

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
імені ВОЛОДИМИРА ДАЛЯ  
(м. Київ)

Факультет Інформаційних технологій та електроніки  
(повне найменування факультету)  
Кафедра Електронних апаратів  
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
до дипломної роботи

освітньо-кваліфікаційного рівня магістр  
(бакалавр, спеціаліст, магістр)  
спеціальності 171 – Електроніка  
(шифр і назва спеціальності)

на тему **Пристрій для контролю температури технологічного процесу в  
хімічній промисловості**

Виконала: студент групи Ел-23дм  
Денисов  
(прізвище, та ініціали) (підпис)

Керівник Тюндер І.С.  
(прізвище та ініціали) (підпис)

Завідувач кафедри Паєранд Ю.Е.  
(прізвище та ініціали) (підпис)

Рецензент Самойлова Ж.Г.  
(прізвище та ініціали) (підпис)

Київ - 2023

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
імені ВОЛОДИМИРА ДАЛЯ  
(м. Київ)

Факультет Інформаційних технологій та електроніки  
Кафедра Електронних апаратів

Освітньо-кваліфікаційний рівень магістр (бакалавр, спеціаліст, магістр)  
Спеціальність 171 – Електроніка  
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ЕА  
Ю.Е. Паеранд  
“ ” 2023 року

З А В Д А Н Н Я  
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА  
Денисов Олег Сергійович  
(прізвище, ім'я, по батькові)

1. Тема роботи Пристрій для контролю температури технологічного процесу для хімічної промисловості

керівник проекту (роботи) Тюндер Ірина Сергіївна, старший викладач  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від 21.11.2023 року № 52/14.04

2. Строк подання студентом проекту (роботи) 05.12.2023

3. Вихідні дані до роботи Матеріали науково-дослідної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) Аналітичний огляд

2) Розробка пробної схеми

3) Вибір мікроконтролера для майбутнього приладу

4) Створення принципіальної схеми приладу

5) Експериментальні дослідження

5. Дата видачі завдання 01.09.2023

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Загальний розгляд питань і завдань	01.09.23 - 02.09.23	
2	Види датчиків температури	03.09.23 - 10.10.23	
3	Розробка схеми приладу	11.10.23 - 25.10.23	
4	Код тестової програми	26.10.23 - 27.10.23	
5	Проблеми переходу з плат ARDUINO	28.10.23- 30.10.23	
6	Принципова схема приладу та її опис	01.11.23- 10.11.23	
7	Написання коду основної програми	11.11.23 - 25.11.23	
8	Різниця у розключенні між платою ARDUINO та мікроконтролеру	26.11.23 - 27.11.23	
9	Оформлення пояснювальної записки	29.11.22 - 30.11.22	
10	Передача роботи на перевірку та рецензування	01.12.23 - 05.12.23	

Студент \_\_\_\_\_ Денисов О. С. .  
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) \_\_\_\_\_ Тюндер І.С.  
(підпис) (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка: 59 сторінок, 27 рисунків, 1 таблиць, 9 посилання.

**Тема наукової роботи:** «Пристрій для контролю температури технологічного процесу в хімічній промисловості».

**Об'єктом дослідження** є розробка приладу для контролю температури технологічного процесу.

**Мета роботи** – полягає в розробці приладу для контролю температури який був би сумісний з приладами для автоматизації, які вже використовуються у технологічних процесах.

**В процесі роботи** проведено аналітичний огляд існуючих методів заміру температури, існуючих методів передачі даних, представлена принципова схема приладу, а також код програми для роботи приладу.

**В результаті роботи** запропоновано прилад, що може використовуватись для роботи у технологічних процесах хімічної промисловості, який сумісний з вже існуючими методами обробки даних, візуалізації та автоматизації технологічного процесу.

Робота виконана за період навчання в магістратурі з 01 серпня 2023 року по 05 грудня 2023 року на кафедрі «Електронних апаратів» (завідувач кафедри, професор, канд. техн. наук Ю.Е. Паеранд).

## ЗМІСТ

Вступ	6
1. Види датчиків температури	7
1.1 Види промислових передач даних	13
1.2 Висновок вибору датчику температури та методу передачі даних	16
1.3 Промислові типи датчиків температури	20
2. Розробка схеми приладу	23
2.1 Підбір компонентів для приладу	23
2.2 Вибір мікроконтролеру та платформи	26
2.2.1 Створення/розробка пробної схеми	29
2.2.2 Код тестової програми	31
2.2.3 Проблеми переходу з плат ARDUINO	33
2.3 Підбір елементів живлення для приладу	35
2.3.1 Вибір мікросхеми для розробки послідовного інтерфейсу RS-485	37
2.3.2 Принципова схема приладу та її опис	39
2.4 Код на ARDUINO C для приладу	43
2.4.1 Підготовка мікроконтролеру під мову ARDUINO C	43
2.4.2 Написання коду основної програми	47
2.5 Різниця у розключенні між платою ARDUINO та мікроконтролеру	52
3 Практичне використання приладу	54
4 Можливе використання приладу	57
Висновок	58
Література	59

## ВСТУП

На сьогоднішній день, визначення температури є для нас повсякденням, ми бачимо показники температури, коли дивимось зранку на термометр, що знаходиться у нас на вікні за для упевненості, що одягти нам перед виходом на вулицю, або коли дивимось прогноз погоди на завтра. Для заміру температури використовуємо різні методи, це може бути термометр, або датчик температури, що заміряє температуру, але суть у цих приладах одна, поміряти температуру навколишнього середовища, та передати ці дані в вигляді цифр для обробки користувачу, тобто нам.

Датчики температури зайняли широку нішу у нашому житті, це як вище вже розказувалось, замір температури навколишнього середовища для метеостанцій та повсякденного життя, це може бути термометр для заміру температури тіла, які вже приходять у наше життя на заміну ртутних термометрів що були до цього, також для заміру температури у промисловості, починаючи заміром двигуна під час роботи, та його токоведучих частин, закінчуючи судами під тиском у хімічній промисловості, металургії, харчової промисловості, для контролю та управління процесами. Окрім цього датчики температури використовують навіть для визначення косвених параметрів, наприклад тиску, тиск та температура пов'язані та якщо виходить з ладу датчик тиску оператор може оперуватись на температуру, як на датчик тиску, окрім датчика рівня, також використовуються датчики температури в так званих датчиках потоку, суть такого датчику є замір витрати рідини, зазвичай води, де менше температура, тим сильніше вважається витрата води, хоча не дає точну витрату рідини. Тому на сьогоднішній день є потреба не тільки в вимірюванні температури, а ще й з універсальною передачею вимірювальних даних. Окрім цього прилад було легко повторити для комерційного використання чи побутового використання.

## 1 ВИДИ ДАТЧИКІВ ТЕМПЕРАТУРИ

Термометри - це найпоширеніші датчики температури, що зустрічаються при простих повсякденних вимірах температури. Два приклади термометрів - це заповнена система та біметалічні термометри.

### Заповнений термометр системи

Звичний рідкий термометр складається з рідини, укладеної в трубку. Обсяг рідини змінюється в залежності від температури. Збільшення молекулярного руху зі збільшенням температури змушує рідину розширюватися і переміщатися по каліброваним маркуванням з боку трубки. Рідина повинна мати відносно великий коефіцієнт теплового розширення, щоб невеликі зміни температури призвели до помітних змін об'єму. Загальним матеріалом трубки є скло, а звичайна рідина - спирт. Ртуть раніше була більш поширеною рідиною, поки не була реалізована її токсичність. Хоча термометр із заповненою системою є найпростішим і дешевим способом вимірювання температури, його точність обмежується калібрувальними відмітками по довжині трубки. Оскільки заповнені системні термометри зчитуються візуально і не виробляють електричних сигналів, їх важко реалізувати в управлінні процесом, які значною мірою покладаються на електричне та комп'ютеризоване управління.

На відміну від термометра, термометра супротиву і термопари, пірометри (безконтактні датчики температури) вимірюють кількість випромінюваного тепла, а не кількість тепла, проведеного і конвекційного до датчика. Існують різні типи пірометрів, такі як загальне випромінювання та фотоелектричні пірометри. Нижче наведена схема оптичного пірометра на рисунку 1.

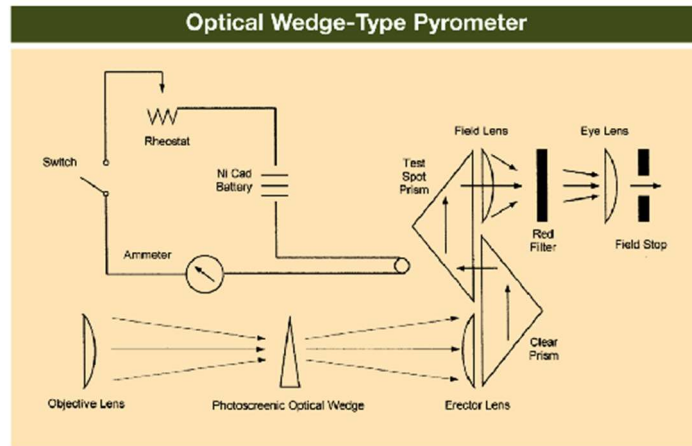


Рис 1 -Принципова схема оптичного пірометра

Ці пірометри відрізняються за типом випромінювання, яке вони вимірюють. Існує багато факторів, які впливають на кількість виявленого випромінюваного тепла, тому існує багато припущень, які необхідно зробити щодо випромінювальної здатності або міри способу випромінювання тепла об'єкта. Ці припущення ґрунтуються на способі випромінювання тепла, а також геометрії об'єкта. Оскільки температура залежить від випромінювальної здатності тіла, ці припущення щодо випромінювальної здатності вносять невизначеності та неточності в показаннях температури. Тому через помилки, пов'язаної з ними, пірометри не часто використовуються в промисловості.

Існує кілька різних видів пірометрів. Розрізняють оптичні і радіаційні пірометри.

Як працюють оптичні пірометри:

Порівнює колір видимого світла, що виділяється об'єктом, з кольором електрично нагрітого дроту

Провід може бути попередньо встановлений на певну температуру

Провід можна регулювати вручну, щоб порівняти два об'єкти

Як працюють радіаційні пірометри:

Цей датчик працює шляхом вимірювання випромінювання (інфрачервоного або видимого світла), яке виділяє об'єкт

Випромінювання нагріває термопару в пірометрі, яка, в свою чергу, індукує струм.



Чим більше індукований струм, тим вище температура

Пірометри зазвичай використовуються при дуже високих температурах, але можуть використовуватися і при більш холодних температурах. Існує безліч промислових застосувань для пірометрів. Оператори заводу можуть використовувати пірометри, щоб зрозуміти, при якій температурі працюють певні процеси. Мінусом пірометрів є те, що вони не дуже точні, як термопари або термометрів супротиву. Це тому, що вони покладаються на кількісне визначення кольорів світла.

У нашому повсякденному житті ми навіть не даємо оцінку видам датчиків температури, що використовуємо діляться на декілька груп:

1. Основані на ефекті Зеєбека
2. На зміні супротиву матеріалу
3. Напівпровідники
4. Мікросхеми

В основі ефекту Зеєбека полягає різномірність матеріалів, які спаяні між собою, таким чином місце спайки називається гарячим спаєм, а на кінцях такого пристрою утворюється ЕДС у вигляді мВ/°С та називається холодним спаєм, цей пристрій називається термопарою (рис. 2), та використовується у промисловості для заміру високих температур, так як на цих температурах термопари найбільш точні. Але недоліками термопар є сам холодний спай, який видає ЕДС і на термопарі до приладу потрібно до самого холодного спаю, що приводе до проблем підключення, це вирішили за допомогою компенсаційного провідника, що зроблений з так самого матеріалу, що термопара, окрім цього на холодному спаї, для зменшення похибки підчас заміру почали використовувати допоміжний датчик температури.

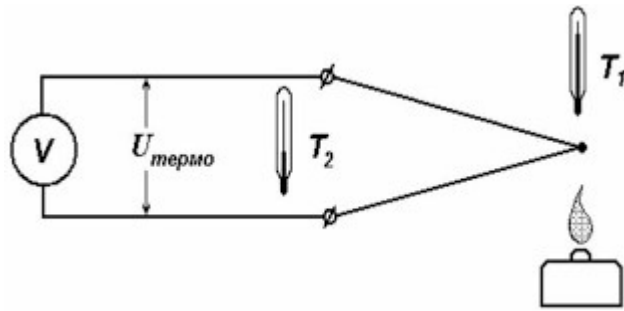
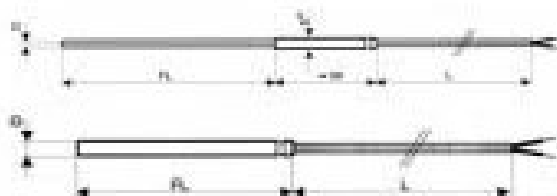


Рис. 2 – Принцип роботи термопар

Згідно фізики в залежності від температури, змінюється супротив матеріалу, зазвичай у радіоелектроніці та електриці з зміною супротиву від температури борються так як це призводить до втрат енергії під час виконання роботи пристрої, передачі електроенергії по високовольтним лініям електропередачі, у трансформаторах і тд., але такий метод знайшов своє використання для визначення температури. Таким прикладом є термометр супротиву (Рис 3). Цей датчик в залежності від температури міняє свій супротив Ом/°С, прилад показує температуру, але на високих температурах термометри супротиву неефективні, а саме дають високу похибку, але на менших температурах, більш точні ніж термопар. Окрім цього термометри супротиву діляться на декілька груп, за матеріалом (мідь, платина), а також по методу заміру супротиву (двохпроводні, трьохпроводні, чотирипроводні). Якщо за матеріалом ще більш менш зрозуміло, але під провідністю не досить добре. Двох провідні термометри супротиву це датчики по яким подається живлення і з цих же провідників міряється супротив. Але недоліком таких датчиків є те що, провідники живлення самі вносять супротив у замір, тому додають, ще провідники до самого датчику, для компенсації супротиву ось так і з'явилися трьох та чотирипроводні термометри супротиву.



