |
| 25.00   | 63.98  | 317.00 | 0.44                         |
| 30.00   | 65.35  | 323.00 | 0.36                         |
| 35.00   | 66.72  | 327.00 | 0.52                         |
| 40.00   | 68.15  | 331.00 | 0.54                         |
| 45.00   | 69.52  | 336.00 | 0.40                         |

Рисунок 3.7 – Результати симуляції у середовищі Tinkercad

За результатами експерименту, можемо порівняти значення температурного коефіцієнту з табличним значенням. Невідомим металом є мідь, з ТКО  $\approx 0.43 \cdot 10^{-3}$ . Програма працює правильно і дозволяє проводити визначення ТКО. Головним недоліком симуляції є відсутність підключення мультиметру до Arduino і необхідність піднімання параметрів на датчиках вручну, через що падає точність симуляції через людський фактор. Але слід зазначити, що цей факт ніяк не вплине на зібрану схему, так як усі дані будуть збиратися автоматично і потребують від користувача лише запуску програми і правильного збору схеми. Ця програма може бути використана для вимірювання температурного коефіцієнту опору металу в лабораторних умовах. Використовуючи цей код, можна збільшити точність вимірювання за рахунок автоматизації процесу збору даних та обробки. Застосування мікроконтролера Arduino для збору та обробки даних може бути особливо корисним для вимірювання параметрів в реальному часі та забезпечення автоматичного контролю над параметрами.



## ВИСНОВКИ

1. Розглянуто історію та загальні відомості щодо мікроконтролерів типу Arduino, їх характеристики. На основі Arduino UNO R3 розглянута система роботи пам'яті Гарвард і блок-схему роботи чипу ATmega328P.

2. Проаналізовано використання платформи Arduino для використання, як системи збору даних у фізичних експериментах. Головною перевагою є доступність створення будь-яких платформ, невелика ціна, можливість швидкого перемодельювання і розробки та попереднього моделювання у веб-середовищах. Отже, можна зробити висновок, що використання Arduino для збору та обробки даних у фізичних експериментах є ефективним і зручним інструментом, що дозволяє швидко і точно збирати та обробляти дані, а також підвищує точність і повторюваність експериментів.

3. Проаналізована методологія була проаналізована методологія визначення параметру температурного коефіцієнту опору та його фізичні принципи виникнення. Було розглянуто основні теоретичні підходи до визначення властивостей матеріалу під дією температури, а також проведено дослідження практичного застосування Arduino для збору даних у фізичних експериментах.

4. Було створено систему автоматичного збору та обробки даних на основі Arduino Uno R3 та датчику температури і мультиметру для визначення температурного коефіцієнту опору. Система автоматично вимірює показники, розраховує ТКО і поступово збільшує напругу на нитці розжарювання для збільшення температури.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Monk S. Programming arduino: getting started with sketches, 3rd Edition: McGraw-Hill Education, 2022. 176 p.
2. Pearson D. Arduino: Beginner's Guide to Learn Arduino Programming Step by Step: Independently published, 2019. 61 p.
3. Banzi M., Shiloh M. Getting started with arduino: the open source electronics prototyping platform: Maker Media, Incorporated, 2021. 256 p.
4. Isizoh A.N. and other. Design and Construction of Arduino-Based Experimental Training Kit // International Journal of Research Publication and Reviews. 2022. Vol 3, no 4. P. 389-399.
5. Thorpe E. Arduino: simple and effective strategies to arduino programming: Independently Published, 2019. 228 p.
6. Arduino memory guide. Arduino Documentation. URL: <https://docs.arduino.cc/learn/programming/memory-guide#memory-architectures-101> (date of access: 21.04.2023).
7. Pereira, P.F.; Ramos, N.M.M. Low-Cost Arduino-Based Temperature, Relative Humidity and CO2 Sensors-An Assessment of Their Suitability for Indoor Built Environments // J. Build. Eng. 2022. Vol. 60. P. 105151.
8. Z. Li et al. Design of office intelligent lighting system based on arduino // Procedia computer science. 2020. Vol. 166. P. 134–138.
9. Vallejo W., Diaz-Uribe C., Fajardo C. Do-it-yourself methodology for calorimeter construction based in Arduino data acquisition device for introductory chemical laboratories // Heliyon. 2020. Vol. 6, no. 3. P. e03591.
10. Lay K. S., Li L., Okutsu M. High altitude balloon testing of Arduino and environmental sensors for CubeSat prototype // HardwareX. 2022. Vol. 12. P. e00329.
11. P. Bora. Smart real time health monitoring system using Arduino and Raspberry Pi // Materials today: proceedings. 2021. Vol. 46. P. 3855–3859.

12. Kumar Sai K. B., Mukherjee S., Parveen Sultana H. Low cost iot based air quality monitoring setup using arduino and MQ series sensors with dataset analysis // Procedia computer science. 2019. Vol. 165. P. 322–327.
13. J. Cannon. Applications of low-cost environmental monitoring systems for fine-scale abiotic measurements in forest ecology // Agricultural and forest meteorology. 2022. Vol. 321. P. 108973.
14. Wirarespati A., Rasjid Z. E. Automotive security with authorization and tracking via GPS // Procedia computer science. 2019. Vol. 157. P. 72–78.
15. Л.Д. Дідух. Електрика та магнетизм: підручник. Тернопіль : Підручники і посібники, 2020. 464 с
16. К.С. Дрозденко. Фізичні основи електроніки: курс лекцій: навч. посіб. для студ. спеціальності 171 «Електроніка». Київ: КПІ ім. Ігоря Сікорського, 2021. 153 с.
17. Бойко О. В. Функціонально-інтегровані сенсори термічного аналізу на структурах твердотільної електроніки : автореф. дис на здобуття наук. ступеня канд.-мат. наук: спец. 05.27.01 - Твердотільна електроніка. Львів, 2019. 40 с.
18. Скіцько, І. Ф. Фізика. Практикум: навчальний посібник 2-ге вид., перероб. і доп. Київ : КПІ ім. Ігоря Сікорського, 2020. 614 с.
19. IRF3205 pdf, IRF3205 Description, IRF3205 Datasheet, IRF3205 view. ALLDATASHEET.COM - Electronic Parts Datasheet Search. URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/68131/IRF/IRF3205.html> (date of access: 25.06.2023).
20. TMP36 pdf, TMP36 Description, TMP36 Datasheet, TMP36 view. ALLDATASHEET.COM - Electronic Parts Datasheet Search. URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/49108/AD/TMP36.html> (date of access: 25.04.2023).
21. Tinkercad. From mind to design in minutes. URL: <https://www.tinkercad.com/> (date of access: 01.05.2023).

22. Sinisalo H., Nieminen J., Ilmoniemi R. Waveform simulation and pulse-width-modulation approximations with multi-locus tms // Brain stimulation. 2021. Vol. 14, no. 6. P. 1634.
23. І.А. Дичка, В.П. Тарасенко, М.В. Онай. Основи прикладної теорії цифрових автоматів: підручник. Київ: КПІ ім. Ігоря Сікорського, 2019. 508 с.
24. Arduino High-Current Interfacing - Transistors & MOSFETs. DroneBot Workshop. URL: <https://dronebotworkshop.com/transistors-mosfets/> (date of access: 02.05.2023).