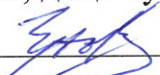


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ У МЕДИЦИНІ ТА ТЕЛЕКОМУНІКАЦІЯХ

«Допущено до захисту»

Завідувач кафедри
комп'ютерних технологій у
медицині та телекомунікаціях
к.т.н., доц. Чухов В.В.


« 11 » 06 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

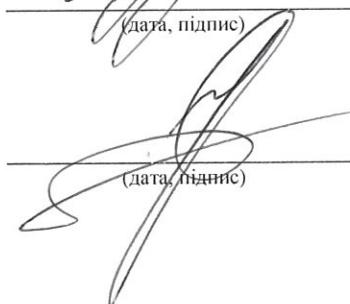
на тему Проектування медичного лабораторного smart-термостата
за спеціальністю 163 «Біомедична інженерія»
освітній ступінь «бакалавр»

Виконавець роботи
студент 4 курсу, гр.БІ-20-1

Керівник роботи
к.т.н., доц.


(дата, підпис)

Шевчук А.Л.


(дата, підпис)

Коломієць Р.О.

Житомир
2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ У МЕДИЦИНІ ТА ТЕЛЕКОМУНІКАЦІЯХ

Пояснювальна записка до кваліфікаційної роботи бакалавра

на тему Проектування медичного лабораторного smart-термостата

Виконав: студент 4 курсу,
групи БІ-20-1 спеціальності
163 «Біомедична інженерія»

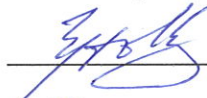
Шевчук А.Л.

Керівник Коломієць Р.О.

Рецензент Чухов В.В.

«Затверджено»

Завідувач кафедри комп'ютерних
технологій у медицині та
телекомунікаціях

 Владислав ЧУХОВ
« 02 » 02 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

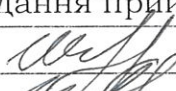
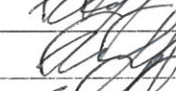
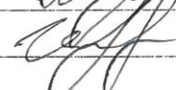
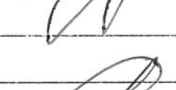
Студента Шевчука Артура Леонідовича
Тема роботи Проектування медичного лабораторного
smart - термостата

Затверджена Наказом університету від « ____ » ____ 20 ____ р. № ____

Термін здачі студентом завершеної роботи червень 2024 р.

Вихідні дані роботи: пристрій призначений для підтримання стабільної
температури в.г. +5°C до +60°C; умови експлуатації - ста-
ціонарний прилад, лабораторні умови; зв'язок з користувачем через
Telegram-bot; індикація результатів у цифровому вигляді;

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що
їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1.	Коломієць Р.О.	лютий 2024 р.	+ 
2.	Коломієць Р.О.	березень 2024 р.	+ 
3.	Коломієць Р.О.	березень 2024 р.	+ 
4.	Коломієць Р.О.	квітень 2024	+ 
5.	Коломієць Р.О.	травень 2024	+ 

Керівник  (підпис)

Календарный план

[illegible]

Студент

(підпис)

Керівник

(підпис)

РЕФЕРАТ

Метою роботи є проектування ефективного та надійного smart-термостата для медичних лабораторій, який забезпечить оптимальні умови зберігання та обробки біологічних зразків. Для досягнення цієї мети необхідно виконати наступні завдання:

1. Аналіз вимог до медичних лабораторних термостатів.
2. Розробка технічних вимог та специфікацій для smart-термостата.
3. Розробка апаратної та програмної частини термостата.
4. Налаштування та оптимізація роботи smart-термостата.

Зміст роботи

Перший розділ присвячено аналізу вимог до медичних лабораторних термостатів. У цьому розділі розглядаються існуючі рішення на ринку, та варіанти їх покращення та модернізації.

Другий розділ містить технічне завдання.

Третій розділ містить дослідження технічних вимог та розробку структурних схем для блоків медичного лабораторного smart-термостата. Визначаються основні функціональні можливості, необхідні для забезпечення стабільного температурного режиму.

Четвертий розділ включає розробку апаратної частини термостата. Описується вибір компонентів, таких як сенсори температури, мікроконтролери, компонентів блока живлення, а також схема електричних з'єднань. Також описується розробка програмного забезпечення для управління smart-термостатом. Описуються алгоритми контролю температури, обробки даних з сенсорів, віддаленого моніторингу та управління через інтернет.

П'ятий розділ містить загальну конструкцію приладу, розташування компонентів, та налаштування інтерфейсу користувача через Telegram bot.

ЗМІСТ

Вступ		8
1	Аналітичний огляд літературних джерел	10
1.1	Загальні відомості про медичний лабораторний термостат	10
1.2	Застосування медичного лабораторного термостата	14
1.3	Різновиди медичних лабораторних термостатів	16
1.4	Варіанти покращення та вдосконалення конструкції медичних лабораторних термостатів	20
1.5	Вимоги до smart-термостата	21
1.6	Висновки	22
2	Технічне завдання	24
3	Розроблення структурної схеми	25
3.1	Опис структурної схеми блоку живлення	25
3.2	Опис структурної схеми медичного лабораторного smart-термостата з мікроконтролером ESP	28
3.3	Опис архітектури програмного коду	29
3.3.1	Логічна структура коду	31
3.4	Структура та алгоритм роботи Telegram бота	31
3.4.1	Алгоритми роботи бота	32
3.4.2	Взаємодія модулів	33
4	Синтез і розрахунок електричної принципової схеми	34
4.1	Синтез та розрахунок електричної принципової схеми блоку живлення	34

					ІКБІ.480 004.005 - ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Проектування медичного лабораторного smart - термостата Літ. Арк. Акрушів 6 92 Державний університет «Житомирська політехніка» ФІКТ, гр БІ-20-1		
Розроб.		Шевчук А.П.					
Перевір.		Коломієць Р.О.					
Реценз.		Чухов В.В.					
Н. Контр.							
Затверд.		Чухов В.В.					

4.2 Синтез та розрахунок електричної принципової схеми медичного лабораторного smart-термостата з мікроконтролером ESP	42
4.3 Програмне забезпечення	45
4.4 Схема роботи Telegram бота	48
4.4.1 Код для ESP з використанням бібліотеки UniversalTelegramBot	48
4.4.2 Код програми з використанням таймера	52
5 Конструкторська частина	57
5.1 Загальна конструкція приладу	57
5.1.1 Розташування компонентів	62
5.2 Налаштування та підключення до Telegram-бота	63
5.3 Оцінка надійності приладу	66
Висновки	69
Перелік використаних джерел	70
Додатки	72
Додаток А	73
Додаток В	76
Додаток С	77
Додаток D	80

ВСТУП

Предметом даної роботи є розробка медичного лабораторного smart-термостата. Актуальність такої розробки обумовлена необхідністю забезпечення точного та надійного контролю температури в лабораторних умовах, що має ключове значення для успішного проведення медичних досліджень та діагностичних процедур. Сучасні медичні дослідження все частіше вимагають застосування високотехнологічних засобів, які дозволяють забезпечити стабільні умови зберігання та аналізу біологічних зразків. Термостатування є однією з найважливіших функцій, необхідних для підтримання оптимальних умов під час проведення лабораторних тестів, вирощування клітинних культур, зберігання реактивів та інших біоматеріалів.

На сьогоднішній день розробка та впровадження smart-термостатів набуває все більшого поширення завдяки їх високій точності, надійності та можливості автоматичного регулювання температури відповідно до заданих параметрів. Це дозволяє зменшити ймовірність помилок, пов'язаних із людським фактором, і забезпечити стабільність проведення експериментів протягом тривалого часу. Крім того, інтеграція термостатів з цифровими системами керування та моніторингу дозволяє здійснювати віддалений контроль та аналіз даних, що є особливо важливим у контексті сучасних тенденцій розвитку медицини та біотехнологій.

На ринку медичного обладнання представлена певна кількість моделей термостатів, проте не всі вони відповідають вимогам сучасної лабораторної практики. Це може бути пов'язано з тим, що розробка такого обладнання вимагає врахування специфічних потреб медичних лабораторій, зокрема високої точності регулювання температури, можливості автоматизації процесів та забезпечення безпеки при експлуатації.

Дана робота представляє спробу розробити медичний лабораторний smart-термостат, який буде відповідати всім вказаним вимогам. В роботі проведено аналіз існуючих конструкцій термостатів, на основі аналізу розроблені

					ІКБІ.480 004.005 - ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

структурна та функціональна схеми пристрою, синтезована та розрахована електрична принципова схема, а також розроблена конструкція приладу.

Можливо, у майбутньому smart-термостати стануть невід'ємною частиною медичних лабораторій ХХІ століття, сприяючи підвищенню якості та надійності медичних досліджень.

					ІКБІ.480 004.005 - ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1.1 Загальні відомості про медичний лабораторний термостат

Медичні лабораторні термостати - це високоточні прилади, призначені для підтримки стабільної температури в робочій камері протягом тривалого періоду. Вони є критично важливими у багатьох медичних та наукових дослідженнях, де необхідна постійна температура для досягнення точних та повторюваних результатів. Термостати використовуються в різних галузях, включаючи бактеріологію, вірусологію, біологію, фармацевтику та контроль якості.

Основні галузі застосування термостатів:

1. Бактеріологія:

- Вирощування та культивування бактерій: Термостати забезпечують оптимальні умови для росту бактерій, що необхідно для вивчення їх фізіології, розробки антибіотиків та інших медичних досліджень.
- Інкубація культур: Використовуються для підтримки сталих температур під час інкубації бактеріальних культур, що є важливим для їх правильного розвитку та аналізу.

2. Вірусологія:

- Дослідження вірусів та їх життєвого циклу: Термостати дозволяють підтримувати постійну температуру, необхідну для дослідження вірусів, їх реплікації та взаємодії з клітинами-господарями.
- Культивування вірусів: Використовуються для створення сприятливих умов для культивування вірусів в клітинних культурах, що необхідно для розробки вакцин та антивірусних препаратів.

					ІКБІ.480 004.005 - ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Біологія:

- Дослідження ферментативних реакцій: Підтримка стабільної температури є критично важливою для вивчення ферментативної активності та інших біологічних процесів, що залежать від температури.
- Розвиток ембріонів: Термостати використовуються для інкубації ембріонів при сталих температурах, що важливо для досліджень у галузі розвитку та генетики.

4. Фармацевтика:

- Тестування нових ліків та вакцин: Термостати застосовуються для проведення стабільних та контрольованих температурних умов під час тестування ефективності та безпечності нових фармацевтичних препаратів.
- Зберігання зразків: Використовуються для зберігання чутливих до температури зразків, таких як реагенти, біологічні матеріали та зразки ліків.

5. Контроль якості:

- Тестування продуктів харчування: Підтримка сталих температур дозволяє проводити мікробіологічні аналізи харчових продуктів для забезпечення їх безпечності та якості.
- Медичні вироби та інші матеріали: Термостати застосовуються для тестування стабільності та надійності медичних виробів, косметики та інших продуктів в умовах різних температур.

Ключові характеристики медичних лабораторних термостатів:

- Точність температурного контролю: Високоточні сенсори та системи регулювання, що забезпечують підтримку заданої температури з мінімальними коливаннями.

					ІКБІ.480 004.005 - ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

- Стабільність температури: Можливість підтримки постійної температури протягом тривалого часу без значних змін.
- Широкий діапазон температур: Можливість встановлення та підтримки як дуже низьких, так і високих температур, залежно від потреб дослідження.
- Гнучкість налаштувань: Регулювання параметрів для специфічних досліджень та експериментів.
- Надійність та довговічність: Використання якісних матеріалів та технологій для забезпечення тривалого терміну служби приладу.

Медичні лабораторні термостати є важливими інструментами, що забезпечують точність, надійність та ефективність у проведенні різноманітних наукових та медичних досліджень.

Призначення та функції

Основне призначення медичного лабораторного термостата - підтримка стабільної температури в робочій камері з високим ступенем точності. Це досягається за допомогою системи нагрівання, охолодження та регулювання, яка забезпечує однорідність температурного режиму по всій камері.

Функції медичного лабораторного термостата можуть варіюватися залежно від моделі, але зазвичай включають:

- Встановлення та підтримка заданої температури: термостат може бути запрограмований на певну температуру, яку він буде підтримувати протягом певного періоду часу.
- Контроль та моніторинг температури: термостат постійно контролює та відображає температуру в камері.
- Сигналізація про перевищення температури: термостат може подавати сигнал тривоги, якщо температура в камері виходить за встановлені межі.
- Програмування: деякі термостати дозволяють програмувати цикли температури та часу, що може бути корисно для певних досліджень.

					ІКБІ.480 004.005 - ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- Запис даних: деякі термостати можуть записувати дані про температуру та час протягом певного періоду часу.
- Підключення до комп'ютера: деякі термостати можуть підключатися до комп'ютера для запису даних та моніторингу.

Класифікація

Медичні лабораторні термостати можна класифікувати за кількома характеристиками, включаючи:

- Тип:
 - Сухоповітряні: використовують повітря для нагрівання та охолодження камери.
 - Водяні: використовують воду для нагрівання та охолодження камери.
 - Вакуумні: створюють вакуум в камері, що дозволяє проводити дослідження при низькому тиску.
 - Стерилізаційні: використовуються для стерилізації лабораторного посуду та інструментів.
 - Криотермостати: підтримують дуже низькі температури, до -80°C або нижче.
 - Інфрачервоні: використовують інфрачервоне випромінювання для нагрівання зразків.
- Об'єм камери:
 - Малі (до 10 літрів)
 - Середні (10-50 літрів)
 - Великі (понад 50 літрів)
- Система керування:
 - Аналогове
 - Цифрове
 - Комп'ютерне

					ІКБІ.480 004.005 - ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаткові функції:

- Сигналізація про перевищення температури
- Освітлення камери
- Порт для підключення до комп'ютера
- Запис даних
- Програмованість

Вибір термостата

При виборі медичного лабораторного термостата важливо врахувати такі фактори:

- Тип дослідження: тип дослідження, яке буде проводитися, визначить тип термостата, який потрібен.
- Діапазон температур: необхідний діапазон температур дослідження визначить діапазон температур термостата.
- Об'єм камери: необхідний об'єм камери визначить розмір термостата.
- Функції: необхідні функції термостата визначають модель термостата.
- Ціна: ціна термостата буде залежати від його

1.2 Застосування медичного лабораторного термостата

1. Мікробіологія

У мікробіології термостати використовуються для інкубації різних зразків, включаючи культури бактерій, грибків та інших мікроорганізмів. Вони забезпечують необхідні температурні умови для росту і розвитку мікроорганізмів. Термостати підтримують стабільну температуру, необхідну для вирощування мікробіологічних культур. Це важливо для діагностики інфекційних захворювань та дослідження антибіотикорезистентності. Лабораторні термостати використовуються для перевірки стерильності медичних матеріалів та препаратів.

					ІКБІ.480 004.005 - ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Багато біохімічних тестів вимагають інкубації зразків при певній температурі для виявлення ферментативної активності або метаболічних процесів.

2. Клінічна біохімія

У клінічній біохімії термостати використовуються для проведення різноманітних аналізів, таких як гормональні дослідження, інкубація зразків крові при певних температурах дозволяє визначити рівні різних гормонів, що важливо для діагностики ендокринних порушень. Аналізи ферментів, вимірювання активності ферментів при стабільній температурі дозволяє оцінити функціональний стан органів, таких як печінка та підшлункова залоза. Імунологічні дослідження: Багато імунологічних тестів, таких як ELISA, потребують інкубації при постійній температурі для виявлення антитіл або антигенів.

3. Генетика та молекулярна біологія

В генетичних та молекулярно-біологічних дослідженнях термостати застосовуються для PCR (Полімеразної ланцюгової реакції): Для ампліфікації ДНК потрібні циклічні зміни температури, які забезпечуються спеціалізованими термостатами, такими як термоциклер. Гібридизації - ідтримання стабільної температури необхідне для процесів гібридизації ДНК та РНК, що є критичним етапом в багатьох молекулярно-біологічних методах. Електрофорез - термостати можуть використовуватися для контролю температури під час електрофорезу, що важливо для чіткого розділення нуклеїнових кислот або білків.