### Рис 3 – Термометр супротиву двохпровідний Pt. 100

Напівпровідникові датчики температури теж прийняли той самий ефект, що датчики супротиву, але в якості матеріалів, не використовуються металеві провідники, а напівпровідники, і в залежності від матеріалу, може різко змінюватися супротив в залежності від температури, навіть складати від декількох Ом до МОМ, вони називаються терморезисторами (Рис 4). Такі датчики дають змогу використовувати такі датчики температури на платах у якості теплового захисту плати, та встановлюються, послідовного на монтажній платі, хоча їх є можливість використовувати для контролю температури у побутовій сфері, але не в промисловості, тому що такі датчики температури створюють досить сильну похибку при вимірюванні у сосудах під тиском.

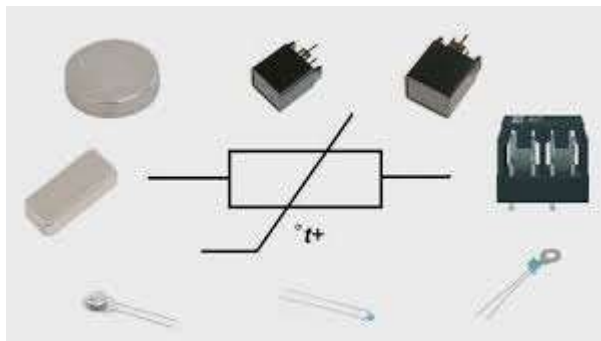


Рис 4. – Терморезистор

Одним із найсучасним, але не досить поширеними видами датчиків температури є мікросхемні (Рис 5), вони створені у корпусах транзисторів ТО-92 то у типу корпусів SOP, SOIC, для монтажу на поверхню печатної плати. Це тип датчиків температури не відрізняється від вище заявлених типів датчиків за методом заміру, але завдяки сучасним технологіям з створення мікросхем, такі датчики стали використовуватися у сучасних приладах, для заміру температури, недоліками таких типів датчиків є вузька ніша використання, а саме побут, та прилади під монтаж на плату, але деяку частину у промисловості

все ж таки такі датчики зайняли, а саме у приладах для заміру температури, де використовуються термопари [1].

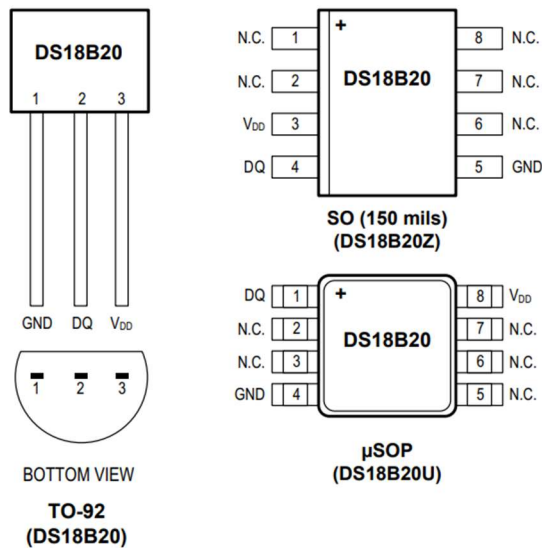


Рис. 5 – Мікросхемні датчики температури

Ці датчики потрібні для заміру температури холодного спаю, та таких прилад може компенсувати різність температури, для зменшення похибки вимірювання. Окрім цього ще одним покращенням є те що, такі мікросхеми можуть давати одразу цифровий сигнал, а вище сказані датчики дають аналоговий. Це дає змогу зменшити прилад для заміру температури, де не потрібно робити схему частини приладу для перетворення мВ/°С або розраховувати втрати електроенергії на супротив і створювати залежність Ом/°С [2].

## 1.1 ВИДИ ПРОМИСЛОВИХ ПЕРЕДАЧ ДАНИХ

У промисловості на сьогоднішній використовуються декілька методів передачі даних, вони поділяються на аналогові та дискретні (цифрові) сигнали, а також протоколи передачі даних утворених на послідовних інтерфейсах RS-232, RS-485, RS-422. До аналогових сигналів можна віднести струмову петлю.

Струмова петля це використання значення струму, а не напруги для захисту від зовнішніх наведень, від наприклад високочастотних антен, трансформаторів (наприклад трансформатор току для плавлення сталі), тощо. Таким чином зменшується похибка від зовнішніх наведень, але недоліком таких пристроїв є те що вони повинні завжди бути в роботі та потребувати енергії не тільки для живлення, а ще й для перетворення сигналів у токову петлю. Токова петля по кордонам поділяється на 3 види, а саме: 0-4 мА, 4-20 мА, а також 0-20 мА. В залежності від завдань використовують свої кордони значень перетворення, але в промисловості використовують найчастіше токову петлю 4-20 мА, це пов'язано з тим що існують прилади які можуть живлитися від струмової петлі та на неї передавати значення, наприклад рівень, витрату, рН-вимірюваного середовища, тиск, тощо.

Послідовний інтерфейс RS-232, RS-485, RS-422 створені на базі протоколу передачі даних UART, але при цьому використовується мікросхеми MAX для перетворення цифрового сигналу у послідовний інтерфейс, наприклад у RS-232 напруга передачі даних становить від -15В до +15 В, та заміряється від землі, тому на RS-232 для передачі даних використовуються 3 провідника RX, TX, GND. Окрім цього обмежень по швидкості немає, але максимальна довжина передачі даних максимум 15 м. Послідовний інтерфейс RS-485, вже має залежність швидкість/довжина тому при швидкості в 95 Кб/с довжина передачі даних може сягати до 1200м, окрім цього піднаймі від RS-232 передача даних йде по двом провідникам А та В, по напрузі +5В та -5В, та крім цього приймаючий сигнал пристрій заміряє напругу між цими провідниками, тому при зміщенні напругу на 200 мВ та більше такі сигнали нерозпізнаються контролером та відсікаються, що окрім

екранування та інших способів захисту від зовнішнього впливу поміж підвущує передачу даних, крім цього на кінці провідників встановлюють так звані «термінатори» це резистори на 120-150 Ом, що діють, як маркер того що це кінець передачі даних.

Окрім цього послідовний інтерфейс RS-485 обзавівся й своїми протоколами, найпоширеними є MODBUS, PROFIBUS, FIELDBUS, слід зазначити що протокол MODBUS використовується і на послідовних інтерфейсах RS-232 та RS-422. Протоколи PROFIBUS та FIELDBUS є розповсюдженими, але не відкритими та для їх використання у розробці пристроїв потрібна ліцензія віднаймні від MODBUS. В усіх цих протоках створюється передача даних Master-Slave, де Master це основний прилад, якому підпорядковуються інші прилади так звані Slave, основна задача Master проводити опитування приладів підпорядкованих йому, отримувати від них інформацію, обробляти інформацію та віддавати команди своїм підпорядкованим пристроям. Для того щоб зв'язок Master-Slave створювався усім пристроям у цій системі дається свій унікальний адрес, зазвичай відрахування починається з 1, тому що 0 це адреса Master, але в залежності від створення сеті передачі даних адресація може бути різною.

В протоколі PROFIBUS є зв'язок Slave-Slave ця передача даних використовується для обміну даних між рівними за собою приладами у підключеної сеті не тільки для обробки даних, тобто є 2 CPU-300 (ПЛК, Виробництво SIEMENS), що рівні між собою, але їм обоім підпорядковані свої прилади, але мають свого Master наприклад CPU-400 або CPU-1200, але щоб не передавати дані на свого Master, щоб він вже віддав команду інший CPU-300. Тому для прискорення роботи CPU-300 передає сигнал на такий самий CPU-300, якому підпорядкован вже виконавчий пристрій.

Також існує зв'язок Master-Master, тобто у сеті можливий 1 Master, але 2 Master, може бути ПК, де вже людина проводить конфігурацію сеті, перегляд інформації, тощо.

Якщо брати сімейство ПЛК SIEMENS, то в них Master може бути CPU-300, CPU-400, CPU-1200, окрім цього ще й HMI (Human-Machine Interface) [3].

## 1.2 ВИСНОВОК ВИБОРУ ДАТЧИКУ ТЕМПЕРАТУРИ ТА МЕТОДУ ПЕРЕДАЧІ ДАНИХ

З усіх вищезазначених датчиків температури можна виділити лише 2 основних метода вимірювання температури, а саме зміна супротиву датчика у вимірюваному середовищі та генерація ЕДС під впливом вимірювального середовища. Окрім цього якщо почати конструювати схему обробки сигналу з датчику. Вслід за цим виникає що в залежності від типу датчику, потрібно посилювати ЕДС, для заміру вимірювального середовища, як показано на рис 6, ду у якості посилювача ЕДС для термопарі використовується операційний підсилювач:

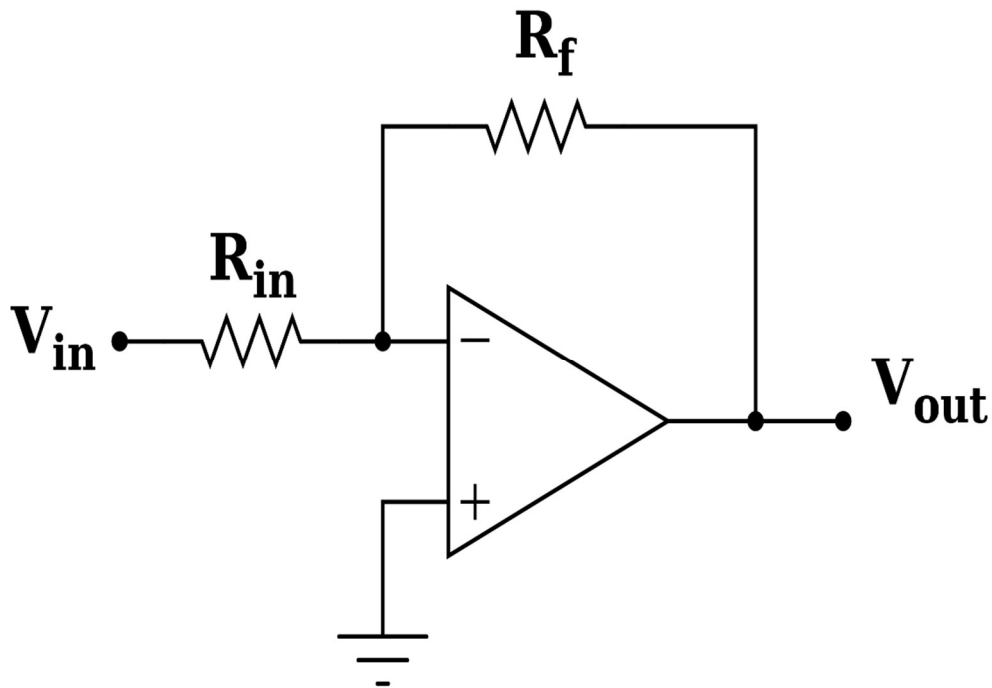


Рис 6 – використання Операційно посилювача у режимі інвертованого підсилення.

Схемі на операційному посилювачі представлена на рис 5 є лише універсальним прикладом того що посилення ЕДС для термопарі є необхідним, що підвищує площу плати майбутнього приладу, навіть при використанні SMD-компонентів, підвищує вартість приладу, окрім за все для підвищення точності приладу потрібно створювати допоміжний вузол



замірювання температури на «холодному» спаї, тобто на кінцях терморпарі що будуть підключатися до клем приладу, що в свою чергу підвищує час на розробку, вартість, ємність плати готового приладу.

Наступною універсальною схемою яка є поширена у промисловості це замір супротиву. З одного це набагато легше для реалізації від заміру та посилення ЕДС, але є свої нюанси, а саме потрібно слідкувати за стабільністю напруги, що приходить від такого датчика, потрібно враховувати не тільки супротив датчика, а ще й провідників, до яких під'єднан датчик супротиву, якщо подивитись на універсальну схему підключення, рис 7, то досить легко зрозуміти, що обчислення лежать у межах Закону Ому, де знаючи струм та напругу, що живлять датчик супротиву, можна обчислити супротив та порівняти з таблицею залежності температури від супротиву, що надає виробник для свого датчику у документації.

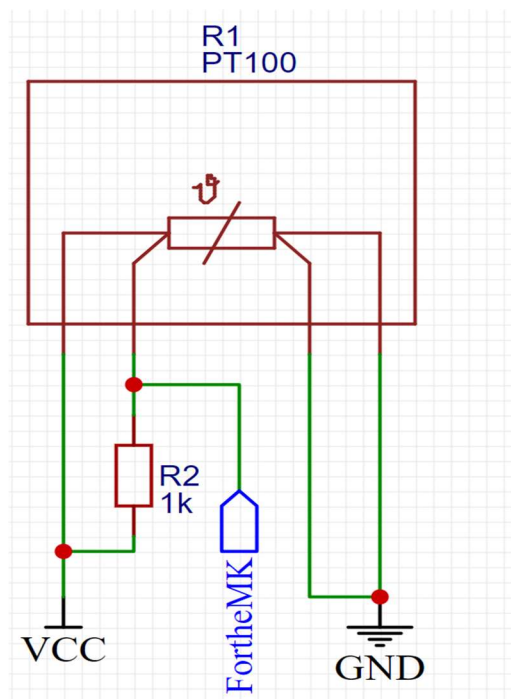


Рис 7 – універсальна схема підключення термометру супротиву 4-х провідного Pt100 до мікроконтролеру

Хоча на рис 6 представлена схема для 4-провідного датчику температури, це не змінює її використання для 2- або 3-провідного датчику супротиву. Окрім

цього ця схема хоч і виглядає досить легкою, від використання схеми термопар, але обоє цих підключень є досить розмитими так як у промисловості ці датчики температури можуть розташовуватися під високочастотними генераторами, як трансформатор, окрім цього може бути неякісний монтаж, тощо. Тому для цих типів датчику насамперед необхідна допоміжна схема для фільтрації сигналу, використання компенсаційних кабелів для компенсації втрати ЕДС на відстані від термопар до вимірювального приладу, це теж стосується та для датчику супротиву, але компенсаційний кабель компенсує різницю втрати супротиву.

