Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА

Кафедра
автоматизации
и управления

МИКРОПРОЦЕССОРНЫЕ КОМПЛЕКСЫ В СИСТЕМЕ УПРАВЛЕНИЯ

Методические указания к выполнению лабораторных работ для студентов специальности
5В070200 – Автоматизация и управление

Алматы 2020

СОСТАВИТЕЛИ: Копесбаева А.А., Ким Е.С. Микропроцессорные комплексы в системах управления. Методические указания к выполнению лабораторных работ для специальности 5В070200 – Автоматизация и управление. – Алматы: АУЭС, 2020 – 39 с.

Методические указания содержат описание к 7 лабораторным работам на базе программного обеспечения TIAPortal. Лабораторные работы предназначены для студентов специальности 5В070200 - Автоматизация и управление.

Табл. 14, ил. 26, библиогр.- 5 наим.

Рецензент: Елеукулов Е.О.

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи имени Гумарбека Даукеева» на 2020г.

© НАО «Алматинский университет энергетики и связи имени Гумарбека Даукеева», 2020 г.

Содержание

1 Лабораторная работа № 1. Знакомство с программным обеспечением TIA Portal, конфигурирование и параметрирование микропроцессорных систем. Логические функций. 4

2 Лабораторная работа № 2. Программирование с использованием логических функций и операций с памятью. 14

3 Лабораторная работа № 3. Программирование с использованием таймерных функций 17

4 Лабораторная работа № 4. Программирование с использованием счетчиков и функций счета. 22

5 Лабораторная работа № 5. Программирование арифметических операций. Чтение и нормирование аналоговых сигналов. 25

6 Лабораторная работа № 6. Программирование функций и функциональных блоков. 30

7 Лабораторная работа № 7. Визуализация технологических процессов. Основы работы со SCADA системами. 33

Список литературы 38

1 Лабораторная работа № 1. Знакомство с программным обеспечением TIA Portal, конфигурирование и параметрирование микропроцессорных систем. Логические функции.

Цель работы: ознакомление с составом программного обеспечения TIA Portal, приобретение навыков работы с прикладной программой, создание проекта на языках Step7, знакомство с методами конфигурирования и параметрирования аппаратной части проекта, приобретение навыков работы в режиме Online.

Оборудование и программное обеспечение: контроллер Simatic 312C фирмы Siemens и программное обеспечение TIA Portal.

1.1 Задание к лабораторной работе

1. Создать проект.
2. Конфигурировать микропроцессорную систему.
3. Параметрировать модули.
4. Записать программу в соответствии с вариантом задания преподавателя для OB1 организационного блока.
5. Исследовать процессы в режиме Online.
6. Сделать выводы. Заполнить отчет.

1.2 Порядок выполнения работы

1.2.1 Создание нового проекта

Запустить на Рабочем столе программу TIA Portal. В открывшемся окне программы, необходимо создать новый проект.

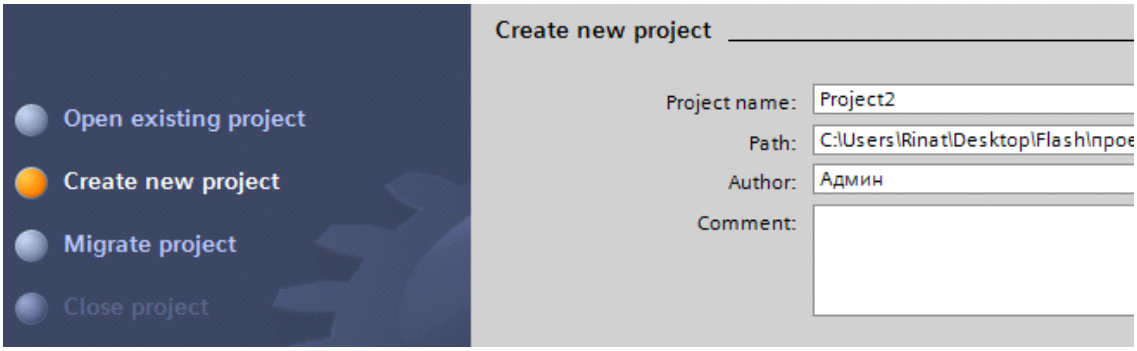


Рисунок 1.1 Пример окна проекта

Создание проекта производится в окне *Create new project* нажатием на аналогичную команду. Необходимо дать название проекту в окне *Project name*. Желательно не менять место сохранения проекта *Path*, так как для хранения файлов отведено программой специальное место на компьютере. Окно проекта представлено на рисунке 1.1. На лабораторной установке к MPI шине подключен персональный компьютер. После завершения всех настроек нажать кнопку *Create* для создания проекта.

1.2.2 Конфигурирование системы

Под «конфигурированием системы» подразумевается размещение носителей модулей, модулей и интерфейсных модулей в конфигурационной таблице. Для этого в TIAPortal, после создания проекта, откроется окно пошагового конфигурирования системы, для оптимизации работы с проектом (рисунок 1.2).

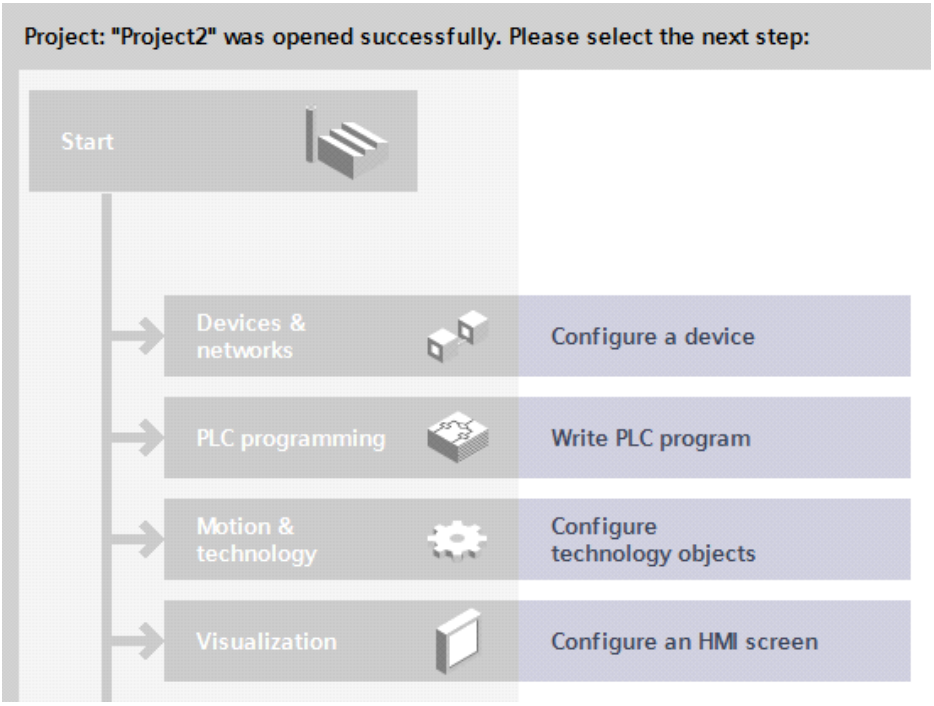


Рисунок 1.2 – Окно пошаговой конфигурации проекта

Для создания аппаратной конфигурации оборудования необходимо выбрать пункт *Devices & Networks >> Configure a device*. Далее выбрать пункт *Add new device***.** Из трех предложенных типов устройств выбрать тип PLC, и далее определить конфигурацию на основе конфигурации лабораторного стенда (рисунок 1.3). После завершения выбора ЦПУ необходимо нажать кнопку *Add*, после чего выбранный ЦПУ будет добавлен в проект. Все ЦПУ разделены на поколения, начиная с Simatic S7-300, 400, заканчивая контроллерами нового поколения Simatic S7-1200 и 1500. Выбор контроллера для проекта сопровождается информацией, с описанием типа интерфейса, количеством дополнительных встроенных модулей, с дискретными и аналоговыми сигналами. Также важно учитывать версию операционной системы ЦПУ, для корректной работы всей установки.

