

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет комп'ютерно-інтегрованих технологій, мехатроніки і
робототехніки
Кафедра робототехніки, електроенергетики та автоматизації
ім. проф. Б.Б. Самотокіна

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи магістра

на тему: Лабораторний стенд дослідження законів керування на основі
автоматизованої системи керування температурою у термошафі

Виконав студент 2 курсу, групи АТ-28м
спеціальності 174 “Автоматизація,
комп'ютерно-інтегровані технології
та робототехніка”

Мельник М.М.

(прізвище та ініціали студента)

Керівник: канд. техн. наук,

старший викладач кафедри

Підтиченко О.В.

(науковий ступінь, вчене звання, посада,
прізвище та ініціали)

Рецензент: доктор техн. наук, професор
завідувач кафедри інформаційно-
вимірювальних технологій

Подчашинський Ю.О.

(науковий ступінь, вчене звання, посада,
прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ “ЖИТОМИРСЬКА ПОЛІТЕХНІКА”
Факультет комп’ютерно-інтегрованих технологій, мехатроніки і
робототехніки
Кафедра робототехніки, електроенергетики та автоматизації
ім. проф. Б.Б. Самотокіна

Спеціальність 174 “Автоматизація, комп’ютерно-інтегровані технології
та робототехніка”

ЗАТВЕРДЖУЮ
Завідувач кафедри
робототехніки, електроенергетики
та автоматизації
ім. проф. Б.Б. Самотокіна
к.т.н., доц. Громовий О.А.

« _____ » _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

Студент _____ Мельник Матвій Михайлович

1. Тема роботи: Лабораторний стенд дослідження законів керування на основі автоматизованої системи керування температурою у термошафі

Затверджена Наказом по університету від «12» листопада 2024 р. № 646/с

2. Термін здачі закінченої роботи на кафедру «20» грудня 2024 р.

3. Вихідні дані до роботи: Технічна документація на контролери платформи Arduino та інші електронні компоненти. Опис та принципи роботи розподілених АСУ ТП (телемеханічних систем) Технічний опис інтерфейсу UART

4. Зміст пояснювальної записки (перелік питань, які підлягають розробці)

1) Аналіз законів керування. 2) Розробка структурної схеми системи керування.

3) Розробка структурної та електричної принципової схеми системи керування.

4) Розробка алгоритмів роботи системи

5. Перелік презентативного матеріалу (з точним визначенням обов’язкових слайдів/креслень)

1. Структурна схема системи керування. 2. Електрична принципова схема системи керування. 3 Алгоритм роботи контролера системи керування.

6. Консультанти з кваліфікаційної роботи магістра із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технічна частина	Підтиченко О.В.	10.10.24	

7. Дата видачі завдання 10.10.2024 р.

Календарний план

№ з/п	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз об'єкту дослідження	01.04.24 – 20.04.24	
2.	Опис та принципи роботи розподілених АСУ ТП	21.10.24 – 25.10.24	
3.	Розробка структурної схеми автоматизованої системи керування, вибір технологічних засобів автоматизації	26.10.24 – 01.11.24	
4.	Розробка електричної принципової схеми автоматизованої системи керування	12.10.24 – 17.10.24	
5.	Розробка алгоритмічного забезпечення для розробленої системи керування	16.11.24 – 25.11.24	
6.	Побудова ТМС на базі лабораторного стенду	06.12.24 – 20.12.24	
7.	Оформлення проекту	03.12.24 – 20.12.24	

Студент

_____ (підпис, дата)

Мельник М.М.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис, дата)

Підтиченко О.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

В даній магістерській дипломній роботі розроблено (удосконалено) лабораторний стенд дослідження законів керування на основі автоматизованої системи керування температурою у термошафі. Розроблений дозволяє виконувати відповідні дослідження для навчальних цілей.

Спираючись на отримані знання та інформацію в результаті роботи було виконано вибір елементної бази, а саме ряду виконавчих пристроїв. В проекті виконано розробку: структурної та електричної принципової схем лабораторного стенду, алгоритмічного та програмного забезпечення. Розроблена система керування стендом дозволяє здійснювати роботу в декількох встановлених режимах.

.

ABSTRACT

In this master's thesis, a laboratory stand for the study of control laws based on an automated temperature control system in a thermochamber has been developed (improved). The developed stand allows for the performance of relevant research for educational purposes.

Based on the knowledge and information obtained as a result of the work, the element base was selected, namely a number of actuators. The project includes the development of: structural and electrical schematic diagrams of the laboratory stand, algorithmic and software. The developed stand control system allows for work in several installed modes.

РЕФЕРАТ

Дана магістерська робота представлена в обсязі 73 сторінки формату А4, 1 лист специфікації. Дипломна робота складається зі вступу, 4 розділів, висновків, списку використаних джерел та додатків. Все це разом надає вичерпну інформацію про лабораторний стенд дослідження законів керування на основі автоматизованої системи керування температурою у термошафі.

В першому розділі розглядається аналіз законів керування, їх загальна характеристика, переваги та недоліки, місця застосування. В даному розділі описано загальні підходи до побудови АСУ ТП.

У другому розділі представлено загальну структурну схему ТМС. та описано роботу даної схеми.

Третій розділ присвячений розробці загальної електричної принципової схеми системи керування. Тут наведено основну обрану елементну базу, та описано роботу схеми.

У Четвертому розділі представлено алгоритмічне, та програмне забезпечення роботи системи.

ЗМІСТ

ВСТУП	7
1. АНАЛІЗ ЗАКОНІВ КЕРУВАННЯ	8
1.1. Опис існуючих законів керування їх переваги та недоліки.....	7
1.2. Загальні поняття та визначення структури АСУТП та ТМС.....	12
1.3. Структура ієрархічної розподіленої АСУ ТП.....	16
1.4. Постановка задачі	21
2. РОЗРОБКА СТРУКТУРНОЇ СХЕМИ СИСТЕМИ КЕРУВАННЯ.....	24
2.1. Протокол передачі даних UART.....	24
2.2. Розробка структурної схеми системи керування.....	27
2.3 Висновки до розділу 2.....	29
3. РОЗРОБКА ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ СХЕМИ СИСТЕМИ КЕРУВАННЯ.....	30
3.1. Опис елементної бази.....	30
3.1.1. Апаратно програмна платформа Arduino.....	30
3.1.2. LCD Keypad Shield.....	31
3.1.3. Датчики.....	32
3.1.3. Блок живлення та конвертори напруги.....	35
3.1.4. Виконавчі механізми.....	36
3.2. Розробка електричної принципової схеми системи керування...	39
3.3. Висновки по розділу 3.....	40
4. РОЗРОБКА АЛГОРИТМІВ РОБОТИ СИСТЕМИ.....	41
4.1. Опис принципу роботи установки та реалізованих режимів роботи.....	41
4.2. Вигляд інтерфейсу меню на керуючому модулі.....	44
4.3. Програмне забезпечення Arduino IDE.....	52
4.4. Розробка програмного забезпечення.....	53
ЗАГАЛЬНІ ВИСНОВКИ	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТКИ.....	73

ВСТУП

Керування температурою в наші дні це надзвичайно поширений процес, що зачіпає майже всі галузі людської діяльності. Починаючи від металургійної промисловості з різноманітним литтям металів, хімічного та фармацевтичного виробництва де температура відіграє важливу роль, закінчуючи енергетикою та сільським господарством в якому контроль температури в оптимальних межах забезпечує краще вирощування рослин, та їх зберігання. Такий перелік можна продовжувати досить довго.

Основними завданнями при розробці, побудові, розрахунку параметрів керування, налагодженні та програмуванні систем автоматичного та автоматизованого керування об'єктами є ідентифікація об'єктів керування, вибір найкращого закону керування, визначення оптимальних налаштувань регулятора (із визначенням коефіцієнтів і параметрів законів керування). , визначення показників якості системи тощо. У процесі навчання здобувачів за даною та іншими спеціальностями важливим є вміння виконувати відповідні проектні задачі. Це пов'язано з необхідним розумінням розробки, побудови та експлуатації різноманітних систем автоматизованого керування об'єктами та процесами.

Саме тому, дослідження процесів керування температурою технічних та технологічних об'єктів, вибір законів керування, визначення показників якості та створення навчальних стендів та лабораторних установок для навчання майбутніх фахівців, проведення різного роду досліджень в межах освітнього процесу є завжди **актуальною задачею**.

1. АНАЛІЗ ЗАКОНІВ КЕРУВАННЯ

1.1. Опис існуючих законів керування їх переваги та недоліки

Релейний закон керування – найпростіший з законів керування, де відбувається перемикання регулятора між двома фіксованими величинами в залежності від відхилення

Особливостями є дуже проста реалізація, до недоліків відноситься можливість спричиняти коливання навколо встановленої точки.

Пропорційний закон керування або Р закон керування - це один із законів що реалізується регуляторами в системах автоматизованого керування.

Регулятори, що його реалізують також називають регуляторами з жорстким зворотнім зв'язком, або статичними регуляторами.

Його рівняння та передаточна функції мають вигляд:

$$\mu = k_p \Delta \quad (1.1)$$

$$W(p) = k_p \quad (1.2)$$

Де μ – це вплив регулятора на регулюючий орган(РО) за допомогою виконавчого механізму.

Δ – це сигнал розузгодження.

Перехідна функція П регулятора і вид перехідного процесу представлені на рис.1.1.

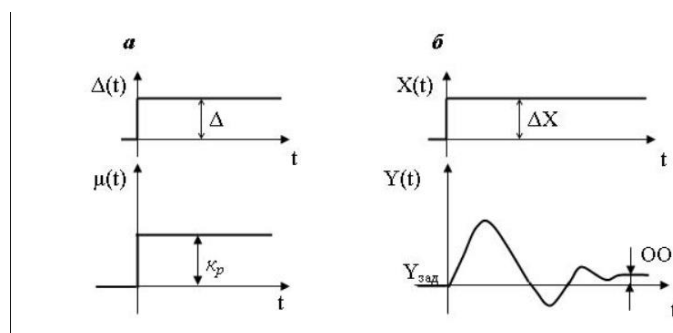


Рис. 1.1. Перехідна функція П регулятора і вид перехідного процесу

Принцип його дії полягає у зміні керуючого впливу пропорційно до величини помилки, тобто відхилення між заданим і фактичним значенням.

Його особливостями є :

- Простота реалізації.
- Швидке реагування на помилку.

Недоліками є можливе залишкове відхилення тобто статична помилка.

Доволі часто використовується для регулювання температури в простих обігрівачах.

Інтегральний закон керування або ж І – закон регулювання з це закон, у якому керуючий сигнал з регулятора, дорівнює інтегралу від розузгодження в часі. Тобто Регулятор враховує накопичену помилку за певний час і коригує керуючий вплив , для її усунення.

$$U = K_2 \int \varepsilon dt. \quad (1.3)$$

Даний закон реалізується І регулятором з параметром налаштування K_2 , або астатичним регулятором.

Його особливостями є:

- Усунення статичної помилки .
- Та повільніше реагування на зміну параметрів.

Недоліками цього регулятора є можливість перерегулювання тобто значні коливання перед стабілізацією.

Диференціальний закон керування або ж D регулятор. Даний вид регулятора реагує на швидкість зміни помилки, що дозволяє прогнозувати розвиток та вчасно корегувати керуючий вплив.

$$u(t) = K_d \frac{de(t)}{dt}. \quad (1.4)$$

Його особливостями є: швидке реагування на динамічні зміни системи.

До недоліків відноситься підвищена чутливість до шумів у вимірювальній системі. В чистому вигляді Диференціальних регуляторів не існує.

Пропорційно-інтегральний закон регулювання являється свого роду поєднанням двох інших регуляторів. Регулятори, що працюють по даному закону, виконують переміщення регулюючого органу пропорційно сумі відхилення та інтеграла відхилення регульованої величини. Тобто поєднують в собі пропорційну частину для швидкої реакції та інтегральну для усунення статичної помилки. Його математична модель представляється у вигляді:

$$u = K_1 \varepsilon + K_2 \int \varepsilon dt = K_1 \left(\varepsilon + \frac{1}{T_i} \int \varepsilon dt \right). \quad (1.5)$$

Реалізується ПІ закон за допомогою гнучкого зворотного зв'язку. У динамічному відношенні ПІ-регулятор подібний системі з двох паралельно включених регуляторів. Його особливістю є збалансоване керування між швидкістю реакції та точністю.

Одним з недоліків є проблематичність у роботі з швидкими змінами параметрів системи.

Пропорційно-диференціальний закон керування або ПД закон регулювання, реалізується в регуляторах, що по своїй суті можуть бути представлені у вигляді системи з двома паралельно працюючими типовими ланками: пропорційною та диференціальною.

Принцип їх роботи полягає у використанні пропорційної частини для базового регулювання, а диференціальної для передбачення динамічних змін у системі. Позитивною стороною їх використання є те. Що регулювання відбувається з урахуванням як величини відхилення параметра що регулюється так і швидкості його зміни.

Його перехідна функція зображена на рис.1.2.

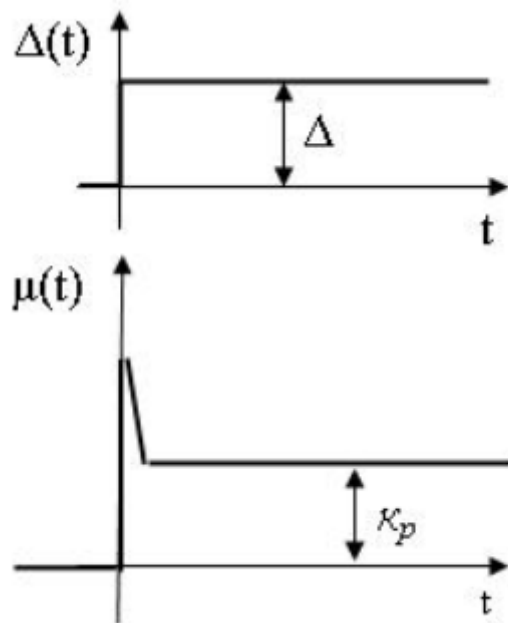


Рис. 1.2. Перехідна функція ПД регулятора

Його особливостями є те що він підходить для динамічних систем.

До недоліків відноситься відсутність зниження статичної помилки.

Пропорційно-інтегрально-диференціальний закон керування або ж PID регулятор. Регулятор, що поєднує в собі три складові попередніх регуляторів.

Його перехідна функція зображена на рис 1.3.

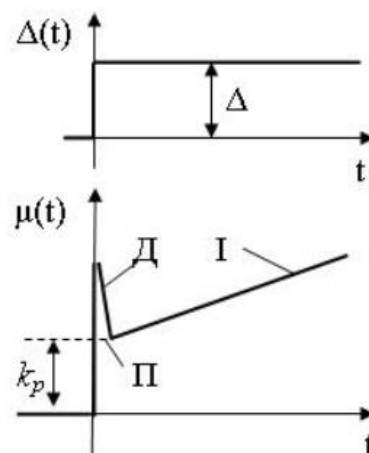


Рис. 1.3. Перехідна характеристика PID регулятора

Математична модель що поєднує три складові представлена нижче.

$$u = K_p \cdot e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \quad (1.5)$$

Особливостями цього регулятора є те що він підходить для складних систем з високими вимогами до точності та надійності.

Недоліком є складність налаштування коефіцієнтів що відповідають за кожну складову.

1.2. Загальні поняття та визначення структури АСУТП та ТМС

АСУТП або ж автоматизована система управління технологічними процесами – це певний комплекс програмних та технічних засобів, що призначений для прийняття та реалізації керуючої дії на певний технологічний об'єкт згідно прийнятих параметрів керування.

Під АСК ТП зазвичай розуміється комплексне рішення, що забезпечує автоматизацію основних технологічних операцій на виробництві в цілому або якійсь його ділянці. Складовими частинами АСУ ТП можуть бути окремі системи автоматичного управління (САУ) та автоматизовані пристрої, пов'язані в єдиний комплекс. Такі як системи диспетчерського управління та збору даних (SCADA), розподілені системи управління (DCS), і інші дрібніші системи управління (наприклад, системи на програмованих логічних контролерах (PLC)).

АСУ ТП зазвичай мають у своєму складі єдину систему операторського управління технологічним процесом, яка включає один або декілька пультів управління, засоби обробки та архівування інформації про хід процесу та типові елементи автоматики, такі як датчики, пристрої управління та виконавчі пристрої. Промислові мережі служать в даних системах для передачі інформації між підсистемами.

Велика різноманітність АСУ, відмінність їх функціонального призначення та вирішуваних задач, характеристик і розміщення елементів АСУ, реакції на обробку вхідного потоку інформації, часу на вироблення рішень і доведення їх до виконавців, технічні характеристики засобів автоматизованого керування призводять до необхідності класифікації АС. Класифікація дозволяє методологічно більш послідовно підійти до вирішення проблем проектування АС, так як вибір технічних засобів в значній мірі визначається належністю АС до того чи іншого класу. Класифікація АС може бути вельми різноманітною, проводиться по найрізноманітнішим ознакам і тому є достатньо умовною. Наведемо приклади класифікацій АСУ за різними ознаками, рис.1.4:

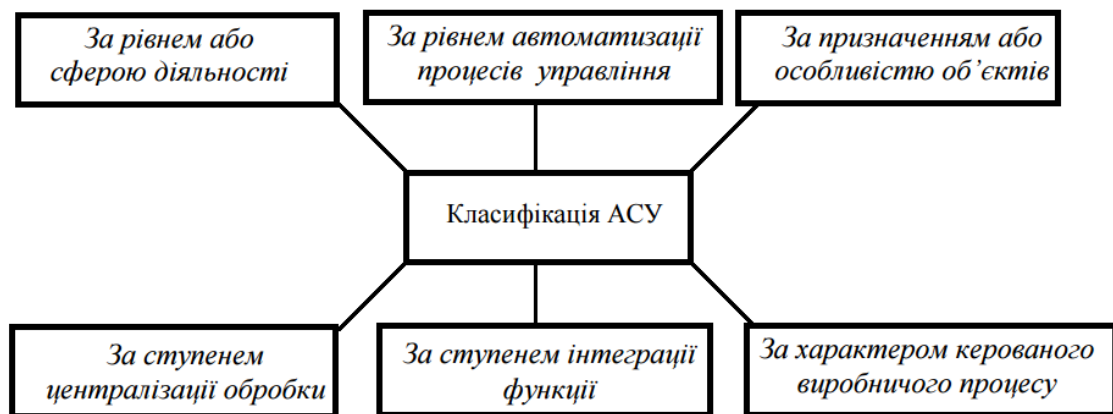


Рис.1.4. Класифікації АСУ за різними ознаками

1. За рівнем або сферою діяльності: державні, територіальні (регіональні), галузеві, підприємств або організацій, технологічних процесів.
2. За рівнем автоматизації процесів управління: інформаційно-пошукові, інформаційно-керівні, інформаційно-довідкові, СППР, інтелектуальні, експериментальні тощо.
3. За призначенням або особливістю об'єктів управління: адміністративні, виробничо-технічні, соціальні, транспортні тощо.

4. За ступенем централізації обробки інформації: централізовані, децентралізовані, колективного використання.
5. За характером керованого виробничого процесу: для виробництва з дискретним процесом (машинобудування, приладобудування); з неперервним процесом (хімічна, нафтопереробна і газова промисловості); з неперервно- дискретним процесом (металургійна, цементна, харчова промисловість).
6. За ступенем інтеграції функцій: багаторівневі з інтеграцією за рівнями планування, багаторівневі з інтеграцією за рівнями управління; комплексні (інтеграція з відмінними задачами).

Розподілена система управління - це комплекс технічних і програмних рішень для побудови АСУ ТП, характерною рисою якої є децентралізована обробка даних і наявність розподілених систем введення і виводу інформації.

