

Програмування ПЛК систем локальної автоматизації. Програмування САР рівня рідини та температури продукту.

10.1. Стабілізаційне регулювання рівня рідини

Завдання.

Створити проект для ПЛК М340, для реалізації програми управління установкою, що описується наступним алгоритмом (рис.10.1). Після нажимання кнопки "СТАРТ" відкривається клапан набору першого продукту. Після досягнення середнього рівня клапан 1-го продукту закривається, відкривається клапан набору 2-го продукту. Після спрацювання сигналізатору верхнього рівня закривається клапан набору 2-го продукту, відкривається клапан пари на 100% (діапазон виходу 0-100%). Після досягнення температури 95°C (діапазон датчику 0-150°C) включається етап витримки. Витримка повинна тривати 3 хвилини, в цей час регулятор повинен підтримувати температуру на заданому рівні. Задане значення температури визначає оператор. Програму перевірити та налагодити з використанням операторських екранів.

Після закінчення витримки, рідина зливається з апарату. Після відключення сигналізатору нижнього рівня, цикл повторюється у випадку якщо кнопка СТОП не нажата. Якщо СТОП нажата – клапан зливу закривається. У ПЛК поступає сигнал від датчика рівня з діапазоном вимірювання 0-5 м.

Рішення.

Модифікована частина програми показана на рис.10.2, перелік змінних та екземплярів функціональних блоків наведений на рис.10.3 та рис.10.4.

Змінна що відповідає за управління виконавчим механізмом *VR_par_R* управляється регулятором *TIC1*, який реалізовує ПІ закон управління. Регулятор викликається кожні 100 мс, оскільки його вхід *EN* підключений до блоку *SMPL1*. Ручне управління клапаном *VR_par_R* забезпечується через функцію *SEL* (".16"), який перемикає *VR_par_R* на значення *VR_par_MAN*. Для безударності переходу замість *SEL* можна використання блок *MS*.

Регулятор *TIC1* працює в режимі регулювання тільки при (*StepProg=4*), в інших випадках він працює в режимі слідкування, тобто на вихід *TIC1.OUT* буде подаватися значення *TIC1.TR_I*. Таким чином на усіх етапах роботи програми *VR_par_R* буде управлятися значенням з *MUX*, тільки на етапі витримки буде включатися режим регулювання.

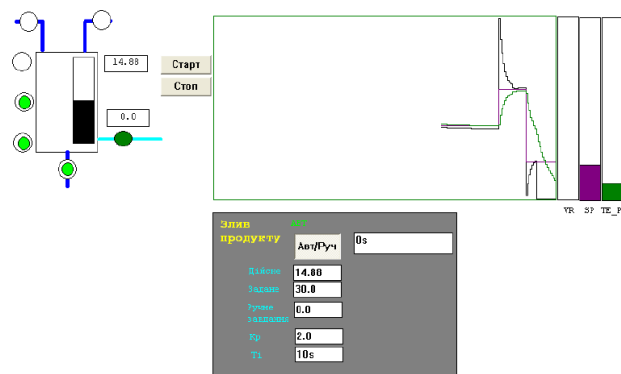


Рисунок 10.1– Приклад операторського екрану до поставленої задачі.

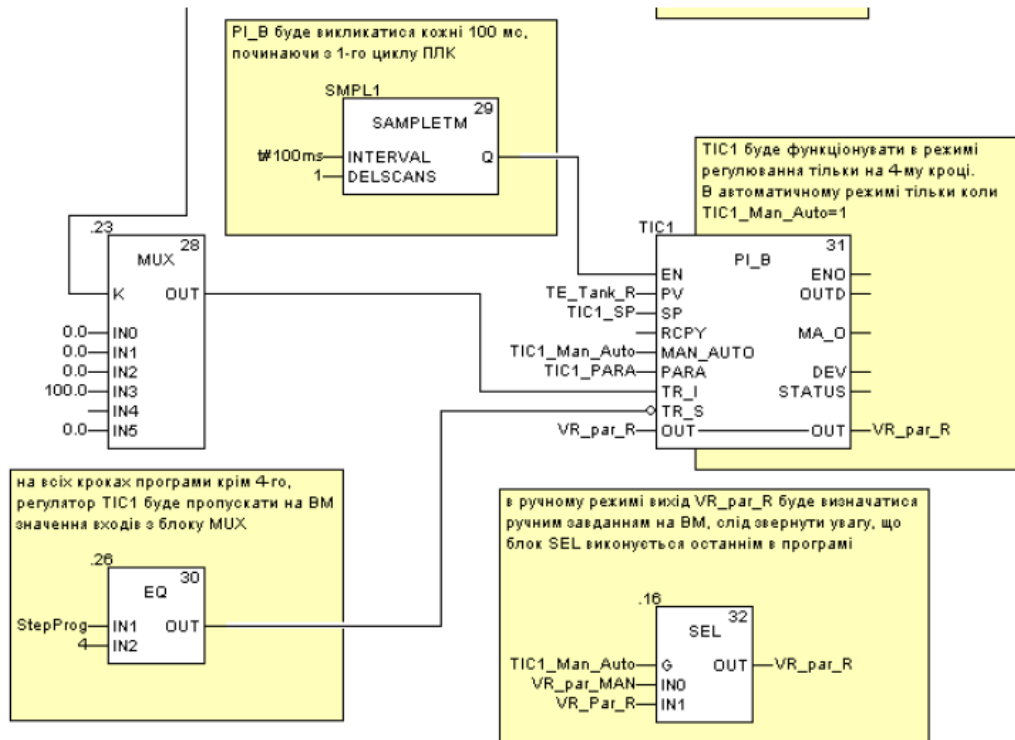


Рисунок 10.2– Модифікована частина секції програми

Name	Type	Addr...	Value	Comment
LE_Tank	INT	%IW0.1.0		Датчик рівня
LE_Tank_R	REAL			Відмасштабований Рівень 0-5 м
LS_nyz	EBOOL	%I0.2.0		Сигналізатор нижнього рівня
LS_ser	EBOOL	%I0.2.1		Сигналізатор середнього рівня
LS_verh	EBOOL	%I0.2.2		Сигналізатор верхнього рівня
SB_Start	BOOL			Кнопка запуску процесу
SB_stop	BOOL			Кнопка зупинки процесу
SCALE_PARA_L	Para_SCALING			
SCALE_PARA_T	Para_SCALING			
StepProg	INT		0	Крок програми
TE_Tank	INT	%IW0.1.1		Датчик температури
TE_Tank_R	REAL			Відмасштабована Температура 0-150 гр.С
TIC1_Man_Auto	EBOOL		1	Режим руч/авт для регулятора температури
TIC1_PARA	Para_PI_B			
id	UINT			
pv_inf	REAL		0.0	обмеження по мінімуму вхідної величини
pv_sup	REAL		150.0	обмеження по максимуму вхідної величини
out_inf	REAL		0.0	обмеження по мінімуму вихідної величини
out_sup	REAL		100.0	обмеження по максимуму вихідної величини
rev_dir	BOOL		false	режим прямої роботи ПІ-регулятора
en_rcpy	BOOL		false	не використовувати RCPY
kp	REAL		1.0	коефіцієнт пропорційності
ti	TIME		t#0s	час інтегрування
dband	REAL		0.2	зона нечутливості
outbias	REAL		50.0	зміщення виходу регулятора в П-режимі
TIC1_SP	REAL		80.0	Задане значення на регулятор температури
VA_nabor1	EBOOL	%Q0.2.16		Клапан набору 1-го продукту
VA_nabor2	EBOOL	%Q0.2.17		Клапан набору 2-го продукту
VA_sliv	EBOOL	%Q0.2.18		Клапан сливу
VR_par	INT	%QW0.1.4		Клапан пари
VR_par_MAN	REAL			Вихід на ВМ в ручному режимі
VR_Par_R	REAL			Відмасштабоване значення ВМ 0-100%

Рисунок 10.3– Перелік змінних.