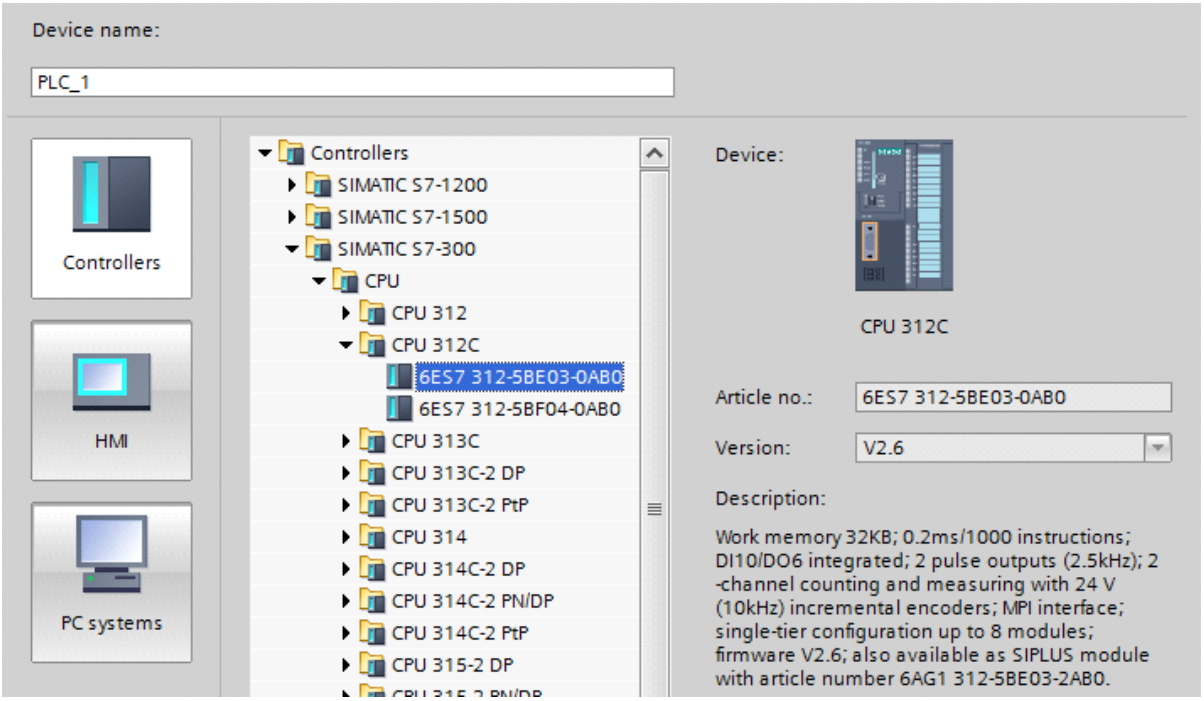


Рисунок 1.3 – Выбор ЦПУ

При составлении аппаратной конфигурации необходимо придерживаться следующей последовательности действий:

- конфигурировать и параметрировать центральный процессор CPU;
- определить тип сигнальных модулей;
- определить дополнительные модули, имеющиеся на профильной шине лабораторного стенда;
- сохранить проект;
- загрузить конфигурацию в центральный процессор CPU.

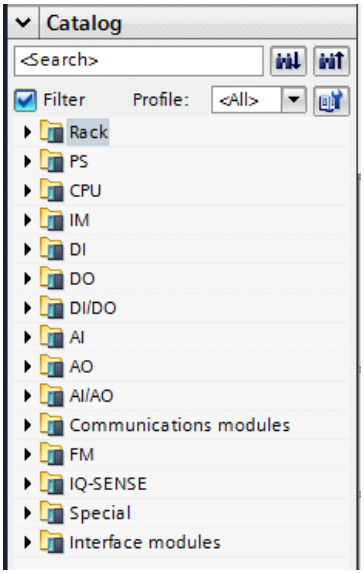


Рисунок 1.4 – Каталог оборудования

Модули аппаратного обеспечения устанавливаются на слоты в строго определенном порядке. В соответствии с реальным местом модуля составляется конфигурационная таблица. Все модули выбираются из каталога устройств, расположенного с правой стороны окна (рисунок 1.4).

DI/DO/AI/AO сигнальные модули. DI/DO – дискретные входные и выходные, AI/AO – аналоговые входные и выходные модули. Окончательный вид конфигурации приведен ниже (рисунок 1.5). Первый слот предназначен для блока питания, для подачи напряжения на ЦПУ и идущие за ним модули. Для составления правильной аппаратной конфигурации требуется произвести выбор оборудования в соответствии с реальной конфигурацией лабораторного стенда.

Рисунок 1.5 – Вид аппаратной конфигурации

Конфигурационная таблица (рисунок 1.6) отражает все выбранное оборудование в табличном виде. В таблице *Device overview* содержится краткая информация о типе ЦПУ, и всех подключенных к профильной шине устройств, включая сигнальные модуля и коммуникационные модули.

Device overview

...	Module	Rack	Slot	I address	Q address	Type
		0	1			
	PLC_1	0	2			CPU 312C
	MPI interface_1	0	2 X1			MPI interface
	DI 10/DO 6_1	0	2 2	124...125	124	DI 10/DO 6
	Count_1	0	2 4	768...783	768...783	Count
		0	3			
	DO 16x24VDC/0.5A_1	0	4		0...1	DO 16x24VDC/0.5A
	DI 8/DO 8x24VDC/0.5A_1	0	5	4	4	DI 8/DO 8x24VDC/...
	CP 343-1 Lean_1	0	6	288...303	288...303	CP 343-1 Lean
	PROFINET interface_1	0	6 X1	1023*		PROFINET interface
		0	7			

Рисунок 1.6 – Конфигурационная таблица

Остальные элементы конфигурационной таблицы заполняете в соответствии с реальной установкой. Необходимо помнить, что в конфигурационной таблице, как и на реальной установке первое место занимает блок питания, *а сразу за ним на двух местах устанавливается CPU,* затем остальные модули. Запишите адреса, которые присвоены каждому из модулей.

1.2.3 Параметрирование модулей

Таблица 1.1 – Таблица настройки адресации

Тип сигнала	Стартовый адрес
DI	0..(в зависимости от типа модуля)
DO	4..(в зависимости от типа модуля)

Теперь выберите на вкладке Device network ЦПУ, в конфигурационной таблице *Device overview* назначьте входные и выходные адреса дискретным модулям.

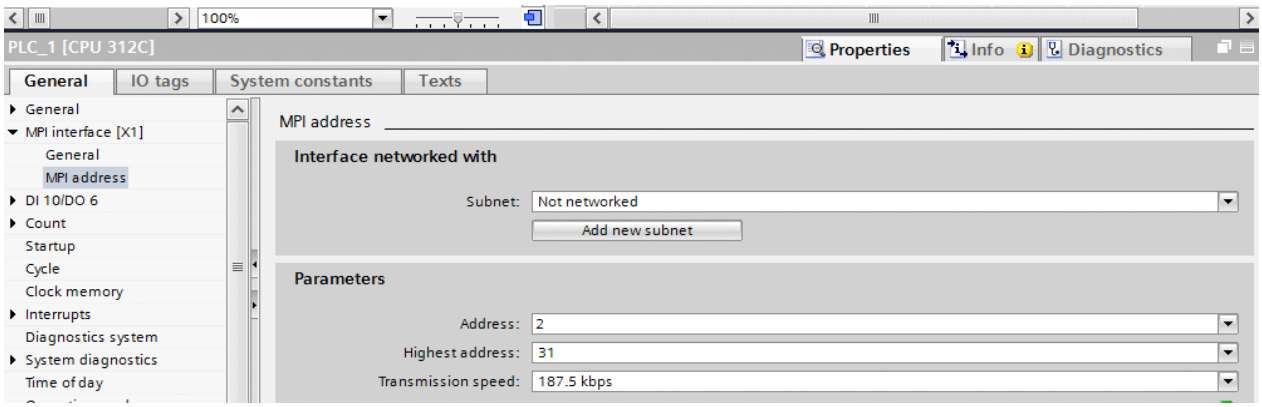


Рисунок 1.7 – Окно настроек

Во вкладке *Properties* внизу окна редактора расположено окно настроек, где можно просмотреть все свойства любого объекта, и изменить их.

На рисунке 1.7 приведены свойства ЦПУ, в частности свойства сети, настройки времени, распределение памяти и настройки безопасности.