Якщо почати порівнювати вище зазначене, як недоліки до датчиків температури ЕДС та супротиву і порівняти з датчиком температури розробленому у вигляді мікросхеми рис 8:

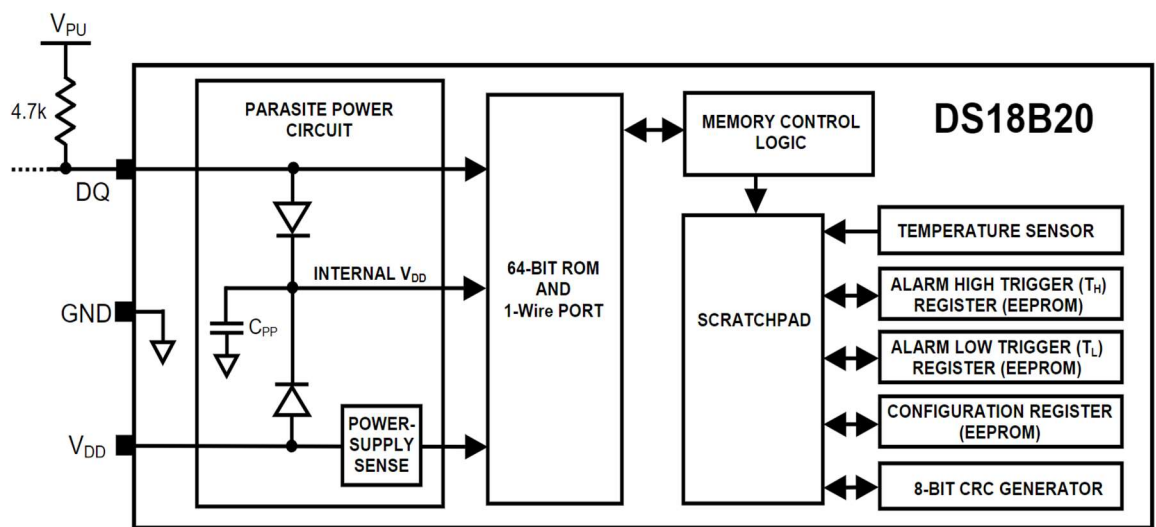


Рис 8 – блок схема мікросхеми датчику температури

Та порівняти рис 8, з вище зазначеними схемами можна зрозуміти вже недоліки датчиків термопар та датчиків супротиву, а саме у тому що мікросхема має вже встановлений датчик температури, а також схему для перетворення/фільтрації сигналу/посилення сигналу при цьому бути в корпусі ТО-92, або взагалі у вигляді SMD-компоненту. Це не тільки спрощує розробку, тому ще вже не потрібно думати над посиленням, або компенсувати втрату

сигналу, де розбіжність навіть у декілька мВ може зробити похибку вимірювання в залежності від завдання, від 0.1 до 10 °С, а може навіть більше.

Окрім цього у датчиків температури на базі мікросхем є ще свої недоліки, так як вони передають вже перетворений сигнал у цифру, то це теж має свої втрати, але втрати вже бітів, що теж можуть давати похибку, але озируючись на документацію від виробника можна полегшити собі завдання в залежності від кінцевої цілі, тобто якщо діапазон температур та дистанція між розміщенням датчика та приладу співпадає з заявленими від виробника то слід вже подумати лише над захистом від зовнішніх факторів які можуть впливати на передачу сигналу, наприклад використання екранованого кабелю, що теж може спростити розробку, а також інші методи фільтрації сигналу, наприклад «Репітер» - прилад, що ретранслює сигнал та фільтрує, але це вже інший пристрій тому на нього не орієнтуємося. Та не слід забувати ще й про енергоефективність використання датчиків температури на базі мікросхем, тому що живлення в них може бути від самого приладу та становити до 10 В, а опираючись на нелінійність вимірювання у часі то можна навіть відключати датчик температури від приладу самим приладом або переводити прилад стану до моменту опитування датчику, що підвищує енергоефективність приладу.

Вибір інтерфейсу даних досить скудний у тому що більшість інтерфейсів передачі даних є комерційної таємницею, тому просто так доступу до документації є неможливим, це стосується протоколу PROFIBUS та FIELDBUS. Тому вибран був протокол передачі даних MODBUS.

### 1.3 ПРОМИСЛОВІ ТИПИ ДАТЧИКІВ ТЕМПЕРАТУРИ

На сьогоднішній день рішення вимірювання температури, а саме виробництво датчиків є досить поширеними, тому кількість компаній, що надають рішення досить багато, але представлена тут лише маленька частка того що надає ринок.

Виробники:

SIEMENS

DANFOSS

ТЭРА

ENDRESS+HAUSER

SIDERMES

Це лише частка представлених датчиків температури, які мені довелося працювати, окрім цього компанію надають різні типи кріплення датчиків температури, такі як:

Погружні

Зовнішні

Датчики з гнучким кабелем

Перетворювачі (перетворюють дані температури у токову петлю або в RS-485)

Слід виділити ще той факт, що усі вище надані компанії випускають не тільки датчики темперари, а ще й датчики тиску, рівня, витратоміри, рН, кондуктометри, перетворювачі частоти і тд., що ставить досить важким гравцем на ринку.

За зовнішнім виглядом, промислові датчики виглядають так:

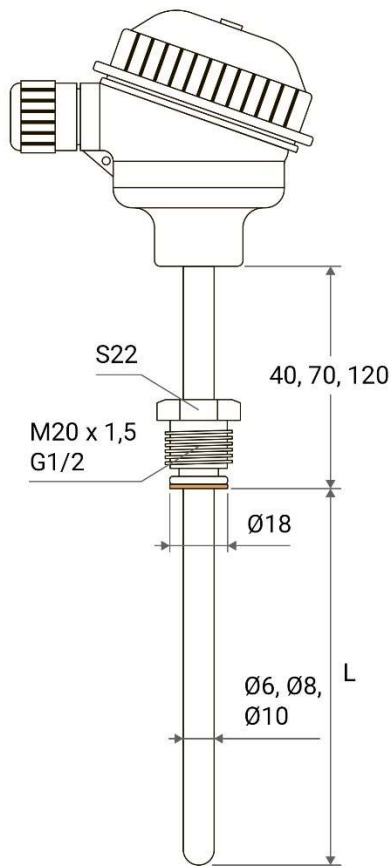


Рис 9 – датчик температури виробництва ТЭРА

Також є досить інтересні рішення від компанії SIDERMES, а саме погрузні термопари для заміру температури у рідкій сталі. Термопара поділяється на 2 частини, перша це зонд у якому є роз'єм, а друга частина це целюлозний кожух один кінець якого це гарячий спай термопари, а другий під'єднується при одяганні на зонд до роз'єму. Потім зонд заміру через компенсаційний кабель підключається до приладу, який вже показує заміряну перетворює сигнал та показує температуру, а також отримані дані передає по токовій петлі. Під час заміру захисний кожух згорає з гарячим спаєм, але сам зонд залишається не пошкодженим, що дозволяє у тривалому темпі заміряти температуру (більше 50 замірів) таким зондом до технічного обслуговування. Окрім цього компанія SIDERMES має рішення для заміру не тільки

температури, а ще вуглецю та окислюванності у рідкої сталі, при цьому одним приладом та спеціальним захисним кожухом зонду.

Окрім цього є ще й безконтактні прилади для заміру температури, це пірометри та тепловізори. Такі прилади носять у собі ефект інфрачервоного випромінювання. Тобто коли нагрівається тіло то воно випромінює енергію у інфрачервоному спектрі, а такого приладу встановлена лінза яка фокусує на випромінювання на термопарі розташованому у приладі, яке в залежності від нагріву створює ЕДС, а потім прилад перетворює отримані дані у температуру для користувача.

## 2. РОЗРОБКА СХЕМИ ПРИЛАДУ

### 2.1 ПІДБІР КОМПОНЕНТІВ ДЛЯ ПРИЛАДУ

У якості мікросхеми для заміру температури був використана мікросхема DS18D20 (рис. 10). DS18D20 це цифровий 9/12-бітний датчик температури, що не тільки заміряє температуру у навколишньому, а ще передає дані температури у вигляді 9/12-бітного цифрового сигналу, це зменшує елементарну базу компонентів для фільтрації, посилення аналогового сигналу, аналогове перетворення сигналу (АЦП).

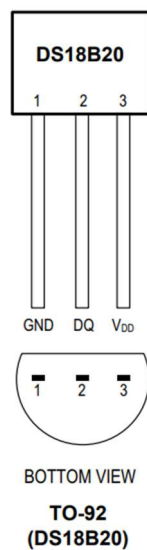


Рис. 10 – цифровий датчик температури DS18D20

GND – земля; DQ – цифровий вихід даних;  $V_{DD}$  – +3-5 В живлення;

Цей цифровий датчик має живлення +3-5 В, окрім має можливість живлення у так званому «Паразитному» режимі, цей режим дозволяє живлення датчику температури від цифрового виходу мікроконтролера, що може підвищити енергоефективність майбутнього приладу, а ще й з економити на трасуванні та проектуванні живлення для датчику. А корпус самого датчику виготовлюється у корпуса TO-92, SO, а також у водозахистному корпусі.

Окрім цього DS18D20, має можливість використовувати шину даних 1-Wire, ця шина передачі даних дозволяє підключати на 1 пін мікроконтролера

більше 50 датчиків. Дані щодо можливостей датчику DS18D20, наведені до Таблиці 1.

Таблиця 1 – характеристики DS18D20

Найменування параметру	Значення	Похибка	Одиниці вимірювання
Температура	-10...+85	+/-0.5	°C
	-55...+125	+/-2	°C
Напруга живлення	3.3...5.0		В
	2.2...5.0		В
Живлення струмом	750...1000		nA
	1...1.5		mA
	5		µA
Час опитування	95-750		мС

Таблиця 1 створена на базі документації виробника, а саме Dallas Semiconductor, окрім цього з документації, були взяті принципові схеми підключення DS18D20 до мікроконтролеру, а також при підключенні до шини 1-Wire (Рис. 11), а також для підключення в «паразитному» живленні (Рис. 7).

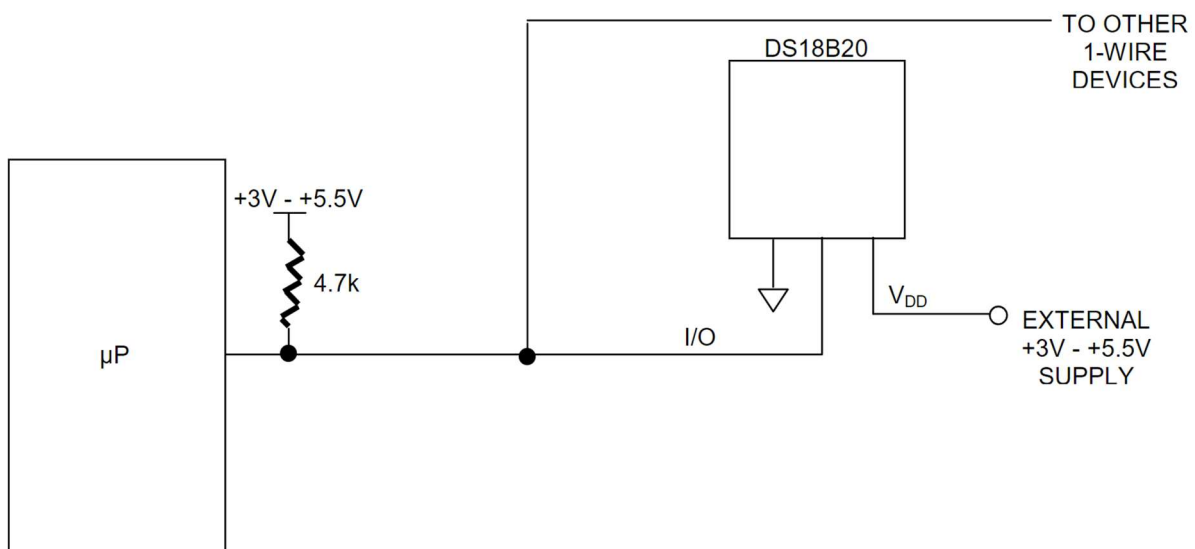


Рис 11 – Підключення DS18D20 до піну мікроконтролеру



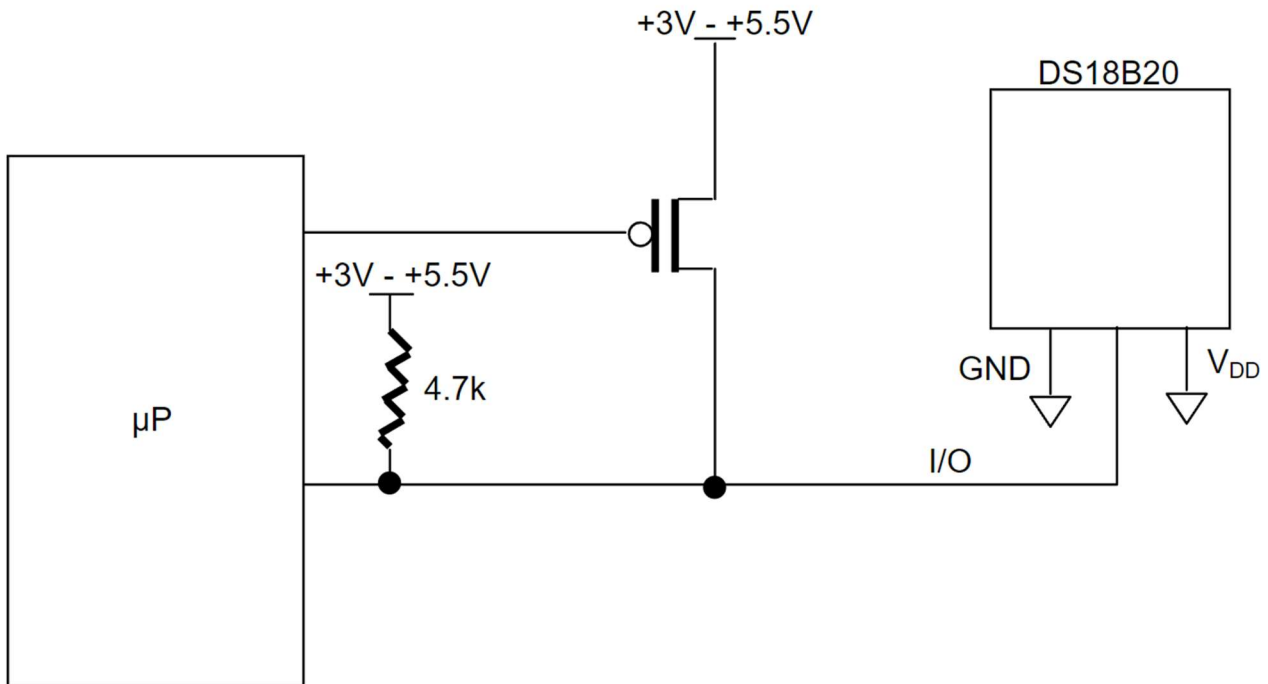


Рис 12 – підключення DS18D20 у «паразитному» режимі

Слід зауважити, що можливість використання DS18D20, як по шині даних 1-Wire, а також у «паразитному» підключенні живлення не використовувалось, окрім цього є досить привабливим для подальшого розвитку використання DS18D20 для промисловості, як наприклад харчова або направлення, де температура робочої зони досягатиме температури рекомендованої розробником.

