

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
Факультет інформаційної безпеки та електронних комунікацій
(повне найменування факультету)
Кафедра інформаційної безпеки та наноелектроніки
(повне найменування кафедри)

Пояснювальна записка
до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему «Дистанційний моніторинг автоматизованого контролю температури
приміщення»

(назва теми)

Виконала: студентка II курсу, групи БК-313-м
Спеціальності 176 Мікро- та наносистемна техніка
Освітня програма (спеціалізація)

Мікро- та наноелектронні прилади і пристрої

ЧАГОВЕЦЬ А.О.

(ПРИЗВИЩЕ та ініціали)

Керівник

СНІЖНОЙ Г. В.

(ПРИЗВИЩЕ та ініціали)

Рецензент

ПРАВДА М. І.

(ПРИЗВИЩЕ та ініціали)

Міністерство освіти і науки України

Національний університет «Запорізька політехніка»

Факультет Факультет інформаційної безпеки та електронних комунікацій

Кафедра Кафедра інформаційної безпеки та наноелектроніки

Ступінь вищої освіти магістр

Спеціальність 176 Мікро- та наносистемна техніка

(код і найменування)

Освітня програма (спеціалізація) Мікро- та наноелектронні прилади і пристрої

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ІБтаН

Андрій КОРОТУН

« _____ » _____ 20__ року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЧАГОВЕЦЬ Ангеліни Олександрівни

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дистанційний моніторинг автоматизованого контролю температури приміщення, Remote Monitoring System for Automated Room Temperature Control.

керівник проєкту (роботи) д. техн. наук, проф. СНИЖНОЙ Геннадій Валентинович

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затвержені наказом закладу вищої освіти від « 5 » грудня 2024 року № 507

2. Строк подання студентом проєкту (роботи) 15 грудня 2024

3. Вихідні дані до проєкту (роботи) Arduino, датчик температури TMP36, Arduino IDE, платформа Tinkercad, Wi-Fi модуль ESP8266, макетна плата, додаток Telegram.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Проаналізовано системи моніторингу температури та окреслено доцільність використання платформи Arduino для системи. Створено модель дистанційного моніторингу оповіщення системи контролю температури за допомогою інтеграції її з Telegram.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

16 слайдів у презентації PowerPoint.

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	Сніжної Г.В., доктор. техн. наук, проф.		
Нормо-контроль	Корольков Р.Ю., канд.тех.наук, доцент		

7. Дата видачі завдання «02» _____ вересня _____ 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
	Огляд літератури за темою проєкту. Складання та систематизація змісту проєкту	14.09-28.09	
	Аналіз можливостей платформи Arduino для систем моніторингу.	03.10-11.11	
	Розробка коду для зчитування даних з мікроконтролеру та для інтеграцій з Telegram API	12.11-16.11	
	Розробка прототипу дистанційного моніторингу та оповіщення системи контролю температури на платформі Arduino	18.11-20.11	
	Оформлення звіту	25.11-15.12	

Студент(ка) _____

(підпис)

Ангеліна ЧАГОВЕЦЬ

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи) _____

(підпис)

Геннадій СНІЖНОЙ

(Ім'я ПРИЗВИЩЕ)

АНОТАЦІЯ

Пояснювальна записка до магістерської роботи: 80 с., 6 табл., 22 рис., 3 дод., 24 джерела.

МОНІТОРИНГУ, ОПОВІЩЕННЯ TELEGRAM, СИСТЕМА, ARDUINO, ARDUINO UNO, TELEGRAM.

Об'єкт дослідження – платформа Arduino , моніторингові системи температури та додаток Telegram.

Мета проекту – дослідити та проаналізувати існуючі системи моніторингу температури в приміщенні, дослідити та обґрунтували доцільність використання платформи Arduino для систем моніторингу температури приміщення. Розробити проект дистанційного моніторингу температури в приміщенні з оповіщенням за допомогою telegram-бота.

Метод дослідження – аналіз з подальшим моделюванням.

Результати: Складена схема апаратного забезпечення для системи моніторингу на основі плати Arduino Uno, розроблений код для збору й обробки даних з датчика температури, а також розроблений спосіб інтеграції з Telegram системи моніторингу для оповіщення користувача.

ANNOTATION

The explanatory note to the master's thesis: 80 pages, 6 tables, 22 figures, 3 appendices, 24 references.

MONITORING, TELEGRAM NOTIFICATIONS, SYSTEM, ARDUINO, ARDUINO UNO, TELEGRAM.

The object of the research is the Arduino platform, temperature monitoring systems, and the Telegram application.

The aim of the project is to investigate and analyze existing indoor temperature monitoring systems, study and justify the feasibility of using the Arduino platform for indoor temperature monitoring systems, and develop a project for remote indoor temperature monitoring with notifications via a Telegram bot.

Research method – analysis followed by modeling.

Results: A hardware schematic for a monitoring system based on the Arduino Uno board has been developed, code for collecting and processing data from the temperature sensor has been created, and a method for integrating the monitoring system with Telegram for user notifications has been designed.

ЗМІСТ

	С.
Перелік умовних скорочень	7
Вступ	8
1 Аналіз існуючих систем моніторингу температури у приміщеннях	9
1.1 Аналіз стандартів та нормативів щодо контролю температури	9
1.2. Аналіз існуючих засобів контролю кліматичних показників в приміщеннях	14
1.3. Узагальнення результатів аналізу та постановка завдань дослідження	16
1.4. Вимоги до програмного модуля моніторингу параметрів системи	18
1.5. Висновок	19
2 Доцільність використання платформи Arduino для систем моніторингу температури приміщення	20
2.1 Особливості апаратної частини та можливості платформи Arduino	20
2.2 Види мікроконтролерів та їх можливості для створення систем збору даних	27
2.3 Принцип роботи Arduino	33
2.4 Принципи роботи сенсорів температури	37
2.5 Висновки до розділу	44
3 Реалізація дистанційного моніторингу та оповіщення системи контролю температури	45
3.1 Схема апаратного забезпечення	45
3.2. Розробка коду для збору даних з датчиків і їх обробки	47
3.3. Інтеграція з Telegram API для сповіщення користувача	59
Висновки	63
Перелік джерел посилань	64
Додаток А Схеми підключень до плати	67
Додаток Б Програмний код	71
Додаток В Програмний код інтернет модуля для підключення його до Telegram	79

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЦП – Аналого-цифровий перетворювач

ШИМ – широтно-імпульсної модуляція

GPIO – General-purpose input/output .Інтерфейс введення/виведення загального призначення.

I2C –Inter-Integrated Circuit

IoT – Internet of Things. Інтернет-речі

NTC – Negative Temperature Coefficient. Від'ємний температурний коефіцієнт

PTC – Positive Temperature Coefficient. Позитивний температурний коефіцієнт.

PWM – див. ШИМ

RTD – Резистивні сенсори температури

UART – Universal Asynchronous Receiver-Transmitter, універсальний асинхронний приймач-передавач.

ВСТУП

У сучасному світі автоматизація і дистанційний моніторинг стають невід'ємними складовими багатьох галузей, включаючи управління кліматом у приміщеннях. Ефективне регулювання температури не тільки підвищує комфорт для людей, але й має значний вплив на енергозбереження та екологічну стійкість. Зі зростанням попиту на розумні системи управління будівлями дедалі актуальнішим стає впровадження інноваційних рішень, які поєднують у собі автоматизований контроль і можливість дистанційного моніторингу. Традиційні системи опалення, вентиляції та кондиціонування часто працюють за статичними алгоритмами, які не враховують змінних зовнішніх і внутрішніх факторів. Це може призводити до перевитрат енергії, зниження ефективності роботи систем та небажаного впливу на навколишнє середовище.

У той же час розвиток технологій Інтернету речей (IoT) створює нові можливості для оптимізації кліматичних систем. Завдяки IoT пристроям, що здатні обмінюватися даними в режимі реального часу, з'являються рішення, які дозволяють не лише автоматично регулювати температуру, але й забезпечувати дистанційний доступ до даних та управління системою. Такі системи можуть відслідковувати зміни температури, аналізувати отриману інформацію і приймати оптимальні рішення для забезпечення комфортного середовища з мінімальними енергозатратами.

Метою даної магістерської роботи є розробка із можливим впровадження системи дистанційного моніторингу автоматизованого контролю температури приміщення, на основі платформи Arduino, та використання можливостей платформи для сполучення системи з додатком Telegram.

1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ МОНІТОРИНГУ ТЕМПЕРАТУРИ У ПРИМІЩЕННЯХ

1.1 Аналіз стандартів та нормативів щодо контролю температури

Контроль температури у виробничих і житлових приміщеннях є важливою умовою забезпечення якості продукції, комфорту та безпеки. У різних секторах використовуються стандарти й нормативи, які регламентують допустимі температурні діапазони, методи контролю та вимоги до обладнання.

Санітарні норми поширюються на мікрокліматичні умови в робочих приміщеннях виробничих об'єктів, таких як підприємства, установи, громадські заклади тощо незалежно від їх форм власності та належності. Ця специфікація не поширюється на підземні та шахтні роботи, мобільний транспорт, тваринницькі та птахофабрики, місця зберігання сільськогосподарської продукції, холодильники, склади та інші місце де має бути специфічний мікроклімат, а також місця, де параметри мікроклімату встановлюються відповідно до вимог процесу [1].

Параметри мікроклімату, які встановлюються відповідно до технологічних вимог, зазвичай регламентуються для приміщень, де кліматичні умови безпосередньо впливають на якість продукції, процес виробництва, або умови зберігання.

У різних типів приміщення є свої особливості з яких випливають і вимоги температурного контролю в них. Під терміном «виробничі приміщення» за санітарними нормами мікроклімату, постанова ДСН 3.3.6.042-99 мається за увазі замкнутий простір в спеціально призначених будинках та спорудах, в яких постійно, по змінах, або періодично, протягом частини робочого дня, здійснюється трудова діяльність людей [1].

До виробничих приміщень належать:

- цехи з високотемпературними процесами (металургія, ливарне виробництво, коксохімічне виробництво);

- хімічна промисловість;
- текстильні фабрики (обробка тканин і ниток);
- виробництво фармацевтики (чисті приміщення, асептичні умови);
- виробництво електроніки(збирання мікросхем, збирання техніки).

Металургійні заводи, коксохімічні підприємства, скловиробництво характеризуються інтенсивним виділенням тепла при виробленні своїх продуктів. Це вимагає до наступних кліматичних параметрів, а саме – температура повітря має бути знижена за допомогою вентиляції та охолоджувальних систем. Прикладом де це вимагає є ливарний цех, на якому виробляють автомобільні диски, процеси їх плавлення вимагають високих температур, тому у цехах організуються система охолодження для підтримання комфортної температури на рівні 20-25°C [2].

Складські приміщення:

- холодильні камери;
- склади сировини(деревина, хімікати, меблі) ;
- склади готової продукції (одяг, меблі) .

Склади для зберігання продукції підприємств, вимагають чітких температурних режимів для запобігання фінансових трат у разі псування продуктів. Наприклад в приміщенні де зберігаються хімікати потрібно дотримуватися термічних норм бо не дотримання може привести до аварій на виробництві від вибуху до хімічного отруєння [3].

Громадські приміщення з вимогами до експлуатуванням:

- спортивні комплекси та басейни;
- музеї, архіви, бібліотеки.

Будівлі по типу музею, архівів та бібліотек, дотримуються температурних норм, через експонатів які в них знаходяться, і якщо буде помилка, то це може призвести до фінансових штрафів або культурних втрат для країни та світу в цілому.

Основні вимоги до температурних параметрів встановлюються нормативними документами, зокрема санітарними нормами, які визначають

оптимальні та допустимі величини мікрокліматичних показників залежно від виду виконуваної роботи [4].

Згідно постанови міністерства охорони здоров'я в Україні Санітарні норми мікроклімату виробничих приміщень включають в себе оптимальні та допустимі величини температури в робочій зоні виробничих приміщень.

Що встановлюється та визначається в подібних документах:

- робоча зона – приміщення, де знаходиться робоче місце постійного або непостійного перебування працівників;

- постійні та непостійні робочі місця – місця які визначаються процентом робочого часу який витрачається на роботу, якщо більше 50% від робочого часу без пауз то це постійне а якщо менше то непостійне;

- мікроклімат виробничих приміщень - умови внутрішнього середовища цих приміщень, що впливають на тепловий обмін працюючих з оточенням. Ці умови визначаються поєднанням температури оточуючих людину поверхонь та інтенсивністю теплового (інфрачервоного) опромінення;

- оптимальні та допустимі кліматичні умови – поєднання параметрів мікроклімату, які при тривалому та систематичному впливі на людину забезпечують нормальній тепловий стан організму без активізацій механізмів терморегуляції, або у разі виникнення погіршення стану активізуються до нормалізації середовища в межах фізіологічної адаптації.;

- теплий\холодний період року – періоди року які характеризуються середньодобовою температурою зовнішнього повітря вище +10 °C у разі теплового періоду та рівно або менше +10 °C у разі холодного;

- середньодобова температура зовнішньої повітря – визначена середня температура, яку виміряли у певний термін та через однакові інтервали часу;

- категорія робіт – класифікація робіт за їх важкістю та енерговитрату для організму під час виконання;

- легкі фізичні роботи категорії Ia та Ib- види діяльності, в яких витрата енергії організму дорівнює 105-175 Вт (90-150 ккал/год). Де категорія Ia це вид

робіт які виконуються сидячи і не потребують фізичних навантажень, та категорія Іб в якій супроводжуються невеликими фізичними навантаженнями;

- середня важкість фізичних робіт категорії Па та Пб – види діяльності при яких витрати енергії організмом становить 176-290 Вт (151-250 ккал/год). Категорія Па це праця пов'язана з фізичним переміщенням у просторі дрібних до 1 кг виробів або предметів в положенні стоячи або сидячи і що потребують певного навантаження на організм. До категорії Пб що виконуються стоячи, з переміщенням вантажів до 10 кг і супроводжуються помірним фізичним навантаженням;

- висока важкість фізичних робіт категорії ІІІ- види діяльності, в яких витрати організмом енергії становить 291-349 Вт(251-300 ккал/год). До цієї категорії робіт належать постійні переміщення важких вантажів.

Таблиця 1.1 – Оптимальна температура повітря в робочій зоні виробництва.

Типи робіт	ЄС	Україна
1	2	3
Легка Іа(тепл\холод)	19-23°C	22-25°C
Легка Іб(тепл\холод)	Сидячий, стоячий тип роботи	21-24°C
Серед.важк ІІа(тепл\холод)	17-19°C	19-23°C
Серед.важк ІІб(тепл\холод)	Сидячий, стоячий тип роботи	19-22°C
Важка ІІІ(тепл\холод)	До 12°C Стоячий, активний тип роботи	16-20°C

Для робочої зони виробничих приміщень встановлюються оптимальні мікрокліматичні умови з урахуванням важкості виконуваної роботи та періоду року.

При одночасному виконанні в робочій зоні робіт різної категорії важкості рівні показників мікроклімату повинні встановлюватись з урахуванням найбільш чисельної групи працівників [5].