4. Виробництво біопрепаратів

Виробництво біопрепаратів, таких як вакцини, антитіла та ферменти, часто вимагає використання термостатів для контролю умов ферментації і очищення. Ферментація - термостати забезпечують оптимальні умови для росту мікроорганізмів або клітинних культур, які виробляють необхідні біологічні

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

продукти. Зберігання зразків, контрольована температура потрібна для зберігання зразків під час різних етапів виробництва, щоб запобігти їх деградації.

5. Патологія та гістологія

У патології та гістології термостати використовуються для обробки тканинних зразків. Фіксація тканин для правильного збереження зразків тканин перед подальшим дослідженням необхідно дотримуватись певного температурного режиму. Інкубація з барвниками: У процесі гістологічного фарбування зразків необхідна інкубація з різними барвниками при конкретних температурах для виявлення різних клітинних компонентів.

6. Дослідження в фармакології

У фармакологічних дослідженнях термостати використовуються для тестування стабільності ліків. Інкубація фармацевтичних зразків при різних температурах дозволяє визначити їх стабільність і термін придатності. Дослідження метаболізму термостати допомагають моделювати умови, за яких відбувається метаболізм лікарських засобів в організмі. Медичні лабораторні термостати є незамінним інструментом у різних галузях медичних досліджень і діагностики. Вони забезпечують точний контроль температури, що є критично важливим для точності і надійності результатів. Завдяки різноманітності їх застосування, термостати сприяють розвитку науки та покращенню медичних послуг.

1.3 Різновиди медичних лабораторних термостатів

Медичні лабораторні термостати складаються з різноманітної апаратури, яка забезпечує точний контроль температури для проведення різноманітних медичних та наукових досліджень. Основні компоненти та типи апаратури включають наступне.

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

1. Інкубатори

Інкубатори – це закриті камери, які підтримують стабільну температуру і, часто, вологість. Вони використовуються для вирощування мікроорганізмів, клітинних культур і ембріонів.

Основні характеристики включають контроль температури. Високоточні термостати забезпечують постійну температуру з мінімальними відхиленнями. Контроль вологості, деякі інкубатори мають системи для підтримання необхідного рівня вологості. Фільтрація повітря, використовуються НЕРА-фільтри для забезпечення стерильності середовища.

2. Водяні бані (водяні термостати)

Водяні бані – це пристрої, які використовують нагріту воду для підтримання стабільної температури зразків. Вони ідеальні для: інкубації біохімічних та ферментативних реакцій, теплової обробки зразків, розчинення речовин при контрольованій температурі.



Рисунок 1.1 – Баня водяна БН-09.2

3. Сухі термостати (Сухі Бані)

Сухі термостати або сухі бані використовують нагрітий повітря для контролю температури. Вони часто застосовуються для інкубації пробірок та мікропробірок, підготовки зразків для PCR, зберігання реагентів при заданій температурі.



Рисунок 1.2 – Лабораторний сухий інкубатор MD-08

4. Термоциклер

Термоциклер – це спеціалізований тип термостата, який використовується для полімеразної ланцюгової реакції (PCR). Основні функції включають програмовані цикли нагріву та охолодження для ампліфікації ДНК. Висока точність контролю температури з можливістю швидкого змінення температури.



Рисунок 1.3 – Термоциклер C1000

5. Холодильні термостати

Холодильні термостати підтримують знижені температури для зберігання зразків, які можуть деградувати при вищих температурах. Вони важливі для зберігання зразків крові, плазми та інших біологічних матеріалів. Консервації ферментів та реактивів. Підтримання умов для довгострокового зберігання зразків.

6. Термостати з мікропроцесорним управлінням

Термостати з мікропроцесорним управлінням оснащені електронними контролерами, які дозволяють точне програмування і моніторинг температури. Основні переваги включають висока точність і стабільність температури. Можливість програмування температурних профілів для складних експериментів. Інтеграція з комп'ютерними системами для автоматизації процесів і збору даних.

7. Портативні термостати

Портативні термостати забезпечують контроль температури в польових умовах або в умовах обмеженого простору. Вони використовуються для транспортування зразків при контрольованій температурі, польових досліджень та експериментів, експрес-аналізів на місці проведення досліджень.



Рисунок 1.4 - Портативний термостат Minitube зі встановленою температурою +37°C

Медичні лабораторні термостати є важливим інструментом у сучасних медичних та наукових дослідженнях. Різноманітність типів та моделей термостатів дозволяє забезпечити точний контроль температури для найрізноманітніших застосувань, від мікробіології до молекулярної біології та фармакології. Ці пристрої допомагають підвищити точність і надійність результатів досліджень, що, в свою чергу, сприяє розвитку науки та покращенню медичних послуг.

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

1.4 Варіанти покращення та вдосконалення конструкції медичних лабораторних термостатів

1. Підвищення точності та стабільності температури

Використання високоточних температурних сенсорів з більш швидким відгуком для забезпечення більш точного контролю температури. Впровадження багатозонного контролю температури, який дозволить підтримувати різні температури у різних частинах термостата.

2. Інтеграція з розумними технологіями

Проведення технологій IoT для віддаленого моніторингу та управління термостатами через інтернет. Це дозволить контролювати пристрої з будь-якої точки світу. Використання алгоритмів машинного навчання для оптимізації режимів роботи термостатів на основі історичних даних і прогнозів.

3. Покращення Енергоефективності

Використання матеріалів з високою теплоізоляцією для зменшення втрат енергії та підвищення ефективності нагрівання та охолодження. Впровадження інноваційних технологій для зниження споживання енергії, таких як теплові насоси або рекуператори тепла.

4. Збільшення Функціональності

Розробка термостатів з можливістю одночасного виконання кількох функцій, наприклад, інкубації та охолодження. Впровадження модульних конструкцій, що дозволить легко замінювати або додавати різні компоненти залежно від потреб користувача.

5. Покращення Зручності Використання

Розробка конструкцій з урахуванням ергономічних принципів для зручного використання та обслуговування. Створення зручних користувацьких інтерфейсів з сенсорними екранами та можливістю голосового управління.

6. Підвищення Надійності та Безпеки

Впровадження систем самодіагностики, які автоматично виявляють і повідомляють про несправності або відхилення від нормальних параметрів

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

роботи. Використання екологічно чистих і безпечних матеріалів, що не шкодять здоров'ю користувачів і пацієнтів.

7. Розширення Можливостей Досліджень

Розробка термостатів, що можуть працювати в більш широкому діапазоні температур, що дозволить проводити дослідження в екстремальних умовах. Виготовлення спеціалізованих моделей для певних видів досліджень, таких як генетика, біохімія або фармакологія.

Вдосконалення конструкції медичних лабораторних термостатів сприятиме підвищенню точності та надійності досліджень, енергоефективності, зручності використання та безпеки. Використання сучасних технологій і матеріалів, а також інтеграція з розумними системами дозволить значно розширити функціональні можливості цих пристроїв, забезпечуючи тим самим більш високий рівень наукових та медичних досліджень.

1.5 Вимоги до smart-термостата

У цьому розділі детально розглянуті вимоги до розробки та функціональності smart-термостата для медичного лабораторію. Під час етапу проектування та розробки такого пристрою визначаються основні функціональні та технічні характеристики, які є ключовими для створення ефективного та функціонального пристрою, відповідного потребам медичного лабораторію.

Основні функціональні вимоги:

Smart-термостат повинен забезпечувати точне та стабільне регулювання температури у медичній камері для зберігання зразків та реагентів. Пристрій має постійно вимірювати температуру та вологість у медичній камері і передавати ці дані для обробки системі керування або відображати їх на інтерфейсі користувача. Smart-термостат повинен мати можливість підключення до мережі Інтернет або до системи керування лабораторією для забезпечення дистанційного моніторингу та керування. Програмне та апаратне забезпечення

					ІКБІ.480 004.005 - ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

пристрою мають працювати надійно протягом тривалого часу без виникнення непередбачених ситуацій або збоїв.

Основні технічні вимоги:

Сенсори температури та вологості, які використовуються в пристрої, повинні мати високу точність для забезпечення достовірних даних про умови зберігання. Smart-термостат повинен мати можливість бездротового підключення до мережі Wi-Fi або інших пристроїв для обміну даними та керування. Інтерфейс користувача повинен бути зрозумілим та зручним у використанні для операторів, щоб швидко та ефективно налаштовувати та контролювати параметри пристрою. Smart-термостат повинен мати вбудовану систему живлення з резервним джерелом, що забезпечить безперервну роботу пристрою, навіть у випадку виникнення перебоїв у живленні.

1.6 Висновки

Медичні лабораторні термостати є важливими інструментами в наукових і медичних дослідженнях, забезпечуючи точний контроль температури для досягнення стабільних і повторюваних результатів. Вони використовуються в різних галузях, таких як бактеріологія, вірусологія, біологія, фармацевтика та контроль якості. Завдяки своїй здатності підтримувати постійну температуру, термостати сприяють ефективному проведенню досліджень, вирощуванню культур, дослідженню вірусів, проведенню ферментативних реакцій, тестуванню нових ліків і багато іншого.

Основні типи термостатів включають інкубатори, водяні та сухі термостати, термоциклер, холодильні термостати, термостати з мікропроцесорним управлінням та портативні термостати. Вони відрізняються за типом використання, об'ємом камери, системою керування та додатковими функціями. Вибір термостата залежить від специфіки дослідження, діапазону температур, об'єму камери та необхідних функцій.

					ІКБІ.480 004.005 - ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Покращення конструкції термостатів включає підвищення точності та стабільності температури, інтеграцію з розумними технологіями, покращення енергоефективності, збільшення функціональності, зручності використання, надійності та безпеки. Використання сучасних матеріалів, інноваційних технологій та алгоритмів машинного навчання сприятиме покращенню точності та ефективності роботи термостатів, що, в свою чергу, підвищить якість наукових та медичних досліджень.

					ІКБІ.480 004.005 - ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ТЕХНІЧНЕ ЗАВДАННЯ

1. Призначення та мета створення виробу: пристрій призначений для підтримки стабільної температури в робочій камері з високим ступенем точності. Він використовується для забезпечення оптимальних умов для різних медичних та наукових досліджень, таких як вирощування мікроорганізмів, інкубація зразків, проведення біохімічних тестів та контроль якості медичних матеріалів.

2. Технічні характеристики:

- Діапазон температури, що підтримується: від $+5^{\circ}\text{C}$ до $+60^{\circ}\text{C}$.

- Точність підтримки температури: Від $\pm 0.1^{\circ}\text{C}$ до $\pm 1.0^{\circ}\text{C}$.

- Об'єм робочої камери: 20 л.

- Тип контролю температури: Електронний з цифровим дисплеєм. Програмовані цикли нагріву/охолодження.

- Система управління: Інтегрований мікропроцесорний контролер, кнопкова панель. Підтримка підключення до комп'ютера або мережі для віддаленого управління та моніторингу (Wi-Fi, Ethernet).

- Безпека: Автоматичне відключення при перевищенні заданої температури.

- Рівномірність розподілу температури: Примусова циркуляція повітря для забезпечення однорідного температурного поля.

- Інтерфейси: USB, Ethernet, Wi-Fi для передачі даних та програмування.

- Додаткові функції: Таймер. Запис даних. Можливість калібрування.

3. Кінцевий результат проектування: електрична принципова схема пристрою, друкована плата, ескіз конструкції, програмний код.

					ІКБІ.480 004.005 - ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБЛЕННЯ СТРУКТУРНОЇ СХЕМИ

Розроблення структурної схеми медичного лабораторного smart термостата включає визначення основних функціональних блоків і їх взаємодії.



Рисунок 3.1 - Структурна схеми медичного лабораторного smart - термостата

3.1 Опис структурної схеми блоку живлення

Виходячи з принципів функціонування медичних термостатів, розглянутих у розділі 1, можливо дійти до висновку, що структурна схема медичного лабораторного smart-термостата буде складатися блоку живлення включає визначення основних функціональних елементів та їх взаємодії.

Блок живлення починається з вхідного фільтра, який призначений для згладжування електромагнітних перешкод та високочастотних шумів, що можуть надходити з мережі живлення. Це важливий компонент, який захищає електронні частини пристрою від потенційно шкідливих перешкод.

Далі, електричний сигнал проходить через трансформатор. Трансформатор знижує вхідну змінну напругу до рівня, прийнятного для подальшої обробки. Він також забезпечує ізоляцію високовольтової частини від низьковольтової для забезпечення безпеки.

Після трансформатора сигнал потрапляє у випрямляч. Випрямляч перетворює змінний струм на постійний. Існують два основних типи випрямлячів: однопівперіодний і двопівперіодний (містковий). Зазвичай у випрямлячах використовуються діоди або діодні мости для реалізації цієї функції.

Після випрямлення струму необхідно згладити пульсації постійного струму. Цю функцію виконує фільтр, який складається з великих конденсаторів. Ці конденсатори забезпечують рівний постійний струм, усуваючи залишкові коливання після випрямлення.

Наступним елементом у схемі є регулятор напруги. Регулятор стабілізує вихідну напругу до необхідного рівня, незалежно від коливань вхідної напруги або навантаження. Регулятори напруги можуть бути лінійними, наприклад, на основі стабілітронів, або імпульсними, як-от ШІМ-регулятори.

Вторинний фільтр: Цей фільтр додатково згладжує пульсації після регулятора напруги для забезпечення стабільного струму.

DC/DC перетворювач: Він конвертує напругу до необхідних рівнів для різних частин системи, таких як 5V, 3.3V тощо.

На виході блоку живлення встановлені захисні компоненти. Вони захищають систему від перевантажень по струму, коротких замикань та перенапруг. Зазвичай ці компоненти включають запобіжники, варистори та діоди захисту.

					ІКБІ.480 004.005 - ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Запобіжники захищають систему від надмірного струму, варистори захищають від стрибків напруги, а діоди захисту запобігають зворотній полярності.

В результаті, кожен з цих функціональних елементів відіграє ключову роль у забезпеченні стабільного і безпечного живлення медичного лабораторного смарт термостата. Всі елементи працюють разом, щоб забезпечити надійну роботу термостата в широкому діапазоні умов.

