МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп’ютерних систем та інфокомунікацій

Кафедра комп’ютерних систем, мереж і кібербезпеки

**Курсова робота**

|  |  |
| --- | --- |
| з | *Операційні системи* |
|  | (назва дисципліни) |
| на тему | «Система управління теплообмінником» |

ХАІ.503.525ст1.123.9631823ПЗ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Виконав: здобувач (ка) | | | *3* | курсу групи № | | | | 525ст1 |
| напряму підготовки (спеціальності) | | | | | | | | |
| *123 – комп’ютерна інженерія* | | | | | | | | |
| (шифр і назва напряму підготовки (спеціальності)) | | | | | | | | |
| *Михайлов Олександр Олексійович* | | | | | | | | |
| (прізвище й ініціали здобувача) | | | | | | | | |
| Керівник: | *доцент закладу вищої освіти, к.т.н. Бабешко Є.В.* | | | | | | | |
| (посада, науковий ступінь, прізвище й ініціали) | | | | | | | | |
| Національна шкала: | | | | |  | | | |
| Кількість балів: | | | | |  | | | |
| Члени комісії: | |  | | | |  |  | |
|  | | (підпис) | | | |  | (прізвище й ініціали) | |
|  | |  | | | |  |  | |
|  | | (підпис) | | | |  | (прізвище й ініціали) | |
|  | |  | | | |  |  | |
|  | | (підпис) | | | |  | (прізвище й ініціали) | |

Харків – 2023

ЗМІСТ

[ЗМІСТ 2](#_Toc135864597)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ 3](#_Toc135864599)

[1.1 Аналіз предметної області 3](#_Toc135864600)

[1.2 Постановка задачі 4](#_Toc135864601)

[2 ПРОЕКТУВАННЯ 5](#_Toc135864602)

[2.1 Програмна частина: 5](#_Toc135864604)

[3 РОЗРОБЛЕННЯ 7](#_Toc135864605)

[3.1 Розробка проекту для CoDeSys 7](#_Toc135864610)

[3.1.1 Загальні вимоги 7](#_Toc135864611)

[3.1.2 Кодування програми 7](#_Toc135864612)

[3.1.3 Проектування візуалізації 7](#_Toc135864613)

[4 ТЕСТУВАННЯ 11](#_Toc135864614)

[4.1 Тестування візуалізації 11](#_Toc135864618)

[ВИСНОВКИ 17](#_Toc135864619)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 18](#_Toc135864620)

[ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ. 19](#_Toc135864621)

[ДОДАТОК Б. ВИХІДНІ КОДИ. 20](#_Toc135864627)

[ДОДАТОК В. НАСТАНОВА КОРИСТУВАЧУ. 24](#_Toc135864630)

[ДОДАТОК Г. ПРЕЗЕНТАЦІЯ. 25](#_Toc135864645)

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

# **Аналіз предметної області**

Котельні сучасних будинків – це складні інженерні системи, які включають різноманітне інженерне обладнання: газові та електричні котли, циркуляційні насоси, двоходові / триходові клапани тощо.

Крім цього, в систему опалення часто включають альтернативні теплогенератори: твердопаливні котли, теплові насоси, сонячні панелі. Ефективне управління таким складним обладнанням для забезпечення комфортної температури в будинку вручну практично не можливо і під силу лише автоматичним системам терморегулювання, побудованим з використанням програмованих контролерів.

Сучасні котли, як правило, комплектуються власною автоматикою, функції якої досить обмежені: контроль за безпекою котла, підтримка заданої температури теплоносія, підтримка заданої температури в приміщенні за допомогою термостата.

Однак ця автоматика не дозволяє управляти додатковими джерелами і споживачами тепла. Крім того, потреби в теплі постійно змінюються і на це впливають: різні температурні режими вдень, вночі, в робочі та вихідні дні, зміна вуличної температури, провітрювання приміщень, необхідність в нагріванні гарячої води тощо.

Програмовані контролери для систем опалення дозволяють підтримувати різні температурні режими для декількох опалювальних контурів, забезпечуючи задані добові і тижневі часові програми. Для підтримки заданої кімнатної температури такі системи є погодо залежними, тобто при зміні зовнішньої температури, контролер автоматично змінює температуру теплоносія. Для контролю зовнішньої температури використовується датчик зовнішньої температури, що встановлюється зовні будівлі, з північного боку. Контролер опалювальної системи також може повністю управляти процесом підготовки гарячої води в бойлері і її рециркуляцією. Багато систем побудовані за модульним принципом і дозволяють комплектувати і модернізувати автоматику під конкретні вимоги замовника (підключати додаткові контури шляхом установки відповідного модуля, без заміни панелі управління в цілому), що забезпечує значну економію коштів.

# **Постановка задачі**

Данна курсова робота спрямована на візуалізацію роботи системи управління теплообмінником у будинку з використанням програми CoDeSys за допомогою якої буде розроблена та візуалізована програма.

Задача полягає у наступному:

* Розробка системи теплообміну для 2-х контурів(опалення та ГВС).
* Автоматизація процесу теплообміну по даним з налаштування.
* Забезпечити оператора статистикою температур з датчиків.

Метою даної програми є візуалізація процесу нагрівання ГВС або опалення

до заданої температури. Очікуваним результатом є підтримка заданої температури у заданій системі.

