

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютерна система моніторингу та регулювання
температури теплоносіїв у газових котлах*

Виконав: студент 4 курсу, групи СІ-41
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Жук І.Д. І.

(прізвище та ініціали)

Керівник

(підпис)

Яцишин В. В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Луцин Н. С.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Мудрик І. Я.

(прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
«27» січня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

Студента Жука Івана-Дмитра Івановича
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система моніторингу та регулювання температури теплоносіїв у газових котлах

Керівник роботи Яцишин Василь Володимирович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » січня 2025 року № 4/7-53

2. Термін подання студентом завершеної роботи 19.06.2025

3. Вихідні дані до роботи Технічне завдання, перелік елементів, код програмного забезпечення

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз вимог до комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газовому котлі

2. Проектування системи моніторингу та регулювання температури теплоносіїв у газовому котлі

3. Розробка алгоритму та програмного забезпечення системи моніторингу та регулювання температури теплоносіїв у газовому котлі

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Структурна схема

2. Схема електрична принципова

3. Блок-схема алгоритму

4. Схема електрична монтажна (з'єднань).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	Завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М. І. д.т.н., проф. каф. МТ.</i>		

7. Дата видачі завдання 27.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	<i>27.01 – 02.02</i>	<i>Викон.</i>
2.	<i>Аналіз вимог до комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах</i>	<i>03.02 – 28.02</i>	<i>Викон.</i>
3.	<i>Проектування системи моніторингу та регулювання температури теплоносіїв у газових котлах</i>	<i>01.03 – 30.03</i>	<i>Викон.</i>
4.	<i>Розробка алгоритму та програмного забезпечення системи моніторингу та регулювання температури теплоносіїв у газових котлах</i>	<i>01.04 – 30.04</i>	<i>Викон.</i>
5.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>01.05 -15.05</i>	<i>Викон.</i>
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>16.05 – 30.05</i>	<i>Викон.</i>
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>9.06 – 15.06</i>	
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06 – 22.06</i>	<i>Викон.</i>
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>26.06</i>	<i>Викон.</i>

Студент

(підпис)

Жук І-Д. І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Яцишин В. В.

(прізвище та ініціали)

АНОТАЦІЯ

Жук І-Д. І. Комп'ютерна система моніторингу та регулювання температури теплоносіїв у газових котлах: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: регулювання температури, моніторинг, збільшення, сервопривід, реле, мікроконтролер, витік газу.

Кваліфікаційна робота присвячена розробці та реалізації системи автоматичного контролю температури та контролю витоків газів в котлі газового опалення. Для досягнення цієї мети, використовується мікроконтролер STM32, сенсори та сервопривід. Завдяки цій системі в приміщенні, яке опалюється котлом газового опалення, буде підтримуватися комфортна температура для проживання людей і забезпечуватися безпека, щодо аварійних ситуацій. Дана система регулює температуру до комфортної для людини. В випадках витоку газу чи перевищення критичної температури система має захисні сценарії, завдяки яким ризик смертельних ситуацій стає приблизно рівний нулю через професійно сплановані алгоритми роботи схеми, які були створенні з урахуванням правил техніки безпеки. Уся інформація нашого газового котла відображається на LCD дисплеї, також передаються всі дані нашої системи на головний комп'ютер через засоби UART, де можна побачити всю хронологію подій. Система автоматично регулює температуру шляхом зміни подачі об'єму газу в котел газового опалення. Розроблена система здатна забезпечувати безпеку від витоків газу, а також підтримання комфортної температури приміщення через котел газового опалення.

ANNOTATION

Zhuk I-D.I. Computerized system for monitoring and regulating the temperature of heat carriers in gas boilers: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: temperature control, monitoring, increase, servo drive, relay, microcontroller, gas leak.

The qualification work is devoted to the development and implementation of a system for automatic temperature control and gas leak control in a gas heating boiler. To achieve this goal, an STM32 microcontroller, sensors and a servo drive are used. Thanks to this system, a comfortable temperature for people to live in will be maintained in a room heated by a gas heating boiler and safety will be ensured in case of emergency situations. This system regulates the temperature to a comfortable temperature for a person. In cases of gas leakage or exceeding the critical temperature, the system has protective scenarios, thanks to which the risk of fatal situations becomes approximately zero due to professionally planned algorithms of the circuit operation, which were created taking into account safety regulations. All information from our gas boiler is displayed on the LCD display, all data from our system is also transmitted to the main computer via UART, where you can see the entire chronology of events. The system automatically regulates the temperature by changing the gas supply volume to the gas heating boiler. The developed system is able to ensure safety from gas leaks, as well as maintaining a comfortable room temperature through the gas heating boiler.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ ТА РЕГУЛЮВАННЯ ТЕМПЕРАТУРИ ТЕПЛОНОСІЇВ У ГАЗОВИХ КОТЛАХ.....	12
1.1 Аналіз вимог до комп'ютерної системи	12
1.2 Аналіз можливих рішень поставленого завдання	14
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА РЕГУЛЮВАННЯ ТЕМПЕРАТУРИ ТЕПЛОНОСІЇВ У ГАЗОВИХ КОТЛАХ .	18
2.1 Розробка узагальненої структури комп'ютерної системи.....	18
2.2 Обґрунтування вибору апаратного забезпечення проєктованого комп'ютерного засобу	18
2.2.1 Мікроконтролер	18
2.2.2 Сенсори для вимірювання температури.....	21
2.2.3 Дисплей.....	23
2.2.4 Реле.....	24
2.2.5 Послідовний інтерфейс I2C	25
2.2.6 Сервопривід.....	26
2.2.7 Датчик газу	27
2.2.8 UART модулі передачі даних	28
2.3 Проєктування електричної принципової схеми	29
2.3.1 Середовище проєктування електричних схем	29

					КС КРБ 123.205.00.00 ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Жук І-Д.І.			Комп'ютерна система моніторингу та регулювання температури теплоносіїв у газових котлах			Літ.		Арк.	Аркушів	
Перевір.		Яцишин В.В.								6		
Реценз.		Мудрик І. Я.						ТНТУ, каф. КС, гр. СІ-41				
Н. Контр.		Луцик Н.С.										
Затверд.		Осухівська Г.М.										

2.4	Проектування комп'ютерного засобу	29
2.4.1	Вибір алгоритму «регуляції»	29
2.4.2	Розробка схеми підключення сенсорів до мікроконтролера.....	30
2.4.3	Опис електричної принципової схеми.....	32
РОЗДІЛ 3 РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ ТА РЕГУЛЮВАННЯ ТЕМПЕРАТУРИ ТЕПЛОНОСІЇВ У ГАЗОВИХ КОТЛАХ.....		33
3.1	Реалізація і моделювання проектних рішень	33
3.1.1	Налаштування середовища розробки програмного забезпечення.....	35
3.1.1.1	Налаштування тактування мікроконтролера	35
3.1.1.2	Налаштування комунікаційних інтерфейсів	37
3.1.1.3	Налаштування портів GPIO	39
3.1.1.4	Налаштування таймерів.....	41
3.1.1.4.1	TIM2.....	41
3.1.2	Опис програми	43
3.1.2.1	Функції	46
3.1.2.2.1	Основні функції.....	47
3.1.3	Інтеграція механічних елементів з мікроконтролером	50
3.1.3.1	Живлення насоса для циркуляції теплоносія.....	50
3.1.3.2	Керуючий сигнал сервопривода.....	51
3.2	Тестування	51
3.2.1	Інтерфейс	52
3.2.2	Тестування режимів роботи.....	53
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ..		55
4.1	Моделювання та прогнозування небезпечних ситуацій.....	55

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

4.2	Особливості заходів електробезпеки на підприємствах	57
4.3	Організація безпечної роботи електроустановок	59
	ВИСНОВКИ.....	61
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
	Додаток А Технічне завдання	
	Додаток Б Перелік елементів	
	Додаток Б Код програмного забезпечення	

					КС КРБ 123.205.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

GPIO – General-Purpose Input/Output

I²C – Inter-Integrated Circuit

LCD – High Definition Multimedia Interface

PWM – Network Attached Storage

RTC – System-on-a-chip

RCC – Universal Serial Bus

UART – Universal Asynchronous Receiver-Transmitter

МК – Мікроконтролер

ШИМ – Широтно-Імпульсна Модуляція

					КС КРБ 123.205.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Через необхідність зимового опалення, яке зумовлене фактором неблагополучних природних умов для проживання людства в простих, а також суворих кліматичних зонах в холодних умовах зими, крайнє необхідним є підтримання теплих кімнатних температур. За це відповідають системи опалення. Однією із них являється проста котлова система газового опалення. Вона на території нашої країни і на територіях сусідніх країн Європи є саме найбільш поширеною і найбільш практичною, як в умовах прожиття в одиночних жилих будинках, так і в функціонуванні великих будівель. Газові котли є центральним і найголовнішим елементом систем даного опалювання. Головною функцією газових котлів являється підтримання зручних температур для прийняттого існування живих організмів в стані доброго фізичного здоров'я. Комфортні температури для існування різних тварин і людей різні. Для зоомагазинів в вбудованих тераріумах температура мусить становити від +25° до +28 °С для холонокровних тварин, активний спосіб життя яких вимагає високих температур. Для підтримки існування худоби в сільськогосподарських фермах і м'ясокомбінатах являється від +4 °С до +20 °С, хоча й оптимальною температурою є +15 °С. Крім того вологість повітря мусить не перевищувати 60% (при занадто низьких температурах та й високій вологості призводяться до переохолодження тіла худоби).

Є доволі багато законних проєктів, які в свою чергу стосуються до правил гуманного поводження і умов існування тварин в умовах проживання, як в жилих будинках, так і в промислових районах. До даних проєктів належать: закон України «Про захист тварин від жорстокого поводження», в своєму числі статтею під номером 9 цього Закону в якому прямо йдеться, що, власники домашніх тварин мусять підтримувати умови необхідні, що є рівними для умов біологічним, видовим та індивідуальним особливостям.

Враховуючи ці законні проєкти, а також ті які являються подібними до них за змістом необхідно підтримувати комфортну температурну умову

					<i>КС КРБ 123.205.00.00 ПЗ</i>	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

всередині приміщень в яких вони перебувають. Крім того, на мою думку, більш важливим це питання полягає щодо людей та умов їхнього проживання, навчання й місць робочої діяльності. Оскільки питання існування й якості життя людини собою являються врази більш серйозними. Крім питання опалення й підтримання вигідної температури в будинках проживання людей є важливим й питання правильності роботи й забезпечення безпеки в аварійних і небезпечних ситуаціях роботи газового котла опалення. Нажаль існує доволі велика кількість летальних випадків в подібних системах опалення. Вони призводять до зруйнування, вибухів й пожеж будівель, де вони встановленні, а також до численних жертв в таких ситуаціях. Одним з факторів цих небезпек є сам газ в даних котлах опалювання.

Метою кваліфікаційної роботи є розробка та реалізація системи моніторингу та регулювання температури теплоносіїв у газових котлах. Для досягнення цієї мети використовується мікроконтролер (МК) STM32, сенсори та сервоприводи. Дана система дозволить ефективно регулювати температуру приміщення, та забезпечувати безпечні умови функціонування газового котла, без аварійних й небезпечних випадків.

Цей проєкт забезпечить користувачу даної системи опалення умови безпечного й комфортного існування без переймань щодо побутових питань про керуванням температури теплоносіїв у газових котлах, а також забезпечить правильне функціонування з найменшим ризиком поганих ситуацій.

Для реалізації мети проєкту слід реалізувати наступні задачі:

- виконати огляд і аналіз роботи котлів газового опалення й інших схожих систем;
- розробити структурну схему системи моніторингу та регулювання температури теплоносіїв у газових котлах;
- розробити електричну принципову схему проєктованої системи;
- розробити алгоритм щодо забезпечення запропонованої системи;
- створити програму для коректної роботи усіх складових системи.

					<i>КС КРБ 123.205.00.00 ПЗ</i>	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ ТА РЕГУЛЮВАННЯ ТЕМПЕРАТУРИ ТЕПЛОНОСІЇВ У ГАЗОВИХ КОТЛАХ

1.1 Аналіз вимог до комп'ютерної системи

Аби вдало забезпечити потрібний стан температури в житлових приміщеннях використовують багато різновидів зимового, в деяких з них є схожий принцип поширення тепла по площі будівлі. Який складається з батарей з'єднаними між собою сталевими трубами по яких часто тече сильно розігріта вода, температура котрої здебільшого регулюється окремим механізмом в залежності від виду опалення й самого котлу, яким воно здійснюється. Мінімальною необхідною температурою для комфортного перебування в середині житлового будинку становить $+18^{\circ}\text{C}$ або ж $+20^{\circ}\text{C}$ [2].

При роботі, в нашому випадку, з газовим котлом необхідно слідкувати за витоками суміші газів пропану та бутану, а при роботі з сильно нагрітою водою в всій системі опалення варто вважати аби її температура не стала набо високою і не причинила тиск в цілій системі й не стала головною причиною вибуху нашого житлового будинку чи будь-якого іншого приміщення. Найбільш головним фактором в цьому проєкті є нормальна температура у механізмі зберігання води, що сприятиме якості та адекватності її часу праці, а ще забезпечить комфортну температуру приміщення.

В більшій частині систем газового опалення відсутні прилади, які забезпечать нам необхідний рівень безпеки й ефективності їхньої роботи, аби це виправити буде достатньо одного мікроконтролера і його програмної частини.

					КС КРБ 123.205.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	<i>Аналіз вимог до комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах</i>		
Розроб.		Жук І-Д.І.					
Перевір.		Яцишин В.В.					
Реценз.		Мудрик І. Я.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						12	
					ТНТУ, каф. КС, гр. СІ-41		

Основні фрагменти програмної частини контролера:

- його робота за чіткими алгоритмами;
- контроль теплообміну приміщення;
- автономне вмикання / вимикання системи та контроль стабільної роботи встані «безпечного / аварійного» алгоритму, завдяки дуже стабільному процесу виправлення різних станів ситуацій;
- великий діапазон режимів роботи залежно від температури повітря в приміщенні відносно температури води;
- ведення графіку історії роботи пристрою;
- відображення станів системи з мінімальним інтервалом;
- постійний моніторинг рівня газу в повітрі;
- вимкнення системи при виявленні критичного рівня витоку газу;
- відключення роботи системи при ввімкненні аварійного режиму.

Дані фрагменти були сформовані відповідно джерела [3].

Сучасні ревізії газових котлів оснащені програмними й приладовими удосконаленнями. Більшість з них оснащені навіть цифровими дисплеями. Наприклад, котел ROCTERM Diamond TD2 з дисплеєм, як на рисунку 1.1.



Рисунок 1.1 – Газовий котел ROCTERM Diamond TD2

Система роботи теплоносіїв газового котла виконується в декілька етапів:

- спеціально розміщені датчики збирають всі потрібні дані

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

температури внутрішньої води;

- температурні дані обчислюються автоматично пристроєм;
- в випадку різного результату автоматично виконується відповідний

- сценарій дій відносно даного випадку.

Головною метою автоматизації газового котла опалення є максимальна автоматизація рутинних процесів опалення.

1.2 Другий підрозділ

Більшість житлових, відпочинкових і промислових будівель опалюються за допомогою газового котла. Одним з найбільш поширених аспектів його великого діапазону використання є простота в встановленні і сильною ефективністю в нагріванні приміщення. Воно є просте в обслуговуванні, а також має максимально простий механізм у застосуванні. Його принцип роботи є широковідомим, також він є дуже простим в розумінні. Основа його роботи являється спалювання суміші газів в середині топки з великим виділенням його високотемпературної теплової енергії. Завдяки чому котлова деяка рідина (незамерзаюча рідка речовина або вода) нагрівається й циркулюється. Цей процес детальніше зображено на рисунку 1.2 [4].

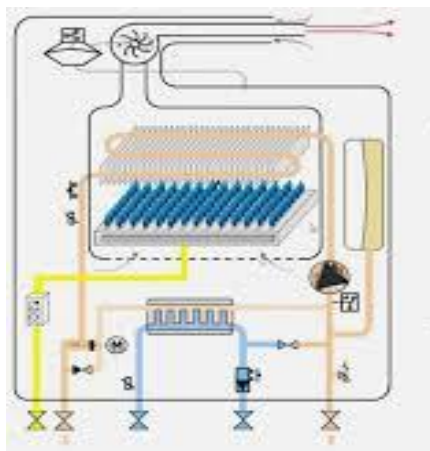


Рисунок 1.2 – Механізм роботи газового котла

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Цей фрагмент механізму функціонування базується на двох принципах: процес горіння та перетворення теплової енергії. Він починається з того що в камеру згоряння потрапляє суміш пропану й бутану для наступного процесу горіння. Наступним етапом є початок процесу запалювання, що в свою чергу включає процес горіння газу. В наслідок наступного процесу створюється водяна пара високої температури і в наслідок хімічної реакції утворюється карбон чотири оксид. Задля найбільш ефективного конвертування енергії в енергію теплообміну даний процес мусить відбуватися регулярно з неймовірною швидкістю, завдяки чому енергія стабільно створюється та цим створюються умови для нормативного опалення в газовому теплообміні в середині котла.