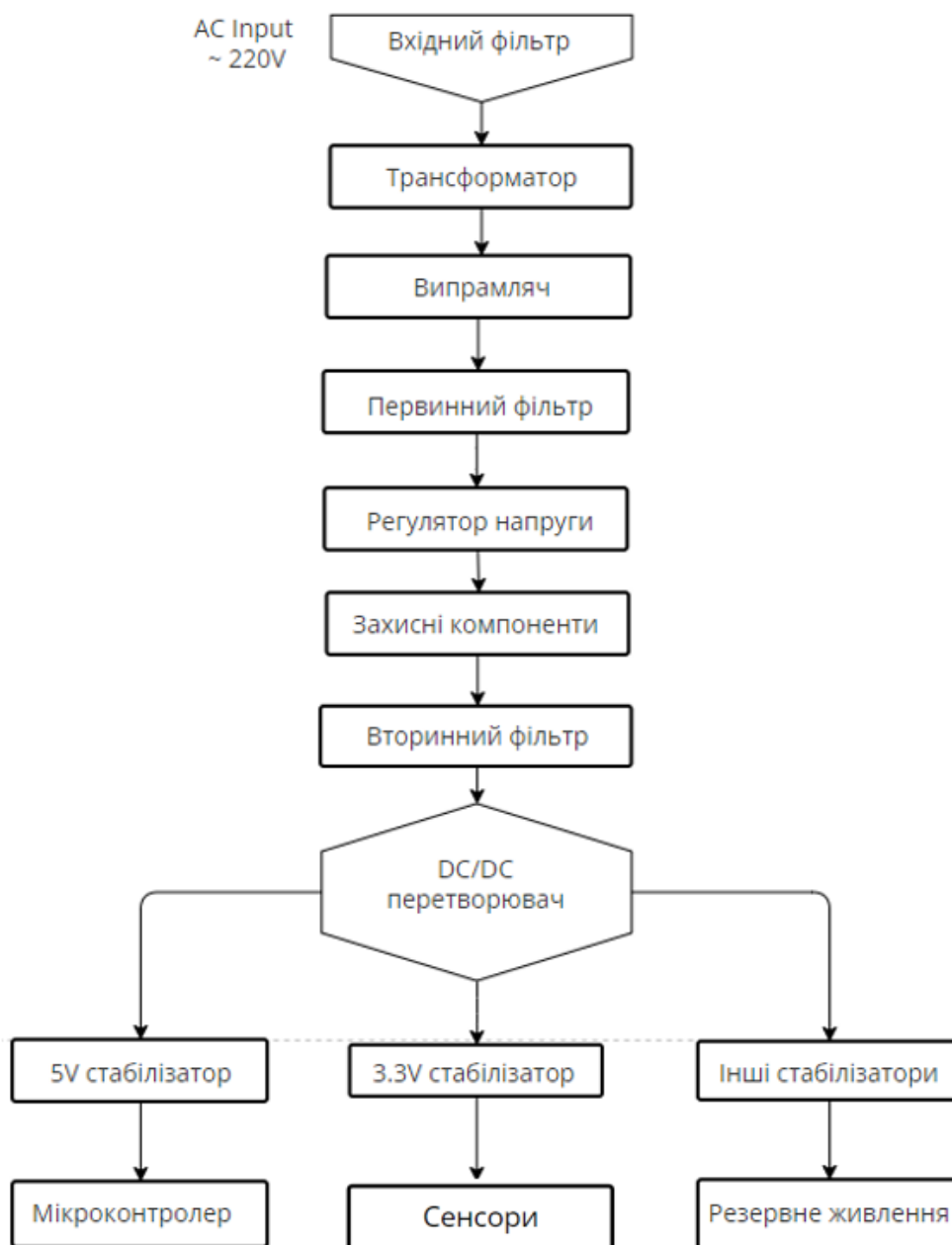


Рисунок 3.2 – Структурна схема блоку живлення

3.2 Опис структурної схеми медичного лабораторного smart-термостата з мікроконтролером ESP

Розглянемо структурну схему медичного лабораторного смарт термостата з мікроконтролером ESP та сенсорами, а також з нагрівальними та охолоджувальними елементами.

Мікроконтролер ESP8266 або ESP32 відповідає за керування та моніторинг різних функцій термостата. Отримує дані від сенсорів температури, вологості та інших, і виконує алгоритми регулювання температури. Взаємодіє з іншими компонентами через різні інтерфейси (наприклад, UART, SPI, I2C). Температурні сенсори (наприклад, термістори або термопари): вимірюють температуру в різних ділянках термостата та зразків. Сенсори вологості: вимірюють вологість у робочій камері. Інші сенсори: можуть вимірювати, наприклад, рівень CO₂, рівень освітленості тощо.

Нагрівальні елементи (наприклад, резистивні нагрівачі): використовуються для підтримки необхідної температури. Система охолодження - вентилятори за потреби охолоджує робочу камеру термостата.

Wi-Fi або Ethernet: для підключення до мережі Інтернет. USB: для підключення до комп'ютера для програмування та налагодження. Інші зовнішні інтерфейси: можуть бути додані для підключення зовнішніх пристроїв, таких як додаткові датчики або індикатори.

Дисплей для відображення інформації про стан та налаштування термостата. Кнопки для управління та налаштування параметрів.

Блок живлення включає в себе всі компоненти, які забезпечують живлення для всіх електронних частин термостата, включаючи мікроконтролер, сенсори та нагрівальні/охолоджувальні елементи. Ця структурна схема дозволяє термостату ефективно керувати температурою та вологістю у медичному лабораторії, забезпечуючи стабільні умови для експериментів та дослідів.

					ІКБІ.480 004.005 - ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

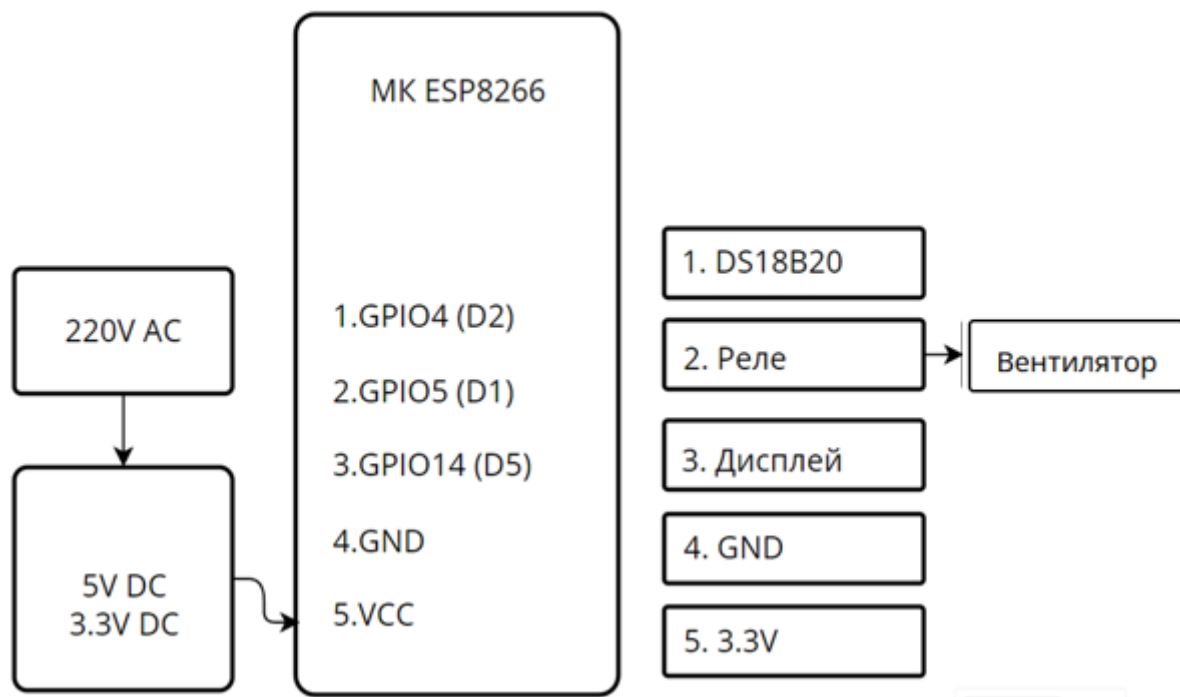


Рисунок 3.3 - Структурна схеми медичного лабораторного smart – термостата з мікроконтролером ESP

3.3 Опис архітектури програмного коду

Головний цикл виконує ініціалізацію системи, запуск основних підсистем та безперервний моніторинг і керування термостатом. Він включає виклики до інших модулів для обробки датчиків, оновлення інтерфейсу користувача та обміну даними.

Модуль ініціалізації відповідає за початкову настройку мікроконтролера та всіх підключених компонентів, таких як сенсори температури, вологості, інтерфейс користувача та мережеві модулі.

Модуль обробки датчиків забезпечує збір даних з різних сенсорів, таких як сенсори температури та вологості. Він обробляє та перетворює отримані дані у формат, зручний для подальшого використання іншими модулями.

Модуль керування нагріванням та охолодженням на основі даних, отриманих від модуля обробки сенсорів, керує нагрівальними та охолоджувальними елементами для підтримання потрібної температури.

Модуль інтерфейсу користувача відповідає за відображення інформації на дисплеї та обробку введення від користувача через кнопки. Він забезпечує взаємодію користувача з пристроєм.

Модуль мережевої взаємодії забезпечує підключення до Wi-Fi та взаємодію з віддаленими серверами або хмарними сервісами для моніторингу та керування термостатом дистанційно.

Модуль збереження даних відповідає за збереження історії даних з сенсорів, налаштувань користувача та подій системи у внутрішню пам'ять або на зовнішній накопичувач.

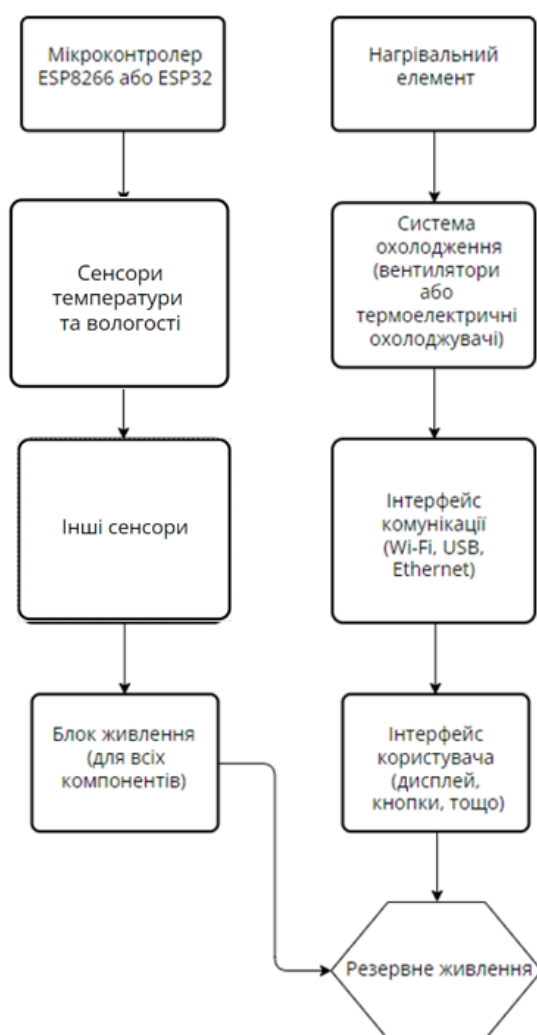


Рисунок 3.4 - Структурна схема архітектури медичного лабораторного smart – термостата з мікроконтролером ESP

3.3.1 Логічна структура коду

Головний цикл (Main Loop) включає виклик модуля ініціалізації, модуля обробки сенсорів, модуля керування нагріванням та охолодженням, модуля інтерфейсу користувача, модуля мережевої взаємодії та модуля збереження даних.

Модуль ініціалізації включає налаштування мікроконтролера, ініціалізацію сенсорів, ініціалізацію інтерфейсу користувача та налаштування мережевих підключень.

Модуль обробки сенсорів забезпечує збір даних з температурних сенсорів, збір даних з датчиків вологості, обробку та фільтрацію даних, а також передачу даних до головного циклу.

Модуль керування нагріванням та охолодженням включає аналіз даних з сенсорів, визначення необхідності нагрівання або охолодження, активацію нагрівальних або охолоджувальних елементів та моніторинг стану нагрівальних та охолоджувальних елементів.

Модуль інтерфейсу користувача включає відображення температури та вологості, обробку введення від користувача, налаштування параметрів роботи термостата та оновлення дисплея.

Модуль мережевої взаємодії забезпечує підключення до Wi-Fi, обмін даними з віддаленими серверами, оновлення програмного забезпечення та відправку даних на хмарні сервіси.

Модуль збереження даних включає запис даних з сенсорів, збереження налаштувань користувача, архівування подій системи та передачу збережених даних при запиті.

3.4 Структура та алгоритм роботи Telegram бота

Telegram бот для медичного лабораторного smart-термостата на базі мікроконтролера ESP забезпечує віддалене керування та моніторинг термостата

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

через інтерфейс Telegram. Нижче наведено детальний опис структури та алгоритмів роботи цього бота.

Головний модуль (Main Module) відповідає за ініціалізацію бота, обробку вхідних повідомлень та взаємодію з іншими модулями. Модуль команд (Commands Module) займається обробкою команд користувача та виконанням відповідних дій. Модуль зв'язку з мікроконтролером (ESP Communication Module) забезпечує обмін даними з ESP та надсилання команд до мікроконтролера. Модуль обробки даних (Data Processing Module) обробляє дані, отримані від ESP, та готує їх для відправки користувачу. Модуль збереження даних (Data Storage Module) зберігає історію даних та налаштувань. Модуль інтерфейсу користувача (User Interface Module) формує повідомлення для користувача та відправляє їх.

3.4.1 Алгоритми роботи бота

1. Ініціалізація бота. Спочатку запускається головний модуль, який ініціалізує Telegram API та отримує ключ доступу. Після цього ініціалізуються інші модулі: команд, зв'язку з ESP, обробки даних, збереження даних та інтерфейсу користувача.
2. Обробка вхідних повідомлень. Головний модуль отримує повідомлення від користувача через Telegram API. Перевіряється тип повідомлення (текстове повідомлення, команда, фото, документ тощо) і передається до модуля команд для обробки.
3. Обробка команд користувача. Модуль команд отримує повідомлення від головного модуля. Перевіряється команда користувача (наприклад, /start, /status, /set_temp) та виконуються відповідні дії. Команда /start відправляє привітальне повідомлення та інструкції, /status викликає функцію отримання поточного стану термостата, а /set_temp зберігає нову температуру та передає команду на ESP.

					ІКБІ.480 004.005 - ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Обмін даними з ESP. Модуль зв'язку з ESP отримує запит на обмін даними (наприклад, отримання поточного стану термостата), надсилає відповідний запит до ESP через Wi-Fi, отримує відповідь від ESP та передає дані до модуля обробки даних.
5. Обробка даних. Модуль обробки даних отримує дані від модуля зв'язку з ESP, обробляє їх (наприклад, формування читабельного формату температури та вологості) та передає оброблені дані до модуля інтерфейсу користувача для відправки користувачу.
6. Формування та відправка повідомлень користувачу. Модуль інтерфейсу користувача отримує оброблені дані або відповідь на команду, формує повідомлення для користувача (наприклад, поточна температура, налаштування успішно змінені) та відправляє повідомлення користувачу через Telegram API.
7. Збереження даних. Модуль збереження даних отримує дані для збереження (наприклад, історію температур або налаштування користувача), зберігає дані у внутрішню пам'ять або на зовнішній накопичувач та надає доступ до збережених даних для інших модулів при необхідності

3.4.2 Взаємодія модулів

Головний модуль виступає центральним вузлом, координуючи взаємодію між модулями. Модуль команд взаємодіє з усіма іншими модулями для виконання команд користувача. Модуль зв'язку з ESP забезпечує безперервний зв'язок з мікроконтролером. Модуль обробки даних отримує та обробляє дані від ESP та модулю команд. Модуль інтерфейсу користувача забезпечує взаємодію користувача з ботом через повідомлення. Модуль збереження даних зберігає та надає доступ до історичних даних та налаштувань.

Ця структура та алгоритми забезпечують ефективну роботу Telegram бота для віддаленого керування та моніторингу медичного лабораторного smart-термостата на базі мікроконтролера ESP.

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

РОЗДІЛ 4

СИНТЕЗ І РОЗРАХУНОК ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ СХЕМИ

Синтез та розрахунок електричної принципової схеми проводиться у відповідності з блоками структурної та функціональної схем, розроблених у розділі 3.

4.1 Синтез та розрахунок електричної принципової схеми блоку живлення

Розрахунки для блоку живлення медичного лабораторного smart-термостата. Для виконання розрахунків необхідно враховувати кілька факторів, таких як потужність споживання кожного компонента, напруга на виході кожного блоку, а також ефективність перетворення енергії. Наведемо розрахунки для кожного блоку.

Вхідний фільтр

Мета: Захист від високочастотних перешкод і імпульсних завад.

Розрахунок: Вибір конденсаторів та індуктивностей фільтра, здатних витримати напругу 220 В.

Вхідна напруга: ~ 220 В (змінний струм)

Неперервність струму у контактній мережі при транзисторному регулюванні навантаження забезпечується Г-подібним фільтром $L_{\phi}C_{\phi}$. Незалежно від стану поривача, струм у контактній мережі має контур для замикання: по ланцюгу тягових двигунів або по ланцюгу фільтруючого конденсатора. У інтервалі $1 - \gamma$, не залежно від того, що тягові двигуни відключені від контактної мережі, відбувається споживання енергії від джерела живлення, яка не витрачається, а накопичується у фільтруючому конденсаторі.