# 2 ПРОЕКТУВАННЯ

# **Програмна частина**

Для програмування нашої системи теплообміну ми використали програму CoDeSys, що є програмним забезпеченням для автоматизації та управління промисловими процесами. Він базується на міжнародному стандарті IEC 61131-3, який визначає п'ять мов програмування для промислових автоматів. CoDeSys включає всі ці мови, що робить його потужним інструментом для розробки автоматизованих систем.

Однією з головних можливостей CoDeSys є його графічний інтерфейс програмування (Grafcet), який дозволяє розробникам створювати складні діаграми станів та процесів за допомогою графічних елементів. Це зручно та зрозуміло для користувачів без глибоких знань програмування, а також дозволяє швидко та легко розробляти програми.

Крім того, CoDeSys має широкий набір інструментів для налагодження та тестування програм, а також інтегрується з різними промисловими пристроями, що робить його гнучким та універсальним інструментом для розробки програмного забезпечення для промислових автоматизованих систем.

Однією з інших важливих можливостей CoDeSys є можливість використовувати функціональні блоки (Function Block Diagrams), які дозволяють створювати складні програми зі зручним графічним інтерфейсом. Крім того, CoDeSys підтримує мови програмування Structured Text (ST), Sequential Function Chart (SFC) та Instruction List (IL), що дозволяє розробляти програми різними способами залежно від вимог проекту.

Зважаючи на вимоги проекту, ми використали мову Structured Text (ST) в CoDeSys для програмування нашої автоматизованої системи.

Structured Text (ST) - це мова програмування вищого рівня, яка зазвичай використовується в промисловій автоматизації та системах керування, особливо для програмованих контролерів логіки (ПЛК). Вона є текстовою мовою, яка використовує структурований синтаксис і базується на стандарті IEC 61131-3.

ST надає структурований та ефективний спосіб розробки керуючої логіки для автоматизованих систем. Вона нагадує синтаксис мови Pascal і дозволяє програмісту писати складні алгоритми, використовуючи комбінацію структурованих конструкцій програмування, таких як цикли, умовні оператори, функції та типи даних.

Деякі ключові особливості та характеристики мови Structured Text (ST) включають:

* Синтаксис: ST використовує комбінацію ключових слів, операторів та символів для визначення логіки програми. Вона використовує принципи структурованого програмування, такі як послідовний потік виконання, прийняття рішень та циклічні конструкції.
* Типи даних: ST підтримує різні типи даних, включаючи елементарні типи (наприклад, BOOL, INT, REAL), похідні типи (наприклад, масиви, структури) та користувацькі типи.
* Змінні та константи: У ST можна оголошувати змінні та присвоювати їм значення. Константи, зі свого боку, містять фіксовані значення протягом виконання програми.
* Оператори: ST надає широкий спектр операторів для математичних, логічних та інших операцій.

Structured Text базується на принципах структурованого програмування та має стандартизований синтаксис, що полегшує розробку програмного забезпечення для промислових систем автоматизації. Вона є однією з п'яти мов програмування, визнаних стандартом IEC 61131-3 для програмування ПЛК

# 3 РОЗРОБЛЕННЯ

# Розробка проекту для CoDeSys

# 3.1.1 Загальні вимоги

Створення проекту в середовищі CODESYS починається зі створення нового проекту та додавання потрібних бібліотек до нього. У проекті для нашого робота були додані такі бібліотеки, як Standart, Util, VisuDialogs.

Після додавання бібліотек необхідно було додати файл Simulation типу ST-POU для коду, файл Glob\_Var для глобальних змінних, 2 файли для візуалізації

та файл для вистежування змінних.

# 3.1.2 Кодування програми

За допомогою ST-POU було розроблено код програми для управління теплообміном там було зчитано усі данні котрі вводились у систему за допомогою панелі у візуалізації налаштувань. Після цього через декілька перевірок даних включається котел та починається нагрів, також реалізовано остигання після передачі тепла у теплообміннику, або через стіни до зовнішнього простору.

# 3.1.3 Проектування візуалізації

За допомогою CoDeSys було зроблено код для систем теплообміну[1], а вже на його основі розробляється візуалізація основної програми котра має такі елементи:

* 4 які працюватимуть як датчики температури
* 2 котрі будуть теплообмінниками
* 1 працюючий як котел
* 2 як датчики потоку води
* 1 як кран
* 2 як клапани
* 2 як насос