Враховуючи даний процес дуже корисно би було повністю автономно регулювати кількість подачі газу. Завдяки чому процес стане кращим завдяки виправленню і вдосконаленню багатьох параметрів:

- виконуваність роботи пальника і якість ;
- надійність системи газопостачання;
- якість роботи механізму димовидалення;
- точністю регулювання та стабілізацією параметрів;
- тепло ефективністю;
- низька ймовірність небезпеки;
- простота режимів праці;
- зрозумілість системи.

Найбільш часте використання газового опалення і через це найбільшої необхідності його вдосконалення потребують наступні ситуації:

- якщо система опалення є дуже старою і нестабільною, що її заміна є неможливою і вона потребує більш стабільної роботи;
- якщо є потреба частої зміни рівня подавання газу;
- якщо система створює зависоку чи недостатню температуру всередині.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

До інших переваг даної системи не зв'язаних з якістю роботи чи обміну тепла належать наступні випадки застосування:

- для запобігання великого й швидкого збільшення тиску в цій системі в наслідок випаровування води;
- для контролю рівня витоків газу і його припинення чи запобігання;
- задля виправлення нестабільності в роботі системи у всіх випадках які цього потребують.

Газові системи опалення дуже широко використовуються у важливих для життя громадян установах, таких як: школи, лікарні, місця тимчасового перебування, а також коледжі й вузи. Виробництво пропонує нам різні за розмірами та областю застосувань газові котли. До найвідоміших побутових котлів належать міні Eolo 28 та КС-Г-100 . Для великих підприємств доволі часто використовують котел марки Колві модель RS, який є високоефективним в наслідок великого об'єму обробки палива, а саме від 0,25 до 5,5 т/год. Приклад цього котла зображений на рисунку 1.3 [5].



Рисунок 1.3 – Котел Колві RS

Найбільш важливим аспектом безпеки не враховуючи контролю витоку газу є регулювання температури води в цій системі. Стабільною температурою води для внормованої роботи системи являється +60 °С. Аварійно-критичною

					КС КРБ 123.205.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

температурою роботи вважається від +80 °С, чого необхідно запобігати в чому стоїть головна задача в контролі температури води.

Підводячи підсумок потрібно зазначити, що створена система моніторингу та регулювання температури теплоносіїв у газових котлах дозволяє ефективно регулювати параметри середовища його застосування, забезпечуючи його автономну й стабільну свою роботу й безпечне виконання функцій.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА РЕГУЛЮВАННЯ ТЕМПЕРАТУРИ ТЕПЛОНОСІЇВ У ГАЗОВИХ КОТЛАХ

2.1 Розробка узагальненої структури комп'ютерної системи

На рисунку 2.1 зображена структурна схема комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах.

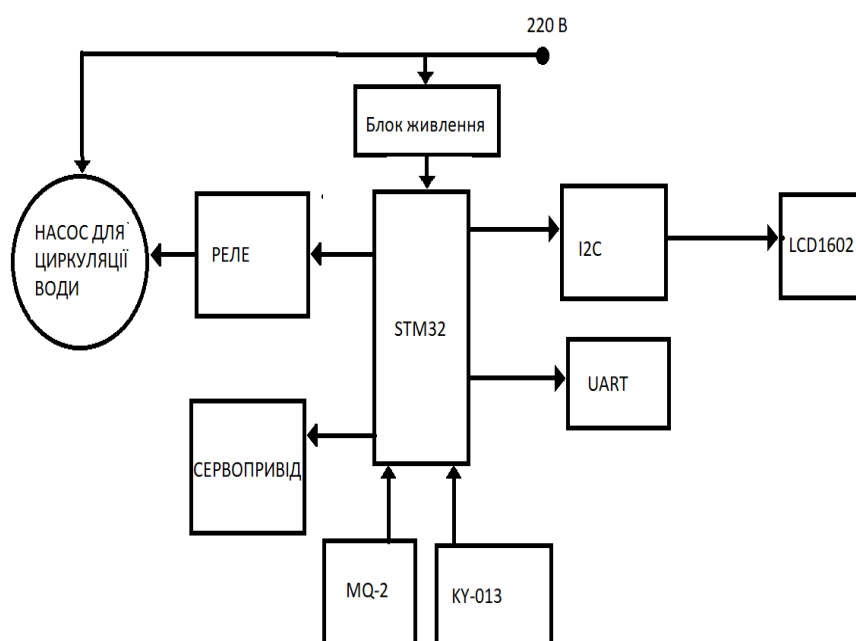


Рисунок 2.1 – Структурна схема комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах

Дана система складається з наступних складових:

- STM32 – це серце і мозок системи. МК виконує зчитування інформації подальшу її обробку та керує зовнішніми пристроями газового котла;
- дисплей LCD1602 – містить 2 рядки по 16 символів тексту;

					КС КРБ 123.205.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Жук І.Д.І.			Проектування системи моніторингу та регулювання температури теплоносіїв у газових котлах	Літ.	Арк.
Перевір.		Яцишин В.В.					18
Реценз.		Мудрик І. Я.				ТНТУ, каф. КС, гр. СІ-41	
Н. Контр.		Луцук Н.С.					
Затверд.		Осухівська Г.М.					

- блок живлення – подає напругу 5 В на STM32, що вимагає більшість важливих пристроїв в системі, таких як дисплей, проте інші живляться від напруги в 3.3 В, що є можливим завдяки стабілізатору напруги;
- MQ-2 – датчик газу, призначений виявляти витoki газу в котлі;
- UART – призначений для швидкісної передачі інформації та зберігання й ведення історії подій в системі й у самому газовому котлі;
- насос для циркуляції води – призначений для обміну води в системі опалення будинку, в випадку великих приміщень їх є декілька. Для ефективної роботи системи необхідно деколи ввімкнути даний насос аби знизити температуру теплоносія газового котла, що в свою чергу допомагає регулювати температуру приміщення, а також запобігає критичні ситуації. Завдяки даному насосу дана система опалення й має нормальний принцип роботи, який забезпечений алгоритмом роботи при перегрівих;
- сервопривід – призначений для регулювання подачі газу у котел;
- I2C – шина обміну даних призначена для спрощення роботи з екраном;
- Реле – призначене для ввімкнення насоса для циркуляції води;
- KY-013 – призначений для вимірювання температури води, теплоносія газового котла.

2.2 Обґрунтування вибору апаратного забезпечення проєктованого комп'ютерного засобу

2.2.1 Мікроконтролер

Для використання системи було обрано МК STM32F103C6T6. Дана плата для розробки містить 32-бітний мікропроцесор, вона зображена на рисунку 2.2.

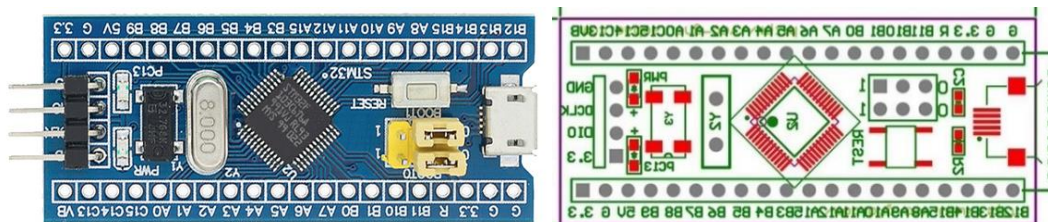
На платі є всі необхідні елементи для роботи з цим МК:

- 37 портів GPIO, що розведені на окремі контактні майданчики;

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

- має доволі малий форм-фактор, тому її друга назва Blue Pill (таблетка);
- потужне ядро ARM Cortex-M3 з частотою 72 МГц, яке забезпечує;
- обчислювальну потужність значно кращу за Arduino UNO чи 16/8-бітні аналоги;
- робоча напруга 2.0-3.6В і широкий температурний діапазон від -40 до +85°C;
- два 12-бітних АЦП, три 16-бітних таймери, PWM-таймер, інтерфейси;
- I2C, SPI, USART, USB та CAN для максимальної гнучкості проектів;
- роз'єм micro USB (з'єднаний з апаратним USB інтерфейсом);
- присутні пара світло-діодів: червоний відповідає за живлення, зелений відповідно до розпіновки за роботу порта GPIO C13;
- два ряди пінів забезпечують доступ до всіх функцій мікроконтролера при збереженні компактності для інтеграції в будь-які проекти.

Прошивати даний МК можна через програматори типу SWD, найкращий з яких ST-Link, також можливий варіант прошивання використовуючи необхідні USB-UART конвертери. Його прошивка унікальна, бо підтримує такі програми як: Arduino IDE, Mbed, STM32Cube, PlatformIO і багато інших програм забезпечують гнучкість їхнього вибору.



Таблиця 2.1 – Плата для розробки STM32F103C6T6

					КС КРБ 123.205.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Характеристики МК STM32F103C6T6 [6]

Ядро	ARM 32-bit Cortex-M3
Напруга живлення плати	2,0...5 В
Максимальна тактова частота	72 МГц
Об'єм флеш пам'яті	64 Кб
Об'єм оперативної пам'яті SRAM	8 Кб
Кількість портів GPIO	37
Кількість АЦП	1х 12bit (16 каналів)
Діапазон робочих температур	-40°C..+85°C
Комунікаційні інтерфейси:	до 2-х I ² C до 3-х USART до 2-х SPI 1× CAN 1× USB (Full Speed)

Завдяки даним характеристикам цей МК повністю підходить для системи.

2.2.2 Сенсори для вимірювання температури

Для цього проєкту було обрано модуль термістора під назвою KY-013 відповідно до [7]. Модуль термістора KY-013 зображений на рисунку 2.3 – це датчик, що вимірює температуру в режимі реального часу, це відбувається за допомогою терморезистора. Плата модуля складається з SMD резистора з значенням в 10 кОм та датчика температури KY-013. Даній резистори зібрані по схемі резисторного ділення напруги.

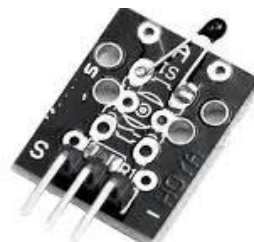


Рисунок 2.3 – Датчик температури KY-013

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Характеристики датчика температур КУ-013 показані в таблиці 2.2.

Таблиця 2.2 – Характеристики датчика КУ-013 [8]

Межі вимірювання температури	від -55° до +125°
Точність вимірювання температури	0.5°
Межі вимірювання вологості	від 0 до 100%
Напруга живлення	від 3,3 до 5 В
Розміри модуля	19 x 15 x 13 мм

Принцип роботи датчика полягає в тому, що він зібраний по схемі резисторного ділення напруги. Тобто, при зміні значення температури прямопропорційно міняється значення температурного датчика, що й змушує до зміни напруги при виведенні даних. Контролер отримавши дані від даного датчика, й отримані дані про напругу сенсора конвертує в градуси використовуючи формули логарифмічного перетворення. Тобто він значення напруги переводить в температурні значення.

В нього також є широкий спектр способів до застосування. Він може вимірювати температуру повітря в середовищі, а також в рідинах, якщо його ізолювати в термоклей. Крім цього він здатен вимірювати температуру поверхонь твердих тіл. Для цього лише необхідно його прикріпити за допомогою спеціальних прикріплень і нанести шар термопровідної пасти. В випадку відсутності заданої пасти його можна приклеїти на клей. Завдяки цьому датчику можна буде вимірювати стан компонентів комп'ютерних чи вичислюючих систем, забезпечуючи їм захист від перегріву [9].

На заданому модулі є три виходи для підключення до пінів МК. Контакт VCC відповідає за подачу струму на модуль, відповідно контакт GND є заземленням. По контакту S передається вся аналогова інформація до МК.

Для виводу правильної інформації з сенсору при його початку експлуатації нам необхідно дотримуватись наступних кроків:

- використовувати теплопровідні шари речовин для відповідності

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

вимірів до дійсних температур;

- необхідно подати живлення відповідно до необхідного діапазону значень, для більшого діапазону краще подати живлення в 3,3 В.

Сайт обчислення параметрів терморезистора зображено на рисунку 2.4 [10].

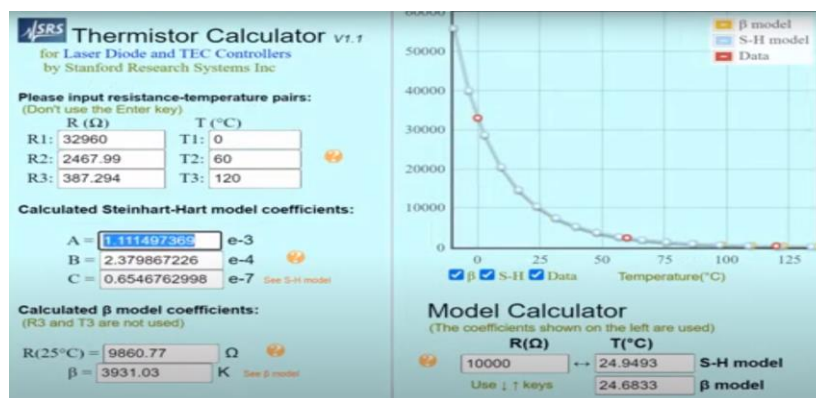


Рисунок 2.4 – Сайт Thermistor Calculator

Перед використанням необхідно виміряти опір терморезистора при трьох різних температурах, щоб отримати три різні сталі зміни, завдяки яким під час програмування в певній формулі буде обчислюватися точна температура.

2.2.3 Дисплей

Було прийнято рішення обрати для проєкту LCD1602 дисплей. Його технічні характеристики відображено в таблиці 2.3.

Таблиця 2.3 – Характеристики дисплея LCD1602 [11]

Назва	LCD1602
Живлення	5 В
Кількість символів	32
Інтерфейс	I2C
Робоча температура	До +50 °

Сам дисплей зображений на рисунку 2.5.

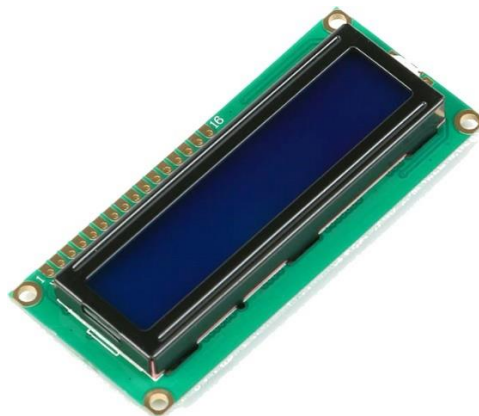


Рисунок 2.5 – Вигляд LCD1602 дисплея

Сам дисплей містить по 16 символів в кожному з двох рядків, чого повністю достатньо аби виводити всю потрібну інформацію.

2.2.4 Реле

Для регулювання температури в теплоносії газового котла необхідно буде використовувати насос для циркулювання теплоносія. Для здатності керувати насосом в нашій системі необхідне реле. Реле що розроблені для використання з МК містять в схемі гальванічну розв'язку з живленням (рис. 2.6). Через що це сильно зменшує навантаження на стабілізатори напруги 3,3 В, саме реле потребує більшу напругу на живлення, а саме 5 В.



Рисунок 2.6 – Реле вигляд моделі для МК

Саме реле буде повністю живитися від окремого джерела, а від мікроконтролера буде подаватись лише напруга в 3,3 В. МК STM32 буде

живити тільки один резистор, керуючий оптопарний та сигнальний світлодіоди.

Реле комутує максимального споживача з наступними властивостями:

- Максимальна напруга: 250 В;
- Максимальний струм: 10 А;
- Максимальна потужність: 2500 Вт.

2.2.5 Реле

Щоб упростити процес роботи з дисплеєм на рівні апаратному, так і на рівні програмному, тобто під час програмування, необхідно використовувати послідовний інтерфейс I2C. Даний інтерфейс зображений на рисунку 2.7.

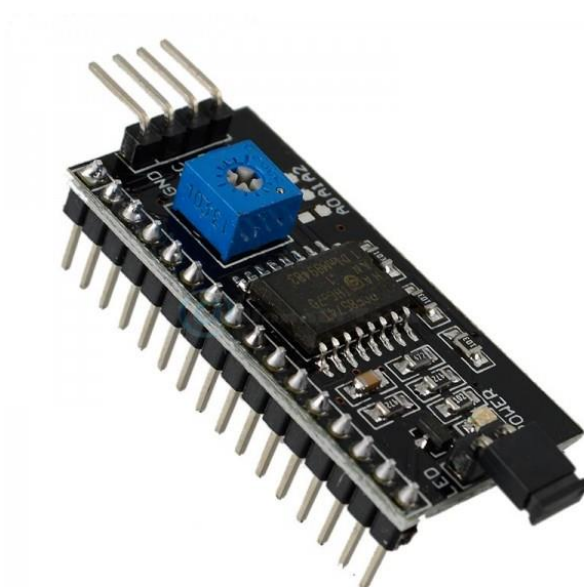


Рисунок 2.7 – Приклад моделі послідовного інтерфейсу I2C для дисплею

Відповідними плюсами в його використанні є те, що завдяки ньому буде використовуватися менше пінів МК. Також для налаштування контрастності екрану використовується регулятор в вигляді змінного резистора, в свою чергу регуляція відбувається за допомогою викрутки або подібного до неї інструменту.