					ІКБІ.480 004.005 - ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

У інтервалі періоду γ в ланцюзі тягових двигунів поступає енергія як від джерела живлення, так і від фільтруючого конденсатора. Шляхом підбору параметрів фільтра $L_{\phi}C_{\phi}$ максимальні пульсації струму обмежуються на рівні 1-2 % струму $I_{\text{ср}}$, так влюбий момент часу значення струму у контактній мережі незначно відрізняється від середнього $I_{\text{ср}}$.

Фільтр повинен витримувати струм до 1 А (резерв).

Трансформатор

Мета: Зниження напруги з 220 В до необхідного рівня для випрямлення.

Розрахунок: Визначення коефіцієнта трансформації та потужності трансформатора.

Вхідна напруга: 220 В АС

Вихідна напруга: 24 В АС

Потужність: $P = U \cdot I$, де U - вихідна напруга, I - струм.

Якщо загальне споживання складає 20 Вт, то струм становитиме

$$I = \frac{20V}{24V} = 0.833 \text{ А}$$

Випрямляч

Мета: Перетворення змінного струму на постійний.

Розрахунок: Вибір діодів із відповідними параметрами.

Вихідна напруга після випрямлення: 24 В DC

Струм через діоди: 0.833 А

Вибір діодів: діоди повинні витримувати зворотну напругу щонайменше 35 В і прямий струм понад 1 А.

Первинний фільтр

Мета: Згладжування пульсацій після випрямляча.

Розрахунок: Вибір конденсатора для фільтрації.

Розрахунок ємності: $C = \frac{I}{f \cdot \Delta V}$, де I - струм, f - частота пульсацій (100 Гц після

повнохвильового випрямлення), ΔV - допустиме рябіння.

Для $\Delta V = 1 \text{ В}$,

$$C = \frac{0.833A}{100\text{Гц} \cdot 1V} = 8330 \text{ мкФ}.$$

Регулятор напруги

Мета: Стабілізація вихідної напруги.

Розрахунок: Вибір стабілізатора напруги.

Припустимо, потрібна стабілізована напруга 12 В.

Вибір стабілізатора: 7812 для 12 В із струмом до 1 А.

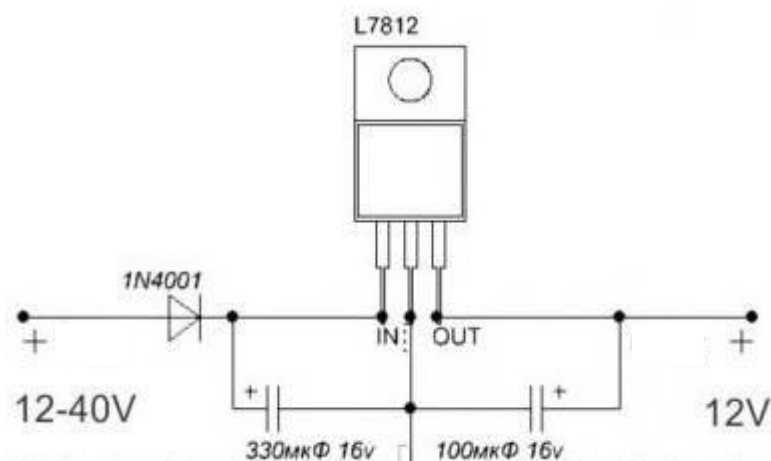


Рисунок 4.1 - Лінійний стабілізатор L7812CV

Захисні компоненти

Мета: Захист від перенапруги та перевантаження.

Розрахунок: Вибір запобіжників та варисторів.

Запобіжник на 1 А.

Варистор із номіналом 14 В для захисту стабілізатора.

Для захисту від перенапруги та перевантаження в електричних системах зазвичай використовують запобіжники та варистори. Запобіжники призначені для захисту від перевантаження та короткого замикання, вони переривають електричне коло, коли струм перевищує допустиме значення.

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Варистори ж використовуються для захисту від перенапруги шляхом поглинання надлишкової енергії.

Розрахунок захисних компонентів: Запобіжник на 1 А:

Запобіжник обирається на основі максимального допустимого струму, який може пройти через систему без пошкодження обладнання. Якщо максимальний струм, який може пройти через систему, менше 1 А, то застосовується запобіжник на 1 А, оскільки він захистить систему від перевантаження та короткого замикання.

Варистор із номіналом 14 В для захисту стабілізатора:

Варистор обирається на основі максимальної напруги, яку він повинен захищати. Варистор пропускає струм тільки тоді, коли напруга на ньому перевищує його номінальну напругу. У вашому випадку, вибираємо варистор із номінальною напругою 14 В для захисту стабілізатора.

Пояснення розрахунків: Запобіжник на 1 А:

Розрахунок запобіжника зазвичай базується на допустимому струмі обладнання або електричної системи. Якщо максимальний струм, який може пройти через систему, менше 1 А, то застосовується запобіжник на 1 А, оскільки він не допустить перевантаження та коротке замикання.

Розрахунок варистора відбувається на основі максимальної напруги, яку він повинен захищати. Якщо напруга в системі перевищує номінальне значення варистора (у випадку 14 В), варистор переходить в режим провідності і поглинає надлишкову енергію, захищаючи таким чином обладнання від перенапруги.

Вторинний фільтр

Мета: Згладжування залишкових пульсацій після стабілізатора.

Розрахунок: Вибір конденсатора для фільтрації.

Одним із ключових компонентів вторинного фільтра є конденсатор. Його ємність визначає ефективність фільтрації пульсацій.

Для розрахунку необхідної ємності конденсатора можна використовувати наступну формулу:

$$C = \frac{I \cdot t}{V_{pp} \cdot \Delta V}$$

де:

- C - ємність конденсатора (в мікрофарадах, мкФ)
- I - струм навантаження (в амперах, А)
- t - період пульсацій (в секундах, с)
- V_{pp} - пікове значення пульсацій (в вольтах, В)
- ΔV - допустиме значення пульсацій на виході (в вольтах, В)

Вибір конденсатора

Важливо вибрати конденсатор з відповідними характеристиками, такими як:

- Тип: Електролітичні конденсатори найчастіше використовуються у вторинних фільтрах.
- Робоча напруга: Напруга конденсатора повинна бути не меншою, ніж максимальна напруга на виході стабілізатора.
- Ємність: Як вже згадувалося, рекомендується використовувати конденсатор з більшою ємністю, ніж розрахункове значення.
- Форм-фактор: Конденсатори доступні в різних форм-факторах, таких як радіальний та осьовий. Вибір форм-фактора залежить від доступного місця на платі.

Додаткові міркування

Окрім конденсатора, у вторинному фільтрі можуть використовуватися й інші компоненти, такі як індуктивні котушки та резистори. Ці компоненти можуть допомогти в подальшому згладжуванні пульсацій та покращити загальну стабільність живлення.

Важливо правильно підібрати та розмістити всі компоненти вторинного фільтра, щоб забезпечити його ефективну роботу. Вторинний фільтр є важливим компонентом джерела живлення, який допомагає отримати чисте та стійке живлення для чутливого електронного обладнання.

					ІКБІ.480 004.005 - ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Розрахунок ємності конденсатора та вибір відповідних компонентів є ключовими факторами для забезпечення ефективної роботи фільтра.

Ємність фільтра: 1000 мкФ на 25 В.

DC/DC перетворювач

Мета: Перетворення 12 В на необхідні 5 В та 3.3 В.

Розрахунок: Вибір перетворювача.

Вхідна напруга: 12 В DC

Вихідна напруга: 5 В та 3.3 В

Струм навантаження: до 1 А для кожного каналу.

Перетворювачі: наприклад, LM2596 для 5 В та 3.3 В.

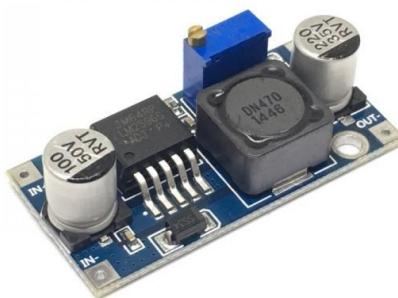


Рисунок 4.2 - LM2596

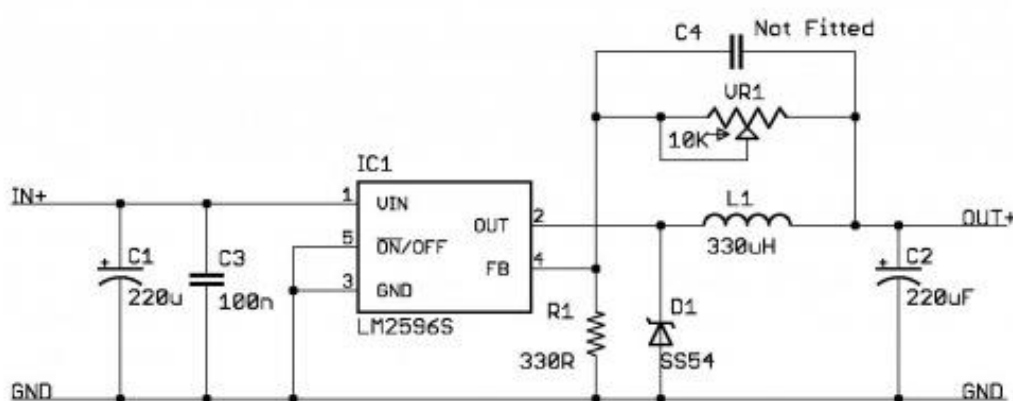


Рисунок 4.3 - Схема перетворювача DC-DC на LM2596

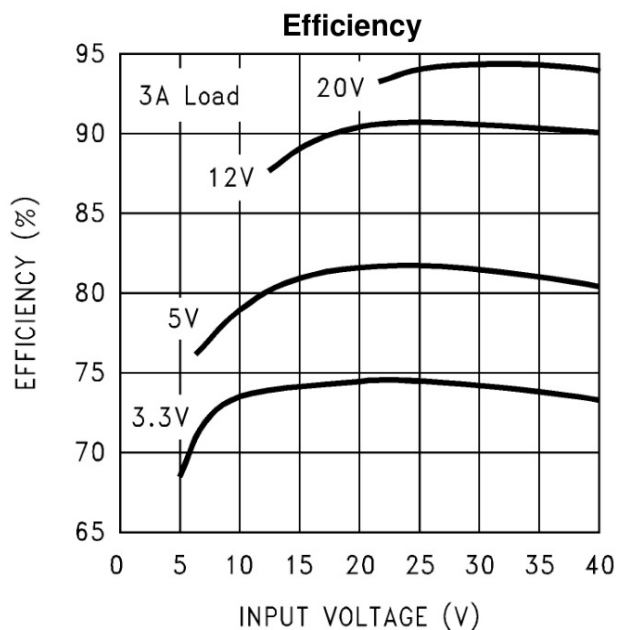


Рисунок 4.4 – Графік стабілізації напруги перетворювача DC-DC на LM2596

5V стабілізатор

Мета: Стабілізація 5 В для мікроконтролера.

Розрахунок: Вибір стабілізатора.

Вибір стабілізатора: 7805 із струмом до 1 А.

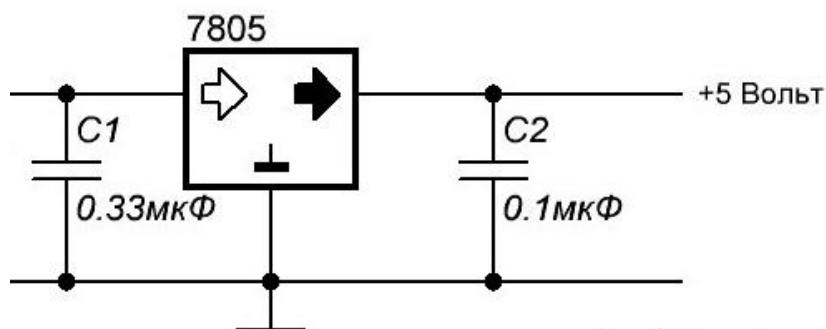


Рисунок 4.5 - Схема стабілізатора 7805

3.3V стабілізатор

Мета: Стабілізація 3.3 В для датчиків.

Розрахунок: Вибір стабілізатора.

Вибір стабілізатора: LD1117 для 3.3 В із струмом до 1 А.



Рисунок 4.6 - Схема стабілізатора LD1117

Блок живлення повинен бути розрахований на загальне споживання всіх компонентів із достатнім запасом.



Рисунок 4.7 - Схема блоку живлення на LM7805

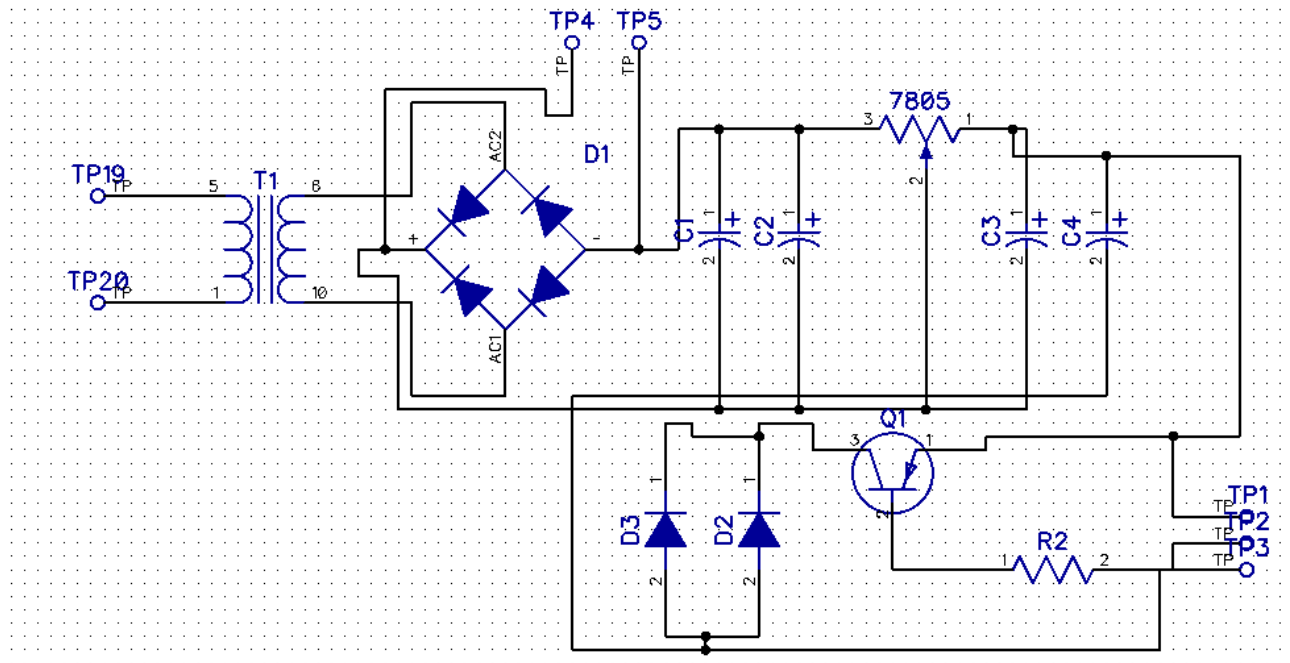


Рисунок 4.8 - Електрична принципова схема блоку живлення

4.2 Синтез та розрахунок електричної принципової схеми медичного лабораторного smart-термостата з мікроконтролером ESP

Створення електричної принципової схеми медичного лабораторного smart-термостата на базі мікроконтролера ESP вимагає розробки кількох основних компонентів. Нижче наведено опис ключових елементів та їх взаємозв'язків у схемі.

Основні компоненти:

Мікроконтролер ESP (ESP8266 або ESP32), сенсор температури DS18B20, нагрівальний елемент ТЕН, реле або транзистор для керування нагрівальним елементом, блок живлення, елементи захисту (запобіжники, діоди), інтерфейс користувача (кнопки, дисплей), схема підключення:

Підключення мікроконтролера ESP:

VCC мікроконтролера підключається до позитивного виходу блоку живлення (3.3V або 5V в залежності від моделі ESP).