Name	r	Type	v	Comment
RS_Nabor1		RS		Тригер для управління VA_Nabor1
RS_Nabor2		RS		Тригер для управління VA_Nabor2
RS_Sliv		RS		Тригер для управління VA_Sliv
SCALE_L		SCALING		екземпляр функц блока масштабування для LE_Tank
SCALE_T		SCALING		екземпляр функц блока масштабування для TE_Tank
SMPL1		SAMPLETM		
T_DELAY		TON		екземпляр таймера на витримку
TIC1		PI_B		екземпляр функц болка ПІ-регулятора

Рисунок 10.4– Перелік екземплярів функціональних блоків.

10.2. Каскадне регулювання температури

Завдання.

Необхідно створити проект в UNITY PRO для реалізації поставленої задачі (рис.10.5) з використанням FFB бібліотеки *ControllIB*.

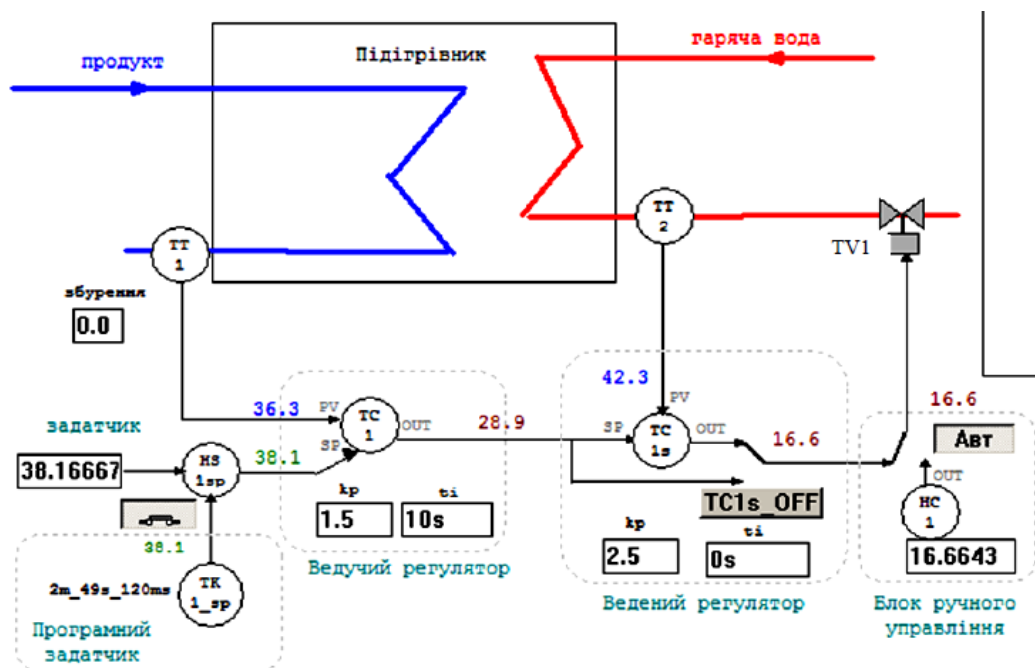


Рисунок 10.5– Операторський екран для контролю та управління процесом нагрівання

Технологічна установка являє собою теплообмінник для підігріву рідкого продукту (рис.10.5) за допомогою гарячої води. Температура продукту на 4 виході вимірюється датчиком температури *TT1* (0-100°C, вхід ПЛК *%IW0.1.2*) який вмонтований на трубопроводі на відстані кількох метрів від підігрівника, а регулюється витратою гарячої води регулюючим клапаном *TV1* на виході підігрівника (0-100% ХРО, вихід *%QW0.2.0*). Додатково вимірюється також температура води в трубопроводі безпосередньо на виході підігрівника за допомогою датчика *TT2* (0-100°C, вхід ПЛК *%IW0.1.1*). Інерційність об'єкта по каналу *TT2* менша ніж по *TT1*.

Система управління повинна забезпечити регулювання температури $TT1$ з підтримкою наступних функцій:

- 1) стабілізація температури на виході підігрівника на заданому оператором або програмним задатчиком значенні, з використанням каскадного регулятора, де:
 - a. ведучий регулятор (ПІ режим) стабілізує температуру $TT1$;
 - b. ведений регулятор (ПІ режим) служить для швидкої реакції контуру регулювання на зміну малоінерційного $TT2$;
- 2) можливість настройки коефіцієнтів K_p та T_i з операторського екрану;
- 3) можливість ручного управління виконавчим механізмом $TV1$ з операторського екрану при переключенні в ручний режим з забезпеченням безударності переходу ;
- 4) можливість формування завдання програмним задатчиком в залежності від часу по залежності показаній на діаграмі рис. 10.6.
- 5) можливість переключення з ручного завдання на програмне управління та навпаки в будь який момент часу з забезпеченням безударності переходу;
- 6) можливість виключення веденого регулятора з каскаду (з забезпеченням безударності переходу), тобто переводу контуру в режим прямого ПІ-регулювання по температурі $TT1$;
- 7) при ручному управлінні виконавчим механізмом ведучий регулятор повинен фіксувати своє вихідне значення в останньому положенні.

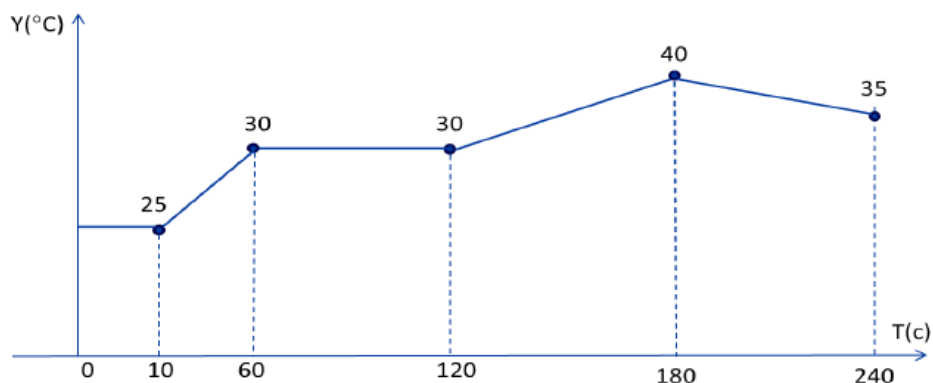


Рисунок 10.6– Залежність завдання температури від часу

Рішення.