Структура розподіленої АСУ складається із наступних рівнів.

Нижній рівень системи:

Містить у собі всі апаратні засоби системи, призначені для перетворення фізичних величин в електричні сигнали, сприймані нижнім рівнем системи. У польовий рівень входять датчики, що нормалізують перетворювачі, мультиплексори, контролери з програмувальною логікою, джерела живлення та інше необхідне устаткування. Крім того, у польовий рівень системи входять елементи керування технологічними процесами.

Середній рівень системи:

Містить у собі мікропроцесорний пристрій (пристрою) нижнього рівня, задачею якого є збір даних від польового рівня системи і керування його роботою. Нижній рівень забезпечує збір даних, їхню первинну обробку і збереження в енергонезалежній пам'яті для наступної передачі верхньому рівню системи. Передача даних від нижнього рівня верхньому здійснюється по підтримуваним нижнім рівнем каналам зв'язку, до яких відносяться

- нуль-модемний зв'язок з використанням інтерфейсу RS232;
- телефонний зв'язок, що комутується, з використанням модему;

- радіозв'язок з використанням радіо модему;
- зв'язок з використанням виділеної телефонної лінії;
- зв'язок з використанням засобів локальної обчислювальної мережі.

При необхідності нижній рівень системи дозволяє відображати хід процесу збору даних і значення, отримані в результаті їхньої обробки, на екрані монітора персонального комп'ютера. Крім того, забезпечується надання інформації про стан каналів зв'язку і використовуваного для збереження облікової й іншої інформації дискового нагромаджувача.

Верхній рівень системи:

Верхній рівень системи утворений поруч автоматизованих робочих місць, що об'єднані в локальну обчислювальну мережу. До задач верхнього рівня відносяться :

- одержання від нижнього рівня системи облікового даного і даного телеспостереження по наявних каналах зв'язку;
- одержання дані телеспостереження від інших локальних систем обліку відповідно до обговорених протоколів надання даних;
- вторинна обробка отриманих даних відповідно до заданого користувачем алгоритмами;
- ведення бази облікових даних і формування елементів захисту даних комерційного обліку;
- ведення вторинної бази даних для користувачів адміністративного рівня системи;
- формування і відправлення дані телеспостереження на адресу інших систем обліку відповідно до обговорених протоколів надання даних;
- надання користувачам системи дані телеспостереження;
- надання дані телеспостереження користувачам вилучених станцій спостереження;
- формування і підготовка команд керування, їхнє відправлення
- відповідному адресату нижнього рівня з наступним контролем виконання;
- ведення протоколів функціонування системи;

- проведення розрахунків з метою формування документів комерційного і технічного обліку;
- підтримка проведення робіт з перевірки системи, опису вимірювальних каналів системи, опису параметрів системи, у тому числі алгоритмів обробки даних, аналізу роботи каналів зв'язку, їхньої конфігурації і перепризначенню, забезпеченню вірогідності даних, їхньому відновленню, обслуговуванню баз даних;
- формування різних документів, що стосується описи системи, її вимірювальних каналів і параметрів.

Адміністративний рівень системи:

До адміністративного рівня системи умовно відноситься існуюча локальна обчислювальна мережа підприємства. У загальному випадку наявність цього рівня системи необов'язково і визначається існуючою інфраструктурою підприємства. Окремі робочі станції цієї мережі мають можливість :

- вести спостереження за станом об'єктів, що обслуговуються системою;
- одержувати інформацію про стан устаткування системи;
- робити розрахунок інтегральних характеристик роботи об'єктів, що обслуговуються системою.

При наявності на адміністративному рівні системи комунікаційного чи сервера його аналога можливе ведення між мережевого обміну даними телеспостереження й обліку як у виді довільних текстових файлів, так і у виді взаємопогоджуваних протоколів надання даних.

1.3. Структура ієрархічної розподіленої АСУ ТП

Розподілені АСУ, які базуються на телемеханічних комплексах, можуть виконувати завдання диспетчеризації, керування віддаленими об'єктами та збору та передачі інформації з/до віддалених об'єктів. В цьому випадку АСУ будуються як багаторівневі розподілені системи з ієрархічною вузловою структурою. Загальну структуру багаторівневої ієрархічної розподіленої АСУ ТП показано на рис. 1.5.

В ролі вузлів виступають так звані «пункти управління» (ПУ) та «контрольовані пункти» (КП). Верхній рівень АСУ утворений пунктом управління та автоматизованим робочим місцем оператора (та при необхідності, диспетчерським щитом), а нижні рівні утворені мережею контрольованих пунктів, що виконують функції передачі даних від одного вузла до іншого (ретрансляції) та/або взаємодіють з об'єктами керування.

Таким чином, в сучасній ієрархічній АСУ можна виділити 3 рівні:

1) Рівень керування (верхній). На даному рівні виконується загальне керування технологічними процесами (визначення рецептів, керування стадіями процесів, активація / деактивація тих чи інших ділянок чи одиниць обладнання цеху, визначення заданих значень керованих параметрів тощо), що реалізується шляхом формування інформаційних посилок, їх видачі на підлеглі вузли, та отримання посилок-відповідей з підлеглих вузлів. Також на рівні керуючої ЕОМ виконуються задачі виведення алфавітно-цифрової та графічної інформації про стан об'єктів, архівація даних, інша обробка інформації. Як правило, в існуючих телемеханічних системах функції керування виконуються лише на верхньому рівні, до якого зводиться інформація від об'єктів про їх поточний стан. У відповідь з верхнього рівня надсилаються посилки про зміну стану виконавчих механізмів. Представляється доцільним збільшити ефективність роботи АСУ (оперативність реагування на зміни стану об'єктів) шляхом перенесення деяких функцій керування з верхнього рівня до місцевих контролерів, на які надсилати посилки про задані значення та отримувати з них інформацію про об'єкти лише з метою видачі довідникової інформації для оператора.

2) Рівень реалізації інформаційної взаємодії. На даному рівні забезпечується передачі інформаційних посилок (пакетів) від вузлів верхніх рівнів до вузлів нижчих рівнів та у зворотному напрямку. Передача може виконуватися широкосповіщувальним (широкомовним) чином – всі підлеглі вузли отримують розмножену інформацію з верхнього вузла, поки пакет не потрапить у потрібний вузол. Рівень взаємодії з об'єктами. На даному рівні відбувається видача керуючих впливів безпосередньо на виконавчі механізми

об'єктів керування, а також збір (зняття) інформації з датчиків та вимірювальних систем. Також на даному рівні є можливим підключення локальних САК, що мають інтерфейси інформаційної взаємодії (наприклад RS-232) для їх інтеграції в централізовану АСУ.

При цьому сучасним підходом до побудови вузлів (ПУ та КП) є використання набору уніфікованих функціональних блоків або модулів. Таким чином, кожен КП (а також ПУ) має модульну структуру, утворену функціональними модулями, які з'єднані внутрішньою шиною даних. Функціональність вузла забезпечується шляхом комбінації функціональних модулів потрібних типів, що визначається призначенням (функціональністю вузла). Модулі беруться в такій кількості, яка визначається необхідністю реалізувати потрібну кількість каналів введення чи виведення даних (каналів керування). В ПУ та КП, що виступають в ролі проміжних вузлів ретрансляції даних, застосовуються відповідні модулі зв'язку. В КП, які безпосередньо взаємодіють з технічними об'єктами, застосовують модулі взаємодії з об'єктами (введення / виведення аналогової та/або дискретної інформації, спряження по спеціальним інтерфейсам з пристроями, датчиками тощо), а також один або декілька модулів зв'язку. Таким чином, будь-який КП може виконувати або лише задачі передачі інформації до / з підлеглих вузлів (ретрансляції даних), або лише задачі керування об'єктами, або суміщати в собі обидві задачі.

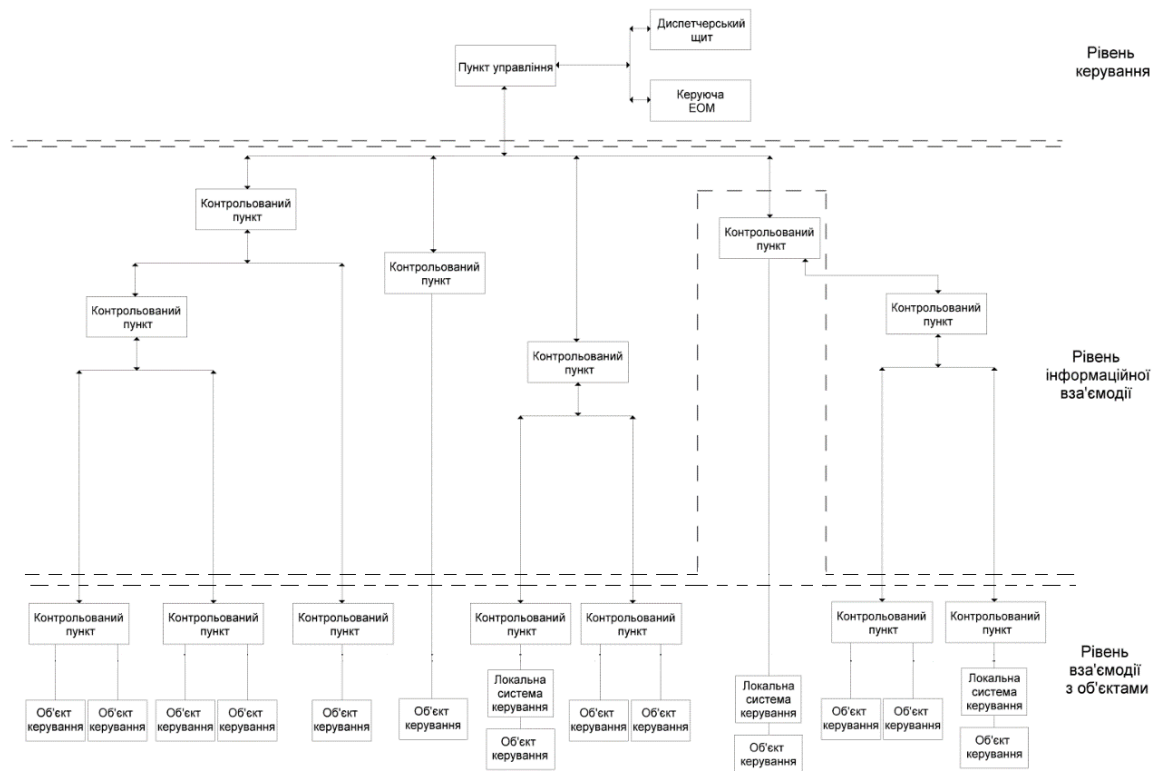


Рис. 1.5. Загальну структура розподіленої АСУ ТП

Контрольований пункт, тобто вузол АСУ, виконаний як модульний комплекс програмно-технічних засобів, призначений для роботи на технологічних об'єктах, вимірювальних вузлах та інших об'єктах автоматизації. Модульний принцип побудови дозволяє створювати на базі вузла систему керування об'єктом відповідно поставленим завданням. Дане рішення дозволяє підвищити ефективність керування, оскільки для кожного технологічного об'єкта можливо індивідуально обрати функціонал відповідного вузла керування, встановивши відповідні функціональні модулі. Загальна модульна структура вузла розподіленої АСУ приведена на рис. 1.6.

Модульність кожного вузла забезпечується їх шинною організацією. В якості поширених шин даних можуть виступати шини I²C або SPI тощо. Шинна організація вузлів дозволяє легко нарощувати їх кількість та забезпечувати додавання вузлів АСУ шляхом нарощення кількості модулів зв'язку у вузлі, до якого треба підключити нові вузли. Таким чином, модифікація такої АСУ (додавання вузлів), що має модульну організацію вузлів, є досить простою і

виконується по місцю розташування найближчого територіально вузла, шляхом зміни кількості його модулів зв'язку.

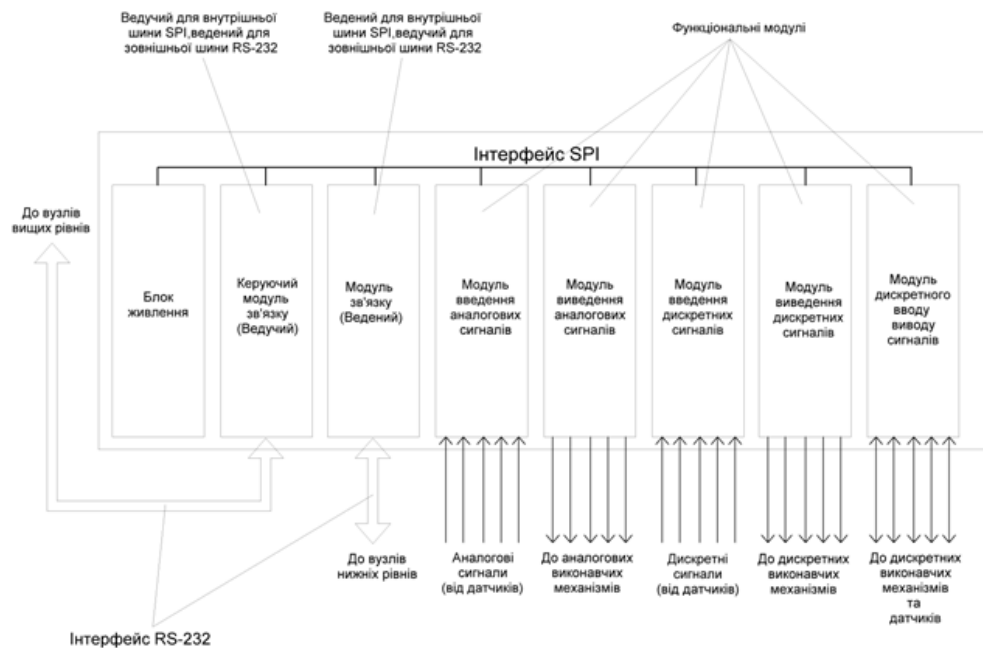


Рис. 1.6. Модульна структура вузла розподіленої АСУ

Обмін між контрольованими пунктами може відбуватися або за допомогою шини (дротової або безпроводного зв'язку), або реалізовуватися через виділені лінії зв'язку. На рис. 1.7. приведено додатковий варіант обміну даними між вузлами АСУ, при якому для обміну із іншим вузлом виділяється окремий порт, що у свою чергу збільшує швидкість обміну даними між нижчими за рівнем вузлами.

Контрольовані пункти можуть обмінюватися за допомогою шини (дротового або безпроводного зв'язку), або вони можуть обмінюватися через спеціально виділені лінії зв'язку. На рис. 1.10 показано ще один спосіб обміну даними між вузлами АСУ. Для обміну даними з іншими вузлами використовується окремий порт, що збільшує швидкість обміну даними між вузлами нижчого рівня.

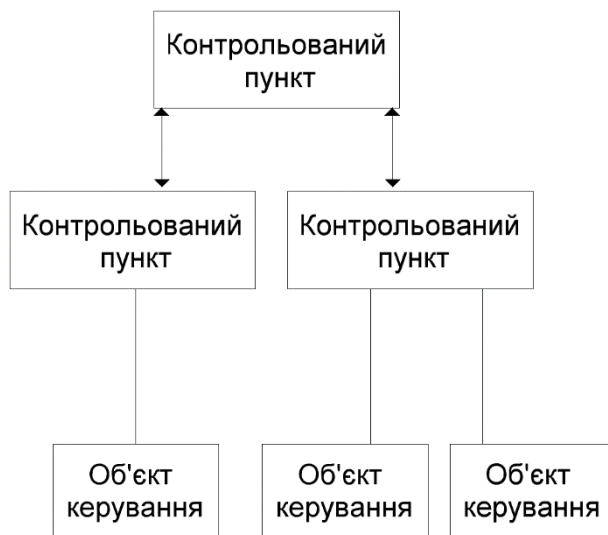


Рис. 1.7. З'єднання головного вузла із вузлами нижчих рівнів

Одним із найбільш поширених способів реалізації виділених провідних ліній зв'язку є використання інтерфейсу RS-232. В такому випадку принцип «точка до точки» використовується для зв'язку кожного підлеглого вузла нижчого рівня з вузлом, вищим за ієрархією. Тому, верхній вузол повинен мати окремі порти для кожного підключеного вузла нижнього рівня. Щоб використовувати модулі зв'язку з одним портом, необхідно встановити певну кількість модулів зв'язку, яка визначається кількістю підлеглих КП.

1.4. Постановка задачі дипломної роботи:

З огляду на те що розподілені АСКТП можуть бути реалізованими за допомогою різних архітектурних підходів:

1) Промислової автоматики блочно-модульного типу – програмованих логічних контролерів (PLC), в тому числі модульного типу, панелей візуалізації та спеціальних модулів (сервоприводів, зважувальних систем тощо), зв'язаних промисловими мережами різних видів;

2) Керуючих ЕОМ (робочих станцій оператора), зв'язаних промисловими мережами та інтерфейсами з об'єктами за допомогою стандартизованих моноблочних модулів-перетворювачів сигналів та інтерфейсів;

3) Телемеханічних систем та комплексів.

В даній роботі буде розглядатись саме принцип побудови талемеханічних систем та комплексів для впровадження даного підходу при виконанні лабораторного стенду дослідження законів керування на основі автоматизованої системи керування температурою у термошафі у вигляді модуля телемеханічної системи АСУТП.

Відповідна система керування має забезпечити наступну функціональність:

- 1) Забезпечувати постійну передачу інформації про поточну температуру в середині термошафи, як в режимі очікування так і під час безпосереднього відтворення роботи законів керування.
- 2) Забезпечувати передачу інформації від керуючого модуля до підлеглого, про зміну параметрів законів керування в реальному часі та про вибір відповідного режиму роботи..

Керованим параметром є в даній системі є температура – T .

Метою даної магістерської роботи є розробка лабораторного стенду дослідження законів керування на основі автоматизованої системи керування температурою у термошафі шляхом розробки мікропроцесорного контролеру (функціонального модуля для вузлів ТМС), що реалізує задачі локального керування – вибору закону керування , налаштування необхідних параметрів, а також має можливість інтеграції з іншими контролерами як модуль ТМС за допомогою відповідних інтерфейсів зв'язку.

Для досягнення поставленої мети необхідно виконати ряд задач:

1. Виконати аналіз Законів керування їх переваги та недоліки, обрати необхідні для подальшої реалізації в установці.
2. Виконати аналіз існуючих принципів побудови розподілених АСУТП та ТМС та комплексів.
3. Обрати підхід до виконання автоматизації керування об'єктом.

4. Розробити функціональну схему системи керування температурою в стенді.
5. Розробити структурну схему системи керування лабораторною установкою, що побудована на основі ТМС.
6. Розробити електричну-принципову схему системи керування лабораторним стендом.
7. Розробити алгоритмічне та програмне забезпечення функціональних модулів ТМС.
8. Практично реалізувати виконані розробки у вигляді лабораторного стенду та реалізувати керування нею за допомогою контролерів на базі апаратно-програмної платформи Arduino.

Наукова новизна: Вперше запропоновано застосувати підхід до побудови розподілених АСУТП у вигляді модульних ТМС для автоматизації процесу керування температурою, в лабораторному стенді дослідження законів керування на основі автоматизованої системи керування у термошафі, де функції оперативного керування перенесено з верхнього рівня на нижні рівні керування до місцевих контролерів (функціональних модулів ТМС), що забезпечує кращу ефективність роботи АСУТП.

Практична цінність даної роботи полягає в:

1. Розробці алгоритмічного та програмного забезпечення мікропроцесорних контролерів для керування об'єктом .Де керування відбувається на основі підходу до побудови розподілених АСУТП у вигляді таких ТМС, в яких функції керування перенесено з верхнього рівня на нижні рівні , до місцевих контролерів. Для забезпечення приросту ефективності роботи АСУТП.