Якщо взяти наприклад хімічну промисловість, то такі датчики своє призначення знайшли б у ректифікаційних колонах при температурах роботи від +70...+120 °С, окрім цього використання шини 1-Wire, допомогло б з економити місце для підключення іншого обладнання під автоматизацію ректифікаційної колони (датчики тиску, датчики рівня, витратоміри, тощо).

## 2.2 ВИБІР МІКРОКОНТРОЛЕРУ ТА ПЛАТФОРМИ

Зважаючи на те що у нашому житті на сьогоднішній момент тема з використанням мікроконтролерів стала поширеною, завдяки платформі ARDUINO, використання плат RASBERRY PI, ORANGE PI та дешевості мікроконтролерів сімейства PIC, ATMEGA, ATTINY та мікропроцесорів промислових STM, все більше заохочуваних займатися розробкою своїх приладів, маючи при цьому майже ніяких знань, створюючи свої розумні годинники, системи гідропоніки для автоматизації ферм, системи розумного будинку де йде моніторинг за мікрокліматом, годування улюбленців у автоматичному режимі, тощо. Дає все це широкий асортимент для розробки приладу насамперед на простоту розробки програмного забезпечення, для спрощення створення прототипу який швидко запрацює на колінці, а також дасть змогу використання отриманих результатів для покращення майбутнього приладу.

Насамперед критеріями вибору платформи та мікроконтролеру є:

1. Простота ПО для створення програми.
2. Гнучкість створення коду без зайвих «костилів».
3. Отриманий код можна перенести з легкістю з платформи до мікроконтролеру.
4. Дешева вартість платформи та мікроконтролеру.

Ці критерії за якими проводився вибір платформи та мікроконтролерів. Плати RASBERRY PI, ORANGE PI відпали одразу за двома причинами, насамперед ці плати хоч і гнучкі, але ціна 2500 грн за плату досить велика, що ставить 1 проблему, по друге плата досить велика для приладу, окрім цього не слід забувати, що обробка даних цих плат досить велика та витратити на прилад для заміру температури є лише втратою часу, ресурсів, що не є розумним.

Використання платформи ARDUINO, має свої переваги, а саме вартість плати до 300 грн, що ставить вже на перше місце за для створення своїх приладів, окрім цього сама платформа безкоштовна, а при цьому матеріалу від розробників

досить багато, як і від сторонніх розробників (платформа дозволяє можливість створювати копії плат, платформи та комерціалізувати завдяки ліцензії від розробника), а також від інших людей, що займаються розробкою ПО, інтуїтивний інтерфейс та одразу можливість розбирати приклади прямо у програмі, досить привабливі. Це ставить платформу ARDUINO на 1 місце для вибору під розробку приладу. Але платформа має свої недоліки, а саме проблеми з енергоефективністю, але для створення пробної версії, та при цьому різний форм фактор плат прямо приваблює, а також недоліком платформи є використання мікроконтролерів сімейства Atmel, що призводить до штучного зменшення використання пам'яті мікроконтролера, а ще взяти за увагу можливість використання резервування пам'яті під назвою BOOTLOADER – це резервація пам'яті під код, який буде незмінений під час перезапису мікроконтролеру (якщо потреби у цьому не буде), але при цьому пам'ять що доступна, ще зменшується для використання, але для використання під вимірювання температури цим можна знехтувати. Також слід використання платформи призводить до проблем з зменшенням швидкості роботи мікроконтролеру, але вимірювання температури є лінійним процесом та при цьому нешвидким, також цим недоліком є можливість знехтувати. Також платформа ARDUINO є досить гнучкою у плані коду, були випадки коли платформу ставили на мікропроцесори сімейства ESP (навіть не розглядалися по причині зайвих можливостей у вигляді WI-FI, BLUETOOTH так як такі можливості у промисловості не потрібні), а використання мікроконтролерів сімейства ATMEGA робить легкість переносу коду на чистий камінь без проблем.

Вибір мікроконтролерів для пристрою теж підпорядковувався вище зазначеним критеріям, але насамперед вибір мікроконтролеру був вже відомий лише завдяки вибору платформи, але все ж потрібно порівняти, чому вище зазначені мікроконтролери були не вибрані у якості приладу.

Почнемо з сімейства мікроконтролерів PIC, ці мікроконтролери були одразу не вибрані з причини відсутності потрібного програматору, а створення на базі платформи ARDUINO, затягнула би розробку приладу. Хоча PIC може

похизуватися й видами серій мікроконтролерів вже з захистом від копіювання, тобто те що було «зашито» у мікроконтролер вже не буде прочитано, а також не буде скопійовано. Також перевагою мікропроцесору можна назвати стійкість, так як мікропроцесор може витримувати досить великі стрибки напруги, великий діапазон температур без втрат на свою роботу.

Наступними є сімейство ATTINY це молодша версія сімейства ATMEGA, тому все що є у ATMEGA є у ATTINY, але менше, наприклад ATTINY13 має 4 Кб пам'яті, а ATMEGA328 має 32 Кб, при цьому і споживає менше, якщо брати умови відбору та завдання це сімейство досить добре підходить, але є головний недолік так як сімейство ATTINY це молодше сімейство від ATMEGA тому знайти потрібний мікроконтролер з потрібними функціями під завдання досить важко. Також є недоліки використання платформи ARDUINO, як вище вказувалось пам'ять, але ще й недоліком є те що розробку переносу платформи ARDUINO на сімейство ATTINY займаються ентузіасти, тому досить багато «костилів» від самих розробників, що ведуть перенос платформи, а це накладає допоміжні кордони на готовий прилад та його можливості, що не є досить добре.

Використання мікропроцесорів STM для майбутніх проектів не тільки для цієї роботи є досить перспективним завдяки досить великим перевагам, а саме вартість плати з таким мікропроцесор сягає приблизно 500 грн (компанія ST, власники виробництва STM випускають свою платформу та плати для створення/наладки приладів), також потужністю так як вже використовується технології близькі для розробки процесорів тому й потужність мікропроцесору сягають до 500 МГц (в залежності від версії ядра). Також під мікропроцесор є своя платформа під назвою CUBEMX, що дозволяє не так легко як платформа ARDUINO створювати та налагоджувати прилади, але все ж таки дозволяє від наймні від ARDUINO платформа дозволяє налагоджувати ті процеси, які не доступні в платформі ARDUINO, а саме швидкість мікропроцесору де можна настроїти як потрібно, реалізувати режим сну. Це ставить досить доброї перевагою перед платформою ARDUINO, але є досить проблемними недоліками, а саме гнучкість, так як створений код на платформі CUBEMX досить

проблемно переносити з одного мікропроцесору на іншій (наприклад з STM32F103C8T6 на STM32F103UCC8), тому що код потрібно копіювати, створювати новий проект під новий мікропроцесор та конфігурувати заново мікропроцесор. Також недоліком платформи є момент де при написанні коду, а саме коли створюється проект, при цьому сама платформа пише код під мікропроцесор, тому є спеціальні маркери де можна писати свій код, якщо не там то при компілюванні проекту платформа їх видалить, що не зручно, але можна звикнути.

Якщо брати заувагу використання мови програми то слід зауважити що у всіх вищезазначених платформах є використання C подібної, що ставить гнучкість «прошивання» мікроконтролеру/мікропроцесору/процесору на одну позицію, хоча можливі казуси.

Виходячи з всього вище описаного перевагу у розробці тестового приладу та приладу взагалом було віддано платформі ARDUINO [4,5,6].

## 2.2.1 СТВОРЕННЯ/РОЗРОБКА ПРОБНОЇ СХЕМИ

В якості початку роботи, було вирішено використання платформи ARDUINO, на це було декілька причин:

1. Цифрові датчики DS18D20 вже використовуються у хобійних проектах.
2. Платформа ARDUINO, досить гнучка для роботи з мікроконтролерами не на професійному рівні, але дає змогу завдяки модулям, а також відкритому коду та поширюваності, досить швидко створювати свої проекти та навчатись.
3. Також платформа дає змогу швидко створити пристрій та під час створення наладити цей пристрій, окрім цього така швидкість дає змогу створювати перші зразки приладів для інвесторів, що дає змогу показати реалізацію проекту для залучення інвестицій.

Для реалізації, було вибрано плату ARDUINO NANO, тактова кнопка, а також сам датчик температури DS18D20 та 2 резистори на 4.7 Ком та 10 КОм, над вибором живлення для схеми, не має потреби, тому що плата ARDUINO NANO при підключенні до ПК, живлється від порту USB, що становить від 4.5...5 В, в залежності від характеристик блоку живлення ПК. Таким чином принципова схема пристрою зображена на рис.13.

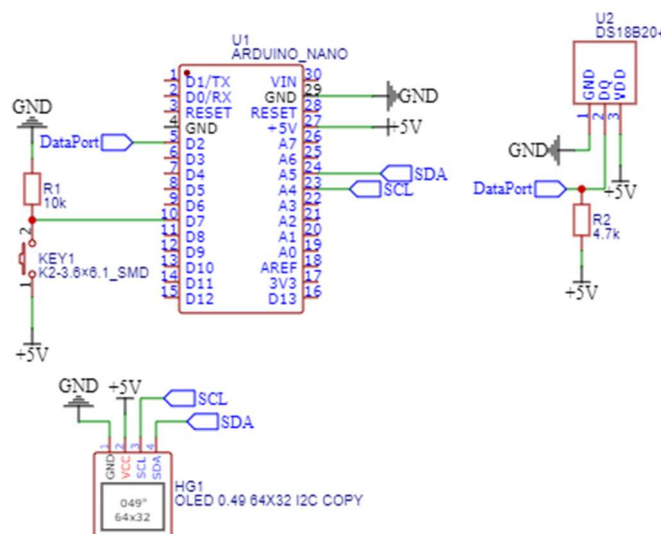


Рис. 13 – Підключення DS18D20 до плати ARDUINO NANO

З рис.13 маємо такий список компонентів та їх маркування:

1. U1 – плата ARDUINO NANO
2. U2 – цифровий датчик DS18D20
3. KEY1 – Тактова кнопка
4. HG1 – OLED дисплей 64x32, що працює по I2C інтерфейсу
5. R1 – резистор на 10 КОм для стягування тактової кнопки з мінусом
6. R2 – Резистор на 4.7 КОм для стягування цифрового піну DS18D20 до плюсу живлення.

Так як плата ARDUINO NANO (далі U1) підключена под ПК, то її піни GND та +5 використовуються для створення стабілізованного живлення 5 В, до цих пінів під'єднуються пін VDD від елементу U2, VCC від елементу HG1, а також провідник від елементу KEY1, DQ від елементу U2, під'єднується до елементу U1 на пін D2 (5), де до пінів DQ та VDD під'єднується елемент R2. Елемент KEY1 під'єднується до U1 на пін D6 (10) та на цей же пін під'єднується елемент R1, який іншим кінцем під'єднується до GND від U1. HG1 під'єднується до U1 своїми піна SCI до A4, а SDA до A5, свободні піни GND від U2 та HG1 під'єднуються до піну GND від U1, утворюючи мінус живлення.

## 2.2.2 КОД ТЕСТОВОЇ ПРОГРАМИ

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <avr/sleep.h>

/*Підключаємо потрібні нам бібліотеки для роботи з датчиком температури
DS18B20,
а також з екраном та режимом сну*/
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define SCREEN_ADDRESS 0x3C
#define OLED_RESET -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16

/*Вказуємо адресу екрану, та його властивості, як ширина та висота*/
#define ONE_WIRE_BUS 5 //Змінна, якою ми підключаємо датчик DS18B20
#define TEMPERATURE_PRECISION 12 //Змінна, якою ми задаємо точність
датчику

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensor(&oneWire);
int Button_Temperature = 10; //змінна, якою ми підключаємо кнопку для роботи з
датчиком
```



```

int Button_State = 0; //Змінна, якою ми приравнюємо початковий стан кнопки

void setup() {
  pinMode(Button_Temperature, INPUT);
  sensor.begin();
  display.begin();

  /* Блок програми в якому присвоєно включення потрібних бібліотек та пінів
  МК,
  окрім цього цей блок програми МК робить лише один раз за весь час роботи*/
}

void loop() {

  /* Основний блок програми, це бескінечний цикл, в якому є умова
  для перевірки стану кнопки, окрім цього, в залежності від стану кнопки,
  працюють потрібні алгоритми програми*/

  if (Button_State == 0) {
    Sleep(); //Алгоритм сну
  } else {
    Temperature0(); //Алгоритм роботи датчику DS18b20
    delay(60000); //Стан роботи, приблизно 1 хвилина
  }
}

void Temperature0() { //блок програми, що відповідає за замір температури, а
також вивід даних на дисплей
  sensor.requestTemperatures(); //Під'єднання до датчику DS18b20
  float temperatureCelsius = 0; //Початкова температура змінної

```