Загальні принципи роботи контуру. Функціональна схема контуру регулювання температури продукту показана на операторському екрані (рис.10.5).

Контур включає:

- канали вимірювання ($TT1$, $TT2$);
- канал формування завдання, який складається з ручного задатчика, програмного задатчика $TK1_{sp}$ (формування програмного управління по часу), перемикача ручного/програмного завдання $HS1sp$ (включена кнопка – запуск програмного задатчика) ;
- ведучого регулятора $TC1$ (ПІ закон), який на основі сформованого завдання (вхід SP) та вимірювального значення ($TT1$ на вхід PV), формує сигнал управління (вихід OUT), який подається на ведений регулятор $TC1s$; для регулятора налаштовуються K_p та T_i ;

- веденого регулятора $TC1s$ (П закон), який на основі сформованого завдання ($TC1.OUT$ на вхід SP) та вимірювального значення ($TT2$ на вхід PV), формує сигнал управління (вихід OUT), який подається на виконавчий механізм $TV1$ через перемикач $TC1s_OFF$; перемикач $TC1s_OFF$ дозволяє переключити $TV1$ на вихід ведучого регулятора $TC1.OUT$, тим самим відключивши ведений регулятор з контуру управління; для регулятора налаштовується Kp а також Ti , ненульове значення якого дозволяє перевести регулятор в ПІ-режим;

- блок ручного управління $HC1$, який дає можливість переключитися в режим ручного управління виконавчим механізмом $TV1$ з операторського екрану; кнопка "АВТ" для переключення режиму (нажата - автоматичний режим);

- канал управління ($TV1$);

Перелік змінних та екземплярів функціональних блоків наведений на рис.10.7.

Name	r	Type	\	Comment
HC1		MS		Блок ручного управління
smHeater1		smHeatExch		Імітаційна модель підігрівача
HS1sp		SP_SEL		Перемикач ручного/програмного завдання
smkl1		smValve		Імітаційна модель клапану
smDEL1		smDELAY		Імітаційна модель блоку запізнювання
SMPL1		SAMPLETM		Диспетчер виклику
TC1		PI_B		Ведучий регулятор
TC1s		PI_B		Ведений регулятор
TK1		TON		Таймер програмного задатчика

Name	/	Type	Value	Comment
smInit		BOOL		Ініціалізація імітаційної моделі
TC1_AUTO		BOOL		1 - включити автоматичний режим роботи контуру
TC1s_OFF		BOOL		1 - відключити ведений регулятор
TK1_ON		BOOL		1 - включити програмний задатчик
smZ1		REAL		імітація збурення
TC1_OUT		REAL		Вихід ведучого регулятора
TC1_SP		REAL	20.0	Уставка для ведучого регулятора
TC1s_OUT		REAL		Вихід веденого регулятора
TK1_SP		REAL		Вихід завдання з програмного задатчика
TT1		REAL		T продукту на виході підігрівача
TT2		REAL		T гарячої води на виході підігрівача
TV1		REAL		Клапан подачі гарячої води

Variables DDT Types Function Blocks DFB Types				
Filter				
Name	Type	Value	Comment	
HC1_PARA	Para_MS		Параметри ручного задатчика	
out_min	REAL	0.0		
out_max	REAL	100.0		
inc_rate	REAL			
dec_rate	REAL			
outbias	REAL			
use_bias	BOOL	FALSE		
bumpless	BOOL	FALSE		
TC1_PARA	Para_PI_B		Параметри вдеучого регулятора	
TC1s_PARA	Para_PI_B		Параметри веденого регулятора	
HS1sp_PA...	Para_SP_SEL		Параметри перемикача завдань	
sp_min	REAL	0.0		
sp_max	REAL	100.0		
bump	BOOL			
track	BOOL			
rate	REAL			

Рисунок 10.7– Змінні проекту

Загальні принципи роботи програми. Для реалізації даної задачі використовуються 4-ри секції (рис.10.8): секція "*INPUTS*" – для обробки вхідних каналів вимірювання (датчики температури); секція "*OUTPUTS*" – для обробки вихідного каналу правління; секція "*CTRL1*" – для реалізації контуру регулювання. Секція "*Simulation*" призначена тільки для імітації об'єкта при налагодженні програми і є необов'язковою.

У секції "*INPUTS*" оцифровані значення аналогових входів масштабуються шляхом множення на коефіцієнт (діапазон 0-10000 в 0-100°C). У секції "*OUTPUTS*" аналогічним чином проводиться масштабування для значення виконавчого механізму *TV1*, тільки в зворотному напрямку (діапазон 0-100%XPO в 0-10000).

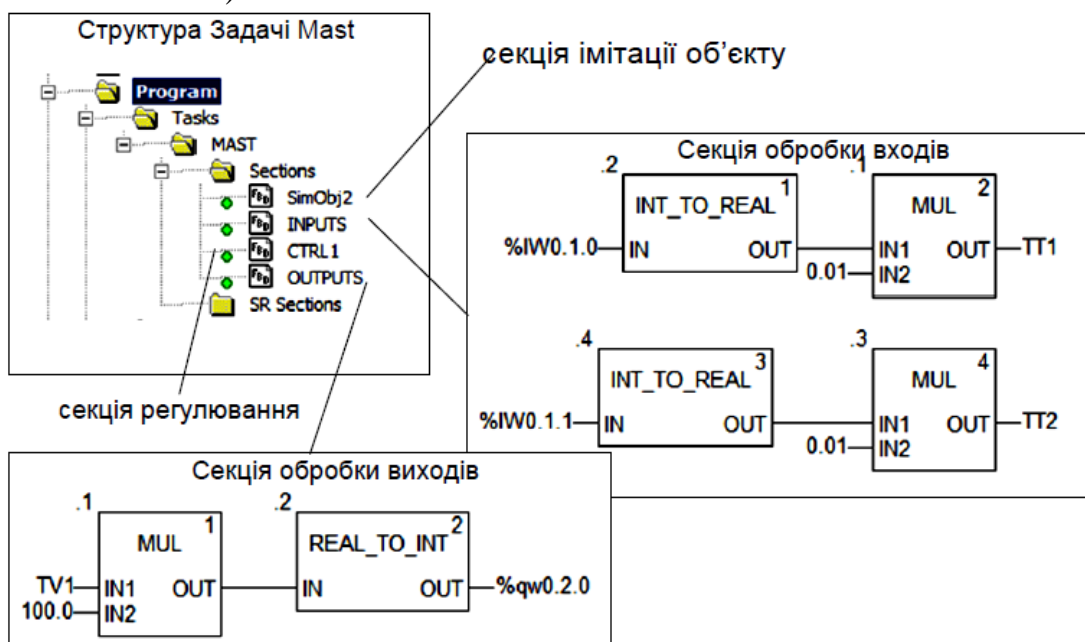


Рисунок 10.8– Структура задачі MAST

Реалізація програмного задатчика. Для реалізації програмного задатчика використана процедура *LOOKUP_TABLE1*, яка реалізовує кусочно-лінійну інтерполяцію за заданими вузловими точками. Вузлові точки задаються парами

значень $XiYi$, перше з яких – час в секундах, друга – задане значення температури в цій точці. Таким чином на вході X процедура буде отримувати значення часу роботи програмного задатчика в форматі *REAL*, в залежності від якого на виході Y буде формуватися задане значення (уставка для регулятора) $TK1_SP$.