После всех настроек аппаратной конфигурации необходимо загрузить проект в PLC. Для загрузки необходимо произвести начальную настройку соединения с устройством. Для проверки соединения и доступных устройств нажмите на кнопку *Accessible devices* на панели инструментов или *Online>>Accessible devices,* как показано на рисунке 1.8. При этом выбирается тип интерфейса (соединения с контроллером). Выбрать интерфейс MPI. В окне проверки доступных устройств появятся устройства подключенные к данной сети, с указанием адреса, серийного номера, названия контроллера. Если соединение с контроллером лабораторного стенда установлено, закройте окно и загрузите проект в контроллер.

Загрузите конфигурационную таблицу командой *PLC-Download* (на панели инструментов или *Online>>Download to device*). При этом всегда сохраняйте проект нажатием кнопки *Save Project на* панели инструментов. Сохраняя проект сохраняются все изменения, внесенные в проект (рисунок 1.9).

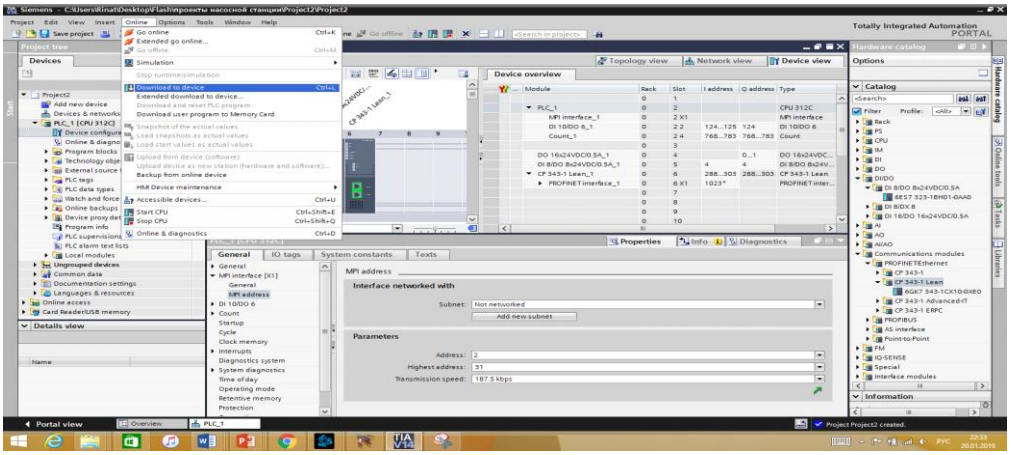


Рисунок 1.8 – Поиск доступных устройств

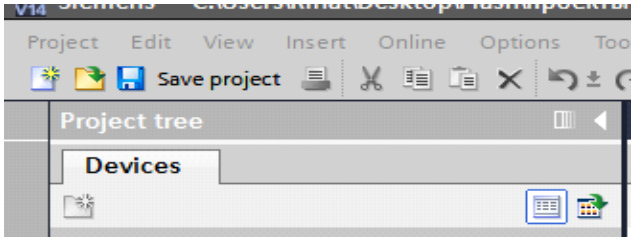


Рисунок 1.9 – Сохранение проекта

При этом все установленные параметры становятся действующими. В случае ошибки на CPU загорится сигнал ошибки SF. SF возникает в случае аппаратной ошибки оборудования, такие как обрыв провода, короткое замыкание, изъятие, повреждение модуля или его отсутствие. Если ошибка возникла после загрузки проекта, то необходимо проверить правильность составления конфигурации в проекте и загрузить проект еще раз.

В случае успешной загрузки в диалоговом окне появится запрос на перевод CPU из режима STOP в режим RUN, поставив галочку в соответствующем поле. Для того, чтобы в дальнейшем проверить выполнение прикладной программы переведите контроллер в режим RUN. Закройте окно конфигураций.

1.3 Запись алгоритма для основной программы в организационном блоке OB1

В результате созданной конфигурации и загрузки ее в промышленный контроллер, в окне проекта наряду с таблицей конфигурации появится PLC

CPU (рисунок 1.10). Внутри CPU находятся: программные блоки, таблица тегов, технологические объекты, и объекты создания и управления средой разработки. Откройте программные модули, они состоят из источника, звеньев и символьного редактора.

Задание: написать программу управления прессом. Пресс запускается нажатием кнопки Старт или при срабатывании датчика. В случае нажатия кнопки Стоп пресс останавливается.

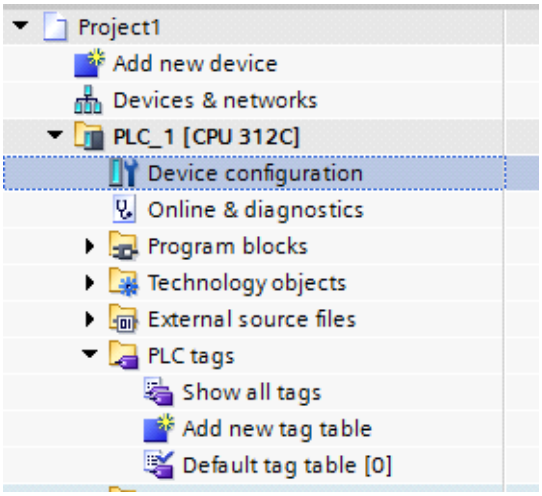


Рисунок 1.10 – PLC-проекта

Прежде чем приступить к написанию программы необходимо создать таблицу переменных в разделе *PLC tags*, и инициализировать те сигналы, которые будут использоваться в программе. Открыть раздел *Default tag table* и переименовать, кликнув один раз или правой кнопкой мыши функция *Rename*. Таблица *PLC tags предназначена* для присваивания сигналам адреса, типа данных и имя для работы в проекте.

Добавьте переменные согласно рисунку 1.11.

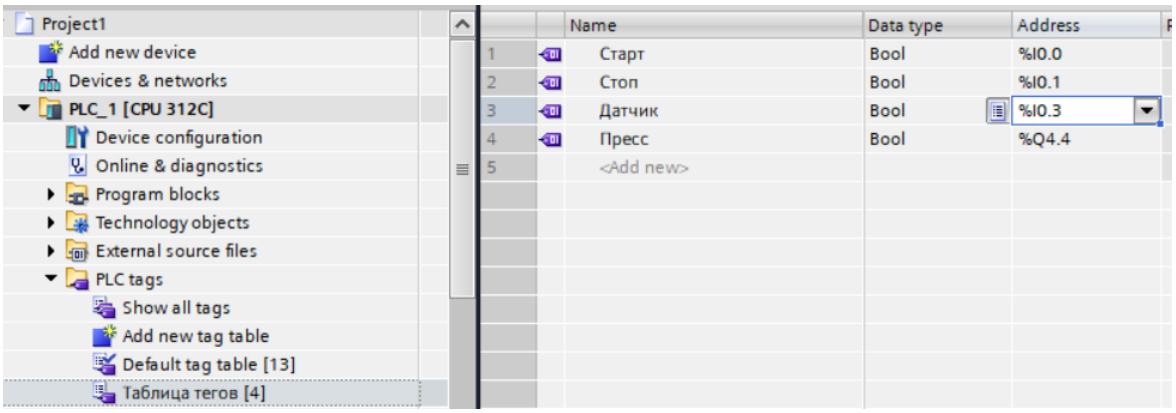


Рисунок 1.11 – PLCtags

Где тип данных Bool (Boolean) означает логический тип данных. Сигнал имеющий логический тип данных имеет два состояния 0 или 1 (0-false,1-true). Адрес дискретных сигналов указывает на тип сигнала I-input (входной сигнал), Q- output (выходной сигнал).

Для составления программы следует перейти во вкладку *Program Blocks*. И открыть OB1 для написания программной части.OB1 (Main)– это организационный блок, который циклически считывает программу и выполняет алгоритм, основанный на командах.

Все команды расположены в правом окне карты задач *Instructions* (рисунок 1.12). Логические операции с битами включают простейшие команды. Для добавления команды в рабочую область, необходимо выбрать нужный контакт и перетащить, зажав левой кнопкой мыши.

