

**Course Project Report**

Mixing and Heating of Liquid Components

Discipline High-level Design of Information and Control Systems

|  |  |
| --- | --- |
| Prepared by | Sergei Ilin |
| Submitted to | Professor Vladimir N. Khokhlovsky |
|  | May 17, 2023 |

St. Petersburg

2022

Contents

[1. Task. 3](#_Toc135220508)

[2. High-level description of the task by means of the UML language. 4](#_Toc135220509)

[2.1. Use Case Diagram (UML). 4](#_Toc135220510)

[2.2. Component (Class) Diagram (UML). 5](#_Toc135220511)

[2.3. Activity Diagram (UML). 5](#_Toc135220512)

[2.4. State Machine Diagram (UML). 6](#_Toc135220513)

[3. Description of state diagrams using CODESYS programming languages. 8](#_Toc135220514)

[3.1. The Control System by means of the SFC language (CODESYS). 8](#_Toc135220515)

[3.2. The Object of Control by means of the SFC language (CODESYS). 10](#_Toc135220516)

[4. Visualization. 13](#_Toc135220517)

[5. Conclusion. 13](#_Toc135220518)

[6. Used materials. 14](#_Toc135220519)

[Appendixes 15](#_Toc135220520)

# Task.

Mixing and Heating of Liquid Components.

**Purpose:**

The purpose of this task is to control the mixing and heating process of liquid components in a technological installation. The installation consists of two identical tanks for mixing the liquid components. Additionally, there is a receiving tank with a smaller volume than the combined volume of the two mixing tanks.

**Composition:**

2 mixing tanks

**Principle:**

The principle of the mixing and heating process of liquid components involves controlling the flow and temperature of the components to achieve a desired mixture. The process follows a series of steps:

Start: The process begins by pressing the START button, which initiates the sequence.

Component Mixing: The first product's valve opens, allowing the liquid to flow into the mixing tank. Once the average level is reached in the first product tank, the valve closes, and the valve for the second product opens. This ensures proper mixing of the components.

Level Monitoring: The level of the liquid components is continuously monitored using level sensors. If the upper level signal is triggered, indicating that the maximum level in the second product tank is reached, the valve for the second product closes.

Heating: After the valve for the second product closes, the steam valve opens at 100%. The steam enters the tank and heats the mixture. The temperature is monitored using a temperature sensor. Once the temperature reaches 95 degrees Celsius, the steam valve is adjusted to remain open at 20% for an additional 10 seconds to maintain the desired temperature.

Holding Period: After the heating process, a holding period is observed to allow for any necessary reactions or settling. During this time, the liquid remains in the tank.

Draining: The liquid is drained from the tank through a drain valve. The drain valve is closed if the lower level signal is deactivated, indicating that the liquid level is below the threshold.

Cycle and Stop: The entire process repeats unless the STOP button is pressed. If the STOP button is pressed, the drain valve is closed, and the process comes to a halt.

The principle relies on the accurate control of valves, level sensors, temperature sensors, and the coordination of their actions through a programmable logic controller (PLC). The level and temperature data are continuously monitored, and appropriate actions are taken to maintain the desired mixing and heating conditions.

# High-level description of the task by means of the UML language.

UML (Unified Modeling Language) is a visual representation language used for illustrating object-oriented concepts in software development, system design, business process modeling, and organizational structure visualization.

In the context of this project, UML serves three key objectives:

* It provides a comprehensive description of the core functionalities offered by the system.
* It presents a clear depiction of the system's structure, including its principal classes and components.
* It captures the dynamic behavior and interactions of the system, showcasing how its various elements operate and respond to stimuli.

# Use Case Diagram (UML).

Use Case Diagram describes the general functionality of the system (Figure 1). Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

*Figure 1.*

The diagram shows the actors and their functions.

# Component (Class) Diagram (UML).

The diagram illustrates the system's structure and the controlled object (Figure 2). It visually represents the flow of information between the control object's sensors and the program controller through input and output signals. Additionally, it showcases the transmission of commands from the controller to the control object's elements.