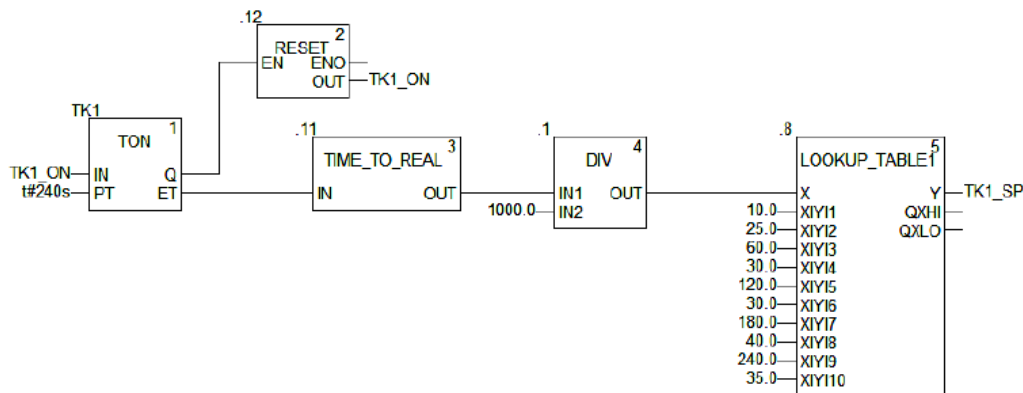


Рисунок 10.9– Реалізація програмного за датчика

Для формування часу роботи задатчика використовується таймер $TK1$, який стартує по команді запуску задатчика $TK1_ON$, та налаштований на повний час його роботи (240с). Спрацювання виходу $TK1.Q$, тобто по закінченню роботи програмного задатчика, приводить до скидання команди $TK1_ON$. У зв'язку з тим, що вихід таймеру $TK1.ET$ видає значення типу *TIME*, воно перетворюється в тип *REAL*, а отримані мілісекунди шляхом ділення на 1000 переводяться в секунди.

Загальні принципи реалізації контуру регулювання. Фрагмент програми реалізації контуру регулювання показана на рис.10.10. Слід зазначити, що старші версії UNITY PRO (<6.0) не підтримують можливість безпосереднього з'єднання в FBD виходів типу *INOUT* та *IN* (або *OUT*).

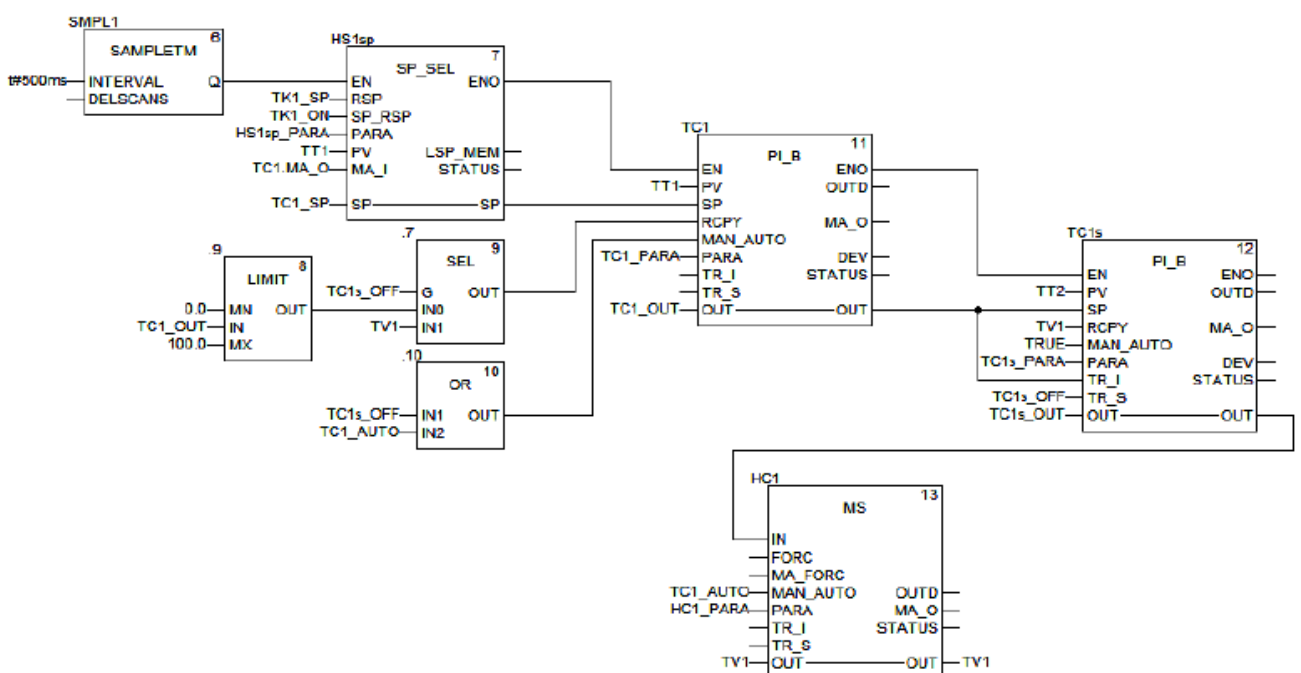


Рисунок 10.10– Фрагмент секції з реалізацією контуру регулювання

(для UNITY PRO ≥ 6.0)

Використовуючи *SMPL1* та входи *EN* функціональних блоків, блоки *HS1sp*(перемикача завдань), *TC1*(ведучий регулятор) та *TC1s*(ведений регулятор) викликаються періодично з часом 500 мс, послідовно один за одним. Це дає змогу зменшити навантаження на ресурси ПЛК та виконавчих механізмів в автоматичному режимі. Однак для ручного режиму необхідна швидка реакція системи на дії оператора, тому блок ручного завдання викликається з кожним циклом (вхід *EN* не задіяний).

Реалізація каналу формування завдання. Вихід блоку *HS1sp.SP* через змінну *TC1_SP* формує завдання для блоку *TC1*, вихід якого в свою чергу формує завдання *TC1s*. Таким чином формується каскадне регулювання.

Блок перемикавання завдання *HS1sp.SP* на вході RSP (дистанційне завдання) отримує значення в змінній *TK1_SP* з програмного задатчика. При активації оператором програмного управління (*TK1_ON=TRUE*), через вхід *SP_RSP* блок *HS1sp* переключиться в дистанційний режим, при якому *HS1sp.SP=HS1sp.TK1_SP*. Це ж значення копіюється в змінну *TC1_SP*, так як вона прив'язана до входу/виходу *SP*. У локальному режимі (*TK1_ON=FALSE*) оператор може змінювати уставку безпосередньо змінюючи *TC1*.