При цьому роботодавці зобов'язані проводити оцінку ризиків, включно з впливом температури, і впроваджувати профілактичні заходи, як-от системи вентиляції або перерви для працівників, щоб зменшити ризик теплового стресу.

Допустимі величини мікрокліматичних умов встановлюються у випадках, коли на робочих місцях не можна забезпечити оптимальні величини мікроклімату за технологічними вимогами виробництва, технічною недосяжністю та економічно обґрунтованою недоцільністю.

Величини показників, які характеризують допустимі мікрокліматичні умови, встановлюються для постійних і непостійних робочих місць, які наведені в табл. 1.2 [5].

Перепад температури повітря по висоті робочої зони при забезпеченні допустимих умов мікроклімату не повинен бути більше 3°C для всіх категорій робіт, а по горизонталі робочої зони та протягом робочої зміни – виходити за межі допустимих температур для даної категорії роботи, вказаних в табл. 1.2

Порівняння стандартів.

Порівнюючи нормативи України та ЄС, можна виділити такі ключові аспекти:

- В Україні встановлені конкретні температурні межі залежно від категорії фізичного навантаження;
- У країнах ЄС більше уваги приділяється адаптації умов праці для запобігання тепловому чи холодовому стресу;
- Українські норми більш деталізовані у визначенні параметрів мікроклімату, тоді як європейські стандарти частіше орієнтовані на ризик-орієнтований підхід.

Таблиця 1.2 Допустимі величини температури повітря в робочій зоні.

Період року	Категорія робіт	Температура °С			
		Верхня межа		Нижня межа	
		На постійних Робочих місцях	На непостійних робочих місцях	На постійних Робочих місцях	На непостійних робочих місцях
1	2	3	4	5	6
Холодний період року	Легка Іа	25	26	21	18
	Легка Іб	24	25	20	17
Теплий період року	Легка Іа	28	30	22	20
	Легка Іб	28	30	21	19
	Середньої важкості Іа	27	29	18	17
	Середньої важкості Іб	27	29	15	15
	Важка ІІІ	26	28	15	13

1.2. Аналіз існуючих засобів контролю кліматичних показників в приміщеннях

Сучасні житлові будинки оснащені рядом технічних рішень для створення сприятливого мікроклімату в будинку – це системи вентиляції, опалення, кондиціонування, пристрої зволоження повітря. Вимагає контролю та уваги: налаштування режиму, включення та виключення тощо. Проте з системою

«розумний дім» усіма пристроями можна керувати через браузер ПК, планшета чи мобільного телефону [6].

Можна централізовано задати бажану температуру в кожному приміщенні, відстежити поточну, а також статуси кліматичних приладів. Запрограмувати час включення і виключення клімат-контролю за таймером.

Система «Розумного будинку» позбавляє від зайвих заходів і дозволяє використовувати кліматичний комплекс максимально ефективно – працюватимуть лише ті пристрої та прилади, які повинні працювати в даний момент.

Як і житлові приміщення, виробничі комплекси також мають подібну систему. Системи управління кліматом використовується для підтримки і зміни параметрів температури, вологості і циркуляції свіжого повітря всередині приміщень.

Сучасні системи клімат-контролю є досить складними системами автоматичного регулювання, які підтримують контрольовані параметри повітря в заданих межах за сигналами, отриманими від датчиків температури.

У роботі описана система моніторингу та системи оповіщення в приміщенні за допомогою датчика температури з звуковим сповіщувачем на програмно-апаратних засобах платформи Arduino з мікроконтролером Atmega328.

Мікроконтролер, в свою чергу, на стороні прийому виробляє обробку отриманої інформації з подальшим виведенням на монітор і на семисегментний індикатор пристрою.

Перевагою цього пристрою буде те що він може інформувати в разі критичних показників, за допомогою звукового сповіщення в його модулі та через чат бота в телеграмі. Пристрій може запобігти псуванню продуктів, що в свою чергу дає змогу нам вчасно вирішити проблему та зменшити фінансові витрати. Це дуже вигідно для маленьких\середніх виробництв а також для звичайних громадян через його економну вартість.

1.3. Узагальнення результатів аналізу та постановка завдань дослідження

Аналіз сучасних систем контролю кліматичних показників свідчить про важливість інтеграції автоматизованих рішень як у побуті, так і на виробництві. Системи «Розумного будинку» забезпечують централізоване управління кліматичними пристроями, дозволяючи знижувати витрати енергії та забезпечувати комфортний мікроклімат. Водночас, виробничі комплекси потребують точного моніторингу параметрів температури, для уникнення технічних і фінансових ризиків.

Розробка системи моніторингу на базі платформи Arduino дозволяє створити економічно вигідне рішення, що може бути адаптоване до потреб користувачів завдяки модульній архітектурі [8].

Таке рішення актуальне як для побутового використання так і для виробничих приміщень, де стабільність температури є критично важливою. Вчасне сповіщення про відхилення температури дозволяє швидко та чітко реагувати на проблему, зменшуючи фінансові втрати.

Наприклад, використання таких датчиків у побуті є доцільним у випадках, коли необхідно виявити несправність, як-от у ситуації з поламаним холодильником, де визначення проблеми без відкривання дверцят є складним завданням.

Момент відкриття холодильника може статися в будь який час- через пів години від моменту поломки так і через декілька годин коли всі продукти зіпсуються. І тому завдяки розробляємої системи в цьому проекті, людина дізнається вчасно про зміни температури та зменшити фінансові трати на продукти, до поки не відновить роботу холодильнику або купівлю нового. Цей приклад може статися не тільки в побуті але й на виробництвах де такі проблеми можуть призвести до дуже великих фінансових витрат, що до закриття підприємства буде не далеко.

Ціна на комплектуючі для зборки системи температурного моніторингу на базі Arduino в Україні залежить від обраних компонентів.

Arduino-плати:

- Arduino Nano V3.0 – від 113 грн;
- Arduino Uno KIT Mega Pack – стартовий набір з платою, датчиками та аксесуарами – близько 2152 грн.

Датчики температури:

- DS18B20 – близько 90 грн;
- DHT11/22 – від 70 до 150 грн залежно від точності та комплектації.

Додаткові компоненти: Кабелі, макетні плати та інші аксесуари – від 50 до 300 грн за одиницю залежно від типу.

Вартість такої системи в базовій конфігурації становить близько 500–700 грн, а зі стартовими наборами – від 2000 грн.

Аналогічні варіанти подібних пристроїв є але в них мінуси — ті які можуть дешевше за збірку на Arduino, вони односторонньої дії та без тонких настройок, і тому програють збірці в тому що вони не модульні, і щоб мати багатий функціонал потрібно докупляти окремі прилади які спеціалізовані і врешті решт сума буде в 1.5 або в 2 рази більша за вартість Arduino.

Мінуси інших варіантів в тому що їх ціна починається від 14000 гривень і більше, та такий варіант не всім вигідний. Система Arduino має значні плюси – вона може сповіщати про критичні зміни температури за допомогою звуку та тексту на панелі, а також використовує інтеграцію чат-ботом в телеграмі для сповіщень у реальному часі [9].

Аналіз стану існуючих засобів контролю кліматичних показників в приміщеннях відображає те, що на сьогодні відсутні дешеві та ефективні рішення для створення комфортних температурних умов в приміщеннях. І основним завданням цього дослідження буде розробка системи моніторингу температури з можливістю звукового та текстового сповіщення, яка базується на легкодоступній платформі Arduino та може бути адаптована до широкого спектру застосувань [10].

1.4. Вимоги до програмного модуля моніторингу параметрів системи

На основі можливих областей застосування системи за допомогою вимірювання параметрів середовища можна визначити багато вимог до розробленої системи.

Мобільність – система вимірювань має бути мобільною, що полегшить роботу користувачів і забезпечить більший охоплення. Це все може бути забезпечено за допомогою бездротової передачі даних від вузла Системи до комп'ютера і порівняно невеликими габаритними розмірами.

Масштабованість – дозволяє збільшувати кількість функцій, при цьому продуктивність її не погіршується. Повинна бути реалізована можливість інтегрування розробленої системи до інших. Для забезпечення масштабованості системи вимірювань повинна бути реалізована можливість додавання і виключення датчиків з її складу.

Модульність – повинна бути можливість використання розроблюваної системи індивідуально або в якості модуля в складі інших систем. Таким чином, необхідно забезпечити можливість інтегрування системи, що розробляється в інші системи.

Швидкодійність – необхідно забезпечити менші часові витрати на функціонування системи, а також можливість одночасного вимірювання декількох різнорідних фізичних величин. Це може бути досягнуто наявністю в складі системи набору датчиків, що працюють паралельно [11].

Формування звітності повинно відбуватися автоматично. Протокол вимірювань повинен містити всю необхідну інформацію в зручному для користувача вигляді, наприклад, у формі таблиць і графіків, та мати можливість відображатися на екрані смартфона або ПК(персональних комп'ютер).

На підставі вимог, що пред'являються до системи при вимірюванні параметрів мікроклімату, був сформований список фізичних величин, які будуть вимірюватися розробленою системою.

1.5. Висновок

Аналіз існуючих стандартів і рішень моніторингу температури свідчить, що дотримання нормативів мікроклімату є критично важливим для забезпечення безпеки, комфорту та якості продукції в різних секторах. Сучасні системи контролю на основі мікроконтролерів, таких як платформи Arduino, демонструють високу ефективність і можуть бути інтегровані в різноманітні сфери завдяки своїй мобільності, масштабованості та низькому енергоспоживанню.

Для забезпечення можливості використання системи контролю температури в різних типах приміщень формуються вимоги до системи. Вони мають включати в себе: модульність, швидкодіяльність, мобільність та зручний формат отримання звіту про стан температури.

Аналіз стану існуючих засобів контролю кліматичних показників в приміщеннях дає нам зрозуміти, що на сьогодні відсутні дешеві та ефективні рішення для задачі дистанційного моніторингу автоматизованого контролю температури приміщення. Платформа Arduino ж надає можливість значно зробити доступнішим систему контролю температури.

2 ДОЦІЛЬНІСТЬ ВИКОРИСТАННЯ ПЛАТФОРМИ ARDUINO ДЛЯ СИСТЕМ МОНІТОРИНГУ ТЕМПЕРАТУРИ ПРИМІЩЕННЯ.

2.1 Особливості апаратної частини та можливості платформи Arduino

Проекти на Arduino є економічно вигідними для задач, які не потребують складних обчислень або обробки великих обсягів даних. Arduino дозволяє створювати ефективні, дешеві і малоспоживаючі системи для автоматизації, моніторингу, керування та прототипування. У випадках, коли немає потреби в серйозних обчислювальних потужностях, вибір Arduino допомагає значно зекономити ресурси і кошти.

В основному це пов'язано саме з особливостями апаратної частини платформи. Апаратною основою платформи є плата із розміщеним на ній мікроконтролером і розведеними (тобто розпаяними) за фіксованою електричною схемою контактами. Програмна частина представлена середовищем розробки Arduino IDE [18].

Оригінальні плати Arduino мають чотири типи, що залежать від габаритів та кількості контактів, доступних для підключення (входи та виходи). Найпоширенішою у використанні є плата Arduino Uno, що містить 20 контактів, при стандартному розмірі. (рис. 2.1).

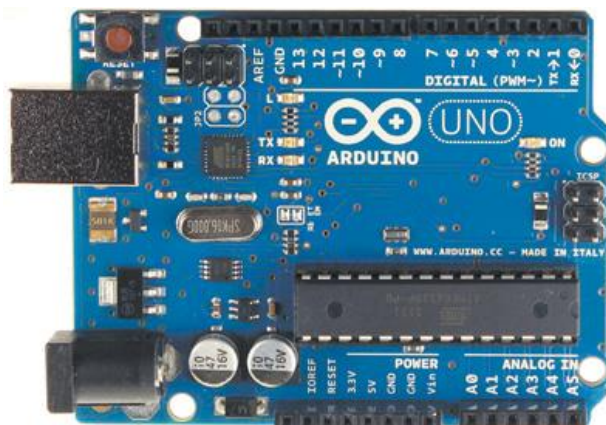


Рисунок 2.1 – плата Arduino Uno

Arduino Mega: збільшений розмір, 70 контактів (рис.2.2)

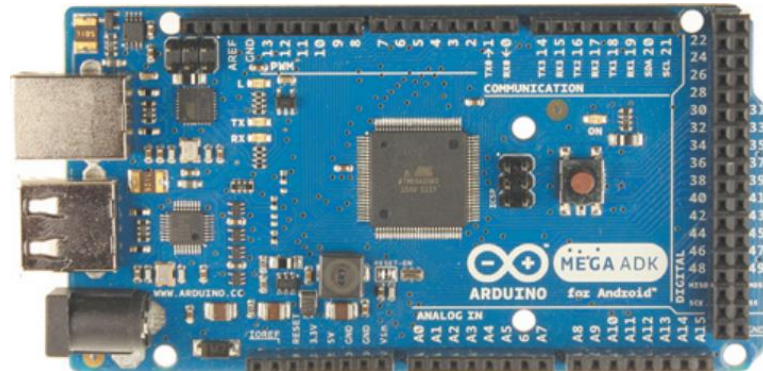


Рисунок 2.2 – Arduino Mega

Arduino Nano: зменшений розмір, 22 контакти (рис. 2. 3)

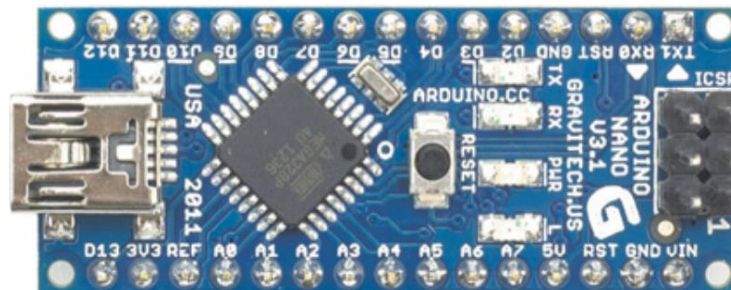


Рисунок 2.3 – Arduino Nano

Arduino Micro: мініатюрна версія, 20 контактів (рис. 2.4), відсутній DATA-інтерфейс USB (потрібний зовнішній програматор, USB забезпечує лише живлення плати).

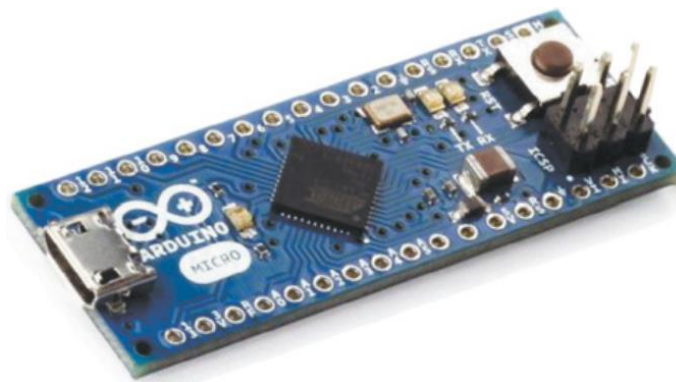


Рисунок 2.4 – Arduino Micro

Arduino Uno належить до робототехнічних платформ відкритого типу. Вона керується мікроконтролером ATmega328, що має характеристики: 2 Кб оперативної пам'яті, 1 Кб пам'яті на енергонезалежному носії EEPROM, 32 Кб пам'яті для завантаження програм, з яких 0,5 Кб зайнято самим завантажувачем. Мікроконтролер має 2 Кб пам'яті SRAM та 1 Кб EEPROM, з якої зчитують або записують інформацію за допомогою бібліотеки EEPROM. Вхідна напруга від джерела живлення становить 7-12 В. Підключені датчики і модулі управляються струмом 40 мА для основної робочої напруги 5 і 50 мА для додаткової напруги 3,3 В. Тактова частота мікропроцесора дорівнює 16 МГц.