```

temperatureCelsius = sensor.getTempCByIndex(0); //переназначення датчику до
змінної
display.clearDisplay(); //спочатку очищаємо дисплей
display.setTextSize(2); //Встановлюємо висоту тексту
display.setCursor(0, 0); //початкову позицію тексту
display.print("Temperature: "); //Вивід тексту
display.print(temperatureCelsius); //Вивід змінної де збережені дані температури
display.println("°C"); //Вивід тексту
display.display(); //Закінчення роботи з дисплеєм
}
void Sleep() { //Блок програми що працює зі сном
sleep_cpu (); //Створення рижиму сна
}

```

Вище створен код програми, як працюють компоненти програми написано у коментарях, що починаються з символів // та \*\. Алгоритм коду в данному випадку простий, а саме це опитування датчику температури, отримання даних з нього, та вивід отриманого значення на дисплей, окрім цього передбачено режим сну, але не до кінця реалізовано, це вже пов'язано з недоліками самої платформи.

### 2.2.3 ПРОБЛЕМИ ПЕРЕХОДУ З ПЛАТ ARDUINO

Використання ARDUINO NANO хоч і досить дешево, тому що над блоком живлення потрібно навіть і думати, це вже є на платі, хоч і в якості живлення виступає ПК, але для приладу, який має бути універсальний, це не є досить добре, тому використання плат ARDUINO відміняється, а в якості мікроконтролеру буде використовуватись ATMEGA328P/ATMEGA328PB, ці мікроконтролери виробництва MICROCHIP, раніше вони були виробництва ATMEL, але компанія MICROCHIP викупила та поглинула компанію ATMEL. Чому саме ATMEGA328P, а не інший мікроконтролер? Відповідь полягає у тому, що виробники платформи ARDUINO використовують ці ж самі мікроконтролери, тому над переробкою коду з мови ARDUINO C на C++ або на чистий C не потрібно, потрібен лише програматор (можна використати ту ж саму ARDUINO NANO у якості програматора), та «зашити» програму у мікроконтролер. З одного боку це виглядає досить просто, хоча є свої підводні камні.

Основна проблема вже з'являється у створенні блоку живлення та створення стабілізованою напруги живлення, при чому таке, від якого мікроконтролер буде житись, стабілізоване, а на сам перед захищене від короткого замикання, або від інших проблем з живленням. Окрім цього взяти за увагу, що живлення буде йти від промислового живлення, а це 24В, хоча всі БП на підприємствах створені спеціально під 24В на 15А та вище, при цьому захищеними від короткого замикання та інших форсмажорних обставин, але перестраховатись потрібно. Тому для стабілізації напруги з найменшими потребами по грошам це використання мікросхем LM317, TL431, LM338, та інших мікросхем, але вибір впав на TL431. Чому саме TL431? Вибір простий, якщо відкрити документацію на цю мікросхему то можна подивитись що ця мікросхема створена для стабілізації напруги, що вже вирішує проблему з

перетворенням 24В на 5В, та його стабілізації, хоча не вирішує проблему з коротким замиканням, неполярним підключенням приладу живлення.

Наступною же проблемою є саме розробка печатної плати, до цього при використанні плати ARDUINO можна було використовувати без пайкову макетну плату, але не для готового пристрою, де потрібно зробити трасування таким чином, щоб траси провідників витримували подану напругу.

Наступною проблемою це є передача даних, хоча від початку такої цілі проекту не було, так як для цього був передбачен OLED-дисплей, але все ж таки був обрана ще й можливість передачі даних по протоколу MODBUS, на послідовному інтерфейсі RS-485, що призводить до використання порту RJ-45 та використання мікросхеми MAX485 зі своєї обв'язкою.

### 2.3.1 ПІДБІР ЕЛЕМЕНТІВ ЖИВЛЕННЯ ДЛЯ ПРИЛАДУ

Так як вище зазначено було використання TL431 то виберемо вже її для використання, мікросхема є у декількох версіях, а саме TO-92, SO-89, SOT-23, SOP и SOIC. Для приладу був використан корпус TO-92, що зображен нижче, рис 14:

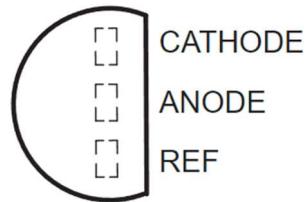


Рис 14 - TL431 у корпусі TO-92

Також, якщо подивись на рис 13, а також у документацію, то розпіновка мікросхеми виглядає так:

CATHODE – Катод; ANODE - Анод; REF – Опорний пін;

Сама же мікросхема працює до 36В, що для проекту тільки плюс при максимальному використанні в 24В, дає ще й запас по перенапрузі, а також по напрузі навантаження йде у 100 мА, що для пристрою саме те, навіть з використанням OLED-дисплею. Принципова схема була взята з документації, та показана на рис 15.

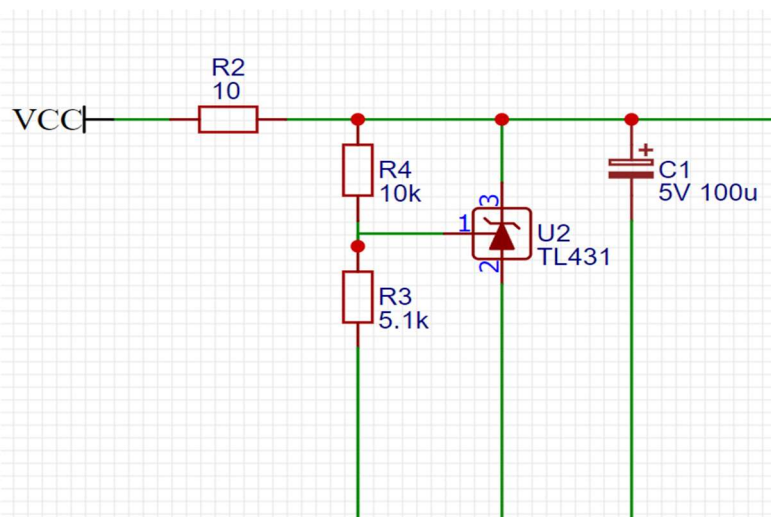


Рис 15 – Принципова схема підключення мікросхеми TL431

У цій обв'язці для TL431 використані резистори R4, R3 для створення поділювання напруги для створення стабілізації 5В, а резистор R2 був використан як запобіжник, за потреби можна залюбки замінити на запобіжник на 0.2 А. Резистори R4, R3 були вируховані по формулі з документації з мікросхеми на рис 16:

$$V_O = \left(1 + \frac{R1}{R2}\right)V_{ref}$$

Рис 16 – Формула розрахунку стабілізації напруги TL431

## 2.3.2 ВИБІР МІКРОСХЕМИ ДЛЯ РОЗРОБКИ ПОСЛІДОВНОГО ІНТЕРФЕЙСУ RS-485

Використання інтерфейсу RS-485 необхідно для передачі даних між пристроями. Чому саме RS-485 використовується у проекті? Тому що цей інтерфейс досить поширений, навіть якщо буде використовуватися у в ланцюгу з іншими приладами, це надасть можливість використовувати у якості Мастера інший прилад по цьому інтерфейсу. Також інтерфейс використовує 2 провідника для з'єднання до мережі пристроїв, а обмеженням інтерфесу є довжина до 1200м. У якості мікросхеми перетворювача UART - RS-485, виступає MAX485, це досить поширена мікросхема перетворювача. Підключення до мікроконтролеру показано нижче на рисунку 17.

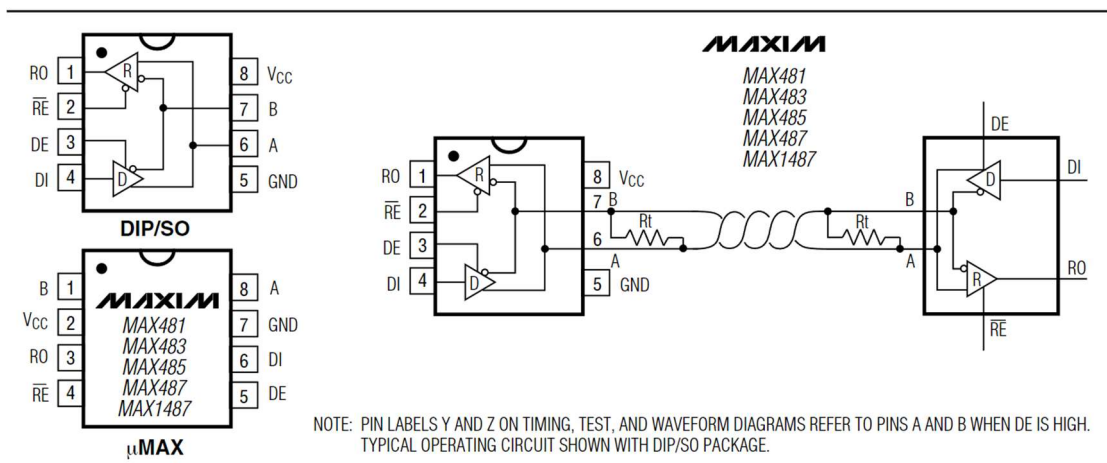


Рис 17 – схема розклучення MAX485 до мікроконтролеру.

1 та 2 пін мікросхеми під'єднується до цифрового піну мікроконтролеру, для реагування на послідовність бітів, а саме на біти, що відповідають за адресу пристрою. 3 та 4 піни під'єднуються до пінів мікроконтролеру, які відповідають за інтерфейс UART, 5 та 8 це живлення мікросхеми, а 7 та 6 це піни що відповідають за утворення каналу даних А та В. Пояснення з даташиту стосовно резисторів, це кінцеві резистори (термінатори), ставляться на останньому

підключеному пристрої до інтерфейсу та цим резистором помічається, що надалі передачі даних не буде по інтерфейсу (це потрібно для конфігурування мережі, а також для підвищення стабільності сигналу). Під'єднання до мікроконтролера використовувалось, що використовується у цьому проекті видно на рисунку 18.

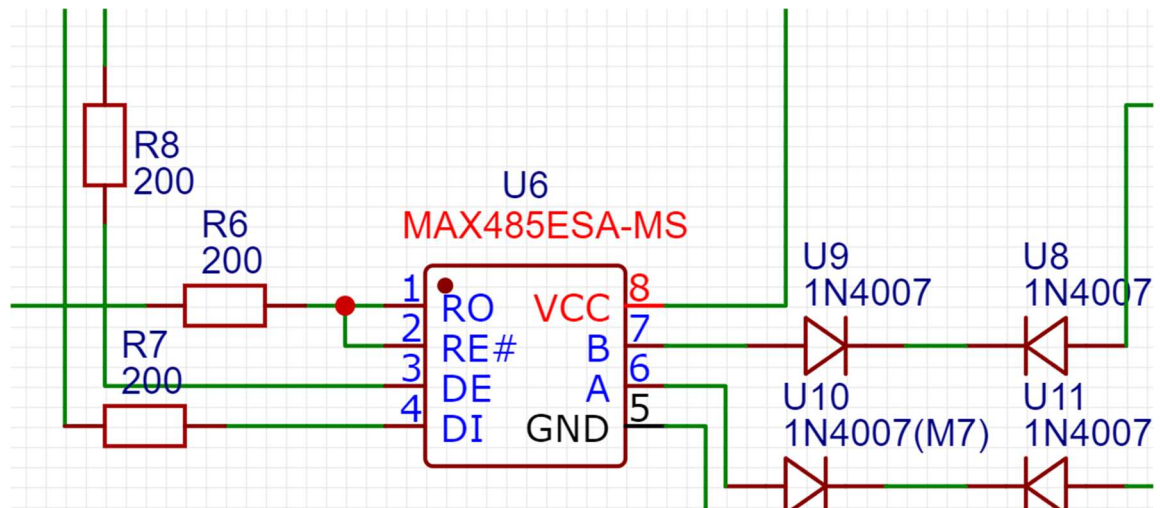


Рис 18 – використання мікросхеми МАХ485 у проекті

З рисунку 16 мікросхема МАХ485 має допоміжну розв'язку у вигляді резисторів R6-8, ці резистори потрібні для фільтрації сигналу від пінів мікроконтролера, а діоди U9-11 потрібні для захисту при неправильному з'єднанні [7, 8].



2.3.3 ПРИНЦИПОВА СХЕМА ПРИБОРУ ТА ЇЇ ОПИС

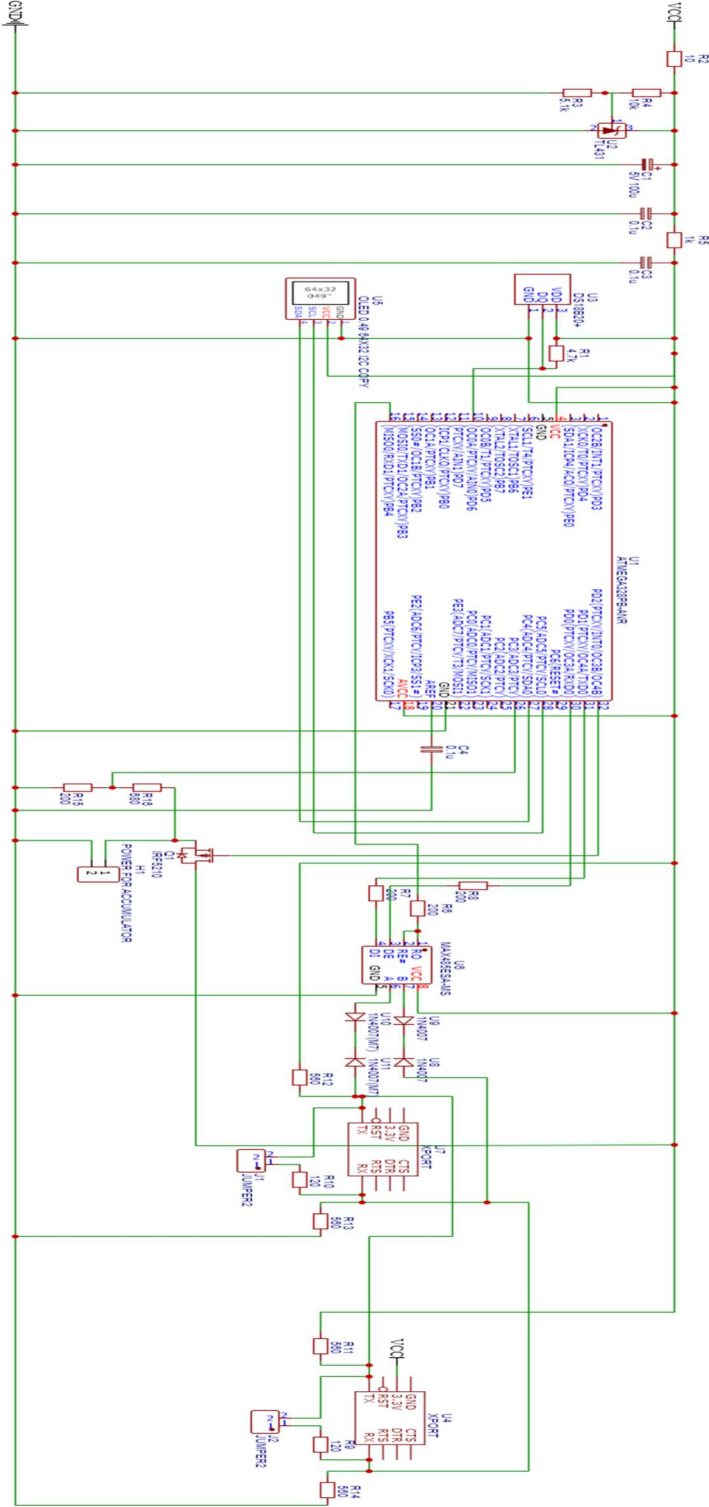


Рис 19 – принципова схема приладу.

Компоненти приладу, що зображені на рис 19:

R1 – резистор на 4.7 КОм

R2 – Резистор на 10 Ом

R3 – Резистор на 5.1 КОм

R4 – Резистор на 10 КОм

R5 – Резистор на 1 КОм

R6-R8 – Резистори на 200 Ом

R9-R10 – Резистори на 120 Ом

R11-R14 – Резистори на 560 Ом

R15 – Резистор на 200 Ом

R16 – Резистор на 680 Ом

C1 – Електролітичний конденсатор на 5 В та 100 мкФ

C2-C4 – Керамічний конденсатор на 0.1 мкФ

Q1 – MOSFET-транзистор IRF5210

U1 – Мікроконтролер ATMEGA328PB

U2 – Мікросхема стабілізації напруги TL431

U3 – Датчик температури DS18B20

U4 – Роз'єм RJ-45 для кабелю ETHERNET

U5 – OLED-дисплей 0.94 дюйма

U6 – Мікросхема MAX485

U7 – Роз'єм RJ-45 для кабелю ETHERNET

H1 – Роз'єм підключення для акумулятору

J1-J2 – Перемички для кінцевих резисторів.

D1-D4 – діоди 1N4007

Живлення приладу 24 В, що приходять по означенню VCC та GND, проходить через R2, що використовується у якості запобіжника та при перевищенні по току, виходить зладу. Після цього живлення проходить через резистори R3 та R4, утворюючи розділювач напруги між яким утворюється напруга до 2.5 В, що подаються на катод U2, де мікросхема