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

2.2.6 Реле

Найкращим варіантом у виборі сервоприводу для проєкту є модель під назвою Servo Tower Pro 9g SG90 (рис. 2.8). Це один з найпоширеніших і один з найвідоміших сервоприводів, які можна використати в даній роботі. Ще важливою перевагою даного сервоприводу являється його низька ціна. Поворот механізму можливий на кут до 180° , що повністю задовольняє потреби нашої системи у контролі пристрою для подачі газу. Даний SG90 містить наступні характеристики:

- Робоча напруга: 3.5-5 В
- Зусилля утримання (при 4.8В): 2 кг/см
- Робоча швидкість (при 4.8В): 0.12 сек/60 град без навантаження
- Споживаний струм утриманні: 5-10 мА
- Кут повороту: 90/180 °
- Робоча температура: -30°C - $+60^{\circ}\text{C}$
- Споживаний струм у русі: 50-80 мА
- Ширина мертвої зони: 4 мікросекунди
- Розміри: 3.3 см x 3 см x 1.3 см
- Вага: 9 грам



Рисунок 2.8 – Сервопривід SG90

Механічний регулятор подачі газу (рис. 2.8) знаходиться в передній стінці більшості моделей газового котла. Враховуючи це монтаж сервоприводу на сам важіль є простою задачею.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.9 – Терморегулятор для газового котла Житомир 2

Верхній важіль в регуляторі буде місцем де буде вмонтовуватися сам сервопривід для контролю подачі газу.

2.2.7 Датчик газу

Датчиком газу для системи було вибрано MQ-2 [12]. Однією з переваг даного датчику в нашій системі є наявність двох виходів для передачі усієї інформації на мікроконтролер. Він виводить аналогові дані про вміст газу в повітрі, а також і цифрові дані. Це означає що є режим виводу стану коли присутній газ і коли його немає по принципу бінарної системи як 1 і 0, а також він може мати певне числове значення вмісту газу і тоді можна буде поставити межові значення для цього при програмуванні нашої системи згідно з [13]. Даний модуль має швидкий час реакції і він має велику чутливість. Даний модуль має наступні характеристики:

- Детектований газ: Горючий газ (вуглеводневі гази), дим;
- Діапазон чутливості: 300-10000 ppm;
- R_s (опір чутливого елемента) 1...20 кОм 50ppm толуол;
- Час відгуку ≤ 10 с
- R_h (опір нагрівача) $31\Omega \pm 3\Omega$;

					КС КРБ 123.205.00.00 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

- I_h (струм нагрівача) $\leq 180\text{mA}$;
 - V_h (напруга нагрівача) $5\text{V} \pm 0,2\text{V}$;
 - P_h (потужність нагрівача) $\leq 900\text{mW}$;
 - V_c (напруга схеми) $\leq 24\text{V}$;
 - Умови зберігання Температура: $-20 \sim +70^\circ\text{C}$, вологість: $\leq 70\%\text{RH}$;
- Даний модуль зображений на рисунку 2.10.



Рисунок 2.10 – Модуль MQ-2

Також до переваг даного модуля можна віднести й те, що він може виміряти вміст диму й інших газів, також.

2.2.8 Датчик газу

В нашій системі буде використовуватися USB-UART конвертер на чіпі FTDI FT232RL. Це пов'язано з його найбільшою поширеністю. Даний інтерфейс використовуватиметься для передачі даних через нього на комп'ютер з метою ведення історії подій і станів теплоносія газового котлу. Даний конвертер підтримує роботу від 3,3 В до 5 В. Принцип його роботи полягає в перетворенні паралельних даних процесора в послідовний потік бітів для передавання. Сама передача відбувається з відсутністю тактового сигналу проте там використовуються стоп і стартові біти для синхронізації процесу передачі даних.

Його дизайн є мінімалістичним і сильно підходить для даної системи. Даний конвертер зображений на рисунку 2.11.

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28



Рисунок 2.11 – USB-UART конвертер

Також для простоти розуміння його принципу роботи він містить 2 світлодіодні індикатори роботи.

2.3 Проектування електрично принципової схеми

2.3.1 Середовище проектування електричних схем

Для розробки електричної принципової схеми комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах було обрано середовище проектування EasyEDA. Це ПЗ є призначеним для спрощення процесу розробки електричних схем та створення монтажних плат.

Основним критерієм вибору ПЗ є його активна підтримка користувачами, і в разі відсутності певного компоненту в стандартних бібліотеках, є високий шанс що він знаходиться в користувацьких.

2.4 Проектування комп'ютерного засобу

2.4.1 Вибір алгоритму «регуляції»

Алгоритм процесу регуляції теплоносія полягає в тому, щоб виміряти температури (t , $^{\circ}\text{C}$) та вміст газу (C , $[\text{кг}/\text{см}^3]$) у середовищі газового котла для окремого середовища де встановлений сам газовий котел, щоб забезпечити повний контроль над усіма витоками газу опалення, також необхідно буде вимірювати температуру в теплоносії газового котла. Враховуючи виміряні

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

значення необхідно визначити стан газового котла. Порівнявши порогові значення з даними на цей момент вимірювання й виконувати дії необхідні для складеної ситуації. В ситуаціях відповідних до критичних температур для газового котла буде необхідно виконати циркуляцію води теплоносія в приміщенні чи в будинку, відповідно увімкнувши необхідний насос.

Для того аби знати точні температури теплоносіїв необхідно розрахувати за формулою відношення опорів терморезисторів:

$$T = 320 - (1 / (a + (b \cdot \ln(R)) + (c \cdot \ln^3(R))), \quad (2.1)$$

де:

$$R = (33000 / ((R_{term} \cdot U_{term}) / 4095)) - 10000, \quad (2.2)$$

R – обрахований опір терморезистора ,

R_{term} – не обрахований опір терморезистора,

U_{term} – напруга живлення терморезистора (залежить від необхідного діапазону значень температури, 5 В для меншого діапазону, а 3,3 В для більшого)

a , b , c – необхідні сталі коефіцієнти терморезистора для конвертування явної температури, визначаються за допомогою трьох дослідів, де записується різний опір даного резистора при різних температурах середовища, обчислюється за допомогою спеціальної програми.

Описаний вище алгоритм мусить спрацьовувати з чіткою періодичністю в 3 секунди для стабілізації точних вимірів температури теплоносія відповідно до заданої в теперішніх.

2.4.2 Розробка схеми підключення сенсорів до мікроконтролера

Вибір портів GPIO для рішення задачі проєкту в табличному вигляді зображено в таблиці 2.4.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.4 – Розподілення портів GPIO

Змінна в програмі	Порт STM32	Порт пристрою
Датчики газу та терморезистора		
gas1	PA0	MQ-2 датчик газу
adc_raw	PA1	KY-013 датчик температури
LCD дисплей підключено до комунікаційного інтерфейсу I2C		
LCD_SCL	PB6	SCL дисплея
LCD_SDA	PB7	SDA дисплея
UART для передачі даних		
RX_TX	PA10	TX
TX_RX	PA9	RX
Керування сервоприводами регулятора подачі газу		
TIM2_CH2	PB3	Регулятор подачі газу
Керування реле		
PIN_RELAY	PB12	реле подачі живлення на насос для циркуляції теплоносія
Вбудований світлодіод		
SD_LED	PC13	Світлодіод на МК світиться при критичних ситуаціях

В випадку використання датчика газу використовується або аналогове або цифрове обчислення значень. Для точності визначення вмісту газу в середовищі рекомендовано використовувати цифрову форму. Перевагою такого способу є перед всім можливість встановлення порогового значення при якому значення вмісту газу в приміщенні є критичним. Однак у випадку необхідності додавання додаткового датчику у новіших версіях даної комп'ютеризованої системи варто використовувати аналогове обчислення значень. Тому що це дасть змогу підключити ще один датчик, наприклад температури зовнішнього приміщення до ШІМ. Для зміни стану зчитуємої інформації необхідно лише апаратно поміняти певний порт підключений до МК. В цьому випадку значення вмісту газу будуть рівні 1, якщо газ є в середовищі, 0, якщо газ відсутній в середовищі.

В новіший версіях даної системи можна буде покращувати не тільки кількість датчиків, но і якість індикатора для відображення інформації.

2.4.3 Опис електрично принципової схеми

Дана електрична схема комп'ютеризованої системи була розроблена на основі структурної схеми даної системи (див. рис. 2.1). Головним компонентом цієї схеми є МК STM32, який позначений на схемі під найменуванням U1. Напруга +5 В надається через блок живлення.

На рисунку 2.14 зображено електричну принципову схему системи моніторингу та регулювання температури теплоносіїв у газових котлах.

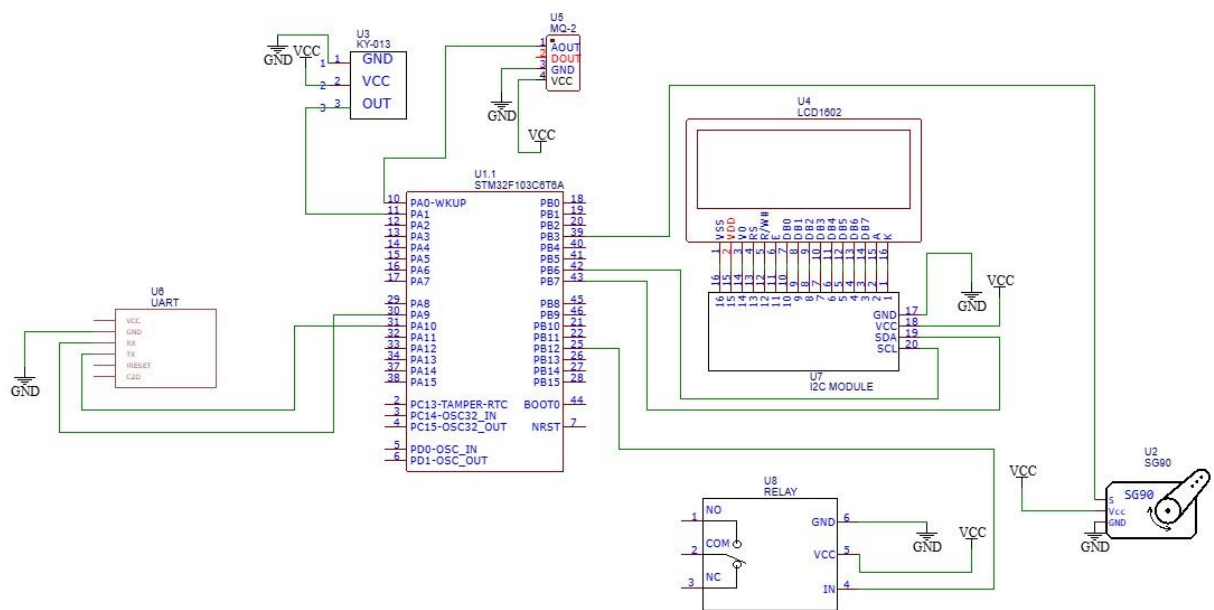


Рисунок 2.14 – Електрична принципова схема системи моніторингу та регулювання температури теплоносіїв у газових котлах

Завдяки даній електричній принциповій схемі простота складання даної системи є повністю зрозуміла і спрощує реалізацію даної системи.

РОЗДІЛ 3 РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ ТА РЕГУЛЮВАННЯ ТЕМПЕРАТУРИ ТЕПЛОНОСІЇВ У ГАЗОВИХ КОТЛАХ

3.1 Реалізація і моделювання проєктних рішень

На рисунках 3.1 і 3.2 було зображено детальну блок-схему алгоритму праці програми для МК даної системи.

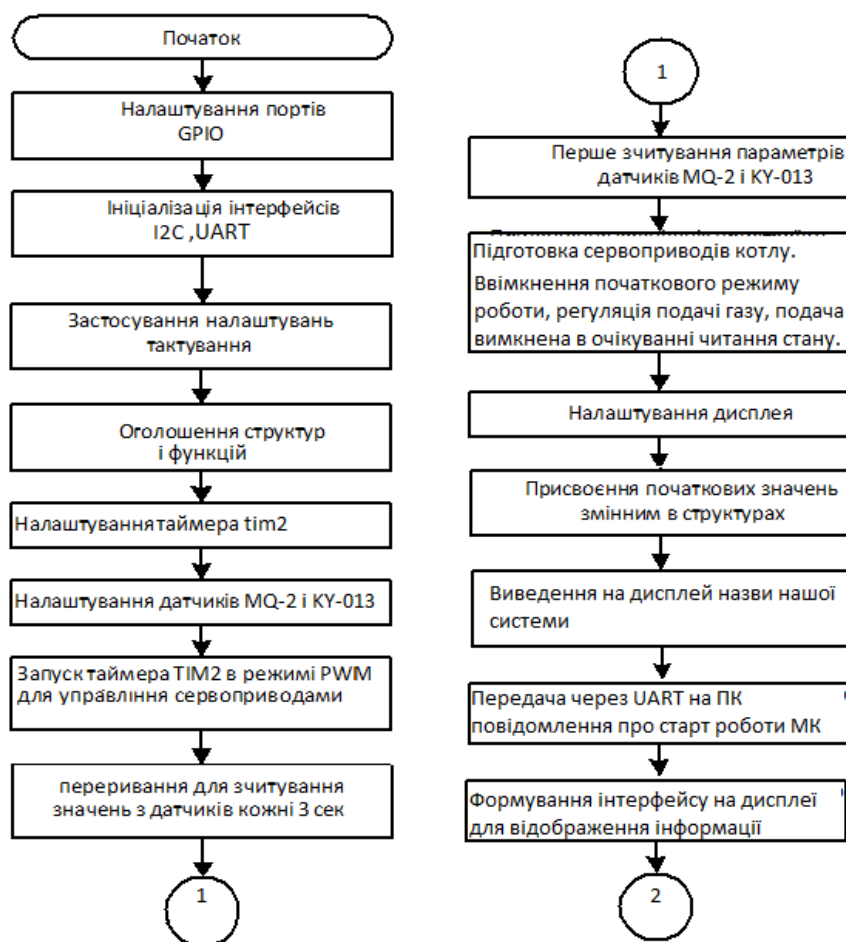


Рисунок 3.1 – Алгоритм роботи системи моніторингу та регулювання температури теплоносіїв у газових котлах

					КС КРБ 123.205.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Жук І-Д.І.			Розробка алгоритму та програмного забезпечення системи моніторингу та регулювання температури теплоносіїв у газових котлах	Літ.	Арк.
Перевір.		Яцишин В.В.					33
Реценз.		Мудрик І. Я.				ТНТУ, каф. КС, гр. СІ-41	
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					

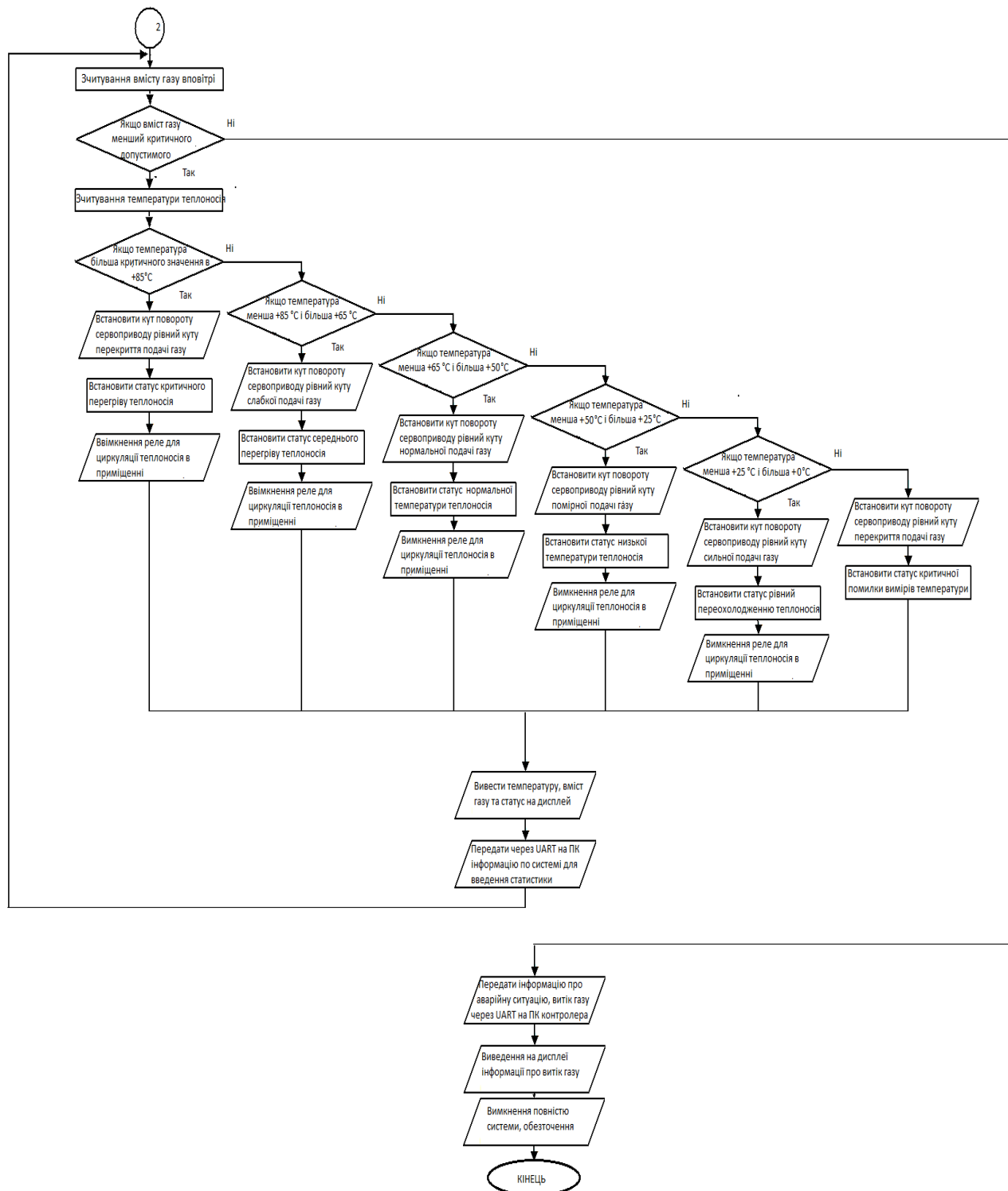


Рисунок 3.2 – Алгоритм роботи системи моніторингу та регулювання температури теплоносіїв у газових котлах (продовження)

Опис роботи даного алгоритму було детально приведено в пункті 3.1.2 «Опис програми».