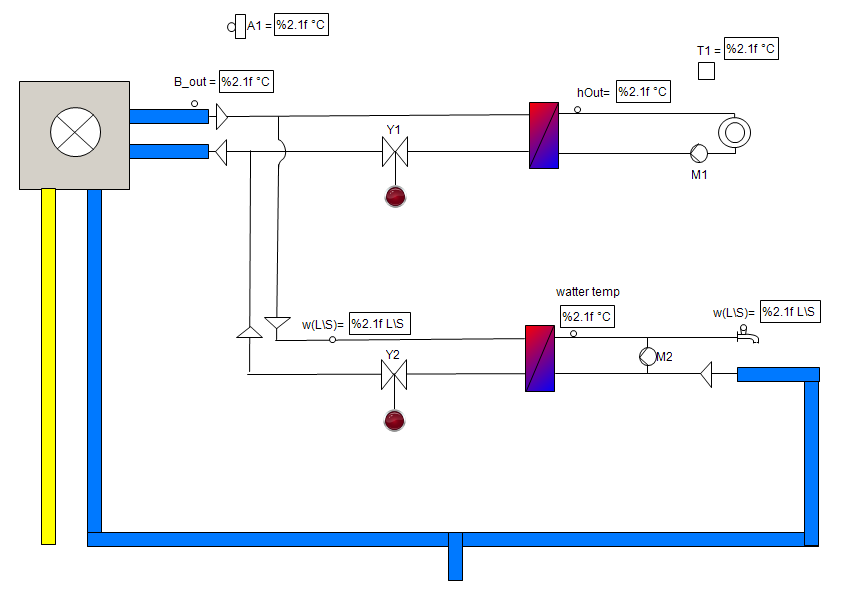


Рисунок 3.1 Візуалізація основної програми

А також візуалізація програми налаштування котра має такі елементи:

* 4 потенціометра котрі працюють як задання температур та потоку води
* 3 перемикача котрі потрібні за для включення подачі води до одного з контурів або включення котла
* 3 індикатори котрі відображають стан перемикачів
* 1 круговий індикатор для температури у котлі
* 5 панелі на які виводяться данні з температур у кімнаті, температури води, температури у котлі, максимальної температури у котлі, напору води



Рисунок 3.2 Візуалізація програми налаштування

Ще було добавлено вікно у котрому відображається уся статистика за час праці.