стабілізує напругу до 5 В на аноді та REF. Потім напруга знижена до 5 В живить C1, використовується для накопичення заряду та для підвищення можливостей стабілізатору по струму живлення. В свою чергу C2 та C3 використовуються для зменшення стрибків по напрузі при живленні компонентів приладу. Після конденсаторів, напруга стабільна та відфільтрована потрапляє на живлення інших компонентів схеми, а саме U1, U3, U5, U6, а також H1, який використовується для під'єднання резервного живлення від акумулятору (3.7В), та у разі вимкнення, стане живити систему, живлення на компоненти U4 та U7 поступає не за для живлення, а у якості опорної напруги через R11-R14, це потрібно для зменшення чистоти напруги по каналам А та В. До U1, окрім живлення що поступає на пін 4, 5, 18 та 21, поступають дані з U3 на пін 10, який в свою чергу стягнутий R1 до живлення для стабілізації передачі даних. Пін 20 від U1 під'єднан до GND послідовно через C4, для стабілізації опорної напруги U1, 1.3В за для коректної роботи аналогових входів U1, пін 26, який є аналоговим під'єднан до подільовача напруги, для вимірювання напруги на H1, та у разі потреби через під 32, який контролює Q1 та відключити акумулятор від аварійного живлення приладу, за для того щоб не перевантажувати систему стабілізації напруги від якою живиться прилад. Пін 27 та 28 це шина даних I<sup>2</sup>C або TWI, використовується ця шина за для передачі даних для візуалізацію даних з датчику U3 по місцю розташуванню приладу. 30 та 31 пін U1 це шина даних UART, що передає дані від U1 до пінів 3 та 4 елементу U6 через резистори R7 та R8, окрім цього на 1 та 2 пін елементу U6 під'єднані до пін 16 через резистор R6, та відповідає для отримання дозволу на передачу даних.

Елемент U6 отримавши дані від U1 конвертує їх з UART у RS-485 на пінах 7 (B) та 6 (A), де послідовно підключені елементи D1-D4, вони використовуються у ролі у разі якщо користувач сплутає під'єднання RS-485, та захистить від переполюсовки, далі сигнал після захисту

потрапляє на піни RX та TX, елементів U4 та U7, що в свою чергу є лише вивідними роз'ємами з приладу та передають сигнал на кабель зв'язку (кручена пара). Окрім цього на U4 та U7 на піни RX стягнути резистори R12, R11 до VCC, а на TX стягнути R13, R14 до GND, це потрібно для меншого розбігу по напрузі між каналами А та В. Також до пінів RX на елементах U4 та U7 під'єднані елементи R9 та R10 ці елементи виконують роль термінаторів, а вони в свою чергу під'єднані до перемичок J1 та J2, котрі у разі потреби, точніше у разі закінчення послідовності приладів у мережі замикаються на TX.

## 2.4 КОД НА ARDUINO C ДЛЯ ПРИЛАДУ

### 2.4.1 ПІДГОТОВКА МІКРОКОНТРОЛЕРУ ПІД МОВУ ARDUINO C

В приладі перехід від простого приладу, тестового зразка який був створений на «колінці» зазнав не тільки як у залізі, а також у кодї, хоча слід зауважити, що використання ARDUINO C не є досить привабливим для готового приладу, вже вище зазначеним причинами, але використовується завдяки тому що, мову ARDUINO C, є досить легко записати до мікроконтролеру, що використовується, для цього потрібен «програмактор» або навіть інша плата ARDUINO, що й використаємо для прикладу простого «прошивання» мікроконтролерів. Спочатку під'єднаємо плату ARDUINO до ПК через кабель USB, він виступить у якості живлення плати та мікроконтролеру. Наприклад у якості плати ARDUINO, використати ARDUINO NANO, тому що мікроконтролер легше буде розмістити на неспаяній макетній платі, як і саму плату ARDUINO NANO. Потім відкриваємо ARDUINOIDE, це компілятор ARDUINO C та заходимо до прикладу ARDUINOSPI, цим кодом потрібно спочатку «прошити» плату ARDUINO NANO, де знаходиться приклад для прошивки вказано на рис 20.

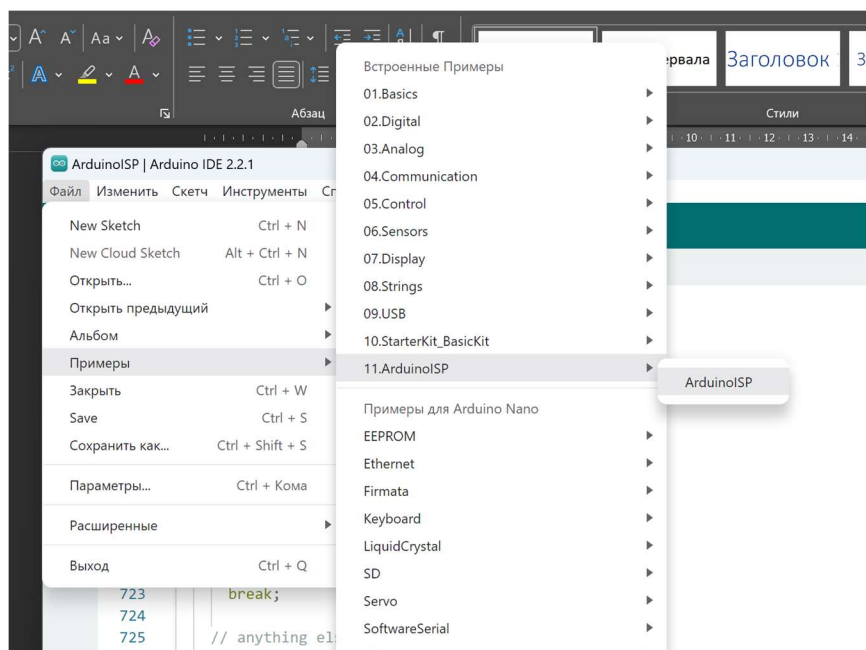


Рис 20 – шлях до коду для «прошивання» плати ARDUINO NANO

Після цього прошивши плату під'єднуємо мікроконтролер до плати ARDUINO NANO згідно рис 21, у моєму випадку це ATMEGA328PB.

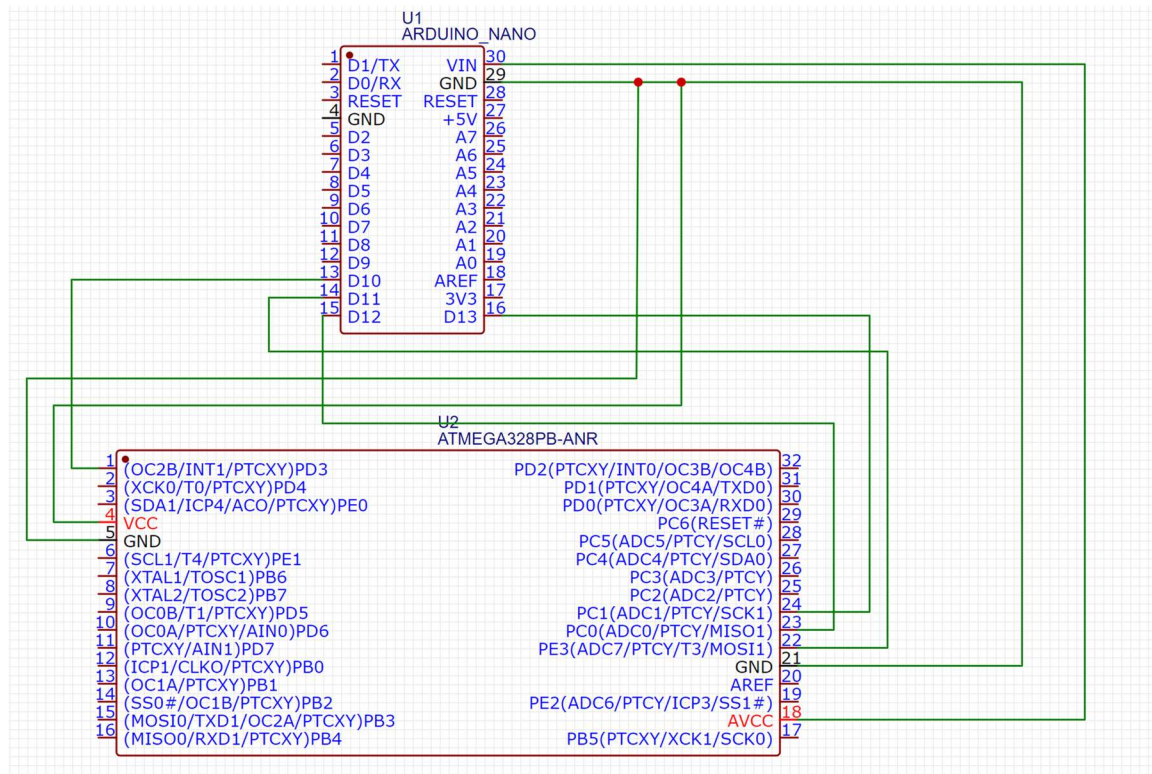


Рис 21 – схема підключення ATMEGA328PB до ARDUINO NANO

А саме 10 пін плати до RESET, 11 до MOSI, 12 до MISO, 13 до SCK, Vin та GND це живлення мікроконтролеру стабільними 5 В, якщо потрібно менше, наприклад 3.3 В то можна скористатись піном 3.3 В для живлення мікроконтролеру, також не слід забувати і про підключення кварцевого генератору на потрібні піни мікроконтролеру, у моєму випадку це 7 та 8 пін на який я підключав кварцевий генератор з частотою на 16 МГц. Потім, як все підключили, підготували плату, переходимо до налаштувань ARDUINOIDE під наш мікроконтролер, для цього переходимо у вкладку ІНСТРУМЕНТИ – ПРОГРАМАТОР та обираємо ARDUINO AS ISP, приклад доданий на рис 22.

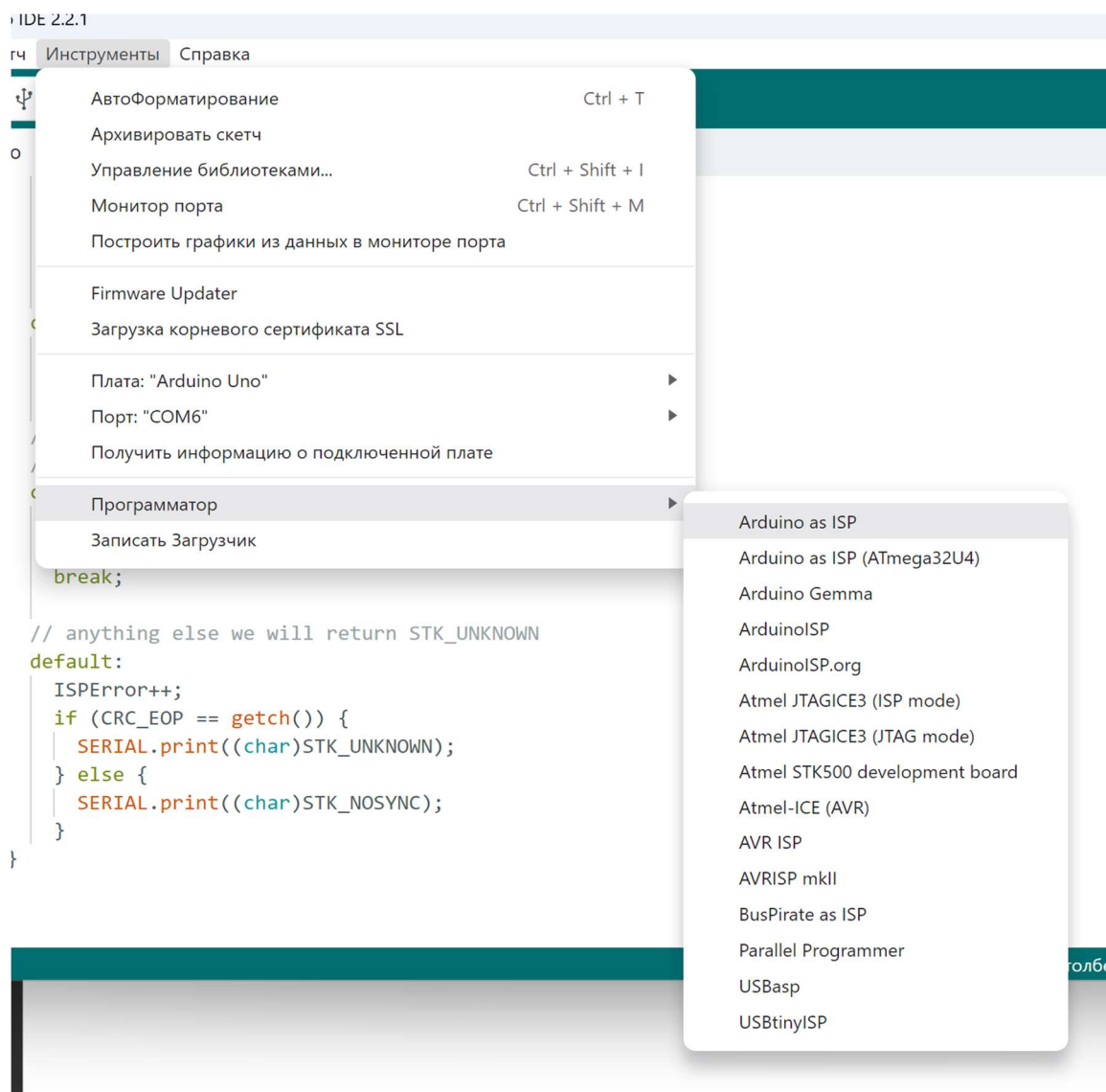


Рис 22 – приклад обирання програматору для прошивання мікроконтролеру через ARDUINO.

Після чого все готове для прошивання мікроконтролеру, але слід бути уважним, тому що у звичайній ARDUINOIDE є можливість обрати лише мікроконтролери сімейства ATMEGA, які й використовуються у платах ARDUINO, а якщо це сімейство ATTINY або PIC то вже потрібно шукати як це зробити. Потім після вибору програматору, починаємо записувати BOOTLOADER, для цього заходимо до ІНСТРУМЕНТИ – ЗАПИСАТИ ЗАГРУЗЧИК, як показано на рис 23.

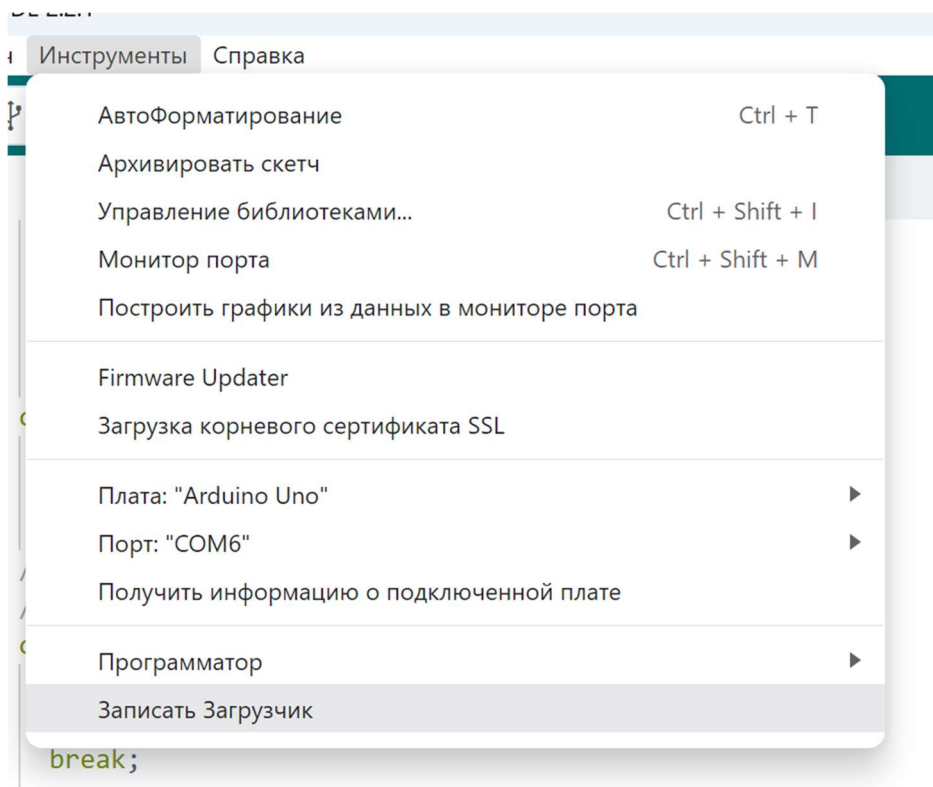


Рис 23 – вибір «ЗАПИСУ ЗАГРУЗЧИКУ»

Цікавий факт, BOOTLOADER це функція резервування пам'яті у сімейства ATMEGA та ATTINY, та в залежності від мікроконтролеру це може бути різна кількість пам'яті EEPROM, якщо використовувати C++/C то є можливість взагалі, написати ті функції та параметри роботи, які будуть зарезервовані і не будуть видаляться, при кожному прошиванні, таким чином при заглибленому використанні мікроконтролерів цих сімейств дає змогу писати навіть свою програму, яка віддалено буде нагадувати операційну систему, що й зробили виробники ARDUINO.

Після того як зашили в мікроконтролер BOOTLOADER, переходимо до написання коду.



## 2.4.2 НАПИСАННЯ КОДУ ОСНОВНОЇ ПРОГРАМИ

До програмного коду описаний 2.2.2, додається підключення бібліотеки MODBUSRTU. Тому новий код виглядає так:

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ArduinoRS485.h>
#include <ArduinoModbus.h>

/*Підключаємо потрібні нам бібліотеки для роботи з датчиком
температури DS18b20,
а також з екраном та режимом сну*/
constexpr auto baudrate { 19200 };
constexpr auto bitduration { 1.f / baudrate };
constexpr auto wordlen { 9.6f }; // try also with 10.0f
constexpr auto preDelayBR { bitduration * wordlen * 3.5f * 1e6 };
constexpr auto postDelayBR { bitduration * wordlen * 3.5f * 1e6 };
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define SCREEN_ADDRESS 0x3C
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);
#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16

/*Вказуємо адресу екрану, та його властивості, як ширина та висота*/
```

```

#define ONE_WIRE_BUS 5 //Змінна, якою ми підключаємо датчик
DS18B20

#define TEMPERATURE_PRECISION 12 //Змінна, якою ми задаємо
точність датчику

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensor(&oneWire);
float rawParameters = 0;
float Parameters = 0;
float temperatureCelsius = 0; //Початкова температура змінної
void setup() {
    sensor.begin();
    display.begin();
    RS485.setDelays(preDelayBR, postDelayBR);
    ModbusRTUClient.begin(baudrate, SERIAL_8E1);

    /* Блок програми в якому присвоєно включення потрібних бібліотек та
пінів МК,
    окрім цього цей блок програми МК робить лише один раз за весь час
роботи*/
}

void loop() {
    /* Основний блок програми, це бескінечний цикл, в якому є умова
для перевірки стану кнопки, окрім цього, в залежності від стану
кнопки,
    працюють потрібні алгоритми програми*/
    ModbusRTUClient.requestFrom(1, HOLDING_REGISTERS, 0x00, 2);
    rawParameters = temperatureCelsius;
    rawParameters = Parameters;
    Parameters = ModbusRTUClient.coilWrite(1, 0x00, 0x01);