Реалізація каскадного регулятора. Ведучий регулятор *TC1* реалізований через функціональний блок типу *PI_B*. Його задача забезпечити значення температури *TT1* на рівні заданому *TC1_SP*.

Ведений регулятор *TC1s* теж реалізований через функціональний блок типу *PI_B*, однак по замовченню він працює в П-режимі, так як інтегральна складова відключена ($T_i=0$). Його задача швидко зреагувати на зміну менш інерційної змінної процесу *TT2*. Таким чином велику частину збурень швидко компенсує ведений регулятор, а ведучий регулятор забезпечить точність регулювання.

Враховуючи вимоги до функцій контуру регулювання, ведучий регулятор повинен мати можливість працювати в каскаді з веденим регулятором (*TC1s*), або самостійно без нього. У каскадному режимі вихід регулятора підключається до *TC1s.SP* через змінну *TC1_OUT*, тим самим формуючи завдання веденому регулятору. Виведення з каскаду веденого регулятора проводиться шляхом переключення *TC1s* в режим слідкування (*Tracking*), подавши на його вхід *TR_s* змінну *TC1s_OFF* (*TC1s.TR_s=TC1s_OFF*). Тобто в режимі слідкування на вихід *TC1s.OUT* буде подаватися значення *TC1s.TR_I=TC1_OUT*.

Враховуючи вимоги до безударності переходів, та різні режими роботи контуру регулювання, обидва регулятори *TC1* та *TC1s* працюють в режимі використання входу *RCPY* (*en_rcpy=TRUE*). Це значить, що нове значення виходу *OUT* регулятори будуть розраховувати на базі значення входу *RCPY*. Для веденого регулятора *TC1s* на вхід *RCPY* завжди подається значення, яке йде на виконавчий механізм *TV1*. Тобто, якщо ведений регулятор виведений з каскаду, він все одно буде формувати нове значення виходу внутрішнього регулятора на базі плинного *TV1*, що забезпечить безударний перехід при повторному вводити його в каскад. Для ведучого регулятора значення *RCPY* залежить від того, чи включений в контур управління ведений регулятор *TC1s*. Якщо ведений регулятор виведений із каскаду, то *RCPY=TV1*, так як *TC1*

безпосередньо управляє виконавчим механізмом, отже нове розраховане значення базується на значенні $TV1$. Якщо ведений регулятор в складі контуру, то $RSPY=LIMIT(TC1_OUT)$. Функція обмеження використовується тому, що в режимі ($en_rcpy=TRUE$) обмеження на вихід блоку PI_B не діють.

Реалізація ручного управління. Ручне управління контуру реалізоване через блок ручного управління HCI та відповідних режимів роботи регуляторів $TC1$ та $TC1s$. Значення виходу $HCI.OUT$ зв'язане зі змінною $TV1$. У автоматичному режимі роботи контуру ($TC1_AUTO=TRUE$) вихід $HCI.OUT=TC1s_OUT$. У ручному режимі вихід $HCI.OUT$ може бути змінений оператором. При цьому, враховуючи що значення $TV1$ повторюється на входах $RSPY$ блоків регулювання забезпечується безударний перехід.

Ведений регулятор завжди знаходиться в автоматичному режимі, так як для ручного режиму передбачений блок HCI . Ведучий регулятор, якщо він знаходиться в режимі каскаду з веденим, в ручному режимі повинен фіксувати своє вихідне значення в останньому положенні. Це забезпечується шляхом переключення його в ручний режим. Таким чином ведучий регулятор працює в автоматичному режимі, коли працює в автоматичному режимі весь контур, або коли ведений регулятор виведений з каскаду.

10.3. Регулювання з використанням виконавчих механізмів типу реверсивний двигун

Завдання.

Необхідно створити проект в UNITY PRO для реалізації поставленої задачі (рис.10.11 та рис.10.12) з використанням FFB бібліотеки *ControlLIB*. Налаштування проекту зробити з використанням готового програмного імітатора об'єкту та операторського екрану.

Технологічна установка являє собою два теплообмінника для підігріву рідкого продукту за допомогою гарячої води. У першому підігрівнику (рис.10.11) температура продукту на виході вимірюється датчиком температури $TT1$ (0-100°C, вхід ПЛК $\%IW0.1.2$) який вмонтований на трубопроводі на відстані кількох метрів від підігрівника, а регулюється витратою гарячої води на виході підігрівника з використанням клапану $TV1$ і приводом типу МЕО (вихід ПЛК $\%Q0.3.16$ – "більше", $\%Q0.3.17$ – "менше"). Додатково вимірюється також температура води в трубопроводі безпосередньо на виході підігрівника за допомогою датчика $TT1a$ (0-100°C, вхід ПЛК $\%IW0.1.3$). Виконавчий механізм $TV1$ має показчик положення регулюючого органу (0-100%, вхід ПЛК $\%IW0.1.4$). Час повного відкриття клапану – 10с, мінімальний імпульс – 250 мс.

Температура продукту на виході другого підігрівника (рис.3.70) вимірюється датчиком температури $TT2$ (0-100°C, вхід ПЛК $\%IW0.1.6$), а регулюється витратою гарячої води на виході підігрівника з використанням клапану $TV2$ з приводом типу МЕО (вихід ПЛК $\%Q0.3.18$ – "більше", $\%Q0.3.19$ – "менше"). Додатково вимірюється також температура води в трубопроводі безпосередньо на виході підігрівника за допомогою датчика $TT2a$ (0-100°C, вхід ПЛК $\%IW0.1.7$). Виконавчий механізм $TV2$ має датчики кінцевого положення регулюючого органу: "повністю відкритий" - вхід ПЛК $\%IO.3.0$, "повністю закритий" - вхід ПЛК $\%IO.3.1$. Час повного відкриття клапану – 10с, мінімальний імпульс – 250 мс.

Система управління повинна забезпечити регулювання температури $TT1$ та $TT2$ з підтримкою наступних функцій:

- 1) стабілізація температур на виходах підігрівників на заданому оператором значенні, з використанням ПІ регуляторів та блоків управління серводвигунами;
- 2) можливість настройки коефіцієнтів K_p та T_i з операторського екрану;
- 3) можливість ручного управління виконавчими механізмами $TV1$ та $TV2$ з операторського екрану при переключенні в ручний режим з забезпеченням безпечності переходу.

Рішення.

Перелік змінних та екземплярів функціональних блоків наведені на рис.10.13.