2. Виконані розробки реалізовано у практичному вигляді, а саме лабораторному стенді дослідження законів керування на основі автоматизованої системи керування температурою у термошафі.

2. РОЗРОБКА СТРУКТУРНОЇ СХЕМИ СИСТЕМИ КЕРУВАННЯ

2.1. Протокол передачі даних UART

Universal Asynchronous Receiver-Transmitter -універсальний асинхронний приймач-передавач (UART) може передавати дані без зовнішнього тактового сигналу. Він перетворює дані в послідовний вид, що дозволяє передавати їх по цифровій лінії іншому схожому пристрою. В комп'ютерній техніці метод перетворення широко використовується і добре стандартизований.

Оскільки крім бітів даних інтерфейс UART вставляє в потік біти синхронізації: стартовий і стоповий, для передачі байта інформації потрібно 10 бітів, а не 8.

Передача даних в UART здійснюється по одному біту в рівні проміжки часу. Цей часовий період визначається заданою швидкістю UART і для конкретного з'єднання вказується в бодах. Швидкість передачі визначає, як довго передавач тримає послідовну лінію в високому / низькому стані.

Для передачі байта інформації потрібно 10 бітів, а не 8, оскільки інтерфейс UART вставляє в потік біти синхронізації: стартовий та стоповий.

UART передає дані в рівні проміжки часу по одному біту. Цей період часу визначається заданою швидкістю UART. Те як довго передавач триматиме послідовну лінію у високому чи низькому стані залежить від швидкості передачі.

Важливою вимогою є те, щоб обидва пристрої працювали одночасно. Швидкість передачі 9600 біт/с є однією з найпоширеніших. Логічно, що швидкість передачі корелює з швидкістю прийому та передачі даних.

У кожному пакеті даних є блок даних, який є найважливішим, оскільки містить корисну інформацію. Кожен пакет може містити будь-яку кількість даних, від п'яти до дев'яти біт. Хоча 8-розрядний байт є стандартним розміром даних, можна використовувати будь-які інші розміри.

Через UART обмін інформацією може відбуватися в дуплексному режимі, тобто передача даних може відбуватися одночасно з прийомом. Для цього в інтерфейсі UART є два сигнали:

TX (вихід для передачі даних); RX (вхід для прийому даних).

При з'єднанні двох UART пристроїв вихід TX одного пристрою з'єднується з входом RX іншого. А сигнал TX другого UART підключається до входу RX першого.

Особливості роботи з UART:

- в неактивному режимі вихід UART знаходиться в високому стані;
- передача байта починається зі стартового біта (низького рівня);
- передача байта закінчується стоповим бітом (високого рівня);
- дані передаються молодшим бітом вперед;
- для передачі байта зазвичай потрібно 10 бітів;

У процесі розвитку напрямку створення ТМС було розроблено протокол інформаційного обміну як між окремими модулями ТМС, так і між вузлами (контрольованих пунктів).. Незважаючи на інтерфейс, даний формат може бути використаний для пересилання багатобайтних посилки. Таким чином, у деяких реалізаціях ТМС міжмодульний обмін всередині контрольного пункту може бути здійснений через інтерфейс SPI, UART або RS-485, серед інших, а міжмодульний обмін через інтерфейси RS-232 або RS-485, залежно від ситуації.

Протокол обміну при цьому передбачає 3 рівні формату посилки:

1) Перший рівень визначає посилку фіксованої довжини, яку формує керуюча ЕОМ у вигляді команди, яка має в кінцевому рахунку надійти на конкретний модуль для виконання. Ці команди формуються в залежності від необхідності згідно функціональності кожного конкретного модуля і можуть включати в себе команди ввімкнення (01); вимкнення (02); команди задання заданих керованих значень (коди 1х) для величин, які вимірюються та

опитуються, команди для задання режимів роботи (коди 6х), які відтворюють константи, які не змінюються та не опитуються; команди опитування (2х); команди-відповіді із поточними значеннями (3х), а також інші в залежності від необхідності (наприклад команди для маршрутизації – прив'язки номерів підключених КП до номерів модулів зв'язку). Такі посилки не передаються в цьому вигляді; замість цього вони завжди вкладаються в посилки 2-го або 3-го рівня, які забезпечують адресацію та захисні коди.

2) другий рівень визначає посилку варійованої (змінної) довжини, які передається між вузлами КП, зокрема між підлеглим МЗ верхнього КП та між головним МЗ нижнього залежного КП. Посилка включає два рівня адреси – адресу КП та адресу МЗ, для кого призначена команда. Ці адреси формуються у керуючій ЕОМ, яку вона передає спочатку на самий верхній вузол ТМС. Також посилка включаю в себе «код посилки» та власне дані першого рівня. Також посилка захищається контрольною сумою. Посилка з даними містить 7 байт. Також передбачається посилка фізичного опитування (містить 4 байти – два рівня адреси, команду «опитування» та захисний код). У відповідь на фізичне опитування може бути з нижніх КП передані на верхні КП посилки-відповіді: або така сама 7-ми байтна посилка з даними, або 4-байтна посилка з командою-відповіддю «немає даних».

3) На третьому рівні здійснюється обмін між модулями всередині вузлів. Він дозволяє обмінюватися посилками з однорівневою адресацією в межах певного вузла. Якщо головний модуль вузла отримує дані, призначені для модулів вузла з верхнього рівня, отримана посилка упаковується у формат у формат посилки-даних (10 байт), тобто посилки-даних, що містить захисний код, адресу конкретного модуля вузла та команду «дані». Якщо головний модуль вузла отримує дані для модулів іншого вузла з верхнього рівня, ця посилка упаковується у формат посилки-даних розміром десять байт і надсилається на підлеглий модуль зв'язку, до якого підключений даний вузол.

Адреса модуля отримується з таблиці маршрутизації. Таким чином, до посилки додається захисний код, команда «дані» та адреса конкретного модуля

роботи, необхідних параметрів (що закладені в програмну частину) та виведення інформації про поточний режим роботи та температуру.

Для роботи людино-машинного інтерфейсу, або ж пульта керування, використовується модуль Keypad Shield LCD. Модуль механічно встановлюється згори процесорного модуля, що дозволяє з'єднати керуючі сигнали з цифровим портом, а частина виводів модуля служить для підключення шини даних до індикаторного модуля. Клавіатура інтерфейсного модуля передає аналоговий сигнал на аналоговий порт A0.

Всі керуючі та комутаційні елементи схеми, а також більшість виконавчих пристроїв живляться блоком живлення 12В від промислової мережі змінного струму 220В. Блок живлення також подає живлення вентиляторам та елементу охолодження, також відомому як елемент Пельтьє. Нагрівальний елемент, в якості якого використовується лампа розжарювання, також живиться від промислової мережі змінного струму.

Процесорний модуль отримує живлення від блоку живлення 12В за допомогою додаткового конвертора, який зменшує напругу до 9В [9].

Інтерфейсний модуль і модуль реле отримують загальне живлення від процесорного модуля через відповідні порти живлення.

Загальний блок живлення 12В забезпечує живлення вентиляторів. Елемент Пельтьє також живиться від цього блоку живлення, але за допомогою додаткового конвертора [9], який знижує напругу до 9В.

Цифровий датчик DS18B20, який позначений D1, підключений до дискретного порту 2. Інфрачервоний дискретний датчик перепони D2, підключений до одного з виводів цифрового порту, виконує функцію визначення стану дверей (відкрита-закрита).

Комутаційні релейні модулі також отримують керуючі сигнали через цифрові порти процесорного модуля.

2.3 Висновки до розділу 2

В даному розділі було проаналізовано та описано роботу протоколу передачі даних UART його особливості та рівні формату посилок. Спираючись на всі переваги даного протоколу, було прийнято рішення використовувати саме цей інтерфейс у даній роботі . Було також розроблено структурну схему роботи лабораторного стенду із застосуванням даного інтерфейсу.

3. РОЗРОБКА ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ СХЕМИ СИСТЕМИ КЕРУВАННЯ

3.1. Опис елементної бази

3.1.1. Апаратно програмна платформа Arduino.

Апаратно програмна платформа Arduino на сьогоднішній день налічує в своєму арсеналі великий асортимент контролерів, що можуть забезпечувати найрізноманітніший функціонал. Серед найбільш поширених та використовуваних виділяються Arduino Micro, Arduino Mini, Arduino Nano що отримали свою популярність завдяки низькій вартості та простоті в роботі.

Але в подальшій роботі буде розглядатися плата Arduino UNO (рис.3.1) як найбільш оптимальна для досягнення поставлених цілей.

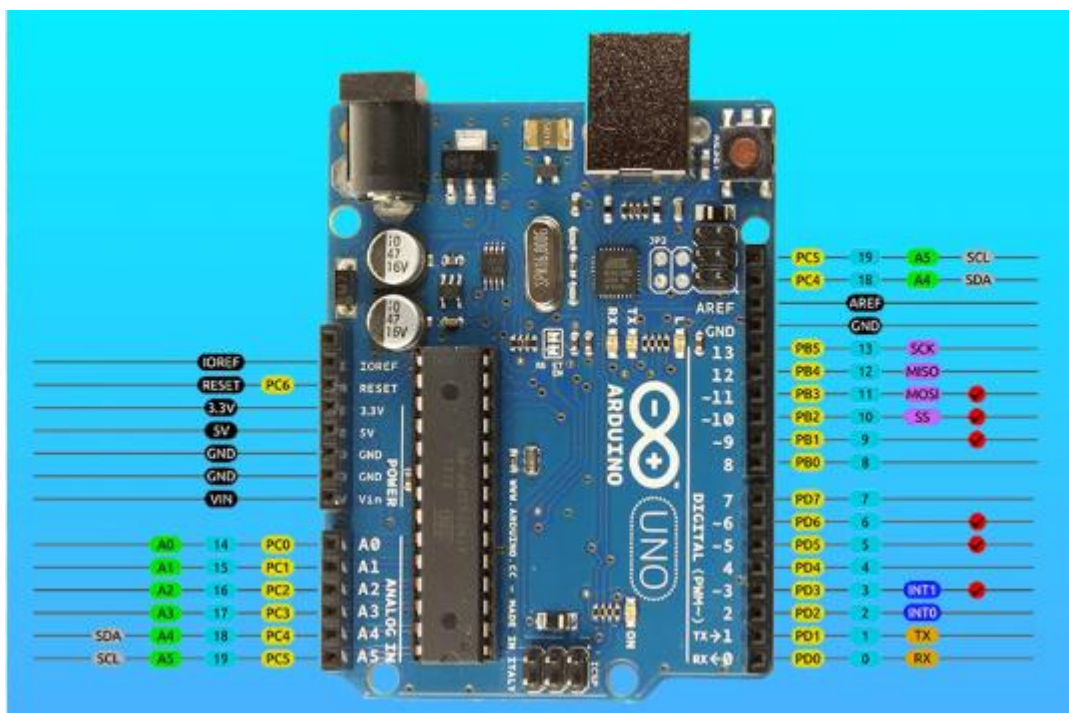


Рис. 3.1. Загальний вигляд плати Arduino Uno

Arduino Uno — це плата, що має в своїй основі мікроконтролер ATmega328P. Вона містить все, що потрібно для зручної роботи з мікроконтролером: 14 цифрових входів/виходів (6 з них можуть використовуватися як ШИМ-виходи), 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм для живлення, роз'єм для внутрішнього схемного

програмування (ICSP) і кнопка скидання. Для того, щоб пристрій міг працювати, достатньо підключити його до комп'ютера за допомогою кабелю USB або AC/DC-адаптера або батарейки.

Плата має розміри 6,9 на 5,3 см. За межі плати на декілька міліметрів виступають гніздо живлення та USB, також на платі передбачені спеціальні місця для кріплення.

В платі реалізовано підтримку декількох інтерфейсів для обміну даними:

1. UART або Universal Asynchronous Receiver-Transmitter .

Під нього виділені піни RX(D0),TX(D1). Через нього також відбувається програмування плати.

2. SPI або ж Serial Peripheral Interface, що призначений для швидкісної послідовної передачі даних між платами Arduino або іншою периферійними пристроями.

Під нього виділено піни:

MOSI(D11), MISO(D12), SCK(D13), SS(D10)

3. I²C або Inter-Integrated Circuit

Піни SDA(A4), SCL(A5).

3.3.2. LCD Keypad Shield

На рис. 3.2 показано, як виглядає даний модуль, що встановлюється прямо на плату Arduino, щоб розширити вихід і введення даних. Він використовує LCD-дисплей TC1602 для індикації даних, а вбудовані кнопки забезпечують введення та керування. Крім того, на модулі є потенціометр, який дозволяє змінювати яскравість дисплею.

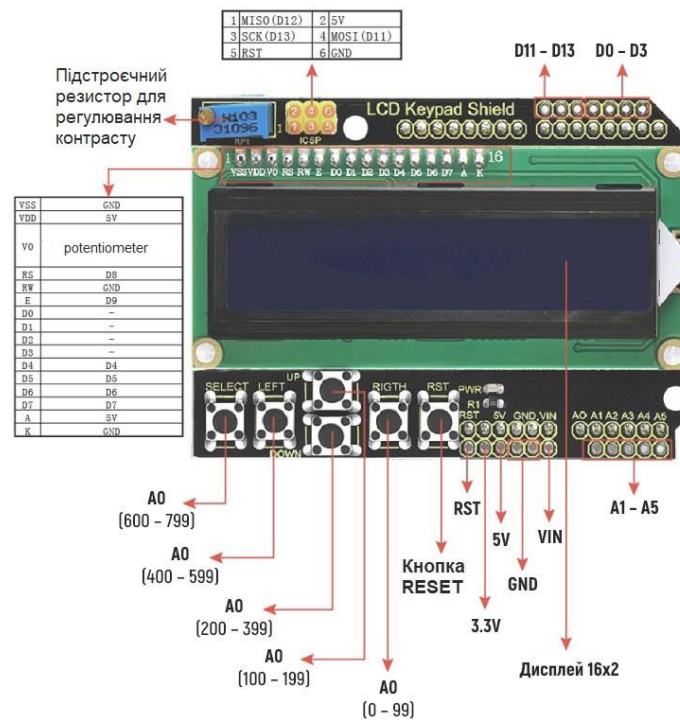


Рис. 3.2. LCD Keypad Shield та його виводи

Роз'єми на платі дозволяють підключати різні пристрої та датчики. Для роботи з екраном використовуються піни з 4 по 10. Один аналоговий пін A0 використовується для визначення натискання кнопок. Цифрові піни 0-3, 11-13 і аналогові піни A1-A5 є вільними.

Опис виводів дисплею також наведений в на рис.3.3.

Перевагами використання даного модуля є його швидке монтування, що не потребує пайки, економія аналогових пінів під підключення кнопок та доволі широкі можливості застосування в якості пульта керування та відображення інформації про стан виконавчих механізмів та датчиків.

До недоліків можна віднести те що хоч і економиться певна кількість аналогових виходів, але разом з цим зменшується кількість цифрових.

3.3.3. Датчики

Датчик температури DS18B20 цифровий. Датчик може бути виконаний в різних варіаціях. У цьому конкретному прикладі у водонепроникному корпусі (рис. 3.3). Він може вимірювати температуру з точністю $\pm 0,5$ °C у діапазоні, що

не перевищує значення від від -10 до 85 °С. Але загальні межі вимірювань лежать від -55 до 125 °С. Вимірювальний елемент розташований всередині герметичного корпусу, що дозволяє вимірювати температуру при повному зануренні в рідину та в будь-яких умовах вологості та запиленості. Трьома провідниками здійснюється під'єднання. Для виведення даних використовується жовтий провід, червоний провід живлення, чорний провідник приєднується до землі. Крім того, для нормальної роботи датчика необхідно підключити резистор 4,7 кОм. до виводу живлення та лінії передачі даних. Якщо потрібно працювати з даними датчиками за допомогою інтерфейсу Arduino, також можливо підключити бібліотеку OneWire. Технічні характеристики представлені нижче таб.3.1.

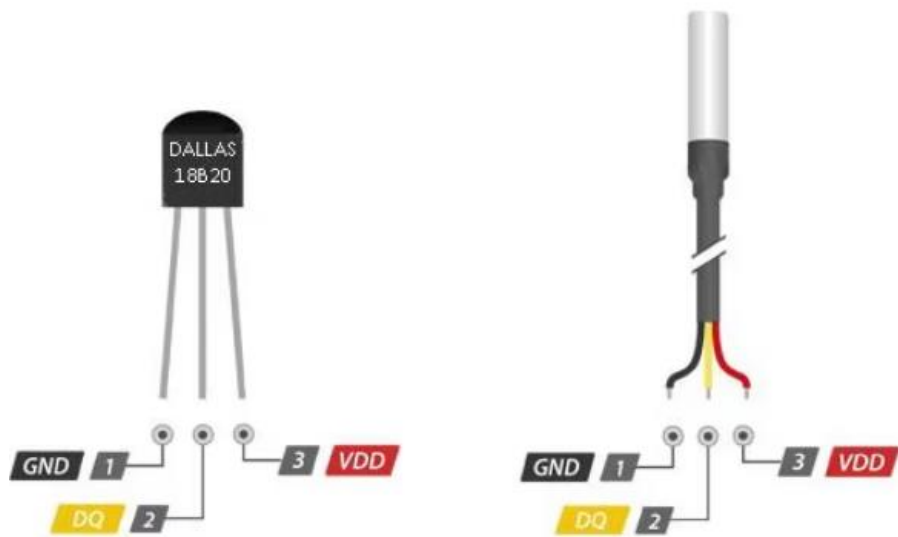


Рис. 3.3.Різні форм фактори DS18B20 під різні задачі

Таблица 3.1

Технічні характеристики датчика температури в порівнянні з іншою моделлю :

Маркування	DS1820	DS18B20
Корпус	TO-92	TO-92
Розрядність	9-біт	9-12біт

Час перетворення	750nS (max)	750nS (max)
Точність вимірювання $\pm 0,5\%$	-10+85°C	-10+85°C
Напруга живлення для точності вимірювання $\pm 0.5\%$	3,0-5,5V	3,0-5,5V

Датчик інфрачервоний датчик перешкод— використовується в проектах на контролерах, для обходу перешкод роботом або визначення перешкод, контролю габаритів, оптичного вимикача світла та інших пристроїв.

Модуль працює таким чином: інфрачервоний світлодіод випускає невидимі інфрачервоні промені. Якщо на шляху променю виникає перешкода, промінь відбивається від перешкоди і потрапляє на фототранзистор. Який у свою чергу, залежно від інтенсивності випромінювання та налаштувань компаратора відкривається. І це дозволяє сенсору виявити перешкоду. Його зображення приведені на. (рис 3.4). Технічні характеристики наведені в табл.3.2.



Рис.3.4. Інфрачервоний датчик перешкод

Таблиця 3.2

Технічні характеристики датчику перешкод

Живлення	Від 3,3 до 5 В
Дальність	Від 2 до 80 см

Розміри	31 14 x 7 мм
---------	--------------

3.3.3. Блок живлення та конвертори напруги

Даний тип блоків живлення призначений для живлення різних електричних пристроїв. В лінійці блоків представлені різні варіації різною кількістю виводів, та різною вихідною напругою від 12 до 24 В. Даний блок представлений на рис. 3.5.



Рис.3.5. Блок живлення S-120-12

В технічних характеристиках даного приладу вказана потужність, що становить 120 Вт. Вихідна напруга рівна 12В. В блоці реалізовано захист від короткого замикання і перевантаження. Даний блок рекомендується використовувати в захищених конструкціях або приміщеннях. Ступінь захисту по волості IP20

Понижувальний DC-DC конвертор на XL4015CC CV зображений на рис. 3.6.