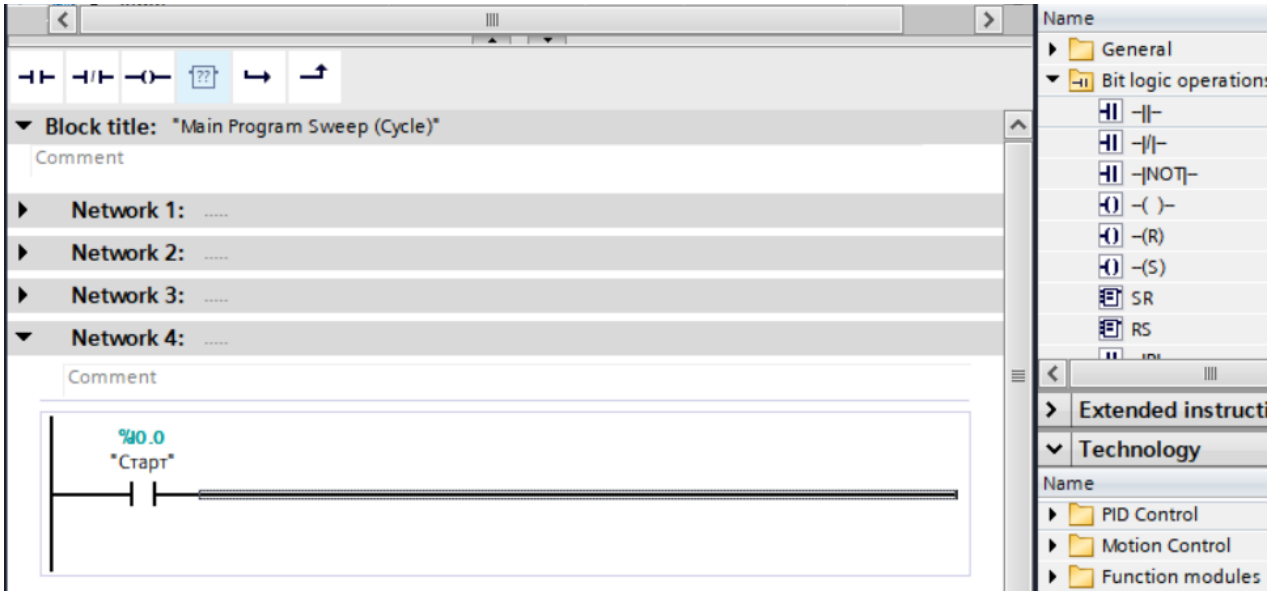


Рисунок 1.12 – Команды LAD

На рисунке 1.13 показана страница программы, написанная на языке LAD.

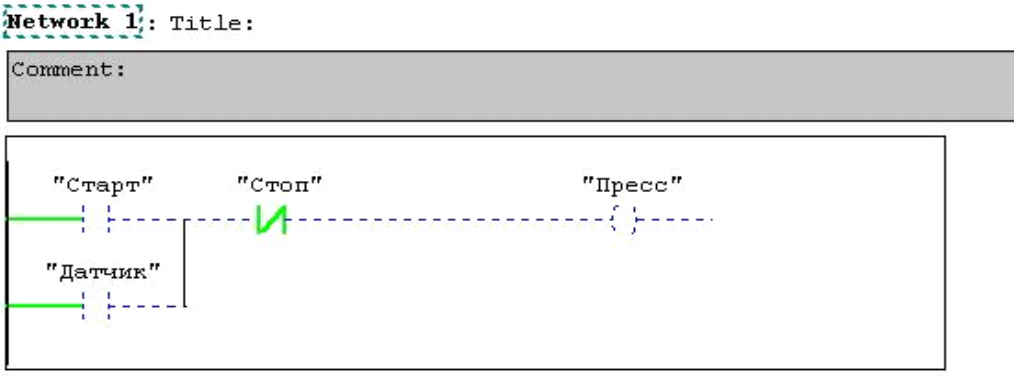


Рисунок 1.13 – Пример программы на языке LAD

Представленные программы на языке LAD напоминают коммутационную схему. Элементы схемы объединяются в сеть. Последовательность сетей образует операторскую часть программы. Язык контактного плана построен на принципе протекания тока в цепи, которая присоединена к левой шине плана. Правая часть должна заканчиваться катушкой или блоком.

Программа пока должна включать в себя хотя бы один контакт и катушку. Сохраните программу и закройте её.

1.4 Исследовать процессы в режиме online

Для исследования переменных в режиме online нужно создать таблицу переменных *Watchtable* (с контекстного меню объекта *Watchandforcetables*). Создайте в ней перечень переменных для наблюдения (рисунок 1.14). В этом случае это все элементы вашей программы. Для перехода в онлайн режим необходимо нажать на панели инструментов кнопку «Monitor all» (очки), и проверить статус сигналов и их изменение в таблице.

Теперь поступайте по привычной для вас схеме: сохраните, закройте окно редактора LAD, загрузите программу в CPU. Теперь создайте режим Online для проверки прикладной программы. Для этого вам следует открыть OB1, и “надеть очки”. Окно редактора при этом изменится, а процесс работы CPU отражается мигающим сигналом в строке состояний. Попробуйте имитировать ключами с модуля имитации дискретного сигнала наличие или отсутствие входного дискретного сигнала, наблюдая при этом выходной сигнал.

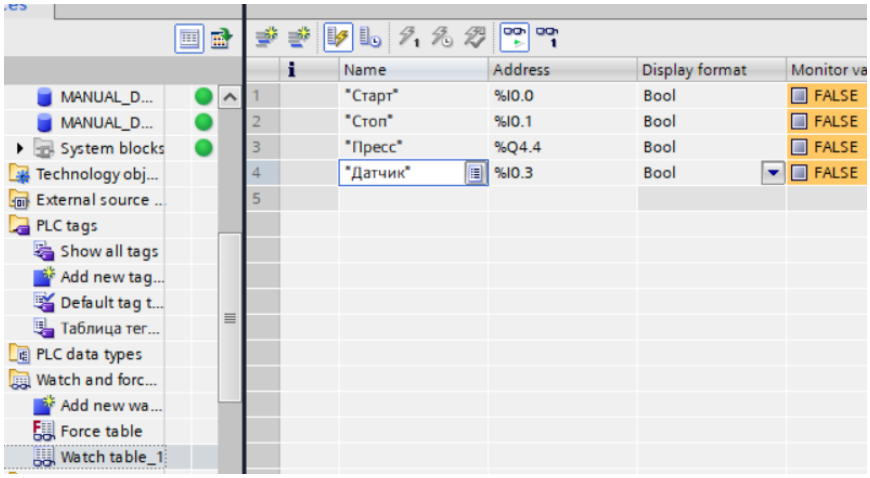


Рисунок 1.14 - Таблица переменных Watch_table

1.5 Заполнить таблицу со всеми вариантами срабатывания сигналов

Заполните таблицу сигналов на основе исследования программы в Online режиме.

Таблица 1.2 – Таблица результатов

Входные сигналы			Выходной сигнал
Датчик	Стоп	Старт	Пресс
I0.3	I0.0	I0.1	Q4.4

1.6 Отчет по результатам лабораторной работы

Сформировать отчет с результатами выполнения лабораторной работы.

Отчет должен содержать:

- все результаты создания проекта, формирования аппаратной конфигурации;
- таблицу символов;
- программный блок OB1 с программой;
- результаты исследования онлайн режима;
- все таблицы с результатами;
- программу, конвертированную на язык STL;
- выводы по работе.

1.7 Контрольные вопросы

1. Из каких, составляющих состоит проект в TIAPortal?
2. Для чего служит таблица символов (PLC tags)?
3. Возможно, ли наблюдение за переменными в режиме Online времени без таблицы переменных и без таблицы символов?
4. Из каких элементов состоит программа проекта?
5. Как проверить правильность составленной таблицы конфигурации?
6. Как выбираются и как обозначаются в основной программе дискретные входы и выходы?
7. Как назначаются адреса аппаратных средств проекта?
8. Как активировать готовый проект? Из какого окна происходит загрузка основной программы в контроллер?
9. Для чего служит таблица переменных (Watch Table)?
10. Как выбирается центральный процессор в таблице конфигурации?
11. Как выбираются блоки питания и где они устанавливаются в таблице конфигурации?
12. Какие типы адресации используются в проекте?
13. Назовите такие составляющие проекта, без которых невозможно выполнение программы.
14. Как в TIAPortal фиксируется орфографическая, логическая ошибка и ошибка в конфигурации?
15. Назначение организационного блока. Какие типы адресации используются при программировании на языке LAD?
16. Назовите основные составляющие проекта, который загружается в контроллер?

Лабораторная работа № 2. Программирование с использованием логических функций и операций с памятью

Цель работы: получить навыки программирования в программной среде TIA Portal. Программирование с использованием логических функций на базе языка LAD.

