

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних систем та технологій

(повна назва кафедри (предметної, циклової комісії))

Дипломна робота

на здобуття ступеня вищої освіти «бакалавр»

(освітньо-кваліфікаційний рівень)

на тему Розробка апаратно-програмного комплексу системи автоматизації

(назва українською)

Development of automation hardware and software

(назва англійською)

Виконав: студент заочної форми навчання
спеціальності 123 Комп'ютерна інженерія

(шифр і назва напрямку підготовки, спеціальності)

Ільїних Анна Анатоліївна

(прізвище, ім'я, по-батькові)

Керівник

ст. викл. Берков Ю.М.

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

Рецензент

к.т.н., доцент Левченко А.О.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ від « » 2021 р.

Завідувач кафедри

Ю.О. Гунченко

(підпис)

(прізвище, ініціали)

Захищено на засіданні ЕК №

протокол № від « » 2021р.

Оцінка / /

(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

Н.Ф. Казакова

(підпис)

(прізвище, ініціали)

Одеса - 2021

АНОТАЦІЯ

Темою даної дипломної роботи є «Розробка апаратно-програмного комплексу системи автоматизації».

У якості процесу, що автоматизується, було обрано управління твердопаливним котлом. Мета роботи полягає у розробці такої автоматичної системи управління, що матиме необхідний функціонал для безпечного, комфортного та економного користування твердопаливним котлом. Для досягнення поставленої мети було:

- досліджено особливості використання твердопаливних котлів та пов'язаних з ними потреб користувачів;
- проаналізовано представлені на ринку прилади;
- спроектовано автоматичну систему управління котлом;
- розроблено прототип апаратно-програмного рішення поставленої задачі.

Основою створеного пристрою є мікроконтролер Arduino Uno, до якого під'єднано дисплей LCD1602 Keypad SHIELD, на який виводяться поточні результати вимірювання з датчиків, діючий режим роботи котла та поточний кут повороту шибери. За допомогою меню користувач може налаштувати бажаний режим роботи котла. Для попередження користувача у разі аварійної ситуації використовуються світлові та звукові сигнали.

Структурно робота складається з вступу, трьох розділів, висновків та списку використаної літератури. У додатках представлено код створеної програми.

Ключові слова: автоматика, мікроконтролери AVR, Arduino, твердопаливний котел, температура теплоносія, димові гази, режим горіння.

ANNOTATION

The topic of the submitted thesis is Development of automation hardware and software. The objective of the work is to develop an automation system for solid fuel boiler, the provided system having all the functionality required to use the boiler safely, comfortably, and economically. The adverted objective requires fulfilling the following tasks:

- researching specifics of solid fuel boilers operation and related usage demands;
- analysing appliances present on the market;
- designing the automation system for solid fuel boiler management;
- developing the prototype of automation hardware and software solution to the task.

Arduino UNO microcontroller is used as the base of the created appliance. It is connected to the LCD1602 Keypad SHIELD showing current sensors data (i.e. water and flue gas temperatures and CO level), boiler operation mode, and the rotation angle of the flue damper. Display menu is used to choose the boiler operation mode. In case of an emergency, alarm sound and LED signal are activated to warn user about danger.

Structurally the paper consists of the introduction, three chapters, conclusion, bibliography. Appendix shows the created software code.

Keywords: automation, AVR microcontrollers, Arduino, solid fuel boiler, heat transfer fluid temperature, flue gas, combustion mode.

АННОТАЦИЯ

Темой данной дипломной работы является «Разработка аппаратно-программного комплекса системы автоматизации».

В качестве процесса, который автоматизируется, было выбрано управление твердотопливным котлом. Цель работы заключается в разработке такой автоматической системы управления, которая будет обладать всем необходимым функционалом для безопасного, комфортного и экономного пользования твердотопливным котлом. Для достижения поставленной цели нами были:

- исследованы особенности использования твердотопливных котлов и связанные с ними потребности пользователей;
- проанализированы представленные на рынке приборы;
- спроектирована автоматическую систему управления котлом;
- разработан прототип аппаратно-программного решения поставленной задачи.

Основой созданного устройства является микроконтроллер Arduino Uno, к которому подключен дисплей LCD1602 Keypad SHIELD, на который выводятся текущие результаты измерения с датчиков, действующий режим работы котла и текущий угол поворота шибера. С помощью меню пользователь может настроить желаемый режим работы котла. Для предупреждения пользователя в случае аварийной ситуации используются световые и звуковые сигналы.

Структурно работа состоит из введения, трех глав, выводов и списка использованной литературы. В приложениях представлен код созданной программы.

Ключевые слова: автоматика, микроконтроллеры AVR, Arduino, твердотопливный котел, температура теплоносителя, дымовые газы, режим горения.

ЗМІСТ

ВСТУП.....	6
1 СПОСОБИ КОРИСТУВАННЯ ТВЕРДОПАЛИВНИМ КОТЛОМ	8
1.1 Способи опалення приватного будинку	8
1.2 Використання твердопаливного котла для опалення	10
1.3 Огляд наявних апаратно-програмних рішень	13
1.4 Деталізація поставленої задачі	16
2 ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ	18
2.1 Вибір необхідного апаратного забезпечення	18
2.2 Програмне забезпечення, необхідне для виконання задачі	29
3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	32
3.1 Опис апаратної частини	32
3.2 Алгоритм роботи програми	38
ВИСНОВКИ	45
СПИСОК ЛІТЕРАТУРИ	46
ДОДАТОК А	49

ВСТУП

У сучасному світі тенденція до автоматизації поширюється щороку швидше. Побутова автоматизація уже стала досить звичною для багатьох людей, однак її створення та встановлення для певних приладів досі може бути окремою задачею для користувача, тобто автоматика не поставляється «з коробки». Одним із прикладів таких приладів є твердопаливний котел, що є популярним рішенням для опалення приватного будинку в Україні.

Деякі постачальники котлів пропонують купити додатково автоматику для управління ним. У разі ж її відсутності, або якщо постачальник не пропонує такий додатковий товар, управління даним приладом покладається на ручну регуляцію сили тяги чи доступу кисню до камери згорання.

Пристрій автоматизації управління котлом найчастіше всього оснащений рядом датчиків (пр. температури води, що нагрівається котлом для подальшого постачання у систему опалення, температура димових газів тощо), реле для запуску насосу, управління вентилятором для швидшого розпалу. Додаткові системи сповіщення у разі перевищення температури або системи, що дозволяють скоротити витрати палива завдяки більш відповідальному використанню, рідше зустрічаються, а автоматика, оснащена подібними системами, має більшу вартість. Також слід зазначити, що через різну конфігурацію котлів покупна автоматика не завжди чітко підходить до певної моделі та все одно потребує додаткового налаштування та заміни датчиків.

Ціна власне елементів/модулів автоматики є невисокою. Тому створення власної автоматики є водночас більш економним методом, а також дозволить управляти приладом більш точно, враховуючи бажані можливості та особливості конфігурації конкретної моделі котла.

Таким чином, актуальність даної роботи зумовлена затребуваністю на ринку автоматичних систем управління твердопаливним котлом.

Мета роботи полягає у розробці такої автоматичної системи управління, що матиме необхідний функціонал для безпечного, комфортного та економного користування твердопаливним котлом.

Досягнення поставленої мети потребуватиме виконання наступного ряду завдань:

- дослідження твердопаливних котлів, особливостей їх використання та пов'язаних з ними потреб користувачів;
- аналіз представлених на ринку приладів;
- проектування автоматичної системи управління котлом;
- розробка апаратно-програмного рішення поставленої задачі.

1 СПОСОБИ КОРИСТУВАННЯ ТВЕРДОПАЛИВНИМ КОТЛОМ

Поширення та популяризація технологій разом зі зменшенням їх вартості, а також вартості встановлення та підтримки призвело до збільшення попиту на системи автоматизації. Перш за все ми маємо на увазі автоматизацію управління приватним будинком, що часто називають просто «розумним будинком». Такий будинок може мати різні особливості, визначені власником. Як видно з рис. 1.1, у даному випадку управління опаленням розглядається як одна з систем управління водою.

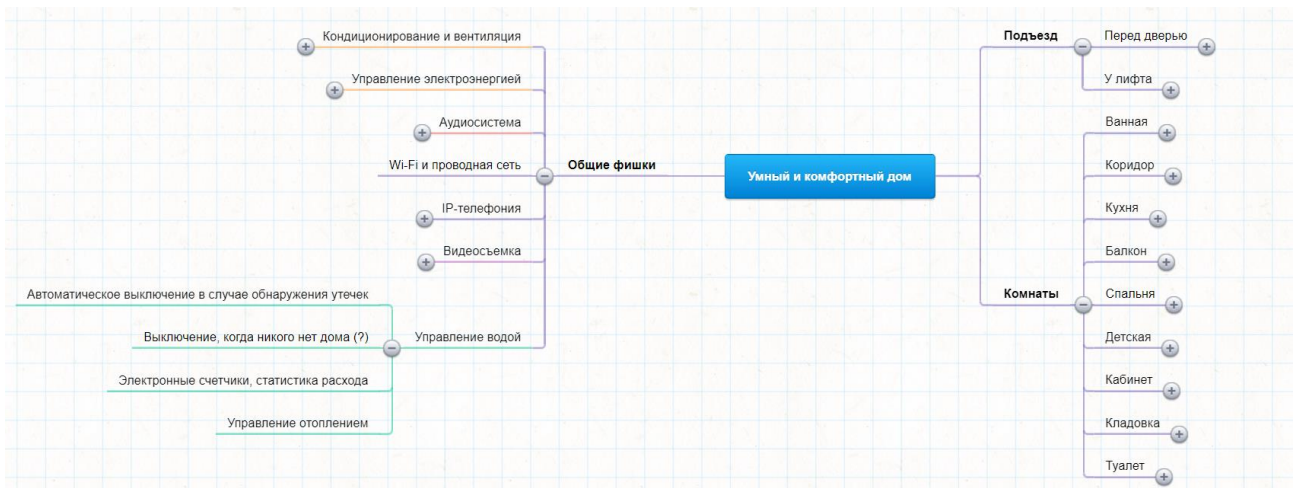


Рисунок 1.1 – Схема автоматизації приватного будинку [3]

1.1 Способи опалення приватного будинку

У даному розділі ми розглянемо способи опалення приватного будинку. У сучасному світі існує безліч таких способів, однак вибір оптимального є важким завданням. Перш за все, потрібно визначити, яка енергія буде обігрівати будинок. Останнім часом все більшу популярність здобувають сонячні батареї, однак використання даного способу досить обмежене: територія, де знаходиться будинок, має бути досить сонячною (приблизно 20 сонячних днів на місяць) [4], також важливо розмістити самі батареї так, аби на них попадало найбільше променів, враховуючи

площу будинка, що потрібно опалювати, та яку кількість енергії батареї зможуть постачати. Звичайно, їх потрібно обслуговувати. Зважаючи на всі ці фактори, в Україні сонячні батареї частіше використовуються як додаткове джерело електроенергії.

Теплова помпа (насос) – ще один екологічний та економний спосіб опалення будинку. Існує декілька типів насосів, і вибір конкретного спирається на особливості зовнішньої території, законодавства та, звичайно, будівлі, що опалюється. Потрібно також зазначити, що для опалення за допомогою теплової помпи слід використовувати теплі поли/підлоги, а не звичні радіатори. Наразі поширенню даного способу опалення заважає висока вартість теплових насосів [5].

Частіше за все в Україні для опалення використовуються газові, електричні або твердопаливні котли. Вони відрізняються не тільки типом палива, а й мають кожні свої особливості встановлення та потрібної додаткової апаратури. Наприклад, для твердопаливного котла найчастіше виділяють окрему кімнатку, оскільки він громіздкий, і до того ж під час роботи утворюються відходи палива (попіл, сажа). Однак такі котли використовують досить часто, оскільки на відміну від електричного та газового, вони не залежать від мережі, і (в залежності від площі, що опалюється, та доступності) саме паливо (дрова, пелети, вугілля) дешевше, ніж при використанні електричного/газового котла [6, 7].