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.205.00.00 ПЗ

Арк.

34

3.1.1 Налаштування середовища розробки програмного забезпечення

Програмне забезпечення розроблялась в спеціальному середовищі STM32CudeIDE. Для При виборі мови програмування найкращим варіантом являється вибір мови C. Одними з її переваг є її поширеність в даній галузі і її час швидкодії, котрий являється одним з найвищих серед усіх мов програмування. Після детального аналізу доступних джерел в мережі інтернет було вибрано необхідні бібліотеки та модулі для UART, дисплея, датчиків з джерел: [14, 15, 16], котрі було додано до проєкту, котрий створеного в програмному середовищі STM32CudeIDE.

Перед створенням програмного забезпечення даної системи в середовищі STM32CudeIDE було ввімкнено та налаштовано:

– Необхідні комутаційні інтерфейси:

а) I2C1 для підключення LCD1602 дисплея;

б) UART для модуля передачі даних.

– Порти GPIO у відповідності з таблицею 2.4.

– Необхідні таймери:

а) TIM2, канал 1 в режимі роботи генерації широтно-імпульсної модуляції, ШІМ (PWM – англ. pulse-width modulation) для контролю положення сервоприводу;

б) RTC, (англ. Real time clock) годинник реального часу.

3.1.1.1 Налаштування тактування мікроконтролера

Система для управління над тактуванням в середовищі розробки STM32CudeIDE називається RCC (reset and clock control). Для здобуття максимально можливої частоти на МК було налаштовано RCC (рис. 3.3). В першу чергу було обрано кварци для генерації тактових сигналів. Для кожного з генераторів було вибрано наступні зовнішні кварци: High-speed external (HSE) і Low-speed external (LSE), оскільки вони мають найкращу стабільність роботи. Крім того, для точної та коректної роботи в годиннику реального часу (RTC) необхідно обрати зовнішній кварц LSE.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

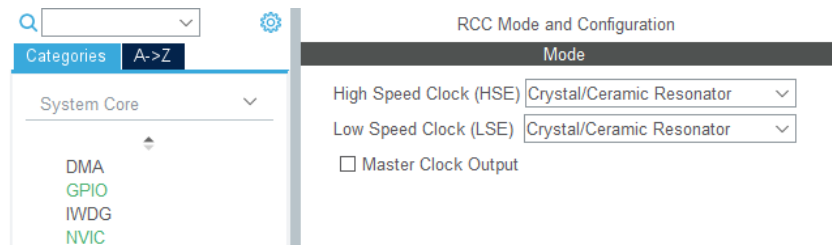


Рисунок 3.3 – Налаштування RCC

Було налаштовано RTC (календар та годинник), як показано на рисунку 3.4.

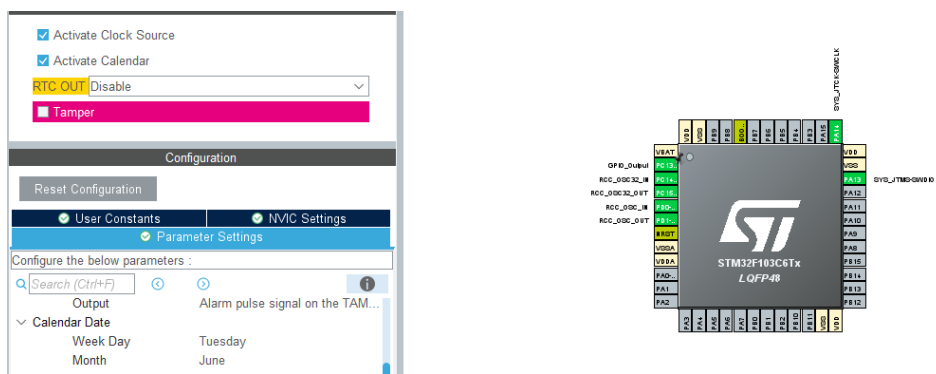


Рисунок 3.4 – Налаштування RTC

Результат налаштування тактування зображено на рисунку 3.5.

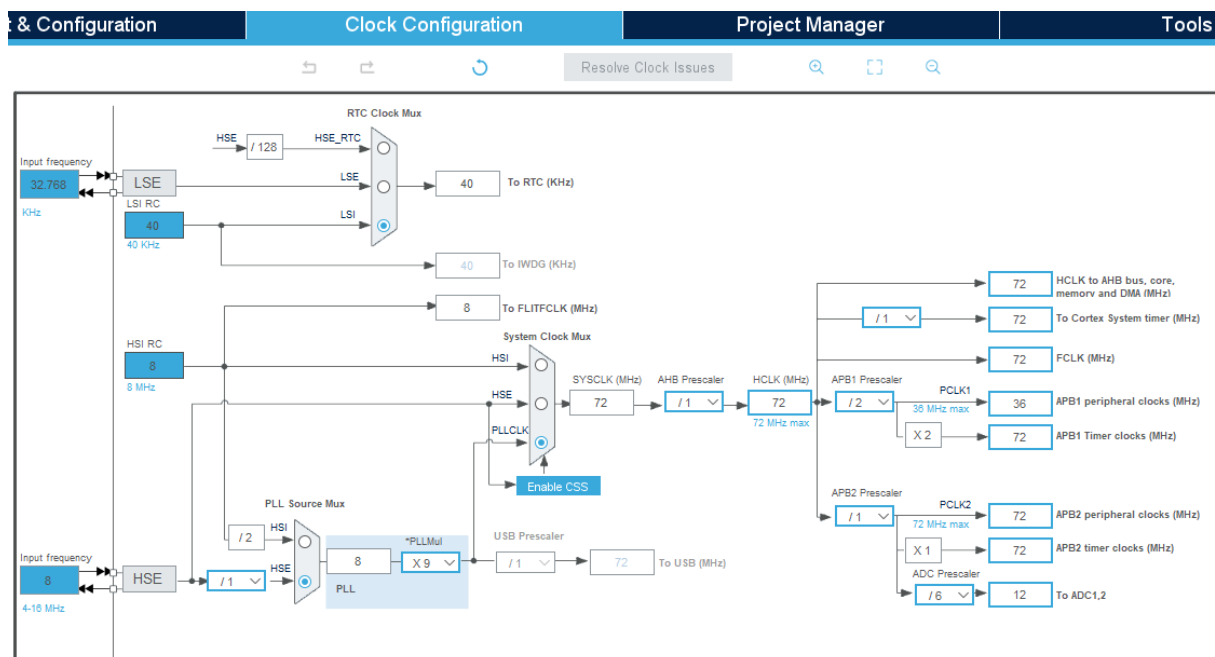


Рисунок 3.5 – Налаштування тактування

На мікроконтролерах серії STM32f103 максимальна частота тактовання є 72 MHz. Джерелами тактування являються вбудований і зовнішні кварци HSI і HSE відповідно. Також для тактування використовується PLL. PLL – це множник, котрий збільшує значення тактування від x2 до x16 разів. Щоб частота була максимальною необхідно поставити множник на x9.

3.1.1.2 Налаштування тактування мікроконтролера

Оскільки для підключення LCD дисплею було використано I2C модуль, що дозволило використовувати меншу кількість з'єднань, то для підключення даного дисплею через I2C модуль необхідно було увімкнути комунікаційний інтерфейс I2C, так як зображено на рисунку 3.6.

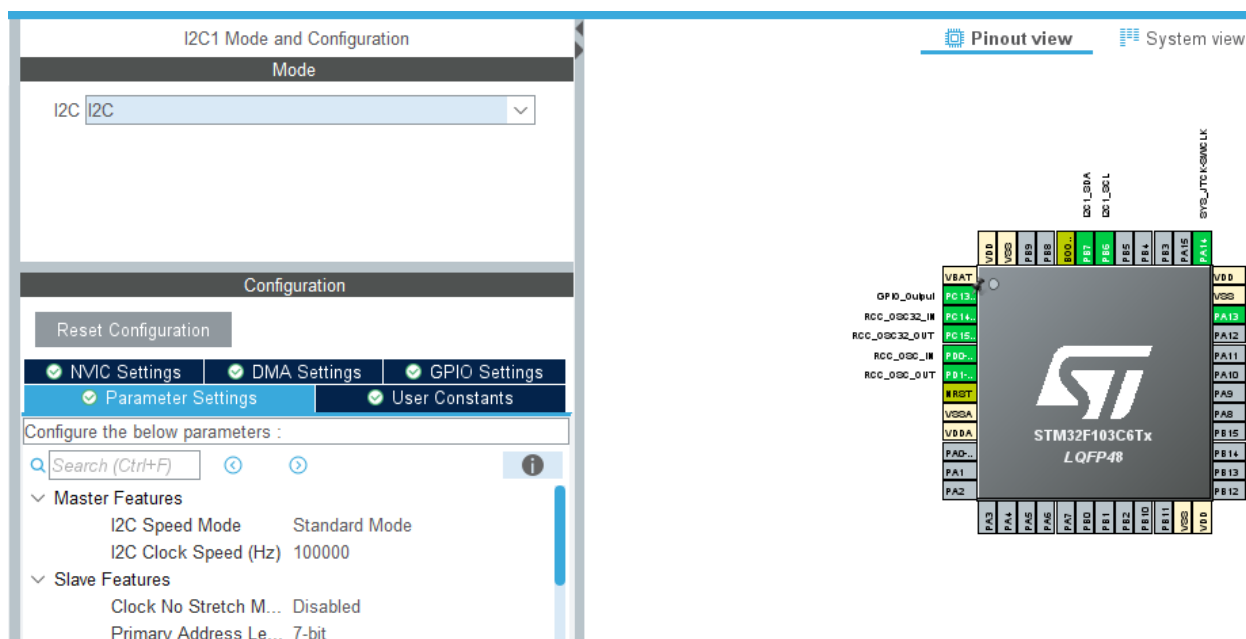


Рисунок 3.6 – Увімкнення та налаштування шини I2C

UART (Universal Asynchronous Receiver/Transmitter) являється апаратним модулем, котрий використовується для послідовного асинхронного обміну даними.

Для підключення модуля передачі даних налаштовано перший комунікаційний інтерфейс USART, як зображено на рисунку 3.7.

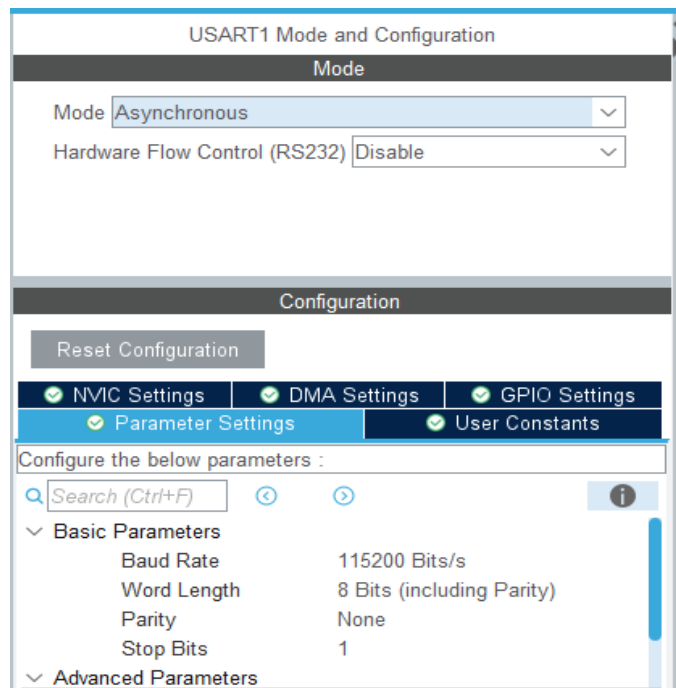


Рисунок 3.7 – Увімкнення та налаштування шини USART1

Завдяки ньому відбувається передача або ж приймання даних з комп'ютера. Його принцип роботи є доволі простим, оскільки UART використовує тільки два сигнали, а саме: TX (Transmit) є для передачі даних, а RX (Receive) в свою чергу є для їхнього приймання. Обмін відбувається асинхронно, тобто без окремого сигналу тактування, що відрізняє його від SPI та I2C.

Під час передачі даних кожен байт інформації, котра передається, автоматично додається старт-біт в вигляді логічного нуля. Після чого далі передаються решта бітів даних, в більшості випадків передаються 8 бітів, після чого передається стоп-біт в вигляді логічної одиниці. Кількість бітів, що передаються на пряму залежать від самих налаштувань модулю. Для передачі використовується регістр USARTx->DR.

Під час отримання даних приймач UART очікує на старт-біт, після чого отримує всі решта бітів. Після прийому байту всі дані передаються в буфер і після цього встановлюється прапор прийому даних. Дані можна зчитати з того ж самого регістру USARTx->DR.

До основних регістрів UART (USART) належать наступні регістри:

USARTx->SR, USARTx->DR , USARTx->BRR, USARTx->CR1, CR2, CR3. Регістр USARTx->SR використовується для перевірки прапорів, таких як: TXE, RXNE, ORE і інші. USARTx->DR використовується для читання або передачі даних. Регістр USARTx->BRR встановлює швидкість передачі інформації (англ. baud rate). USARTx->CR1, CR2, CR3 – це регістри керування, котрі використовують для вмикання UART, виконання переривань або встановлення кількості стоп-бітів.

У вікні можна керувати завантаженнями файлів: додавати нові torrent-файли, видаляти завантаження, зупиняти і запускати завдання.

3.1.1.3 Налаштування портів GPIO

Налаштування для портів вводу-виводу GPIO було виконано у відповідності з таблицею 2.4. Результат налаштування зображено на рисунках 3.8 – 3.12.