2.1 Задание к лабораторной работе

1. Создать проект.
2. Конфигурировать оборудование.
3. Составить таблицу символов.
4. Записать программу в соответствии с заданием в OB1 организационном блоке.
5. Исследовать процессы в режиме Online и проверить программу на правильность.
6. Выполнить задание по варианту и заполнить таблицу результатов.
7. Сформировать отчет по результатам лабораторной работы.

2.2 Порядок выполнения работы

2.2.1 Создайте проект, добавьте аппаратную конфигурацию и произведите настройки адресации согласно пункту п.1.2.

2.2.2 Изучите, функции логических преобразований и операций (BitLogic, WordLogic), операции сравнения (Compare, StatusBit), операции переходов (Jump). Все функции расположены на карте задач во вкладке *Instructions*.

2.2.3 Согласно заданию (логическая схема управления конвейером на рисунке 2.1) составьте программу на языке LAD, соблюдая правила написания программ и пользуясь полным перечнем операций языка (приложение В).

Задание: На рисунке 2.1 представлен конвейер, который запускается с помощью электродвигателя. В начале ленты находятся две кнопки: S1 ПУСК и S2 СТОП. В конце ленты находится кнопка S3 ПУСК. Лента может запускаться с обоих концов. Останов ленты осуществляется по кнопке S2 СТОП или датчику S4, если предмет на ленте доходит до конца.

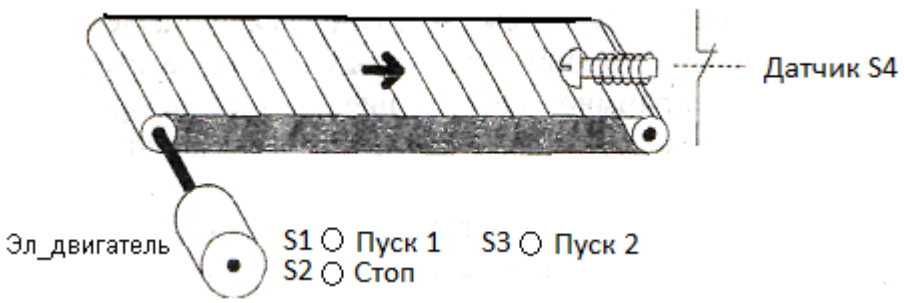


Рисунок 2.1 - Управление конвейером

2.2.4 Согласно заданию, составьте таблицу символов, выбрав в проекте назначив адресацию согласно таблице 2.1. Необходимо учитывать адресацию при конфигурации ЦПУ и сигнальных модулей.

Таблица 2.1 – Адресация сигналов

Сигнал	Абсолютное обозначение
Кнопка ПУСК1	I 0.4
Кнопка СТОП	I 0.5
Кнопка ПУСК2	I 0.6
Датчик	I 0.7
Мотор	Q 4.4

2.2.5 Составьте программу на языке LAD. Используя операции *Bit Logic Operations Составьте* программу для запуска двигателя конвейера. Проверьте программу на наличие синтаксических ошибок, используя команду *Compile*, расположенной на панели инструментов.

2.2.6 Загрузите программу в контроллер операцией *Online>>Download To Device*.

2.2.7 Наблюдайте процесс выполнения программы в режиме *Online*, имитируя входные и выходные сигналы с блока имитации дискретного сигнала. Переключая тумблеры блока, необходимо записать результаты в таблицу 2.2 и состояние сигналов, при выполнении программы. Исследуйте программу в новом окне, создавая различные ситуации: включение конвейера в начале транспортной ленты, отключение в конце ленты; включение в начале ленты и отключение по датчику и т.д.

2.2.8 Переведите язык программирования на язык STL. Для этого кликните правой кнопкой мыши на блок OB1, в открывшемся меню выберите команду *Switch Programming Language*.Из выпавшего списка выберите необходимый язык. Исследуйте программу в режиме *Online*. Определите в чем разница работы в разных режимах при работе с языком LAD и STL.

Таблица 2.2 - Результаты работы логической схемы

Входные сигналы				Выходной сигнал
I 1.4	I 1.5	I1.6	I1.7	Q 5.4

2.2.9 По заданию преподавателя выполните задание для закрепления материала:

Задача Л-1: Перемещение детали с одного транспортера на другой производится пневматическим цилиндром. Конечные положения штока цилиндра контролируются датчиками В1 и В2. Запустить работу цилиндра кнопкой однократного нажатия «ПУСК». Чтобы не произошло случайного

движения цилиндра вперед, контролируется начальное положение цилиндра датчиком B1. Управление цилиндром вперед осуществляется выдачей дискретного сигнала на выход контроллера. При достижении цилиндром конечного положения вернуть его в исходное положение по сигналу датчика B2.

Задача Л-2: Осуществляется штамповка деталей пневматическим цилиндром. Присутствие детали штамповки контролируются датчиком B1. Нанесение штампа контролируется датчиком B2. Запустить штамповку кнопкой «ПУСК». Закончите штамповку кнопкой «СТОП». Если деталь присутствует, а также нажата кнопка «ПУСК» - начать штамповку. Управление цилиндром вперед осуществляется подачей дискретного сигнала на выход контроллера. При достижении цилиндром конечного положения по сигналу датчика B2, а также по кнопке «СТОП» вернуть его в исходное положение.

Задача Л-3: Управлять работой двигателя маломощного насоса. Включить насос по кнопке «ПУСК». Если уровень в баке достигнет максимального, то сработает дискретный датчик уровня B1. По его сигналу или по сигналу кнопки «СТОП» остановить насос. Если уровень в баке меньше минимального срабатывает датчик уровня B2. По его сигналу выполнить автоматический запуск насоса.

2.3 Отчет по результатам лабораторной работы

Сформировать отчет с результатами выполнения лабораторной работы.

Отчет должен содержать:

- все результаты создания проекта, формирования аппаратной конфигурации;
- таблицу символов;
- программный блок OB1 с программой;
- результаты исследования онлайн режима;
- все таблицы с результатами;
- программу, конвертированную на язык STL;
- выводы по работе.

2.4 Контрольные вопросы

1. Какие логические операции представлены в разделе Bit Logic Operations?
2. Для чего предназначен блок OB1?
3. Особенности языка LAD?
4. Особенности языка STL?
5. Как производиться переключение программных языков в TIAPortal?
6. Какой тип данных применяется для логических операций?
7. Как можно произвести проверку программы на наличие синтаксических ошибок?
8. Сколько способов определения функциональных ошибок программы?

9. Какие настройки необходимо произвести перед загрузкой программы ЦПУ?

10. Какие компоненты программы загружаются в контроллер?

Лабораторная работа № 3. Программирование с использованием таймерных функций

Цель работы: получить навыки программирования с помощью таймерных функций. Изучить все виды таймерных функций.

3.1 Задание к лабораторной работе

1. Создать проект.
2. Конфигурировать оборудование.
3. Составить таблицу символов.
4. Записать программу в соответствии с заданием в OB1 организационном блоке с использованием таймерных функций.
5. Исследовать принцип работы тактового генератора на базе таймерных функций.
6. Выполнить задание по варианту и заполнить таблицу результатов.
7. Сформировать отчет по результатам лабораторной работы.

3.2Программирование с использованием таймеров

3.2.1 Изучите, просмотрев в разделе *Time Operations* таймерные функции и пользуясь диаграммой таймерных функций (рисунок 3.1).Диаграмма отображает зависимость работы таймера от входного сигнала и принцип действия таймеров. Все функции расположены на карте задач во вкладке *Instructions*.

3.2.2 Для получения навыков работы с таймерными функциями, необходимо составить программу со всеми типами таймеров на языке LAD, в программном блоке OB1. Согласно рисунку 3.1. следует заполнить таблицу символов, как указано в таблице 3.1.