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

GND мікроконтролера підключається до негативного виходу блоку живлення. GPIO-піни мікроконтролера використовуються для підключення сенсорів, реле, дисплея та кнопок. Підключення датчика температури.

VCC датчика температури підключається до 3.3V. GND сенсора температури підключається до GND. Data вивід сенсора підключається до одного з GPIO-пінів ESP (наприклад, GPIO4), через резистор підтягується до 3.3V (зазвичай 4.7 кОм). Підключення реле для нагрівального елемента.

VCC реле підключається до 5V. GND реле підключається до GND. IN реле підключається до GPIO мікроконтролера (наприклад, GPIO5). NO (Normally Open) контакт реле підключається до одного кінця нагрівального елемента. Інший кінець нагрівального елемента підключається до живлення (наприклад, 220V). Блок живлення забезпечує стабільне живлення для всіх компонентів (5V для реле, 3.3V для ESP і сенсорів). Захист та інтерфейс користувача:

Запобіжники та діоди встановлюються для захисту від перенапруги. Кнопки підключаються до GPIO-пінів для керування термостатом вручну. Дисплей LCD підключається до I2C або SPI інтерфейсу мікроконтролера для відображення температури та інших параметрів.

Сенсор температури DS18B20 є популярним вибором для багатьох застосувань завдяки його точності, надійності та простоті інтеграції. Ось деякі ключові характеристики та переваги цього сенсора. Ключові характеристики DS18B20: тип сенсора - цифровий сенсор температури; інтерфейс - працює через інтерфейс 1-Wire, що дозволяє використовувати лише одну лінію даних для з'єднання декількох датчиків з мікроконтролером; діапазон вимірювань - від -55°C до $+125^{\circ}\text{C}$; точність - $\pm 0.5^{\circ}\text{C}$ в діапазоні від -10°C до $+85^{\circ}\text{C}$; роздільна здатність - налаштовується від 9 до 12 біт; живлення - напруга живлення 3.0V - 5.5V; ідентифікація - кожен сенсор має унікальний 64-бітний серійний номер, що дозволяє легко ідентифікувати кожен датчик у мережі. для інтеграції цього датчика в систему: підключення до мікроконтролера - підключіть лінію

					ІКБІ.480 004.005 - ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

даних сенсора до мікроконтролера через резистор підтягування (зазвичай 4.7 кОм); читання даних - використовуйте бібліотеки для роботи з 1-Wire інтерфейсом, наприклад, для Arduino є бібліотека DallasTemperature.

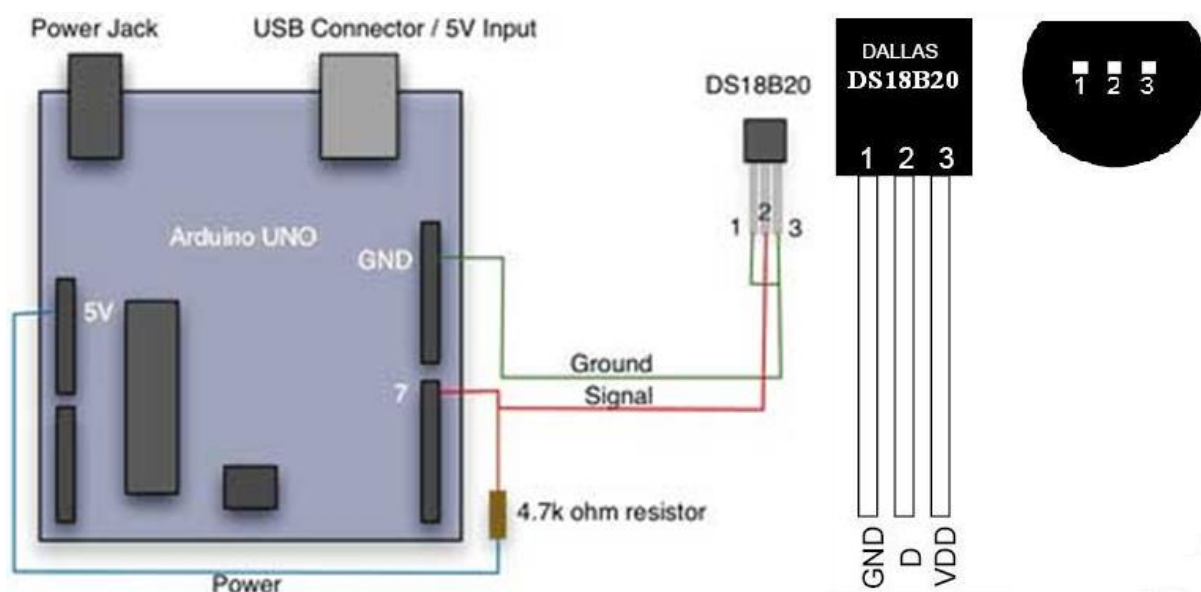


Рисунок 4.9 – Схема підключення DS18B20

Початковий код (Arduino IDE)

```
#include <ESP8266WiFi.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#define ONE_WIRE_BUS 2

#define RELAY_PIN 5

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {

    pinMode(RELAY_PIN, OUTPUT);

    digitalWrite(RELAY_PIN, LOW);

    sensors.begin();

    lcd.begin();

    lcd.backlight();

}
```

					ІКБІ.480 004.005 - ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void loop() {

    sensors.requestTemperatures();

    float tempC = sensors.getTempCByIndex(0);

    lcd.setCursor(0, 0);

    lcd.print("Temp: ");

    lcd.print(tempC);

    lcd.print(" C");

    if (tempC < 37.0) {

        digitalWrite(RELAY_PIN, HIGH); // Turn on heating

        lcd.setCursor(0, 1);

        lcd.print("Heating ON ");

    } else {

        digitalWrite(RELAY_PIN, LOW); // Turn off heating

        lcd.setCursor(0, 1);

        lcd.print("Heating OFF");

    }

    delay(1000);

}

```

Ця схема та код є основою для реалізації smart-термостата. В реальних умовах можна додати додаткові функції, такі як підключення до мережі Wi-Fi для віддаленого моніторингу та керування, або використання більш складних датчиків та елементів управління.

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

4.4 Схема роботи Telegram бота

Початкова конфігурація. Реєстрація Telegram-бота: створіть нового бота в Telegram через BotFather та отримайте токен доступу. Конфігурація ESP: прошийте мікроконтролер ESP кодом, що дозволяє йому з'єднуватися з Wi-Fi мережею та взаємодіяти з Telegram API. Взаємодія з Telegram-ботом. Користувач надсилає команду або запит через Telegram до бота. Telegram API приймає запит та пересилає його на сервер або безпосередньо на мікроконтролер ESP (якщо використовується WebSocket або подібний метод).

Обробка запиту. Сервер або ESP отримує запит від Telegram-бота. Код на ESP перевіряє команду та виконує відповідні дії: зчитування температури з датчика, керування нагрівальним елементом через реле (вмикання або вимикання), відправка даних (наприклад, поточної температури) назад до користувача через Telegram.

Відправка відповіді. ESP або сервер відправляє відповідь назад через Telegram API. Telegram-бот пересилає відповідь користувачу.

4.4.1 Код для ESP з використанням бібліотеки UniversalTelegramBot

```
#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#include <ArduinoJson.h>

#include <OneWire.h>

#include <DallasTemperature.h>

// Wi-Fi налаштування

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

					ІКБІ.480 004.005 - ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const char* password = "YOUR_PASSWORD";

// Telegram налаштування
const char* botToken = "YOUR_BOT_TOKEN";
const char* chat_id = "YOUR_CHAT_ID";

// Підключення до Wi-Fi
WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);

OneWire oneWire(D2);
DallasTemperature sensors(&oneWire);

const int relayPin = D1;

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
    Serial.println("Connected to WiFi");
}

```

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

sensors.begin();

pinMode(relayPin, OUTPUT);

digitalWrite(relayPin, LOW);
}

void loop() {

    if (WiFi.status() == WL_CONNECTED) {

        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        while (numNewMessages) {

            Serial.println("got response");

            handleNewMessages(numNewMessages);

            numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        }

    }

}

void handleNewMessages(int numNewMessages) {

    for (int i = 0; i < numNewMessages; i++) {

        String chat_id = String(bot.messages[i].chat_id);

        String text = bot.messages[i].text;

        if (text == "/start") {

            String welcome = "Welcome to Smart Thermostat Bot";

```

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

        bot.sendMessage(chat_id, welcome, "");
    }

    if (text == "/temperature") {

        sensors.requestTemperatures();

        float temperature = sensors.getTempCByIndex(0);

        String temp = "Current temperature: " + String(temperature) + " °C";

        bot.sendMessage(chat_id, temp, "");

    }

    if (text == "/heat_on") {

        digitalWrite(relayPin, HIGH);

        bot.sendMessage(chat_id, "Heating is now ON", "");

    }

    if (text == "/heat_off") {

        digitalWrite(relayPin, LOW);

        bot.sendMessage(chat_id, "Heating is now OFF", "");

    }

}

}

```

Ключові команди бота

1. /start - початкова команда для привітання користувача.
2. /temperature - команда для отримання поточної температури.

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

3. /heat_on - команда для увімкнення нагрівального елемента.

4. /heat_off - команда для вимкнення нагрівального елемента.

Ця схема роботи забезпечує простий та ефективний спосіб керування медичним лабораторним smart-термостатом через Telegram.

Для реалізації таймерів на ESP з використанням функціоналу Arduino можна використовувати бібліотеку Ticker. Вона дозволяє легко налаштовувати періодичні або одноразові виклики функцій. Нижче наведено приклад коду для роботи з таймером в контексті Telegram-бота для керування медичним лабораторним smart-термостатом.

4.4.2 Код програми з використанням таймера

Цей код включає функціонал для періодичного зчитування температури та відправки її через Telegram-бота.

```
#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#include <ArduinoJson.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include <Ticker.h>

// Wi-Fi налаштування

const char* ssid = "YOUR_SSID";

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

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

```

// Telegram налаштування

const char* botToken = "YOUR_BOT_TOKEN";

const char* chat_id = "YOUR_CHAT_ID";


// Підключення до Wi-Fi

WiFiClientSecure client;

UniversalTelegramBot bot(botToken, client);


OneWire oneWire(D2);

DallasTemperature sensors(&oneWire);


const int relayPin = D1;

Ticker temperatureTicker;

Ticker botTicker;


void setup() {

    Serial.begin(115200);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        delay(1000);

        Serial.println("Connecting to WiFi...");

    }

    Serial.println("Connected to WiFi");

```

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

sensors.begin();

pinMode(relayPin, OUTPUT);

digitalWrite(relayPin, LOW);


// Налаштування таймерів

temperatureTicker.attach(30, readAndSendTemperature); // Викликати
кожні 30 секунд

botTicker.attach(2, handleBot); // Перевіряти повідомлення кожні 2
секунди
}

void loop() {

    // Основний цикл порожній, оскільки все виконується у таймерах
}

void readAndSendTemperature() {

    sensors.requestTemperatures();

    float temperature = sensors.getTempCByIndex(0);

    String temp = "Current temperature: " + String(temperature) + " °C";

    bot.sendMessage(chat_id, temp, "");
}

void handleBot() {

    if (WiFi.status() == WL_CONNECTED) {

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

```

int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

while (numNewMessages) {

    Serial.println("got response");

    for (int i = 0; i < numNewMessages; i++) {

        String chat_id = String(bot.messages[i].chat_id);

        String text = bot.messages[i].text;

        if (text == "/start") {

            String welcome = "Welcome to Smart Thermostat Bot";

            bot.sendMessage(chat_id, welcome, "");

        }

        if (text == "/temperature") {

            sensors.requestTemperatures();

            float temperature = sensors.getTempCByIndex(0);

            String temp = "Current temperature: " + String(temperature) + "
°C";

            bot.sendMessage(chat_id, temp, "");

        }

        if (text == "/heat_on") {

            digitalWrite(relayPin, HIGH);

```

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

```

        bot.sendMessage(chat_id, "Heating is now ON", "");
    }

    if (text == "/heat_off") {

digitalWrite(relayPin, LOW);

        bot.sendMessage(chat_id, "Heating is now OFF", "");
    }
}

numNewMessages = bot.getUpdates(bot.last_message_received + 1);

}

}
}

```

1. Бібліотеки та змінні: Імпорт необхідних бібліотек та оголошення змінних для Wi-Fi, Telegram, датчика температури та реле.
2. Функція setup: Підключення до Wi-Fi, ініціалізація сенсора температури та реле, налаштування таймерів.
3. Функція loop: Основний цикл порожній, оскільки всі дії виконуються в таймерах.
4. Функція readAndSendTemperature: Зчитування температури та відправка її через Telegram кожні 30 секунд.
5. Функція handleBot: Перевірка нових повідомлень від Telegram-бота кожні 2 секунди та обробка команд.

Цей код дозволяє автоматично зчитувати та відправляти температуру кожні 30 секунд, а також регулярно перевіряти повідомлення від користувачів, забезпечуючи ефективне дистанційне керування термостатом.

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

РОЗДІЛ 5

КОНСТРУКТОРСЬКА ЧАСТИНА

5.1 Загальна конструкція приладу

В цьому розділі розробляється конструкція пристрою в цілому.

Загальна конструкція приладу медичного лабораторного smart-термостата з мікроконтролером ESP з керуванням через Telegram-бот включає кілька основних модулів та елементів, кожен з яких відповідає за певні функції. Нижче наведено докладний опис цих модулів та їх розташування в конструкції приладу.

Корпус приладу виготовлений з міцного та термостійкого матеріалу. Внутрішня частина корпусу розділена на кілька секцій для розміщення електронних компонентів, датчиків, нагрівального елемента та інтерфейсу користувача.

Мікроконтролер ESP (ESP8266 або ESP32) розміщується на друкованій платі, яка закріплюється всередині корпусу. Він підключений до блоку живлення, датчиків, реле та інших периферійних компонентів.

Мікроконтролер ESP (ESP8266 або ESP32)

- Розміщується на друкованій платі, яка закріплюється всередині корпусу.
- Підключений до блоку живлення, датчиків, реле та інших периферійних компонентів.

Сенсор температури DS18B20

- Встановлюється в зоні, де необхідно вимірювати температуру.
- Підключений до мікроконтролера за допомогою відповідних проводів.

					ІКБІ.480 004.005 - ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Нагрівальний елемент ТЕН

- Закріплений у відсіку, який забезпечує рівномірне нагрівання.
- Підключений до реле або транзистора, який керує його роботою.

Реле або транзистор

- Використовується для керування нагрівальним елементом.
- Розташований на друкованій платі поруч з мікроконтролером.
- Підключений до відповідних GPIO-пінів мікроконтролера.

Блок живлення

- Забезпечує стабільне живлення для всіх компонентів (3.3V для ESP і датчиків, 5V для реле).
- Розташований в окремому відсіку корпусу для мінімізації електромагнітних перешкод.

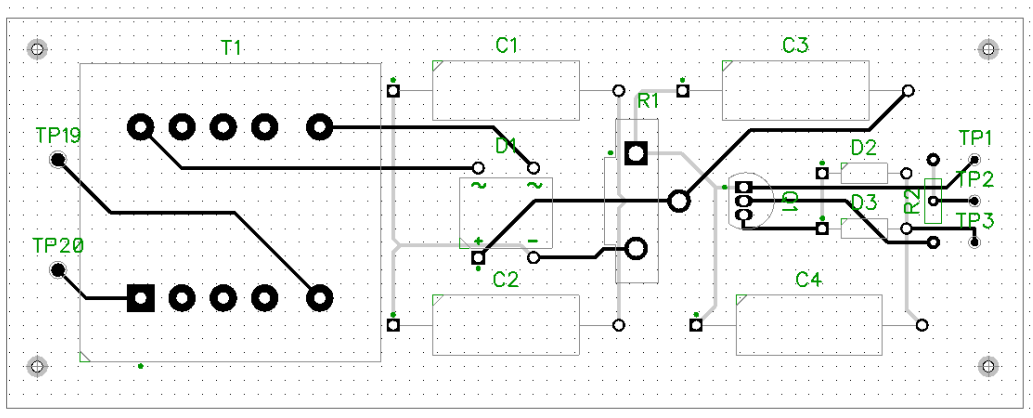


Рисунок 5.1 – Друкована плата блоку живлення

Елементи захисту (запобіжники, діоди)

- Встановлені на друкованій платі для захисту від перенапруги та зворотної полярності.
- Розташовані поруч з відповідними компонентами.