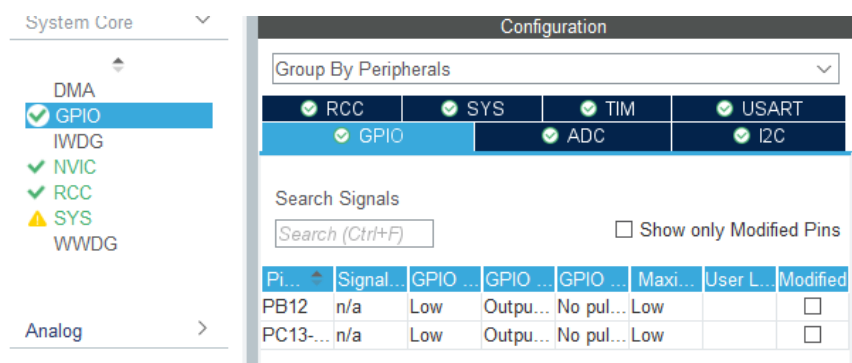


Рисунок 3.8 – Налаштування GPIO

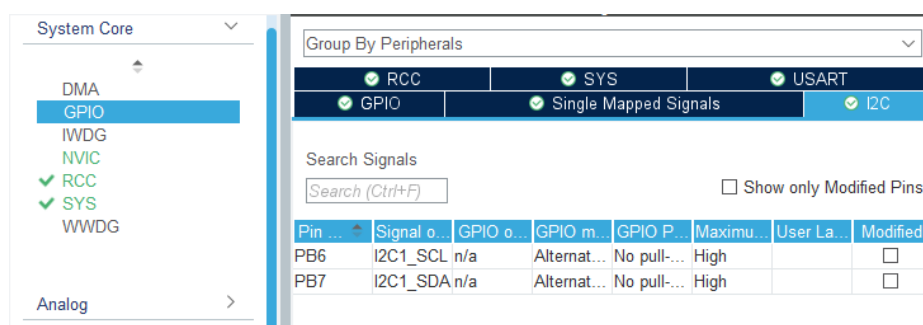


Рисунок 3.9 – Налаштування портів I²C

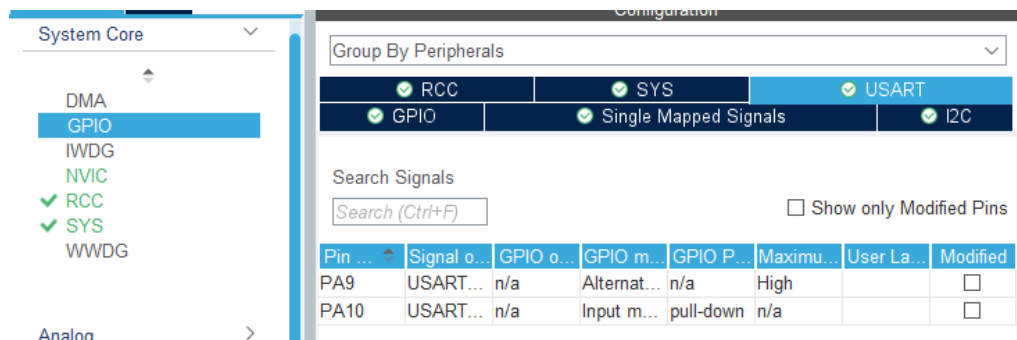


Рисунок 3.10 – Налаштування портів USART

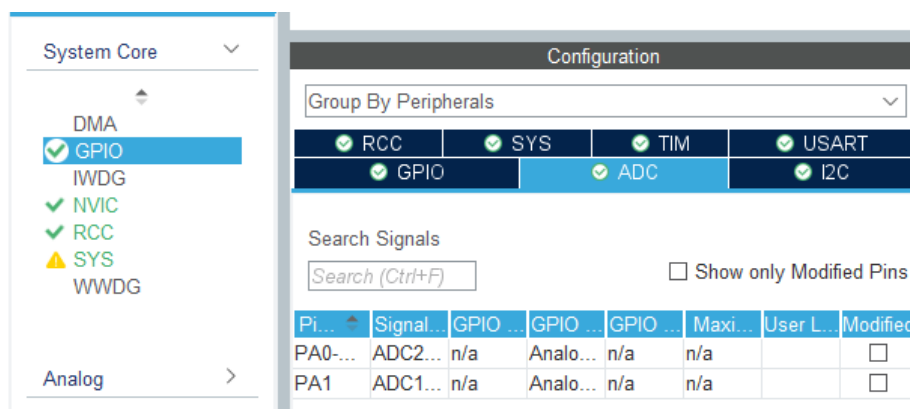


Рисунок 3.11 – Налаштування портів ADC

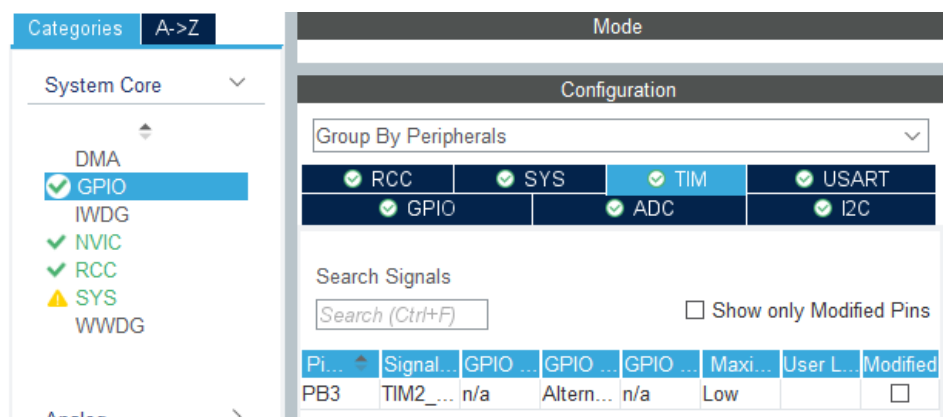


Рисунок 3.12 – Налаштування портів, що використовуються таймерами

GPIO порти використовуються для підключення реле. Для реле використовується режим підключення GPIO_output, даний режим також використовується для вбудованого в мікроконтролер світлодіода PC13. Для підключення сервоприводу використовується таймер TIM2_CH2.

Для зчитування інформації з датчиків використовується режим

ADC1_IN5 і ADC2_IN3. З датчиків передається інформація про температуру теплоносія, а також про вміст газу в повітрі в середовищі розміщення газового котла з метою виявлення і забезпечення безпеки в небезпечних ситуаціях.

Для передачі інформації про наш газовий котел підключений до нашої системи використовується два порти для USART: PA9, PA10. Перший порт, котрий підключений в режимі USART1_TX і другий порт знаходиться в режимі USART1_RX. Для виведення інформації на дисплей використовуються порти PB7, PB6 в режимах I2C1_SCL і I2C1_SDA відповідно.

3.1.1.4 Налаштування таймерів

3.1.1.4.1 TIM2

Використовуючи таймер TIM2 реалізується генерація ШІМ для керування сервоприводом SG90. Його налаштування показано на рисунку 3.13.

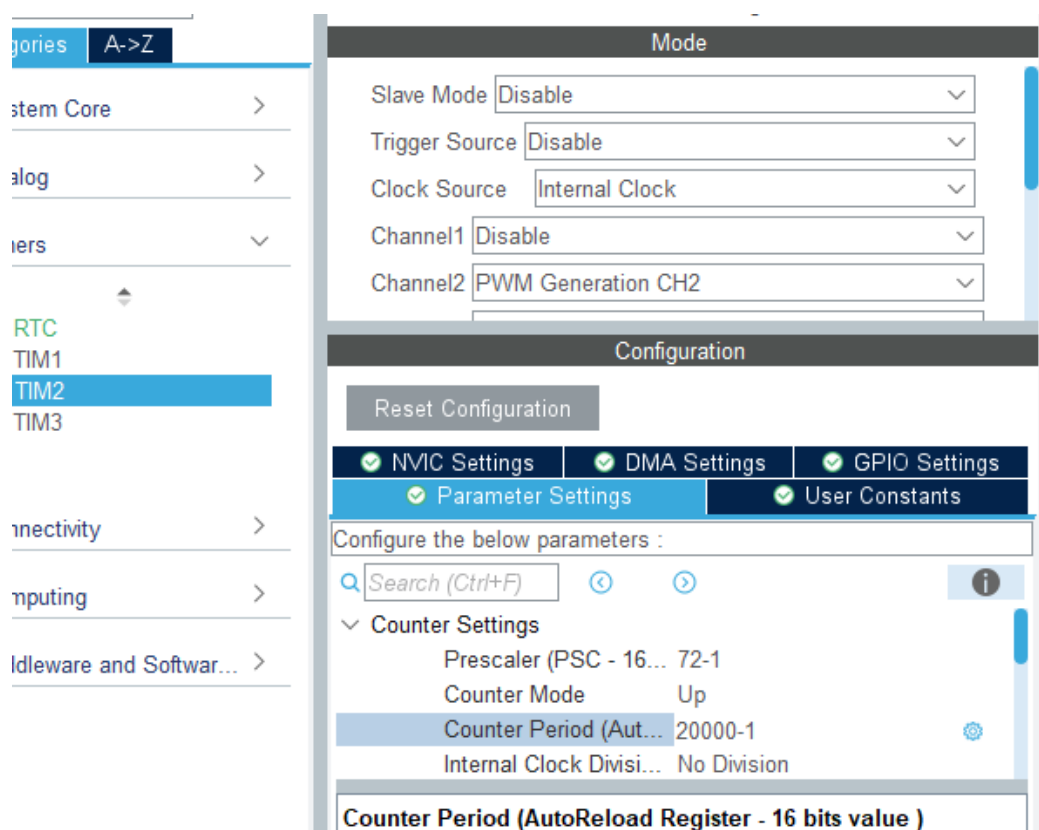


Рисунок 3.13 – Налаштування таймера TIM2

Для налаштування таймера, щоб той використовувався для сервопривода необхідно його правильно налаштувати. Сервопривід працює в режимі ШІМ при сигналі в частоті приблизній 50 Гц.

$$Prescaler = \frac{Timer\ clock}{Frequency \times Counter\ period}, \quad (3.1)$$

де:

- Timer clock – це тактова частота самого МК STM32F103C6T6 є 72 МГц;
- Frequency – це необхідна частота для таймера;
- Prescaler – це передільник, який зменшує частоту подаванню на таймер;
- Counter period – це період підрахунку, котрий визначає кількість імпульсів таймера до повного циклу.

Для даної формули було виставлено наступні значення параметрів: Counter period в 20000 і Prescaler в 72. Таким чином було отримано значення 50 Гц для роботи ШІМ. Проте, оскільки в ПЗ STM32CubeIDE рахунок починається з нуля, то необхідно від цих значень відняти одиницю.

В коді програми для мікроконтролера необхідно додати одну строку аби ініціалізувати таймер в проєкті (рис. 3.14).

```
/* USER CODE BEGIN 2 */

// Таймер htim2

// запуск таймера 2 в режимі ШІМ для сервопривода
HAL_TIM_PWM_Start_IT(&htim2, TIM_CHANNEL_2);
```

Рисунок 3.14 – лістинг фрагменту коду для активації ШІМ

ШІМ розшифровується як штучно імпульсна модуляція. ШІМ – це метод керування над середнім значенням напруги або потужності, котрий реалізується шляхом швидкого вмикання чи вимикання з різною тривалістю його вмикання в одному періоді.

Контролери серії STM32 мають апаратні таймери, котрі можуть

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

автоматично генерувати ШІМ-сигнали без участі ЦП. Для цього необхідно встановити частоту і тривалість імпульсу, а таймер сам буде формувати прямокутні імпульси на конкретному вибраному перед тим порті МК.

У коді необхідно додати ще один фрагмент для передавання значення ШІМ на сервопривід SG90 для зміни кута повороту самого сервопривода. За допомогою цього буде регулюватися кількість подачі гасу в котел. Даний фрагмент зображено на рисунку 3.15.

```
//Встановлення куту повороту сервоприводу
__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_2, j);
```

Рисунок 3.15 – фрагмент коду для управління сервоприводу

Зміна “j” є зміною, котра вказує кут повороту сервопривода, котрий встановлюється в декілька відповідних кутів відповідно до статусу системи. Для критичного перегріву подача газу перекривається, отже кут повороту сервопривода встановлюється рівним нулю, і відповідно значення даної змінної теж рівне нулю.

3.1.2 Налаштування середовища розробки програмного забезпечення

За допомогою програми STM32CubeIDE було згенеровано шаблон коду з усіма налаштуваннями проекту.

Головною функцією програми є функція main(). На початку функції було написано ініціалізацію портів GPIO, таймерів, усіх потрібних комунікаційних інтерфейсів, структури та змінні, котрі були запрограмовані в коді програми. Також в початку цієї функції є виведення на дисплей і через UART початкового повідомлення.

Далі в коді програми буде цикл while в котрому буде виконуватимуться регулювання та моніторинг над температурою теплоносія в газовому котлі. Проте, даний цикл не буде безкінечним, перед початком циклу і після його початку буде проводитися перевірка на витоки газу у середовищі розміщення газового котла. Головною умовою цього циклу буде чи вміст газу не

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

перевищує порогові значення, в протилежному випадку цикл завершиться й проведеться аварійний завершальний протокол його роботи.

В основному циклі роботи системи передбачено п'ять станів газового котлу і ще один при виникненні помилки в вимірах. Стани встановлюються відповідно до температури теплоносія і відповідно до станів виконуються певні алгоритми, котрі забезпечують безпеку ситуацій експлуатації газового котла та дотримання комфортної температури для роботи газового котла і відповідно для комфортної температури приміщення. Присутні два стани перегріву теплоносія: сильного і помірного, - при їхньому виявленні комп'ютеризована система вмикатиме насос для циркуляції теплоносія, тим самим охолоджуючи теплоносію, а також буде або перекриватись подача газу, або виставлятиметься мінімальне значення відповідно до цих двох станів. В випадку переохолодження теплоносія в котлі є також два стани системи, а саме: сильного й слабого переохолодження. В обох цих станах буде вимикатись насос для циркуляції теплоносія в приміщенні, оскільки потреби в його охолодженні немає. В випадку сильного переохолодження рівень подачі газу в нашу систему буде близький до високих значень, аби якомога швидше встановити комфортні температури в котлі і в приміщенні. А в випадку слабшого переохолодження теплоносія газового котла рівень подачі газу буде вищим за середній, щоб якомога безпечніше перейти в комфортний статус даної комп'ютеризованої системи. В комфортному статусі температура теплоносія буде підтримуватись в значеннях приблизних до 60°C, відповідно рівень подачі газу буде середнім, а насос для циркуляції теплоносія задля захисту від переохолодження буде вимкнено. Для визначення необхідного стану системи буде зчитуватись значення з терморезистора KY-013, і відповідно до них буде встановлюватись стан системи. Після встановлення стану системи буде виводитись інформація про газовий котел і про комп'ютеризовану систему, а саме її стан. Дана інформація буде відображатись в нижній частині дисплею і в верхній частині буде відображатись інформація про температуру теплоносія, а також про вміст газу

					КС КРБ 123.205.00.00 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

в повітрі. Також дані дані будуть передаватись в термінал головного ПК через модуль передачі даних UART. Весь цикл while буде повторюватись з періодичністю в декілька секунд, приблизно в дві чи три, котрі будуть необхідні для точного функціонування системи.

Після даного циклу реалізований фрагмент коду для дій в випадку виявлення аварійної ситуації витoku газу в системі опалення газового котлу. Першою й основною дією в даній ситуації буде перекриття подачі газу аби забезпечити його різке зростання витoku, або ж й перекрити сам витік. Далі буде виведене повідомлення про високу небезпеку витoku газу на дисплей і також дане повідомлення буде передане на ПК керуючого аби попередити його про дану ситуацію аби було вжито запобіжних засобів, провітрення приміщення й виклику службу газу за номером 104 і також евакуюватися з приміщення. Останнім, но найважливішим аспектом є обезструмлення системи, спочатку вимкнеться насос для циркуляції теплоносія, якщо він був ввімкнутий, потім вимкнеться сама система, щоб зменшити ймовірність вибуху в приміщенні.

Даний код повністю зображений в додатку В. В ньому вказано всі функції, параметри і структури. Також в цьому коді повністю детально описано всі частини коду коментарями для кращого розуміння написаного коду. Завдяки даним коментарям стає більше зрозуміло сам принцип роботи, а також, спрощує подальшу можливість доопрацювання і покращення коду в новіших версіях даної комп'ютеризованої системи.

Дана версія коду була провірена дебагером і там були виправленні всі помилки, включаючи також і небезпеки, завдяки чому даний код повністю робочий і підготовлений до роботи з даним мікроконтролером й усіма складовими системи.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.2.1 Функції

Для коректної роботи даної програми було написано функції, весь їхній перелік, а також призначення було приведено в таблиці 3.1 згідно з [17].

Таблиця 3.1 - Перелік основних функцій, створених в програмі

Функція	Призначення
void LSD_WRITE_I2C (uint8_t bt);	передача одного байта через шину I2C, встановлює адресу I2C
void E_set (uint8_t data);	встановлює біт E = 1, потрібний для зчитування команди
void E_reset (uint8_t data);	встановлює біт E = 0, закінчуючи подачу команди
void RS_set(uint8_t data);	встановлює біт RS = 1, режим даних
void RS_reset (uint8_t data);	встановлює біт RS = 0, режим команди
void sendhalfbyte (uint8_t c, uint8_t mode);	передає пів байта інформації на шину I2C
void sendbyte (uint8_t c, uint8_t mode);	передає повний байт на LCD дисплей
void LCD_SenChar (char ch);	виведення символу типу char
void LCD_Clear (void);	очищення дисплею від усіх виведених на нього символів
void LCD_init (void);	ініціалізує LCD дисплей для роботи з ним
void LCD_Set_Pos (uint8_t x, uint8_t y);	виставлення курсору на дисплеї для виведення символу
void LCD_String (char* st);	виведення набору символів на дисплей
void LCD_Gradusi(uint16_t pwm);	виведення кута повороту сервоприводу на дисплей
void LCD_Run(uint8_t krug, char word[]);	виведення повідомлення біжучої стрічки на екран
void send_data_UART(float temp, float gas, char *stan);	передавання інформації про систему через UART
uint64_t convHexToASCII (uint16_t value);	конвертування чисел з десяткових до ASCII таблиці
void LCD_Numb (int numner);	виведення числа на дисплей
void LCD_NumbwithDot (float numner);	виведення числа з біжучою комою на екран
void TempMONITOR();	вимірювання температури за допомогою терморезистора
void kutpovorotu(int grad);	встановлення кута повороту сервоприводу

Більшість функцій виконуються для керування LCD-дисплея на шині I2C. Основні функції були описані в розділі 3.1.2.1.1.

3.1.2.1.1 Функції

Головною функцією програми є функція `int main(void)`. В програмі вона є стартовою для системи. Першими в даній функції буде виконуватися наступне:

- Процес ініціалізації таймерів, всіх портів GPIO та всі комунікаційні інтерфейси (UART, I2C);
- Встановлюються початкові значення для змінних;
- виконуються необхідні налаштування;
- вмикається АЦП для подальшої роботи;
- отримання перших вимірів вмісту газу в середовищі розміщення системи.

Після чого виконуються цикл з умовою щоб кількість газу була менша небезпечних значень, в котрому виконується зчитування значень температури і вмісту газу, і керування газовим котлом за допомогою елементів системи.

В кінці функції встановлений код для повідомлення про витік газу і виконання вимкнення системи для зменшення ймовірності детонації приміщення внаслідок загоряння газу від електричного струму.