Після вимкнення живлення, EEPROM дозволяє зберігання невеликих обсягів даних, таких як:

- Налаштування пристрою (конфігурація);
- Результати вимірювань або останній стан системи.

Через обсяг пам'яті в 1 Кб EEPROM має відносно кількість циклів запису/стирання, зазвичай до 100 000.

Arduino Uno – еталонна модель платформи Arduino і є останньою в серії USB-плат. Вона має 14 цифрових входів/виходів з них 6 можуть використовуватися як ШІМ-виходів, оскільки вони використовують широтно-імпульсну модуляцію (PWM) для імітації аналогового сигналу.

Оскільки Arduino – це платформа відкритого типу для створення інтерактивних електронних пристроїв, яка поєднує апаратну частину з програмним

забезпеченням. Її популярність зумовлена простотою використання, універсальністю та доступністю компонентів, що дозволяють проектам на Arduino бути економічно вигідними. Апаратні компоненти та можливості Arduino Uno, які дозволяють ефективно реалізовувати задачі моніторингу, автоматизації та IoT.

Arduino Uno забезпечує різноманітні комунікаційні інтерфейси та функції, що робить її універсальною для створення інтерактивних проектів. Серійний порт UART використовується для обміну даними між платою і комп'ютером або іншими пристроями, забезпечуючи простоту передачі інформації. Інтерфейси I2C та SPI дозволяють підключати периферійні модулі, такі як дисплеї, пам'ять, сенсори або інші мікроконтролери, що робить можливим створення складних систем.

Arduino підтримує два варіанти живлення: через USB-порт, що на платі позначається 5 В, для зручного підключення до комп'ютера або через зовнішнє джерело на 7–12 В для автономної роботи. Інтегрований стабілізатор напруги забезпечує захист від коливань напруги, що гарантує стабільну роботу плати навіть у несприятливих умовах. Останній фактор особливо важливий в нинішній напруженій ситуації в енергетиці України, та може бути вкрай актуальним для використання автоматизованих проектів на Arduino у складських чи інших виробничих приміщеннях.

Серед додаткових функцій варто виділити вбудований світлодіод на піні 13, що використовується для тестування або індикації. Кнопка Reset дозволяє перезапустити програму без перепідключення живлення. Для програмування через USB передбачено перетворювач USB-to-Serial, що забезпечує зручний обмін даними між платою і комп'ютером. Максимальна довжина та ширина друкованої плати Uno складає 6.9 см і 5.4 см, з урахуванням роз'єму USB та роз'єму живлення, які виступають за межі плати. Чотири отвори кріплення дозволяють прикріплювати плату до поверхні або корпусу. При цьому особливу увагу фізичних параметрів плати заслуговує відстань між цифровими виводами 7 і 8, що не є традиційною 2.54 мм, а становить 4 мм.

Ці особливості роблять Arduino Uno універсальною та доступною і простою у використанні платформою для навчання, автоматизації та інтеграції в проекти

IoT, надаючи широкі можливості для створення інтерактивних проектів. Вона підтримує інтеграцію з численними периферійними пристроями, включаючи температурні датчики (наприклад DS18B20), датчики вологості (DHT22), модулі зв'язку, такі як Wi-Fi (ESP8266 ver ESP-07), Bluetooth або GSM, і дисплеї, зокрема LCD та OLED, для виведення інформації. Також важливим електронним компонентом для нашого проекту моніторингу температури є п'єзoeлемент, що підключається до плати. Він працює за принципом п'єзoeфекту, при якому механічна деформація матеріалу генерує електричний заряд або навпаки.

Wi-Fi модуль ESP8266 буде використовуватися для передачі даних отриманих мікроконтролером від датчика температури. Цей модуль є периферійним пристроєм, що буде підключатися до плати. Вигляд ESP8266 можемо бачити на рисунку 2.5. Для забезпечення коректної роботи модуль має бути підключений до спеціальних портів UART, а саме TX та RX (D0, D1), які відповідають за серійну передачу даних. Прийом та віддача сигналів Wi-Fi модулем ESP8266 можемо бачити на рисунку 2.6.



Рисунок 2.5 – Wi-Fi модуль ESP8266 версії ESP-07

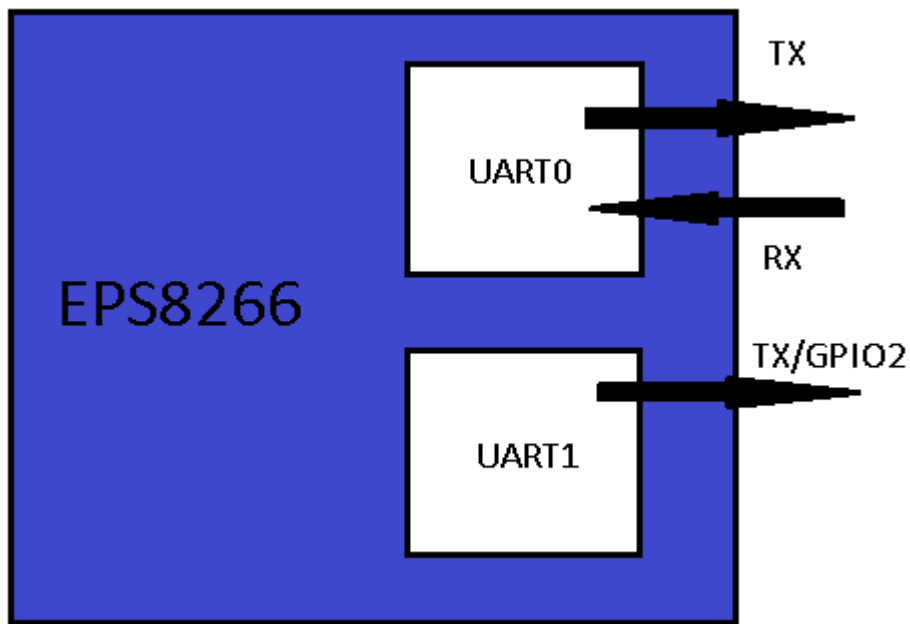


Рисунок 2.6 – Прийом та віддача сигналів Wi-Fi модулем ESP8266

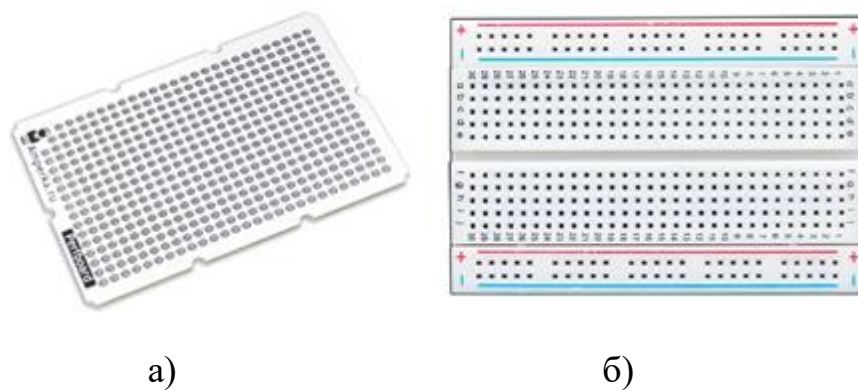
Під час передачі UART перетворює паралельний сигнал, від мікроконтролера, у послідовний для надсилання. Під час прийому UART виконує зворотну операцію, та зазвичай використовується для зв'язку між мікроконтролером і самим модулем, або для програмування і налаштування модуля. Крім основного послідовного порту, ESP8266 має другий UART, який переважно використовується для виведення діагностичної та налагоджувальної інформації. Для правильного функціонування UART необхідно встановити однакову швидкість передачі даних на обох пристроях. Під час завантаження ESP8266 автоматично намагається визначити швидкість підключеного пристрою і синхронізуватися з нею. Бездротова передача даних з мікроконтролера на комп'ютер відбувається у вигляді двійкового коду, який і передає інформацію з Arduino Uno у вигляді послідовності 0 та 1. Двійковий формат мінімізує обсяг передаваних даних, та має обмежену кількість комбінацій, що мають свої конкретні значення.

Ще одним цікавим елементом є пьезоелемент. Цей пристрій можна використати для звукового сигналу, про порушення нормального температурного діапазона.

Завдяки подачі широтно-імпульсної модуляції (ШІМ) на пьезоелемент можна здобути генерацію звуку. Частота цих імпульсів визначає тон звуку. Коли пьезоелемент використовується для створення звуку в Arduino-проектах, його підключають до одного з цифрових виходів плати. Arduino подає змінні імпульси, що створюють коливання пьезокерамічного шару всередині елемента, внаслідок чого генерується звук.

Arduino Uno активно використовується в IoT-проектах завдяки можливості підключення до хмарних сервісів, як-от ThingSpeak або Telegram, який більш зручний та зрозумілий для широкого кола можливих користувачів, для віддаленого моніторингу та управління системами. Простота платформи робить її популярною у навчальних закладах, де вона використовується для викладання основ програмування, електроніки та робототехніки. А широтно-імпульсна модуляція (ШІМ) дозволяє створювати аналогоподібні сигнали, що застосовуються для регулювання яскравості світлодіодів чи швидкості обертання двигунів. Використовуючи Arduino IDE, розробники можуть швидко писати, компілювати та завантажувати програми, спрощуючи процес розробки.

Для максимальної сумісності плати, не тільки з компонентами платформи Arduino, а й з іншими пристроями використовують макетні плати. Приклад макетної плати можемо побачити на рисунку 2.7. Вони поділяються на макетну плату для прототипування за допомогою паяння та для прототипування без паяння



а – прототип макетної плати з паянням; б - без паяння

Рисунок 2.7 – Макетні плати

Платформа також підтримує автоматизацію, дозволяючи створювати системи для контролю освітлення, моніторингу клімату та інших автоматизованих завдань, що робить її ідеальним вибором для широкого спектру задач, які можна втілювати і в умовах невеликого бюджету підприємства чи малого бізнесу, наближуючи їх до створення системи «розумного будинку».

2.2 Види мікроконтролерів та їх можливості для створення систем збору даних.

Плата мікроконтролера – це основа робототехнічної платформи. Мікроконтролери є основою сучасних електронних пристроїв, зокрема систем збору даних. Їх використовують для зчитування, обробки та передачі інформації, зібраної з різних сенсорів, у реальному часі.

Мікроконтролери застосовуються усюди для вирішення абсолютно несхожих завдань: від автоматизації рутинних процесів у сільському господарстві, обслуговуванні та експлуатації будівель, електрифікації до складних автономних

систем життєзабезпечення бункерів. Скрізь, де автоматизація процесу вимагає великих обчислювальних потужностей, застосовують комп'ютери з урахуванням мікроконтролерів, завдяки їх невеликому енергоспоживанню, компактним розмірам та простоті підключення датчиків для взаємодії з навколишнім середовищем та обробки показань.

Коли необхідно спроектувати систему, що вирішує багатокомпонентну, але все ж таки єдину задачу, мікроконтролер стає ефективним і одночасно недорогим рішенням, яке може значно зменшити витрати при автоматизації виробничих процесів, що в свою чергу дозволить зробити послугу чи товар більш доступним за ціною для широкого кола населення. Потрібно зазначити можливу зацікавленість в такому економічному рішенні малий бізнес, що особливо вразливий в актуальних соціально-економічних умовах України, та має гостру необхідність в налагоджені оптимізації своїх виробничих процесів. Оскільки малий бізнес сприяє швидкій структурній перебудові економіки, насичує ринок товарами та послугами, максимізує зайнятість всього працездатного населення, але, зазвичай, потребує втілювати в життя мінімальну кількість задач, що є взаємопов'язаними, то використання в автоматизації шаблонних та елементарних задач електронно-обчислювальних пристроїв з вагомими обчислювальними потужностями – позбавлено сенсу.

Платформа Arduino має дев'ять плат, та лише шість різних мікроконтролерів. Кожна модель має свої особливості, які роблять її придатною для певних завдань. Нижче наведено види мікроконтролерів Arduino, їх можливості та порівняння.

Як бачимо з таблиці 2.1 одна з найпоширеніших у використанні моделей є Arduino Uno, яка базується на мікроконтролері ATmega328P, оснащена 14 цифровими пінами вводу/виводу, 6 аналоговими входами, а також має 32 КБ пам'яті Flash, 2 КБ SRAM і 1 КБ EEPROM. Простота використання та доступність роблять Arduino Uno ідеальною проектів збору даних із температурних датчиків, сенсорів вологості або освітленості або моніторингу стану обладнання. Проте через низьку продуктивність та обмежений обсяг пам'яті Arduino Uno менш придатна для більш складних систем збору даних. Це пов'язано саме із мікроконтролером

ATmega328 що є однокристальним мікроконтролером, створеним компанією Atmel в семействі megaAVR, та має модифіковане 8-битне RISC-ядро процесора Гарвардской архітектури.

Таблиця 2.1– Порівняння видів мікроконтролерів Arduino та їх можливості

Плата Arduino	Мікроконтролер	Робоча напруга [В]	Цифрові входи/виходи	Виходи із ШІМ	Аналогові входи/виходи	Максимальний струм з піна введення-виводу [мА]	Flash-пам'ять [КБ]	Оперативна пам'ять [КБ]	EEPROM- пам'ять [КБ]	Тактова частота [МГц]	Габарити [мм]	USB-роз'єм
Uno	Atmega328	5	14	6	6	40	32	2	1	16	69×53	USB A-B
Leonardo	ATmega32u4	5	20	7	12	40	32	2.5	1	16	75×53	micro-USB
Nano	ATmega328	5	14	6	8	40	32	2	1	16	18×45	mini-USB
Mega	ATmega2560	5	54	14	16	40	256	8	4	16	102×53	USB A-B
Due	Atmel SAM3X8E ARM Cortex-M3 CPU	3.3	54	12	12	800	512	96	×	84	102×53	micro-USB
Mini	ATmega328	5	14	6	6	40	32	2	1	16	30×18	USB-Serial
Micro	ATmega32u4	5	20	7	12	40	32	2.5	1	16	48×18	micro-USB
M0	Atmel SAMD21	3.3	20	12	6	7	256	32	×	48	69×53	micro-USB
LilyPad	ATmega328p	2.7 - 5.5	20	6	6	40	32	2	1	8	∅ 50	USB-Serial

А ось у Arduino Mega, яка базується на ATmega2560, більш задовольняє своїми можливостями потребу в багатоканальних системах моніторингу, де потрібно одночасно обробляти дані з багатьох джерел, а тому підходить для проектів із великою кількістю датчиків і складними програмними алгоритмами. Ця плата має 54 цифрових пінів, з яких 15 підтримують PWM, і 16 аналогових входів, а також має значно більший обсяг Flash-пам'яті - 256 КБ, 8 КБ SRAM та 4 КБ EEPROM-пам'яті.