Даний DC-DC конвертор XL4015 - призначений для перетворення постійної вхідної напруги сигналу на постійну вихідну напругу сигналу. Цей перетворювач може виробляти регульовану вихідну напругу в діапазоні від 1,25 до 32 Вольт і регульований струм до 5А. При навантаженні, у якому сила струму перевищує 4А, потрібно встановити радіатор на корпус мікросхеми або

плату перетворювача. Дотримуючись ізоляції струмопровідних частин від тепловідведення.

Мікросхема має вбудований захист від коротких замикань та перегріву. Захист спрацьовує при перевищенні температури в 125°C. Модуль також має обмеження по струму, що спрацьовує при 8А. Таблиця 2.6 містить технічні характеристики.

Переваги модуля включають: високу ефективність, широкий діапазон вхідної напруги і можливість регулювання напруги та струму, індикація включення та процесу роботи.

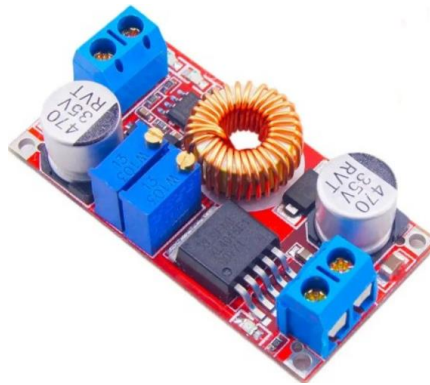


Рис.3.6. Понижуючий DC-DC перетворювач на XL4015CC

3.3.4. Виконавчі механізми

Механічні модулі реле можуть бути виконані як одноманітній компоновці так і містити на одній платі надзвичайно велику кількість реле.

1-канальний модуль реле 5В - це модуль, що як показано на рис. 3.7, містить реле та відповідні виводи для керування ним. Модуль може працювати з різними пристроями, які потребують сигналів контролера для керування.

У представленому модулі є лише один канал реле, що дозволяє йому керувати одним навантаженням. Він працює з напругою 5 Вольт і містить два контакти живлення та один керуючий вивід. Технічні характеристики модуля наведені в табл. 3.3.

В залежності від компонування модулів на них можливе розміщення оптоізоляції, тобто він може бути незахищений від сторонніх впливів таких як

перенапруга, або перевантаження. Параметри що стосуються виводів модуля не повинні перевищувати вказаних на корпусі реле.

Модуль реле з двома каналами використовується для керування пристроями з великими вхідними струмами (рис. 3.7). Для того, щоб модуль міг працювати, на нього потрібно підключити живлення, а також на кожне реле окремо подавати керуючі сигнали. Замикання реле вимагає низького логічного рівня на керований вхід. Таблиця 2.14 містить технічні характеристики. Рис. 2.15 показує електричну схему двоканального модуля реле.

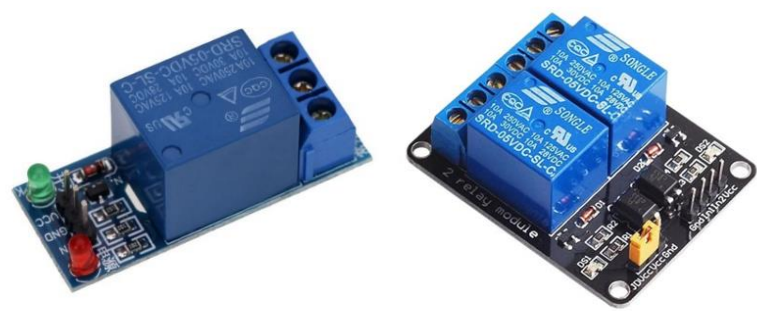


Рис. 3.7. Одноканальний та двоканальний модулі реле

Таблиця 3.3

Технічні характеристики 1- та 2-канального модулів реле

Живлення	5В	5В
Сигнал увімкнення	Логічний нуль/одиниця В залежності від модуля	Логічний нуль/одиниця В залежності від модуля
Максимальна напруга	250 В	250 В
Розмір	43x17x19 мм	50x40мм

Модулі з використанням твердотільного реле

Даний 1 канальний модуль призначений для комутації змінної напруги, (рис.3.8).

Його відмінною особливістю є повна відсутність рухомих контактів, завдяки цьому досягається безшумність та надійність роботи. Для керування даним модулем необхідно подати на виводи цифровий сигнал з мікроконтролера, а також живлення та землю. Технічні характеристики наведені в таблиці 3.4.

Модуль 2-х каналного твердотільного реле побудований на G3MB-202P і призначений для комутації змінної напруги. Його відмінною особливістю є повне відрізання рухомих контактів і контактів взагалі, що робить реле надзвичайно надійним і безшумним. Щоб керувати реле, достатньо подати на нього цифровий сигнал з мікроконтролера.

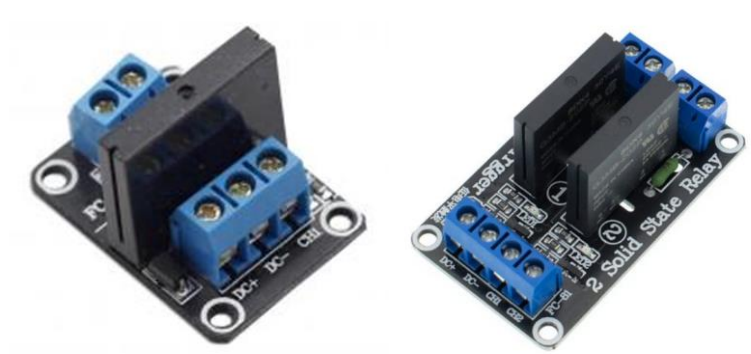


Рис.3.8. Вигляд1 та 2 каналного модулю твердотільного реле

Таблиця 3.4

Технічні характеристики 1 та 2 каналного модулів

Живлення	5В	від 0 до 5В
Сигнал увімкнення	Логічний нуль/одиниця В залежності від модуля	Логічний нуль/одиниця В залежності від модуля
Максимальна напруга	250 В	Від 100 до 240 В
Розмір	57x53x25 мм	55 x 33 x 25 мм
Кількість реле	1	2

3.2. Розробка електричної принципової схеми системи керування

На основі структурної схеми та елементної бази виконано розробку електричної принципової схеми системи керування стендом.

Електронна принципова схема системи керування дає можливість аналізувати роботу системи та дає повне уявлення про всі її компоненти та зв'язки. Основні елементи підписані, та мають відповідні підписані контакти. Електронна принципова схема представлена на рисунку 3.9 і в додатку Б. Перелік елементів міститься в додатку Е.

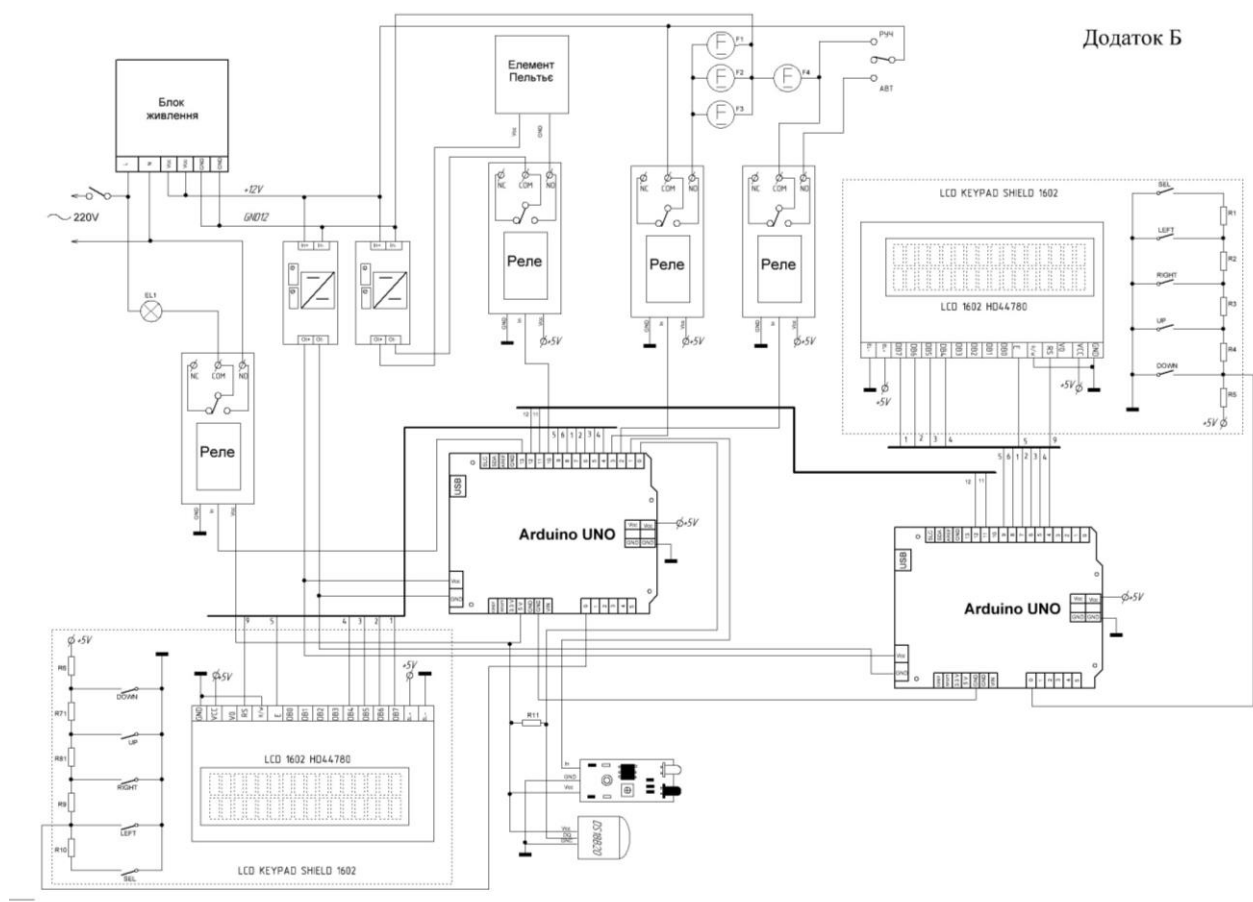


Рис. 3.9. Схема електрична принципова системи керування

Модуль Keypad Shield LCD, який може виконувати функцію панелі оператора, під'єднаний до плати Arduino UNO. На цьому модулі є рідкокристалічний екран, на який виводяться дані з датчиків, а також меню, яке використовується для налаштування параметрів і видачі впливів на виконавчі механізми.

Дана плата Arduino UNO виконує функцію керуючої, на ній відбувається задання параметрів та вибір режимів роботи. Вже з неї через інтерфейс UART відбувається передача відповідної інформації до підлеглої плати, що відповідає за безпосередню видачу керуючих впливів та передачу інформації про об'єкт на Master плату.

Для підключення установки до мережі 220 Вольт схема містить вмикач. Основний модуль реле керує нагрівальним елементом, який, у цьому випадку, є лампою розжарювання, для більшості режимів роботи. Вентилятори для охолодження, вентилятори елементу Пельтьє та сам елемент вмикаються блоками реле.

Поряд з основним вентилятором охолодження знаходиться перемикач, який дозволяє використовувати вентилятор як вручну, так і автоматично. або взагалі не вмикати його.

Блоки реле працюють за допомогою живлення 5 В і виводу GND. що береться з виводів плати Arduino UNO. Для роботи реле окремі порти плати підключені до виводів подачі сигналу. 7 цифровий вивід відповідає за включення основного нагрівача, 6 вивід контролює включення бокового вентилятора охолодження, 5 вивід відповідає за вентилятори елементу Пельтьє, а 4 цифровий вивід вмикає сам елемент.

Датчик відкриття дверей підключений до 3 цифрового виводу плати. Датчик температури підключається до виводу D2.

3.3. Висновки по розділу 3

В даному розділі дипломної роботи було виконано опис елементної бази, що використовується в лабораторному стенді. Описані характеристики елементів, та виконано розробку самої електричної принципової схеми системи керування.

4. РОЗРОБКА АЛГОРИТМІВ РОБОТИ СИСТЕМИ

4.1. Опис принципу роботи установки та реалізованих режимів роботи

Наразі керування температурою термошафи працює в двох режимах. Режим 1 використовує релейний закон керування, а режим 2 використовує пропорційний закон керування та пропорційно інтегральний закон. У режимі 1 є чотири підрежими, а в режимі 2 — три підрежими, що разом дає сім режимів роботи установки. Режими 1.1-1.3 реалізують релейні закони для задачі нагрівання об'єкта. Режим 1.4 реалізує релейний закон з керуванням температурою в сторону охолодження.

Режими 2.1 та 2.2 реалізують пропорційні закони керування для задачі нагрівання об'єкта. А режим 2.3 реалізує пропорційно-інтегральний закон керування.

При цьому виконавчий пристрій може працювати в нереверсивному (вмикається або не вмикається нагрівальний елемент) або реверсивному (вмикається або нагрівальний елемент, або бічний вентилятор охолодження) режимах відповідно 1.1-1.3 та 2.1-2.2. Додатковий апаратний перемикач обирає режим роботи вентилятора. Це дозволяє відключати бічний вентилятор, керувати ним від контролера (для реверсивного варіанту), або вмикати примусову роботу для швидкого охолодження установки або задання постійно діючого збурення.

Далі більш детально представлено всі реалізовані режими.

Перший підрежим — реалізує релейний закон керування з симетричним гістерезисом, де параметрами, що задаються є задана температура T , а також величина гістерезису Δ , в цьому випадку інтервал керування становить від $T - \Delta/2$ до $T + \Delta/2$.

Підрежим два — реалізує релейний закон керування з несиметричним гістерезисом, де параметрами, що задаються є задана температура T , а також

величина гістерезису G , в цьому випадку інтервал керування становить від $T-G$ до T .

Підрежим три – реалізує релейний закон керування з гістерезисом, величина якого задається неявно через граничні межі температури $T_{\min}$ та $T_{\max}$, параметрами, що задаються є граничні межі температури $T_{\min}$ та $T_{\max}$, в цьому випадку інтервал керування становить, очевидно, від $T_{\min}$ до $T_{\max}$.

Підрежим чотири– реалізує релейний закон керування з несиметричним гістерезисом, але з керуванням температурою в сторону охолодження;

Елемент Пельтьє та його відповідні задні вентилятори служать виконавчими пристроями, щоб охолодити гарячу сторону елемента. Параметри, які задаються, це задана температура охолодження T_r (нижче якої вмикається охолодження) і максимальна допустима температура $T_{\max}$ (вище якої вмикається охолодження). У цьому випадку інтервал керування становить від T_r до $T_r+T_{\max}$.

Підрежим 2.1 та 2.2 реалізують пропорційний закон керування за наступною логікою:

$$\begin{aligned}\Delta t &= T_z - T_n, \\ U &= K \cdot \Delta t,\end{aligned}\tag{4.1}$$

де Δt – різниця заданого та поточного значення температури;

T_z – задане значення температури;

T_n – поточне значення температури;

K – коефіцієнт пропорційності для пропорційного закону керування;

U – розрахована величина сигналу керування.

Широтно-імпульсна модуляція (ШІМ) використовується для реалізації пропорційного закону керування на цифровому виході процесорного модуля. Для наглядності ШІМ реалізується програмно з періодом модуляції в декілька секунд, який задається на пульті. Багато промислових контролерів мають подібні режими.

Логіка ШІМ модуляції виконується за очевидною формулою:

$$\begin{aligned} t_{on} &= Ct \frac{U}{U_{\max}} = Ct \frac{K \cdot \Delta t}{U_{\max}}, \\ t_{off} &= Ct - t_{on}, \end{aligned} \quad (4.2)$$

де t_{on} – час ввімкненого стану виконавчого пристрою;

t_{off} – час вимкненого стану виконавчого пристрою;

Ct – період модуляції;

U – розрахована величина сигналу керування;

$U_{\max}$ – умовна максимальна величина сигналу керування;

K – коефіцієнт пропорційності для пропорційного закону керування;

Δt – різниця заданого та поточного значення температури.

При цьому в режимі 2.1 забезпечується гарантоване ввімкнення виконавчого пристрою не менше, ніж на 1 с, та не більше, ніж на $(Ct-1)$ с, кожен період модуляції.

В режимі 2.2 виконавчий пристрій включається, якщо його розрахований час ввімкнення t_{on} не менше 1 с. Інакше активація не відбудеться. Відповідно до кожного періоду модуляції, максимальний час ввімкнення (за умови виконання умови ввімкнення) також обмежений величиною $(Ct-1)$. У цьому режимі можна досліджувати статичну помилку системи за допомогою значення параметрів закону керування.

4.2. Вигляд інтерфейсу меню на керуючому модулі

Взаємодія з інтерфейсом меню відбувається за допомогою кнопок, розташованих на модулі LCD Keypad Shield. При запуску на екран виводиться перша сторінка меню (рис.4.1).



Рис. 4.1. Вигляд першої вкладки меню при увімкненні

Для переходу з першої вкладки меню до вибору між режимами роботи , і наступними вкладками використовується кнопка Select. Для переходу між режимами та між параметрами при виборі налаштувань використовуються кнопки UP та DOWN, для повернення назад потрібно використовувати кнопку Left.



Рис. 4.2. Вигляд меню вибору режиму роботи

Коли відбувся вибір режиму, відбувається перехід до запуску встановленого за замовчуванням підрежиму або вибору налаштування (рис.4.3).



Рис. 4.3. Вигляд меню при входженні в перший режим роботи

Вибір вкладки налаштування переносить спочатку до вибору підрежиму роботи, а при ще одному натисканні кнопки Select. До налаштування параметрів підрежиму. При виборі підрежиму переключення відбувається кнопками Right, Left (рис.4.4).



Рис. 3.4. Вигляд меню при обранні підрежиму

При налаштування параметрів натиснувши кнопку Select одразу відбувається перехід до меню з вибором запуску, або налаштувань (рис.4.5).

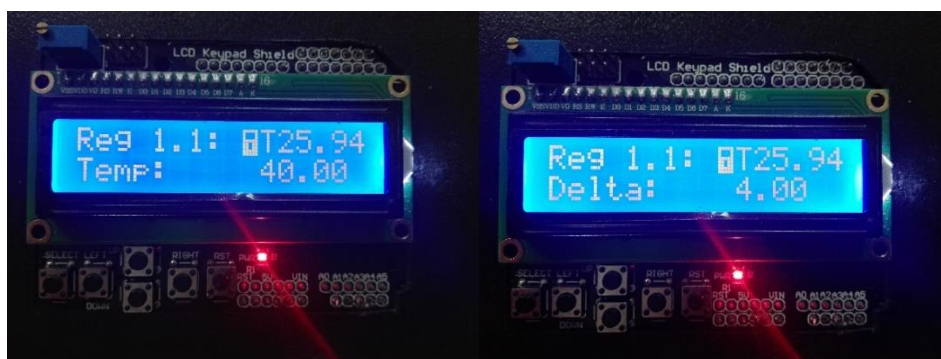


Рис. 4.5. Зовнішній вигляд налаштування параметрів режиму 1.1

При виборі налаштувань другого режиму все відбувається аналогічно першому (рис.4.6.-4.7).