```

```

}

void Temperature0() { //блок програми, що відповідає за замір
температури, а також вивід даних на дисплей
    sensor.requestTemperatures(); //Під'єднання до датчику DS18b20
    temperatureCelsius = sensor.getTempCByIndex(0); //переназначення
датчику до змінної
    display.clearDisplay(); //спочатку очищаємо дисплей
    display.setTextSize(2); //Встановлюємо висоту тексту
    display.setCursor(0, 0); //початкову позицію тексту
    display.print("Temperature: "); //Вивід тексту
    display.print(temperatureCelsius); //Вивід змінної де збережені дані
температури
    display.println("°C"); //Вивід тексту
    display.display(); //Закінчення роботи з дисплеєм
}

```

Такими змінами отримується передача даних по протоколу MODBUS, що надає можливість передавати дані по універсальному протоколу та за можливістю працювати у 1 рівні Автоматичної Систему Управління Технологічного Процесу (АСУ ТП). Окрім цього слід відмітити, що до дисплею, цей дисплей використовується для виведення параметрів по місцю та окрім цього за потреби, як незалежне обладнання.

Також використовується можливість під'єднання аварійного живлення тому додається ще частина коду та фінальна версія коду виглядає так:

```

#include <OneWire.h>
#include <DallasTemperature.h>
#include <SPI.h>
#include <Wire.h>

```

```

#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ArduinoRS485.h>
#include <ArduinoModbus.h>

/*Підключаємо потрібні нам бібліотеки для роботи з датчиком
температури DS18b20,
а також з екраном та режимом сну*/
constexpr auto baudrate { 19200 };
constexpr auto bitduration { 1.f / baudrate };
constexpr auto wordlen { 9.6f }; // try also with 10.0f
constexpr auto preDelayBR { bitduration * wordlen * 3.5f * 1e6 };
constexpr auto postDelayBR { bitduration * wordlen * 3.5f * 1e6 };
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define SCREEN_ADDRESS 0x3C
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);
#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16
/*Вказуємо адресу екрану, та його властивості, як ширина та висота*/
#define ONE_WIRE_BUS 5 //Змінна, якою ми підключаємо датчик
DS18b20
#define TEMPERATURE_PRECISION 12 //Змінна, якою ми задаємо
точність датчику
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensor(&oneWire);
float rawParams = 0;
float Params = 0;
float temperatureCelsius = 0; //Початкова температура змінної

```

```

int R16 = 680;
int R15 = 200;
float U = 0;
float Uin = 0;
int Transistor = 5;
void setup() {
  sensor.begin();
  display.begin();
  RS485.setDelays(preDelayBR, postDelayBR);
  ModbusRTUClient.begin(baudrate, SERIAL_8E1);
  pinMode (Transistor, OUTPUT);
  /* Блок програми в якому присвоєно включення потрібних бібліотек та
пінів МК,
    окрім цього цей блок програми МК робить лише один раз за весь час
роботи*/
  }

void loop() {
  /* Основний блок програми, це бескінечний цикл, в якому є умова
для перевірки стану кнопки, окрім цього, в залежності від стану
кнопки,
    працюють потрібні алгоритми програми*/
  ModbusRTUClient.requestFrom(1, HOLDING_REGISTERS, 0x00, 2);
  rawParametr = temperatureCelsius;
  rawParametr = Parametr;
  Parametr = ModbusRTUClient.coilWrite(1, 0x00, 0x01);
  if (Uin <= 3.3 && Uin >= 4.0) {
    Transistor = 1;
  } else {
    Transistor = 0;
  }
}

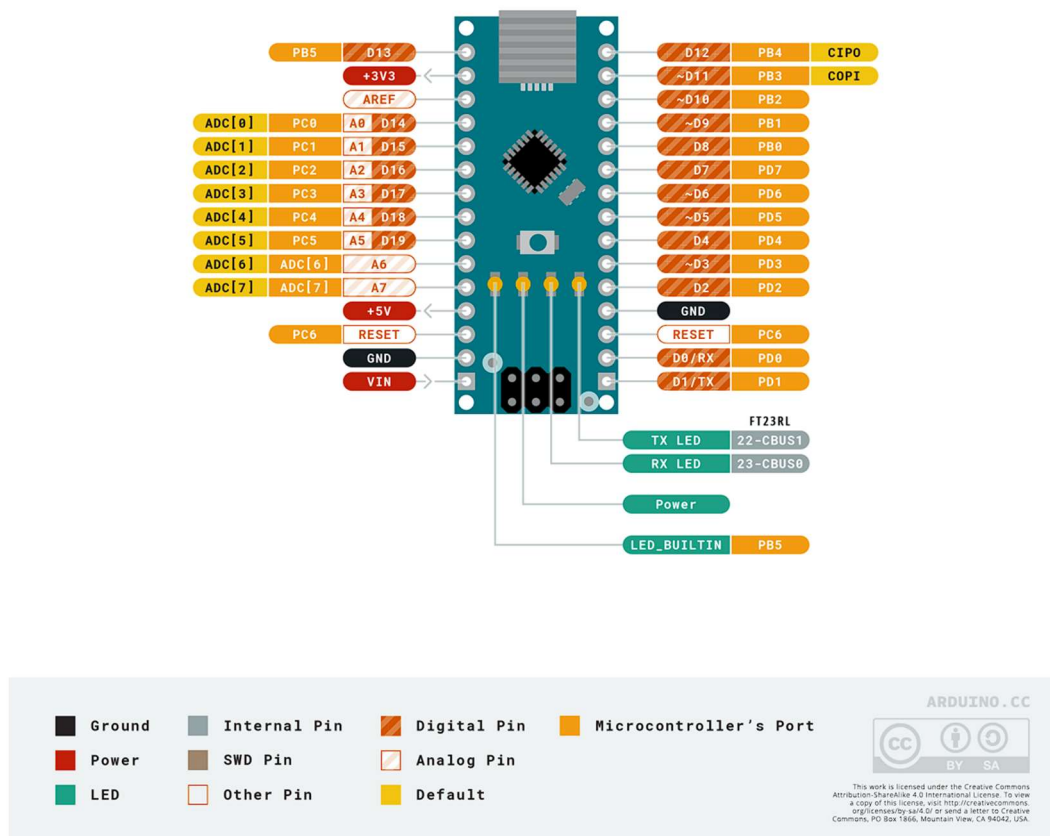
```

```
}  
}
```

```
void Temperature0() { //блок програми, що відповідає за замір  
температури, а також вивід даних на дисплей  
    sensor.requestTemperatures(); //Під'єднання до датчику DS18b20  
    temperatureCelsius = sensor.getTempCByIndex(0); //переназначення  
датчику до змінної  
    display.clearDisplay(); //спочатку очищаємо дисплей  
    display.setTextSize(2); //Встановлюємо висоту тексту  
    display.setCursor(0, 0); //початкову позицію тексту  
    display.print("Temperature: "); //Вивід тексту  
    display.print(temperatureCelsius); //Вивід змінної де збережені дані  
температури  
    display.println("°C"); //Вивід тексту  
    display.display(); //Закінчення роботи з дисплеєм  
}  
void Current() {  
    U = analogRead(A3)/1024;  
    Uin = U * (R15/(R16+R15));  
  
}
```

## 2.5 РІЗНИЦЯ У РОЗКЛЮЧЕНІ МІЖ ПЛАТОЮ ARDUINO ТА МІКРОКОНТРОЛЕРУ

Між платою ARDUINO та мікроконтролером ATMEGA328PB має велику різницю, наприклад плата ARDUINO NANO з сайту розробника виглядає так:



Power Tree of Arduino Nano

Рис 24 – розключення плати ARDUINO NANO

А розключення ATMEGA328PB показано на рис 25, яке було взятє з даташиту цього мікроконтролера.