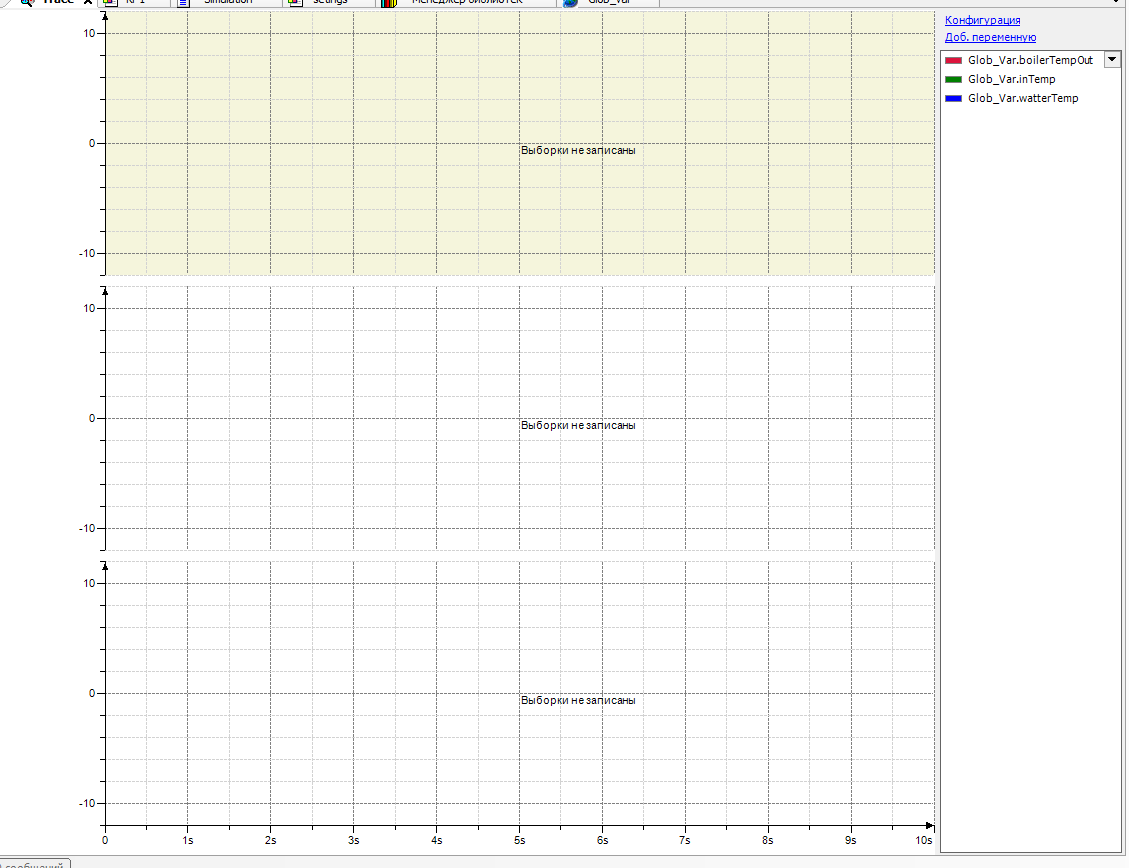


Рисунок 3.3 Візуалізація статистики

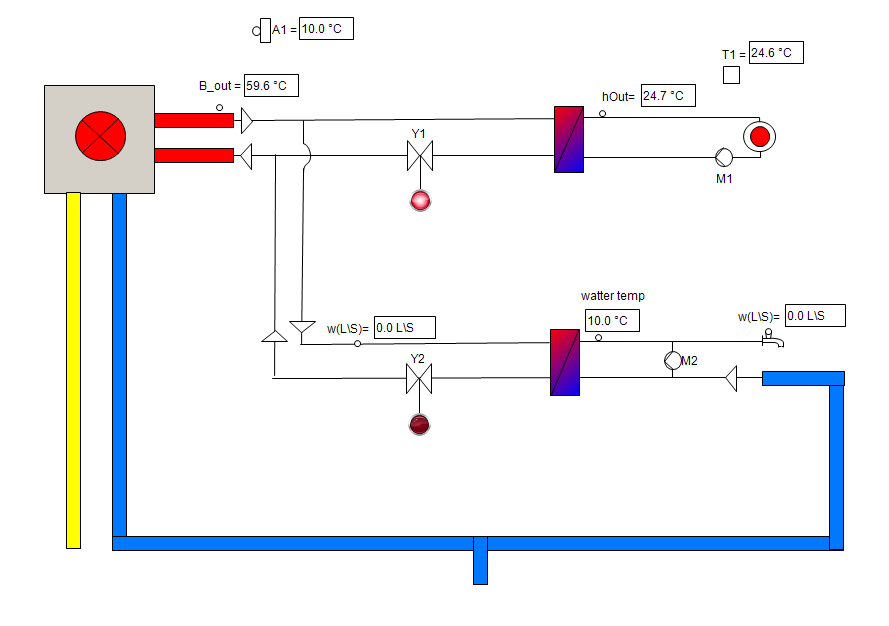
# 4 ТЕСТУВАННЯ

# Тестування візуалізації

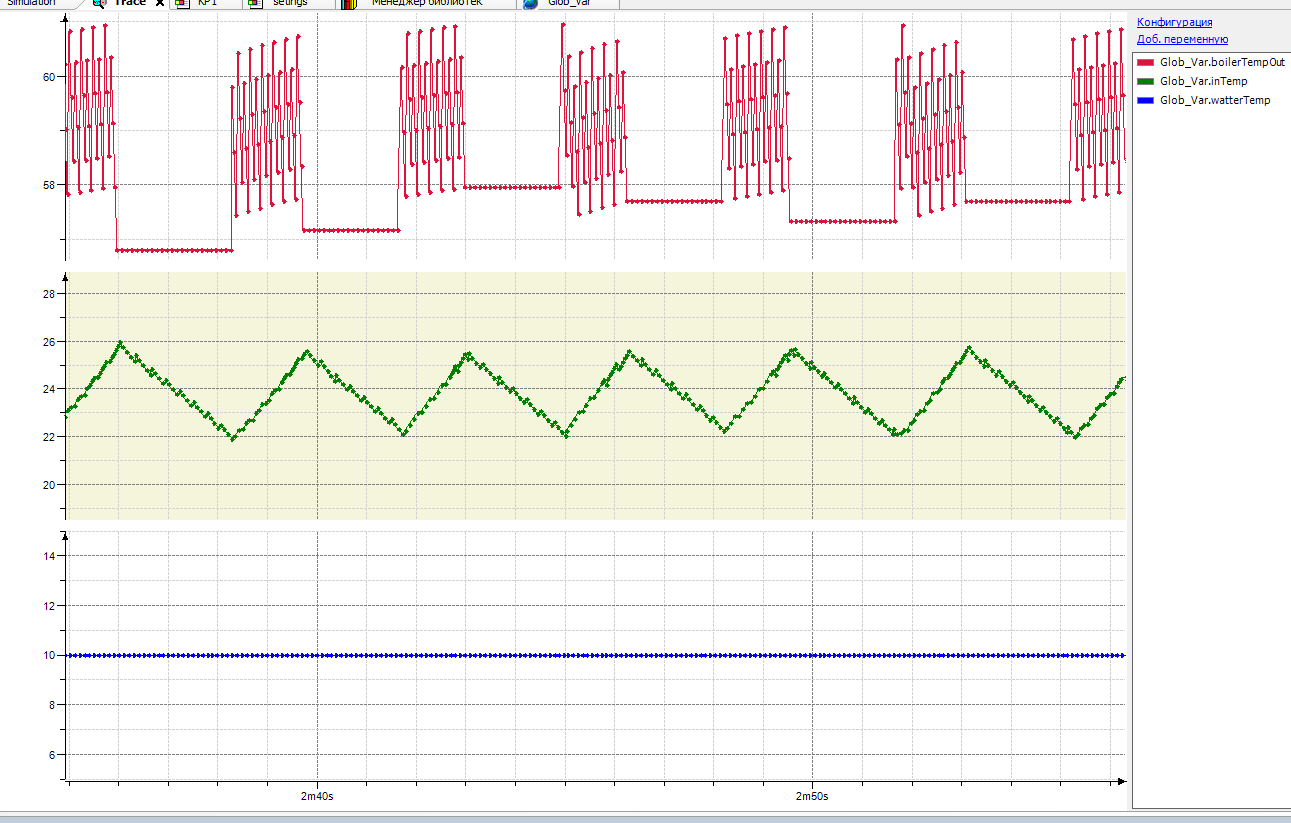
Спочатку проведем тестування опалення:



З початку вмикаємо опалення та задаємо необхідну температуру у кімнаті після чого задаємо максимальну температуру у котлі та система починає працювати. При праці системи є зміни у візуалізації.