На рис. 1.2 представлено статистику щодо вартості опалення будинку площею у 200 м² під час року в залежності від вибору опалювальної системи (ТН – тепловий насос) [6]. Як видно з діаграми, опалення з використанням теплової помпи вважається найдешевшим, однак слід зазначити, що сама система такого опалення є найдорожчою. До того ж, за бажання використовувати такий метод опалення, слід з самого початку проектувати будинок з його урахуванням.

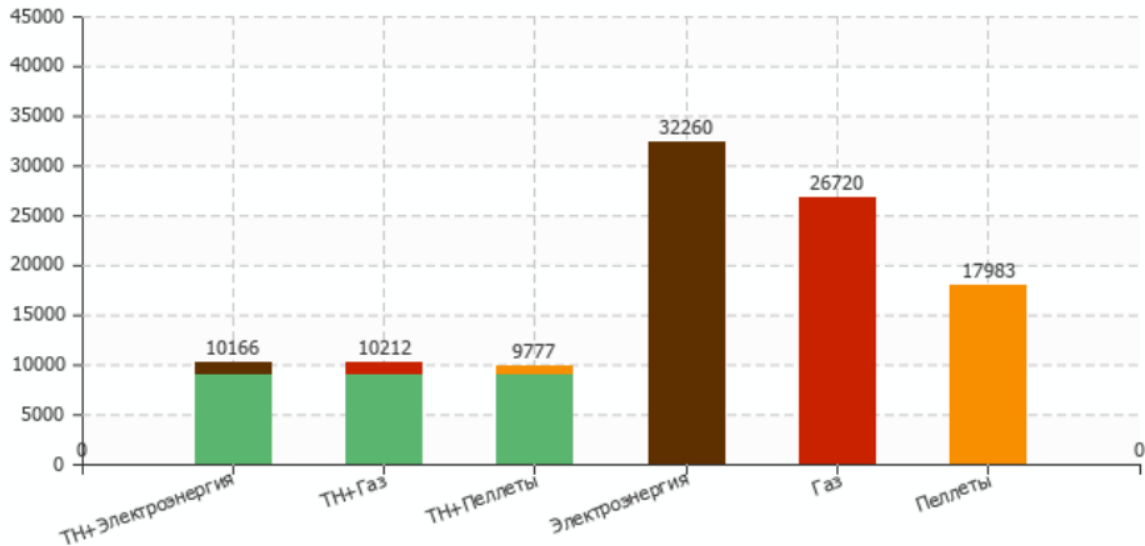


Рисунок 1.2 – Порівняння річних витрат на опалення в залежності від типу опалювальної системи

Серед розглянутих котлів найдешевшим є пелетний твердопаливний котел. Є декілька видів твердопаливних котлів, і вартість їх використання буде також залежати від виду палива. Наприклад, пелетний котел можна топити лише пелетами, в той час як універсальний твердопаливний котел може використовувати у якості палива як пелети, так і дрова, вугілля тощо. У даній роботі ми розглянемо опалення за допомогою твердопаливного котла, а точніше, систему автоматичного управління таким котлом, що можна використовувати для будь-якого з зазначених типів твердопаливних котлів.

1.2 Використання твердопаливного котла для опалення

Слід зазначити, що твердопаливні котли в свою чергу поділяються на пелетні, дров'яні та універсальні [8]. Також котли можуть мати різну конструкцію (одноходові, триходові, з використанням насоса тощо).

На рис. 1.3 показано приклад схеми твердопаливного котла виробника KRONAS. Хоча у складі будь-якого котла є камера згоряння,

шибер, водяна оболонка або окрема водяна камера, конкретна схема котла іншого виробника може суттєво відрізнятися від представленої.



Рисунок 1.3 – Схема твердопаливного котла KRONAS

Триходовий клапан використовується для перемикання напрямку потоку гарячого повітря. Завдяки цьому довше зберігається циркуляція жара у котлі, тож потрібно менше палива для підтримання бажаної температури. Однак у такому разі процес розпалу може бути сповільнений; до того ж, за низької температури котла утворюється конденсат, що може шкідливо позначитися на довговічності та безпечності конструкції. Автоматика може використовуватися для перемикання одно- та триходового режиму котла як зазначено на рис 1.4 [9, 10].

Автоматика використовується для регуляції процесу опалення, що включає в себе регуляцію постачання повітря для горіння шляхом повороту заслонки, за наявності додаткових систем – насосів, вентиляторів, – управління ними. Також автоматика може подати сигнал про несправність обладнання або застережити, якщо температура котла перевищить вказане значення. Таким чином, автоматика забезпечує додатковий рівень комфорту (регуляція температурі для уникнення різких перепадів) та безпеки, а також дозволяє знизити витрати на паливо завдяки більш розумному (відповідальному) споживанню.

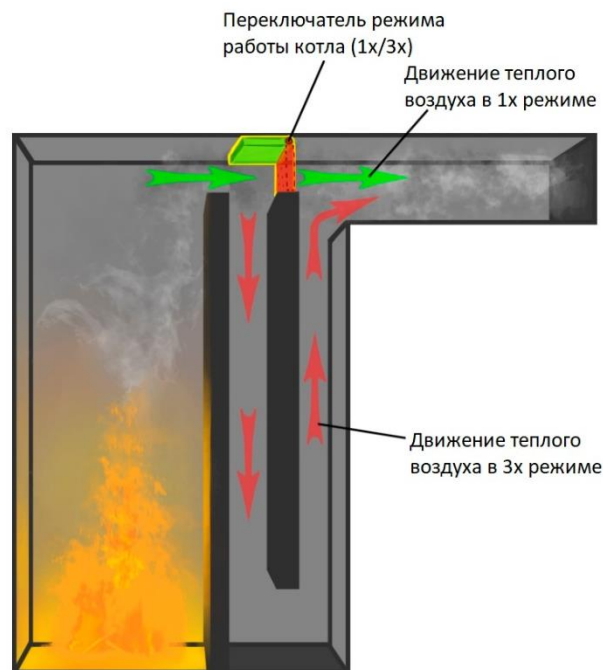


Рисунок 1.4 – Одно-/триходовий режими роботи котла

У регуляції роботи котла також можна використовувати кімнатний термостат [11]. Такий пристрій встановлюється безпосередньо у приміщенні, що опалюється. Він містить датчик температури повітря у приміщенні, і, спираючись на дані поточної температури та вибраної споживачем температури, що має підтримуватися, передає дані щодо необхідності збільшити (зменшити, підтримувати поточний) рівень

опалення. Встановлення чи зміна «оптимальної температури» не потребує безпосередньої роботи з автоматикою котла, адже для цього використовуються налаштування термостата. За такої конфігурації, саме термостат встановлює «вимоги» щодо температури, яку потрібно підтримувати за допомогою твердопаливного котла, а автоматика котла керує пристроями (заслонка, вентилятор, насос), за допомогою яких встановлена температура підтримується.

1.3 Огляд наявних апаратно-програмних рішень

Наразі на ринку представлений широкий вибір автоматики. Вони відрізняються, перш за все, функціоналом (набір підтримуваних пристроїв, можлива точність регулювання, додаткові особливості, наприклад, віддалене управління по Wi-Fi), матеріалами, комплектацією тощо. Також при виборі автоматики слід звернути увагу на особливості конфігурації певного котла, адже на ринку представлено велике розмаїття можливих конфігурацій.

На рис. 1.5 показано пристрій Atos TAL RT22 за виробництвом компанії Kom-Ster [12].



Рисунок 1.5 – Управляющий блок Atos TAL RT22

Даний пристрій має базовий функціонал. Автоматика встановлюється на стіні біля котлу та підключається до електромережі. Датчик температури встановлюється на трубу гарячої води з котла. До автоматики підключається насос (використовується за необхідності швидкого зменшення температури теплоносія) та вентиляційний пристрій (для збільшення потоку повітря для швидшого розпалу палива). Спираючись на дані температури води, що подається з котлу, регулюється рівень горіння палива. Дані представлені на екрані. Також пристрій подає звуковий та LED-сигнал у разі критичного перевищення температури.

До пристрою можна підключити датчик димових газів для додаткового контролю власне процесу горіння палива у котлі. Такий датчик потрібно буде придбати окремо, він не постачається у комплекті автоматики.

CS-20 (Виробник KG Elektronik) зображено на рис. 1.6[13].



Рисунок 1.6 – Управляючий блок автоматики CS-20

Даний пристрій за своїм функціоналом схожий до описаного вище пристрою Atos. На жаль, у інструкції не зазначено, чи можна під'єднати додатковий датчик димових газів до даної моделі. Однак перевагою CS-20

є підтримка декількох режимів, що встановлюються в залежності від виду палива, на якому працює котел.

На рис. 1.7 зображено автоматику Tech ST 81 zPID за виробництвом Tech Sterowniki [14]. Дана модель має більш широкий функціонал, ніж представлені вище. До неї приєднуються декілька датчиків:

- температури теплоносія;
- температури води у резервуарі;
- температури димових газів.



Рисунок 1.7 – Управляючий блок автоматики ST 81 zPID

Окрім насоса примусової циркуляції води, до даного пристрою можна приєднати насос ГВС для бойлеру. Також представлена автоматика має декілька програм для управління котлом та забезпечення його безперебійної роботи навіть у літній час, і підтримує підключення кімнатного термостату.

На рис. 1.8 зображено термостат Pilot R, виробник – Prond [15].

Даний пристрій має два режими роботи:

- контроль температури котла або опалювального контуру, в залежності від встановлення та конфігурації пристрою;
- контроль температури приміщення, де пристрій встановлено.

Екран пристрою виводить інформацію щодо температури, отриманої встановленим датчиком, стану вентилятора, насоса циркуляції та насосу ГВС.



Рисунок 1.8 – Термостат PILOT R

Доцільність використання кімнатного термостату слід визначати з огляду на умови роботи та вподобання користувача. Оскільки робота твердопаливного котлу потребує досить частого ручного заповнення котла паливом, винесення керуючого пристрою для регуляції температури з конкретного приміщення не має таких яскравих переваг, як при використанні інших котлів.

1.4 Деталізація поставленої задачі

У даній роботі ми маємо спроектувати та виготовити прототип системи управління твердопаливним котлом (автоматику). Використання власного контролера дозволяє деталізувати програму та спроектувати пристрій конкретно під модель котла, що використовується, з меншими витратами ніж спеціалізовані девайси виробника (за умови їх наявності).

Представлений пристрій повинен мати наступний функціонал:

- датчик температури, під'єднаний до труби подачі гарячої води у систему опалення з котла є основним джерелом даних для системи автоматизації;
- подача повітря регулюється шляхом повороту замикаючої заслонки (шибера), таким чином корегуючи температуру теплоносія;
- регулятор режиму (одно-/триходовий) використовується для економії палива;
- регулятор роботи циркуляційного насосу використовується для швидшої подачі гарячої води у систему опалення або охолодження води, нагрітої котлом;
- дані датчика температури димових (вихідних) газів використовується для додаткового рівня безпеки (захисту від пожегу);
- подається звуковий сигнал при критичному підвищенні температури води та/або димових газів;
- датчик чадного газу подає попереджуючий сигнал у разі перевищення концентрації чадного газу у приміщенні;
- інформація щодо поточного стану виводиться на екран;
- керувати налаштуваннями можна використовуючи меню на екрані пристрою.

Оскільки ми не маємо можливості протестувати пристрій на конкретному котлі, тестування проводитимуться у домашніх умовах та використані налаштування максимальної/мінімальної/оптимальної температур відрізнятимуться від таких, що використовуються при безпосередній роботі пристрою.