Інтерфейс користувача (кнопки, дисплей)

- Дисплей LCD встановлений на передній панелі корпусу для зручного перегляду інформації.
- Кнопки для ручного керування термостатом також розташовані на передній панелі.
- Підключені до GPIO-пінів мікроконтролера для обробки команд.

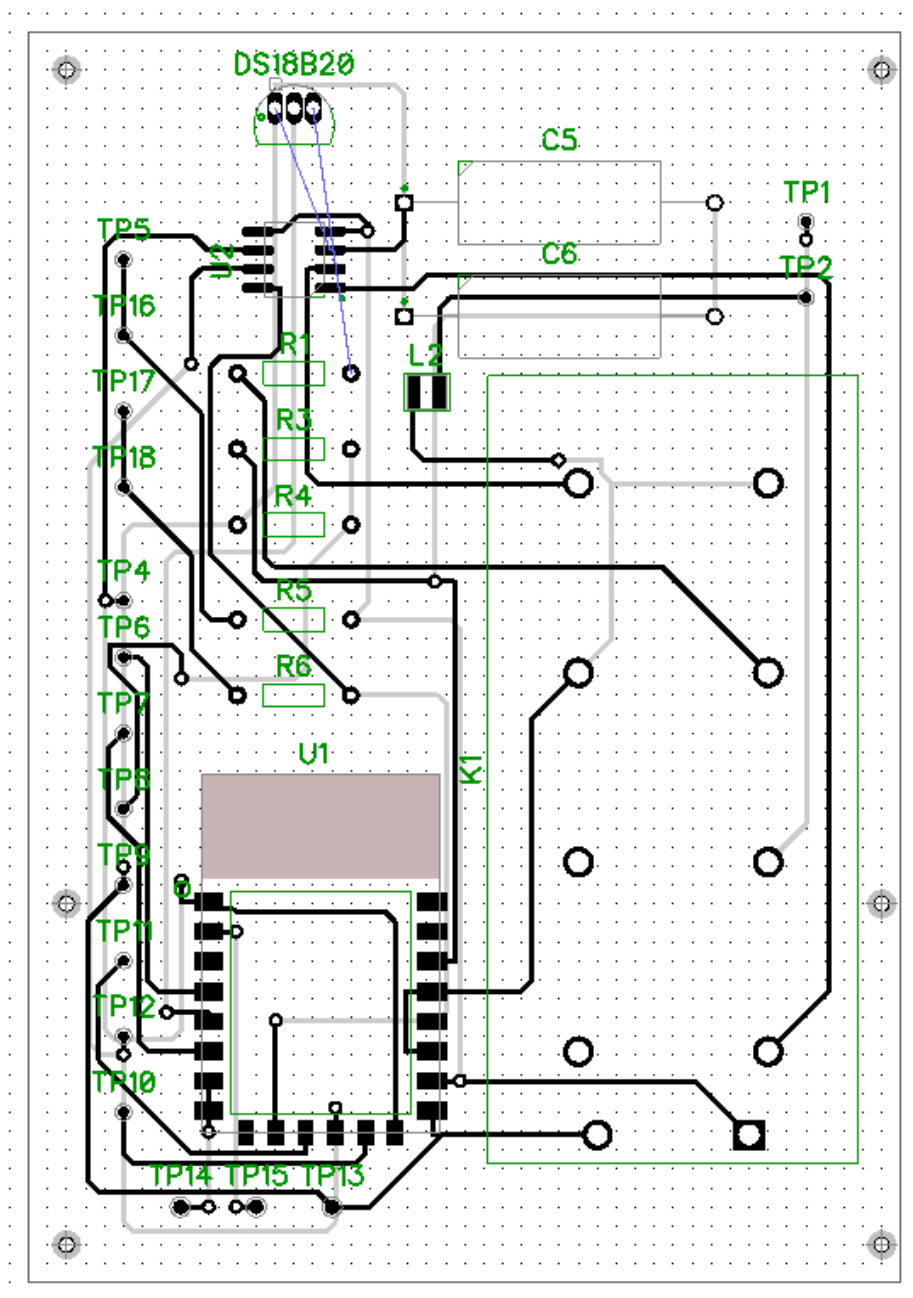
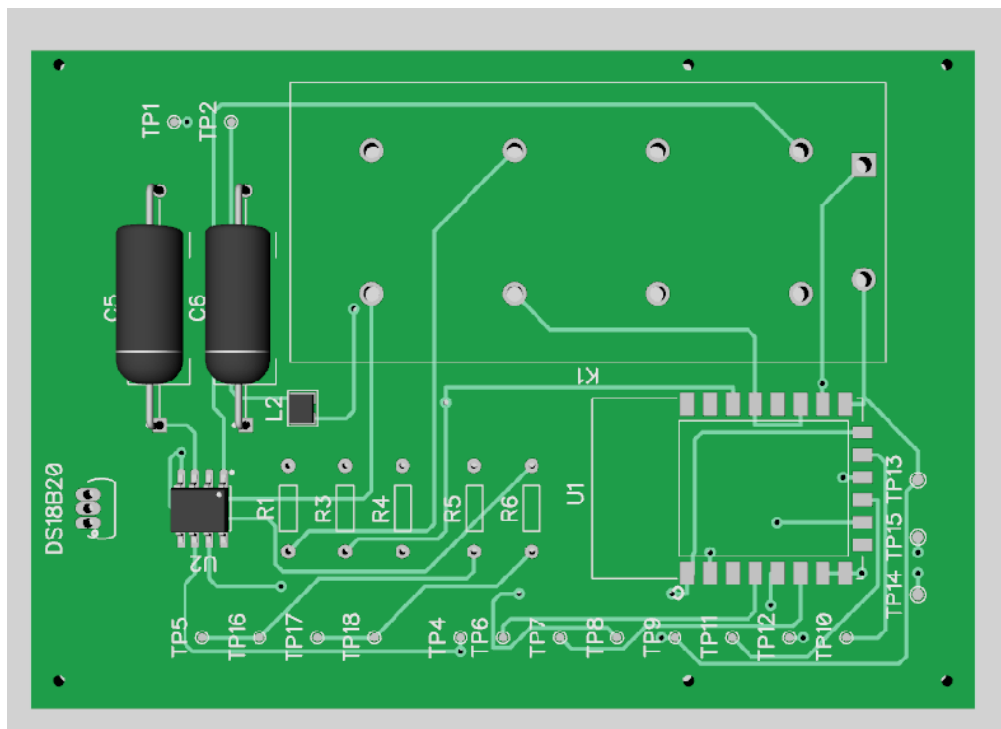
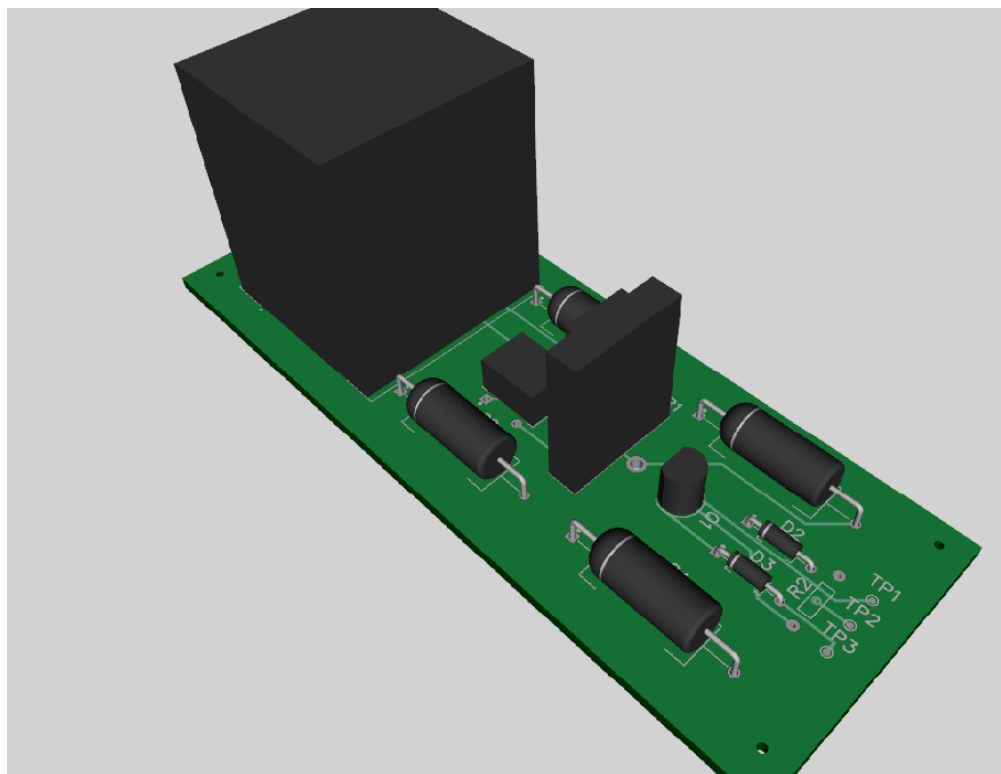


Рисунок 5.2 – Друкована плата основної частини медичного лабораторно smart - термостата



1



2

Рисунок 5.3 – 3D моделі друкованих плат. 1 - основної частини медичного лабораторно smart – термостата, 2 - блоку живлення.

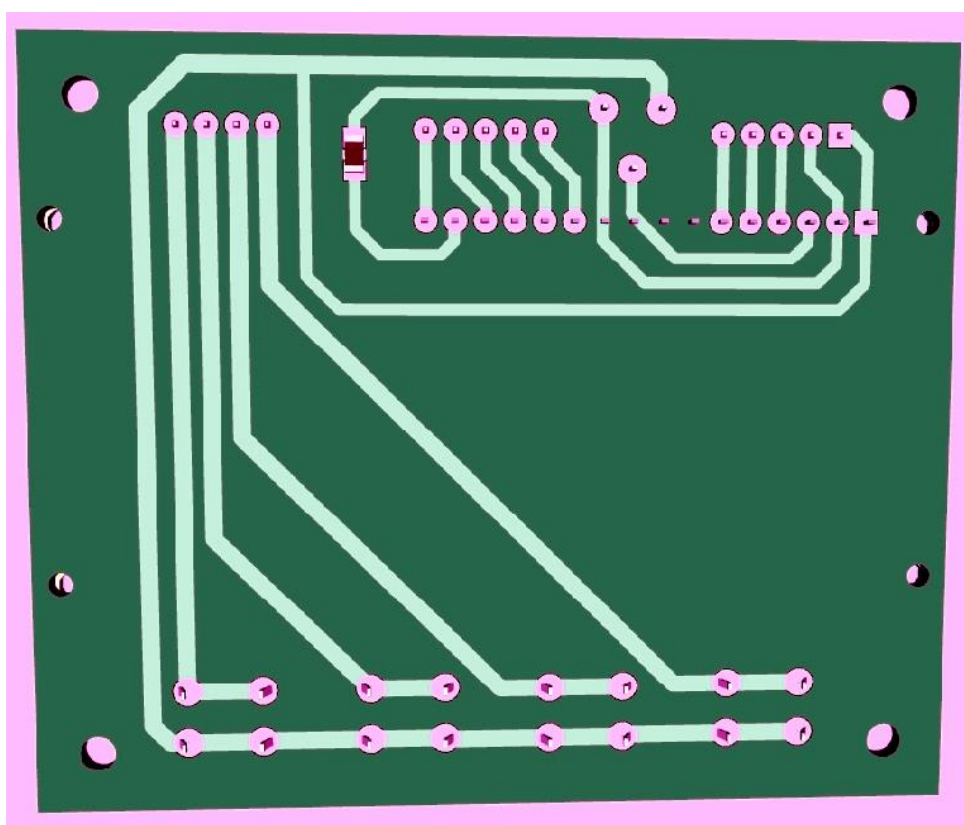
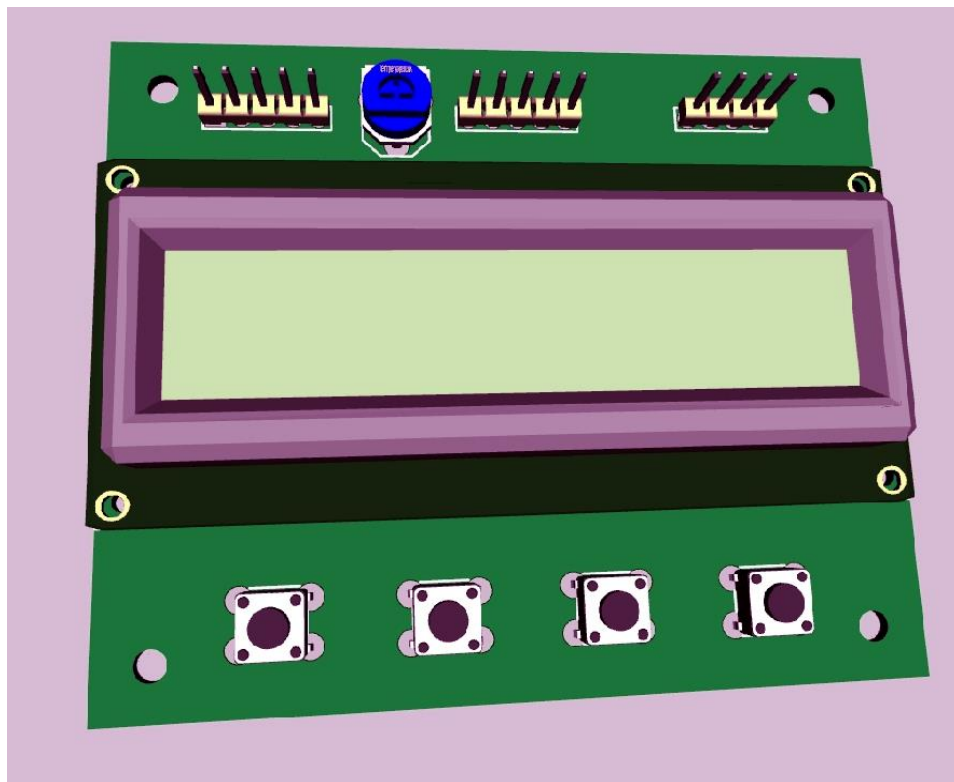


Рисунок 5.4 – 3D модель друкованої плати передньої панелі керування

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

5.1.1 Розташування компонентів

Передня панель містить дисплей для відображення температури та налаштувань і кнопки для керування приладом (вмикання/вимикання, зміна температури).

Внутрішня частина корпусу містить друковану плату з мікроконтролером ESP, реле, елементами захисту та іншими електронними компонентами, блок живлення в окремому відсіку для мінімізації перешкод, сенсор температури закріплений у місці, де вимірюється температура, і нагрівальний елемент, встановлений у спеціальному відсіку для рівномірного нагрівання.

Задня панель містить роз'єми для підключення живлення і отвори для вентиляції та охолодження компонентів.

Всі компоненти підключені до блоку живлення відповідно до їх вимог (3.3V або 5V). Внутрішні з'єднання між мікроконтролером, датчиком температури, реле та інтерфейсом користувача здійснюються за допомогою проводів та роз'ємів для забезпечення надійного контакту. Використовується Wi-Fi для підключення до мережі та взаємодії з Telegram для дистанційного керування термостатом.