Для компактних проектів, наприклад, переносних систем збору даних, доцільно використовувати Arduino Nano. Ця модель має схожі характеристики з Arduino Uno, але її компактний розмір дозволяє інтегрувати її у невеликі пристрої. Nano базується на ATmega328P або ATmega168 і має 14 цифрових пінів, 8 аналогових входів та до 32 КБ Flash-пам'яті [16].

Модель Arduino Leonardo, побудована на базі ATmega32U4, вирізняється тим, що має вбудований USB-контролер. Це дозволяє їй імітувати периферійні пристрої, такі як миша, клавіатура чи джойстик. Вона має 20 цифрових пінів, 12 аналогових входів, а також 32 КБ Flash-пам'яті. Leonardo підходить для проектів,

які вимагають тісної інтеграції з комп'ютером, наприклад, для збору та передачі даних на ПК.

Для складних проектів зі збору даних, які потребують обчислювальної потужності, доцільно використовувати Arduino Due. Ця модель базується на 32-бітному процесорі ARM Cortex-M3 (SAM3X8E) і має 54 цифрових піна, 12 аналогових входів, 512 КБ Flash-пам'яті та 96 КБ SRAM. Особливістю Arduino Due є 12-бітний АЦП, що забезпечує високу точність при зборі аналогових даних. Вона використовується у складних системах моніторингу або в проектах, де потрібна висока швидкість обробки даних.

Серія Arduino MKR розроблена спеціально для проектів Інтернету речей (IoT). Вона базується на ARM Cortex-M0+ (SAMD21) і має вбудовані модулі зв'язку, такі як Wi-Fi, GSM або LoRa, що спрощує передачу даних на сервери або у хмару. Компактний розмір та висока функціональність роблять ці мікроконтролери ідеальними для створення систем віддаленого моніторингу та збору даних у реальному часі.

Для промислових та високопродуктивних додатків використовується серія Arduino Portenta, наприклад, Portenta H7. Ця модель побудована на базі ARM Cortex-M7/M4, має значний обсяг пам'яті (8 МБ Flash) і підтримує багатозадачність. Завдяки високій продуктивності вона підходить для обробки великих обсягів даних, включаючи застосування алгоритмів машинного навчання.

Всі зазначені вище мікроконтролери дають змогу дозволяє передавати зібрані дані до центральних систем або хмарних платформ завдяки підтримці протоколів UART, I2C, SPI, Wi-Fi, LoRa. Завдяки підключенню до платформ, таких як ThingSpeak чи Blynk, можливий віддалений моніторинг і аналіз даних у реальному часі, що дозволяє інтеграцію з IoT, та Telegram. Інтегровані математичні функції дозволяють виконувати базову фільтрацію, обчислення середнього значення або компенсацію шуму, здійснюючи таким чином обробку даних. Arduino Uno не має вбудованого модуля Wi-Fi, проте його можна використовувати для передачі даних на суміжні пристрої через додаткові модулі, такі як ESP8266 або ESP32. Ці модулі підключаються до Arduino Uno через

інтерфейси UART (послідовний порт) або SPI та забезпечують бездротову передачу даних через мережу Wi-Fi. В такому випадку алгоритм передачі даних можливий, і буде виглядати як на схемі (рисунок 2.8)

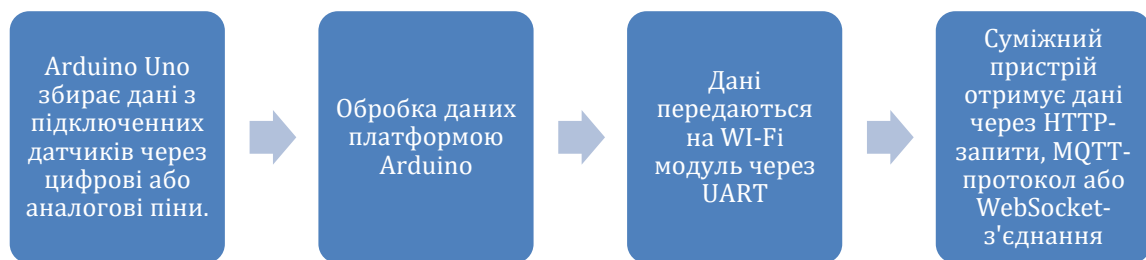


Рисунок 2.8 – Алгоритм передачі даних

Мікроконтролери Arduino забезпечують широкий вибір моделей, які можуть адаптуватися до різного типу завдань систем збору даних. Для економічних та універсальних проектів найкраще підходять Arduino Uno або Nano, тоді як для більш складних систем збору та обробки даних доцільно використовувати Arduino Mega або Due. Моделі серії MKR і Portenta ідеально підходять для IoT-додатків і промислових рішень, але і їх ціна в середньому стартує від ста євро, тоді як Arduino Uno і дешевша і доступніша для покупки на ринку України, тим паче, що і для неї є можливість використання Wi-Fi модуля, що забезпечує можливість передачі даних до суміжних пристроїв. При цьому Arduino Uno та ESP8266 споживають відносно мало енергії, що робить їх придатними для автономних систем та дозволяє ефективно реалізувати передачу даних на суміжні пристрої через бездротові мережі. Ця комбінація підходить для різних завдань: моніторингу, передачі даних у хмару або створення IoT-пристроїв. Окрім того, що до Arduino можна підключати різноманітні датчики, доступними також є і модулі Wi-Fi, Bluetooth, GSM. Для роботи з модулями зазвичай використовуються стандартні бібліотеки, доступні в

Arduino IDE. Особливо нас цікавить бібліотека для роботи з Wi-Fi – ESP8266WiFi. Оскільки для передачі даних Arduino може використовувати послідовний порт UART, I2C або SPI, то, дані з датчиків можуть бути передані через Wi-Fi модуль ESP8266 на сервер або інший суміжний пристрій.

Від вибору мікроконтролера залежить кількість підключених датчиків, обсяг даних для обробки та вимоги до передачі інформації. Оскільки нашим проектом є дистанційний моніторинг автоматизованого контролю температури приміщення, то доцільніше буде вибрати саме універсальну та економічно доступну плату Arduino Uno, з мікроконтролером ATmega328P, оскільки збір температури є відносно простим процесом, а сам дистанційний моніторинг буде відбуватися здебільшого завдяки Wi-Fi модулю.

Arduino Uno має 14 портів вводу/виводу (GPIO). Це цифрові порти, й позначаються на платі D0–D13. А також 6 аналогових портів – A0–A5. Цифрові порти можуть працювати в режимі вводу або виводу. Цифрові порти можуть приймати на себе функції аналогових портів. До них ми відносимо: D3, D5, D6, D9, D10, D11, вони підтримують ШІМ (PWM), що дозволяє генерувати аналогоподібні сигнали. Також маємо три порти для живлення: на компоненти з напругою 5 В, для пристроїв, які працюють на 3,3 В та два порта GND – загальний мінусовий контакт (земля).

Спеціальними портами можна назвати порти TX та RX (D0, D1), що відповідають за серійну передачу даних через UART. Під час передачі UART перетворює паралельний сигнал, наприклад, від мікроконтролера, у послідовний для надсилання. Під час прийому UART виконує зворотну операцію.

SDA та SCL (A4, A5) використовуються для роботи з I2C-пристроями, що дозволяють з'єднувати кілька електронних пристроїв. Ці контакти фізично розташовані на аналогових пінах A4 (SDA) і передає данні, а A5 (SCL) синхронізує передачу, генеруючи тактовий сигнал.

2.3 Принцип роботи Arduino

Актуальність використання Arduino обумовлена зростаючою потребою у доступних та гнучких інструментах для створення електронних рішень. Принцип роботи Arduino може вдовільнити цю потребу в автоматизованій системі, для рішення різного роду побутових задач, зберігаючи такий необхідний сьогодні економ-режим електроенергії та фінансів, оскільки, нагадаємо, компоненти плати Arduino одні з самих доступних на ринку робототехніки. Будь-яка плата Arduino складається з ключових компонентів у вигляді мікроконтролера, живлення, GPIO-пінів, комунікаційних інтерфейсів, які забезпечують зв'язок з периферією або іншими пристроями, та програматора, що дозволяє завантажувати код на мікроконтролер. Спрощений цикл програмування на Arduino можна поглянути на рис. 2.9.

Arduino Uno має USB-порт із чіпом програмування, завдяки якому він під'єднується до комп'ютера. Програмна частина представлена середовищем розробки Arduino IDE, доступ до якої вільний, та включає в себе редактор, препроцесор, монітор послідовного порту і компілятор.

Мова програмування на якій розроблене інтегроване середовище Arduino IDE є стандартним C++ та C, з використанням компілятора AVR-GCC. Середовище розробки доступне для Windows, MacOS та Linux, та призначене для творення та завантаження програм на Arduino-сумісні плати. З'єднання персонального комп'ютера, на який і встановлюється Arduino IDE із мікроконтролером реалізовано через інтерфейс USB. Код мовою C та C++ пишеться в редакторі, в якому є підсвічування команд та перевірка орфографії - спелчекер. У кожен мікроконтролер вшитий завантажувач, який трансформує інструкції, написані людиною, в машинні коди без використання апаратного програматора. Також використовуються різноманітні бібліотеки для роботи з апаратними модулями. Це дозволяє створювати програми без необхідності знати складні низькорівневі деталі

роботи мікроконтролера, та додають функціональність для роботи з датчиками, модулями Wi-Fi, дисплеями, серводвигунами тощо.



Рисунок 2.9 – Спрощений цикл програмування на Arduino

Arduino IDE містить редактор коду, що є основною частиною інтерфейсу, де користувач пише програми. Панель інструментів включає кнопки для швидкого виконання основних дій, таких як компіляція, завантаження коду на плату, створення нового скетчу, відкриття чи збереження файлу. Інформацію про процес компіляції, помилки, а також повідомлення про стан підключення до плати, відображає консоль. Для відлагодження та отримання даних від підключених пристроїв використовується монітор послідовного порту. Все це можна побачити на рисунку 2.10.

The screenshot shows the Arduino IDE window titled "Arduino - 0011 Alpha". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main editor area displays the "Blink" sketch code, which includes a multi-line comment explaining the sketch's purpose and a URL. The code defines a pin, sets it as an output in the setup function, and then alternates the pin state between HIGH and LOW with 1000ms delays in the loop function. The console at the bottom shows the message "Done compiling." and "Binary sketch size: 1098 bytes (of a 14336 byte maximum)". The line number 22 is visible at the bottom left of the console area.

```

/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}

```

Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)

22

Рисунок 2.10 – Інформаційна консоль про стан підключення до плати

Arduino IDE не єдине середовище для програмування доступне для роботи з платами. Програма Processing ґрунтується на Java і має бібліотеку Serial, через яку можна отримувати та надсилати повідомлення, командами: `Serial.write()` та `Serial.print()`. Basic for Arduino (B4R) - середовище розробки, що використовує мову Basic. Це безкоштовна програма, що розглядається як альтернатива Arduino IDE, удосконалена версія Visual Basic. А також деякі навчальні емулятори дозволяють збирати електричні ланцюги та програмувати Arduino, наприклад – Tinkercad.

Програмний код для реалізації тих чи інших апаратних функцій серед Arduino називають скетчем. Зберігаються файлами з розширенням «.ino». Робота програмного коду Arduino побудована на циклічному виконанні інструкцій у поєднанні з багатим вибором моделей і бібліотек, які дозволяють ефективно адаптувати Arduino для збору, обробки та передачі даних.. Основна структура будь-

якої програми для проекту складається з двох основних етапів: «Setup» та «Loop». Setup відповідає за початкову ініціалізацію компонентів, таких як порти введення/виведення або комунікаційні модулі. В свою чергу Loop є основним робочим циклом, де обробляються дані з підключених пристроїв, та виконує логіку керування, забезпечуючи передачу або збереження інформації.

Структура нашої системи контролю та оповіщення температури буде складатися з трьох рівнів: нижнього, середнього і верхнього. Нижній рівень, представлений датчиком температури.

Середній рівень системи, в нашому випадку, представлений у вигляді контролера Arduino UNO, до якого приєднуються датчик і виконавчі елементи, такі як: RGB-світлодіод та 7-сегментний екран.

Верхній рівень, це SCADA, в разі промислових систем, або ж Web-додаток Telegram, що буде використовуватись в нашому випадку, через який здійснюється управління системою.

На даний момент вже існують SCADA-системи розумного будинку, проте дані системи реалізовані на промислових контролерах, що тягне за собою високу вартість кінцевого продукту. Наша ж система реалізована на базі контролера Arduino UNO, вартість якого доступна для середнього класу споживача, в чому і полягає актуальність даної роботи. Зазначимо також, що після підключення датчиків потрібно встановити відповідні для них бібліотеки програмного коду, щоб написати для них програму. Систему можна використовувати управління для мікроклімату в квартирах або будинку, ще можна збільшити площу, підключаючи контролери та інше обладнання для «розумного-будинку». Так як Arduino підтримує обробку, як цифрових, так і аналогових сигналів, цифрові піни можуть бути використані для читання логічних станів HIGH або LOW, керування пристроями, такими як реле, світлодіоди. Аналогові піни забезпечують перетворення аналогових сигналів у цифрові значення (АЦП) з роздільною здатністю 10 біт (0–1023).

2.4 Принципи роботи сенсорів температури

Моніторинг температури навколишнього середовища потрібен для досягнення оптимальних величин температури необхідних для створення ефективних умов праці, або зберігання продукції. При відхиленні встановленої температурної норми на підприємстві, вдома чи в інших приміщеннях застосовуються вже механізми оповіщення та заходи направлені на повернення необхідного температурного режиму.

Для моніторингу виміру теплової енергії та перетворенні її на електричний сигнал, що буде відслідковуватись для моніторингу температури, використовуються сенсори температури. Залежно від методу роботи, сенсори поділяються на контактні та безконтактні, а також на аналогові та цифрові.

Найпоширенішими типами сенсорів можна назвати:

- терморезистори;
- термопари;
- напівпровідникові сенсори;
- інфрачервоні сенсори;
- резистивні сенсори температури (RTD).

Терморезистори працюють за принципом зміни опору залежно від температури та поділяються на NTC (Negative Temperature Coefficient), при яких опір зменшується зі збільшенням температури, та відповідно PTC (Positive Temperature Coefficient), де опір збільшується зі зростанням температури. Хоч терморезистори популярні в системах моніторингу температури в побутових приладах та промисловому обладнанні, але потребують постійного калібрування, оскільки залежність опору від температури нелінійна.

Термопари складаються з двох провідників різних металів, які з'єднані у певних точках. Різниця температур між кінцями провідників створює термоелектричний ефект, що генерує напругу. Термопари мають широкий діапазон температур та використовуються при потребі в високій точності вимірювання,

маючи при цьому мають залежність вихідного сигналу від довжини проводів. Напівпровідникові сенсори використовують залежність струму або напруги в напівпровідникових матеріалах від температури. Такі сенсори поділяються на цифрові та аналогові. Цифрові сенсори мають, порівняно з аналоговими, значно простішу інтеграцію з мікроконтролерами та генерують точний цифровий сигнал, що безпосередньо передається мікроконтролеру. За таким принципом працює DS18B20 – датчик від компанії Dallas, що має діапазон вимірювальних температур від -55 до +125 °C.