2 ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ

2.1 Вибір необхідного апаратного забезпечення

Для виконання поставленого завдання нам перш за все потрібно вибрати мікроконтролер, що буде «серцем» нашої автоматики. Розглянемо декілька доступних мікроконтролерів:

STM32 – серія мікроконтролерів, основних на 32-бітних ядрах ARM Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+ або Cortex-M зі скороченим набором інструкцій. STM32 зображено на рис. 2.1 [16].

Особливості мікроконтролерів STM32:

- низька вартість;
- широкий вибір середовищ розробки;
- чіпи можна замінити (якщо недостатньо ресурсів одного мікроконтролеру, його можна замінити, не заміняючи схему/плату);
- зручне налагодження;
- висока продуктивність;
- високий поріг входження;
- небагато літератури по STM32;
- наявні бібліотеки застаріли, потрібно створювати власні.



Рис 2.1 – Мікроконтролер EASY-STM32

Arduino – це апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++ [5].

До основних переваг даної платформи відносяться [17]:

- низький поріг входу;
- доступність великої кількості модулів, сенсорів тощо, за невисокою ціною;
- проста мова програмування;
- вбудований програматор та велика кількість бібліотек для використання додатків (модулів, дисплеїв тощо);
- можливість автономної роботи;
- наявність різноманітних плат розширення (shields).

На рис. 2.2 зображено детальну схему плати Arduino UNO [18].

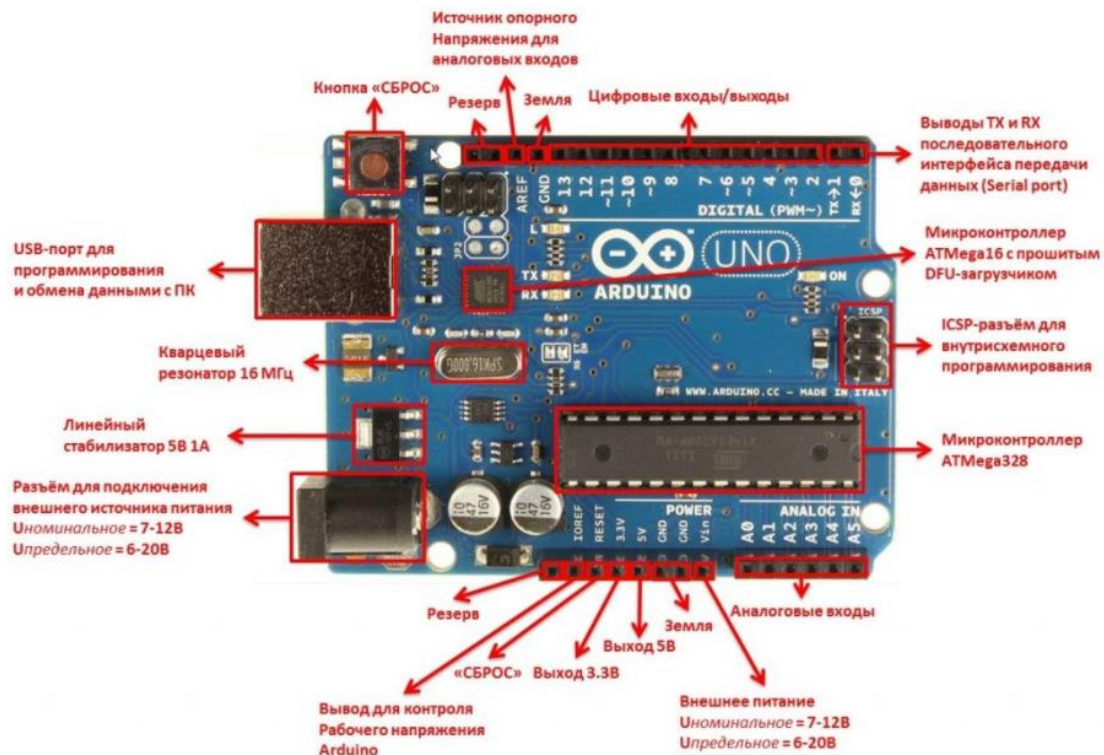


Рисунок 2.2 – Структура плати Arduino Uno

Raspberry Pi – одноплатний комп'ютер, розроблений британським фондом Raspberry Pi Foundation. Raspberry Pi побудований на системі-на-чипі (SoC) Broadcom BCM2835, яка включає в себе процесор ARM із тактовою частотою 700 МГц, графічний процесор VideoCore IV, і 512 чи 256 мегабайтів оперативної пам'яті. Твердий диск відсутній, натомість використовується SD карта. Схему Raspberry Pi показано на рис. 2.3 [18].

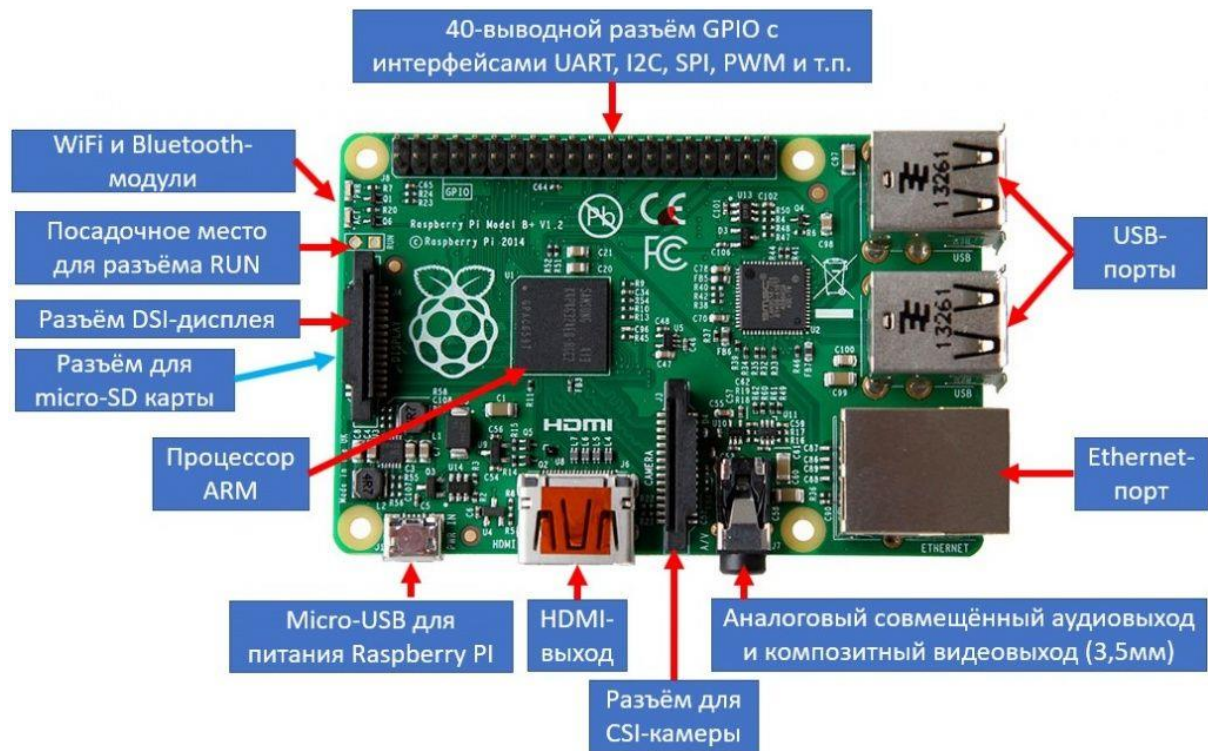


Рисунок 2.3 – Структура мікрокомп'ютера Raspberry Pi B+

Raspberry Pi має значно більший функціонал, аніж розглянуті вище мікроконтролери, однак для нашого завдання немає необхідності у використанні міні-комп'ютера з підтримкою відео через HDMI та встановлення ОС. Пристрій, що розроблюється, має бути недорогим та функціональним, а також для нас важлива можливість живлення автоматики з батареї у разі відключення централізованого електропостачання. Для нашої роботи ми обрали плату Arduino Uno, що є доступною та задовольняє вимогам щодо доступного функціоналу. До

того ж, у мережі інтернет наявна велика кількість матеріалів щодо роботи з Arduino та бібліотек для різноманітних модулів.

Зважаючи на кількість датчиків та систем, якими управлятиме розроблена автоматика, нам потрібно буде використати додаткові плати розширення. Розглянемо детальніше, які модулі потрібно додати до системи:

Датчик температури води у бойлері

Оскільки кінцева мета використання автоматики, як і твердопаливного котла в цілому – обігрів певного приміщення, підтримання заданої температури теплоносія є основною задачею апаратно-програмного забезпечення, що розроблюється. [19] У разі якщо температура буде зависокою, необхідно перекрити подачу кисню аби уникнути аварії.

Ми використовуємо датчик DS18B20 у водонепроникному корпусі з пило-/вологозахистом IP67, що зображено на рис. 2.4.



Рисунок 2.4 – DS18B20 у водонепроникному корпусі

Характеристики:

- робоча напруга: від 3В до 5.5В;
- точність: $\pm 0.5^{\circ}\text{C}$ у діапазоні -10°C до $+85^{\circ}\text{C}$;

- робочий діапазон температур: від -55°C до 125°C ;
- інтерфейс: 1-Wire;
- споживчий струм: 1mA.

Для роботи з датчиком потрібна бібліотека OneWire.h. Для підключення потрібно використати резистор 4.7 кОм.

Датчик температури димових газів

Важливо також підтримувати оптимальну температуру власне у котлі. Датчик димових газів дозволяє вимірювати температуру газів, що виходить через димохід. Якщо температура у котлі занадто низька, утворюватиметься сажа, а вихід шкідливих продуктів розпаду палива збільшується. Занадто висока температура позначатиме підвищену витрату палива. Критичне підвищення температури котла може бути небезпечним, тож у такому випадку слід запустити сигналізацію, а автоматика має закрити доступ кисню до котла, аби найшвидше погасити його.

Оскільки температура газу значно перевищує температуру води у бойлері (температура димових газів не має перевищувати 600°C), звичайний температурний датчик не можна використовувати у даному проекті. У нашому проекті використовуватиметься термопара К-типу з цифровим підсилювачем на мікросхемі MAX6675, яку зображено на рис. 2.5.

Характеристики:

- робоча напруга: від 3В до 5.5В;
- мікросхема перетворювача: MAX6675;
- тип термопари: К-тип;
- компенсація холодного спаю: інтегрована;
- інтерфейс: SPI;
- розрядність перетворювача: 12 біт;

- діапазон вимірюваної температури датчика: від 0°C до 600°C;
- температурна роздільна здатність: 0.25°C;
- діапазон температури компенсації холодного спаю: -20°C ~ +80°C;
- контроль обриву термопари: інтегрований;
- довжина кабелю термопари 50 см;
- різьба для встановлення датчика: M5x1 мм.

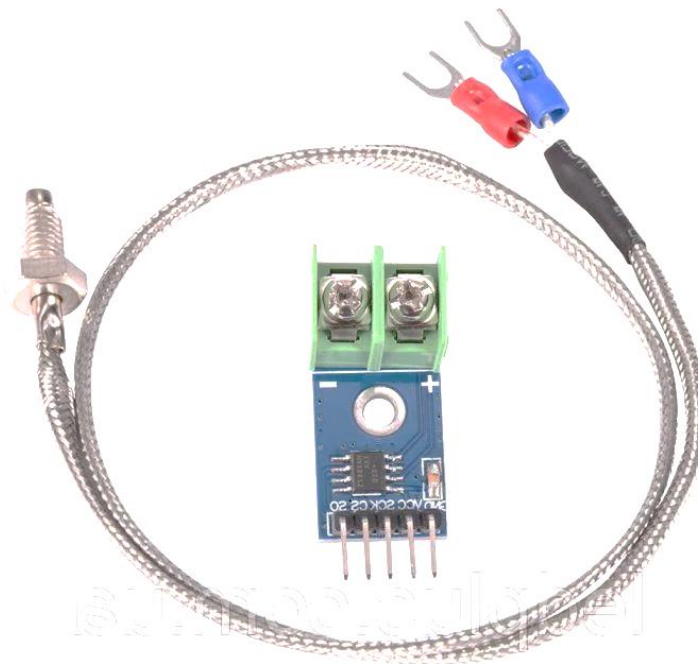


Рисунок 2.5 – Термопара К-типу з підсилювачем MAX6675 [20]

Датчик CO

Під час роботи твердопаливного котла неминуче утворюється певний об'єм чадного газу. Це безбарвний, дуже отруйний газ без запаху. Ми вирішили додати датчик CO у кімнаті, де розташований пристрій, аби завчасно попередити користувача, якщо рівень чадного газу буде критично перевищено.