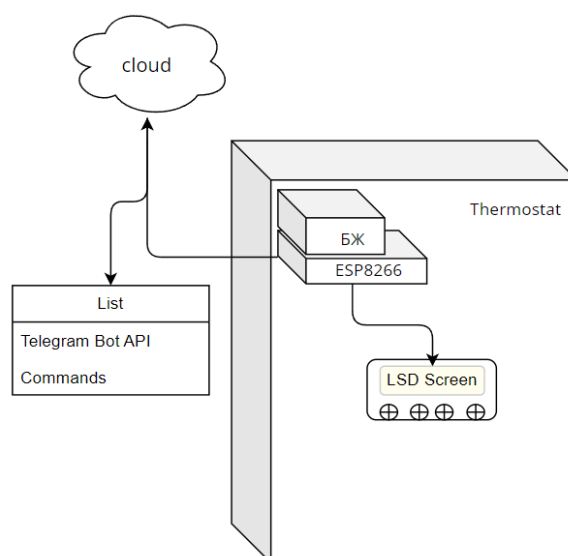


Рисунок 5.5 – Структурна схема розташування компонентів

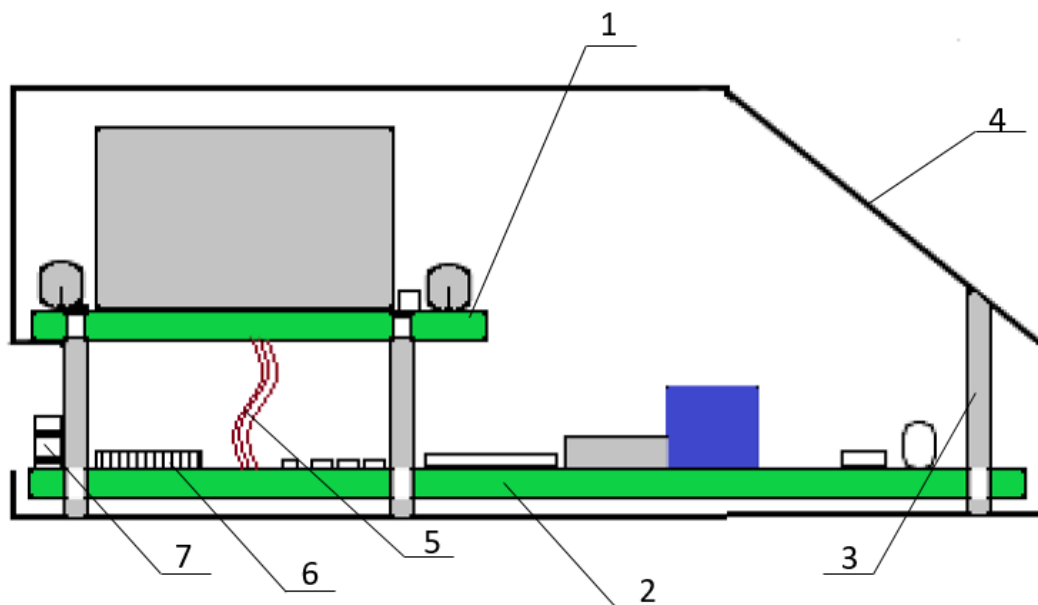


Рисунок 5.6 – Ескіз розташування та кріплення друкованих плат в корпусі: 1 - друкована плата блоку живлення, 2 - друкована плата основної частини медичного лабораторного smart – термостата, 3 - штифти, 4 - корпус, 5 - шина яка з'єднує плату 1 та 2, 6 - піни підключення до друкованої плати передньої панелі керування, 7 - ро'єм для живлення.

5.2 Налаштування та підключення до Telegram-бота

Для налаштування та підключення медичного лабораторного smart-термостата на базі мікроконтролера ESP до Telegram-бота необхідно виконати кілька етапів, які включають створення бота, налаштування ESP та програмування взаємодії між ними. Ось детальний опис кожного з етапів.

Спочатку необхідно створити нового Telegram-бота. Для цього відкрити Telegram та знайти BotFather. Надіслати команду /start, а потім /newbot. Вказати ім'я бота і унікальне ім'я користувача, що закінчується на "bot". Отримати токен доступу, який знадобиться для взаємодії з ботом через Telegram API.

Далі налаштувати мікроконтролер ESP (ESP8266 або ESP32). Прошити його кодом, що дозволяє з'єднуватися з Wi-Fi мережею та взаємодіяти з Telegram API. Використовується Arduino IDE або іншу платформу для написання коду.

Підключити ESP до Wi-Fi. У кодi визначте SSID i пароль вашої Wi-Fi мережі. Налаштувати бібліотеки, такі як ESP8266WiFi.h або WiFi.h для ESP32, а також Universal Telegram Bot для взаємодії з Telegram API.

Прописується код для з'єднання з Telegram-ботом. Використовується токен доступу, отриманий від BotFather, щоб налаштувати бібліотеку TelegramBot. Створюються функції для обробки команд та запитів, які надходять від користувачів через Telegram.

Розробка функції для читання температури з сенсора, керування нагрівальним елементом через реле та надсилання повідомлень користувачам. Вони будуть взаємодіяти з Telegram-ботом, виконуючи відповідні дії за командами користувача. Функція для читання температури та відправлення її значення через Telegram. Код може виглядати так:

```
void handleTemperature() {  
  
    float temperature = readTemperature();  
  
    String message = "Current temperature: " + String(temperature) + "°C";  
  
    bot.sendMessage(chat_id, message, "");  
  
}
```

Розробіть функцію для керування нагрівальним елементом. Вона може виглядати так:

```
void handleHeating(String command) {  
  
    if (command == "on") {  
  
        digitalWrite(HEATER_PIN, HIGH);  
  
        bot.sendMessage(chat_id, "Heating is turned on", "");  
  
    } else if (command == "off") {  
  
        digitalWrite(HEATER_PIN, LOW);  
  
    }  
  
}
```

```

    bot.sendMessage(chat_id, "Heating is turned off", "");
}
}

```

Налаштування основний цикл програми для обробки запитів від Telegram-бота. Використовуйте функцію `bot.getUpdates()` для отримання нових повідомлень та обробляйте їх відповідно до команд:

```

void loop() {
    if (millis() > lastTime + 1000) {
        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        while (numNewMessages) {
            handleNewMessages(numNewMessages);
            numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        }
        lastTime = millis();
    }
}

```

Функція `handleNewMessages()` обробляє нові повідомлення:

```

void handleNewMessages(int numNewMessages) {
    for (int i = 0; i < numNewMessages; i++) {
        String chat_id = String(bot.messages[i].chat_id);
        String text = bot.messages[i].text;

        if (text == "/start") {
            String welcome = "Welcome to the Smart Thermostat bot!";

```

```

bot.sendMessage(chat_id, welcome, "");

    } else if (text == "/temperature") {

        handleTemperature();

    } else if (text.startsWith("/heating")) {

        String command = text.substring(text.indexOf(' ') + 1);

        handleHeating(command);

    }

}

}

```

На завершення перевіряється, чи правильно налаштований Wi-Fi і Telegram-бот. Завантажується код на мікроконтролер ESP і перевіряється, що він успішно з'єднується з мережею та Telegram API. Тестуються функції керування термостатом через команди, надіслані боту. Перевіряємо, що всі функції працюють коректно і реагують на команди користувача.

5.3 Оцінка надійності приладу

Оцінка надійності приладу, такого як медичний лабораторний smart-термостат на базі мікроконтролера ESP, включає розрахунок ймовірності відмов кожного компонента і загальної ймовірності безвідмовної роботи приладу. Для цього використовуються показники надійності кожного компонента, такі як середній час до відмови (MTBF - Mean Time Between Failures).

Середній час до відмови (MTBF): Кожен компонент має свій MTBF, виражений в годинах.

- Мікроконтролер ESP: 50 000 годин
- Сенсор температури (DS18B20): 100 000 годин

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

- Реле: 20 000 годин
- Резистори, конденсатори та інші пасивні компоненти: 200 000 годин
- Блок живлення: 30 000 годин

Кількість компонентів кожного типу:

- Мікроконтролери ESP: 1 шт.
- Сенсори температури: 2 шт.
- Реле: 1 шт.
- Пасивні компоненти (резистори, конденсатори тощо): 70 шт.
- Блок живлення: 1 шт.

Розрахунок загального MTBF приладу: Загальний MTBF для приладу розраховується за формулою:

$$\frac{1}{MTBF_{\text{заг}}} = \sum \left(\frac{1}{MTBF_i} \cdot N_i \right)$$

Де $MTBF_i$ - середній час до відмови окремого компонента, N_i - кількість таких компонентів.

Підставляємо наші значення:

$$\frac{1}{MTBF_{\text{заг}}} = \frac{1}{50000} \cdot 1 + \frac{1}{100000} \cdot 2 + \frac{1}{20000} \cdot 1 + \frac{1}{200000} \cdot 70 + \frac{1}{30000} \cdot 1$$

$$\frac{1}{MTBF_{\text{заг}}} = 0.00002 + 0.00002 + 0.00005 + 0.00035 + 0.0000333$$

$$\frac{1}{MTBF_{\text{заг}}} = 0.0004733$$

$$MTBF_{\text{заг}} = \frac{1}{0.0004733} = 2112 \text{ годин}$$

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

Загальний середній час до відмови (MTBF) нашого приладу становить приблизно 2112 годин. Це означає, що в середньому прилад повинен працювати близько 2112 годин до виникнення першої відмови.

Час роботи медичного термостата на рік залежить від кількості годин, протягом яких він використовується щоденно. Якщо припустити, що термостат працює цілодобово, що є типовим для багатьох лабораторних приладів, розрахунок виглядатиме наступним чином:

1 день = 24 години з них 6 годин робочих на повній ставці лаборанта

1 рік = 365 днів з них робочих днів 275

Отже, кількість годин роботи термостата за рік дорівнює:

6 години/день · 275 днів/рік = 1650 годин/рік

Точна кількість годин залежить від конкретного режиму роботи термостата в вашій лабораторії.

Для оцінки ймовірності безвідмовної роботи протягом певного періоду часу t , використовується експоненційний розподіл:

$$R(t) = e^{-\frac{t}{MTBF}}$$

$$R(1650) = e^{-\frac{1650}{2112}} = e^{-0.79} = 37\%$$

Це означає, що ймовірність того, що прилад не вийде з ладу протягом 1650 годин або 1 року експлуатації, становить приблизно 37%

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

ВИСНОВКИ

У ході даної роботи було розроблено медичний лабораторний smart-термостат. Основною метою дослідження було створення пристрою, який забезпечує точний та надійний контроль температури в лабораторних умовах, що є критично важливим для успішного проведення медичних досліджень та діагностичних процедур.

Спочатку був проведений аналіз існуючих технологій та конструкцій термостатів, визначено основні вимоги до їх функціональності та безпеки. Зокрема, було розглянуто ключові аспекти точного регулювання температури, автоматизації процесів та інтеграції з цифровими системами моніторингу та керування.

На основі отриманих даних були розроблені структурна та функціональна схеми smart-термостата. Вибір даних схем був обґрунтований можливістю забезпечення високої точності температурного контролю та зручності в експлуатації. На основі функціональної схеми було синтезовано та розраховано електричну принципову схему пристрою, що дозволило визначити основні параметри та характеристики термостата.

Кінцевим результатом роботи є розробка конструкції пристрою, яка складається з трьох основних частин: блоку живлення, панелі керування, та власне головного модулю на основі мікроконтролера ESP. Конструкція була розроблена з урахуванням вимог ергономіки та безпеки, що забезпечує зручність та надійність у використанні.

Основні технічні характеристики розробленого термостата включають точність регулювання температури, можливість автоматизації процесів та інтеграцію з цифровими системами. Робоча температура термостата може бути налаштована в діапазоні від 5 до 60 °C з точністю до ± 0.1 °C, що відповідає вимогам сучасних медичних лабораторій. Загальна споживана потужність пристрою не перевищує 100 Вт.

Ймовірність безвідмовної роботи термостата протягом 1650 годин/рік становить приблизно 37%. Всі вимоги технічного завдання були виконані

					ІКБІ.480 004.005 - ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Плавкі вставки і запобіжники. Довідник. Режим доступу: <https://www.voltera.com.ua/uk-ua/catalogue/low-voltage/fuses?ajaxfilter=e1-%2C05>
2. Термостат на Arduino з налаштуванням нагрівання-охолодження). Режим доступу: <https://qazf.com.ua/thermostat-arduino/>
3. Biovet.ua Обладнання для ветеринарії). Режим доступу: <https://biovet.ua/ua/dezinfektsiya/termostati-sukhopovtryan/>
<https://biovet.ua/ua/portativnyy-inkubator-minitube-s-ustanovlennoy-temperaturoy-37c/>
4. Схеми радіоаматорів). Режим доступу: <https://sxem.org/component/tags/tag/termostat>
5. IoT – devices: Модуль живлення для IoT пристроїв на ESP8266 або Arduino). Режим доступу: <https://iot-devices.com.ua/power-supply-module-for-esp8266-arduino/>
6. LD1117 Adjustable and fixed low drop positive voltage regulator). Режим доступу: <https://www.st.com/resource/en/datasheet/ld1117.pdf>
7. Понижувальний DC-DC перетворювач з підтримкою QC3.0, QC4.0). Режим доступу: <https://electronica.in.ua/ua/p1840348275-ponizhayuschij-preobrazovatel-podderzhkoj.html>
8. Visual – paradigm: конструктор структурних схем). Режим доступу: <https://online.visual-paradigm.com/drive/#diagramlist:proj=0&diagram=list>
9. Octopart: Texas Instruments UA7805CKCS). Режим доступу: https://octopart.com/ua7805ckcs-texas+instruments-525328?gad_source=1&gclid=CjwKCAjwydSzBhBOEiwAj0XN4B1_KS_6CI93nboutNUKTUbZQXwCsiuVGysCdXJW2GbyQJhL0r4HcRoCOfQQAyD_BwE
10. Alldatasheet.com: LM2596(164) recommended result). Режим доступу: https://www.alldatasheet.com/view_datasheet.jsp?Searchword=LM2596

					ІКБІ.480 004.005 - ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

ХімтестУкраїна+). Режим доступу:

<https://chemtest.com.ua/ua/>

12. μ A7800 SERIES POSITIVE-VOLTAGE REGULATORS). Режим доступу:

<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>

13. Project Hub: Mini-thermostat with input for a 2-wire NTC thermistor and output for a resistive heating element. Plus code & schematics). Режим доступу:

<https://projecthub.arduino.cc/salhar1/mini-thermostat-f48c67>

14. GrabCAT Community). Режим доступу:

<https://grabcad.com/library>

15. DipTrace forum: LCD connector). Режим доступу:

<https://diptrace.com/forum/viewtopic.php?t=11138>

<https://diptrace.com/forum/viewtopic.php?t=12899>

16. Github: ESP32-WROOM module is added). Режим доступу:

<https://github.com/lexus2k/Espressif->

[diptrace/commit/6911f6a6126a05984c0bc18289552726ef1a2e43](https://github.com/lexus2k/Espressif-diptrace/commit/6911f6a6126a05984c0bc18289552726ef1a2e43)

17. SnapMagic: ESP8266-12E/ESP-12E). Режим доступу:

<https://www.snapeda.com/parts/ESP8266-12E/ESP-12E/AI-Thinker/view-part/?company=ZTU&welcome=home>

18. IMRAD: 2-CH SPDT RELAY CARRIER WITH 12VDC RELAY). Режим доступу: <https://imrad.com.ua/ua/2-ch-spdt-relay-carrier-with-12vdc-relay-3>

19. МВС Україна: Технічні вимоги медобладнання). Режим доступу :

[https://ch-tmo.mvs.gov.ua/files/documents/%D1%82%D0%B5%D1%85%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3%D0%B8\(1\).docx](https://ch-tmo.mvs.gov.ua/files/documents/%D1%82%D0%B5%D1%85%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3%D0%B8(1).docx)