From the Object of Control, the Control System receives information about the temperature and current level of liquids in both tanks, as well as information about the states of the all valves including steaming valve.

Depending on the temperature and liquid value, the control System sends commands to the Control Object to turn on/off valves. Изображение выглядит как текст, диаграмма, снимок экрана, План

Автоматически созданное описание

*Figure 2.*

# Activity Diagram (UML).

The Activity Diagram depicts (Figure 3) the process of mixing and heating liquid components in the mixing tank. It begins with the START button press, opening Valve 1. The system checks until the average level is reached. Then, Valve 1 closes, Valve 2 opens, and the upper level is monitored. If the upper level is reached, Valve 2 closes, the steam valve opens fully, and the temperature is checked. Once the temperature reaches 95 degrees Celsius, the steam valve remains partially open for 10 seconds before draining the liquid. If the lower level signal is detected and the STOP button is not pressed, the drain valve closes, returning to the idle state until the next cycle begins with the START signal.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание*Figure 3.*

# State Machine Diagram (UML).

The State Machine Diagram illustrates the sequential process of mixing and heating liquid components in the mixing tank. It consists of the following states and transitions (Figure 4):

Idle State: Represents the initial state of the system, waiting for the START signal.

Filling L1 State: The tank fills with the first liquid component when Valve 1 opens.

Filled L1 State: Indicates that the tank is filled with the first liquid component after reaching the average level. Valve 1 closes.

Filling L2 State: Valve 2 opens, and the second liquid component begins filling the tank.

Filled L2 State: Signifies that the tank is filled with the second liquid component after reaching the upper level. Valve 2 closes.

Heating State: The tank contents are heated as the steam valve opens fully.

After Heating State: Once the temperature reaches 95 degrees Celsius, the steam valve remains partially open for 10 seconds to stabilize the contents.

Checking CL State: The system checks for the lower level signal.

Draining State: The drain valve opens, and the tank contents are drained.

Error State: Represents any error or abnormal condition that may occur during the process.

These states and transitions enable effective control of the mixing and heating process, ensuring proper mixing and temperature regulation in the mixing tank. Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

*Figure 4.*

# Description of state diagrams using CODESYS programming languages.

To ensure the transition of the display of State Diagrams from UML to their description by means of programming languages, the CODESYS industrial automation software package is used.

Five languages defined by the IEC 61131-3 standard are available for programming in CODESYS. For the purposes of this Project, two of them will be used to varying degrees:

* SFC - Sequential Function Chart;
* ST - Structured Text.

# The Control System by means of the SFC language (CODESYS).

The Control System in this project utilizes the CODESYS programming environment, specifically the Structured Text (ST) language and Function Blocks (FB) to implement the control logic. The implemented part of the system, which is written in CODESYS (Figure 5).

The control logic is organized using Function Blocks, which allow for the creation and inheritance of classes. Each class represents a specific component or functionality of the control system and may include methods and state machines (Figure 5).

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

*Figure 5*

By utilizing CODESYS, the Control System can effectively manage and coordinate the different components and processes involved in the project.

Using Functional blocks to organized control logic with OOP principles let us create programs (in CODESYS terminology PLC\_PRG) by calling ready to use functional blocks, that also allow us to create any quantity of tanks by creating instances (Figure 6).

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

*Figure 6*

# The Object of Control by means of the SFC language (CODESYS).

The Control Object in this project refers to the physical entity or process that is being controlled and monitored by the Control System. In the context of the project, the Control Object is a mixing tank used for blending and heating liquid components.

In CODESYS, the Control Object is created using Function Blocks (FBs) and a composition or aggregation approach. The mixing tank is defined as a Function Block called "MixinTank" which extends the "Tank" FB and includes additional components (also FB) such as valves, level sensors, temperature detectors, and a steam valve.