Для контролю рівню CO ми використовуватимемо датчик MQ7, який зображено на рис. 2.6 [18]. Його характеристики:

- робоча напруга: 5 В;
- споживчий струм: 160 мА;
- діапазон чутливості: 10-10000 ppm (parts per million – часток на мільйон, 1 ppm = 0,0001%. Нормальним рівнем чадного газу є 0,0017% – 170 ppm);
- час накалиу нагрівача: 60 – 90 сек.;
- опір нагрівача: 31 Ом;
- опір датчика: 2-20 кОм.



Рисунок 2.6 – Датчик MQ7

Світлодіоди, динамік для подачі звукового сигналу

Дані з зазначених вище датчиків допоможуть застережити користувача. Явним сигналом несправності буде звуковий сигнал та вмикання світлодіоду. Для подачі звукового сигналу використовуємо п'єзовипромінювач [21].

Характеристики:

- робоча напруга: 4-6 В;
- максимальне споживання струму: 60 мА;

- максимальна сила шуму: 85 дБ;
- опір котушки: 40 Ом;
- частота звукового сигналу: 2731 Гц.



Рисунок 2.7 – П'єзовипромінювач (джерело шумового сигналу)

На рис. 2.8 зображено світлодіод [22]. У нашій роботі використовується аналогічний зображеному світлодіод червоного кольору.

Його характеристики:

- сила світла: 600-800 МКб;
- робочий струм: 20 мА;
- робоча напруга: 1,9-2,1 В.

Для під'єднання необхідно використати резистор (300 Ом-2 кОм).

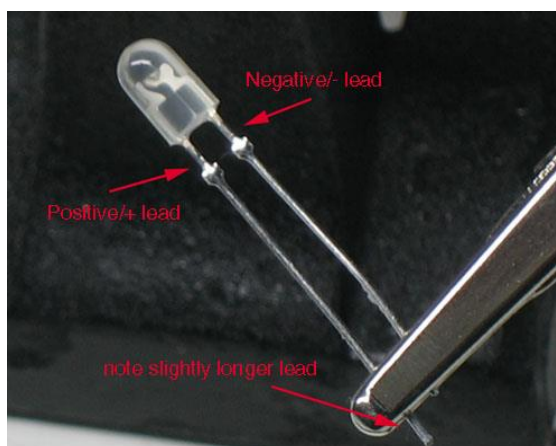


Рисунок 2.8 – Налаштування світлодіоду

Серво-приводи

Для даного проекту потребується серво-привід, що регулюватиме поворот замикаючої заслонки (шибера) для керування подачею кисню до котла. Також серво можна використати для перемикання одно- та трифазового режиму регуляції газів під час роботи котла.

Характеристики:

- робоча напруга: 3,5-5 В;
- споживчий струм під час руху: 50-80 мА;
- споживчий струм для утримання повороту: 5-10 мА;
- кут повороту: 180°;
- обертаючий момент: 2 кг/см.

У даній роботі використовуються два серво-мотори SG90 (див. рис. 2.9). Однак ми створюємо лише прототип автоматики твердопаливного котла. Звичайно, у реальному приладі нам потрібно було б використати серво, що матиме більшу силу (обертаючий момент), адже серво керуватиме важкою заслонкою/перемикачем режиму. Слід також зазначити що більш потужний мотор може потребувати додаткового живлення.



Рисунок 2.9 – Серво-привід SG90

Для роботи з серво-мотором використовуватимемо стандартну бібліотеку Servo.h.

Реле

Нам знадобиться реле для управління циркуляційним насосом. Ми використовуємо двоканальне реле (рис. 2.10) для задання швидкості роботи насосу:

- 00 – насос вимкнено;
- 01 – насос працює на першій швидкості;
- 11 – насос працює на максимальній швидкості.

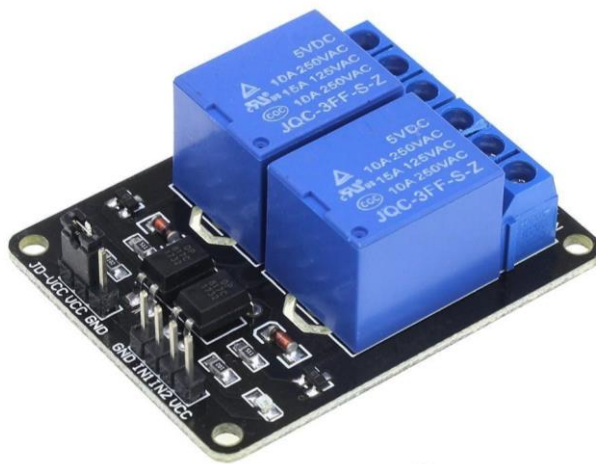


Рисунок 2.10 – Двоканальне реле

У майбутньому аналогічно також можна підключити другий насос (у разі підключення буферної ємності) або вентилятор для швидшого розпалу котла.

Характеристики:

- робоча напруга: 5 В;
- споживчий струм: 15-20 мА;
- каналів керування: 2;
- тип управління: цифровий;
- комутований струм: max 10 А;
- комутована напруга: max 250 V.

При підключенні реле нам знадобиться використовувати окреме

джерело струму для насосу. У разі відсутності електроенергії насос не працюватиме.

Дисплей

Поточний стан температури теплоносія та вихідних газів має відображатися на дисплеї. Через меню на дисплеї потрібно корегувати задану (цільову) температуру, режим роботи котла тощо.

У даній роботі ми використовуємо LCD1602 Keypad SHIELD. Дана плата розширення представляє собою рідкокристалічний дисплей LCD1602 із вбудованими 6 кнопками (5 кнопок для керування пристроєм, та кнопка reset). Даний модуль можна використовувати для візуалізації меню пристрою з можливістю переміщення по пунктам меню [23]. На рис. 2.11 зображено LCD1602 Keypad SHIELD та дана його схема з описом входів/виходів.

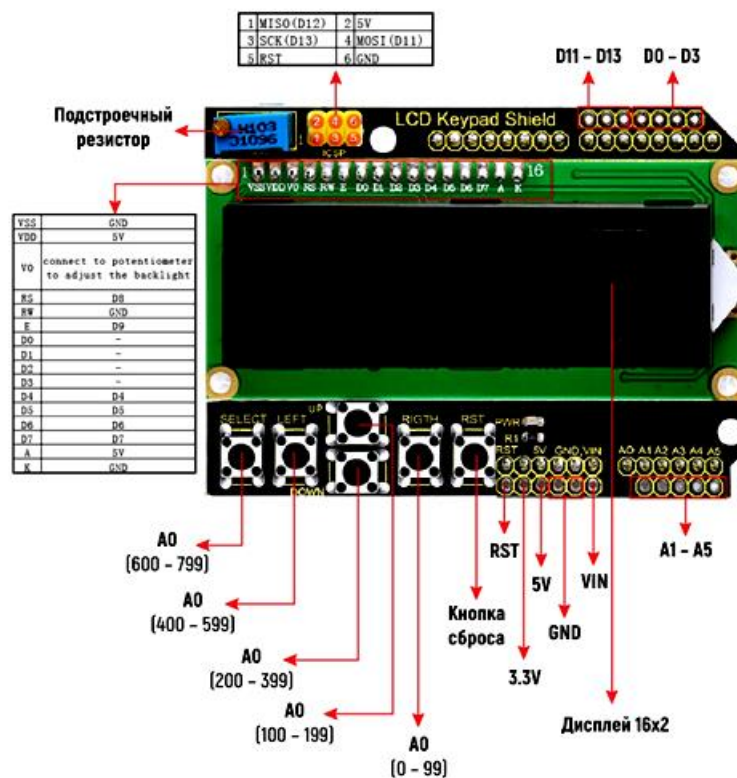


Рисунок 2.11 – Плата розширення LCD1602 Keypad SHIELD

Характеристики:

- робоча напруга: 4,7-5,5 В;
- мікросхема: HD44780;
- колір екрану: синій;
- колір тексту: білий.

Даний модуль підключається до плати Arduino Uno навісним монтажем. Живлення може поступати до модулю від платформи Arduino, інших мікроконтролерних пристроїв чи зовнішнього джерела живлення. Про наявність живлення сигналізуватиме світлодіод PWR.

Для роботи з даним модулем використовується стандартна бібліотека Liquid Crystal.

Додаткові модулі

Звичайно, можливості системи автоматизації на базі Arduino не обмежуються описаними вище модулями. Можна додати можливість підключення автоматки до Wi-Fi мережі для віддаленого управління котлом, підтримку SIM та можливість оповістити у SMS у разі критичної ситуації тощо.

2.2 Програмне забезпечення, необхідне для виконання задачі

Arduino Uno можна під'єднати напряму до ПК через USB порт. Таким чином, живлення плати відбувається через порт, і також ми можемо виконувати програмування даної плати. Arduino IDE – середовище розробки для платформи Arduino, що складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна виводу тексту (консолі), панелі інструментів з кнопками часто використовуваних команд і декількох меню. Для завантаження програм і зв'язку середовище розробки підключається до апаратної частини Arduino.

Інтерфейс програми представлений на рис. 2.12. Програма доступна на платформах Windows, MacOS та Linux.

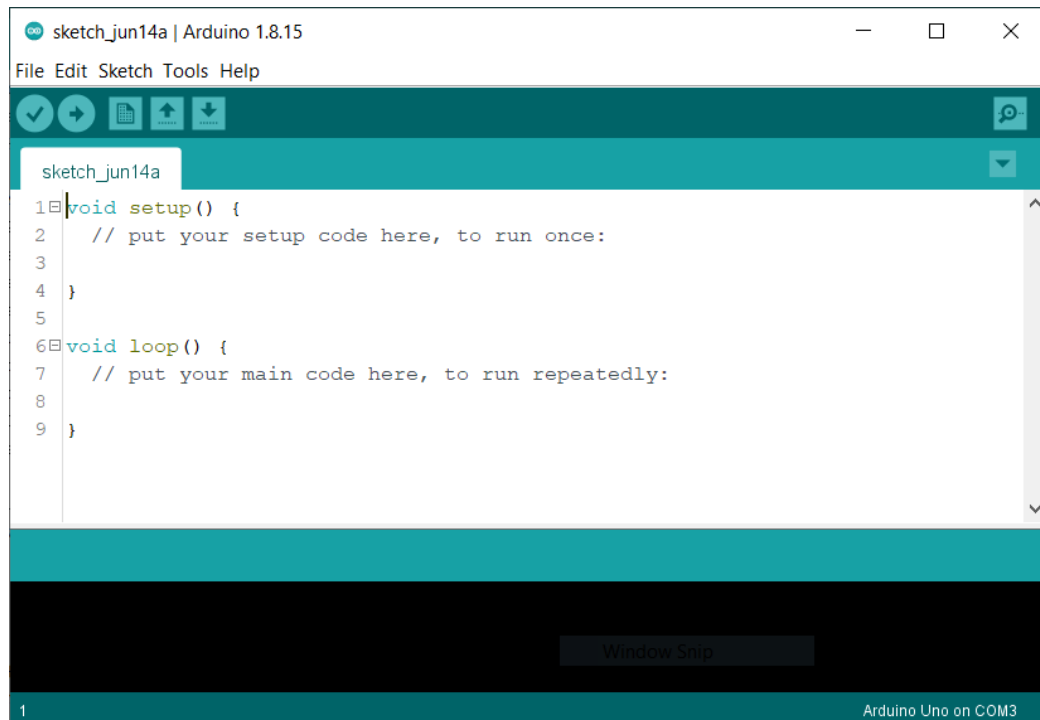


Рис 2.12 – Інтерфейс основного вікна Arduino IDE на платформі Windows

Програма, написана в середовищі Arduino, називається скетч. Скетч пишеться в текстовому редакторі, що має інструменти вирізки/вставки, пошуку/заміни тексту. Під час збереження і експорту проекту в області повідомлень з'являються пояснення, також можуть відображатися виникли помилки. Консоль показує повідомлення Arduino, що включають повні звіти про помилки та іншу інформацію. Кнопки панелі інструментів дозволяють перевірити і записати програму, створити, відкрити і зберегти скетч, відкрити моніторинг послідовної шини. [24]

Перед завантаженням скетчу потрібно задати необхідні параметри в меню Tools> Board і Tools> Serial Port. Після вибору порту і платформи необхідно натиснути кнопку завантаження на панелі інструментів або вибрати пункт меню File> Upload to I/O Board. Сучасні платформи Arduino

перезавантажуються автоматично перед завантаженням. На старих платформах необхідно натиснути кнопку перезавантаження. На більшості плат під час процесу будуть мигати світлодіоди RX і TX. Середовище розробки Arduino виведе повідомлення про закінчення завантаження або про помилки.