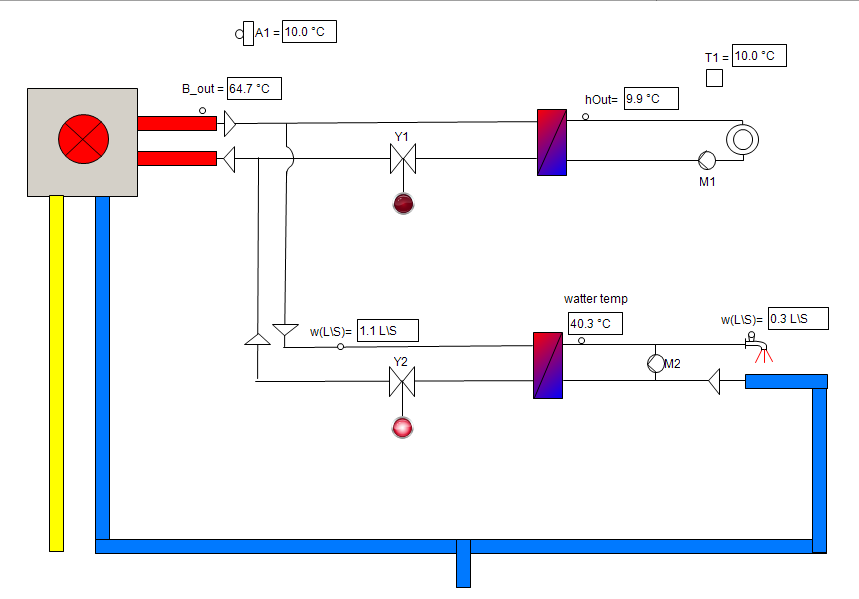
А також записується статистика.



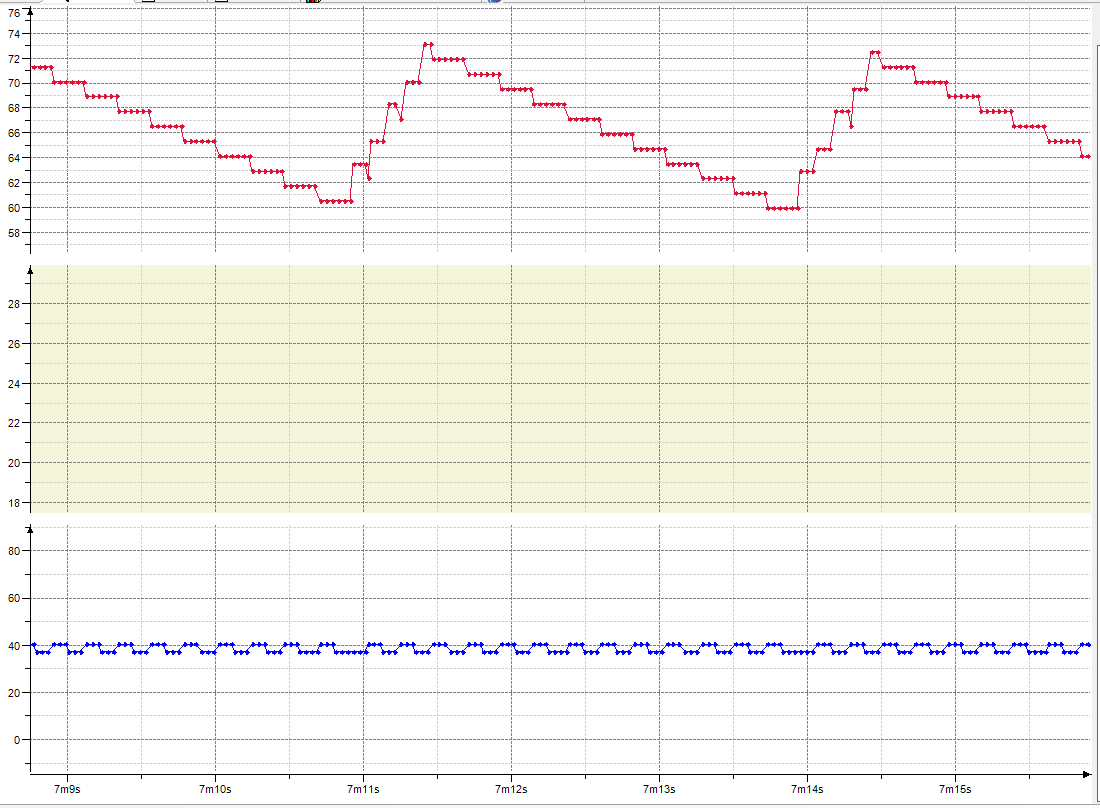
Після перевірки опалення, перевіримо водопостачання.



Тут нам необхідно вже окрім температур, як і у першому разі, ще задати натиск води у крані.



Ось зміна у візуалізації.



Та статистика зміни температур.

# ВИСНОВКИ

У цьому проекті я навчився працювати з програмою CoDeSys. Я ознайомився з принципами планування і розробки промислових систем, відповідно до вимог та потреб замовника. Також було важливим знання з області термодинаміки та схемотехніки, що дозволило мені розуміти роботу окремих компонентів проекту.