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

Додатки

					ІКБІ.480 004.005 - ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

Прошивка LCD

```

void lcd_update() {
    switch (Menu_S) {
        case 0:
            if (state_Work == 1) {                                     // початковий екран
                timeMins = ON_state_current;
                lcd.setCursor(0, 0); lcd.print("WORK P:      ");          // ON
            } else {
                timeMins = OFF_state_current;
                lcd.setCursor(0, 0); lcd.print("PAUSE P:OFF ");          // OFF
            }
            lcd.setCursor(12, 2); lcd.print("W:");
            lcd.print(ON_state); lcd.print("min ");
            lcd.setCursor(0, 2); lcd.print("---Setup--- ");
            lcd.setCursor(0, 3); lcd.print("Temp");
            lcd.print(TM_state, 1); lcd.print(".0\1C ");
            lcd.setCursor(11, 3); lcd.print(" P:");
            lcd.print(OFF_state); lcd.print("min ");
            lcd.display();
            break;
        case 1:                                                         // редактор ON
            lcd.clear();
            lcd.setCursor(2, 0); lcd.print("Setup Time Work");
            lcd.setCursor(0, 1); lcd.print("-----");
            lcd.setCursor(7, 2); lcd.print("> ");
            lcd.print(ON_state); lcd.print(" < min");
            lcd.setCursor(0, 3); lcd.print("Limits: 1 to 600 min");
            break;
        case 2:                                                         // редактор OFF

```

```

    lcd.clear();

    lcd.setCursor(2, 0); lcd.print("Setup Time Pause ");

    lcd.setCursor(0, 1); lcd.print("-----");

    lcd.setCursor(7, 2); lcd.print("> ");

    lcd.print(OFF_state); lcd.print(" < min");


    lcd.setCursor(0, 3); lcd.print("Limits: 1 to 600 min");

    break;

case 3:                                     // редактор температури

    lcd.clear();

    lcd.setCursor(1, 0); lcd.print("Setup Temperature");

    lcd.setCursor(0, 1); lcd.print("-----");

    lcd.setCursor(5, 2); lcd.print("> ");

    lcd.print(TM_state); lcd.print(".0 < \1C");

    lcd.setCursor(1, 3); lcd.print("Limits: 1 to 150\1C");

    break;

case 4:                                     // редактор потужності вентилятора

    lcd.clear();

    lcd.setCursor(2, 0); lcd.print("Setup Power Fan");

    lcd.setCursor(0, 1); lcd.print("-----");

    lcd.setCursor(7, 2); lcd.print("> ");

    lcd.print(Fan_state); lcd.print(" < %");

    lcd.setCursor(1, 3); lcd.print("Limits: 1 to 100%");

```

```

        break;

case 5:                                     // редактор роботи тени

    lcd.clear();

    lcd.setCursor(1, 0); lcd.print("Setup Time UV Lamp");

    lcd.setCursor(0, 1); lcd.print("-----");

    lcd.setCursor(7, 2); lcd.print("> ");

    lcd.print(UV_state); lcd.print(" < min");

    lcd.setCursor(0, 3); lcd.print("Limits: 1 to Time P");

    break;

case 6:

    lcd.clear();

    lcd.setCursor(3, 0); lcd.print("Save settings:");

    lcd.setCursor(0, 1); lcd.print("-----");

    if (ST_state == 1) {

        lcd.setCursor(7, 2);

        lcd.print("> Yes <");

        lcd.setCursor(2, 3);

        lcd.print("Save User setting");

    } else {

        lcd.setCursor(7, 2);

        lcd.print("> No <");

        lcd.setCursor(2, 3);

        lcd.print("Load User setting");

```

```

    }

    break;

default:

    break;

}

}

```

Додаток В

Прошивка PID_OUT

```

void PID_OUT() {

    if (millis() - windowStartTime >= WindowSize) {

        windowStartTime += WindowSize;

        if (windowStartTime > millis()) windowStartTime = 0;

    }

    if (Menu_S == 0 && state_Work == 1) {

        lcd.setCursor(7, 0);

        lcd.print(map(Output, 0, WindowSize, 0, 100)); lcd.print("% ");

        //dimmer.setState(ON);

        //dimmer.setPower(Fan_state_out);

        digitalWrite(Fan_pin, HIGH);

        if (Output < millis() - windowStartTime) {

            digitalWrite(Relay_pin, LOW);

        } else {

            digitalWrite(Relay_pin, HIGH);

        }

    }

}

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    } else {

        //dimmer.setState(OFF);

        digitalWrite(Fan_pin, LOW);

        digitalWrite(Relay_pin, LOW);

    }

    if (state_Work == 0 && UV_state_current > 0) {

        digitalWrite(UV_lamp_pin, HIGH);

    } else {

        digitalWrite(UV_lamp_pin, LOW);

    }

}

```

Додаток С

Прошивка Time_View

```

int redrawDisplay(uint32_t sec, uint32_t timeMins) {

    tf = !tf;

    humidity = dht.readHumidity();

    #if (sensor == 1)

        temperature = dht.readTemperature();                // DHT

    #else

        DS.requestTemp();

        temperature = DS.getTemp();                        // 18B20

    #endif
}

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

lcd.setCursor(0, 1); lcd.print("Temp");

lcd.print(temperature, 1); lcd.print("\1C ");


lcd.setCursor(11, 1); lcd.print(" H:");

lcd.print(humidity, 1); lcd.print("% ");

if (tf == 1) {

    lcd.setCursor(poz + 2, 0);

    lcd.print(" ");

    lcd.setCursor(poz + 5, 0);

    lcd.print(" ");

} else {

    lcd.setCursor(poz + 2, 0);

    lcd.print(":");

    lcd.setCursor(poz + 5, 0);

    lcd.print(":");

}

int timeHours = (timeMins / 60ul) % 24; // Час

if (timeHours < 10) {

    lcd.setCursor(poz, str);

    lcd.print("0");

    lcd.setCursor(poz + 1, str);

    lcd.print(timeHours);

} else {

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    lcd.setCursor(poz, str);

    lcd.print(timeHours);

}

int timeMins1 = (timeMins) % 60ul;                                // Хвилини

if (timeMins1 < 10) {

    lcd.setCursor(poz + 3, str);

    lcd.print("0");

    lcd.setCursor(poz + 4, str);

    lcd.print(timeMins1);

} else {

    lcd.setCursor(poz + 3, str);

    lcd.print(timeMins1);

}

int timeSecs = (sec % 3600ul) % 60ul;                                // Секунди

if (timeSecs < 10) {

    lcd.setCursor(poz + 6, str);

    lcd.print("0");

    lcd.setCursor(poz + 7, str);

    lcd.print(timeSecs);

} else {

    lcd.setCursor(poz + 6, str);

    lcd.print(timeSecs);

}

lcd.display(); lcd.backlight();

}

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток D

Прошивка TRT.v.4

```
#define NAME ">>THERMOSTAT-TIMER<<" // version

#define VERSION "V.4.71"

//////////////////// Sensor MODE //////////////////////

#define sensor      1      // читання температури: 0 = DS18B20, 1 = DHT22

//////////////////// PIN MODE //////////////////////

#define DS_pin      A2                      // 18B20
#define DHT_pin     A3                      // DHT
#define Relay_pin   13                      // нагрів
#define Fan_pin     7                       // вентилятор
#define UV_lamp_pin 10                      // УФ тена
#define pin_A       3
#define pin_B       4
#define button      5
#define zerocross    2

#define WindowSize  512                    // PID: SSD = 512, Relay = 5120

////////////////////

#include <microLiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4); // LCD адрес и розмір SDA = A4, SCL = A5

#include <EEPROM.h>

// #include <RBDdimmer.h>

// dimmerLamp dimmer(Fan_pin);

#include <PID_v1.h>

#include <DHT.h>
```

					ІКБІ.480 004.005 - ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

#include <TimerOne.h>

#include <GyverButton.h>

#if (sensor == 0)

#include <microDS18B20.h>

MicroDS18B20 DS(DS_pin);

#endif

DHT dht(DHT_pin, DHT22);

GButton butt1(button);

int16_t ON_state,

        OFF_state,

        TM_state,

        Fan_state,

        UV_state,

        ON_state_current,

        OFF_state_current,

        UV_state_current,

        Fan_state_out,

        timeMins = 0;

unsigned char ST_state,

        Menu_S = 0,

        encoder_A,

        encoder_B,

        encoder_A_prev = 0,

        seconds = 60,

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Step;

unsigned long loopTime1,

        loopTime2,

        windowStartTime,

        currentTime;

uint8_t simvol[8]   = {B00110,

                        B01001,

                        B01001,

                        B00110,

                        B00000,

                        B00000,

                        B00000,

                        B00000,

                        };                                // символ градуса

uint8_t tf = 1,

        Eprom_SET = 1,

        state_Work = 1;

uint8_t poz = 12, str = 0;                                // позиція и стрічки
часу

double Setpoint,

        Input,

        Output,

        temperature,

        humidity;

double Kp = 25, Ki = 1, Kd = 15, dt = 100;

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void setup()  {

    Serial.begin(9600);                                // Serial.print(' ');
    Serial.println(' ');

    Timer1.initialize(1000);

    Timer1.attachInterrupt(timerIsr);                  // запуск таймера

    lcd.init();                                         // ініціалізація
    lcd

    lcd.init();

    lcd.setCursor(0, 0); lcd.print(NAME);

    lcd.setCursor(7, 1); lcd.print(VERSION);

    lcd.setCursor(4, 2); lcd.print(__DATE__);

    lcd.setCursor(6, 3); lcd.print("MEGA2020");

    myPID.SetOutputLimits(0, WindowSize);

    myPID.SetMode(AUTOMATIC);

    dht.begin();

    #if (sensor == 0)

        uint8_t address[8];                            // масив, адрес датчика

        DS.readAddress(address);

    #endif

    pinMode(Relay_pin, OUTPUT);

    pinMode(Fan_pin, OUTPUT);

    pinMode(UV_lamp_pin, OUTPUT);

    pinMode(pin_A, INPUT);

    pinMode(pin_B, INPUT);

```

```

pinMode(button, INPUT);

// pinMode(sync, INPUT);

currentTime = millis();

loopTime1 = currentTime;

lcd.createChar(1, simvol); // символ градуса

lcd.backlight();

delay(2000);

lcd.clear();

if (digitalRead(button) == LOW) { // ЕЕПРОМ

    EEPROM.write(0, 10); // час роботи

    EEPROM.write(1, 10); // час паузи

    EEPROM.write(2, 25); // температура

    EEPROM.write(3, 50); // потужність
    вентильатора

    EEPROM.write(4, 5); // тена

    lcd.setCursor(2, 1);

    lcd.print("Load User setting");

    lcd.noDisplay();

    delay(300);

    lcd.display();

    delay(300);

    lcd.noDisplay();

    delay(300);

    lcd.display();

```

```

    delay(300);

    lcd.noDisplay();

    delay(300);

    lcd.display();

    delay(300);

}

lcd.noBacklight();

lcd.clear();

lcd_update();

lcd.noDisplay();

//dimmer.begin(NORMAL_MODE, OFF);

}

void timerIsr() {

    buttl.tick();                                // читаємо кнопку
}

void loop() {                                    // читаємо кнопку

    if (Eprom_SET == 1) {

        ON_state = EEPROM.read(0);

        OFF_state = EEPROM.read(1);

        TM_state = EEPROM.read(2);

        Fan_state = EEPROM.read(3);

        UV_state = EEPROM.read(4);

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    ON_state_current = ON_state - 1;

    OFF_state_current = OFF_state - 1;

    UV_state_current = UV_state - 1;

    Eprom_SET = 0;

    lcd_update();

}

if (butt1.isHold()) {

    Step = 10;

} else {

    Step = 1;

}

Input = temperature;

Setpoint = TM_state;


if (butt1.isClick()) {

    switch (Menu_S) {

        case 1:

            Menu_S += 1; state_Work = 1;

            break;

        case 2:

            Menu_S += 1;

            UV_state = OFF_state < UV_state ? OFF_state : UV_state;

            break;

        case 3:

            //dimmer.setState(ON);

            Menu_S += 2;

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        break;

case 4:

    //dimmer initialisation: name.begin(MODE, STATE)

    Menu_S += 1;

    break;

case 5:

    Menu_S += 1;

    break;

case 6:

    Menu_S = 0;


    if (ST_state == 1) {

        EEPROM.write(0, ON_state);

        EEPROM.write(1, OFF_state);

        EEPROM.write(2, TM_state);

        EEPROM.write(3, Fan_state);


        EEPROM.write(4, UV_state);

    }

    Eprom_SET = 0;

    seconds = 60;    //state_Work = 1;

    lcd.clear();

    break;

default: Menu_S += 1;

}

lcd_update();

}

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

currentTime = millis();

if (currentTime >= (loopTime1 + 2)) {

    encoder_A = digitalRead(pin_A);
    encoder_B = digitalRead(pin_B);

    if ((!encoder_A) && (encoder_A_prev)) {

        if (encoder_B) {                                     // збільшення

            switch (Menu_S) {

                case 1:                                       // налаштування часу Work

                    if (ON_state + Step <= 600)

                        ON_state += Step;

                    lcd_update();

                    break;

                case 2:                                       // налаштування часу Pause

                    if (OFF_state + Step <= 600)

                        OFF_state += Step;

                    lcd_update();

                    break;

                case 3:                                       // установка температури

                    if (TM_state + Step <= 150)

                        TM_state += Step;

                    lcd_update();

                    break;

                case 4:                                       // установка потужності вентилятора

                    if (Fan_state + Step <= 100)

```

					ІКБІ.480 004.005 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		88

```

        Fan_state += Step; //dimmer.setPower(Fan_state_out);

        lcd_update();

        break;

case 5:                                // установка часу роботи лампи, тени

        if (UV_state + Step <= OFF_state)

                UV_state += Step;

        lcd_update();

        break;

case 6:                                // збереження "Yes"

        if (ST_state + Step <= 1)

                ST_state += Step;

        lcd_update();

        ON_state_current = ON_state - 1;

        OFF_state_current = OFF_state - 1;

        UV_state_current = UV_state;

        break;

default:

        break;

}

} else {

        switch (Menu_S) {                                // зменшення

        case 1:

                if (ON_state - Step > 0)

                        ON_state -= Step;

                lcd_update();

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        break;

case 2:

    if (OFF_state - Step > 0)

        OFF_state -= Step;

    lcd_update();

    break;

case 3:

    if (TM_state - Step > 0)

        TM_state -= Step;

    lcd_update();

    break;

case 4:

    if (Fan_state - Step > 0)

        Fan_state -= Step;    //dimmer.setPower(Fan_state_out);

    lcd_update();

    break;

case 5:

    if (UV_state - Step > 0)

        UV_state -= Step;

    lcd_update();

    break;

case 6:                                     // збереження "No"

    if (ST_state - Step > 0)

        ST_state -= Step;

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        lcd_update();

        break;

    default:

        break;

    }

}

}

encoder_A_prev = encoder_A;

loopTime1 = currentTime;

}

if (Fan_state < 0) Fan_state = 0;

if (Fan_state > 100) Fan_state = 100;

Fan_state_out = map(Fan_state, 0, 100, 0, 80);

if (Menu_S == 0 && currentTime >= (loopTime2 + 1000)) {    // зворотній
Відлік кожну 1 с

    loopTime2 = currentTime;

    if (seconds - 1 >= 0) {

        seconds -= 1;

    } else {

        seconds = 60;

        if (state_Work == 1) {                                // ON

            if (ON_state_current >= 1) {

                ON_state_current -= 1; timeMins = ON_state_current;

            } else {

                ON_state_current = ON_state - 1;

                OFF_state_current = OFF_state - 1;

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        UV_state_current = UV_state;

        state_Work = 0;

        lcd_update();

    }

} else {

    if (state_Work == 0) {                                     // OFF

        if (OFF_state_current >= 1) {

            OFF_state_current -= 1; timeMins = OFF_state_current;

            UV_state_current -= 1;

        } else {

            ON_state_current = ON_state - 1;

            OFF_state_current = OFF_state - 1;

            UV_state_current = UV_state;

            state_Work = 1;

            lcd_update();

        }

    }

}

    lcd_update();

}

redrawDisplay(seconds, timeMins);

}

myPID.Compute();

PID_OUT();

}

```

					ІКБІ.480 004.005 - ПЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		