Рис. 4.6. Вигляд меню при обранні налаштувань підрежиму 2

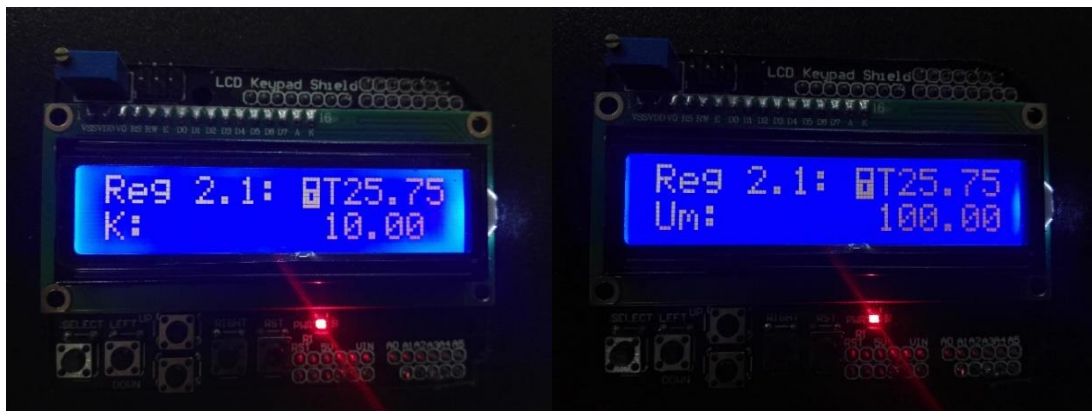


Рис. 4.7. Вигляд меню в режимі налаштувань підрежиму 2.1

При виборі вкладки Start на екран виводиться інформація про поточний підрежим, поточний параметр датчика і задане значення температури що досягається (рис.4.8).



Рис. 4.8. Зовнішній вигляд меню під час роботи в одному з обраних режимів

Алгоритмічне забезпечення роботи контролера представлено структурно – у вигляді загального алгоритму та його окремих фрагментів. З огляду на об'ємність алгоритмів, вони представлені в додатках В, Г та Д.

Алгоритм основного циклу роботи контролера (рис. 4.9) являє собою нескінченний цикл, що продовжується, поки подається живлення на контролер.

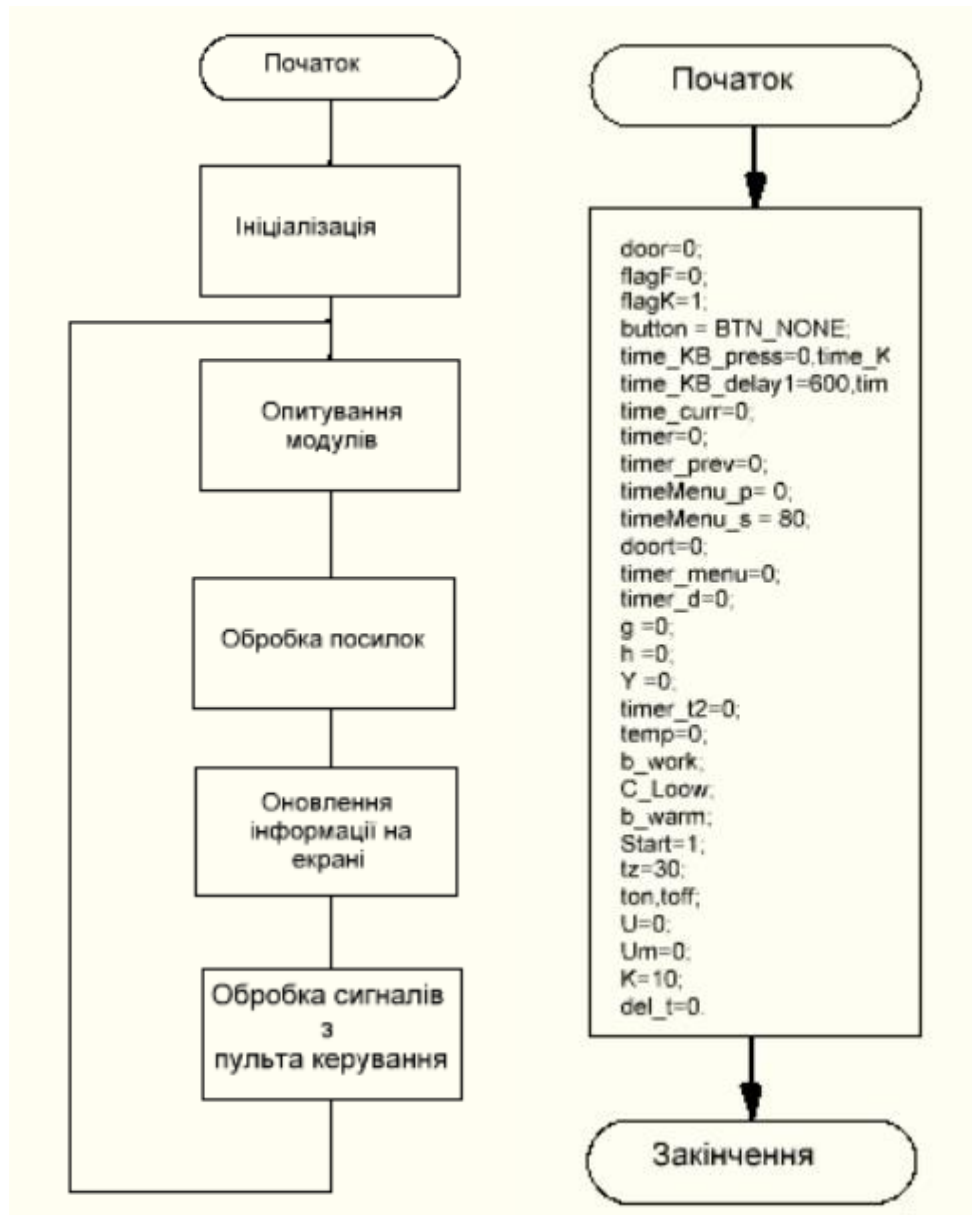


Рис. 4.9. Основний алгоритм роботи та алгоритм ініціалізації основних змінних

Робота алгоритму починається з ініціалізації, яка відбувається лише раз, коли контролер вмикається. Прапорець `bWork` стає рівним нулю під час ініціалізації і використовується для відстеження ввімкнення режиму керування.

У випадку запуску певного режиму роботи в підпрограмі роботи з пультом керування прапорець ставиться в 1. також вимикаються всі виконавчі пристрої (за ввімкнення нагрівача та бічного вентилятора відповідають біти dWarm, dCool відповідно, аналогічно виконуються й інші дії – вимкнення елемента Пельтьє та заднього вентилятора). Також ініціалізується поточна змінна часу time. Крім того, виконується ініціалізація і інших змінних, необхідних для роботи з пультом керування.

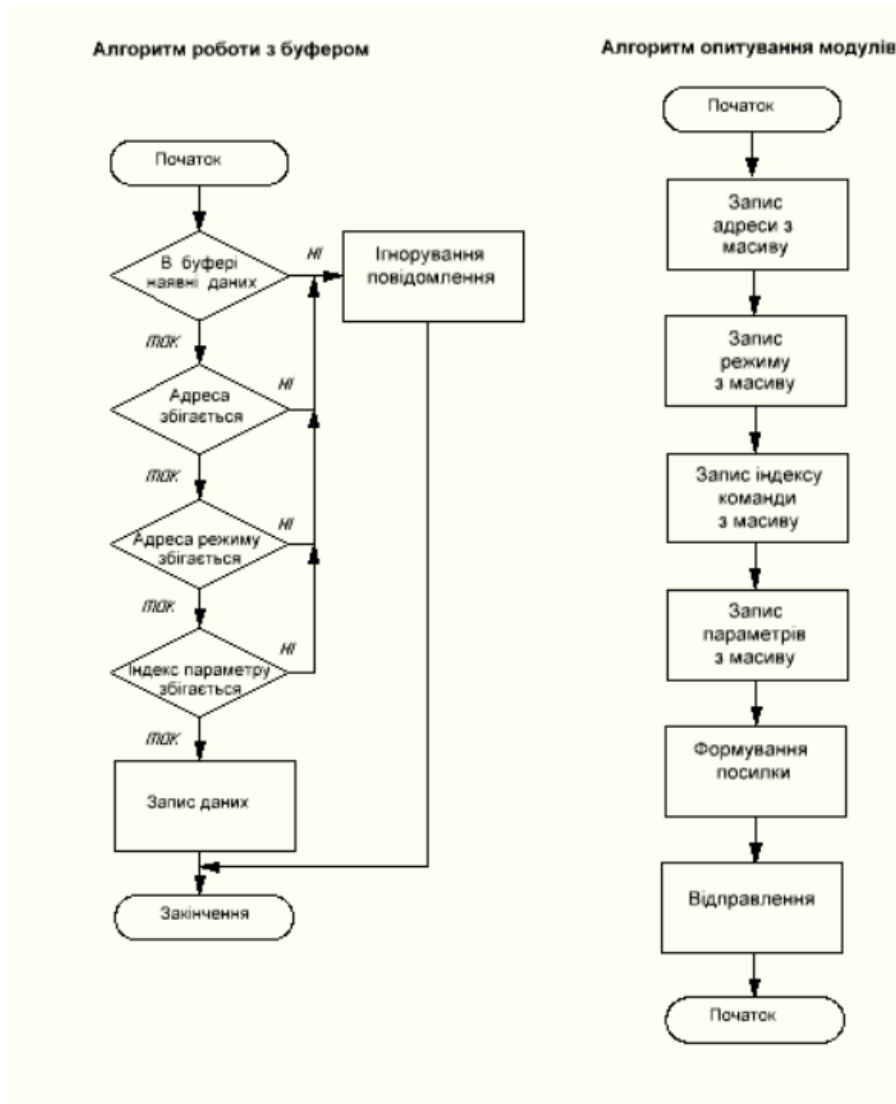


Рис. 4.10. Алгоритм роботи з буфером та опитування модулів

Після ініціалізації відбувається перехід в нескінченний цикл, що відповідає за виконання основних задач: опитування підключених модулів, відправки запиту на отримання поточного значення температури .

На рис. 4.10 зображено цикл роботи підлеглого модуля, що відповідає за виконання основної задачі керування в залежності від обраного режиму роботи. Якщо режим роботи не ввімкнено (прапорець $bWork=0$), усі виконавчі пристрої гарантовано вимкнені. Якщо це не так, виконується задача керування відповідно до обраного підрежиму роботи.

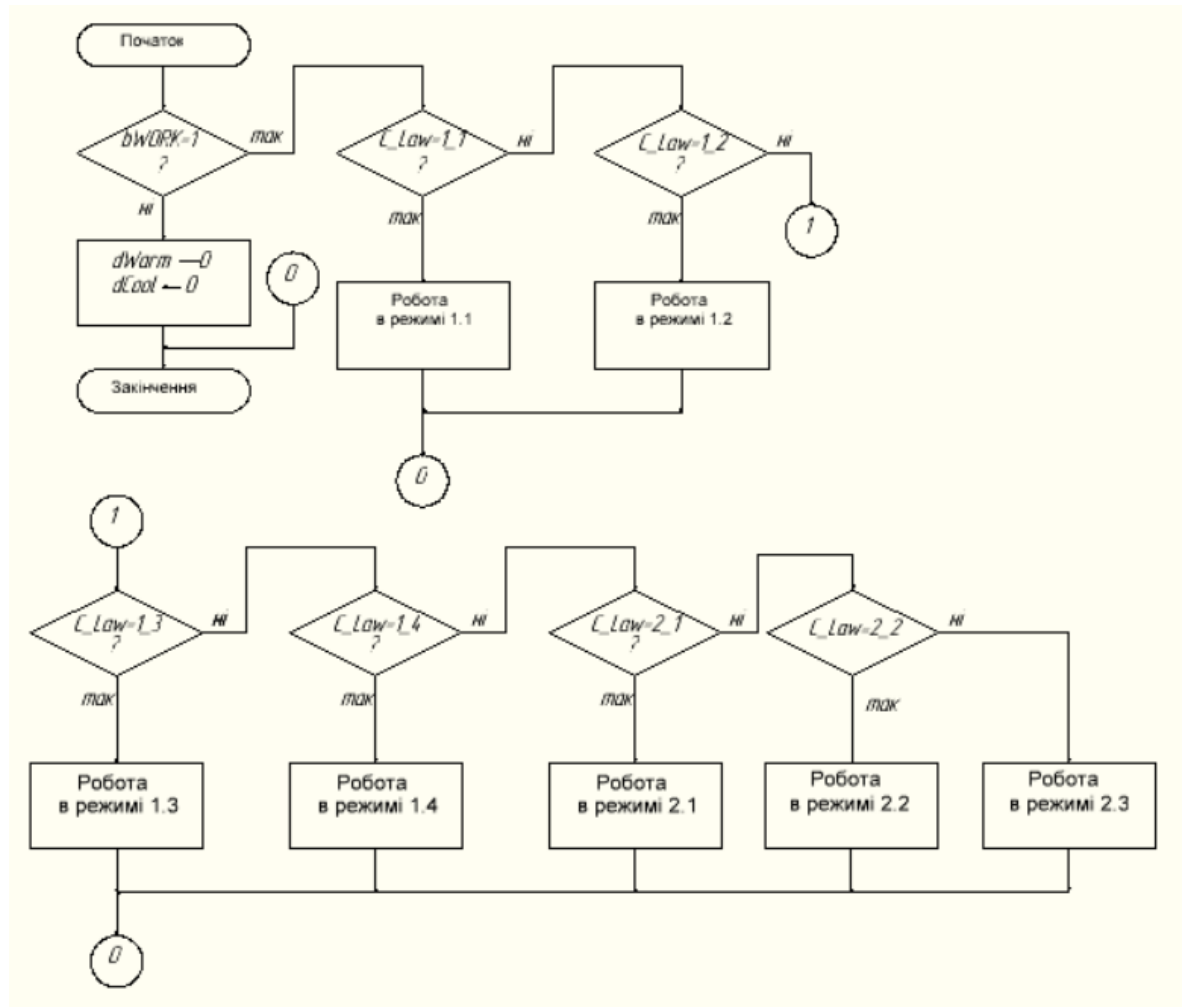


Рис. 4.11. Алгоритм обрання режиму роботи

Алгоритми роботи окремих підрежимів показані на рис. 4.12-4.13.