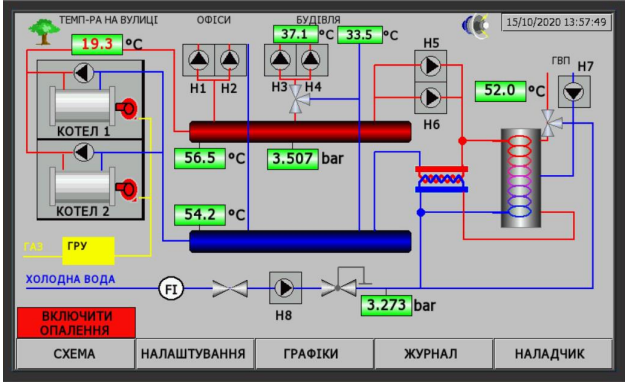
# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

* ChatGPT 3.5. <https://chat.openai.com/>
* CODESY GROUP. (2022). *CODESYS ENGINEERING AND VISUALISATION*. (CODESYS GmbH) Отримано з <https://www.codesys.com/>
* codesysitalia.. Отримано з <https://www.youtube.com/@codesysitalia>
* Документація для роботи бралась з сайту для Контролеру Опалення та ГВП Отримано з <https://energopribor.zp.ua/siemens_rvd140.html>

# ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ.

На підставі завдання потрібно було зробити програму управління теплообмінником та візуалізувати роботу.

Після чого був надано приклад основного екрану де було реалізовано систему з опаленням та ГВП.



Потрібно було на основі даних з датчиків та регуляторів зробити автоматичне опалення та нагрів води й подачу до ГВП.

# ДОДАТОК Б. ВИХІДНІ КОДИ.

Код задання глобальних змінних:

VAR\_GLOBAL

//Inputs

heating: BOOL := FALSE;

watter: BOOL := FALSE;

outTemp: REAL:=10.0000;

inTemp: REAL:=10.0000;

inTempSet: REAL:=10.0000;

watterTempIN: REAL:=10.0000;

watterTemp: REAL:=10.0000;

watterTempSet: REAL:=10.0000;

heatTempOut: REAL:=10.0000;

boilerTempOut: REAL:=10.0000;

boilerMaxTemp: REAL:=100.0000;

Hysteresis: REAL:= 10.0000;

IntempHysteresis: REAL:= 3.0000;

wattersteamIn: REAL:= 0.0000;

wattersteamOut: REAL:= 0.0000;

//Outputs

heatingTap: BOOL := FALSE;

watterTap: BOOL := FALSE;

boiler: BOOL := FALSE;

END\_VAR

Код основної програми та локальні змінні:

PROGRAM Simulation

VAR

TempIncriment: TIME:=T#2MS;

TempExchange: TIME:=T#2MS;

WatterTempExchange: TIME:=T#1S;

BoilerTempIncreased: TIME:=T#100MS;

WatterTempIncreased: TIME:=T#100MS;

WatterTempDecreased: TIME:=T#100MS;

BoilerTempDecreased: TIME:=T#100MS;

Timer1: TON;

Timer2: TON;

Timer3: TON;

Timer4: TON;

Timer5: TON;

Timer6: TON;

Timer7: TON;

Timer8: TON;

xRaiseTemp: BOOL;//Signal for increasing the temperature

xDecriaseTemp: BOOL;//Signal for increasing the temperature

xRaiseTempinheater: BOOL;

xRaiseTempinroomfromout: BOOL;

xDicraiseTempinroomfromout: BOOL;

xRaiseTempinroom: BOOL;

xRaiseWatterTemp: BOOL;

xIncreaseWatterTemp: BOOL;

xDecreaseWatterTemp: BOOL;

xDecreasedBoilerTemp: BOOL;

simulation: BOOL := FALSE;

END\_VAR

simulation := TRUE;

IF Glob\_Var.heating THEN

Glob\_Var.watter := FALSE;

Glob\_Var.heatingTap := TRUE;

Glob\_Var.watterTap := FALSE;

IF Glob\_Var.heatingTap THEN

IF Glob\_Var.inTempSet > Glob\_Var.inTemp + Glob\_Var.IntempHysteresis THEN

IF Glob\_Var.boilerTempOut + Glob\_Var.Hysteresis <= Glob\_Var.boilerMaxTemp THEN

Glob\_Var.boiler := TRUE;

END\_IF

END\_IF

END\_IF

END\_IF

IF Glob\_Var.inTempSet < Glob\_Var.inTemp THEN

Glob\_Var.boiler := FALSE;

END\_IF

IF Glob\_Var.watter THEN

Glob\_Var.heating := FALSE;

Glob\_Var.watterTap := TRUE;

Glob\_Var.heatingTap := FALSE;

IF Glob\_Var.watterTap THEN

IF Glob\_Var.watterTempSet > Glob\_Var.watterTemp THEN

IF Glob\_Var.boilerTempOut + Glob\_Var.Hysteresis <= Glob\_Var.boilerMaxTemp THEN

Glob\_Var.boiler := TRUE;

END\_IF

END\_IF

END\_IF

END\_IF

IF Glob\_Var.heating = FALSE THEN

IF Glob\_Var.watter = FALSE THEN

Glob\_Var.watterTap := FALSE;

Glob\_Var.heatingTap := FALSE;

Glob\_Var.boiler := FALSE;

END\_IF

END\_IF

IF Glob\_Var.inTemp < Glob\_Var.outTemp THEN

Timer4(IN:= TRUE, PT:= TempIncriment\*5, Q=>xRaiseTempinroomfromout);