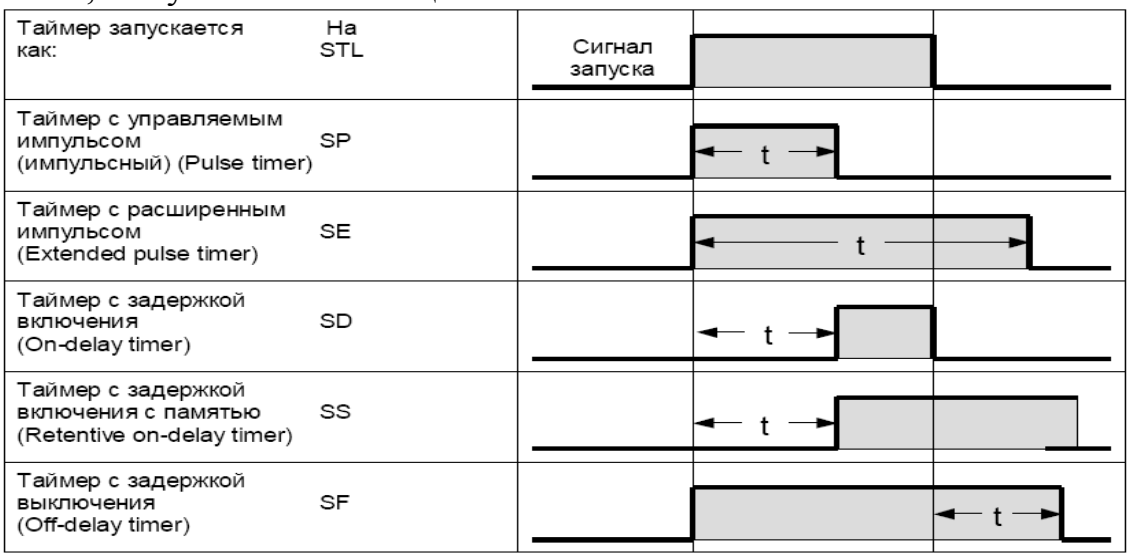


Рисунок 3.1 – Таймерные функции

Таблица 3.1 – Таблица символов

Компонент системы	Адрес
Запуск T1 SP	I 0.1
Запуск T2 SE	I 0.2
Запуск T3 SD	I 0.3
Запуск T4 SS	I 0.4
Запуск T5 SF	I 0.5
Индикатор1	Q 4.1
Индикатор2	Q 4.2
Индикатор3	Q 4.3
Индикатор4	Q 4.4
Индикатор5	Q 4.5

В таблице указаны адреса сигналов запуска и выходных сигналов (индикаторов) для наблюдения за принципом работы каждого таймера. Также в таблицу необходимо добавить пять таймеров, указать адрес Tn и тип данных Timer.

3.2.3 Программу реализации таймерных функций необходимо выполнить в программном блоке OB1, отдельно для каждого типа таймера, как показано на рисунке 3.2.

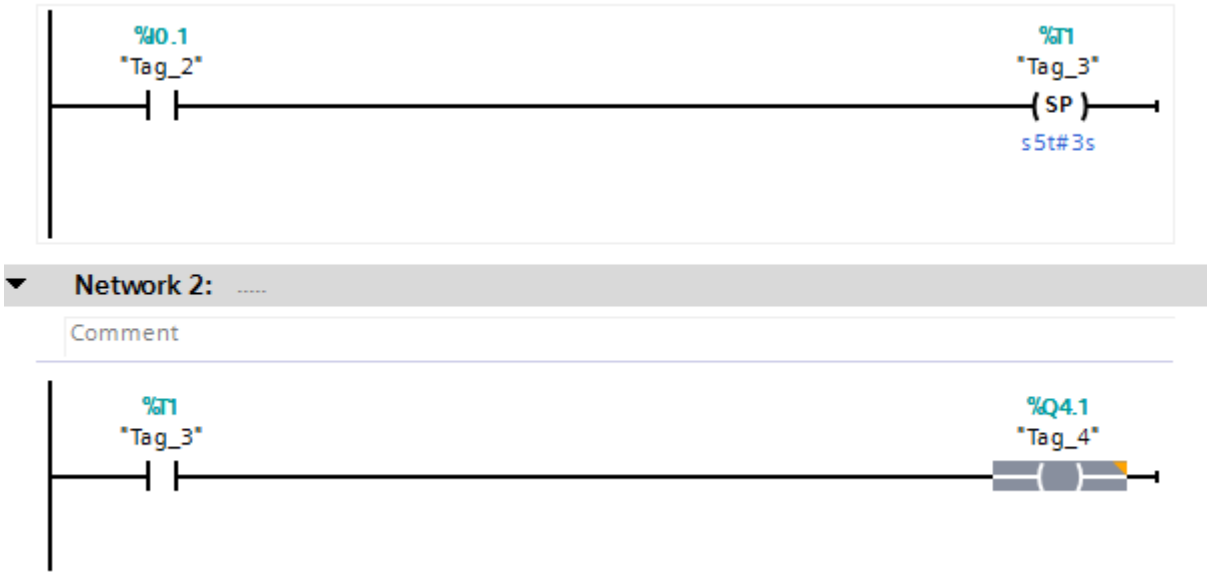


Рисунок 3.2 – Таймерные функции

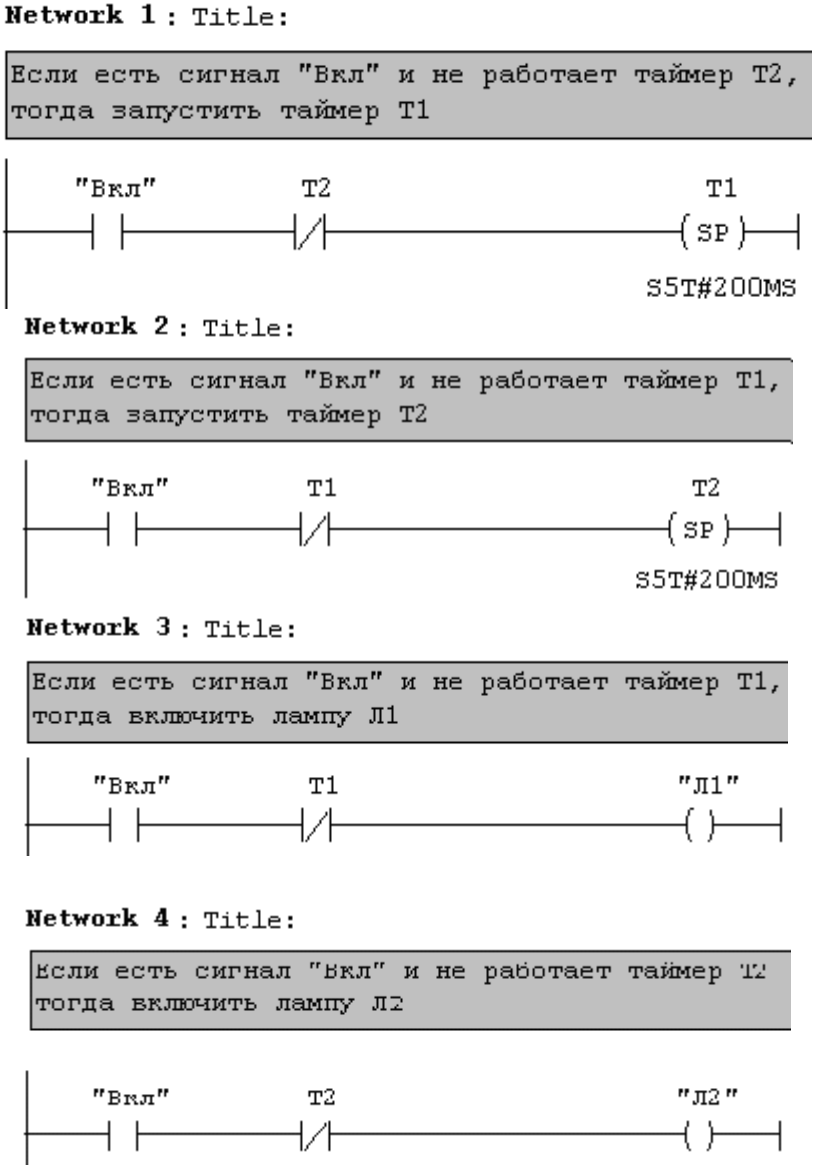
3.2.4 Загрузите готовую программу в контроллер, и проверьте ход исполнения программы подавая сигнал запуска. Необходимо исследовать влияние сигнала запуска на отсчет времени, исследовать процесс остановки и сброса таймера. Заполните таблицу 3.2.

Таблица 3.2 – Результаты исследования таймерных функций

Входные сигналы		Устанавливаемые параметры			Выходные сигналы				
I1.4	……	T1	T2	…..	Q4.1	Q4.2	Q4.3	Q4.4	Q4.5
1		S5T#3 s							
0									

3.2.5 Реализовать программу запуска тактового генератора согласно заданию.