The "MixinTank" FB is instantiated within the Control System program, and the necessary connections are made between the FB and its components. The valves, level sensors, temperature detectors, and steam valve are either declared as input or output variables of the "MixinTank" FB or as separate FB instances that are composed or aggregated within the "MixinTank" FB.

By using composition or aggregation, the Control Object is built by combining multiple FB instances that represent the various components of the mixing tank. This approach allows for modular and reusable code, as well as the ability to configure and customize the Control Object based on specific project requirements.

The Control Object's behavior and functionality are defined through the methods and state machines implemented within the "MixinTank" FB. These methods and state machines handle tasks such as opening and closing valves, monitoring and controlling the level of liquid components, detecting and maintaining the desired temperature, and managing the drainage process.

The Control Object created in CODESYS provides an organized and structured approach to control and monitor the mixing tank. It encapsulates the necessary components and logic required for precise control and efficient operation of the mixing and heating processes.

The State Machines Diagram mentioned in Figure 4 has been implemented with the help of the SFC language (Figure 7). The SFC blocks specify the conditions that are met at the input. These conditions are written in the ST language (Figure 8).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание*

*Figure 7*

The SFC blocks specify the conditions that are met at the input. These conditions are written in the ST language (Figure 8).

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

*Figure 8*

# Visualization.

The human-machine interface is implemented by displaying controls, light indicators and temperature sensor indicators on the dashboard.

Figure 9 shows the control panelИзображение выглядит как снимок экрана, текст

Автоматически созданное описание

*Figure 9.*

# Conclusion.

The project successfully implemented the creation and demonstration of a model for the system of mixing and heating of liquid components. The development process began with a high-level description of the system's main functions and elements. This was followed by a detailed exploration of the interactions between the system elements and the conditions governing their state transitions. The Object of Control and Control System were defined, and inputs and outputs were identified as well.

To simulate the system, programming languages in CODESYS were utilized to describe the behavior of the Object of Control and the Control System. The languages employed included Sequential Function Chart (SFC) and Structured Text (ST). Additionally, a visual representation of the mixing system’s operation was implemented using the Visualization component.

Furthermore, the project involved the modeling, which was realized through the creation of UML diagrams in the Modelio environment. The State Diagram played a significant role in developing an imitation of the control system and the object of control within CODESYS.

Overall, the project provided valuable insights into high-level design principles and the practical application of programming languages in a real-world scenario. The visualization component enhanced the descriptive capabilities, resulting in an illustrative representation of a functioning coal mine.

# Used materials.

* Lectures of the course «High-level Design of Information and Control Systems», 02.2023-05.2023, lecturer Professor Vladimir N. Khokhlovsky;
* Modelio Tutorials, ([Tutorials (modelio.org)](https://www.modelio.org/documentation-menu/tutorials.html));
* «User's Guide to PLC programming in CoDeSys», by 3S - Smart Software Solutions GmbH;
* «Codesys: Sequential Function Chart (SFC) programming», by Tohid Alizadeh ([Tohid Alizadeh - YouTube](https://www.youtube.com/channel/UC_qsTO2Z8tAt3Vx4P2SdH6w));
* V.V.Khokhlovskiy, Formal description of objects’ and systems’ design with UML, 2018

**Appendixes**

**Anylogic modeling**

1. **Market Model**

Изображение выглядит как текст, снимок экрана, число, График

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, диаграмма, Красочность

Автоматически созданное описание

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

1. **Epidemic**

Изображение выглядит как текст, диаграмма, линия, карта

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Изображение выглядит как снимок экрана, текст, программное обеспечение

Автоматически созданное описаниеИзображение выглядит как диаграмма, зарисовка, рисунок, линия

Автоматически созданное описаниеИзображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, диаграмма, число

Автоматически созданное описание

**3. Storage room**

**Изображение выглядит как снимок экрана, текст, Прямоугольник, программное обеспечение

Автоматически созданное описание**

Изображение выглядит как снимок экрана, Прямоугольник, текст

Автоматически созданное описание