При завантаженні скетчу до плати використовується Завантажувач (Bootloader) Arduino, невелика програма, що завантажується в мікроконтролер на платі. Вона дозволяє завантажувати програмний код без використання додаткових апаратних засобів. Завантажувач активний протягом декількох секунд при перезавантаженні платформи і при завантаженні будь-якого з скетчів в мікроконтролер.

Бібліотека – це набір функцій, призначених для того, щоб максимально спростити роботу з різними датчиками, ЖК-екранами, модулями та ін. Наприклад, вбудована бібліотека LiquidCrystal дозволяє легко взаємодіяти з символьними LCD-екранами. Для використання бібліотеки необхідно вибрати меню Sketch> Import Library. Одна або кілька директив `#include` будуть розміщені на початку коду скетчу з подальшою компіляцією бібліотек і разом зі скетчем. Завантаження бібліотек вимагає додаткового місця в пам'яті Arduino. Деякі бібліотеки включені в середу розробки Arduino, інші можуть бути завантажені з різних ресурсів.

Для створення схеми приладу використовуємо програму Fritzing. Це програмне забезпечення з відкритим кодом для віртуального моделювання електричних ланцюгів, схем і електронного обладнання. За допомогою програми можна перетворити прототип на основі Arduino в топологію друкованої плати для серійного виготовлення. Також додаток спрощує налагодження схем за допомогою макетних плат.

3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис апаратної частини

Основою нашого приладу є плата Arduino Uno. До даної плати під'єднаємо LCD1602 Keypad SHIELD. Даний модуль підключається до плати Arduino Uno навісним монтажем.

Серед стандартних модулів, представлених у Fritzing, є декілька дисплеїв. Однак вони суттєво відрізняються від вибраного нами дисплею LCD1602 Keypad SHIELD, тому імпортуємо макет дисплею, знайдений у Інтернеті. [25] На рис. 3.1 зображено схему підключення дисплею до плати Arduino Uno.

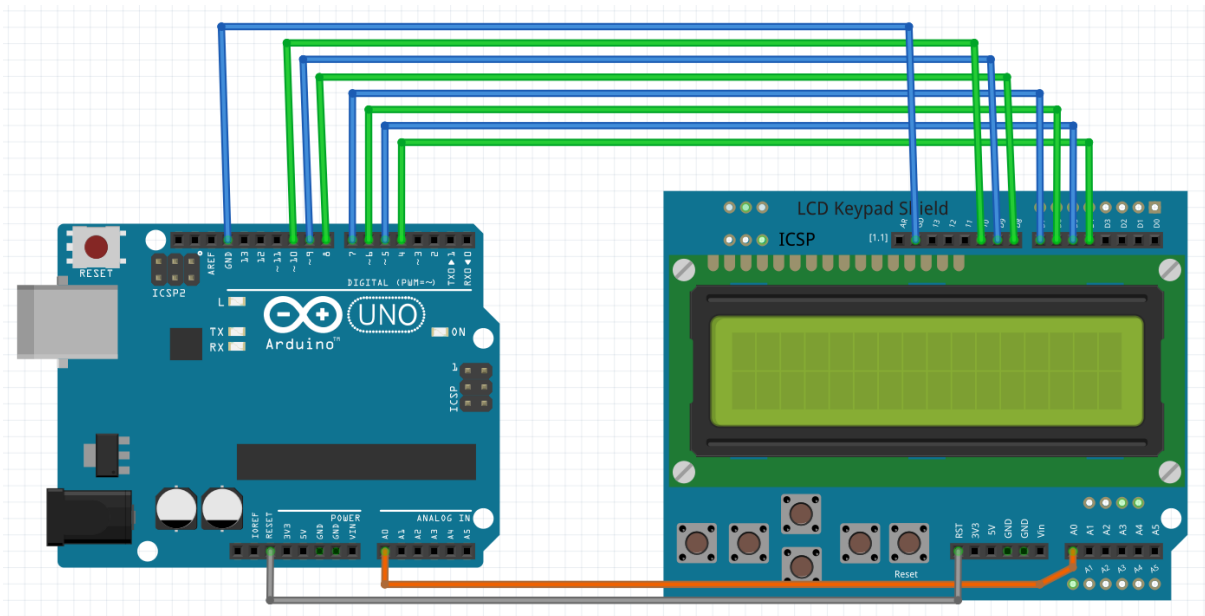


Рисунок 3.1 – Під'єднання шилду дисплея до плати

Знайдений нами макет не є повністю ідентичним платі, що використовується, з огляду на розташування цифрових портів, однак в цілому досить точно відображує схему використаного модулю.

Оскільки нам потрібно буде додавати багато додаткових модулів, що потребують живлення, пари вихідних портів, наявних у даній платі, недостатньо. Тому нам потрібно додати до схеми макетну плату (Breadboard). Основне призначення такої плати – конструювання та налагодження прототипів різних пристроїв. Складається даний пристрій з отворів-гнізд з кроком 2,54мм (0,1 дюйма), саме з таким (або кратним йому) кроком розташовуються висновки на більшості сучасних радіодеталей. Макетні плати бувають різних розмірів.

Схема електричних з'єднань гнізд зображена на рис. 3.2: п'ять отворів з кожного боку в кожному з рядів (в даному випадку 30) електрично з'єднані між собою. Ліворуч і праворуч знаходиться по дві лінії живлення: тут всі отвори в стовпці з'єднані між собою.

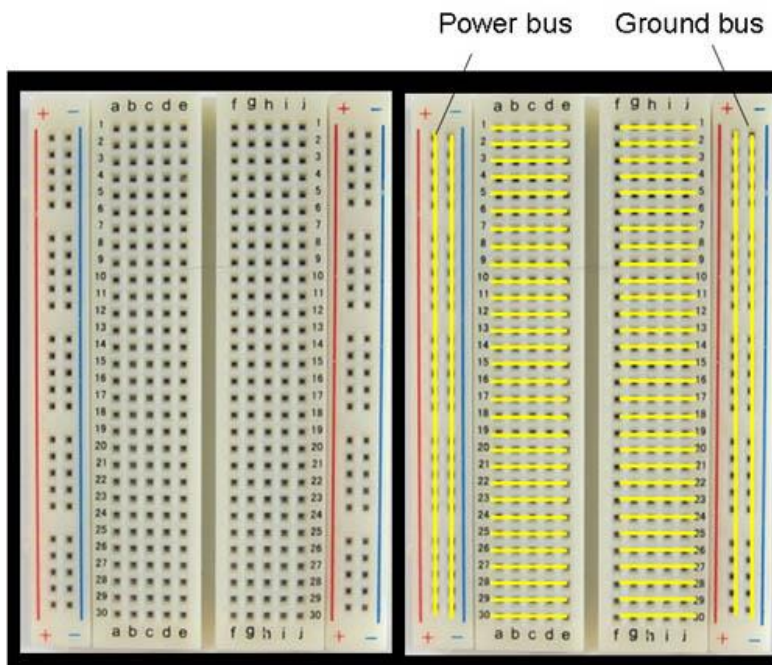


Рисунок 3.2 – Макетна плата

Отже, додаємо до схеми макетну плату та під'єднаємо живлення. На рис. 3.3 червоним зображено дрiт з порту 5В, та чорним – дрiт, що з'єднує GND на шилді з платою.

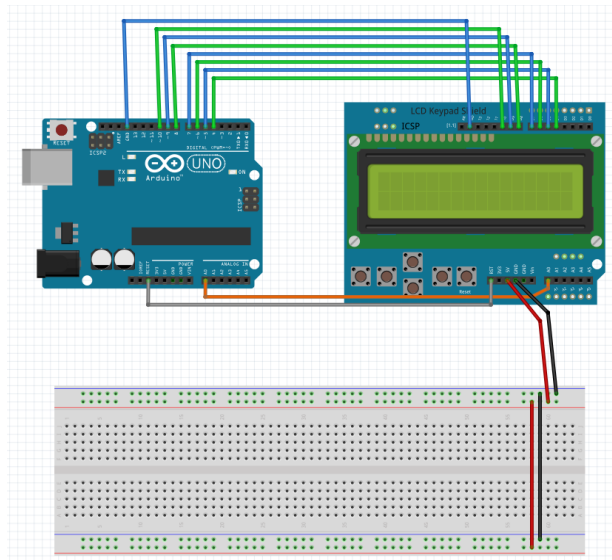


Рисунок 3.3 – Підключення живлення до макетної плати

Перш за все додаємо до схеми датчики. У стандартній бібліотеці Fritzing немає модуля датчика DS18B20 у вологонепроникному корпусі, однак у нашому випадку значення має лише спосіб під'єднання датчика до плати (див. рис. 3.4). При підключенні використовуємо резистор на 4.7 кОм.

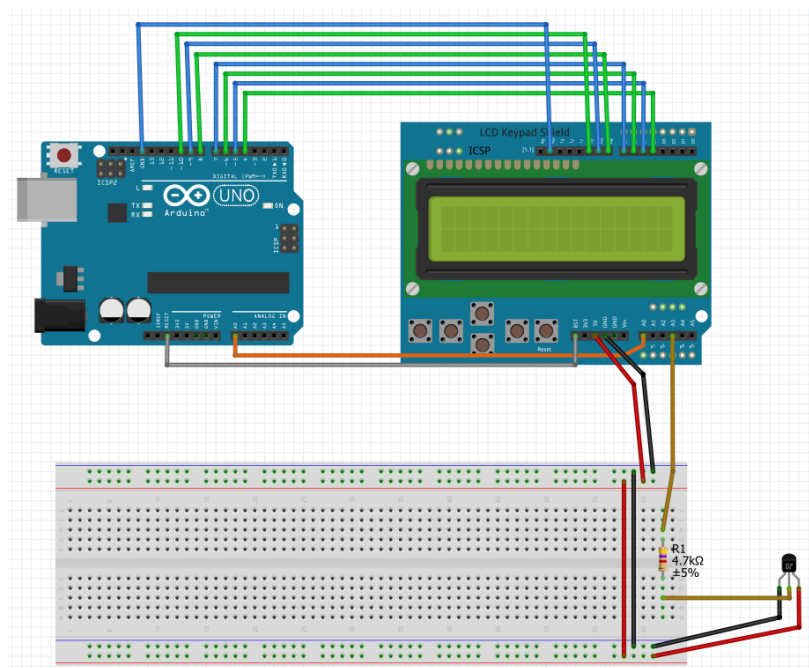


Рисунок 3.4 – Датчик DS18B20 додано до схеми

Коричневим позначено підключення датчику до аналогового порту плати розширення LCD1602 Keypad SHIELD. Даний датчик можна підключати і по цифровому порту.