Зчитування першого значення вмісту газу необхідно виконати перед циклом задля безпеки системи при її ввімкненні. Фрагмент коду в якому виконується перше зчитування вмісту газу представлено на рисунку 3.16.

```
// першочергове зчитування даних вмісту газу
gas1 = HAL_ADC_GetValue(&hadc2);
float voltage = (gas / 4095.0f) * 3.3f;
float ppm = pow(10, ((voltage - 0.1f) * 2.7f)); // груба модель
gas = ppm/4370000; // конвертування значень в шкалу до 100
```

Рисунок 3.16 – першочергове зчитування з MQ-2

Після чого виконується цикл з умовою вмісту газу нижчою за допустиме значення. Даний цикл зображений в додатку В. В ньому усе необхідне підписане коментарями. Варто зазначити, що умова його зображена на рисунку 3.17 і відрізняється від безкінечного циклу згенерованого програмою CubeIDE.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

/* USER CODE BEGIN WHILE */
while (gas < 70)
{
    /* USER CODE END WHILE */

```

Рисунок 3.17 – умова циклу while

Завдяки даному циклу й виконуються основні задачі системи. В даному циклі крім функцій, котрі відповідають за керуванням шиною I2C та LCD дисплеєм є також і дві дуже важливі функції котрі використовуються для керування сервоприводом та модулем UART. Функція для керування сервоприводом зображена на рисунку 3.18. А функція для керування передачею даних через UART зображена на рисунку 3.19.

```

void kutpovorotu(int grad){
    float j;
    if (grad > 180) grad = 180;
    j = (grad * 5.55555555556) + 250; // 5.55555555556 = 1*, from
    250* to 1250 is from 0* to 180*
    __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_2, j);
}

```

Рисунок 3.18 – функція для керування сервопривода

Дана функція конвертує значення кута в градусах в необхідне для сервопривода значення, котре й потрібне для встановлення певного кута. В даній функції отримане число в градусах перетворюється за допомогою деяких математичних функцій. Значення в градусах на котрі може змінювати кут повороту сервопривід це від нуля до 180 о, проте значення, котрі необхідно подавати від МК до SG90 від 250 до 1250, де 250 рівне куту повороту в 0 о, а значення 1250 рівне 180 о.

Далі після формули дане значення присвоюється змінні і вона використовується в керуванні над сервоприводом. Для того, щоб в програмному коді можна було реалізувати всі необхідні функції для даних формул необхідно підключити стандартну бібліотеку мови C під назвою “math.h”.

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48


```

void send_data_UART(float temp, float gas, char *stan) {
    char buffer[100];

    // Температура (1 знак після коми)
    int temp_int = (int)(temp * 10);
    sprintf(buffer, "Temperature: %d.%01d\r\n", temp_int /
    10, temp_int % 10);
    HAL_UART_Transmit(&huart1, (uint8_t*)buffer,
    strlen(buffer), 1000);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_Delay(100);

    // Газ (1 знак після коми)
    int gas_int = (int)(gas * 10);
    sprintf(buffer, "Gas value: %d.%01d\r\n", gas_int / 10, gas_int
    % 10);
    HAL_UART_Transmit(&huart1, (uint8_t*)buffer,
    strlen(buffer), 1000);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_Delay(100);

    // Статус (char)
    sprintf(buffer, "Status of System: %s\r\n", stan);
    HAL_UART_Transmit(&huart1, (uint8_t*)buffer,
    strlen(buffer), 1000);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_Delay(100);
}

```

Рисунок 3.19 – функція для передачі даних по UART

Коментарями в даній функції відзначенні фрагменти коду, котрі відповідають відповідно за передачу певного рядка з певними даними. Завдяки цій функції всі дані про стан системи, температуру теплоносія, вміст газу передаються на ПК, де й видно журнал подій в терміналі.

Також після циклу міститься аварійний фрагмент коду, котрий виконується тільки при великому вмісті газу в середовищі, тобто після завершення циклу. В ньому написані всі необхідні операції дій при виникненні даної надзвичайної ситуації. Даний фрагмент програмного коду представлений на рисунку 3.20.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

LCD_Clear();
//виведення на дисплей помилки
LCD_Set_Pos(0,0);
LCD_String("  -=AVARNING=-  ");
LCD_Set_Pos(0,1);
LCD_String("- VUTIK GAZU -");
strcpy(stan, "!GAS ALERT!");
//передача інформації про небезпечний стан на ПК
sprintf(charData, "VUTIK GAZU, PROVITRIT PRIMISHENYA, CALL
104, GO EVERYONE OUTSIDE NOW %s\r\n", stan);
HAL_UART_Transmit(&huart1, (uint8_t
*)charData, strlen(charData), 1000);
HAL_Delay(2000);
//вимкнення системи
HAL_PWR_EnterSTANDBYMode();

```

Рисунок 3.20 – лістинг аварійного вимкнення й виведення повідомлення про стан небезпеки

Завдяки такому завершенню програма виведе повідомлення, котре може врятувати життя. Дане повідомлення містить вказівки на подальші дії в виникненні такої ситуації, а також містить номер служби газу для їхнього подальшого виклику. Перед вимкненням дана система вимикає насос для циркуляції теплоносія. Також система знеструмиться, щоб не детонувати з витікшим газом.

3.1.3 Інтеграція механічних елементів з мікроконтролером

3.1.3.1 Живлення насоса для циркуляції теплоносія

Насос для циркуляції теплоносія отримує живлення від реле, котре було детально описано в розділі 2.2.4. під назвою «реле». Дані керуючі сигнали для реле передаються відповідно з таблицею 2.4.

Варто зазначити саме реле живиться від напруги 5 В, чого повністю достатньо для його коректної роботи. Принцип керування МК над реле подібний до принципу керування над світлодіодом.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.3.2 Живлення насоса для циркуляції теплоносія

Керуючий сигнал широтно-імпульсної модуляції для сервоприводу використовував згенерований відповідний МК. Практичний досліджування показало, що достатньо логічного рівня 3,3 В для коректного позиціонування кута сервоприводу. Одним з важливих чинників правильного функціонування сервоприводу є необхідність подачі правильної напруги на МК, тобто потрібно подавати на нього напругу в 5 В, замість напруги в 3,3 В котра отримується через програматор ST-Link.

3.2 Тестування

Задля проведення тестування ситеми було виготовлено її тестову версію на макетній платі. Дана система зображена на рисунку 3.21.

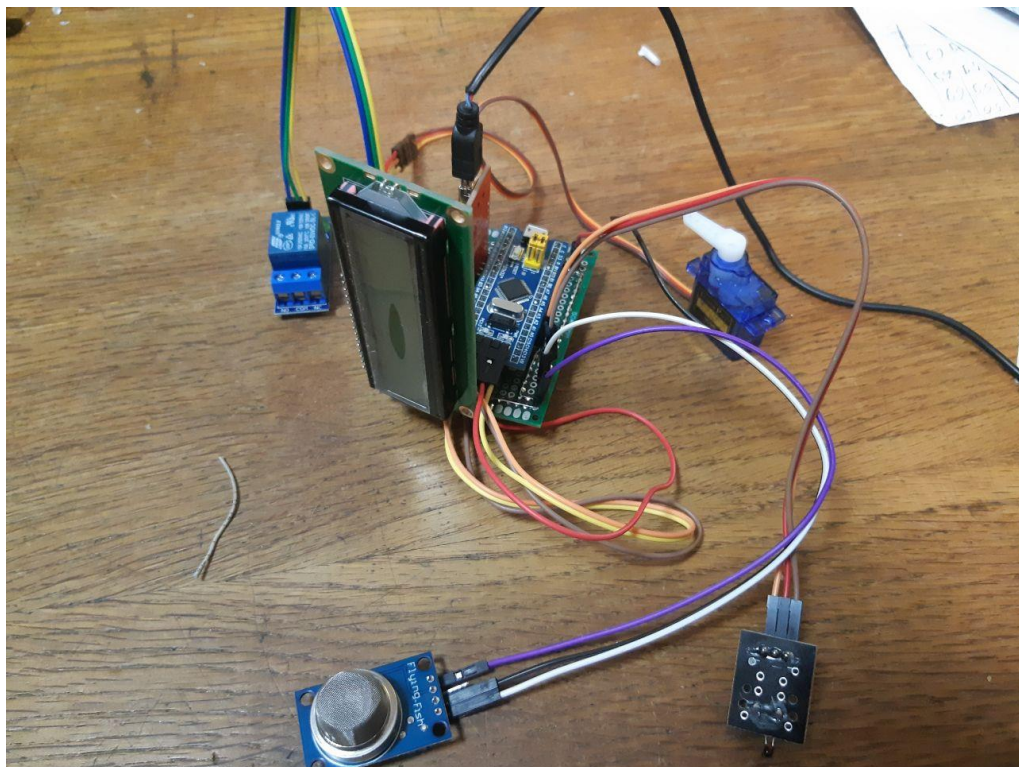


Рисунок 3.21 – Тестова версія системи спроектована на макетній платі

На даній макетній платі були змонтовані всі її елементи, крім реле для керування насоса теплоносія: симулюючи його роботу можна за допомогою

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

світлодіода. При тестуванні датчик KY-013 для зчитування температури теплоносія було виведено на провід.

3.2.1 Інтерфейс

Інтерфейс відображується на LCD1602 дисплеї, він складається з двох інформаційних рядків, котрі містять три параметри.

На інформаційному дисплеї зображено:

- стартове повідомлення з тестовою назвою системи (рис. 3.22);
- інформація про стан роботи газового котла (температура в середині теплоносія, вміст газу в повітрі, назва стану системи) зображено на рисунку 3.23;
- поточні значення всіх параметрів;
- повідомлення про критичний витік газу при виявленні такої ситуації, для повідомлення користувача перед вимкненням системи (рис. 3.24).

На інформаційному екрані завжди, крім випадків ввімкнення системи та виявлення витіку газу, відображається теперішня температура теплоносія і вміст газу разом з повідомленням про стан системи. Завдяки даному інтерфейсу користувач може легко спостерігати за станом і роботою системи і відповідно газового котла.



Рисунок 3.22 – Стартове повідомлення



Рисунок 3.23 – Інтерфейс з параметрами системи



Рисунок 3.24 – Аварійний інтерфейс

Даного інтерфейсу достатньо для виведення всіх необхідних параметрів для даної системи. До даних параметрів належать: значення температури теплоносія, вміст газу опалення в середовищі розташування газового котла і інформація про поточний статус системи.

3.2.2 Тестування режимів роботи

Після завершення коригування датчиків, дану систему було протестовано при різних станах її роботи:

- Сильне переохолодження. Циркуляція теплоносія через насос вимкнена (керуюче реле вимкнено). Високий рівень подачі газу в котел опалення. Працює при відсутності витoku газу опалення та при температурах нижче +25 °C і нище 0 °C.

- Переохолодження. Циркуляція теплоносія була вимкнена (насос через реле вимкнено). Рівень подачі газу в котел вищий середнього. Працює при

відсутності витoku газу опалення і при температурах від +25 °С до +50 °С.

- Нормальна температура. Циркуляція через насос вимкнена (керуюче реле вимкнено). Рівень подачі газу в середній. Працює при відсутності витoku газу опалення й при температурах від +50 °С до +65 °С.

- Перегрів. Циркуляція теплоносія через насос увімкнена (керуюче реле увімкнене). Рівень подачі газу в низький. Працює при температурах від +65 °С до +85 °С та при відсутності витoku газу опалення.

- Критичний перегрів. Циркуляція теплоносія через насос увімкнена (реле увімкнене). Вимкнена подача газу. Працює при температурах вище +85 °С й відсутності витoku газу.

Крім того, для перевірки роботи системи, а саме вмикання чи вимикання насосу і правильності регулювання подачі газу датчик КУ-013 було розташовано в тепле і холодне місце (на електроний камін і в холодильник).

За результатами тестування система показала коректну роботу у всіх станах роботи, тобто правильне керування зовнішніми пристроями (насосом циркуляції та важелем подачі газу) в залежності від значень температури і вмісту газу [18].

					КС КРБ 123.205.00.00 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

У кваліфікаційній роботі описується процес створення комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах, яка відповідно до застосування буде встановлена в приміщенні опалення чи котельній, тобто неподалік від газового котла опалення, тому у розділі розглядаються питання: моделювання та прогнозування небезпечних ситуацій, особливості заходів захисту безпеки роботи, організація безпечної роботи електроустановок. При введенні в експлуатацію системи враховано фактори безпеки й передбачення небезпечних ситуацій в автоматизованому режимі роботи.

4.1 Моделювання та прогнозування небезпечних ситуацій

Небезпечні події з несприятливими наслідками мають свої історичні назви, а саме: аварія, стихійне лихо, катастрофа [19]. Аварія – це небезпечна подія техногенного характеру, що створює загрозу для життя і здоров'я людини, призводить до руйнування будівель, споруд, об'єктів транспортування засобів, порушення виробничого чи транспортного процесу, або завдає шкоди довкіллю. Стихійне лихо – це небезпечна подія природного походження, яка за своєю інтенсивністю, масштабом поширення і тривалістю вражає людей, об'єкти економіки та довкілля. Катастрофа – це великомасштабна аварія, стихійне лихо чи інша подія, що призводить до тяжких, трагічних наслідків. Стан, який виникає після небезпечних подій є надзвичайною ситуацією.

Враховуючи особливу специфіку роботи газового котла необхідно модулювати і прогнозувати всі можливі небезпечні ситуації. Моделювання і

					КС КРБ 123.205.00.00 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Жук І-Д. І.			Безпека життєдіяльності, основи охорони праці	Літ.	Арк.	Аркушів	
Перевірив		Яцишин В.В.						55	
Консульт.		Пилипець М.І.				ТНТУ, каф. КС, гр. СІ-41			
Н. Контр.		Луцик Н.С.							
Затверд.		Осухівська Г.М.							

прогнозування небезпек на практиці проходить три стадії.

На першій стадії визначають матеріальні носії небезпек, тобто небезпечні та шкідливі чинники і умови, за яких вони можуть призвести до небажаних наслідків.

На другій стадії визначається головна небезпечна подія і послідовність інших небезпечних подій та умов, які їй передують. На цій стадії будується логічна схема розвитку небезпеки у вигляді дерева небезпечних подій та причин. Небезпечні події можуть відбуватися послідовно одна за одною, паралельно (одночасно) одна одній, а найчастіше – за змішаною послідовно-паралельною схемою. Для відображення схем реалізації небезпечних подій використовують логічні оператори «І» та «АБО». Логічний оператор «І» показує, що подія А відбудеться, якщо одночасно відбудуться всі події, які їй передують, тобто і Б, і також В [20].

На третій стадії аналізують можливі небажані наслідки і визначають можливі шляхи зменшення їхнього негативного впливу. В основу такого аналізу покладено методику порівняння затрат, спрямованих на відвернення небезпек (зменшення ризику), і вигод, отриманих від зниження рівня ризику. Суть цієї методики полягає в тому, що спочатку визначають затрати DT_j , які скеровані на зниження рівня ризику від j -го чинника (групи чинників) до величини ST_j , а потім оцінюють можливі витрати ST_j які виникають при здобуттому рівні ризику. Якщо можна виразити величини DT_i та ST_j в однакових одиницях виміру, то їхня сума становить загальні витрати суспільства, пов'язані з наявністю цього чинника (групи чинників). Якщо коефіцієнт ризику r_{TI} малий, то очевидно, що витрати ST_j , які виникатимуть унаслідок реалізації небезпек, будуть також малими (мала кількість аварій), але, щоб досягнути цього коефіцієнта ризику необхідні великі затрати DT на відвернення небезпек. І навпаки в разі великого коефіцієнта ризику [21].

В системі газового котла є дві значні матеріальні носії небезпек, а саме: труби подачі газу та середовище теплоносія. В першому випадку виникає проблема з витоком газу у навколишнє середовище, а в другому виникає

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

ймовірність перегріву теплоносія, що призведе до ймовірності детонування системи опалення. Умовами витоку газу є пошкодження труб його надання в наслідок механічного подразника чи в випадку корозії. Відповідними умовами для перегріву теплоносія є зависока температура газового котлу внаслідок зavelикого об'єму подачі газу. Передуючими умовами для цього є неправильне налаштування регулятора подачі газу, первопричиною чого є включення її на максимальні значення і несвоєчасність налаштувань на менші об'єми. Це може статись за умов неуважності і необачності людини, яка займалась регуляцією газового котла.

До небажаних наслідків в випадку перегріву системи опалення є збільшення тиску в середовищі зберігання теплоносія, що в свою чергу приведе до виходу зладу газового котла в наслідок вибуху середовища теплоносія. Серед небажаних наслідків стосовно витоку газу є два випадки. Перший це отруєння витікшим газом, яке може відбутися в наслідок сну, оскільки даний газ в природному вигляді немає запаху, проте аби вміти оприділяти його витoki його було змішано з іншим газом з різким неприємним запахом. Коли людина спить вона не здатна розрізняти запахи, отже це дає можливість отруєння природнім газом. Другим небажаним наслідком є ймовірність детонації приміщення через високий вміст природнього газу в середовищі, також для цього необхідне джерело вогню яким може стати будь який електроний пристрій.

З урахуванням цих небезпек й їхнім моделюванням розроблено в системі алгоритми їхнього застереження й регулювання.

4.2 Особливості заходів захисту безпеки роботи

Електроустановка – це установка, в якій виробляється, перетворюється, передається, розподіляється та споживається (перетворюється) електрична енергія [22].