IF xRaiseTempinroomfromout THEN

Glob\_Var.inTemp := Glob\_Var.inTemp + (Glob\_Var.outTemp \* 0.01);

Timer4(in:= FALSE);

END\_IF

END\_IF

IF Glob\_Var.inTemp > Glob\_Var.outTemp THEN

Timer4(IN:= TRUE, PT:= TempIncriment, Q=>xRaiseTempinroomfromout);

IF xRaiseTempinroomfromout THEN

Glob\_Var.inTemp := Glob\_Var.inTemp - (Glob\_Var.outTemp \* 0.01);

Timer4(in:= FALSE);

END\_IF

END\_IF

IF Glob\_Var.heatTempOut > Glob\_Var.inTemp THEN

Timer3(IN:= TRUE, PT:= TempIncriment, Q=>xRaiseTempinroom);

IF xRaiseTempinroom THEN

Glob\_Var.inTemp := Glob\_Var.inTemp + (Glob\_Var.heatTempOut \* 0.01);

Glob\_Var.heatTempOut := Glob\_Var.heatTempOut - (Glob\_Var.heatTempOut\* 0.02);

Timer3(in:= FALSE);

END\_IF

END\_IF

IF Glob\_Var.heatingTap THEN

IF glob\_var.boilerTempOut >= glob\_var.boilerMaxTemp - glob\_var.Hysteresis THEN

IF Glob\_Var.boilerTempOut > Glob\_Var.heatTempOut THEN

Timer2(IN:= TRUE, PT:= TempExchange, Q=>xRaiseTempinheater);

IF xRaiseTempinheater THEN

Glob\_Var.heatTempOut := Glob\_Var.heatTempOut + (Glob\_Var.boilerTempOut \* 0.01);

Glob\_Var.boilerTempOut := Glob\_Var.boilerTempOut - (Glob\_Var.boilerTempOut \* 0.02);

Timer2(in:= FALSE);

END\_IF

END\_IF

END\_IF

END\_IF

IF Glob\_Var.boiler THEN

Timer1(IN:= TRUE, PT:= BoilerTempIncreased, Q=>xRaiseTemp);

IF xRaiseTemp THEN

Glob\_Var.boilerTempOut := Glob\_Var.boilerTempOut + 3;

Timer1(in:= FALSE);

END\_IF

END\_IF

IF Glob\_Var.boilerMaxTemp <= Glob\_Var.boilerTempOut THEN

Glob\_Var.boiler := FALSE;

END\_IF

IF Glob\_Var.watterTap THEN

IF Glob\_Var.wattersteamOut > 0.0000 THEN

IF glob\_var.boilerTempOut >= glob\_var.boilerMaxTemp - glob\_var.Hysteresis THEN

IF glob\_var.boilerTempOut > glob\_var.watterTemp THEN

IF glob\_var.watterTemp < Glob\_var.watterTempSet THEN

Timer6(IN:= TRUE, PT:= WatterTempIncreased, Q=>xIncreaseWatterTemp);

IF xIncreaseWatterTemp THEN

Glob\_var.wattersteamIn := Glob\_var.wattersteamOut \* (Glob\_var.watterTempSet/12);

Glob\_var.watterTemp := Glob\_var.watterTemp + ((Glob\_var.watterTempSet - Glob\_var.watterTempIN) / 10);

glob\_var.boilerTempOut := glob\_var.boilerTempOut -1.2;

Timer6(in:= FALSE);

END\_IF

END\_IF

END\_IF

END\_IF

END\_IF

END\_IF

IF Glob\_Var.watterTemp >= Glob\_Var.watterTempSet THEN

IF Glob\_Var.wattersteamOut > 0.0000 THEN

Timer7(IN:= TRUE, PT:= WatterTempDecreased, Q=>xDecreaseWatterTemp);

IF xDecreaseWatterTemp THEN

Glob\_var.watterTemp := Glob\_var.watterTemp - ((Glob\_var.watterTempSet - Glob\_var.watterTempIN) / 10);

Timer7(in:= FALSE);

END\_IF

END\_IF

END\_IF

IF Glob\_Var.watterTemp >= Glob\_Var.watterTempIN AND Glob\_Var.watterTap = FALSE THEN

Timer7(IN:= TRUE, PT:= WatterTempDecreased, Q=>xDecreaseWatterTemp);

IF xDecreaseWatterTemp THEN

Glob\_var.watterTemp := Glob\_var.watterTemp - ((Glob\_var.watterTempSet - Glob\_var.watterTempIN) / 10);

Timer7(in:= FALSE);

END\_IF

END\_IF

IF Glob\_Var.watterTap = FALSE AND Glob\_Var.heatingTap = FALSE THEN

Glob\_Var.boiler := FALSE;

IF Glob\_Var.boilerTempOut > Glob\_Var.outTemp THEN

Timer8(IN:= TRUE, PT:= BoilerTempDecreased, Q=>xDecreasedBoilerTemp);

IF xDecreasedBoilerTemp THEN

Glob\_Var.boilerTempOut := Glob\_Var.boilerTempOut - (Glob\_Var.outTemp \* 0.01);

Timer8(in:= FALSE);

END\_IF

END\_IF

END\_IF

# ДОДАТОК В. НАСТАНОВА КОРИСТУВАЧУ.

# 1 Відомість

Ця програма зроблена для ознайомлення з автоматизацією у опалені та ГВП для цього збирається інформація з датчиків та налаштування і після цього програма починає працювати.