Під'єднаємо датчик температури димових газів – термопару К-типу з підсилювачем MAX6675. Підсилювач зазначено на схемі на рис. 3.5

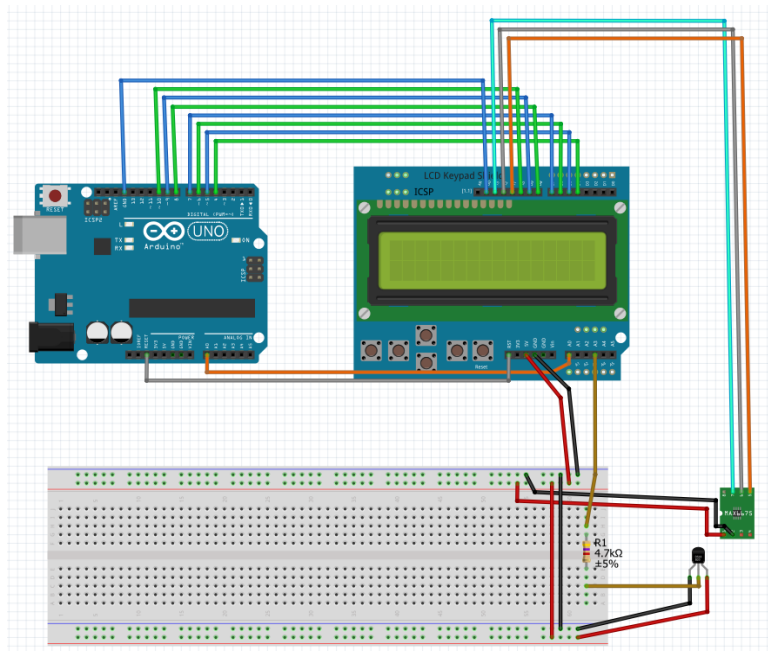


Рисунок 3.5 – MAX6675 додано до схеми

Термопару під'єднано до підсилювача, як показано на рис. 3.6.

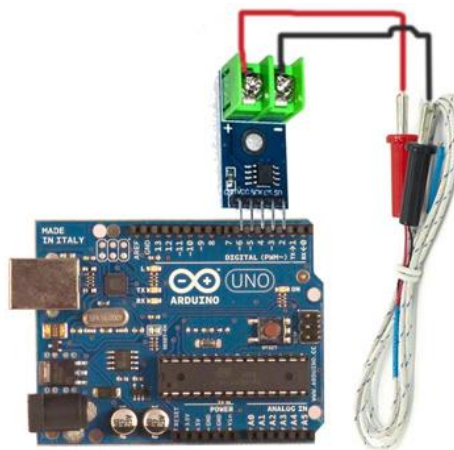


Рисунок 3.6 – Зв'язок термопарі з підсилювачем

Останній датчик, що потрібно під'єднати до схеми – датчик рівня чадного газу у оточуючому середовищі MQ-7. Датчик можна під'єднувати з використанням як цифрового, так і аналогового порту. Наразі використаємо аналоговий порт.

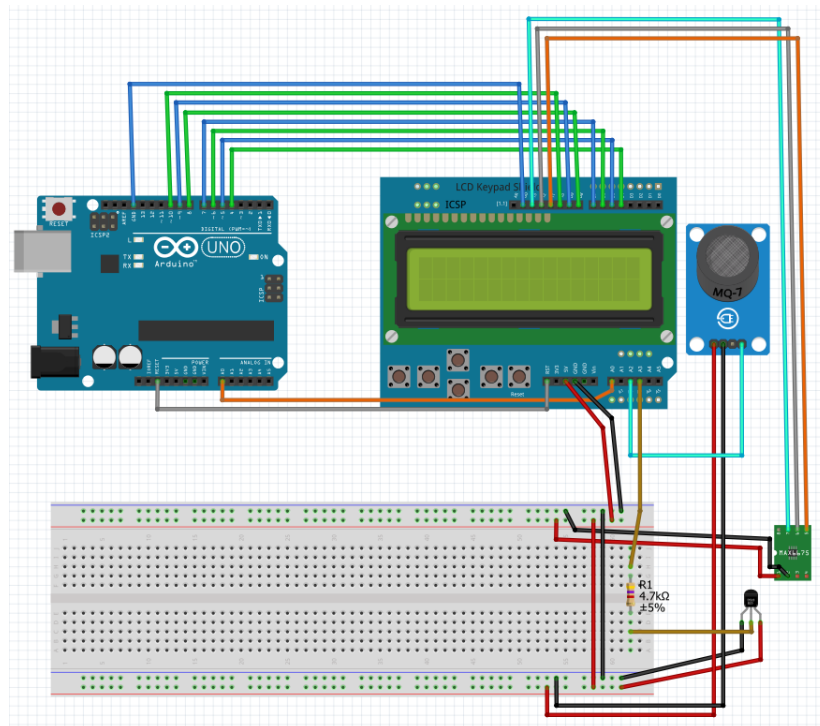


Рисунок 3.7 – Датчик MQ-7 додано до схеми

Таким чином ми додали до схеми усі використані датчики контролю роботи котла. З їх допомогою можна визначити, які дії необхідні для підвищення/зменшення рівню горіння у котлі з метою регуляції температури. Тепер додаємо управляючі модулі, якими керуватиме автоматика.

Перш за все маємо додати до схеми серво-приводи, що регулюватимуть рівень повороту шибера (заслонки) та режимом роботи котла. На рис. 3.8 дроти, що поєднують серво-приводи з цифровими портами плати LCD1602 Keypad SHIELD, показано жовтим (шибер) та сірим (режим роботи котла) кольором.

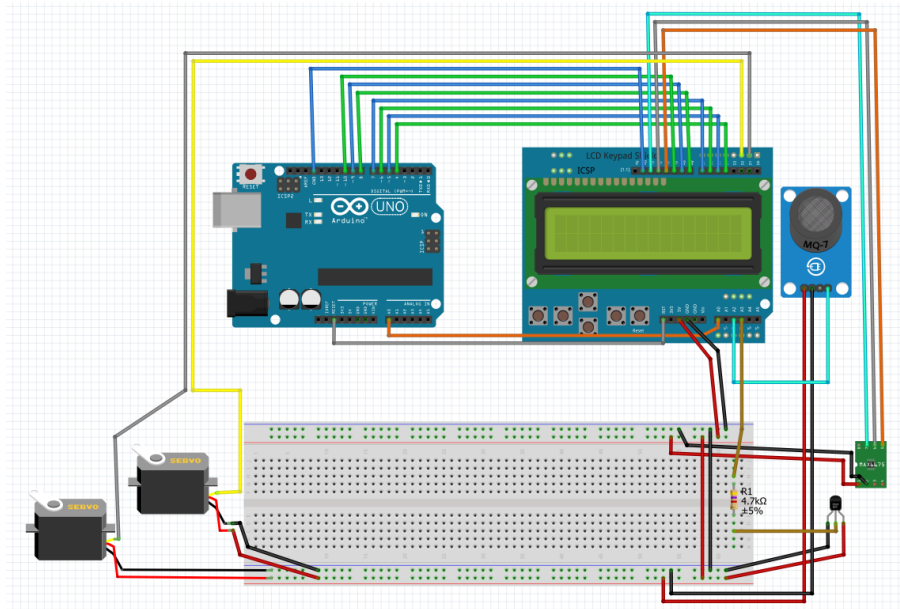


Рисунок 3.8 – До схеми додано 2 серво-приводи

Після підключення серво, додаємо до схеми реле, що управлятиме циркуляційним насосом. Реле під'єднано до плати з використанням двох аналогових портів, як зображено на рис. 3.9. Використовуватимемо подання напруги HIGH/LOW для регуляції швидкості роботи насосу (див. підрозділ 2.1).

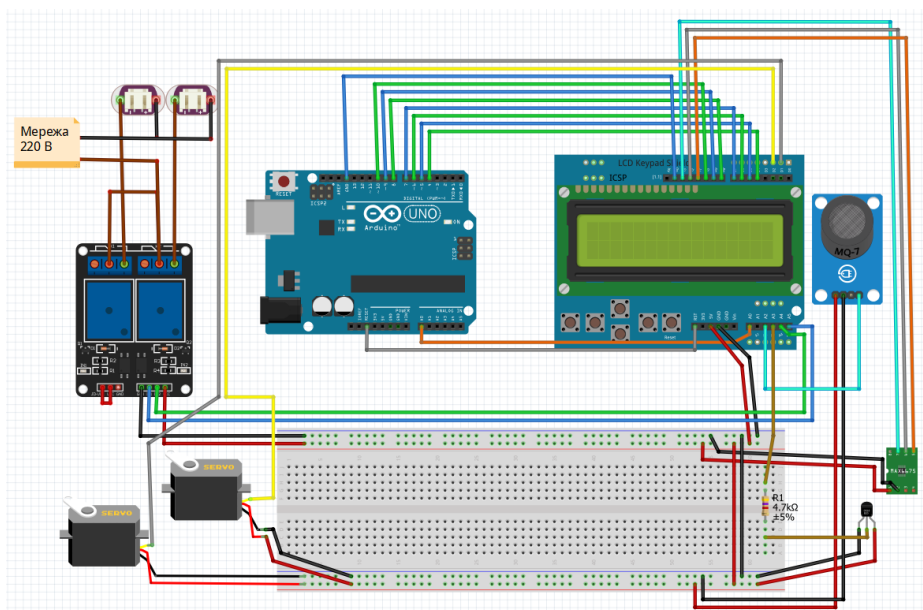


Рисунок 3.9 – Під'єднано двоканальне реле

Останніми додаємо до плати світлодіод та п'єзовипромінювач. При під'єднання їх до макетної плати, використовуємо резистор для підключення сигналу до світлодіоду.

Таким чином, на рис. 3.10 зображена фінальна схема прототипу автоматики твердопаливного котла.

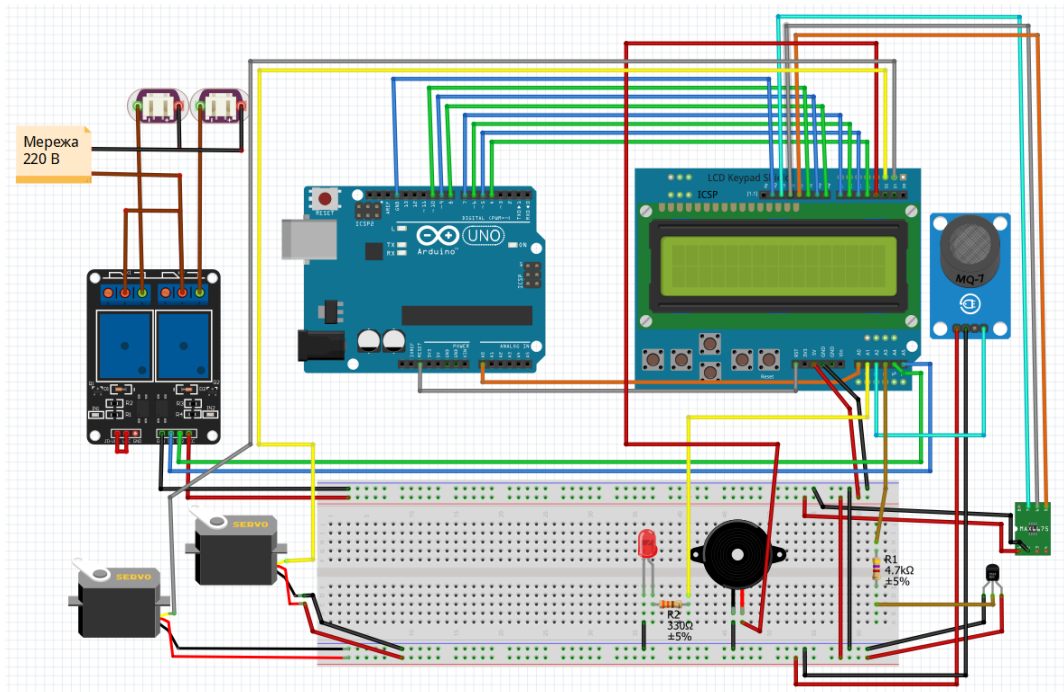


Рисунок 3.10 – Фінальна схема приладу

Для використання даного приладу нам потрібно під'єднати реле до циркуляційного насосу, а сам прилад – до центральної мережі електроенергії. Також до приладу бажано під'єднати додаткове живлення на випадок відключення електроенергії.