Аналогові сенсори, як TMP36, може видавати напругу, що пропорційна температурі оточуючого середовища. Інфрочервоні сенсори (ІЧ-сенсори) працюють за принципом вимірювання кількості інфрачервоного випромінювання, яке випромінює об'єкт, оскільки кількість випромінювання прямо пропорційна температурі об'єкта. Хоч ми і згадуємо про них, та в системах моніторингу температури в житлових та виробничих приміщеннях вони не використовуються, оскільки їх основна задача - моніторинг температури рухомих або небезпечних об'єктів, а, також, вони дуже чуттєві до зовнішнього середовища. Резистивні сенсори температури мають високу точність і стабільність. Але є в найдорожчими, в середньому, у порівнянні з іншими сенсорами. Причина – використання дорогих металів, зазвичай платини, оскільки вона має лінійну залежність опору від температури. Це спрощує калібрування та розрахунки. Опір платинового провідника при температурі 20°C становить 20 Ом, а при температурі 500°C – 59 Ом. Таким чином, резистивні сенсори доцільно використовувати в лабораторних умовах.

Зважаючи на потребу сумісності датчика із нашим проектом дистанційного моніторингу автоматизованого контролю температури приміщення з платою Arduino Uno доцільніше розглянути напівпровідникові сенсори. Докладніше розглянемо їх.

Як вище було зазначено, напівпровідникові сенсори поділяються на цифрові та аналогові, але серед сумісних з платою Arduino є також безконтактний інфрачервоний датчик MLX90614, який можемо побачити на рисунку 2.11.

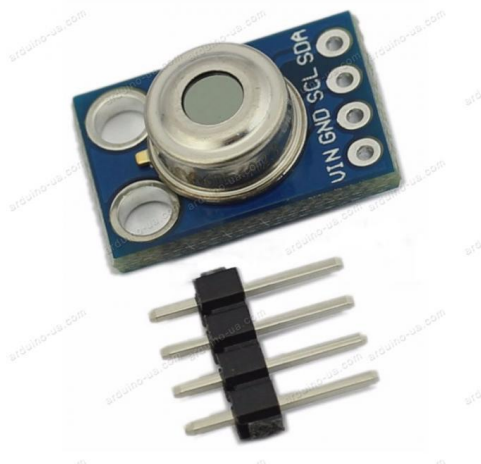


Рисунок 2.11 – Безконтактний інфрачервоний датчик MLX90614

Цей датчик цікавий обробкою температурних даних в два способи: як через цифровий інтерфейс SMBus (аналогічний I2C) так і через ШІМ-вихід із частотою ШІМу 10Гц або 1000Гц. Цей датчик має широкі можливості для програмування, калібрування та конфігурування. Модуль має зручні отвори для кріплення, внутрішній стабілізатор, конденсатори фільтру живлення та підтягуючі резистори на цифровій шині. Цей датчик має власну бібліотеку - `Adafruit_MLX90614` і високу точність. В середньому ціна на нього також висока, серед інших напівпровідникових сенсорів. Власне, для цього проекту використовувати такий датчик не має сенсу, адже потрібні функції доступні і в більш дешевих, та енергоекономічних, а орієнтація проекту на побутових споживачів та складські приміщення не передбачає роботу із настільки широким спектром температур, що охоплює датчик MLX90614. Діапазон температур всіх можливих варіантів датчиків можемо побачити в таблиці 2.11.

Цифровий датчик DHT22 має додаткову властивість – вимірювання вологості. Можемо побачити його на рисунку 2.12. Висока точність, низьке енергоспоживання робить цей датчик привабливим варіантом для проекту дистанційного моніторингу температури, особливо, в житлових або офісних

приміщеннях вимірювання параметру вологості мав би сприяти кращому дотриманню санітарних вимог. Та разом із позитивними, та вельми привабливими сторонами, є і суттєвий недолік. Датчик DHT22 має найповільніший відгук серед представлених типів датчиків – 1 раз на 2 секунди. Для своєї коректної роботи він також потребує підключення окремої бібліотеки DHT.

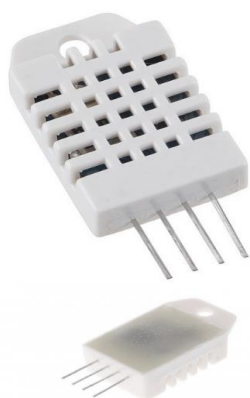
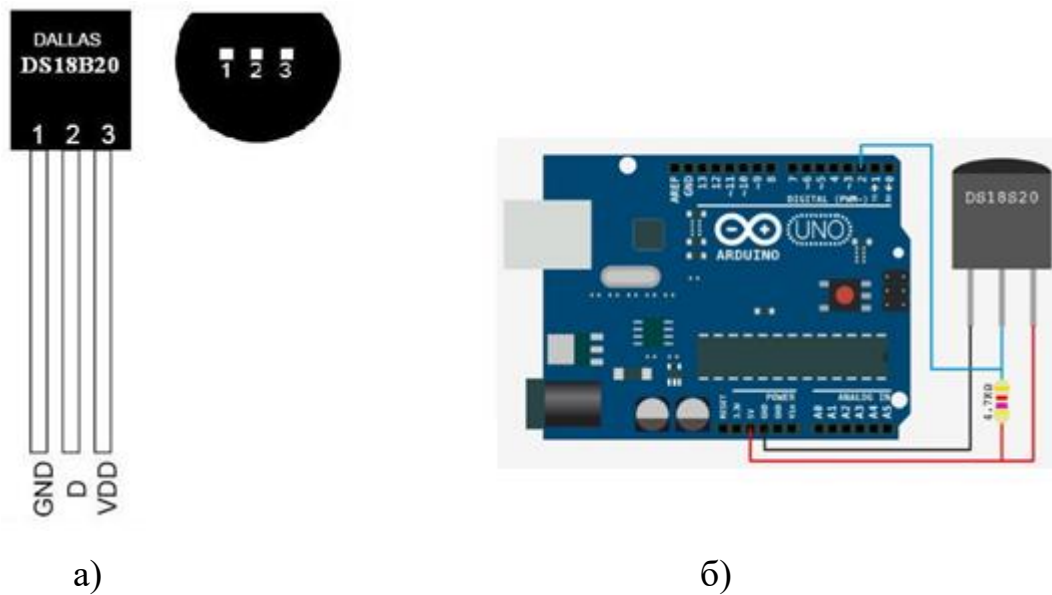


Рисунок 2.12 – Цифровий датчик DHT22

З цифровим інтерфейсом також є DS18B20, представлений на рисунку 2.13 (А). Зчитування інформації з цього датчика здійснюється через спеціальні бібліотеки OneWire, DallasTemperature та команди, які працюють із протоколом. DS18B20 використовує унікальний цифровий інтерфейс, який дозволяє підключити кілька датчиків до одного пину Arduino, а тому і його налаштування більш складні у порівнянні з аналоговими датчиками. Для підключення потрібен резистор 4.7 кОм. Схему підключення можемо побачити на рисунку 2.13 (Б). Окрім цього має можливість паразитарного живлення, що, тим не менш, може призводити до втрати даних, такий самий ризик і через обмеження протоколу. Довжина кабелів між сенсором і мікроконтролером може впливати на якість сигналу.



а - цифровий датчик DS18B20; б - схема підключення до плати

Рисунок 2.13 – Цифровий датчик DS18B20 та схема підключення до плати

Як можемо бачити з таблиці 2.2, аналогові датчики LM35 та TMP36, мають низький рівень споживання енергії, та низьку вартість. Як і TMP36, LM35 видає аналоговий сигнал, але не має початкового зміщення 500 мВ, що робить розрахунок температури простішим. Ці датчики подібні один до одного, та мають ряд відмінностей, найсуттєвіша з яких – в інших схемах підключення, та чутливості до коливань напруги. Основний електричний параметр та виводи можемо детально розглянути на рисунку 2.14, а таблиця 2.2, нам показує що живлення у цього датчика може сягати найвищого, серед наведених датчиків, значення, на відміну від TMP36, що має найнижчі вимоги до живлення.

На рисунку 2.14 також можемо побачити стандартні виводи для передачі даних і підключення до землі: OUT – який видає напругу, пропорційну вимірюваній температурі, та зчитується через аналоговий вхід контролера Arduino за допомогою функції «`analogRead()`», та GND, що в свою чергу відповідає за замкнення електричного кола між джерелом живлення та датчиком.

Таблиця 2.2 Порівняльна таблиця датчиків

Параметр	TMP36	LM35	DS18B20	DHT22	MLX90614
1	2	3	4	5	6
Тип	Аналоговий	Аналоговий	Цифровий	Цифровий (температура + вологість)	Безконтактний інфрачервоний
Діапазон температур	-40°C до +125°C	-55°C до +150°C	-55°C до +125°C	-40°C до +80°C	-70°C до +380°C
Точність	±1°C	±0,5°C при 25°C	±0,5°C	±0,5°C	±0,5°C
Інтерфейс	Аналоговий	Аналоговий	OneWire	Цифровий	I2C
Підключення	1 пін для сигналу	1 пін для сигналу	1 пін для даних	1 пін для даних	2 пини (SDA, SCL)
Живлення	2,7–5,5 В	4–20 В	3–5,5 В	3,3–6 В	3,3–5 В
Використання енергії	Дуже низьке	Дуже низьке	Низьке	Низьке	Відносно високе
Особливості	Простий у використанні	Висока точність	Можна підключати кілька датчиків до одного пину	Вимірює також вологість	Безконтактне вимірювання
Ціна	Низька 79	Низька 77	Низька 26 -55	Середня 210	Висока 375

Лінійне значення вихідної напруги та прецизійне калібрування роблять датчик LM35 вкрай зручним для підключення до вимірювальних ланцюгів, і видає лінійно залежну від температури напругу –10 мВ на кожен градус Цельсія. У

2.5 Висновки до розділу

Моніторинг температури важливий у багатьох сферах: від побутового використання (контроль температури в приміщенні чи холодильнику) до промислових застосувань. Сучасні технології дозволяють створити доступну і зручну систему для дистанційного контролю, а актуальні для України економічно-соціальні умови ставлять додаткову вимогу – економічність в використанні електроенергії, та коштів, затрачених на виготовлення моніторингової системи.

Особливості апаратної частини та можливості платформи Arduino дозволяють економно вирішувати завдання, які не потребують складних обчислень або обробки великих обсягів даних. Arduino Uno активно використовується в IoT-проектах завдяки можливості підключення до хмарних сервісів, а головне дозволяє автоматизувати моніторинг клімату.

Тож, для проекту дистанційний моніторинг автоматизованого контролю температури приміщення буде використовуватися плата Arduino Uno на мікроконтролері ATmega328P. До плати буде підключений датчик температури TMP36, допускається також використання датчика LM35. Виконавчі елементи системи контролю та моніторингу будуть представлені RGB-світлодіодом та 7-сегментний екраном, що будуть візуально подавати інформацію, про відхилення температури від норми. Для надання системі звукового сигналу використаємо пьезоелемент, що буде видавати звук при подачі певних електричних сигналів. Також використаємо Wi-Fi модуль для передачі даних у додаток Telegram.

3 РЕАЛІЗАЦІЯ ДИСТАНЦІЙНОГО МОНІТОРИНГУ ТА ОПОВІЩЕННЯ СИСТЕМИ КОНТРОЛЮ ТЕМПЕРАТУРИ

3.1 Схема апаратного забезпечення

Щоб забезпечити економічно вигідне рішення для автоматизованої системи моніторингу контролю температури, маємо використати доступні можливості платформи Arduino, доцільність використання якої вже зазначили в попередній частині. Щоб розуміти базовий вигляд нашої системи була зроблена візуалізація схеми на платформі Tinkercad, що є одним з емуляторів роботи Arduino.

В своєму початковому варіанті система моніторингу базувалася виключно на цій платформі, та в готовому вигляді мала вигляд як на рис. 3.1. В цьому вигляді модель містить універсальний мікроконтролер Arduino Uno, підключений до нього датчик температур TMP36, базовий п'єзоелемент, RGB-світлодіод, макетну плату, 8-розрядний реєстр здвигу, та три елементи, що складають 7-сегментний екран. Для захисту компонентів від зайвої напруги в модель було додано 25 резисторів. Також можемо бачити, як всі ці елементи підключаються до плати Arduino Uno.

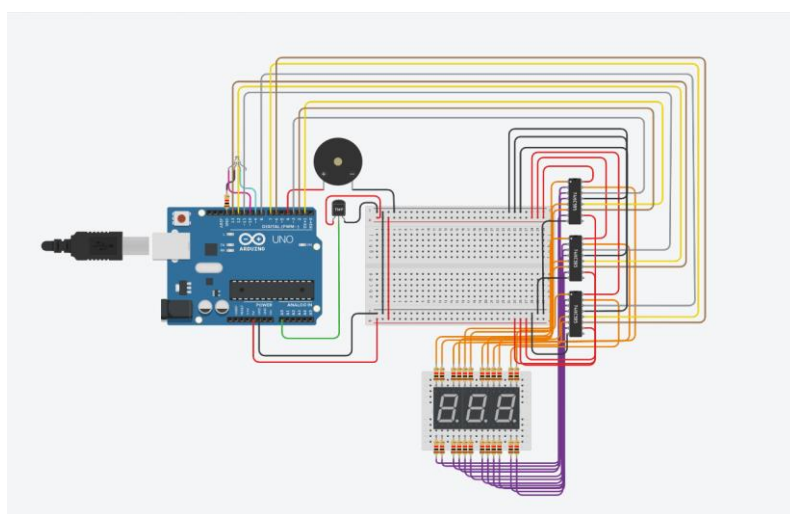


Рисунок 3.1 – Схеми підключень до плати Arduino Uno

Для опису правильного підключення елементів системи моніторингу температури були використані схеми електричного ланцюга, які можемо побачити в додатку Б. Поясненням до переходів на схемі виступає таблиця 3.1, в якій зазначені всі основні використані компоненти, ім'я компонента на схемі та кількість використаних елементів для проекту (додаток А).

Таблиця 3.1. Пояснення переходів на схемі

Ім'я компонента на схемі	Кількість використаних елементів в схемі	Компонент
U1	1	Arduino Uno R3
U2	1	Датчик температури [TMP36]
D1	1	RGB-світлодіод
R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13 R14 R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R25	25	1 k Ω Резистор
PIEZO1	1	П'єзоелемент
U3, U4, U5	3	8-розрядний реєстр здвигу
Digit1, Digit2, Digit3	3	7-сегментний екран

Щоб забезпечити передачу даних з мікроконтролера потрібно використати Wi-Fi модуль ESP8266. Модуль підключається до мікроконтролера через UART, про який говорили у другій частині, з використанням портів TX на передачу і RX на прийом. Але, так як порт TX в нас вже зайнятий, то згадуємо, що цифрові порти здатні приймати на себе функції аналогових портів. Вибираючи з вільних та підходящих портів, на передачу даних підключаємо Wi-Fi модуль в порт D5.