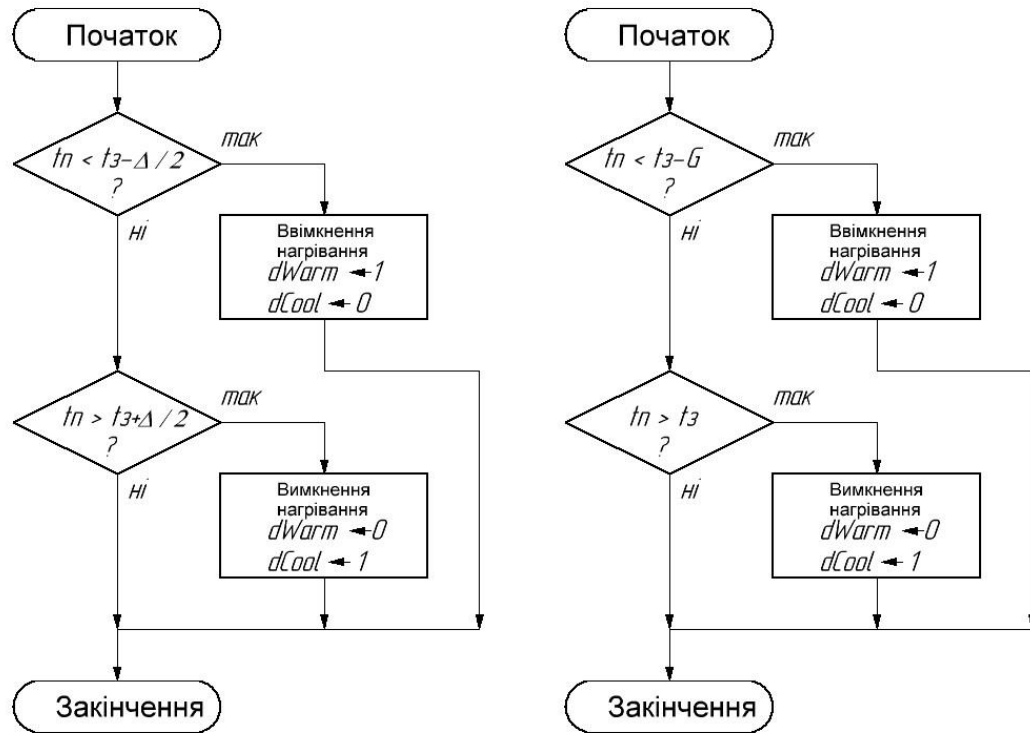


Рис. 4.12. Алгоритм першого та другого режиму роботи

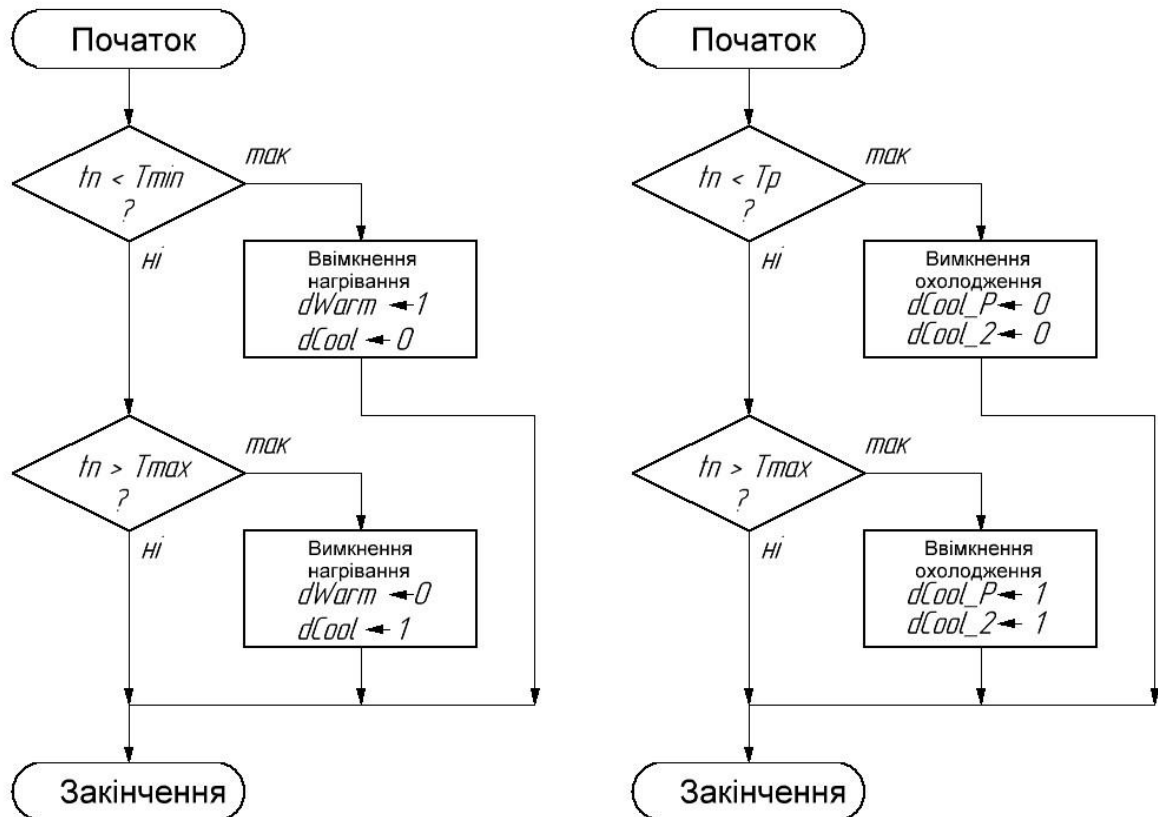


Рис. 4.13. Алгоритм третього та четвертого підрежиму роботи

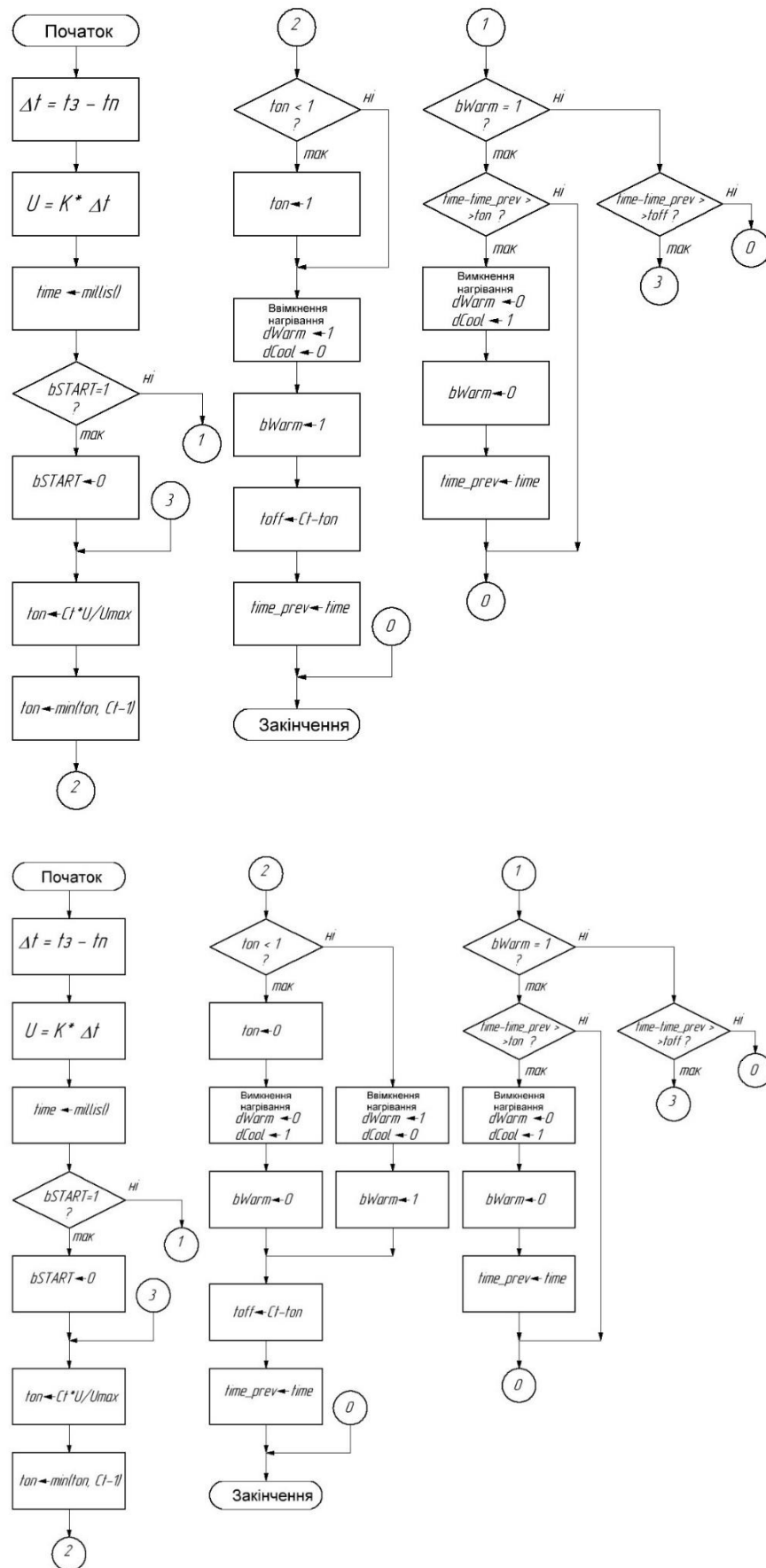


Рис. 4.14. Алгоритм роботи в підрежимі 2.1 та підрежимі 2.2

4.3. Програмне забезпечення з відкритим кодом Arduino IDE

Програмне середовище Arduino IDE буде використано для поставленої мети та задачі. В цьому середовищі є всі необхідні для роботи функції. Починаючи від великої кількості бібліотек і закінчуючи простотою та зрозумілістю інтерфейсу. Це програмне забезпечення є мультиплатформним і досить просте у використанні. На рис.4.16. представлено вигляд основного вікна програми. В залежності від версії воно може трохи відрізнятись.



Рис. 4.16. Середовище розробки Arduino IDE

Завжди при кожному новому сеансі роботи з програмою, при підключенні плати до комп'ютера необхідно перевіряти чи правильно встановлено номер COM-порту і чи обрано відповідну плату Arduino із загального списку представлених пристроїв. Для перевірки необхідно перейти на вкладку “Tools” зверху панелі інструментів. Та перевірити вкладки “Port” та “Board”. Якщо все вказано правильно продовжувати працювати.

4.4. Розробка програмного забезпечення

На основі алгоритмічного забезпечення було написано програмний код, що забезпечує роботу програми з реалізацією необхідних функцій

На початку роботи програми відбувається підключення необхідних бібліотек, ініціалізація глобальних змінних та констант, оголошуються стани потрібних виходів та входів. Серед задіяних бібліотек є бібліотека LiquidCrystal що відповідає за підключення до плати Arduino рідкокристалічного екрану, та виведення на нього інформації.

Після цього йде одноразове оголошення змінних та функцій що знаходяться в строці void setup. Та перехід до виконання основного циклу роботи void loop в якому відбуваються основні операції по відслідковуванню натискання кнопок клавіатури, опитування підключених датчиків у встановлені інтервали часу, та виклик необхідних функцій стану меню та роботи основних режимів

Текст коду управляючої програми підлеглого модуля наведено нижче:

```
String receivedData = "";

#include <SoftwareSerial.h>

SoftwareSerial mySerial(11, 12);

int einc=0;

float zwei;

int drei=0;

char Inform[6]={'A','B','C','D','E'};

float door=0;

#include <OneWire.h>

OneWire ds(9);

unsigned long timer_t2=0;
```

```

unsigned long ti=0;

float temp=0;

#define d_warm 5

#define d_cool 4

int time_KB_press=0,time_KB_delay=30,

time_KB_delay1=600,time_Menu_press=0,time_Menu_deley2=100;

int time_curr=0, timer=0, timer_prev=0, timeMenu_p= 0, timeMenu_s = 80;

unsigned long timer_tempmenu=0;

unsigned long timer_door=0;

unsigned long g =0;

int b_work, C_Loow, b_warm, Start=1;

float tz=30, ton,toff, U, Um, K=10, del_t;

    int OUT, T=10000, d=0;

void setup() {

    Serial.begin(9600);

    mySerial.begin(9600);

    pinMode(2,INPUT);

    pinMode(6,OUTPUT);

    pinMode(7,OUTPUT);

    digitalWrite(6,HIGH);

    digitalWrite(7,HIGH);

    pinMode(4,OUTPUT);

    pinMode(5,OUTPUT);

```

```

digitalWrite(4,HIGH);

digitalWrite(5,HIGH);

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

digitalWrite(10,HIGH);      }

float var_1_1_reg [2]={40,4};

float var_1_1reg_max [2]={100,10};

float var_1_1reg_min [2]={26,0};

float var_1_1reg_step[2]={1,1};

String var_1_1_reg_neme[2]={"Temp:  ","Delta:  "};

float var_1_2reg [2]={40,30};

float var_1_2reg_max [2]={100,96};

float var_1_2reg_min [2]={26,25};

float var_1_2reg_step[2]={1,1};

String var_1_2reg_neme[2]={"Max:","Min:"};

float var_1_3reg [2]={35,3};

float var_1_3reg_max [2]={80,10};

float var_1_3reg_min [2]={25,0};

float var_1_3reg_step[2]={1,1};

String var_1_3reg_neme[2]={"Temp: ","G: "};

float var_1_4reg [2]={24,25};

float var_1_4reg_max [2]={40,45};

float var_1_4reg_min [2]={23,24};

```

```

float var_1_4reg_step[2]={0.1,1};

String var_1_4reg_neme[2]={"Tp:","T max:"};

String var_1_1reg_neme[2]={"Pidregim:"};

int var_1_regv [5]={1,1,1,4};

String var_2_2reg_neme[2]={"Pidregim:"};

int var_2_regv [5]={1,1,1,3};

float var_2reg [4]={33,10,10,100};

float var_2reg_max [4]={80,20,20,400};

float var_2reg_min [4]={25,1,1,10};

float var_2reg_step[4]={1,1,1,10};

String var_2reg_neme[4]={"Temp: ","K: ","T:","Um:"};

float var_2reg2 [4]={40,10,10,200};

float var_2reg2_max [4]={80,20,20,400};

float var_2reg2_min [4]={25,1,1,10};

float var_2reg2_step[4]={1,1,1,10};

String var_2reg2_neme[4]={"Temp:  ","K:  ","T:  ","Um:  "};

float var_3reg2 [3]={30,2,0.5};

float var_3reg2_max [3]={80,10,10};

float var_3reg2_min [3]={20,0,0};

float var_3reg2_step[3]={1,0.1,0.1};

String var_3reg2_neme[3]={"T:  ","Kp:  ","Ki:  "};

void Temp_1(){

unsigned long timer_t1=millis();

```

```

if(timer_t1>timer_t2+800){

    timer_t2=timer_t1;

    byte i, present = 0, type_s;

    byte data[9];

    byte addr[8];

    if ( !ds.search(addr)) {ds.reset_search();

    return; }

    ds.reset();

    ds.select(addr);

    ds.write(0x44, 1);

    present = ds.reset();

    ds.select(addr);

    ds.write(0xBE);

    for ( i = 0; i < 9; i++) { data[i] = ds.read(); }

    int16_t raw = (data[1] << 8) | data[0];

    if (type_s) {

        raw = raw << 3;

        if (data[7] == 0x10) { raw = (raw & 0xFFF0) + 12 - data[6]; } } else {

        byte cfg = (data[4] & 0x60);

        if (cfg == 0x00) raw = raw & ~7;

        else if (cfg == 0x20) raw = raw & ~3;

        else if (cfg == 0x40) raw = raw & ~1; }

    temp = (float)raw / 16.0; } }

```

```

void Temp_2(){
Temp_1();
if(temp>=var_1_2reg [0]){
digitalWrite(d_warm,HIGH);
digitalWrite(d_cool,LOW); }
if(temp<=var_1_2reg [1]){
digitalWrite(d_warm,LOW);
digitalWrite(d_cool,HIGH); } }

void Temp_3(){
Temp_1();
if(temp<var_1_1_reg [0]-(var_1_1_reg [1]/2)){
digitalWrite(d_warm,HIGH);
digitalWrite(d_cool,LOW); }
else if(temp>var_1_1_reg [0]+(var_1_1_reg [1]/2)){
digitalWrite(d_warm,LOW);
digitalWrite(d_cool,HIGH); } }

void Temp_4(){
Temp_1();
if(temp<var_1_3reg [0]-(var_1_3reg [1])){
digitalWrite(d_warm,LOW);
digitalWrite(d_cool,HIGH); }
else if(temp>var_1_3reg [0]){

```

```

digitalWrite(d_warm,HIGH);

digitalWrite(d_cool,LOW); } }

void Temp_5(){

Temp_1();

if(temp>=var_1_4reg [1]){

digitalWrite(6,LOW);

digitalWrite(7,LOW); }

if(temp<=var_1_4reg [0]){

digitalWrite(6,HIGH);

digitalWrite(7,HIGH); } }

void Temp_reg2(){

float tn;

Temp_1();

tn =temp;

del_t=var_2reg [0]-tn; //var_2reg [0]=Tz

U=var_2reg [1]*del_t; //var_2reg [1] =K

timer=millis();

if(Start>0){

Start=0;

ton=(U/var_2reg [3])*(var_2reg [2]*1000); //var_2reg [2]=T ; var_2reg [3]=

Um

if(ton<0.1*(var_2reg [2]*1000)){

ton=1000;}

if(ton>0.90*(var_2reg [2]*1000)){

```

```

ton=0.90*(var_2reg [2]*1000);}

toff=(var_2reg [2]*1000)-ton;

digitalWrite(d_warm,LOW);

digitalWrite(d_cool,HIGH);

b_warm=1;

timer_prev=millis(); }

else { if(b_warm>0){

if(timer-timer_prev>=ton){

    digitalWrite(d_warm,HIGH);

    digitalWrite(d_cool,LOW);

    b_warm=0;

    timer_prev=millis(); } }

else {

if(timer-timer_prev>=toff){

    digitalWrite(d_warm,LOW);

    digitalWrite(d_cool,HIGH);

    b_warm=1;

    ton=(U/var_2reg [3])*(var_2reg [2]*1000);

    if(ton<0.1*(var_2reg [2]*1000)){

        ton=1000;}

    if(ton>0.90*(var_2reg [2]*1000)){

        ton=0.90*(var_2reg [2]*1000);}

    toff=(var_2reg [2]*1000)-ton;

```

```

timer_prev=millis(); } } } }

void Temp_reg2_2(){

float tn;

Temp_1();

tn =temp;

del_t=var_2reg2 [0]-tn; //var_2reg [0]=Tz

U=var_2reg2 [1]*del_t; //var_2reg [1] =K

timer=millis();

if(Start>0){

Start=0;

ton=(U/var_2reg2 [3])*(var_2reg2 [2]*1000); //var_2reg [2]=T ; var_2reg
[3]= Um

if(ton>0.90*(var_2reg2 [2]*1000)){

ton=0.90*(var_2reg2 [2]*1000);}

if(ton<1000){ ton=0;

digitalWrite(d_warm,HIGH);

digitalWrite(d_cool,LOW);

b_warm=0; }

else{ digitalWrite(d_warm,LOW);

digitalWrite(d_cool,HIGH);

b_warm=1; }

toff=(var_2reg2 [2]*1000)-ton;

timer_prev=millis();}

else {if(b_warm>0){

```

```

if(timer-timer_prev>=ton){

digitalWrite(d_warm,HIGH);

digitalWrite(d_cool,LOW);

b_warm=0;

timer_prev=timer;}}

else {if(timer-timer_prev>=toff){ digitalWrite(d_warm,LOW);

digitalWrite(d_cool,HIGH);

b_warm=1;

ton=(U/var_2reg2 [3])*(var_2reg [2]*1000);

if(ton<0.1*(var_2reg2 [2]*1000)){

if(ton>0.90*(var_2reg2 [2]*1000)){

ton=0.90*(var_2reg2 [2]*1000);}

if(ton<1000){

ton=0;

digitalWrite(d_warm,HIGH);

digitalWrite(d_cool,LOW);

b_warm=0;}

else{ digitalWrite(d_warm,LOW);

digitalWrite(d_cool,HIGH);

b_warm=1; }

toff=(var_2reg2 [2]*1000)-ton;

timer_prev=timer; }}}}
```

```
float Upr=0;

float Uint=0;

float Uinti_1=0;

unsigned long currentTim =0;

unsigned long prevTim =0;

float output =0;

void Temp_reg2_3(){

float Kp=var_3reg2 [1];

float Ki = var_3reg2 [2];

float Tzad = var_3reg2 [0];

float Delta_h =0.0;

float integr;

float deltaTim=0.0;

Temp_1();

float Tpot=temp;

Delta_h = Tzad-Tpot;

currentTim=millis();

deltaTim=(currentTim - prevTim)/1000.0;

integr += Delta_h*deltaTim;

prevTim = currentTim;

output = Kp*Delta_h+Ki*integr;

output=constrain(output,0,255);

Serial.println(output);
```

```

if(output==0.0){digitalWrite(10,HIGH);}

analogWrite(10, output);

//var_3reg2 [3]={30,2,0.5};

// Delta_h = Tzad-Tpot;

// Upr=Ki*Delta_h;

// currentTim=millis();

// deltaTim=(currentTim - prevTim)/1000.0;

//Uinti_1=Uint+Ki*Delta_h*deltaTim;

// Uint=(Uinti_1-Uint)/deltaTim;

//prevTim = currentTim;

// output = Upr+Uint;

//output=constrain(output,0,255);

// analogWrite(10, output);}

void Seve_door(){

door = digitalRead(2);

Serial.println(door);

timer_door=millis(); }

int intValue = 0;

void parseData(String data) {

int address = 0;

char Inf = '0';

float value1=0, value2=0, value3=0, value4=0;

int firstComma = data.indexOf(',');

```

```

int secondComma = data.indexOf(',', firstComma + 1);

int thirdComma =data.indexOf(',', secondComma + 1);

address = data.substring(0, firstComma).toInt();

intValue = data.substring(firstComma + 1, secondComma).toInt();

Inf = data.charAt(secondComma + 1);

if(address==12){switch(intValue){

    case(1):{if(Inf=='C'){b_work=1;einc=1;}

if(Inf=='D'){b_work=0; einc=0;

digitalWrite(5,HIGH); //налаштування вимкнення 1 режиму

digitalWrite(4,HIGH);}

if(Inf=='E'){int forComma =data.indexOf(',', thirdComma + 1);

var_1_1_reg [0] = data.substring(thirdComma + 1,forComma).toFloat();

var_1_1_reg [1] = data.substring(forComma + 1).toFloat();

Serial.println(var_1_1_reg [0]);

Serial.println(var_1_1_reg [1]); }

break;}

case(2):{if(Inf=='C'){b_work=1;einc=2;}

if(Inf=='D'){b_work=0;einc=0;

digitalWrite(5,HIGH);

digitalWrite(4,HIGH);}

if(Inf=='E'){int forComma =data.indexOf(',', thirdComma + 1);

var_1_2reg [0] = data.substring(thirdComma + 1,forComma).toFloat();

var_1_2reg [1] = data.substring(forComma + 1).toFloat();

```

```

Serial.println(var_1_2reg [0]);

Serial.println(var_1_2reg [1]);}

break;}

case(3):{if(Inf=='C'){b_work=1;

einc=3;}

if(Inf=='D'){ b_work=0;

einc=0;

digitalWrite(5,HIGH); //налаштування вимкнення 1 режиму

digitalWrite(4,HIGH);}

if(Inf=='E'){ int forComma =data.indexOf(',', thirdComma + 1);

var_1_2reg [0] = data.substring(thirdComma + 1,forComma).toFloat();

var_1_2reg [1] = data.substring(forComma + 1).toFloat(); }

break;}

case(4):{ if(Inf=='C'){ b_work=1;

einc=4;}

if(Inf=='D'){ b_work=0; einc=0;

digitalWrite(6,HIGH);

digitalWrite(7,HIGH);}

if(Inf=='E'){int forComma =data.indexOf(',', thirdComma + 1);

var_1_4reg [0] = data.substring(thirdComma + 1,forComma).toFloat();

var_1_4reg [1] = data.substring(forComma + 1).toFloat();}

break;}

case(5):{if(Inf=='C'){C_Loow=1;einc=5;}

```

```

if(Inf=='D'){C_Loow=0;einc=0;

digitalWrite(5,HIGH);

digitalWrite(4,HIGH);}

if(Inf=='E'){

int forComma =data.indexOf(',', thirdComma + 1);

int faivComma =data.indexOf(',', forComma + 1);

int sixthComma =data.indexOf(',', faivComma + 1);

var_2reg [0] = data.substring(thirdComma + 1,forComma).toFloat();

var_2reg [1] = data.substring(forComma + 1,faivComma).toFloat();

var_2reg [2] = data.substring(faivComma + 1,sixthComma).toFloat();

var_2reg [3] = data.substring(sixthComma + 1).toFloat();

Serial.println(var_2reg [0]);

Serial.println(var_2reg [1]);

Serial.println(var_2reg [2]);

Serial.println(var_2reg [3]); }

break; }

case(6):{if(Inf=='C'){C_Loow=1;

einc=6;}

if(Inf=='D'){ C_Loow=0; einc=0;

digitalWrite(5,HIGH);

digitalWrite(4,HIGH);}

if(Inf=='E'){

int forComma =data.indexOf(',', thirdComma + 1);