3.2 Алгоритм роботи програми

Основою алгоритму роботи автоматики твердопаливного котла є вимірювання температури теплоносія та корегування постачання повітря

до камери горіння (таким чином регулюючи силу горіння та нагріву води у системі опалення).

Також ми використовуємо додатково датчики температури димових газів та датчик чадного газу для забезпечення безпеки користування котлом. Розглянемо алгоритм системи детальніше.

Програма роботи плати Arduino складається з двох частин: `setup()` та `loop()`, перша з яких виконується один раз, а друга виконується циклічно.

У нашому випадку, у частині `setup()` запускаємо модулі, що використовуються у програмі, як показано на рис. 3.11, та зчитуємо з пам'яті останній обраний режим. Якщо це перший запуск програми, вибираємо стандартний режим роботи. Затримка у одну хвилину використовується для розігріву датчиків. Зчитуємо поточні дані з датчиків та зберігаємо їх у пам'яті для можливості порівняння пізніше у циклі.

```

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  ServoShiber.attach(ServoShiberPIN);
  ServoRegime.attach(ServoRegimePIN);
  alarmEnabled=0;
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(Relay1PIN, OUTPUT);
  pinMode(Relay2PIN, OUTPUT);
  currentOperationMode=0;
  if(int(EEPROM.read(0))!=255)
    EEPROM.write(0, byte(currentOperationMode));
  else currentOperationMode=int(EEPROM.read(0));
  OperationModeSelected();
  delay(60000);
  getCurrentData();
  setPreviousToNow();
}

```

Рисунок 3.11 – Код запуску програми

Основний алгоритм задано у циклі `loop()`. Перш за все, щоразу під час запуску циклу зчитуємо дані з датчиків, а також поточний час (див. рис. 3.12). Зазначимо, що у даному випадку зчитується час роботи приладу у мілісекундах, оскільки наш прилад не має годинника у звичайному розумінні користувача, та не під'єднаний до мережі (що дозволило б

синхронізувати час). Для нашої програми нас цікавить час виконання тої чи іншої задачі. Розпал/гасіння палива відбувається не миттєво, що також означає значний час, необхідний для адаптації температури води у бойлері, тому нам потрібно буде порівнювати показники температури за певний час. Для очікування у Arduino реалізована команда `delay()`, що затримує виконання програми. Однак ми не можемо її використовувати, адже вона призупиняє функціонування приладу.

```
//getting data from sensors
void getCurrentData() {
    timeNow=millis();
    timeCheckDelayValue=timeNow-timePrevious;
    tempGasesNow=thermocouple.readCelsius();
    detectWaterTemperature();
    COLevel=analogRead(analogMQ7);
    showCurrentData();
    tempGasesDifference=tempGasesNow-tempGasesPrevious;
    tempWaterDifference=tempWaterNow-tempWaterPrevious;
}
```

Рисунок 3.12 – Код функції отримання даних з датчиків

У рамках отримання поточних даних з датчиків також запускається метод виведення даних на дисплей. Після отримання даних з датчиків порівнюємо їх із заданими показниками для перевірки необхідності запуску тривоги. Дані функції виконуються незалежно від часу роботи програми, щоразу при запуску циклу.

Показники максимальної температури теплоносія та димових газів обираються згідно до вибраного режиму роботи. У якості стандартних показників для першого режиму Default ми встановили 90°C – для води та 580°C – для димових газів. Дані щодо відповідних показників кожного режиму можна змінити використовуючи меню, як показано на рис. 3.13.



Рисунок 3.13 – Меню

У разі якщо норми температури води, димових газів, або норми щодо чадного газу у повітрі перевищені, запускається режим гасіння котла (закриваємо шибер шляхом повороту серво-мотору на 90 градусів). Також запускається звуковий сигнал та червоний світлодіод. На рис. 3.14 показано частину коду методу перевірки необхідності запуску аварійного сигналу.

```

void checkAlarm() {
  //Gas
  if (tempGasesNow>tempGasAlarm)
  {
    Serial.print("Внимание! Высокая температура горения (датчик газов): ");
    Serial.println(tempGasesNow);
    firePutDown();
    enableLedAndAlarm();
    lcd.setCursor(7,1);
    lcd.print("!");
  }
  //Water

```

Рисунок 3.14 – Код перевірки необхідності запуску сигналізації у разі перевищення температури димових газів

Основна частина циклу роботи програми виконується раз у п'ять хвилин, якщо не було запуснено сигналізацію. Опишемо її детальніше.

Перш за все визначаємо етап роботи котла. У нашій роботі визначено 6 можливих етапів, що показано на рис. 3.15.

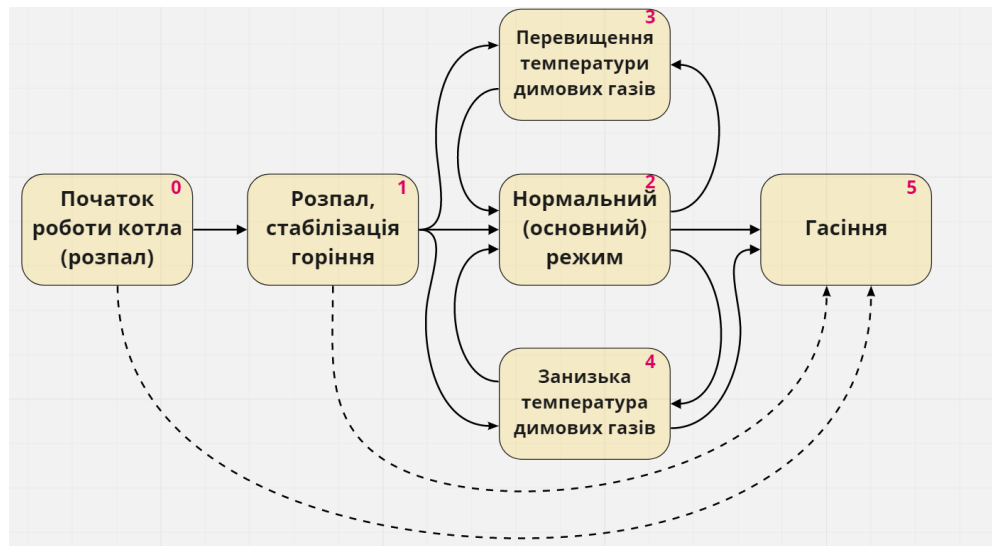


Рисунок 3.15 – Схема режимів роботи котла

0 – початковий етап розпалу котла. Його характеристика: поточна температура димових газів нижча за 200°C (тут і далі приводимо дані щодо температур режиму Default) та вища за температуру газів 5 хвилин тому. На даному етапі встановлюється одноходовий режим роботи котла, кут повороту шиберу – 0. Таким чином ми забезпечуємо найшвидший розпал.

1 – другий етап розпалу. Температура димових газів перевищує 200°C , а у порівнянні з температурою, виміряною 5 хвилин тому, запевняємося, що вона продовжує зростати. Ми продовжуємо використовувати одноходовий режим, однак кут повороту шиберу може змінитися, якщо температура води у котлі значно перевищує цільову (пр. вода у бойлері не встигла охолонути після попереднього запуску котла). У такому випадку ми трохи прикриваємо заслонку (шибер), аби сповільнити розпал та скорегувати температуру теплоносія.

Розглянемо детальніше метод, що керує кутом повороту шиберу для зниження температури теплоносія. Код даного методу представлено на рис. 3.16. Основою є порівняння, наскільки поточна температура у котлі

перевищує цільову. Чим більша різниця, там більший шаг повороту шибера використовується.

```

void getTemperatureDown() {
    Serial.println("Trying to get water temp down");
    if (tempWaterNow-tempWaterTarget > 20) {
        if (ServoShiberValue<70) ServoShiberValue+=15;
        else if (ServoShiberValue>70&&ServoShiberValue<90) ServoShiberValue=85;
    }
    else if ((tempWaterNow-tempWaterTarget) > 10&&(tempWaterNow-tempWaterTarget) < 21) {
        if (ServoShiberValue<=75) ServoShiberValue+=10;
        else if (ServoShiberValue>75&&ServoShiberValue<90) ServoShiberValue=max(80,ServoShiberValue);
    }
    else if ((tempWaterNow-tempWaterTarget) > 5&&(tempWaterNow-tempWaterTarget) < 11) {
        if (ServoShiberValue<=75) ServoShiberValue+=6;
        else if (ServoShiberValue>75&&ServoShiberValue<90) ServoShiberValue=max(80,ServoShiberValue);
    }
    else if ((tempWaterNow-tempWaterTarget) < 6&&(tempWaterNow-tempWaterTarget) > 1) {
        if (ServoShiberValue<85) ServoShiberValue+=2;
    }
}

```

Рисунок 3.16 – Код методу повороту шибера для зниження температури теплоносія в залежності від поточної температури

Оскільки зміна температури відбувається поступово, немає необхідності з першого разу значно змінювати кут повороту шибера. У разі, якщо проведених змін було недостатньо, під час наступного витку п'ятихвилинного циклу збільшимо кут повороту шибера.

З етапу 0 та етапу 1 можна перейти до етапу 5 (гасіння) у випадку критичного перевищення температури теплоносія (автоматично) або при нестачі палива для розпалу (природним шляхом).

2 – нормальний режим роботи. Він може відрізнятися в залежності від котла, однак середнє значення температури димових газів – від 270°C до 520°C [27]. У такому випадку серво використовується для вмикання триходового режиму котла, що забезпечує довше зберігання тепла, а значить – менші витрати палива. У даному випадку зосереджуємося на температурі теплоносія та регулюємо кут повороту шибера з огляду на різницю температур (поточної та бажаної). Для цього використовуємо методи `getTemperatureDown()` (розглянутий на рис. 3.16) та

getTemperatureUp() (аналогічний метод для зменшення куту повороту шиберу у разі необхідності підвищити температуру води).

3 – у разі перевищення температури димових газів (більша за 520°C) переходимо до етапу підвищеної температури горіння. Використовуємо триходовий режим роботи котла. Для уникнення перевищення температури у котлі збільшуємо кут повороту шиберу на 5 градусів (якщо поточна температура теплоносія нижча або дорівнює цільовій) чи більше (використовуючи метод getTemperatureDown()).

4 – у разі зменшення температури димових газів переходимо до етапу 4 та вмикаємо одноходовий режим роботи котла для швидшого розпалу котла та підтримання горіння.

5 – етап гасіння. Якщо температура димових газів менша ніж 150°C, і при цьому температура п'ять хвилин тому була вища, вважаємо, що котел переводиться у режим гасіння.