Задание: Реализуйте функцию тактового генератора импульсов, применив обработку, управляемую временем, в специальных организационных блоках. Представленная на рисунке 3.2 и рисунке 3.3 LAD - программа иллюстрирует использование таймерных функций для генерации такта.



Network 5 : Title:

Если есть сигнал "Вкл" и не работает таймер T1
то не переходить к метке n3

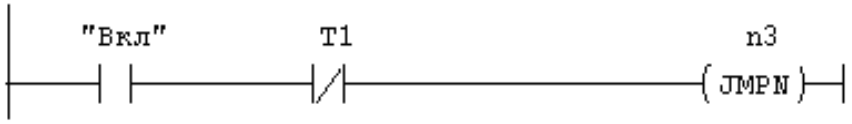
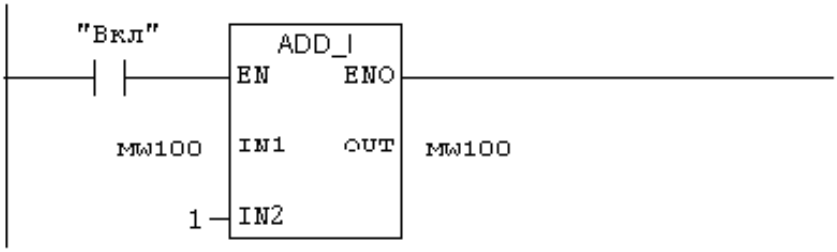


Рисунок 3.3 – Тактовый генератор

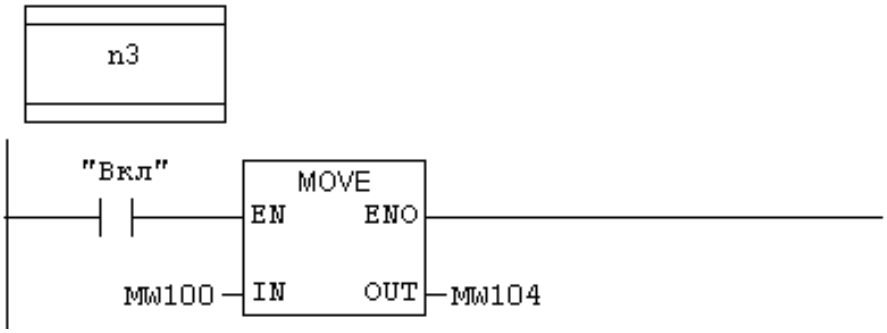
Network 6 : Title:

Если есть сигнал "Вкл" и не работает T1, то выполнять
увеличение значение ячейки MW100 по 1



Network 7 : Title:

Если есть сигнал "Вкл" значение ячейки MW100
переслать в MW104



Network 8 : Title:

Если нет сигнала "Вкл" значение ячейки MW100 и MW104
обнулить

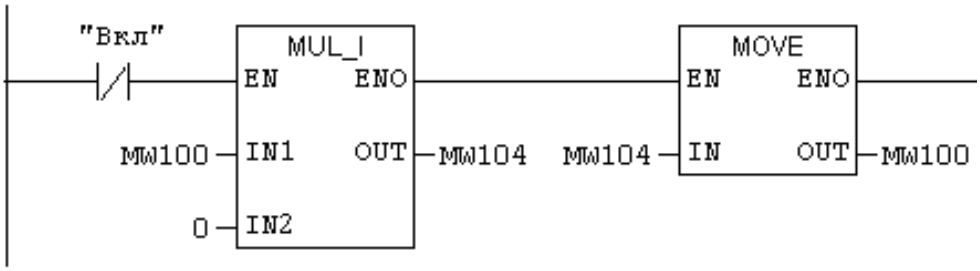


Рисунок 3.4 – Тактовый генератор

3.2.6 Загрузите программу в CPU и проверьте правильность её работы, исследуя операции, которые выполняются в программе.

3.2.7 Результаты оформите в таблицу 3.3:

Таблица 3.3 – Результат работы тактового генератора

Входные сигналы	Устанавливаемые параметры		Результаты			
	T1	T2	MW100	MW104	Л1	Л2

3.2.8 По заданию преподавателя и в качестве самотестирования выполните следующие задачи, и занесите результаты в отчет.

Задача Т-1: Решите задачу Л-1 при условии, что цилиндр перемещает деталь с одного транспортера на другой только по истечению пяти секунд после нажатия кнопки «ПУСК», а при срабатывании датчика В1 и кнопки «ПУСК» по истечению 3 секунд, а возвращается назад немедленно.

Задача Т-2: Решите задачу Л-2 при условии, что штамповка продолжается 5 секунд. Во время нанесения штамповки цилиндр задерживается в крайнем выдвинутом состоянии, а именно при срабатывании датчика В2в течении 4 секунд, а только после начинает штамповку. По окончанию штамповки цилиндр возвращается в исходное состояние.

Задача Т-3. Решите задачу Л-3 при условии, что насос запускается автоматически по истечению 5 секунд после включения датчика нижнего уровня. А в случае нажатия кнопки «Пуск» насос запускается через 6 секунд.

3.2.9 Проверьте выполнение программы на языке STL. Определите отличия в исполнении программы используются следующие операции с таймерами:

Таблица 3.4 – Таблица команд на языке STL

Команда	Описание
RT_n	сброс значения области памяти отведенного для таймера с номером “n”
SP T_n, SE T_n, SD T_n, SS T_n, SF T_n	запуск таймера с номером “n”
A T_n, AN T_n, O T_n, ON T_n, X T_n, XN T_n	опрос таймера с номером “n”
L S5T#5s	загрузка в аккумулятор значение времени
FR T_n	разблокировка таймера для нового запуска

3.3 Контрольные вопросы

1. Какие типы таймерных функций используются в TIAPortal?

2. Приведите примеры, в каких случаях необходимо применять таймеры.

3. Основной принцип работы тактового генератора.

4. В каком случае необходим переход на метку?

5. Особенности инициализации таймеров

6.Какую область памяти занимают таймеры в ЦПУ контроллера?

7.Каким образом задать время таймеру, и в каком формате?

8. Какие программные блоки используются для пересылки значений из одной ячейки памяти в другую?

9. Какой типа данных присущ таймерам?

10. Как осуществить проверку программы на правильность выполнения?

Лабораторная работа № 4. Программирование с использованием счетчиков и функций счета

Цель работы: получить навыки программирования с помощью функций счета. Изучить принцип работы счетчиков.

4.1 Задание к лабораторной работе

1. Создать проект.
2. Конфигурировать оборудование.
3. Составить таблицу символов.
4. Записать программу в соответствии с заданием в OB1 организационном блоке с использованием функций счета.
5. Произвести проверку выполнения функций счета.
6. Выполнить задание по варианту и заполнить таблицу результатов.
7. Сформировать отчет по результатам лабораторной работы.

4.2Программирование с использованием таймеров

4.2.1 Изучите, просмотрев в разделе *Counter Operations* функции счета. В разделе представлено два вида счетчиков, а именно CU (counting up), счета +1, и CD (counting down), счет -1. Все функции расположены на карте задач во вкладке Instructions.

4.2.2 Согласно заданию (рисунок 4.1) составьте программу управления складом. Для этого необходимо составить таблицу символов в соответствием с заданием.

Задание: рисунок 4.1 показывает систему складирования с двумя конвейерами и промежуточной зоной складирования между ними. Конвейер 1 транспортирует пакеты в зону складирования. Фотодатчик в конце конвейера 1 рядом с зоной складирования фиксирует количество пакетов, транспортируемых в зону. Конвейер 2 транспортирует пакеты от этой временной зоны хранения к погрузочной площадке, где они загружаются на грузовой автомобиль для доставки клиенту. Фотодатчик в конце конвейера 2 рядом с зоной промежуточного складирования фиксирует, сколько пакетов транспортируется из зоны складирования к погрузочной площадке.

Информационное табло с пятью лампочками сообщает, насколько заполнена зона складирования. Рисунок показывает LAD – программу активизации индикаторных ламп.