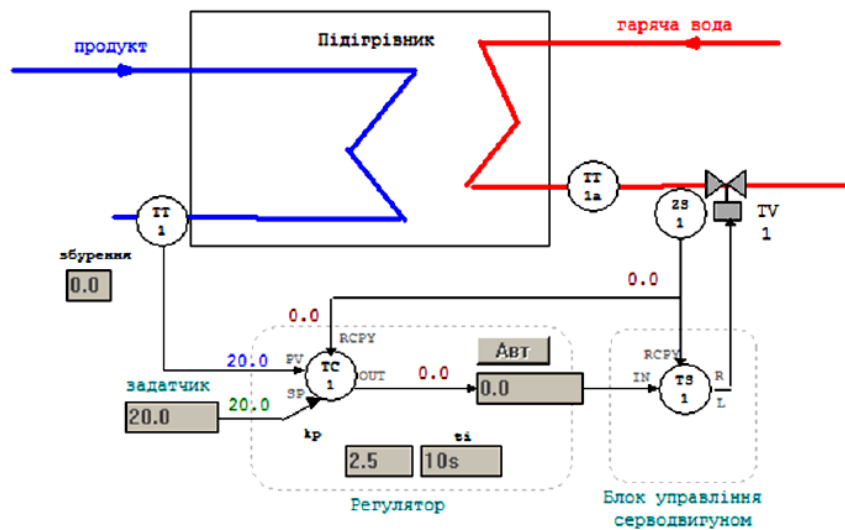


Рисунок 10.11– Операторський екран для контролю та управління процесом нагрівання в підігрівнику 1

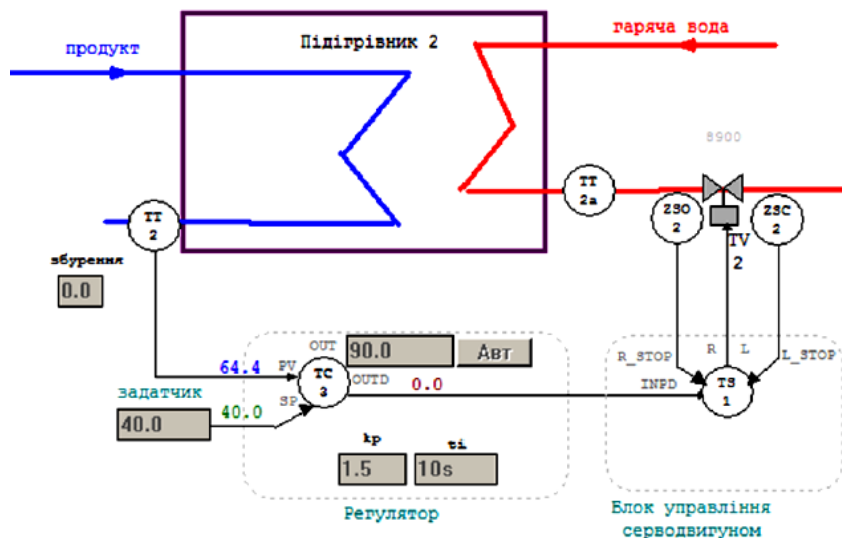


Рисунок 10.12– Операторський екран для контролю та управління процесом нагрівання в підігрівнику 2

Variables | DDT Types | Function Blocks | DFB Types

Filter

Name	Type	Val...	Comment
smInit	BOOL		Ініціалізація імітаційної моделі
TC1_AUTO	BOOL		1 - включити автоматичний режим роботи контуру TC1
TC2_AUTO	BOOL		1 - включити автоматичний режим роботи контуру TC2
TV1_CLS	BOOL		1 - команда на закриття Кл1
TV1_OPN	BOOL		1 - команда на відкриття Кл1
TV2_CLS	BOOL		1 - команда на закриття Кл2
TV2_OPN	BOOL		1 - команда на відкриття Кл2
ZSC2	BOOL		1 - спрацював датчик повного закриття
ZSO2	BOOL		1 - спрацював датчик повного відкриття
smZ1	REAL		збурення для TT1
smZ2	REAL		збурення для TT2
TC1_OUT	REAL		Вихід регулятора TC1
TC1_SP	REAL	20.0	Уставка регулятора TC1
TC2_OUT	REAL		Вихід регулятора TC2
TC2_OUTD	REAL		Інкрементальний вихід регулятора TC2
TC2_SP	REAL	20.0	Уставка регулятора TC1
TT1	REAL		T продукту на виході підігрівача 1
TT1a	REAL		T гарячої води на виході підігрівача 1
TT2	REAL		T продукту на виході підігрівача 2
TT2a	REAL		T гарячої води на виході підігрівача 2
ZS1	REAL		показчик положення регулюючого органу Кл1

Variables | DDT Types | Function Blocks | DFB Types

Filter

Name	Type	Value	Comment
TC1_PARA	Para_PI_B		Параметри регулятора
id	UINT		
pv_inf	REAL	0.0	
pv_sup	REAL	100.0	
out_inf	REAL	0.0	
out_sup	REAL	100.0	
rev_dir	BOOL		
en_rcpy	BOOL	1	
kp	REAL	2.5	
ti	TIME	t#10s	
dband	REAL	0.2	
outbias	REAL		
TC2_PARA	Para_PI_B		Параметри регулятора
TS1_PARA	Para_SERVO		Параметри блоку управління серводвигуном
en_rcpy	BOOL	1	
rcpy_rev	BOOL		
t_motor	TIME	t#10s	
t_mini	TIME	t#250ms	
TS2_PARA	Para_SERVO		Параметри блоку управління серводвигуном
en_rcpy	BOOL	0	
rcpy_rev	BOOL		
t_motor	TIME	t#10s	
t_mini	TIME	t#250ms	

Variables | DDT Types | Function Blocks | DFB Types

Filter

Name	n..	Type	Comment
TC1		PI_B	Регулятор контуру 1
TC2		PI_B	Регулятор контуру 2
TS1		SERVO	Блок управління серводвигуном контуру 1
TS2		SERVO	Блок управління серводвигуном контуру 2

Рисунок 10.13– Змінні проекту

Загальні принципи роботи контуру управління температурю в підігрівнику 1.

Функціональна схема контуру регулювання температури продукту показана на операторському екрані (рис.10.11).

Контур включає:

- канали вимірювання ($TT1$, $TT1a$);
- регулятор $TC1$ (ПІ закон), який на основі сформованого завдання (вхід SP) та вимірювального значення ($TT1$ на вхід PV), формує сигнал управління (вихід OUT), що подається на блок управління серводвигуном $TS1$; на вхід $RCPY$

регулятору заводиться сигнал зворотного зв'язку по положенню регулюючого органу *ZS1*; для регулятора налаштовуються K_p та T_i ; регулятор може працювати в автоматичному або ручному режимі (вмикається/вимикається кнопка "АВТ"); в ручному режимі вихід *OUT* задається безпосередньо оператором;

- блок управління серводвигуном *TS1*, який перетворює числовий сигнал що поступає на вхід *IN* в діапазоні 0-100% у дискретні сигнали відповідної тривалості типу "більше" (вихід R) та "менше" (вихід L); на вхід *RCPY* блоку заводиться сигнал зворотного зв'язку по положенню регулюючого органу *ZS1*;

- канал управління, який окрім клапану з виконавчим механізмом TV1 включає показчик положення регулюючого органу (*ZS1*);

Структура програми. Для реалізації даної задачі використовуються 5-ть секцій (рис.10.14): секція "*INPUTS*" – для обробки вхідних каналів вимірювання; секція "*OUTPUTS*" – для обробки вихідних каналів управління; секція "*CTRL1*" та "*CTRL2*" – для реалізації контурів управління температурою відповідно в підігрівнику 1 та підігрівнику 2. Секція "*Simulation*" призначена тільки для імітації об'єкта.