За видами струму ЕУ поділяють на установки змінного та постійного

					КС КРБ 123.205.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

струму. ЕУ змінного струму поділяють на однофазні і трифазні. Однофазні електричні мережі (ЕМ) – це мережі, які живляться від окремої обмотки джерела живлення (ДЖ – генератора чи трансформатора), яка не має електричних з’єднань з іншими обмотками. Такі ЕМ застосовують вкрай зрідка. В абсолютній більшості випадків застосовуються трифазні ЕМ і трифазні споживачі. Трифазні ЕМ живляться від трьох обмоток і мають певні електричні з’єднання між собою.

Вимоги до заходів захисту від прямого дотику в нормальному режимі роботи електроустановок – це те, що струмовідні частини ЕУ не повинні бути доступні для випадкового прямого дотику до них, а доступні для дотику відкриті і сторонні провідні частини не повинні перебувати під напругою, що становить небезпеку ураження електричним струмом в нормальному режимі роботи і в разі пошкодження ізоляції.

Для запобігання ураженню електричним струмом у нормальному режимі роботи застосовуються окремо або у поєднанні такі заходи захисту від прямого дотику:

- основна ізоляція струмовідних частин;
- огорожі та оболонки в ЕУ;
- бар’єри в ЕУ;
- блокування безпеки в ЕУ.

Захист від прямого дотику не вимагається, якщо номінальна напруга ЕУ не перевищує:

- 25 В змінного або 60 В постійного струму в разі застосування системи ЗННН (система захисної наднизької напруги), якщо обладнання перебуває у зоні дії зрівнювання потенціалів і експлуатується тільки у сухих приміщеннях;
- 6 В змінного або 15 В постійного струму в усіх інших випадках використання пристроїв релейного захисту, що відключають пошкоджені ділянки кола;

Ізоляція – це шар діелектрика або конструкція, виконана з діелектрика, за допомогою якого струмовідні частини відокремлюються одна від одної чи

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

від інших конструктивних елементів обладнання. ЕУ передусім мають робочу ізоляцію – ізоляцію, яка забезпечує протікання струму потрібним шляхом і безпечну експлуатацію обладнання, тобто ізолює струмовідні частини від корпусів [23].

Контроль ізоляції підрозділяють на періодичний (виконують у певні проміжки часу); постійний (виконують впродовж усього періоду роботи ЕУ) і передпусковий (виконують перед пуском ЕУ). Періодичний контроль поділяють на:

- приймально-здавальний – після монтажу, перед введенням в експлуатацію;
- післяремонтні випробування – після капітальних та поточних ремонтів;
- міжремонтні випробування – у терміни, встановлені відповідними правилами.

Захист у разі непрямого дотику – захист, який запобігає ураженню людини електричним струмом у разі одиничного ушкодження. Для запобігання ураження струмом у разі ушкодження ізоляції застосовано окремо або у поєднанні такі засоби захисту в разі непрямого дотику.

4.3 Організація безпеки роботи електроустановок

Персонал, обслуговуючий електрообладнання, може потрапити під напругу внаслідок несправності, аварії або своїх помилкових дій. Безпека обслуговування електрообладнання залежить від його робочої напруги, умов експлуатації і характеру середовища приміщення, в якому воно встановлене.

Значною мірою безпека обслуговування електрообладнання залежить від умов середовища приміщення, в якому воно встановлене, бо ці умови впливають на стан ізоляції і опір шкіри людини. Волога, їдкі пари або газы, струмопровідний пил і висока температура знижують опір ізоляції і руйнують її. Крім того, шкіра людини під впливом вологи і високої температури стає

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

провідною, що зменшує опір тіла людини і підвищує небезпеку ураження електричним струмом.

Струмопровідна підлога (металева, цегляна, бетонна), на якій стоїть людина, різко зменшує опір його кола і підвищує небезпеку дотику до струмопровідних частин, що знаходяться під напругою. Провідний пил осідає на проводах і утворює провідне коло, внаслідок чого можливі замикання на землю і між фазами. їдкі пари або газу (хімічно активне середовище) руйнують ізоляцію і зменшують її опір.

Таким чином, ознаками підвищеної небезпеки є:

- волога (відносна вологість повітря вище 75%) або наявність струмопровідного пилу;
- струмопровідні підлоги;
- підвищена температура повітря (більше +35 ° C);
- можливість одночасного дотику людини до заземлених корпусів обладнання і до частин електрообладнання, що перебуває під напругою [23].

Ознакою особливої небезпеки являється наявність особливої вологості (відносна вологість повітря близька до 100 %) і хімічно активне середовище. За ступенем небезпеки ураження людей електричним струмом приміщення підрозділяють на класи:

- приміщення з підвищеною небезпекою (присутня лише одна ознака підвищеної небезпеки);
- особливо небезпечні приміщення (наявність хоча б однієї ознаки особливої небезпеки або одночасно двох чи більше ознак підвищеної небезпеки).

Об'єм і характер необхідних захисних заходів, що забезпечують безпеку, визначаються залежно від виду електроустановки, номінальної напруги, умов середовища приміщення і доступності електрообладнання.

Вимоги безпеки розробленої комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах враховано в розробленні КС.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було виконано наступне:

- Виконано огляд необхідних умов для функціонування комп'ютеризованої системи моніторингу та регулювання температури теплоносіїв у газовому котлі.
- Було розроблено узагальнену структурну схему даної комп'ютеризованої системи та було зроблено обґрунтування вибору апаратного забезпечення.
- Була розроблена електрично принципову схему системи моніторингу та регулювання температури теплоносіїв у газовому котлі.
- Було описано алгоритм для роботи даної системи, також було написано програмне забезпечення для даної системи.

Розроблена комп'ютеризована система моніторингу та регулювання температури теплоносіїв у газовому котлі. Дана система була зібрана у вигляді тестового прототипу, створена на основі мікроконтролера серії STM32.

За результатами тестування дана система показала повну роботоспособність і вірність роботи всіх станів відповідно до вимог технічного завдан.

Дана комп'ютеризована система готова до виконання задач щодо моніторингу та регулювання температури теплоносіїв в газовому котлі.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.
2. Українська норма: мінімум 18 °C (20 °C в кутових кімнатах) – постанова КМУ № 151. URL: <https://www.vingudpss.gov.ua/news/temperatura-povitrya-v-zhitlovih-primishchennyah/>. (дата звернення: 03.02.2025)
3. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с
4. Принцип роботи газового котла і різновиди опалювального обладнання. URL: <https://upts.com.ua/princip-roboti-gazovogo-kotla-i-riznovidi/>. (дата звернення: 05.02.2025)
5. Промислові парові котли типу RS – технічні характеристики. URL: <https://kolvi.com.ua/uk/produksiya/parovi-kotly/seriya-rs/>. (дата звернення: 06.02.2025)
6. Datasheet STM32F103C6T6.
7. Оконський М.В., Лупенко С.А., Паламар А.М. Комп'ютерна система для моніторингу метеорологічних параметрів на основі IoT. Актуальні задачі сучасних технологій : збірник тез доповідей X міжнародної науково-практичної конференції молодих учених та студентів (Тернопіль, 24–25 листопада 2021 року), Тернопіль: ТНТУ, 2021. С. 112.
8. KY013 Technical Documentation. URL: <https://sensit.joyit.net/en/sensors/>. (дата звернення 05.03.2025)
9. Оконський М.В., Лупенко С.А., Паламар А.М. Інформаційно-вимірювальна система для контролю метеорологічних параметрів на основі Інтернету речей. Матеріали IX науково-технічної конференції "Інформаційні

					КС КРБ 123.205.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

моделі, системи та технології" Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 8–9 грудня 2021 року), Тернопіль: ТНТУ, 2021. С. 118.

10. Thermoresistor Calculator. URL: <https://www.thinksrs.com/downloads/programs/therm%20calc/ntccalibrator/ntccalculator.html>. (дата звернення: 12.03.2025)

11. LCD1602 Module – DFRobot Technical Datasheet. URL: <https://dfimg.dfrobot.com/nobody/wiki/ee1c5bd150fc6b78f8cb8e8306898ab9.pdf>. (дата звернення: 14.03.2025)

12. MQ-2 Datasheets.

13. Ларіоник Р.В., Луцик Н.С., Паламар А.М. Система для моніторингу якості атмосферного повітря на базі IoT. Матеріали IX науково-технічної конференції "Інформаційні моделі, системи та технології" Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 8–9 грудня 2021 року), Тернопіль: ТНТУ. 2021. С. 116.

14. Getting started with UART. URL: https://wiki.st.com/stm32mcu/wiki/Getting_started_with_UART. (дата звернення: 02.04.2025)

15. How to Interface LCD1602 via I2C with STM32. URL: <https://controllerstech.com/2c-lcd-in-stm32/>. (дата звернення: 03.04.2025)

16. Getting started with ADC. URL: https://wiki.st.com/stm32mcu/wiki/Getting_started_with_ADC. (дата звернення: 04.04.2025)

17. Харченко О., Яцишин В. Розробка та керування вимогами до програмного забезпечення на основі моделі якості. Вісник ТДТУ. Тернопіль, 2009. Т. 14. №1. С. 201-207.

18. Слюз І., Жаровський Р. Принципи та основні етапи комплексного тестування комп'ютерної інформаційної системи. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (7-8 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 93.

19. Запорожець О.І. Безпека життєдіяльності. Підручник, 2-е

					КС КРБ 123.205.00.00 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

видання, Центр учбової літератури, 2020. 448 с.

20. Аварії, катастрофи та стихійні лиха: основні принципи і способи захисту населення. Реферат. URL: <https://osvita.ua/vnz/reports/dpju/24307/>. (дата звернення: 22.05.2025)

21. ДСТУ EN IEC 31010:2022. Керування ризиками – методи оцінювання ризику. – Введено в дію 31 грудня 2023 р. – Київ: ДП «УкрНДНЦ», 2023. – 120 с. – (Гармонізований з EN IEC 31010:2019 та IEC 31010:2019).

22. IEC 60038 | Європейські стандарти змінної електричної напруги. URL: <https://mirnyk.com/ua/2024/05/17/iec-60038yevropejskistandartyzminnoyiyelektrychnoyi-naprugy/>. (дата звернення: 25.05.2025)

23. Правила улаштування електроустановок. ПУЕ.

24. Правила безпечної експлуатації електроустановок споживачів: НПАОП 40.1-1.21-98. – Київ: Міністерство праці та соціальної політики України, 1998. – 150 с.

					КС КРБ 123.205.00.00 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“___” _____ 2025 р

КОМП'ЮТЕРНА СИСТЕМА МОНІТОРИНГУ ТА РЕГУЛЮВАННЯ
ТЕМПЕРАТУРИ ТЕПЛОНОСІЇВ У ГАЗОВИХ КОТЛАХ

ТЕХНІЧНЕ ЗАВДАННЯ

на 9 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-41

_____ к.т.н., доц. Яцишин В.В.

_____ Жук І-Д. І.

«___» _____ 2025 р.

«___» _____ 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система моніторингу та регулювання температури теплоносіїв у газових котлах ».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.205.00.00

1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Жук І-Д. І.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-53 від 27.01.2025 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.01.2025 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2025 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи – наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Комп'ютерна система що розробляється призначений для:

- Моніторинг та регулювання температури теплоносіїв.
- Контроль витоку газу.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка та налаштування комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах.

2.3 Характеристика об'єкту

Розробляювана система призначено для використання в домашніх котлах газового опалювання.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Комп'ютерна система повина складатись з:

- Мікроконтролер сімейства STM32, який забезпечує загальне керування функціонуванням системи;
- дисплей з шиною I2C для відображення параметрів, стану системи;
- сенсори для вимірювання температури та вмісту газу;
- UART модуль для передачі даних;
- Сервопривод для регулювання важелю;
- насос для циркуляції теплоносія;
- реле для комутації насосу для циркуляції .

3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Основна вимога, яка ставиться до способів та засобів інформаційного обміну – це використання провідних технологій.

3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два режими функціонування:

- нормальний режим функціонування;
- аварійний режим функціонування.

Основним режимом функціонування є нормальний режим.

Для забезпечення нормального режиму функціонування системи необхідно виконувати вимоги і дотримуватись умов експлуатації програмного забезпечення і комплексу технічних засобів системи, вказані у відповідних технічних документах (технічна документація, інструкції з експлуатації і т. д.).

Аварійний режим функціонування системи характеризується виявленням витoku газу системи опалення при якому дана система обезточується і надсилається повідомлення про аварійну ситуацію.

3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються інструменти діагностування основних процесів системи, які вмонтовані в операційну систему і програмне забезпечення, а також засоби для діагностики апаратного забезпечення.

Інструменти повинні забезпечувати зручний інтерфейс для можливості перегляду діагностичних подій, моніторингу процесу виконання програм.

3.1.5 Перспективи розвитку, проектування системи

Дана система може бути розширена завдяки використанню додаткових програмних компонентів, а також передбачено розширення способів передачі даних системою.

3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного і технічного забезпечення.

3.2.1 Вимоги до надійності

Система повинна бути захищена від фізичних чи механічних пошкоджень на рівні апаратного та програмного забезпечення. Надійність системи повинна забезпечувати відновлюваність функціонування у випадку збою апаратного чи програмного забезпечення. Імовірність безвідмовної роботи системи повинна складати не менше 99,6 %.

3.3 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику, а самі технічні засоби мати занулення або захисне заземлення .

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях в колах навантаження, а також знеструмлення при витоках газу.

Загальні вимоги пожежної безпеки повинні відповідати нормам на побутове електрообладнання. У разі пожежі не мають виділятися отруйні гази і дим. Після зняття електроживлення має бути доступне застосування будь-яких засобів пожежогасіння.

3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від +10°C до +35°C;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск 760 ± 25 мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів, датчики, контролери, системи передачі даних, пристрої безперебійного живлення.

На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів і прийматися заходи по їх ліквідації.

3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні не нижче встановленого вимогами, що пред'являються до категорії 1Д по класифікації документа, що діє, “Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем”.

Рівень захищеності від несанкціонованого доступу засобів обчислювальної техніки, що здійснюють обробку конфіденційної інформації, повинен відповідати вимогам класу захищеності згідно вимогам документу “Засоби обчислювальної техніки. Захист від несанкціонованого доступу до інформації. Показники захищеності від несанкціонованого доступу до інформації”.

3.4.1 Вимоги по збереженню інформації при аваріях

Інформація, при виникненні аварійних ситуацій повинна бути збережена на резервних носіях.

3.4.2 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки і зручності користування за умови комплектування високоякісним обладнанням (ЕОМ, монітор і інше обладнання), що має необхідні сертифікати відповідності і безпеки.

3.4.3 Вимоги до функцій (завдань), що виконуються системою:

- забезпечення зрозумілого інтерфейсу;
- забезпечення точності вимірів;
- регулювання подачі газу у котел;
- контроль живлення насосу для циркуляції теплоносія.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
 - а) Структурна схема.
 - б) Блок схема роботи комп'ютеризованої системи.
 - в) Схема електрична принципова.
 - г) Схема електрична монтажна (з'єднань).

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи
бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання.	27.01.2025р. – 02.02.2025р.
2	Аналіз вимог до комп'ютерної системи моніторингу та регулювання температури теплоносіїв у газових котлах	03.02.2025р. – 28.02.2025р.
3	Проектування системи моніторингу та регулювання температури теплоносіїв у газових котлах	01.03.2025р. – 30.03.2025р.
4	Алгоритму та програмного забезпечення системи моніторингу та регулювання температури теплоносіїв у газових котлах	01.04.2025р. – 30.04.2025р.
5	Безпека життєдіяльності, основи охорони праці	01.05.2025р. - 15.05.2025р.
6	Оформлення пояснювальної записки і графічного матеріалу	16.05.2025р. – 30.05.2025р.
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	9.06.2025р. – 15.06.2025р.
8	Попередній захист кваліфікаційної роботи бакалавра	16.06.2025р. – 22.06.2025р.
9	Захист кваліфікаційної роботи бакалавра	26.06.2025р.

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б
Перелік елементів

Перш. викорис.		Поз. позначення	Найменування					Кіл.	Примітка		
			Мікросхеми								
		U1	STM32F103C6T6					1			
		U2	SG90					1			
		U3	KY-013					1			
		U4	LCD1602					1			
		U5	MQ-2					1			
		U6	UART					1			
Довід. №		U7	I2C шина					1			
		U8	Реле					1			
		Підпис і дата									
Інв. № дубл.											
Зам. інв. №											
Підпис і дата											
Інв. № ориг.		Розроб.	Жук І-Д. І.				Комп'ютерна система моніторингу та регулювання температури теплоносіїв у газових котлах Перелік елементів	Літ	Аркуш	Аркушів	
		Перевір.	Яцишин В.В								
		Реценз.	Мудрик. І. Я.					ТНТУ, каф. КС, гр. СІ-41			
		Н. контр.	Луцик Н. С.								
		Зав. каф.	Осухівська Г. М.								