```

```

int faivComma =data.indexOf(',', forComma + 1);

int sixthComma =data.indexOf(',', faivComma + 1);

var_2reg2 [0] = data.substring(thirdComma + 1,forComma).toFloat();
var_2reg2 [1] = data.substring(forComma + 1,faivComma).toFloat();
var_2reg2 [2] = data.substring(faivComma + 1,sixthComma).toFloat();
var_2reg2 [3] = data.substring(sixthComma + 1).toFloat();

Serial.println(var_2reg2 [0]);

Serial.println(var_2reg2 [1]);

Serial.println(var_2reg2 [2]);

Serial.println(var_2reg2 [3]); }

break;}

case(7){ if(Inf=='C'){ if(C_Loow==0 && einc==0){

digitalWrite(10,LOW);

Serial.println("feouev");}

C_Loow=1;einc=7;}

if(Inf=='D'){b_work=0;einc=0;

digitalWrite(10,HIGH); }

if(Inf=='E'){int forComma =data.indexOf(',', thirdComma + 1);

int faivComma =data.indexOf(',', forComma + 1);

int sixthComma =data.indexOf(',', faivComma + 1);

var_3reg2 [0] = data.substring(thirdComma + 1,forComma).toFloat();

var_3reg2 [1] = data.substring(forComma + 1,faivComma).toFloat();

var_3reg2 [2] = data.substring(faivComma + 1,sixthComma).toFloat();

```

```

zwei = data.substring(sixthComma + 1).toFloat(); }

break; }

case(8):{ if(Inf=='A' && millis()-ti>261){ ti=millis();

    Temp_1();

    int adres1=10, intValue=20;

    char Inf ='B';

    mySerial.print(adres1);mySerial.print(",");

    mySerial.print(intValue);mySerial.print(",");

    mySerial.print(Inf);mySerial.print(",");

    mySerial.print(temp);mySerial.print(";");}

    break;}

case(9):{ if(Inf=='A'){ Seve_door();

    int adres3=10;

    int intValue3=21;

    char Inf3 ='F';

    mySerial.print(adres3);mySerial.print(",");

    mySerial.print(intValue3);mySerial.print(",");

    mySerial.print(Inf3);mySerial.print(",");

    mySerial.print(door);mySerial.print(";");}

    break;}}}}

void loop() {

    while (mySerial.available() > 0) {

        char receivedChar = mySerial.read();

```

```
if (receivedChar == ';') {parseData(receivedData);  
receivedData = ""; }  
  
else {receivedData += receivedChar; }}  
  
if(b_work>0){switch(einc){case(1){Temp_3();  
break;}  
  
case(2){Temp_4(); break;}  
  
case(3){Temp_2(); break;}  
  
case(4){Temp_5(); break;}}}   
  
if(C_Loow>0){switch(einc){  
  
case(5){Temp_reg2();  
  
break;}  
  
case(6){Temp_reg2_2();  
  
break;}  
  
case(7){Temp_reg2_3();  
  
break;}}}}}
```

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання магістерської дипломної роботи розроблено лабораторний стенд дослідження законів керування на основі автоматизованої системи керування температурою у термошафі, що реалізує інформаційний обмін між модулями ТМС на основі платформи Arduino. Даний лабораторний стенд реалізує інформаційний обмін між декількома модулями телемеханічної системи з використанням інформаційних посилок, що забезпечують адресацію модулів, та режимів роботи для передачі необхідної інформації в обох напрямках. Макет складається з двох плат Arduino UNO, що імітують роботу керуючої ЕОМ та функціонального модуля. Керуючий модуль виконує постійне опитування підлеглого модуля про отримані значення з датчиків, та реалізує видачу керуючих сигналів на увімкнення того чи іншого режиму роботи установки. В якості інтерфейсу для обміну інформацією використано UART.

В роботі представлені розроблені структурна та принципова-електрична схеми, а також алгоритмічне забезпечення модулів на основі якого розроблено програмне забезпечення.

Дана робота виконана в рамках загального напрямку створення ТМС, в яких суміщено переваги класичних ТМС та інших архітектурних підходів, для керування технологічними процесами.

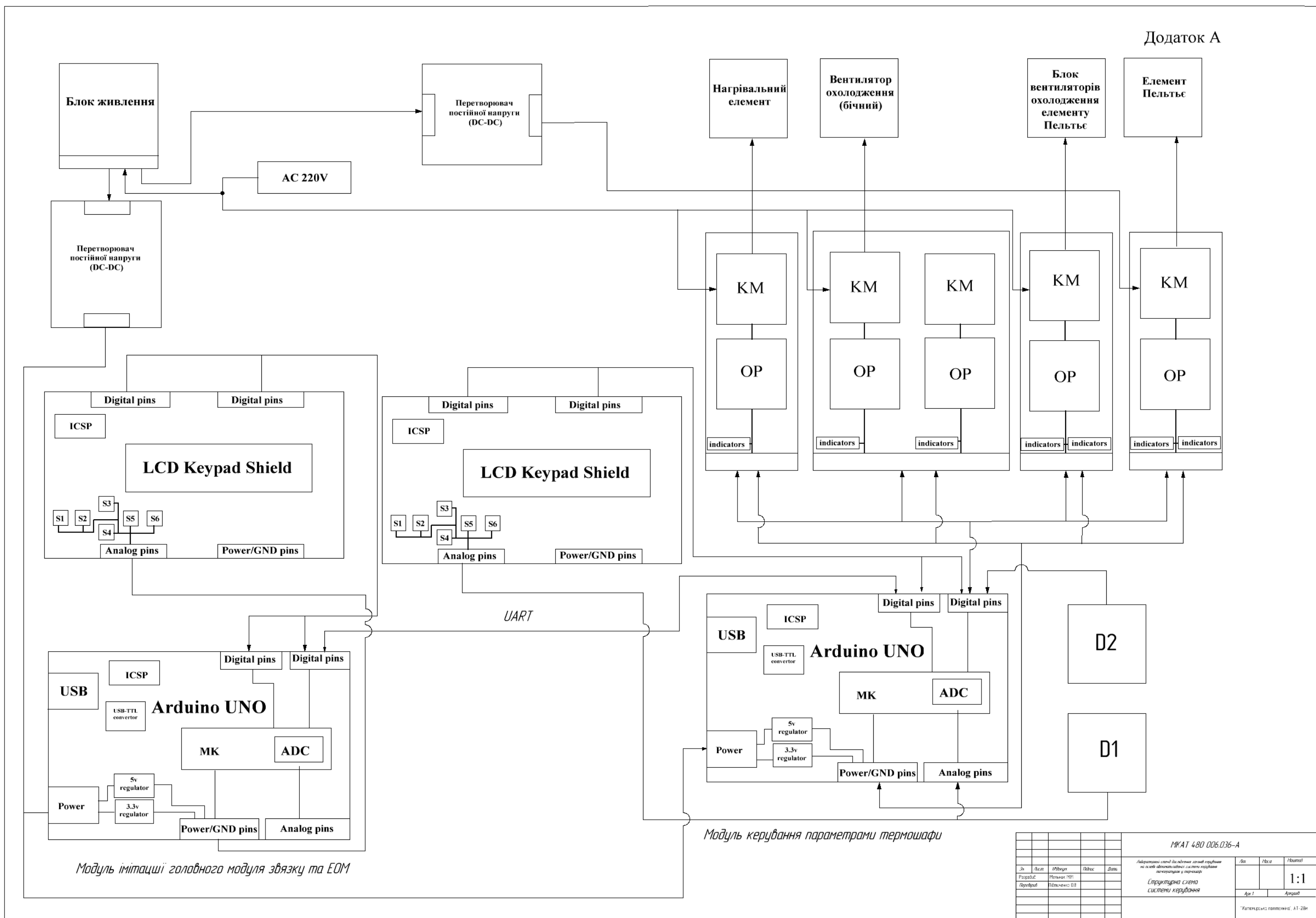
Основою цього методу є використання таких ТМС в яких функції керування об'єктами з верхнього рівня перенесені до місцевих контролерів, що отримують посилки з певними значеннями та отримують інформацію про об'єкти лише з метою надання оператору загальної інформації. У цьому випадку місцеві контролери, які працюють як функціональні модулі вузлів ТМС, виконують функції регулювання.

В результаті виконання роботи у вигляді лабораторного стенду отримано макет, що реалізує закони керування температурою у термошафі для дослідницьких та навчальних цілей.

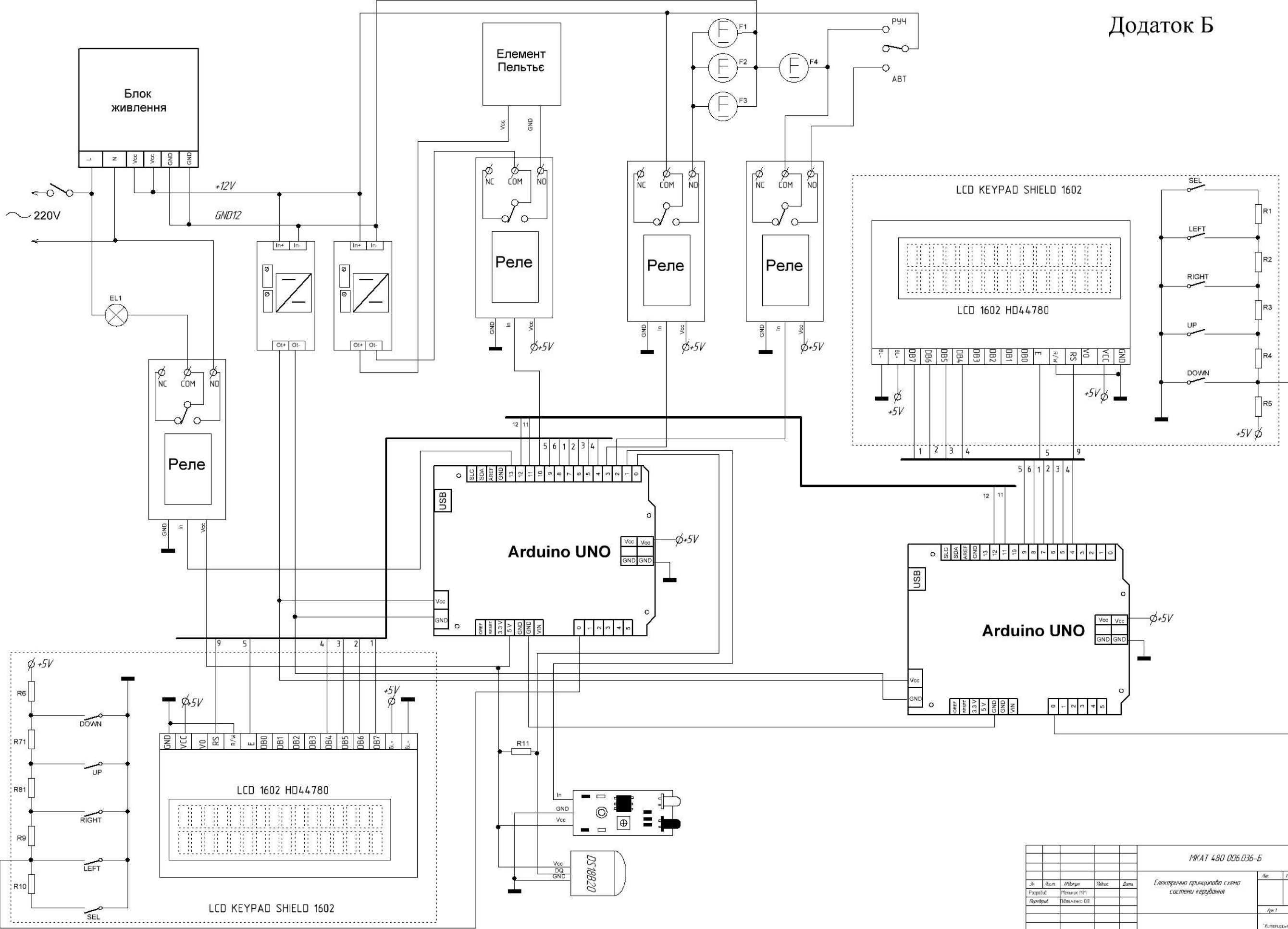
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ПІД закон регулювання: [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki> (дата звернення 16.12.2024).
2. В.А. ВТЮРИН ОСНОВИ АСУТП [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/3908750/> (дата звернення 16.12.2024).
3. UART: [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/UART> (дата звернення 16.16.2024).
4. Arduino Uno: [Електронний ресурс] – Режим доступу: <https://doc.arduino.ua/ru/prog/> (дата звернення 11.12.2024).
5. Блок живлення: [Електронний ресурс] – Режим доступу: https://uamper.com/%D0%91%D0%BB%D0%BE%D0%BA-%D0%BF%D0%B8%D1%82%D0%B0%D0%BD%D0%B8%D1%8F-LED-%D0%B4%D0%BB%D1%8F-%D1%81%D0%B2%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D0%BE%D0%B4%D0%BE%D0%B2-120%D0%92%D1%82-120W?srsId=AfmBOopt-kg92-vGSNohKfjWw1yQFsRuis-icfjlpn_RaFWglMTff3m (дата звернення 17.12.2024).
6. LCD 1602 keypad shield: [Електронний ресурс] – Режим доступу: <https://uamper.com/%D0%A1%D0%B8%D0%BC%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5-%D1%8D%D0%BA%D1%80%D0%B0%D0%BD%D0%B8/LCD-1602-keypad-shield> (дата звернення 11.12.2024).
7. Температурний датчик водонепроникний DS18B20 : [Електронний ресурс] – Режим доступу: <https://arduino.ua/ru/prod414-temperatyrnii-datchik-vodonepronicemii-ds18b20> (дата звернення 12.12.2024).
8. 2-канальний модуль реле 5В: [Електронний ресурс] – Режим доступу: <https://uamper.com/2-%D0%BA%D0%B0%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9-%D1%80%D0%B5%D0%BB%D0%B5> (дата звернення 13.12.2024).
9. XL4015 5А Понижуючий перетворювач з регулюванням напруги, струму: [Електронний ресурс] – Режим доступу: https://radiocron.com.ua/ua/p1061019130-xl4015-ponizhayuschij-preobrazovatel.html?source=merchant_center&gclid=CjwKCAjwyqWkBhBMEiwAp2yUFsxjekAjVSI-48QlJna6TFmgdollarYbUYVloKVnFMnwoTL3msm3dUPBoChOcQAvD_BwE. (дата звернення 15.12.2024).

ДОДАТКИ

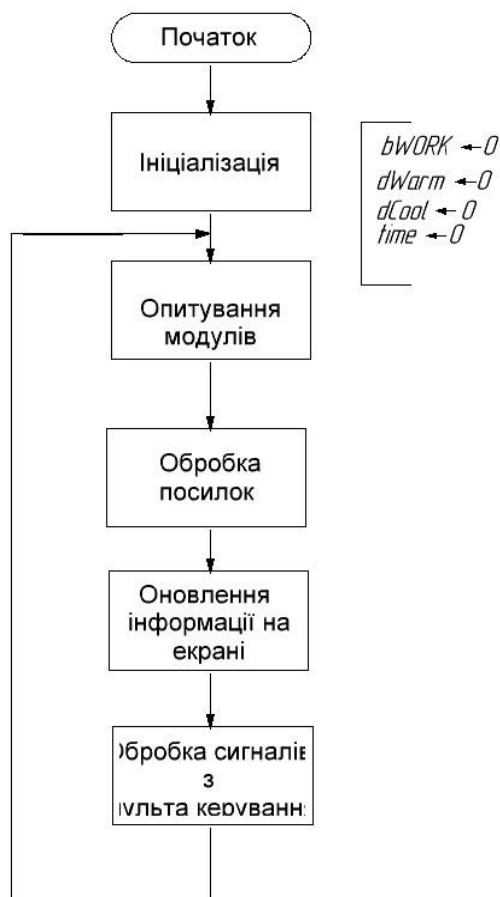


Додаток Б

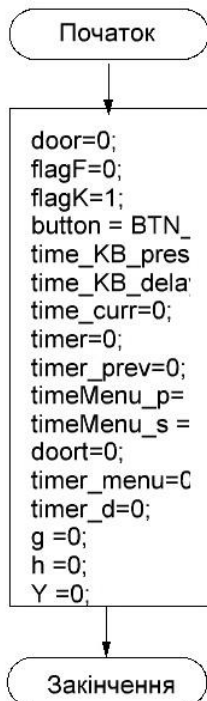


					МКАТ 480 006.036-5		
Зн	Авт	Модифі	Відк	Вит	Електрична принципова схема системи керування		
Розробл	Митченко (М)						
Перевір	Митченко (М)				1:1		
					Арх 1 Арх 2		
					Хитрийська патентна, АТ-284		

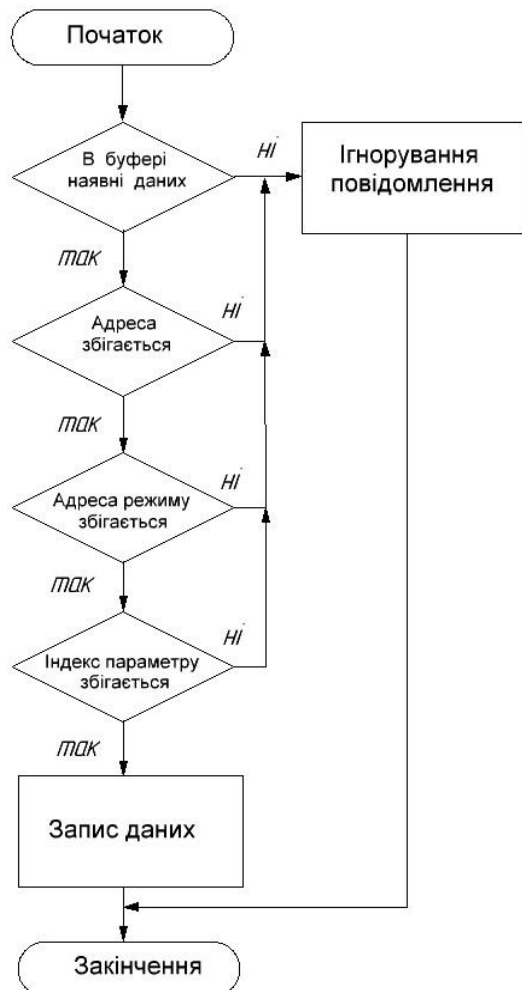
Алгоритм основного цикла работы контролера