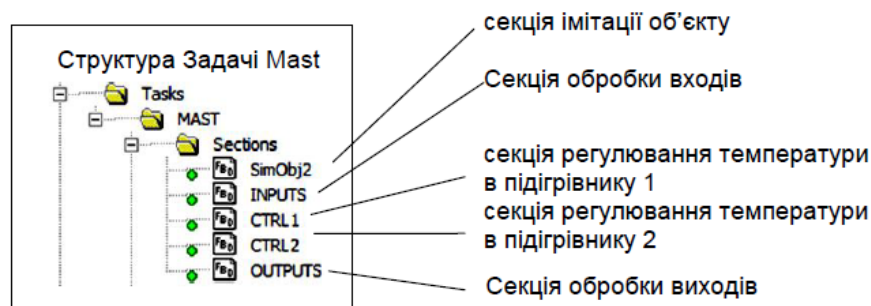


Рисунок 10.14– Структура Задачі MAST

У секції "*INPUTS*" (рис.10.15) оцифровані значення аналогових входів масштабуються шляхом множення на коефіцієнт (діапазон 0-10000 в 0-100°C). До входів контуру з підігрівачем 1 належить також показник положення *ZS1* (діапазон 0-10000 в 0-100 %XPO). До входів контуру з підігрівачем 2 належать також датчики кінцевого положення типу "відкрито" - *ZSO2*, та "закрито" - *ZSC2*.

У секції "*OUTPUTS*" (рис.10.16) для кожного виконавчого механізму на виходи ПЛК подаються сигнали "більше" (*TV1_OPN*, *TV2_OPN*) та "менше" (*TV1_CLS*, *TV2_CLS*).

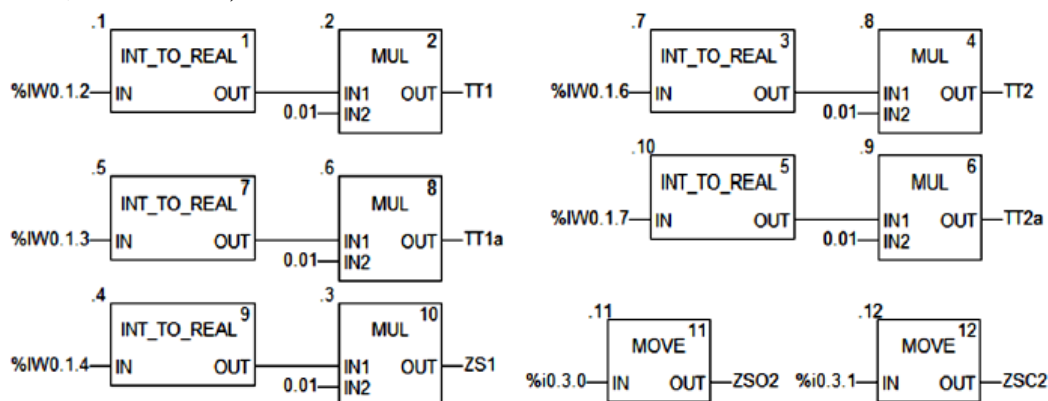


Рисунок 10.15– Секція обробки входів

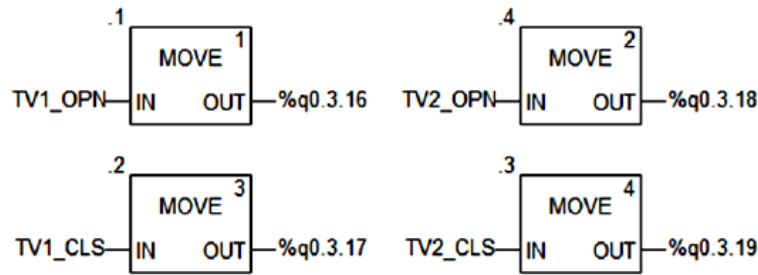


Рисунок 10.16– Секція обробки виходів

Опис роботи програми реалізації контуру управління температурою в підігрівнику 1. Програма секції "CTRL1" наведена на рис.10.17. Для UNITY PRO V<6.0 вхід/вихід *TC1.OUT* треба розірвати від *TS1* та *SERVO*, а зв'язок реалізувати через змінну *TC1_OUT*.

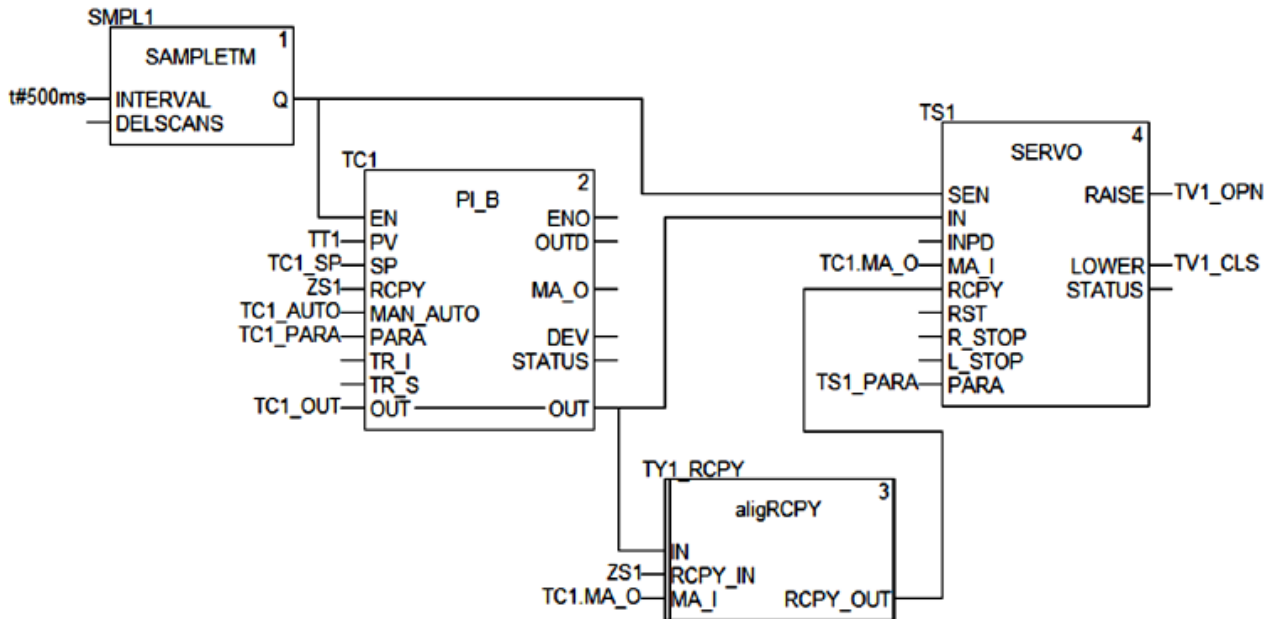


Рисунок 10.17– Секція CTRL1

Основу контуру складає ПІ-регулятор *TC1*, який на основі плинної температури *TT1* та завдання *TC1_SP* формує на виході *OUT* (прив'язаний до *TC1_OUT*) числове значення, яке подається на блок управління серводвигуном *TS1* для формування імпульсів "більше" та "менше". Регулятор *TC1* працює в режимі використання входу *RCPY* (*TC1_PARA.en_rcpy=TRUE*). Це значить, що нове значення виходу *OUT* регулятор буде розраховувати на базі значення входу *RCPY*, на який подається значення показчика положення *ZS1*.