Додаток В

Код программного забезпечення

```
/* USER CODE BEGIN Header */  
/**  
  
*****  
*****  
 * @file           : main.c  
 * @brief          : Main program body  
  
*****  
*****  
 * @attention  
 *  
 * Copyright (c) 2025 STMicroelectronics.  
 * All rights reserved.  
 *  
 * This software is licensed under terms that can be found in the  
LICENSE file  
 * in the root directory of this software component.  
 * If no LICENSE file comes with this software, it is provided  
AS-IS.  
 *  
  
*****  
*****  
 */  
/* USER CODE END Header */  
/* Includes -----*/  
-----*/  
#include "main.h"  
#include "adc.h"  
#include "i2c.h"  
#include "rtc.h"  
#include "tim.h"  
#include "usart.h"  
#include "gpio.h">  
  
/* Private includes -----*/  
-----*/  
/* USER CODE BEGIN Includes */  
#include "string.h"  
#include "stm32f1xx_hal.h"  
#include "stm32f1xx_hal_i2c.h"  
#include <math.h>  
#include "stdio.h"  
#include "stdlib.h"  
#include <stdbool.h>  
void LSD_WRITE_I2C (uint8_t bt);  
void E_set (uint8 t data);
```

```

void E_reset (uint8_t data);
void RS_set(uint8_t data);
void RS_reset (uint8_t data);
void sendhalfbyte (uint8_t c, uint8_t mode);
void sendbyte (uint8_t c, uint8_t mode);
void LCD_SenChar (char ch);
void LCD_Clear (void);
void LCD_init (void);
void LCD_Set_Pos (uint8_t x, uint8_t y);
void LCD_String (char* st);
void LCD_Run(uint8_t krug, char word[]);
void LCD_Gradusi(uint16_t pwm);
uint64_t convHexToASCI (uint16_t value);

uint8_t data [1] = {0x01};

void LCD_Numb (int nummer);

void LCD_NumbwithDot (float nummer);
void TempMONITOR();
void kutpovorotu(int grad);
void send_data_UART(float temp, float gas, char *stan);
/* USER CODE END Includes */

/* Private typedef -----
-----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----
-----*/
/* USER CODE BEGIN PD */
uint32_t countLCD = 0;
/* USER CODE END PD */

/* Private macro -----
-----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----
-----*/

/* USER CODE BEGIN PV */
int adc_raw;

double temp;

double a;
double b;
double c;

```

```

float R = 0;
float gerilim = 0;
double temperature = 0;
double gas1 = 0;
double gas = 0;

/* USER CODE END PV */

/* Private function prototypes -----
-----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----
-----*/
/* USER CODE BEGIN 0 */
int myNum = 2024; // Integer (whole number)
float myFloatNum = 5.98; // Floating point number
char myLetter = 'D';
uint8_t myUint = 21; // Same way for uint16_t or uint32_t
uint8_t myArray[20] = {0};
char charData[100];
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{

    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----
    -----*/

    /* Reset of all peripherals, Initializes the Flash interface and
    the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

```

```

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_ADC1_Init();
MX_ADC2_Init();
MX_I2C1_Init();
MX_TIM2_Init();
MX_USART1_UART_Init();
MX_RTC_Init();
/* USER CODE BEGIN 2 */
HAL_ADC_Start(&hadc1);
HAL_ADC_Start(&hadc2);
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);

HAL_Delay(1000);
HAL_ADC_PollForConversion(&hadc1, 1000);
HAL_ADC_PollForConversion(&hadc2, 1000);

    LCD_init();
    countLCD = HAL_GetTick();
    LCD_Set_Pos(0,0);
    LCD_String(" -=KOTEL CTRL=- ");
    LCD_Set_Pos(0,1);
    LCD_String("    -=V1.0=- ");
    HAL_Delay(1000);
    LCD_Clear();

    char stan[15] = "Normal";

    HAL_UART_Transmit(&huart1, (unsigned char *)"System KOTEL CTRL
turn on\r\n", 27, 1000);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);

    LCD_Set_Pos(0,0);
    LCD_String("TEMP:");
    LCD_Set_Pos(9,0);
    LCD_String("GAS:");
    LCD_Set_Pos(0,1);
    LCD_String("STAN:");
    HAL_Delay(1000);

    a = 0.0004539418411;
    b = 0.0003617759715;
    c = -0.0000005190100184;

    // першочергове зчитування даних вмісту газу
    gas1 = HAL_ADC_GetValue(&hadc2);
    float voltage = (gas / 4095.0f) * 3.3f;
    float ppm = pow(10, ((voltage - 0.1f) * 2.7f)); // груба модель

```



```

    gas = ppm/4370000; // конвертування значень в шкалу до 100
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (gas < 70)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    // Зчитування значень АЦП
    adc_raw = HAL_ADC_GetValue(&hadc1);
    gas1 = HAL_ADC_GetValue(&hadc2);
    //Позрахунок значень датчика KY-013
    gerilim = (adc_raw * 3.3) / 4095;
    R = (33000/gerilim) - 10000;
    temperature = a + (b * (log(R))) + (c * (log(R)) *
(log(R)) * (log(R)));
    temp = 1 / temperature;
    temp = 320 - temp;
    //Позрахунок значень датчика MQ-2
    voltage = (gas / 4095.0f) * 3.3f;
    ppm = pow(10, ((voltage - 0.1f) * 2.7f)); // груба модель
    gas = ppm/4370000; // конвертування значень в шкалу до
100
    // вживання заходів щодо стану системи
    if(temp >= 85){
        //стан критичного перегріву
        kutpovorotu(0);
        strcpy(stan,"VERY HOT");
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,GPIO_PIN_SET);
    }else if(temp < 85 && temp >= 65){
        //стан перегріву
        kutpovorotu(20);
        strcpy(stan,"HOT");
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,GPIO_PIN_SET);
    }else if(temp < 65 && temp >= 50){
        //нормальна температура
        kutpovorotu(45);
        strcpy(stan,"NORM");
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,GPIO_PIN_RESET);
    }else if(temp < 50 && temp >= 25){
        //стан переохолодження
        kutpovorotu(60);
        strcpy(stan,"A BIT COLD");
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,GPIO_PIN_RESET);
    }else if(temp < 25 && temp >= 0){
        //стан сильного переохолодження
        kutpovorotu(75);
        strcpy(stan,"COLD");
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,GPIO_PIN_RESET);
    }else{
        //стан виявлення помилки в вимірах

```

```

        kotpovorotu(0);
        strcpy(stan, "ERROR TEMP");
    }
    //виведення інформації на дисплей
    LCD_Set_Pos(5,0);
    LCD_NumbwithDot(temp);
    LCD_Set_Pos(13,0);
    LCD_Numb(gas);
    LCD_Set_Pos(5,1);
    LCD_String(stan);
    //виведення інформації на ПК
    send_data_UART(temp, gas, stan);
    //затримка
    HAL_Delay(2400);

}
LCD_Clear();
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,GPIO_PIN_RESET);
//виведення на дисплей помилки
    LCD_Set_Pos(0,0);
    LCD_String("  -=AWARNING=-  ");
    LCD_Set_Pos(0,1);
    LCD_String("- _VUTIK GAZU_-");
    strcpy(stan, "!GAS ALERT!");
    //передача інформації про небезпечний стан на ПК
    sprintf(charData, "VUTIK GAZU, PROVITRIT PRIMISHENYA, CALL
104, GO EVERYONE OUTSIDE NOW %s\r\n", stan);
    HAL_UART_Transmit(&huart1,(uint8_t
*)charData,strlen(charData),1000);
    HAL_Delay(2000);
    //вимкнення системи
    HAL_PWR_EnterSTANDBYMode();
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /** Initializes the RCC Oscillators according to the specified
    parameters
    * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType =
RCC_OSCILLATORTYPE_LSI|RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;

```

```

RCC_OscInitStruct.HSISState = RCC_HSI_ON;
RCC_OscInitStruct.LSISState = RCC_LSI_ON;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}

/** Initializes the CPU, AHB and APB buses clocks
*/
RCC_ClkInitStruct.ClockType                                =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK

|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) !=
HAL_OK)
{
    Error_Handler();
}
PeriphClkInit.PeriphClockSelection                        =
RCC_PERIPHCLK_RTC|RCC_PERIPHCLK_ADC;
PeriphClkInit.RTCClockSelection = RCC_RTCCLKSOURCE_LSI;
PeriphClkInit.AdcClockSelection = RCC_ADCPCLK2_DIV6;
if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
{
    Error_Handler();
}
}

/* USER CODE BEGIN 4 */
void kutpovorotu(int grad){
    float j;
    if (grad > 180) grad = 180;
    j = (grad * 5.55555555556) + 250; // 5.55555555556 = 1*, from
250* to 1250 is from 0* to 180*
    __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_2, j);
}

void Delay_us (int time)//??? ?????? 8MHz (???? ???? ?????? ??
????????? ???? ??????????????)
{
    for (;time!=0;time--)
    {
        uint8_t d =0;
        for (;d!=72;d++) {};
    }
}

```

```

}

void LSD_WRITE_I2C (uint8_t bt)
{
data [0] = bt;
HAL_I2C_Master_Transmit(&hi2c1, (uint16_t) 0x8, data, 1,
30); //?????? ???? ? T 0x4e - 111
//HAL_I2C_Master_Transmit(&hi2c1, (uint16_t) 0x70, data, 1, 30);
//HAL_I2C_Master_Transmit(&hi2c1, (uint16_t) 0x72, data, 1, 30);
}

void send_data_UART(float temp, float gas, char *stan) {
char buffer[100];

//дата і час передачі

// Температура (1 знак після коми)
int temp_int = (int)(temp * 10);
sprintf(buffer, "Temperature: %d.%01d\r\n", temp_int / 10,
temp_int % 10);
HAL_UART_Transmit(&huart1, (uint8_t*)buffer, strlen(buffer),
1000);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
HAL_Delay(100);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
HAL_Delay(100);

// Газ (1 знак після коми)
int gas_int = (int)(gas * 10);
sprintf(buffer, "Gas value: %d.%01d\r\n", gas_int / 10,
gas_int % 10);
HAL_UART_Transmit(&huart1, (uint8_t*)buffer, strlen(buffer),
1000);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
HAL_Delay(100);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
HAL_Delay(100);

// Статус (char)
sprintf(buffer, "Status of System: %s\r\n", stan);
HAL_UART_Transmit(&huart1, (uint8_t*)buffer, strlen(buffer),
1000);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
HAL_Delay(100);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
HAL_Delay(100);
}

void E_set (uint8_t data)
{
LSD_WRITE_I2C (data|0x04);
}

```

```

void E_reset (uint8_t data)
{
    LSD_WRITE_I2C (data&~0x04);
}

void RS_set(uint8_t data)
{
    LSD_WRITE_I2C (data|0x01);
}

void RS_reset (uint8_t data)
{
    LSD_WRITE_I2C (data&~0x01);
}

void sendhalfbyte (uint8_t c, uint8_t mode)
{
    c<<=4;
    c |= 0x08;
    if (mode == 0)
    {
        E_set(c);//set line E
        // Delay_us();
        HAL_Delay(1);
        LSD_WRITE_I2C(c);
        E_reset(c);//set line E
        // Delay_us();
        HAL_Delay(1);
    }
    else
    {
        E_set(c|0x01);//set line E
        // Delay_us();
        HAL_Delay(1);
        LSD_WRITE_I2C(c|0x01);
        E_reset(c|0x01);//set line E
        // Delay_us();
        HAL_Delay(1);
    }
}

void sendbyte (uint8_t c, uint8_t mode)
{
    uint8_t hc =0;
    hc = c>>4;
    sendhalfbyte (hc, mode);
    Delay_us(10);
    //HAL_Delay(1);
    sendhalfbyte (c,mode);
    Delay_us(10);
    //HAL_Delay(1);
}

```

```

void LCD_SenChar (char ch)
{
    sendbyte (ch,1);
}

void LCD_Clear (void)
{
    sendbyte (0x01,0);
    HAL_Delay(10);
}

void LCD_init (void)
{
    HAL_Delay(100);
    sendhalfbyte(0x03,0);
    HAL_Delay(4);
    sendhalfbyte(0x03,0);
    HAL_Delay(100);
    sendhalfbyte(0x03,0);
    HAL_Delay(1);

    sendhalfbyte(0x02,0);
    HAL_Delay(1);

    sendbyte(0x28,0);
    HAL_Delay(1);
    sendbyte(0x28,0); //0x28
    HAL_Delay(1);

    sendbyte(0x0c,0); // 0x0c
    HAL_Delay(1);
    sendbyte(0x01,0);
    HAL_Delay(20);

    sendbyte(0x06,0);
    HAL_Delay(1);
    sendbyte(0x02,0);
    HAL_Delay(2);

    //sendbyte (0x0d, 0); //"d"
    //sendbyte (0x00, 1); //00000000
    //sendbyte (0x00, 1); //00000000
    //sendbyte (0x0a, 1); //00001010
    //sendbyte (0x1f, 1); //00011111
    //sendbyte (0x0e, 1); //00001110
    //sendbyte (0x03, 1); //00000100
    //sendbyte (0x00, 1); //00000000
    //sendbyte (0x00, 1); //00000000
}

void TempMONITOR() {

```

```

a = 0.0004539418411;
b = 0.0003617759715;
c = -0.0000005190100184;

adc_raw = HAL_ADC_GetValue(&hadc1);

gerilim = (adc_raw * 3.3) / 4095;
R = (33000/gerilim) - 10000;

temperature = a + (b * (log(R))) + (c * (log(R)) * (log(R)) *
(log(R)));
temp = 1 / temperature;
temp = 320 - temp;

LCD_Set_Pos(0,0);
    LCD_NumbwithDot(temp);

HAL_Delay(2000);
}

void LCD_Gradusi(uint16_t pwm){

}

void LCD_Set_Pos (uint8_t x, uint8_t y)
{
    switch (y)
    {
        case 0:
            sendbyte(x|0x80,0);
            HAL_Delay(1); //delay 1ms
            break;

        case 1:
            sendbyte((0x40+x)|0x80,0);
            HAL_Delay(1); //delay 1ms
            break;

        case 2:
            sendbyte((0x14+x)|0x80,0);
            HAL_Delay(1); //delay 1ms
            break;

        case 3:
            sendbyte((0x54+x)|0x80,0);
            HAL_Delay(1); //delay 1ms
            break;

        default:
            break;
    }
}

```

```

    }
}

void LCD_String (char* st)
{
    uint8_t i=0;
    while (st[i]!=0)
    {
        sendbyte(st[i],1);
        i++;
    }
}

void LCD_Run (uint8_t krug, char word[]) {
    uint8_t k = 16;
    krug *= k;
    uint8_t l = 0;
    uint8_t ik = 0;
    uint8_t jik =0;
    for (k = 0; k < krug + 1; k++ ){

        for(ik=0; ik < strlen(word); ik++){
            if(l > 15) l = 0;

            LCD_Set_Pos(l + ik, 0);
            LCD_SenChar(word[ik]);

            if(l + ik > 15){
                if(jik == strlen(word)) jik = 0;
                jik = l - 16 ;// - strlen(word);
                LCD_Set_Pos(jik + ik, 0);
                LCD_SenChar(word[ik]);
            }
            l++;
            HAL_Delay(300);
            LCD_Clear();
            if (l > 15) l = 0;
        }

    }

}

void LCD_Numb (int nummer)
{
    uint8_t nummer_100 = nummer/100;
    nummer_100 += 48;
    sendbyte(nummer_100,1);
    uint8_t nummer_10 = nummer % 100;
    nummer_10 = nummer_10 /10;
    nummer_10 += 48;
    sendbyte(nummer_10,1);
}

```



```

    uint8_t nummer_1 = nummer % 10;
    nummer_1 += 48;
    sendbyte(nummer_1,1);

}

void LCD_NumbwithDot (float nummer)
{
    nummer = nummer * 10;
    int num = nummer;
    uint8_t nummer_100 = num/100;
    nummer_100 += 48;
    sendbyte(nummer_100,1);
    uint8_t nummer_10 = num % 100;
    nummer_10 = nummer_10 /10;
    nummer_10 += 48;
    sendbyte(nummer_10,1);
    sendbyte(0x2E,1);
    //LCD_SenChar('.');
    uint8_t nummer_1 = num % 10;
    nummer_1 += 48;
    sendbyte(nummer_1,1);

}

uint64_t convHexToASCI (uint16_t value)
{
    uint32_t tmpValue = 0;
    tmpValue = value;
    uint64_t tmp = 0;
    uint64_t rezult = 0;

    tmp = value / 10000;

    if(tmpValue > 9999)
        rezult = tmp | 0x30;
    else
        rezult = 0x20;

    value = value % 10000;
    tmp = value / 1000;
    rezult = rezult << 8;
    if(tmpValue > 999)
        rezult |= tmp | 0x30;
    else
        rezult |= 0x20;

    value = value % 1000;
    tmp = value / 100;

```

```

        result = result << 8;
        if(tmpValue > 99)
            result |= tmp | 0x30;
        else
            result |= 0x20;

        value = value % 100;
        tmp = value / 10;
        result = result << 8;
        if(tmpValue > 9)
            result |= tmp | 0x30;
        else
            result |= 0x20;

        tmp = value % 10;
        result = result << 8;
        result |= tmp | 0x30;

return result;
}
/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error
return state */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source
line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name
and line number,
ex: printf("Wrong parameters value: file %s on line %d\r\n",

```

```
file, line) */  
    /* USER CODE END 6 */  
}  
#endif /* USE_FULL_ASSERT */
```