Окрім визначення режиму та корегування куту повороту шиберу, у рамках даного циклу також управляємо циркуляційним насосом. Алгоритм управління насосом наведено нижче:

- якщо температура теплоносія приблизно дорівнює цільовій температурі (поточна температура відрізняється від цільової не більше ніж на Δt , визначену згідно режиму роботи котла), насос працює на швидкості 1;

- якщо температура теплоносія значно вища, ніж цільова (більше ніж на Δt), насос працює на швидкості 2 задля скорішого зниження температури води та уникнення кипіння;

- якщо вода у бойлері недостатньо прогрілася (температура теплоносія менша за цільову щонайменше на Δt), насос не працює задля економії електроенергії.

ВИСНОВКИ

У даній роботі ми спроектували прототип автоматики управління твердопаливним котлом. Основою даного пристрою є мікроконтролер Arduino Uno, до якого під'єднано дисплей LCD1602 Keypad SHIELD, на який виводяться поточні результати вимірювання з датчиків, вибраний режим роботи котла (одно-/триходовий) та кут повороту заслонки (шиберу).

Пристрій отримує дані з водонепроникного датчика температури теплоносія, датчика температури димових газів (використовується термопара К-типу), та датчика MQ-7, що вимірює рівень чадного газу. Дані щодо температури теплоносія є ключовими. Дані щодо температури димових газів та рівня CO допомагають запевнитися у безпечності використання котлом та попередити користувача у разі аварійної ситуації. Під'єднані до автоматики серво-мотори використовуються для керування кутом повороту шиберу та режимом роботи котла. Двоканальне реле корегує швидкість роботи циркуляційного насосу.

Користувач може встановити бажаний режим роботи котла, використовуючи меню. Налаштування режимів зберігаються у енергонезалежній пам'яті пристрою. Бажано провести налаштування під час першої установки пристрою для конкретного котла з огляду на встановлені для нього температурні норми.

Даний прототип пристрою має базовий функціонал автоматики та може бути розширений шляхом додавання модулів для віддаленого управління автоматикою.

СПИСОК ЛІТЕРАТУРИ

Книга:

1. Brian Evans Beginning Arduino Programming / NY: Apress, 2011. – 271 p.
2. Jonathan Osher, Hugh Blemings Practical Arduino: Cool Projects for Open Source Hardware / NY: Apress, 2009. – 423 p.

Web-сторінка:

3. Умный и комфортный дом [Электронный ресурс] – Режим доступа: <https://mind42.com/mindmap/a83fba6c-dd6e-470f-96d3-3dca31f56af1>
4. 6 сучасних методів опалення будинку без газу [Електронний ресурс] – Режим доступа: <https://evrodim.com/publications/inzhenernye-sistemy/6-suchasnix-metodiv-opalennya-budinku-bez-gazu>
5. Вікіпедія – вільна енциклопедія [Електронний ресурс] – Режим доступа: <https://uk.wikipedia.org/wiki/>
6. Найбільш ефективне та економічне опалення приватного будинку [Електронний ресурс] – Режим доступа: <https://mycond-heatpump.com.ua/ua/naybilsh-efektivne-ta-ekonomichne-opa/>
7. Ефективне опалення приватного будинку. Способи обігріву будинку [Електронний ресурс] – Режим доступа: <https://budprofi.com/efektyvne-opalennya-pryvatnogo-budynku-sposoby-obigrivu-budynku/>
8. Как подобрать твердотопливный котел и систему отопления! [Електронний ресурс] – Режим доступа: <https://opop-servis.com.ua/item/26-kakpodobrattverdotoplivnuykotel.html>
9. Трехходовой клапан для отопления: устройство, принцип работы и схема подключения [Електронний ресурс] – Режим доступа: <https://znayteplo.ru/otoplenie/kak-rabotayut-trexxodovye-klapany-i-servoprivody/>

10. Рабочая температура твердотопливного котла [Электронный ресурс] – Режим доступа: <https://kronas.pro/pravilnaya-ekspluataciya-tverdotoplivnogo-kotla>
11. Что такое комнатный термостат для котла и зачем он нужен? [Электронный ресурс] – Режим доступа: <https://solar-tt.com/что-такое-komnatnyiy-termostat-dlya-kotla-i-zachem-on-nuzhen/>
12. Контроллер TAL RT22 [Электронный ресурс] – Режим доступа: <https://aldentrade.com.ua/content/files/tal-rt22n-41870100.jpg>
13. Инструкция обслуживания контроллера CS-20 [Электронный ресурс] – Режим доступа: <https://aldentrade.com.ua/content/files/nstrukcja-cs-20-64894142.pdf>
14. ST-81 – инструкция обслуживания [Электронный ресурс] – Режим доступа: <http://www.techpribor.com/images/Instrukcii/St81BEZzPID.pdf>
15. Техническая документация панели управления PROND – PILOT R [Электронный ресурс] – Режим доступа: <https://protechplus.com.ua/files/prond-pilot-r.pdf>
16. STM32 – микроконтроллер для начинающих после Arduino [Электронный ресурс] – Режим доступа: <https://arduinomaster.ru/stm32/stm32-mikrokontroller-dlya-nachinayushhih-posle-arduino/>
17. Что такое Ардуино [Электронный ресурс] – Режим доступа: <https://arduinoplus.ru/что-такое-arduino/>
18. База знаний Arduino, 3D принтеры и ЧПУ | 3DIY [Электронный ресурс] – Режим доступа: <https://3d-diy.ru/wiki/>
19. Причины кипения воды в котле [Электронный ресурс] – Режим доступа: <https://termico.com.ua/prichiny-kipeniya-vody-v-kotle/>
20. Термопара К-типу з цифровим підсилювачем на MAX6675 [Электронный ресурс] – Режим доступа: <https://arduino.ua/prod1785-termopara-k-tipa-s-cifrovim-ysilitelem-na-max6675>

21. Stive Hassan Interfacing Buzzer to Arduino [Электронный ресурс] – Режим доступа: <https://www.instructables.com/Interfacing-Buzzer-to-Arduino/>
22. Arduino Tutorial [Электронный ресурс] – Режим доступа: <https://www.ladyada.net/learn/arduino/lesson3.html>
23. LCD1602 Keypad SHIELD Модуль дисплея с кнопками [Электронный ресурс] – Режим доступа: <https://arduino-geek.pp.ua/product/lcd1602-keypad-shield-модуль-дисплея-с-кнопками/>
24. Среда разработки Arduino [Электронный ресурс] – Режим доступа: http://arduino.ru/Arduino_environment
25. Fritzing/LCD Keypad Shield.fzpz [Электронный ресурс] – Режим доступа: <https://github.com/RafaGS/Fritzing/blob/master/LCD%20Keypad%20Shield.fzpz>
26. Breadboard — электронный конструктор для всех [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/150257/>
27. Технические характеристики Твердотопливного котла [Электронный ресурс] – Режим доступа: <http://www.ktto.com.ua/karakteristiki/ktt>

ДОДАТОК А

Фрагмент коду програми пристрою (запуск, цикл, основні функції)

```

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  ServoShiber.attach(ServoShiberPIN);
  ServoRegime.attach(ServoRegimePIN);
  alarmEnabled=0;
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(Relay1PIN, OUTPUT);
  pinMode(Relay2PIN, OUTPUT);
  currentOperationMode=0;
  if(int(EEPROM.read(0))==255)
    EEPROM.write(0, byte(currentOperationMode));
    else currentOperationMode=int(EEPROM.read(0));
  OperationModeSelected();
  delay(60000);
  getCurrentData();
  setPreviousToNow();
}

void loop() {
  getCurrentData();
  //first check if there is something to set alarm upon
  checkAlarm();
  checkDisableAlarm();
  //if there is nothing to set alarm for, check what's going on
  if(timeCheckDelayValue>300000&&!alarmEnabled) {
  //300000 - 5 minutes.
  DefineCurrentMode();
  switch(mode) {
    case 0: //Starting
      ServoRegimeValue=0;
      ServoShiberValue=0;
      if (tempWaterNow>(tempWaterAlarm-tempWaterDelta)
{
          enableLedAndAlarm();
          firePutDown();
          Serial.print("Water temperature is too high
= ");
          Serial.println(tempWaterNow);
        }
      break;
    case 1: //HeatUp
      ServoRegimeValue=0;

```

```

ServoShiberValue=0;
if (tempWaterNow>(tempWaterAlarm-tempWaterDelta))
{
    enableLedAndAlarm();
    firePutDown();
    Serial.print("Water temperature is too high
= ");
    Serial.println(tempWaterNow);
}
else if
(tempWaterNow>(tempWaterTarget+tempWaterDelta))
    getTemperatureDown();
    break;
case 2: //Normal
    ServoRegimeValue=90;
    if (tempWaterNow>tempWaterTarget)
        getTemperatureDown();
    else if (tempWaterNow<tempWaterTarget)
        getTemperatureUp();
    break;
case 3: //HighTemperature
    ServoRegimeValue=90;
    if (tempWaterNow>tempWaterTarget)
        getTemperatureDown();
    else if (ServoShiberValue<85)
        ServoShiberValue+=5;
    break;
case 4: //LowTemperature
    ServoRegimeValue=0;
    if
(tempWaterNow>tempWaterTarget&&tempWaterNow<(tempWaterTarget+t
empWaterDelta))
        getTemperatureDown();
    else if (tempWaterNow<tempWaterTarget)
        getTemperatureUp();
    break;
case 5: //PutFireDown
    firePutDown();
    break;
}
ServoRegime.write(ServoRegimeValue);
ServoShiber.write(ServoShiberValue);
SetCirculationPumpMode(); //реле ВКЛ/ВЫКЛ в
зависимости от температуры воды
setPreviousToNow();
}
Serial.print("Время с последней проверки - ");
Serial.print(timeCheckDelayValue/1000.0);
Serial.println(" секунд.");
Serial.println("=====");
}

```

```

//this one to be executed to save current data for further
reference, before going on the next round to get new data
void setPreviousToNow() {
    tempGasesPrevious = tempGasesNow;
    tempWaterPrevious=tempWaterNow;
    timePrevious = timeNow;
}

int detectWaterTemperature(){
    byte data[2];
    byte tempW;
    oneWire.reset();
    oneWire.write(0xCC);
    oneWire.write(0x44);
    oneWire.write(0xCC);
    oneWire.write(0xBE);
    data[0] = oneWire.read();
    data[1] = oneWire.read();
    tempW = (data[1] << 8) + data[0]; tempW = tempW >> 4;
    tempWaterNow=tempW;
}

//getting data from sensors
void getCurrentData() {
    timeNow=millis();
    timeCheckDelayValue=timeNow-timePrevious;
    tempGasesNow=thermocouple.readCelsius();
    detectWaterTemperature();
    COLevel=analogRead(analogMQ7);
    showCurrentData();
    tempGasesDifference=tempGasesNow-tempGasesPrevious;
}

//printing data on LCD
void showCurrentData() {
    //Water temperature
    Serial.print("Температура воды в котле - ");
    Serial.println(tempWaterNow);
    lcd.setCursor(0,0);
    lcd.print("Wtr=");
    lcd.setCursor(4,0);
    lcd.print(tempWaterNow);

    //show regime used
    lcd.setCursor(8,0);
    if (ServoRegimeValue==0) {
        lcd.print("1x");
    }
    else if (ServoRegimeValue==90) {
        lcd.print("3x");
    }
}

```

```
    }  
  
    //Schieber  
    lcd.setCursor(11,0);  
    lcd.print("Z=");  
    lcd.setCursor(13,1);  
    lcd.print(ServoShiberValue);  
  
    //Gases temperature  
    Serial.print("Температура газа в котле - ");  
    Serial.println(tempGasesNow);  
    lcd.setCursor(0,1);  
    lcd.print("Gas=");  
    lcd.setCursor(4,1);  
    lcd.print(tempGasesNow);  
  
    //CO level  
    Serial.print("CO Level is ");  
    Serial.println(COLevel);  
    lcd.setCursor(8,1);  
    lcd.print("CO=");  
    lcd.setCursor(11,1);  
    lcd.print(COLevel);  
}
```