Керування апаратною частиною проекту буде здійснюватися за допомогою коду.

3.2 Розробка коду для збору даних з датчиків і їх обробки

Для реалізації системи дистанційного моніторингу автоматизованого контролю температури приміщення має бути написаний код, що буде реалізовувати систему збору, обробки та передачі даних, що використовує платформу Arduino з інтеграцією Wi-Fi модуля та Telegram API. Як зазначалося в другому розділі мова програмування на якій здійснюється реалізація проекту є C++.

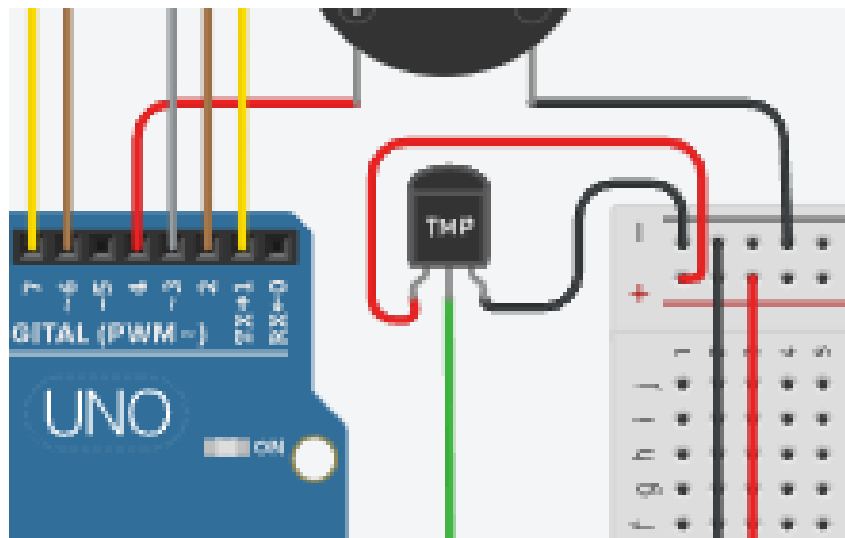


Рисунок 3.2 – Датчик температури

Починаємо код з створення простору імен для всіх периферійних пристроїв, із зверненням до портів їх підключення:

```
#include <WiFi.h>
```

```
#include <UniversalTelegramBot.h>
```

```
#define red 10
```

```
#define blue 9
#define zoomer 4
#define dataED 8
#define SwitchED 7
#define syncED 6
#define dataM 3
#define SwitchM 1
#define syncM 2
#define data 11
#define Switch 12
#define sync 13
#define temp A0
#define RX 0
#define TX 5
```

Підключаємо до мережі Wi-Fi, використовуючи його данні для входу:

```
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";
```

Підключення до telegram-боту використовуючи його токен та ID користувача:

```
const char* botToken = "YOUR_BOT_TOKEN";
const char* chatID = "YOUR_CHAT_ID";
```

Створення об'єкту підключення:

```
WiFiClient client;
UniversalTelegramBot bot(botToken, client);
```

Основна структура складається з функції «void setup()». Це одна з двох основних функцій для роботи Arduino що містить у собі команди які визначають режими роботи для усіх підключених компонентів:

```
void setup()
{
```

До вкладених компонентів належить визначення режиму роботи температурного датчика (рис. 3.2), та передача RGB-світлодіоду відповідну команду увімкнути\вимкнути:

```
pinMode (temp, INPUT);
Serial.begin(9600);
pinMode(red, OUTPUT);
pinMode(blue, OUTPUT);

pinMode(dataED, OUTPUT);
pinMode(SwitchED, OUTPUT);
pinMode(syncED, OUTPUT);
```

Вивід температури здійснюється також і на три 7-сегментних екрана. Зазначаємо, що спочатку 7-сегментний екран вимкнений, щоб уникнути його блимання:

```
digitalWrite(SwitchED, LOW);
```

Створення масиву елементу 7-сегментного екрану

```
shiftOut(dataED,syncED, LSBFIRST, 0b11111100);
```

І задаємо початкове вмикання 7-сегментного екрану:

```
digitalWrite(SwitchED, HIGH);
```

```
pinMode(dataM, OUTPUT);
```

```
pinMode(SwitchM, OUTPUT);
```

```
pinMode(syncM, OUTPUT);
```

Почергове вимкнення елементів 7-сегментного екрану використовується для попередження ефекту блимання:

```
digitalWrite(SwitchM, LOW);
```

```
digitalWrite(SwitchM, HIGH);
```

Знову виставлення режиму роботи 7-сегментного екрану:

```
pinMode(data, OUTPUT);
```

```
pinMode(Switch, OUTPUT);
```

```
pinMode(sync, OUTPUT);
```

```
digitalWrite(Switch, LOW);
```

```
digitalWrite(Switch, HIGH);
```

```
}
```

Прописуємо функцію циклічного виконання ряду вкладених команд:

```
void loop()
```

```
{
```

Створюємо числову змінну, для отримання даних з аналогового піна A0

```
int val = analogRead(A0);
```

Прирівнюємо проміжки значень. 20 = -40, 358 = 125:

```
int tmp = map(val, 20, 358, -40, 125);
```

Використовуємо логічну конструкцію, яка перевіряє значення з датчика на відповідність діапазону нормальних температур:

```
if (tmp >= 5 && tmp <= 70) // Нормальні значення
  analogWrite(zoomer, 0);
  analogWrite(blue, 255);
  analogWrite(red, 0);
  sendTelegramMessage("Все в порядку"); //повідомлення в Telegram, якщо
температура в нормальних значеннях
}
else // Порушення діапазону нормальних температур
{
  analogWrite(zoomer, 255);
  analogWrite(blue, 0);
  delay(200);
  analogWrite(red, 255);
  delay(200);
  analogWrite(red,0);
  sendTelegramMessage("Температура критична, прийміть відповідні заходи "); //
повідомлення в Telegram, якщо значення температури знаходиться за межами
встановленого діапазону
}
```

Змінні, що будуть тримати в собі значення одиниць та десятків значення:

```
int str = tmp%10;
```

```
int strd = tmp/10;
```

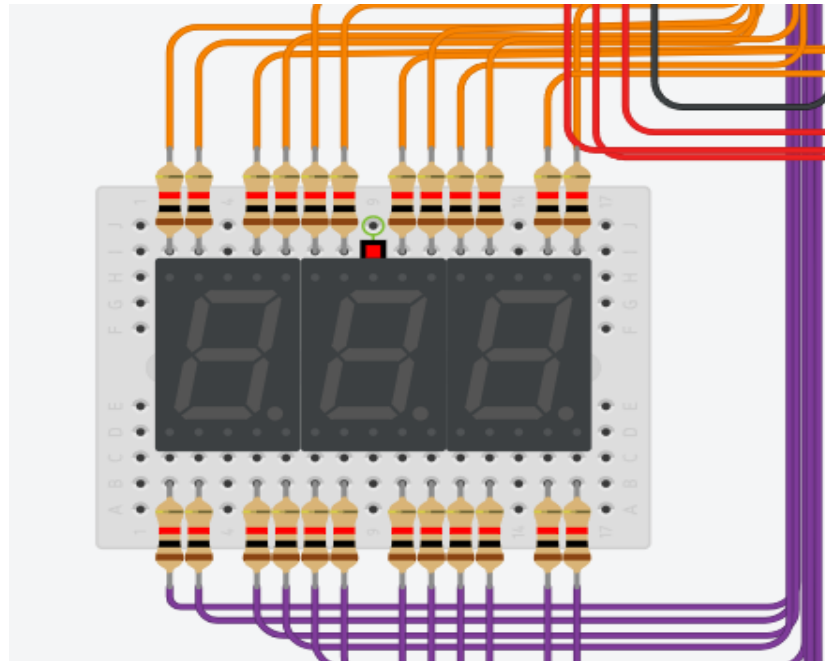


Рисунок 3.3 – 7-сегментний екран

Використовуємо логічну конструкцію для виводу даних про температуру на 7-сегментний екран (рис. 3.3):

```
if (str < 0) { // Виводимо на 1 екран мінус
    digitalWrite(SwitchM, LOW);
    shiftOut(dataM, syncM, LSBFIRST, 0b00000010);
    digitalWrite(SwitchM, HIGH);
}
```

Щоб при виведенні значення з мінусом відображалися дані коректно:

```
str = str*-1;
```

```
strd = strd*-1;
```

```
}
```

Продовження логічної конструкції у разі якщо умова не була виконана:

```
else {
  shiftOut(dataM, syncM, LSBFIRST, 0b11111100); //вивести 0
}
```

Задаємо очікування в 300 мс.:

```
delay(300);
```

Вивід на монітор послідовного інтерфейсу реального значення датчика, для порівняння даних отриманих з 7-сегментного екрану:

```
Serial.println(tmp);
```

Прописуємо комбінації включення елементів 7-сегментного екрану для виведення усіх можливих варіантів арабських цифр одиничних значень:

```
switch(str) {
  case 0: {
    digitalWrite(SwitchED, LOW);
    shiftOut(dataED, syncED, LSBFIRST, 0b11111100);
    digitalWrite(SwitchED, HIGH);
    break;
  }
  case 1: {
    digitalWrite(SwitchED, LOW);
    shiftOut(dataED, syncED, LSBFIRST, 0b01100000);
    digitalWrite(SwitchED, HIGH);
```

```
break;
}
case 2: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b11011010);
digitalWrite(SwitchED, HIGH);
break;
}
case 3: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b11110010);
digitalWrite(SwitchED, HIGH);
break;
}
case 4: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b01100110);
digitalWrite(SwitchED, HIGH);
break;
}
case 5: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b10110110);
digitalWrite(SwitchED, HIGH);
break;
}
case 6: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b10111110);
digitalWrite(SwitchED, HIGH);
```

```

break;
}
case 7: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b11100000);
digitalWrite(SwitchED, HIGH);
break;
}
case 8: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b11111110);
digitalWrite(SwitchED, HIGH);
break;
}
case 9: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b11110110);
digitalWrite(SwitchED, HIGH);
break;
}
}
}

```

Прописуємо комбінації включення елементів 7-сегментного екрану для виведення усіх можливих варіантів арабських цифр десяткових значень:

```

switch(strd) {
case 0: {
digitalWrite(Switch, LOW);
shiftOut(data,sync, LSBFIRST, 0b11111100);
digitalWrite(Switch, HIGH);
}
}

```

```
    break;
  }
case 1: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b01100000);
    digitalWrite(Switch, HIGH);
    break;
}
case 2: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b11011010);
    digitalWrite(Switch, HIGH);
    break;
}
case 3: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b11110010);
    digitalWrite(Switch, HIGH);
    break;
}
case 4: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b01100110);
    digitalWrite(Switch, HIGH);
    break;
}
case 5: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b10110110);
    digitalWrite(Switch, HIGH);
```

```
    break;
}
case 6: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b10111110);
    digitalWrite(Switch, HIGH);
    break;
}
case 7: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b11100000);
    digitalWrite(Switch, HIGH);
    break;
}
case 8: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b11111110);
    digitalWrite(Switch, HIGH);
    break;
}
case 9: {
    digitalWrite(Switch, LOW);
    shiftOut(data, sync, LSBFIRST, 0b11110110);
    digitalWrite(Switch, HIGH);
    break;
}
}
}
```

Такий вигляд коду забезпечує виконання мікроконтролером поставлених перед ним задач по збору даних з датчика температури, обробки цих даних та

відправленні відповідних сигналів через Wi-Fi модуль до telegram-боту. Приклад користувацького інтерфейсу можемо бачити на рис. 3.4.

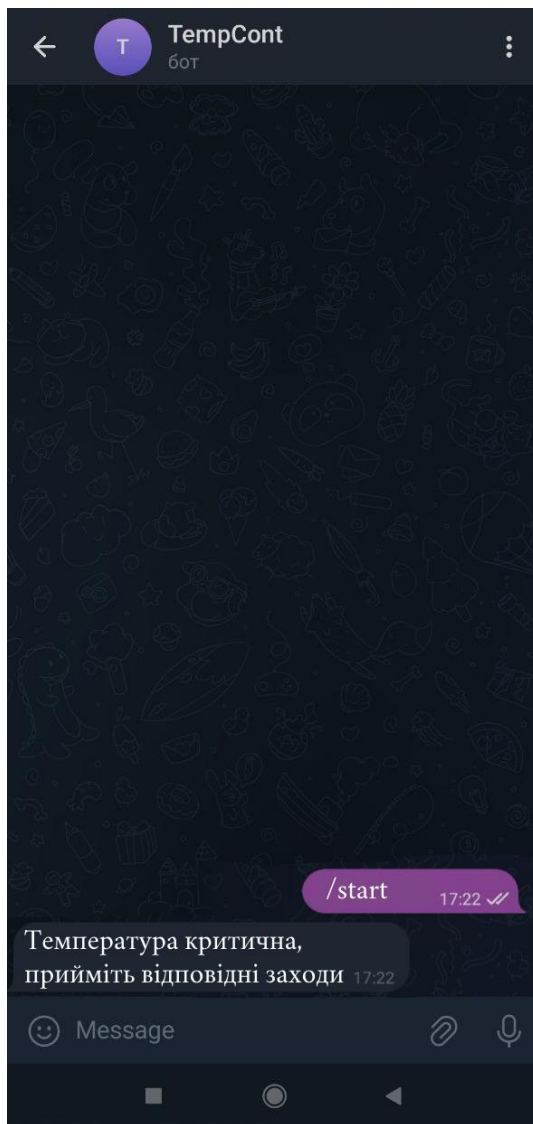


Рисунок 3.4 – Приклад користувацького інтерфейсу в Telegram

3.3. Інтеграція з Telegram API для сповіщення користувача

Wi-Fi модуль ESP8266 забезпечує бездротову передачу даних з мікроконтролера Arduino Uno на комп'ютер. Wi-Fi модуль отримує та передає данні у вигляді двійкового коду, значення якого інтерпретується модулем. Всього таких можливих комбінацій є вісім і їх роз'яснення можемо побачити в Таблиця 3.2.

Wi-Fi модуль не має захисту, у вигляді шифрування, даних. Тому передача даних через Telegram забезпечує певний рівень конфіденційності інформації про стан температури в приміщенні. Оскільки сам додаток Telegram використовує криптографічний протокол за допомогою алгоритмів симетричного шифрування.

Таблиця 3.2 Можливі комбінації двійкового коду

Значення	Десятичне значення	Роз'яснення
1	2	3
000	0	Немає даних
001	1	Завантаження даних по UART. Також дані для прошивки модуля.
010	2	Швидкий старт
011	3	Завантаження з флеш-пам'яті
100	4	SDIO low speed V2
101	5	SDIO high speed V1
110	6	SDIO low speed V1
111	7	SDIO high speed V2

Так як Python оптимальний для створення API для telegram-ботів, оскільки має розширений список бібліотек для виконання цієї задачі. Код для бота нашої системи дистанційного моніторингу температури, наведений нижче, повинен бути

скомпільований у середовищі програмування PyCharm, в якому є список готових бібліотек саме для роботи з Telegram.