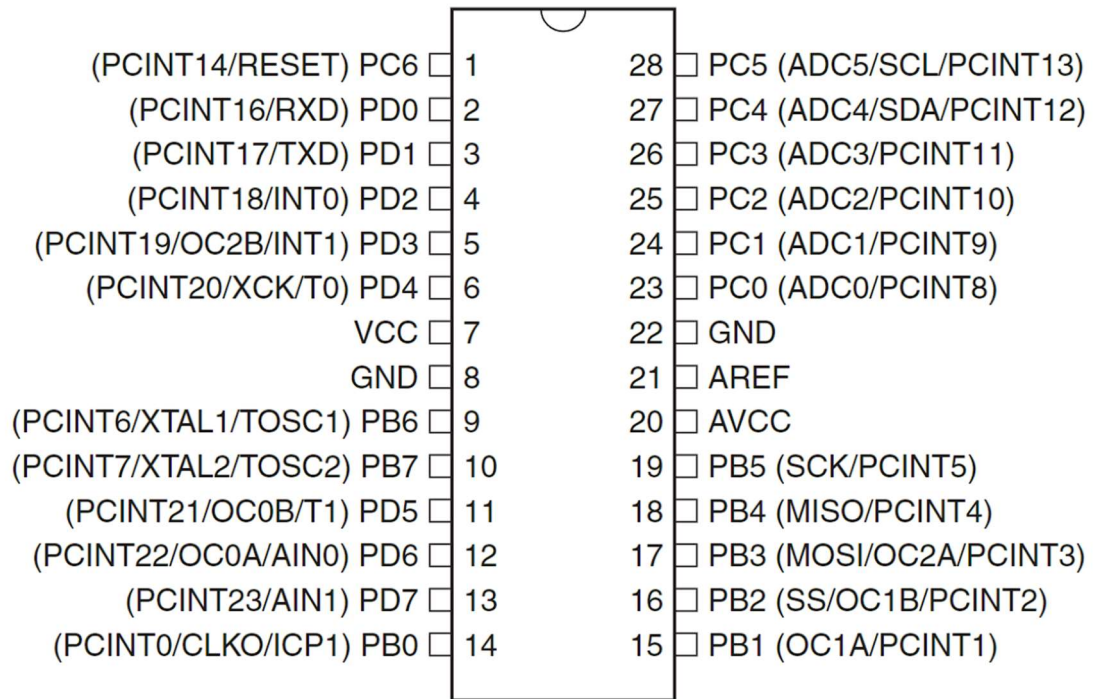


Рис 25 – роз'язучення АТМЕГА328



### 3 ПРАКТИЧНЕ ВИКОРИСТАННЯ ПРИЛАДУ

В цьому пункті буде представлено використання більш ранньої моделі приладу, що використовувався для лабораторних вимірювань під час проведення осадження оксидів цинку.

Для початку потрібно зрозуміти, що для проведення експерименту використовувалась плата ARDUINO UNO та LCD 1602 дисплей з встановленим модулем I<sup>2</sup>C для поліпшення збору приладу.

У якості реагентів використовували нітрат цинку, карбамід та ацетат цинку, рН середовища було приблизно 7, температура наприкінці процесу становила 125-130 °C.

В п'ятигорлу колбу 3, наливали розчин цинк нітрату (цинк ацетату), після чого додавали розраховану кількість карбаміду. Колбу розміщували у піщаній бані 2 з електрообігрівачем 1, після чого у колбу занурювали цифровий датчик температури 4 типу Dallas Temperature 18B20 у водозахисному корпусі. Вимірювання температури здійснювалося з похибкою в  $\pm 0,5$  °C, діапазон температур від -55 до +125 °C. В центральне горло розміщували перехідник 5, а всі інші горла колби закривали гумовими пробками 6. До перехідника 5 під'єднали холодильник 6, з виходу холодильника алонж 8, який одним кінцем був занурений у приймач конденсату 9. Цифровий датчик температури Dallas Temperature 18B20 було під'єднано до плати ARDUINO UNO на базі мікроконтролера ATmega328 - PV 10, у якості дисплея для виведення інформації стосовно температури був використаний дисплей LCD 1602 з шиною I<sup>2</sup>C. Контроль температури здійснювався періодично відповідно до відбору проб.

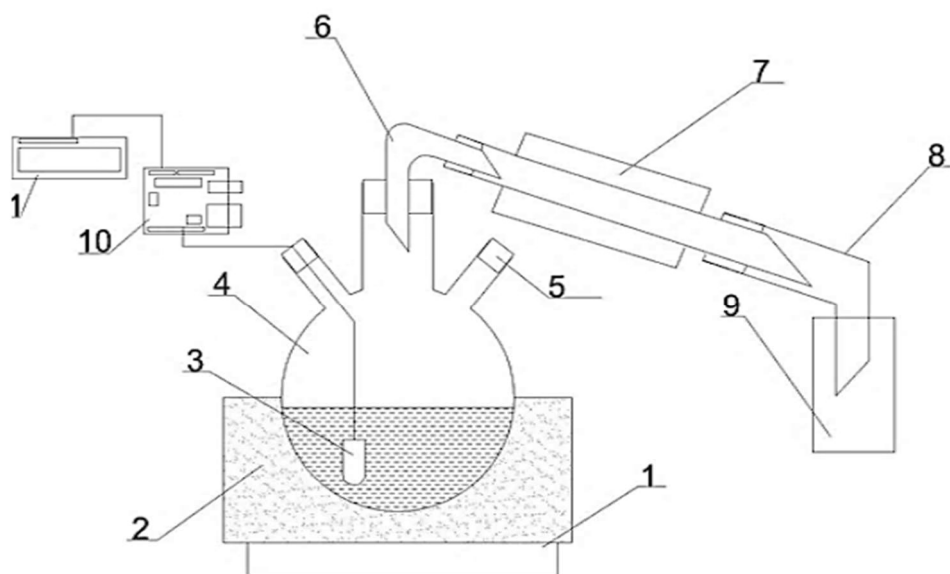


Рис 26 – експериментальна установка для осадження оксидів цинку:  
 1 – електропіч; 2 – піщана баня; 3 – цифровий датчик температури DS18B20; 4 – п'яти горла колба; 5 – пробка гумова; 6 – перехідник; 7 – холодильник; 8 – алонж; 9 – збірник конденсату; 10 – Плата ARDUINO UNO; 11 – LCD дисплей 1602

Отримані дані були використані при написанні статті, що була опублікована у 2019 року. Тому хімічна частина потрібна бути опущена так як не несе потрібної інформації для проекту, але вказати все ж таки необхідно.

У результаті дослідження було виявлено наступне – при температурі 92°C почався гідроліз карбаміду та виділення вуглекислого газу, що створювало видимість кипіння розчину. При температурі 101°C розчин почав кипіти, початок кипіння визначали за появою конденсату у холодильнику. По мірі відгонки конденсату температура кипіння зростала, при цьому безперервно утворювався вуглекислий газ. При температурі 101°C починалось випадіння осаду. Процес осадження цинку з розчину представлено на рис. 27.

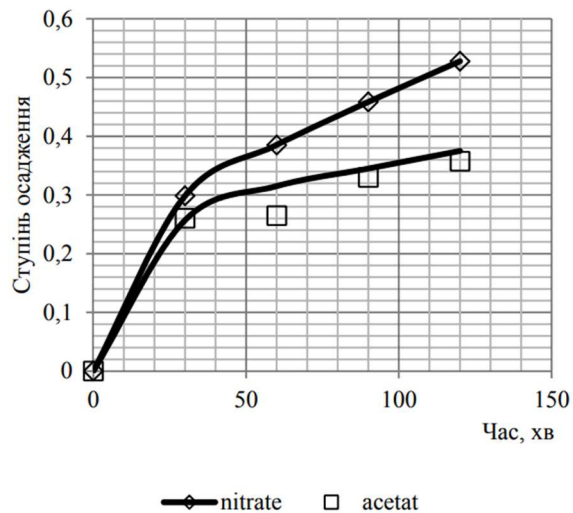


Рис 27 – залежність осадження від часу

Осадження припиняли по досягненні температури для цинк нітрату 103,25 °C і 102,59 °C для цинк ацетату Початок утворення осаду було відзначено через 30 хвилин від початку експерименту.

Також слід зазначити що у якості реагентів використовувались не досить нейтральні речовини, що повинні були пошкодити датчик температури, але пошкоджень датчику не було, що також показало стійкість датчику, та можливість подальшого використання у тож самому середовищі у тривалий час без подальших проблем. Окрім цього проведення випробувань у більш агресивних середовищах для перевірки подальших можливостей для майбутніх пристроїв [9].

#### 4 МОЖЛИВЕ ВИКОРИСТАННЯ ПРИЛАДУ

При вже показаній практиці з більш раннім прототипом все ж таки використання у хімічній промисловості не обмежується, якщо брати рівень побуту то це використання у приміщеннях для клімат контролю, а саме опалення, охолодження приміщення, також може використовуватись як прилад контролю нагріву поверхні, наприклад акумулятору автомобілю, а використання можливості для аварійного живлення, можна так контролювати акумулятор та його нагрів під час заряджання.

Якщо брати за увагу промисловість то одразу використання у харчовій промисловості де не є висока температура, не більше 130 °С, також нейтральність матеріалів датчику та невеликою агресивністю середовища (миючі речовини трубопроводів). Енергетична промисловість у теплообмінниках для нагріву середовища до 130 °С або охолодження, також для розрахування втрат температури. Наприклад є витратомір який обчислює витрату гарячої води, а на вході прилад, що заміряє температуру, а на виході з приладу на трубопроводі стоїть інший витратомір за датчиком температури. Майстер такої зв'язки приладів розраховує різницю витрат та температури і розраховує використання тепла. Це можна використати для локального розрахунку використання газу на об'єкті, також для створення системи моніторингу підприємства для централізованого опалення об'єктів які підпорядковані котельні та ведуть підприємницьку справу.

## ВИСНОВОК

За результатами роботи, було створено пристрій, якій пройшов майже всі стадії розвитку, як для комерційного проекту. А саме було висунуто вимоги до проекту, це:

Легкість у проектуванні;

Використан дешевий датчик температури;

Легкість у повторенні за потреби;

Легкість у створенні програми проекту та його використанні;

Окрім цього прототипи проекту були використані у хімічній інженерії, та використовувалися не в технологічному процесі, а в лабораторних умовах, але це все одно дало результат ефективності для подальшого використання у технологічних процесах хімічної промисловості, хоча питання стійкості корпусу датчику температури до більш агресивних середовищ (кислоти, луги , тощо) ще виникають питання.

Також потрібно відмітити, що використання приладу не завершується лише у хімічній інженерії, а може ще використовуватися для харчової, легкої промисловості, енергетиці (контроль температури гарячої води, нагрів акумулятору підчас зарядки) окрім промисловості також є можливість використовувати й на побутовому рівні.

## ЛІТЕРАТУРА

1. Крамарухин Ю. Е. Приборы для измерения температуры. — М.: Машиностроение, 1990. 208 с: ил.
2. DS18B20 Programmable Resolution 1-Wire Digital Thermometer [Электронный ресурс] // Dallas Semiconductor. – 2008. – Режим доступа до ресурсу: <https://www.rcscomponents.kiev.ua/datasheets/DS18B20-28978.pdf>.
3. Бергер Г. Автоматизация с помощью программ STEP7 LAD и FDB / Ганс Бергер. – Эрланген: SIEMENS CO, 2001. – 605 с. – (SIEMENS).
4. ATmega48A/PA/88A/PA/168A/PA/328/P [Электронный ресурс] // MICROCHIP. – 2020. – Режим доступа до ресурсу: <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>.
5. ARDUINO NANO [Электронный ресурс] // ARDUINO. – 2023. – Режим доступа до ресурсу: <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>.
6. STM32F401xB STM32F401xC [Электронный ресурс] // STlife.augmented. – 2019. – Режим доступа до ресурсу: <https://www.st.com/resource/en/datasheet/stm32f401cb.pdf>.
7. MAX481/MAX483/MAX485/ MAX487–MAX491/MAX1487 [Электронный ресурс] // Maxim Integrated. – 2014. – Режим доступа до ресурсу: [https://pdf.voron.ua/files/pdf/ms-prod/maxim\\_integrated/MAX1487-MAX491-96074.pdf](https://pdf.voron.ua/files/pdf/ms-prod/maxim_integrated/MAX1487-MAX491-96074.pdf).
8. TL431, TL432 Precision Programmable Reference [Электронный ресурс] // Texas Instruments. – 2004. – Режим доступа до ресурсу: [https://www.ti.com/lit/ds/symlink/tl431.pdf?ts=1701546129108&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/tl431.pdf?ts=1701546129108&ref_url=https%253A%252F%252Fwww.google.com%252F).

9. Денисов О. С., Корчуганова О. М. Отримання цинку оксиду: карбамідне осадження. — Вісник СХУ ім. В. Даля, 2019, 30-34 с.