Алгоритм ініціалізації основних змінних контролера



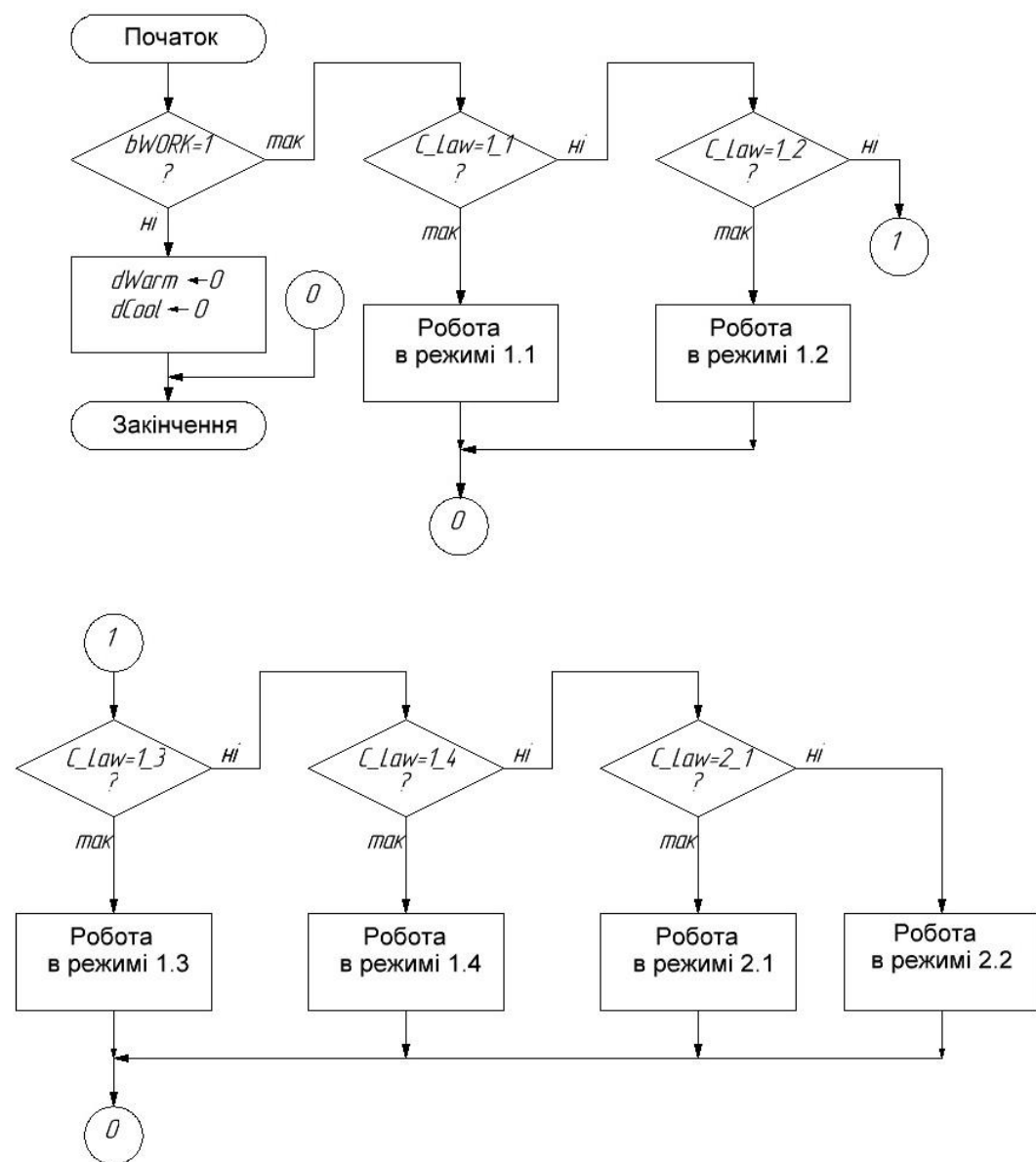
Алгоритм роботи з буфером



Алгоритм опитування модулів

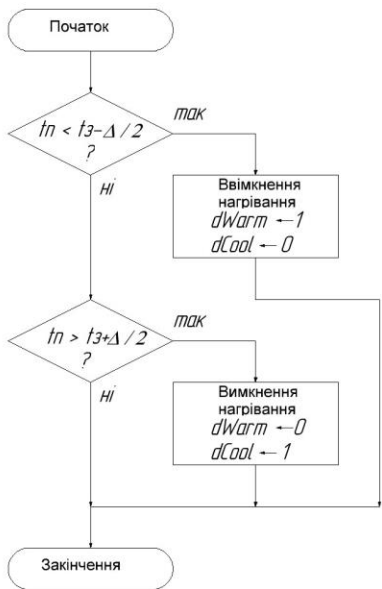


Алгоритм обрання підрежиму роботи контролера

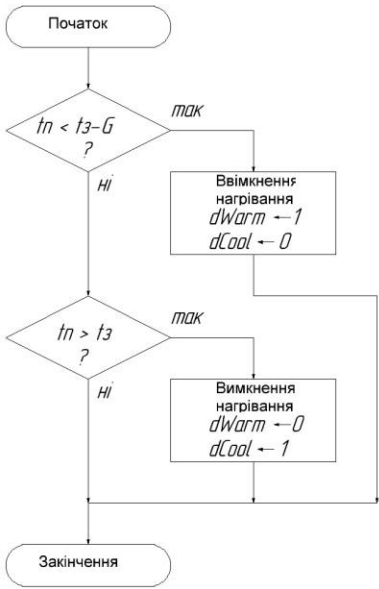


					МКАТ 480 006.036-В		
Зм.	Вс.	Місце	Відео	Дат.	Алгоритми основного циклу роботи, ініціалізації змінних, роботи з буфером, опитування модулів та обрання підрежиму роботи		
Розроб.	Місце	Відео	Дат.	Дат.			
Вір.	Місце	Відео	Дат.	Дат.	1:1		
					Характеристика параметрів, АТ-28		

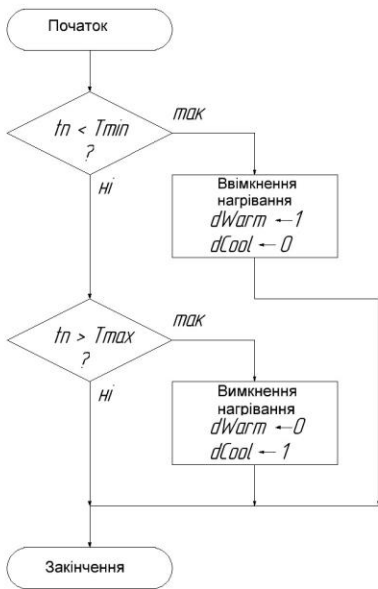
Алгоритми роботи в підрежимі 1.1



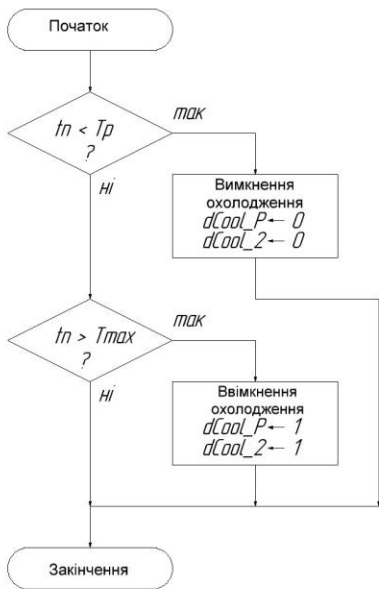
Алгоритми роботи в підрежимі 1.2



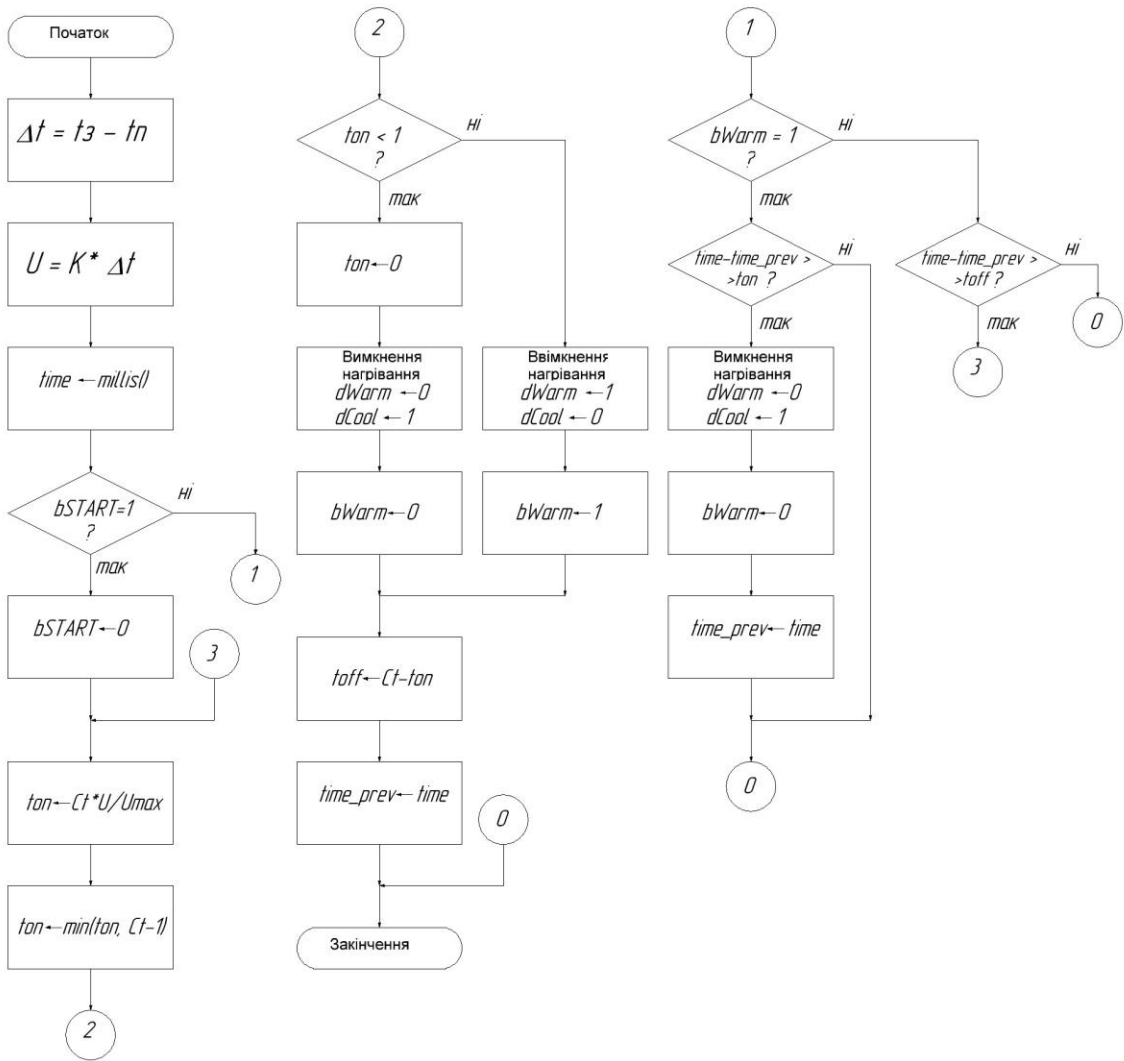
Алгоритми роботи в підрежимі 1.3



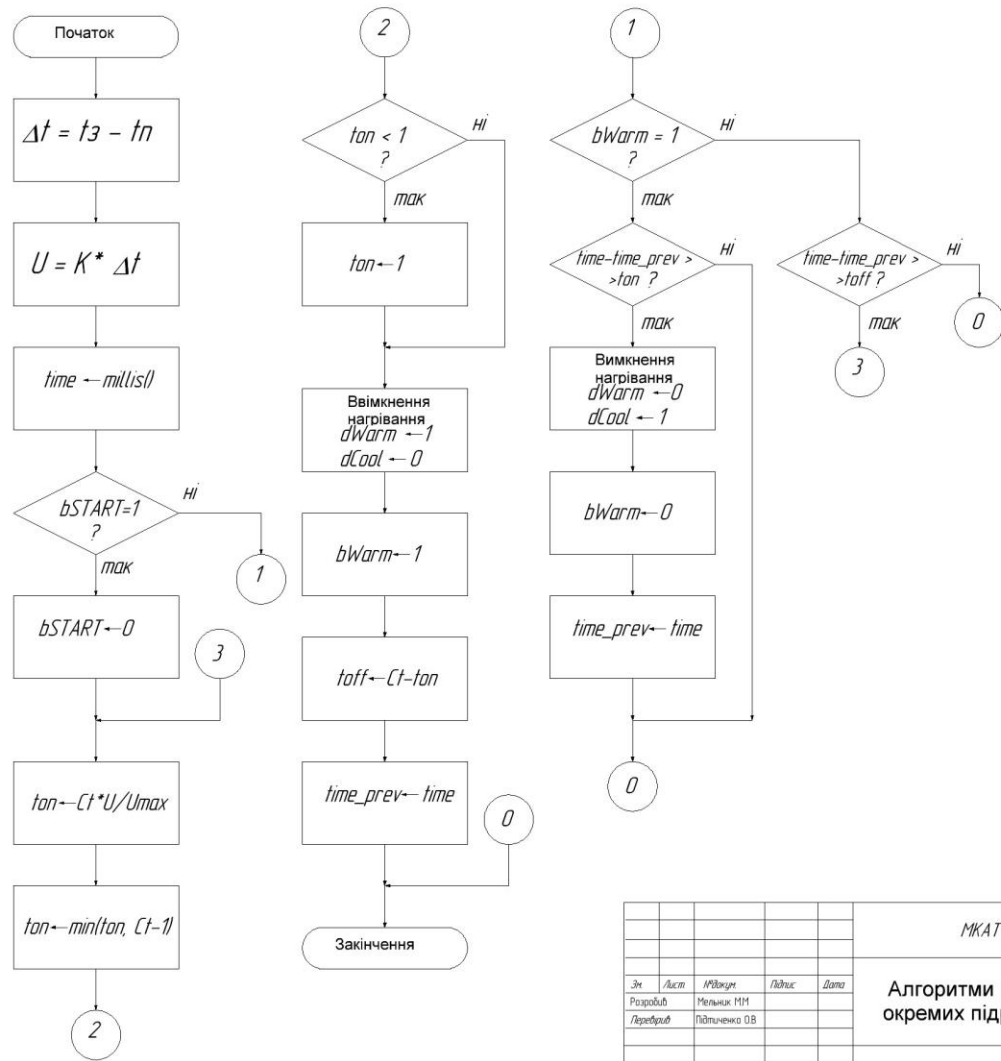
Алгоритми роботи в підрежимі 1.4

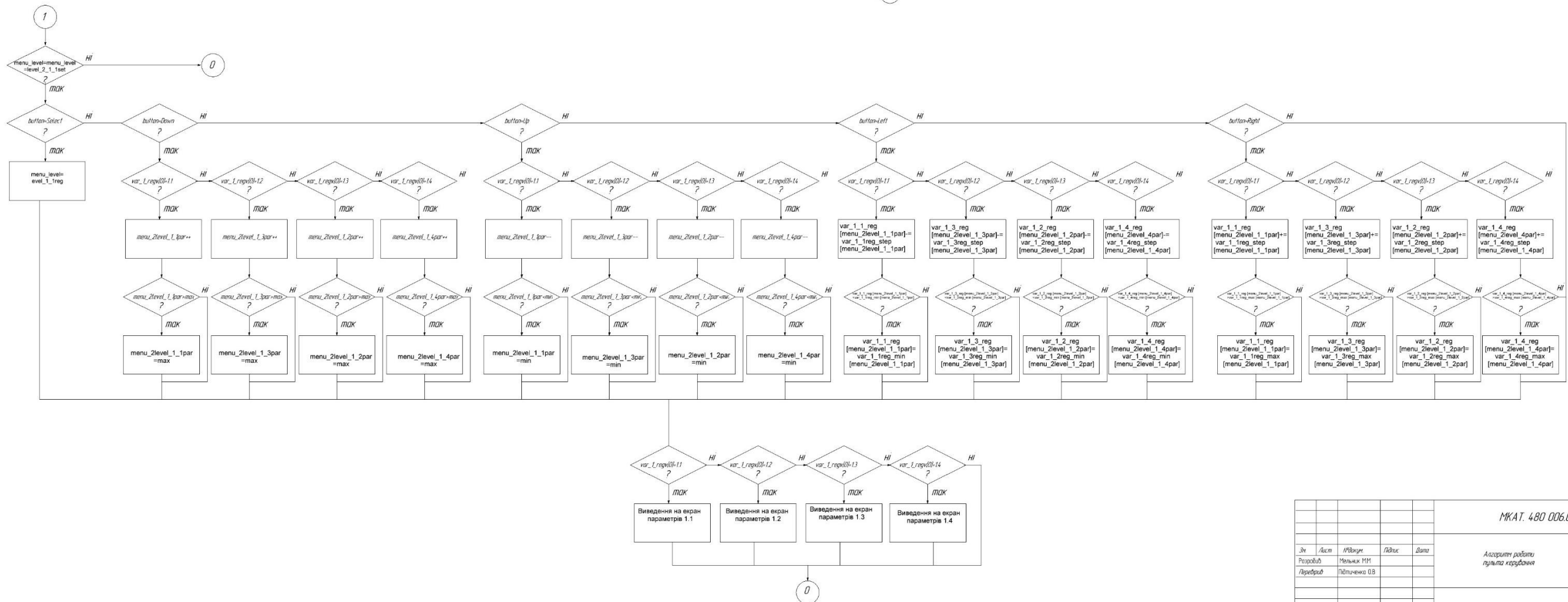
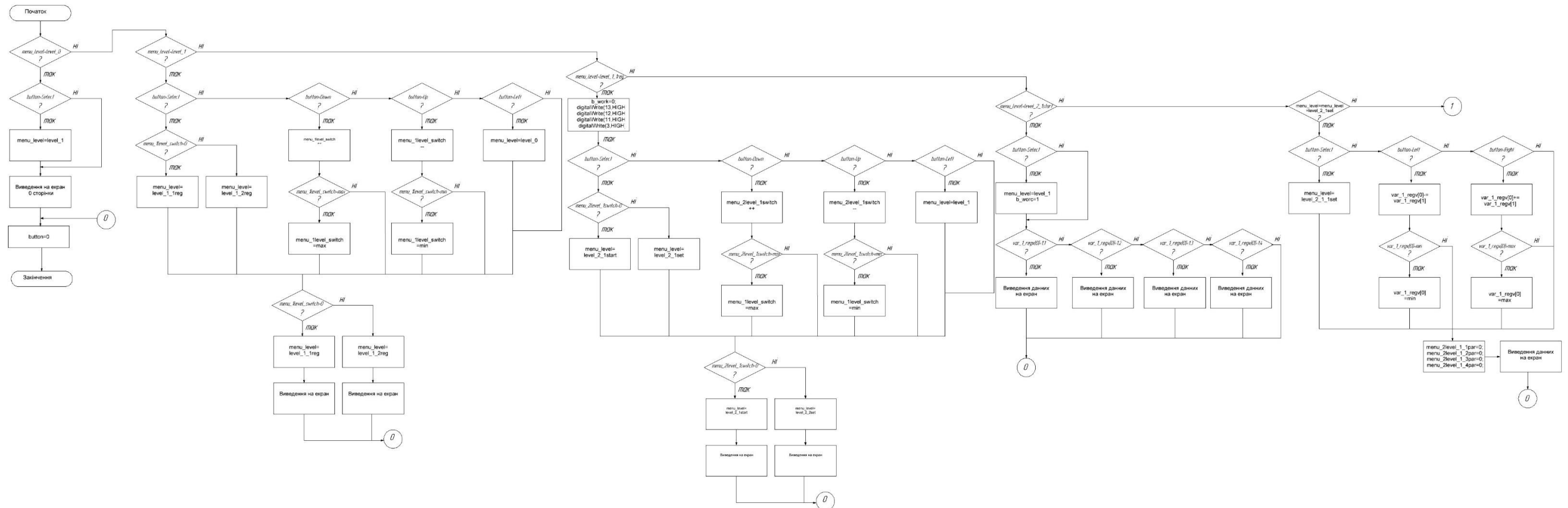


Алгоритми роботи в підрежимі 2.1



Алгоритми роботи в підрежимі 2.2





						МКАТ. 480 006.036 Д			
						Алгоритм роботи пульту керування	Лист	Маса	Назва
Зн	Віст	М'явонг	Підпис	Датум					
Розробів		Мельник ММ							
Перевірив		Поліщук ОВ							
							Арх: і	Архів	
						"Інженерська політехніка". АТ-28а			

Формам.	Зона	Поз.	Позначення	Найменування	Кіл.	Прим.
			DS18B20	Датчик температури	1	
			YL-63	Інфрачервоний датчик	1	
			S-120-12	Блок живлення	1	
			LCD Keypad Shield	Модуль дисплея з кнопками	1	
			Arduino Uno	Контролер	2	
			XL4015CC CV	Понижувальний DC-DC конвертор	1	
			XL4015 5A	Понижувальний перетворювач з регулятором	1	
			M	Вентилятори	4	
			Реле	Реле	4	
			Елемент Пельтьє	Елемент Пельтьє	1	
			EL1	Лампа розжарювання	1	
						</

Загальний вигляд лабораторної установки