Код починаємо з підключення відповідних бібліотек:

```
import os // підзавантаження бібліотек
import logging
from telegram import Update
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters,
CallbackContext
```

Створюємо профіль для користувача бота, де значення «asctime», «name», «levelname», «message» є змінними, оскільки є зверненнями до користувача:

```
logging.basicConfig(format='% (asctime)s - % (name)s - % (levelname)s - % (message)s',
level=logging.INFO)
```

Завдяки токєну буде відбуватися ідентифікація користувача через API до telegram-бота, що з ним пов'язаний, задаючи йому потрібні налаштування та параметри.

```
TELEGRAM_TOKEN = 'YOUR_BOT_TOKEN'
CHAT_ID = 'YOUR_CHAT_ID'
```

Параметри «'YOUR_BOT_TOKEN'» та «'YOUR_CHAT_ID'» – дані з самого бота.

Задаємо функцію для відправлення повідомлення про занадто низьку або високу температуру. Формат повідомлення, звичайно, також адаптується під потреби користувача:

```
def low_temperature_alert():
```

```
message = "Температура критична, прийміть відповідні заходи"
```

Задаємо команду відправлення повідомлень в чат:

```
bot.send_message(chat_id=CHAT_ID, text=message)
```

Ми створюємо функцію яка повідомляє користувачу про початок роботи бота:

```
def start(update: Update, context: CallbackContext) -> None:
```

```
    update.message.reply_text('Бот працює. Ви будете отримувати повідомлення про стан температури.')
```

Обов'язково прописуємо основну функцію, в середині якої будуть викликатися вже названі функції:

```
def main():
```

```
    global bot
```

```
    bot = Updater(TELEGRAM_TOKEN)
```

Підключаємо функцію запуску бота:

```
dp = bot.dispatcher
```

```
dp.add_handler(CommandHandler("start", start))
```

Ініціалізуємо запуск бота:

```
bot.start_polling()
```

```
bot.idle()
```

```
if __name__ == '__main__':
```

```
    main()
```

Даний код забезпечує роботу дистанційного оповіщення в системі дистанційного моніторингу автоматизованого контролю температури приміщення. Таким чином до telegram-боту будуть приходити повідомлення, за умови підключеного до мережі Інтернет комп'ютеру з скомпільованим кодом у середовищі програмування PyCharm та забезпечувати інтеграцію нашої системи з Telegram API для сповіщення користувача.

Таким чином система дистанційного моніторингу автоматизованого контролю температури доволі гнучка та легко адаптується під потреби користувача. І діапазон температур, і форма виводу значень температури до telegram-боту є налаштованими під зручність та вподобання користувача, що робить її доволі адаптивною.

Також, дана система може легко стати частиною більшої системи, виконуючи функцію оповіщення для дистанційного регулювання температури, стаючи таким чином модульною, тобто системою, що може використовуватися як окремо, так і в складі інших систем, легко інтегруючись в систему розумного будинку, але зберігаючи цей актуальний в сучасності швидкий формат звітності про стан температури за допомогою популярного та розповсюдженого додатку – Telegram.

ВИСНОВКИ

Була розроблена система на основі платформи Arduino, що відслідковує температуру у приміщенні та надсилає повідомлення у додаток Telegram, якщо температурний режим у приміщенні, де встановлена ця система відхиляється від заданої норми. Наявність стандартів і рішень моніторингу температури свідчить про важливість цього питання в різних сферах людської діяльності. Актуальність проекту дистанційного моніторингу автоматизованої системи контролю температури в приміщеннях на платформі Arduino зумовлена економічністю в виготовленні та запуску цієї системи, а також її низьким рівнем спожитку електроенергії, що доволі актуально для нинішньої енергетичної ситуації в Україні.. Вибравши універсальний та оптимальний для цього проекту мікроконтролер, було забезпечено підключення всіх елементів, що мали відповідати за різні типи оповіщення користувача про зміни мікроклімату. Збором даних з датчика температури, обробки цих даних та відправленні відповідних сигналів через Wi-Fi модуль до telegram-боту виконує мікроконтролер.

Так як, діапазон нормальних температур встановлюється в кодї, то дану систему можна пристосувати як для побутових задач, так і промислових. Дотримуючись правильної схеми підключення апаратної частини нашої системи, буде коректною і робота програмного коду, оскільки він звертається до всіх периферійних пристроїв, для яких був створений простір імен. Новаторським рішенням серед аналогічних проектів виступає впровадження системи оповіщення через один із найпопулярніших месенджерів – Telegram.

Таким чином дотримуються основні вимоги для забезпечення можливості використання системи контролю температури в різних типах приміщень. Модульність, за рахунок легкої інтеграції між платформою Arduino та додатком Telegram. Швидкодійність, мобільність та зручний формат отримання звіту про стан температури, за допомогою Wi-Fi модулю та telegram-боту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Васильєва Л.В., Кузьменко М.В., Садретдінов Я.С. Online application for calculating optimal microclimate conditions in a computer office // Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems». 2018. № 1 (37). С. 20-27. (дата звернення 16.09.2024).

2. «Основи мікропроцесорної техніки»: Конспект лекцій з дисципліни «Основи мікропроцесорної техніки» / Уклад. Ю.О. Оникієнко, Д.В. Тітков: - Київ.: «КПІ імені І. Сікорського», 2017. - 109 с.(дата звернення 17.09.2024).

3. Олехнович Д.О. Розробка системи безпеки жилого приміщення на базі мікроконтролера / Д.О. Олехнович – Запоріжжя: НУ «Запорізька політехніка», 2022. – 65 с.

4. Брух, С.В. Вплив особливостей метаболізму на повітряний баланс людини / С.В. Брух / АВОК: Вентиляція, опалення, кондиціонування повітря, теплопостачання та будівельна теплофізика. –2005. – № 5. – С.56-60.

5. Нефьодов А.В. Интегральные микросхемы та їхні зарубіжні аналоги. Довідник. Т.5. - М.: КУБК-а, 1997. - 608 с. (дата звернення 19.09.2024).

6. Василенко О.В., Сніжної Г.В., Ямненко Ю.С. Менеджмент цифрового виробництва: монографія. – Запоріжжя: Національний університет «Запорізька політехніка», 2022. – 120 с. ISBN 978-617-529-374-4.

7. Сівко В.І. Безпека виготовлення та експлуатації радіоелектронної апаратури: Навчальний посібник - Житомир: ЖИТИ, 2000 р. - 142 с. (дата звернення 02.10.2024).

8. Сніжної Г.В., Томашевський О. В., Степаненк С. М. Моніторинг і контроль складних стохастичних систем : монографія. – Запоріжжя: Національний університет «Запорізька політехніка», 2022. – 124 с. ISBN 978-617-529-373-7.

9. Chodorow K. MongoDB: The Definitive Guide, Second Edition. – O'Reilly Media, Inc., 2013. –393 с.; Harrison B.L., Consolvo S., Choudhury . Using multi-modal

sensing for human activity modeling in the real world / B.L.Harrison, S.Consolvo, T.Choudhury / Handbook of Ambient Intelligence and Smart Environments, Springer, 2010. - P. 463-478. (дата звернення 16.10.2024).

10. Савчук В. В. Методи та програмно-апаратні засоби систем клімат-контролю : кваліфікаційна робота магістра за спеціальністю 123 — «Комп'ютерна інженерія». Тернопіль: ТНТУ, 2022. - 71 с.

11. Черних, А.А. Блок клімат контролю приміщень / А.А. Черних / Збірник тез доповідей XX міжнародної науково-практичної конференції студентів, аспірантів і молодих вчених "Ресурсоенергозберігаючі технології та обладнання" – К.: «КПІ ім. Ігоря Сікорського», 2021. С.277-278.

12. Моніторинг і керування рівнями фізичних факторів виробничого середовища : навч.пос./ С.В. Сукач, дис. на здобуття наукового ступеня д-р. техн. наук; НАУ.- Київ, 2017, 311 с. (дата звернення 28.09.2024).

13. Система автоматизованої підтримки оптимального мікроклімату виробничого приміщення : навч.пос./ О.В. Строкань / Системи обробки інформації. - 2014. - Вип. 5. - С. 97-100.

14. Структура експертної системи інтелектуального регулювання мікроклімату житлових приміщень : навч.пос./ А. І. Купін, І. О. Музика, Д. І. Кузнєцов. – Запоріжжя, 2017. – 177 с. (дата звернення 14.10.2024).

15. Інформаційні системи і технології [Текст]: навч.пос./ М.М. Бенько. – Київ, 2010. – 325с. (дата звернення 16.10.2024).

16. DS-Electronics.com.ua. Будова та види датчиків температури в кліматичних системах. [Електронний ресурс] Режим доступу: <https://ds-electronics.com.ua/support/blog/termoregulatory/ustrojstvo-i-vidy-datchikov-temperature-v-klimaticheskikh-sistemah/> (дата звернення 28.09.2024).

17. Prom.ua. Пошук за запитом: Huato S500. [Електронний магазин] Режим доступу: https://prom.ua/ua/search?search_term=Huato%20S500 (дата звернення: 11.10.2024).

18. **Telegram (software)//Architecture//Privacy** [Електронний ресурс] Wikipedia// Режим доступу:

[https://en.wikipedia.org/wiki/Telegram_\(software\)#Architecture](https://en.wikipedia.org/wiki/Telegram_(software)#Architecture) (дата звернення: 12.10.2024).

19. **WiseHome.ua.** Що таке протокол для розумного будинку: 6 популярних видів. [Електронний ресурс] Режим доступу: <https://wisehome.ua/ua/scho-take-protokol-dlya-rozumnogo-budinku-6-populyarnih-vidiv/> (дата звернення: 17.10.2024).

20. **Analog Devices. Temperature Sensor Portfolio.** [Електронний документ] –Режим доступу: https://www.analog.com/media/en/news-marketing-collateral/product-selection-guide/TempSensorPortfolio_A.pdf (дата звернення 24.09.2024).

21. **Arduino.ua.** Датчик температури DS18B20. [Електронний магазин] – Режим доступу: <https://arduino.ua/prod190-datchik-temperatyri-ds18b20-cifrovoi> (дата звернення 25.09.2024).

22. **Arduino.ua.** Датчик вологості и температури DHT22. [Електронний магазин] – Режим доступу: <https://arduino.ua/prod301-datchik-vlajnosti-i-temperatyri-dht22> (дата звернення 27.09.2024).

23. **Arduino.ua.** Ethernet модуль ESP8266 версія ESP-07. [Електронний магазин] – Режим доступу: <https://arduino.ua/prod1444-wi-fi-modyl-esp8266-versiya-esp-07> (дата звернення 27.09.2024).

24. **ATmega328.** [Електронний ресурс]// Wikipedia// – Режим доступу: <https://en.wikipedia.org/wiki/ATmega328> (дата звернення 22.09.2024).