2 Умови необхідні для використання :

# Апаратні засоби :

* Комп'ютер з операційною системою, сумісною з вимогами CoDeSys.

## Програмні засоби:

* Остання версія CoDeSys, яку необхідно завантажити та встановити на комп'ютер.

## Документація та ресурси:

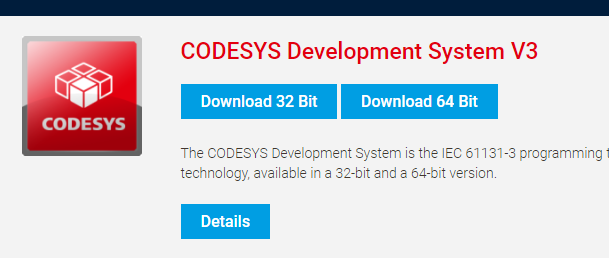
* Документація, що пояснює використання CoDeSys та Arduino, включаючи посібники, інструкції та приклади.
* Ресурси Інтернету для отримання додаткової інформації, підтримки та відповідей на можливі питання.

# Послідовність дій користувача

## Інсталяція CoDeSys

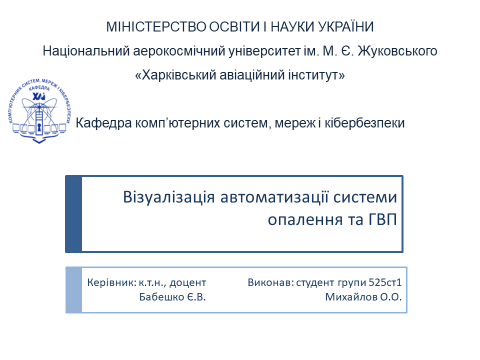
Інструкції щодо встановлення та налаштування CoDeSys:

1. Завантажте останню версію CoDeSys з офіційного веб-сайту виробника.

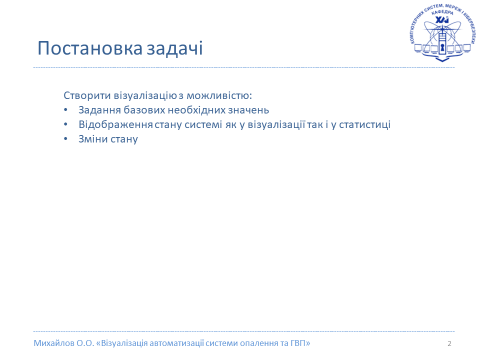


1. Встановіть CoDeSys на ваш комп'ютер.
2. Налаштуйте підключення за допомогою програми PLC Control котра дає змогу програмі під єднатись до комп’ютеру.

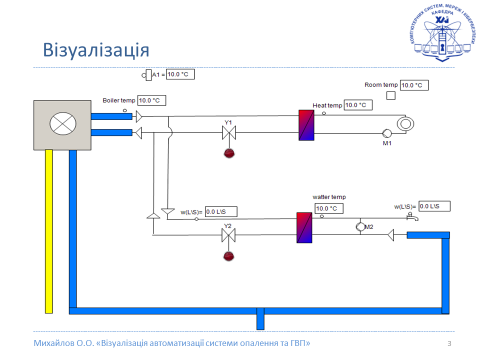
# ДОДАТОК Г. ПРЕЗЕНТАЦІЯ.



Слайд 1



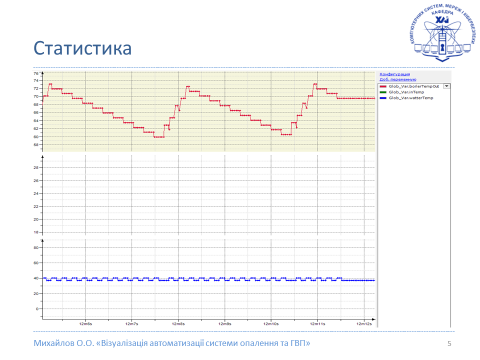
Слайд 2



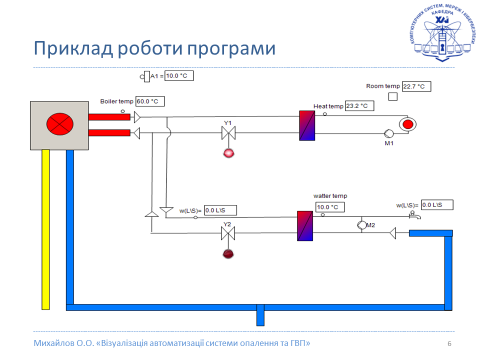
Слайд 3



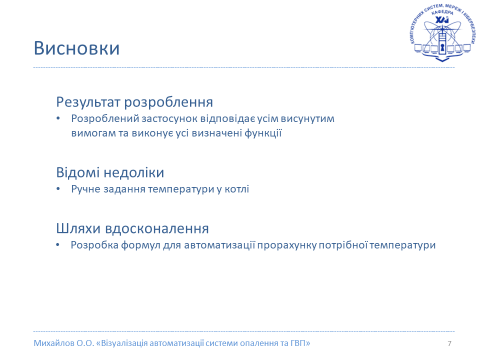
Слайд 4



Слайд 5



Слайд 6



Слайд 7