Блок управління серводвигуном *TS1* теж працює в режимі використання входу *RCPY* (*TS1_PARA.en_rcpy=TRUE*). Це значить, що він буде перетворювати значення різниці *IN-RCPY* у дискретний сигнал *RAISE* або *LOWER* відповідної тривалості. Значення параметрів блоку дорівнюють *TS1_PARA.t_motor=t#10s* та *TS1_PARA.t_mini=t#250ms* відповідно до умов задачі.

Алгоритм роботи блоку *SERVO* працює таким чином, що у ручному режимі він буде видавати сигнали "більше" та "менше" до тих пір, поки *IN* та *RCPY* не будуть рівними. Для того, щоб блок управління серводвигуном в ручному режимі припиняв управління в зоні наближеній до положення *RCPY*, можна створити та використати блок вирівнювання (рис.10.18). Принцип роботи

алгоритму заключається в порівнянні виходу $RCPY_OUT=IN$ в тому випадку, коли в ручному режимі ($MA_I=TRUE$) вхід IN буде в зоні наближення до $RCPY$. Величина зони наближення визначається параметром $deadb$, збільшення значення якого зменшує кількість рухів однак збільшує похибку позиціонування.

The screenshot shows a software interface with tabs for 'Variables', 'DDT Types', 'Function Blocks', and 'DFB Types'. A table lists the structure of the 'aligRCPY' block:

Name	no.	V...	Comment
<inputs>			
IN	1	REAL	управляючий вихід на виконавчий механізм
RCPY_IN	2	REAL	положення виконавчого механізму
MA_I	3	BOOL	стан РУЧ/АВТ регулятора
<outputs>			
RCPY_OUT	3	REAL	вирівняне положення виконавчого механізму
<inputs/outputs>			
<public>			
deadb		REAL	0.25 зона нечутливості
<private>			
<sections>			
aligRCPY			

Below the table, the program code for the 'aligRCPY' block is shown:

```

if ABS(IN-RCPY_IN)<deadb and NOT MA_I then
    RCPY_OUT:=IN;
ELSE
    RCPY_OUT:=RCPY_IN;
end_if;

```

Рисунок 10.18– Структура і програма DFB типу aligRCPY.

Блок вирівнювання TYI_RCPY включається в схему між $TC1_OUT$ та $SERVO.RCPY$. Тобто, коли $ZS1$ знаходиться в зоні $TC1_OUT$, на вхід $TS1.RCPY$ буде подаватися значення $TC1_OUT$ а не $ZS1$, що приведе до відключення виходів LOWER та RAISE.

Опис роботи програми реалізації контуру управління температурою в підігрівнику 2. Програма секції "CTRL2" наведена на рис.10.19.

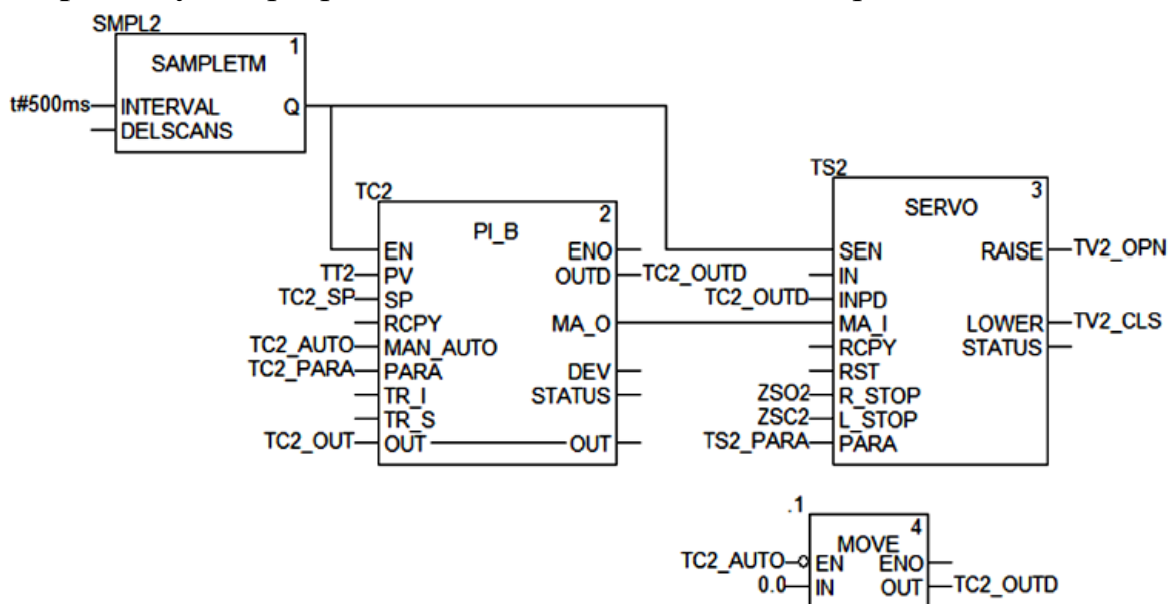


Рисунок 10.19– Секція CTRL2

Оснoву контуру складає ПІ-регулятор $TC2$, який на основі плинної температури $TT2$ та завдання $TC2_SP$ формує на інкрементальному виході $OUTD$ (прив'язаний до $TC2_OUTD$) числове значення, яке подається на блок

управління серводвигуном *TS2* для формування імпульсів "більше" та "менше". Регулятор *TC2* працює в режимі без використання входу *RCPY* (*TC1_PARA.en_rcpy=FALSE*), саме тому використовується зв'язка інкрементального виходу *TC2.OUTD* та *TS2.INPD*. Це значить, що на кожному циклі блок *TC2* буде розраховувати нове інкрементальне значення виходу *OUTD*, а абсолютне значення *OUT* використовуватися не буде.

Блок управління серводвигуном *TS2* теж працює в режимі без використання входу *RCPY* (*TS1_PARA.en_rcpy=FALSE*). Це значить, що він буде перетворювати значення входу *INPD* у дискретний сигнал *RAISE* або *LOWER* відповідної тривалості. Значення параметрів блоку дорівнюють *TS1_PARA.t_motor=t#10s* та *TS1_PARA.t_mini=t#250ms* відповідно до умов задачі.

Алгоритм роботи блоку *SERVO* працює таким чином, що у ручному режимі він буде видавати сигнали "більше" та "менше" відповідно до значення входу *INPD* на кожному циклі. Враховуючи що змінна *TC2_OUTD* оновлюється з періодичністю виклику *TC2* (500 мс), а блок *TS2* з кожним циклом, протягом 500 мс блок *TS2* буде формувати нові імпульси. Для того щоб уникнути цього ефекту, в ручному режимі (*TC2_AUTO=FALSE*) змінна *TC2_OUTD* обнуляється після обробки контуру.