ДОДАТОК А

СХЕМИ ПІДКЛЮЧЕНЬ ДО ПЛАТИ

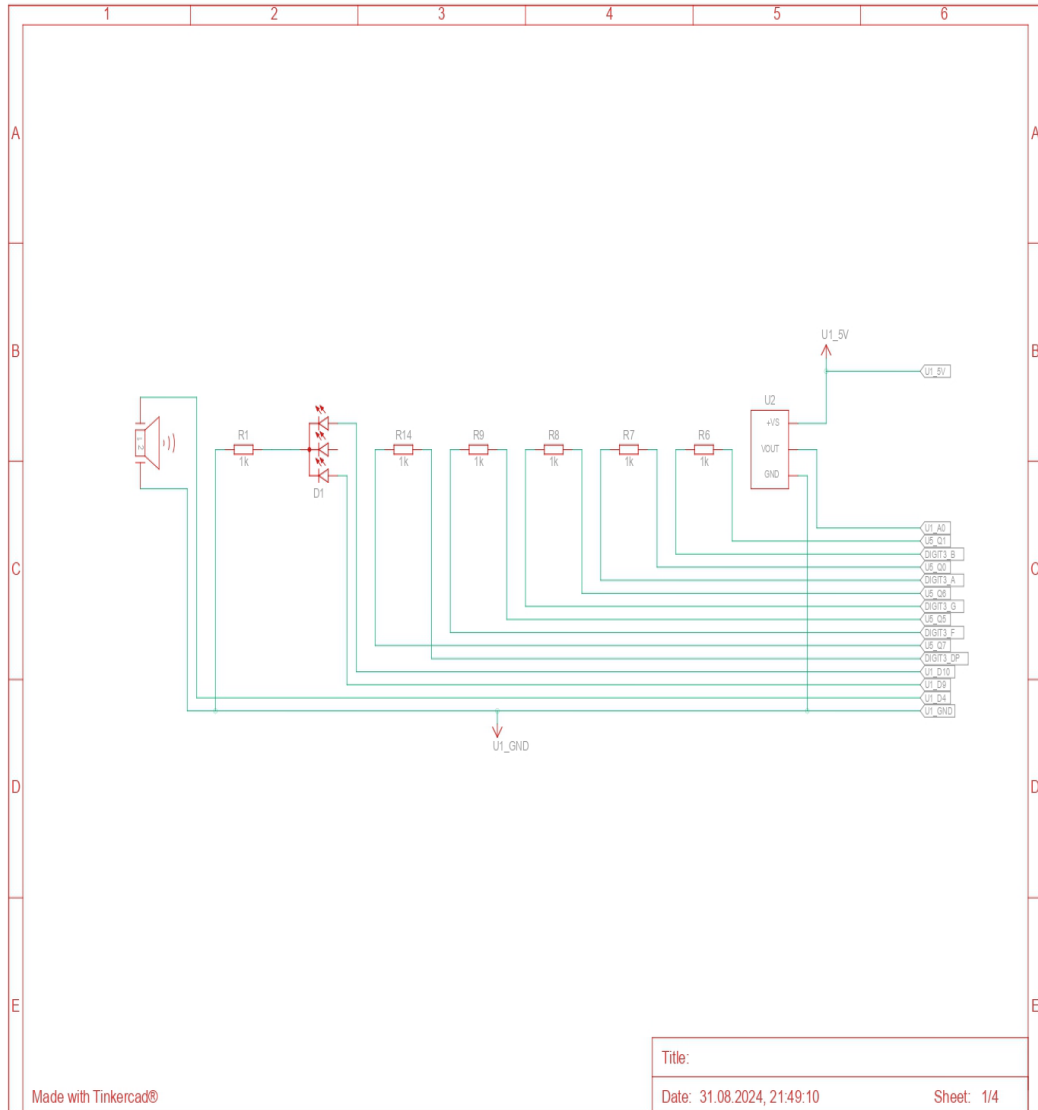


Рисунок А.1 – Схема підключень

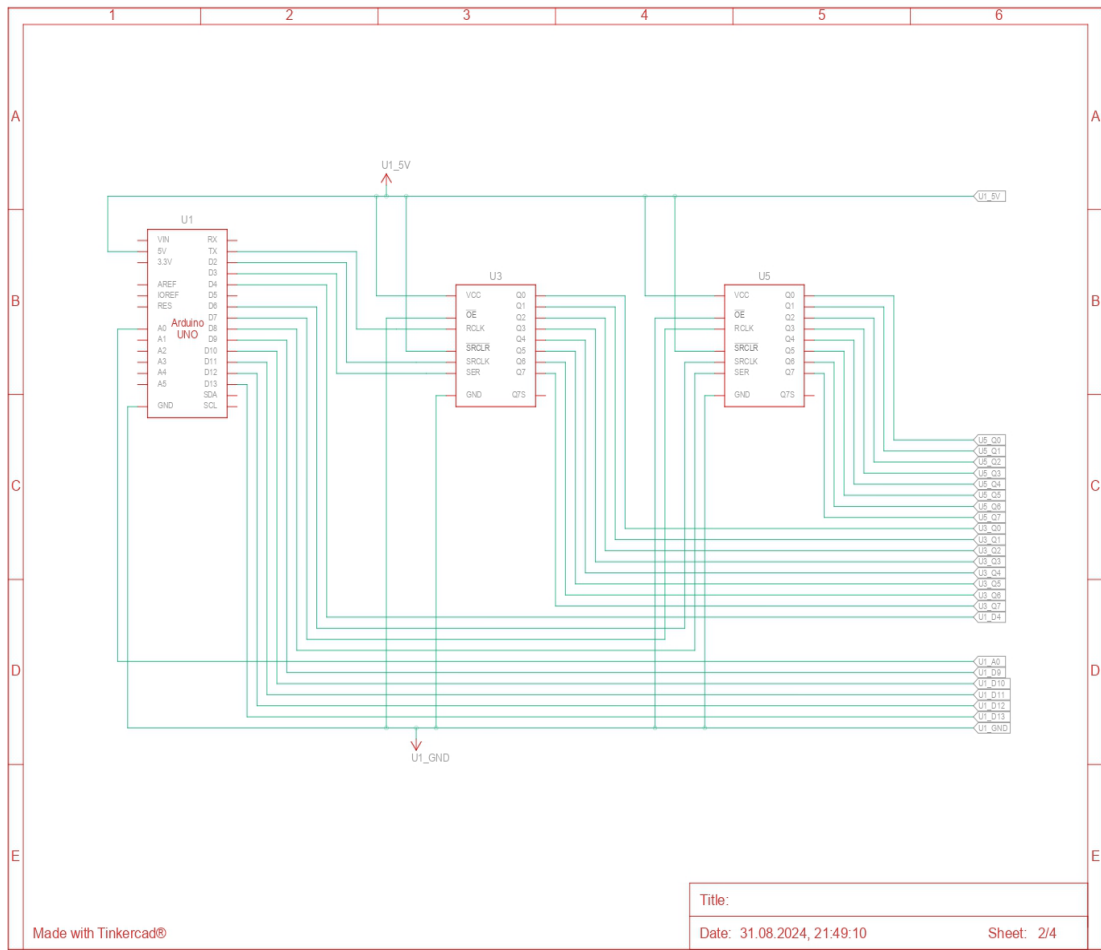


Рисунок А.2 – Схема підключень для Arduino UNO

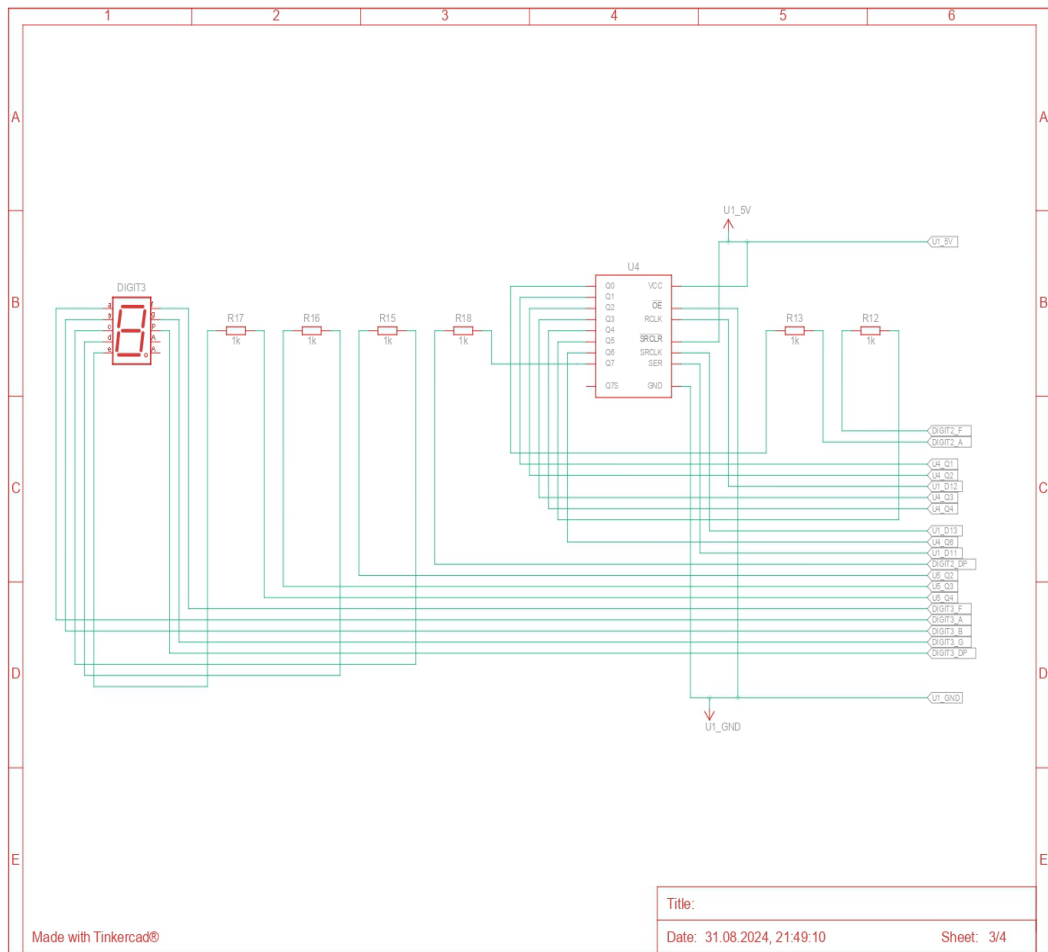


Рисунок А.3 – Схема підключень

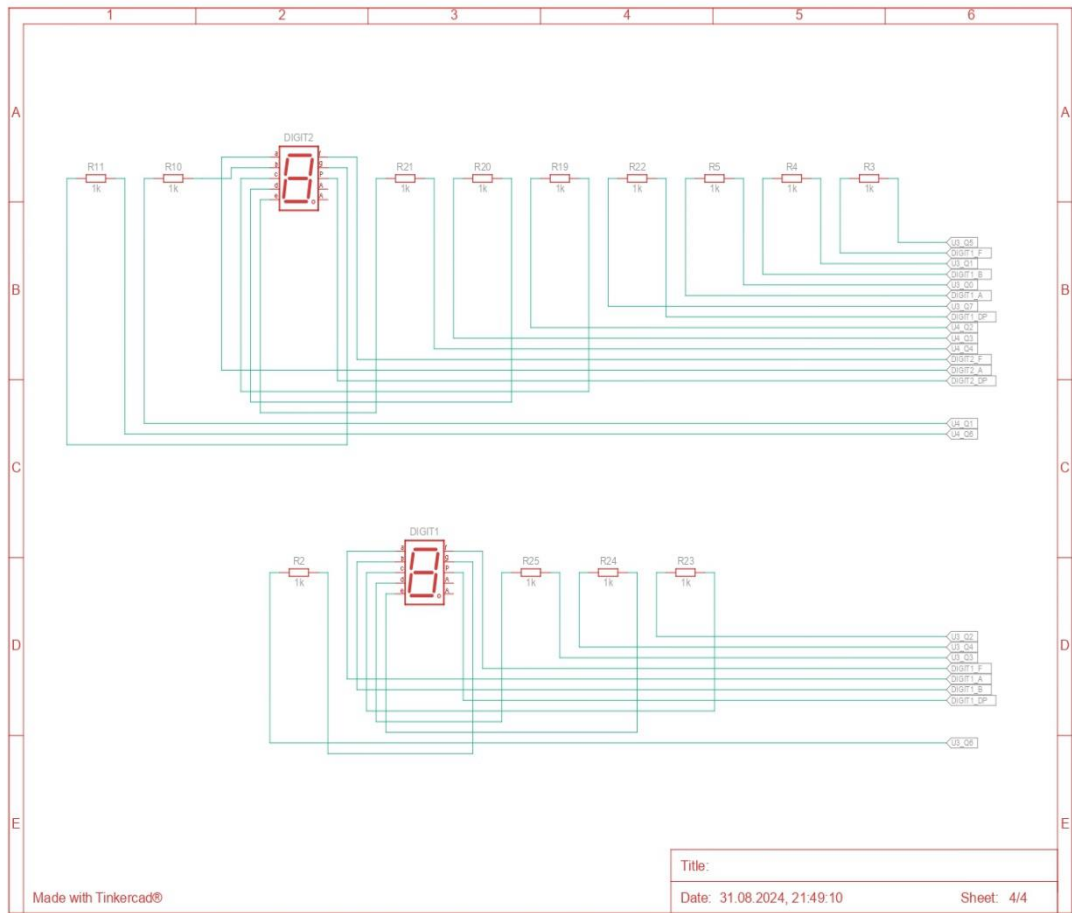


Рисунок А.4 – Схема підключень

ДОДАТОК Б ПРОГРАМНИЙ КОД

```
#include <WiFi.h>
#include <UniversalTelegramBot.h>

#define red 10
#define blue 9
#define zoomer 4
#define dataED 8
#define SwitchED 7
#define syncED 6
#define dataM 3
#define SwitchM 1
#define syncM 2
#define data 11
#define Switch 12
#define sync 13
#define temp A0
#define RX 0
#define TX 5

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

const char* botToken = "YOUR_BOT_TOKEN";
const char* chatID = "YOUR_CHAT_ID";
```

```
WiFiClient client;
UniversalTelegramBot bot(botToken, client);

void setup()
{
  pinMode (temp, INPUT);
  Serial.begin(9600);
  pinMode(red, OUTPUT);
  pinMode(blue, OUTPUT);

  pinMode(dataED, OUTPUT);
  pinMode(SwitchED, OUTPUT);
  pinMode(syncED, OUTPUT);

  digitalWrite(SwitchED, LOW);

  shiftOut(dataED,syncED, LSBFIRST, 0b11111100);

  digitalWrite(SwitchED, HIGH);

  pinMode(dataM, OUTPUT);
  pinMode(SwitchM, OUTPUT);
  pinMode(syncM, OUTPUT);

  digitalWrite(SwitchM, LOW);

  digitalWrite(SwitchM, HIGH);

  pinMode(data, OUTPUT);
```

```
pinMode(Switch, OUTPUT);
pinMode(sync, OUTPUT);

digitalWrite(Switch, LOW);

digitalWrite(Switch, HIGH);

}

void loop()
{
  int val = analogRead(A0);
  int tmp = map(val, 20, 358, -40, 125);

  //tmp%10
  //tmp/10

  if (tmp >= 5 && tmp <= 70) {
    analogWrite(zoomer, 0);
    analogWrite(blue, 255);
    analogWrite(red, 0);
    sendTelegramMessage("Все добре");
  }
  else
  {
    analogWrite(zoomer, 255);
    analogWrite(blue, 0);
    delay(200);
    analogWrite(red, 255);
    delay(200);
  }
}
```

```
    analogWrite(red,0);
    sendTelegramMessage("Температура критична, прийміть відповідні заходи ");
}
int str = tmp%10;
int strd = tmp/10;
if (str <0) {
    digitalWrite(SwitchM, LOW);
    shiftOut(dataM, syncM, LSBFIRST, 0b00000010);
    digitalWrite(SwitchM, HIGH);
    str = str*-1;
    strd = strd*-1;
}
else {
    shiftOut(dataM, syncM, LSBFIRST, 0b11111100);
}
delay(300);
Serial.println(tmp);
switch(str) {
    case 0: {
        digitalWrite(SwitchED, LOW);
        shiftOut(dataED, syncED, LSBFIRST, 0b11111100);
        digitalWrite(SwitchED, HIGH);
        break;
    }
    case 1: {
        digitalWrite(SwitchED, LOW);
        shiftOut(dataED, syncED, LSBFIRST, 0b01100000);
        digitalWrite(SwitchED, HIGH);
        break;
    }
}
```

```
case 2: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b11011010);
digitalWrite(SwitchED, HIGH);
break;
}
case 3: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b11110010);
digitalWrite(SwitchED, HIGH);
break;
}
case 4: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b01100110);
digitalWrite(SwitchED, HIGH);
break;
}
case 5: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b10110110);
digitalWrite(SwitchED, HIGH);
break;
}
case 6: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED,syncED, LSBFIRST, 0b10111110);
digitalWrite(SwitchED, HIGH);
break;
}
```

```
case 7: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED, syncED, LSBFIRST, 0b11100000);
digitalWrite(SwitchED, HIGH);
break;
}
case 8: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED, syncED, LSBFIRST, 0b11111110);
digitalWrite(SwitchED, HIGH);
break;
}
case 9: {
digitalWrite(SwitchED, LOW);
shiftOut(dataED, syncED, LSBFIRST, 0b11110110);
digitalWrite(SwitchED, HIGH);
break;
}
}

switch(strd) {
case 0: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b11111100);
digitalWrite(Switch, HIGH);
break;
}
case 1: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b01100000);
```

```
digitalWrite(Switch, HIGH);
break;
}
case 2: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b11011010);
digitalWrite(Switch, HIGH);
break;
}
case 3: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b11110010);
digitalWrite(Switch, HIGH);
break;
}
case 4: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b01100110);
digitalWrite(Switch, HIGH);
break;
}
case 5: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b10110110);
digitalWrite(Switch, HIGH);
break;
}
case 6: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b10111110);
```

```
digitalWrite(Switch, HIGH);
break;
}
case 7: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b11100000);
digitalWrite(Switch, HIGH);
break;
}
case 8: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b11111110);
digitalWrite(Switch, HIGH);
break;
}
case 9: {
digitalWrite(Switch, LOW);
shiftOut(data, sync, LSBFIRST, 0b11110110);
digitalWrite(Switch, HIGH);
break;
}
}
}
```

ДОДАТОК В-
ПРОГРАМНИЙ КОД ІНТЕРНЕТ МОДУЛЯ ДЛЯ ПІДКЛЮЧЕННЯ ЙОГО
ДО TELEGRAM

```
import os
import logging
from telegram import Update
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters,
CallbackContext

logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
level=logging.INFO)

TELEGRAM_TOKEN = 'YOUR_BOT_TOKEN'
CHAT_ID = 'YOUR_CHAT_ID'

def low_temperature_alert():
    message = "Температура критична, прийміть відповідні заходи"
    # Отправка сообщения в чат
    bot.send_message(chat_id=CHAT_ID, text=message)

def start(update: Update, context: CallbackContext) -> None:
    update.message.reply_text("Бот працює. Ви будете отримувати повідомлення про
стан температури")

def main():
    global bot
```

```
bot = Updater(TELEGRAM_TOKEN)

dp = bot.dispatcher
dp.add_handler(CommandHandler("start", start))

bot.start_polling()
bot.idle()

if __name__ == '__main__':
    main()
```