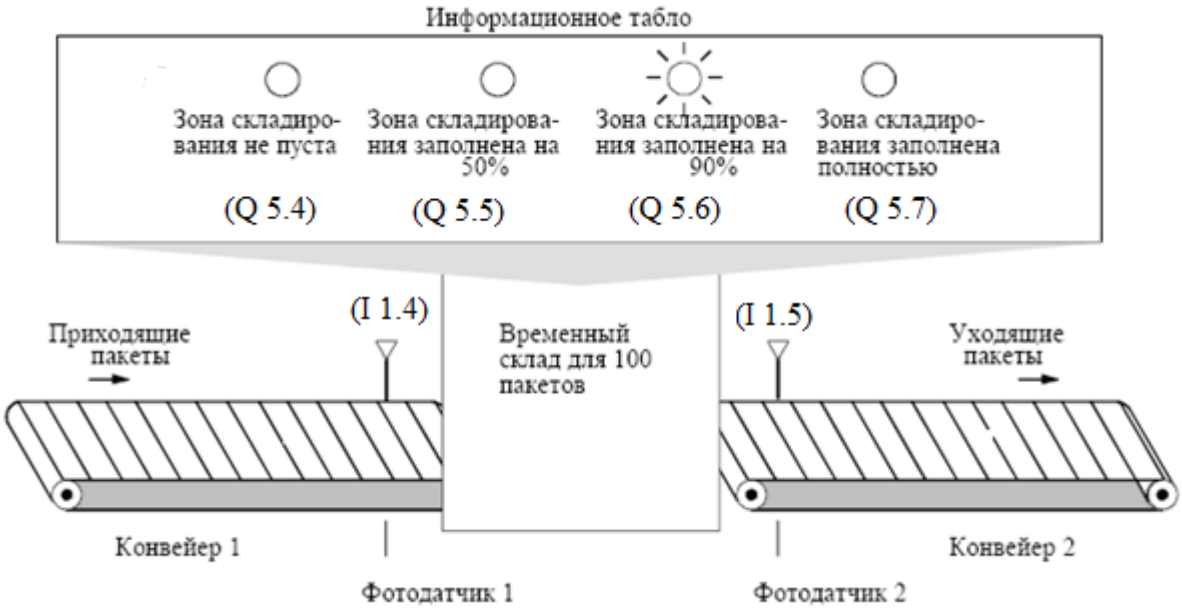


Рисунок 4.1 - Зона складирования

4.2.3 Согласно рисунку 4.2 составьте программу на языке STL, загрузите программу в CPU, и проверьте правильность её работы.

4.2.4 Произведите проверку программы в Online режиме. Проведите исследование счетчиков при прямом и обратном счете. Проверьте каким образом проверить текущее значение счета. Определите в чем разница работы в онлайн режиме на языке STL.

4.2.5 Результаты оформите в таблицу 4.1.Все результаты выполнения программы внести в отчет лабораторной работы.

```
STL             Комментарий
A     I 1.4     //каждый импульс на световом барьере 1
CU    C1        // увеличивает значение счетчика С 1 на 1, выполняя подсчет числа пакетов,
                //поступающих на склад временного хранения.
A     I 1.5     // каждый импульс на световом барьере 2
CD    C1        // уменьшает значение счетчика С 1 на 1, выполняя подсчет числа упаковок,
                //покидающих склад временного хранения.
A     C1        //Если значение счетчика не равно 0,
=     Q 5.4     // Включается лампа "Склад не пустой".
                //
L     50
L     C1
<=I             //При значении счетчика >= 50,
=     Q 5.5     // Включается лампа "Склад заполнен на 50 %".
                //
L     90
L     C1
>=I             // При значении счетчика >= 90,
=     Q 5.6     // Включается лампа "Склад заполнен на 90 %".
                //
L     C1
L     100
>=I             // При значении счетчика >= 100
=     Q 5.7     // Включается лампа "Склад заполнен полностью". (Вы можете также использовать
                выход Q 4.4 для блокировки мотора конвейера 1.)
```

Рисунок 4.2 - STL – программа активизации индикаторных ламп

Таблица 4.1 – Результаты программы управления складом

Входные сигналы			Устанавливаемые параметры			Выходные сигналы					
I1.4	I1.5	……	C1	C2	…..	Q5.4	Q5.5	……	C1	C2	……

4.2.6 По заданию преподавателя и в качестве самотестирования выполните следующие задачи:

Задача С-1: Двери гаража открываются и закрываются пневматическим цилиндром П1. Въезд машин контролирует датчик Д1, а выезд датчик Д2. Когда количество машин достигнет 50, двери гаража автоматически закрываются цилиндром П1. Пневматический цилиндр блокирует въезд пока в гараж полностью заполнен. Двери гаража открываются кнопкой «ПУСК» оператором или изнутри кнопкой «ВЫХОД».

Задача С-2: Мимо турникета, оснащенного датчиком движения Д1, проходят посетители. На выходном табло выведены пять сигнальных ламп Л1-Л5 на каждый десяток посетителей. При достижении количества посетителей 50, загорается последняя сигнальная лампа, и турникет перестает принимать монеты (закрывается задвижка З1). Новый прием посетителей произойдет только тогда, когда все предыдущие посетители покинут помещение. Обратный отсчет посетителей происходит по выходному датчику движения Д2.

Задача С-3. Датчик Д контролирует количество оборотов двигателя, по прохождению через зону обзора датчика штифта, установленного на двигателе. При количестве оборотов 256, двигатель автоматически останавливается. На выходном табло контролируется количество оборотов двигателя в двоичном коде, срабатывание датчика Д. Включение двигателя производится кнопкой «ПУСК». Предусмотрен аварийный останов двигателя кнопкой «СТОП».

4.3 Контрольные вопросы

1. Назовите основные виды счетчиков?
2. Особенности прямого и обратного счета?
3. Какой тип данных присваивается счетчикам?
4. Какое максимальное количество счетчиков можно использовать?
5. Какую область памяти занимают счетчики?
6. В чем особенность работы с языком STL?
7. Как осуществляется сброс счетчика?
8. Каким образом присвоить счетчику начальное значение?
9. Как осуществить проверку текущего значения счета?
10. Необходимо ли учитывать счетчики при составлении аппаратной конфигурации?

Лабораторная работа № 5. Программирование арифметических операций. Чтение и нормирование аналоговых сигналов

Цель работы: получить навыки программирования арифметических функций. Изучить процесс чтения и нормирования аналоговых сигналов, с помощью функции нормирования и масштабирования SCALE.

5.1 Задание к лабораторной работе

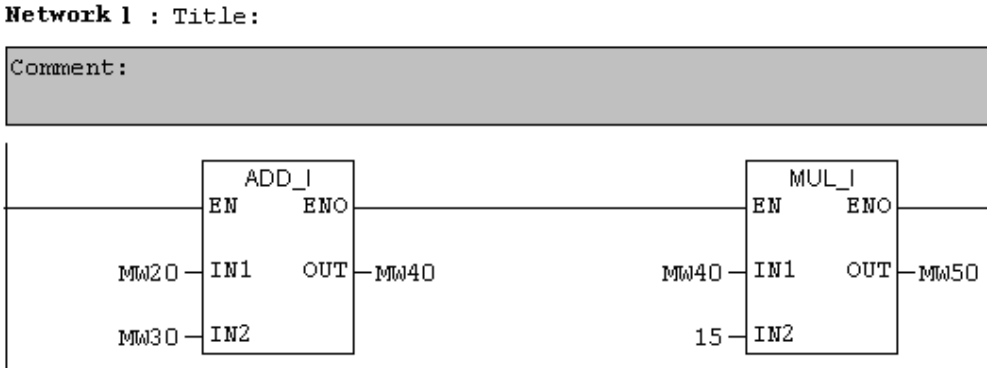
1. Создать проект.
2. Конфигурировать оборудование.
3. Составить таблицу с перечнем всех символов с назначенными адресами входов и выходов отдельно для каждого блока.
4. Записать программу в соответствии с заданием в OB1 организационном блоке для решения арифметического уравнения.
5. Произвести проверку выполнения функций.
6. Выполнить задание по варианту и заполнить таблицу результатов.
7. Реализовать функцию нормирования аналогового сигнала.
8. Выполнить задание по варианту и заполнить таблицу результатов.
9. Сформировать отчет по результатам лабораторной работы.

5.2 Программирование арифметических операций

5.1.1 Изучите, просмотрев в каталоге основные арифметические операции над числами в разделе *Math operations* и *Conversion operations*. TIAPortal обладает большим перечнем арифметических функций.

5.1.2 Составьте программу реализации следующего выражения на языке LAD:

$$MD4 = \frac{(MW20 + MW30)x15}{MD10}. \quad (1)$$



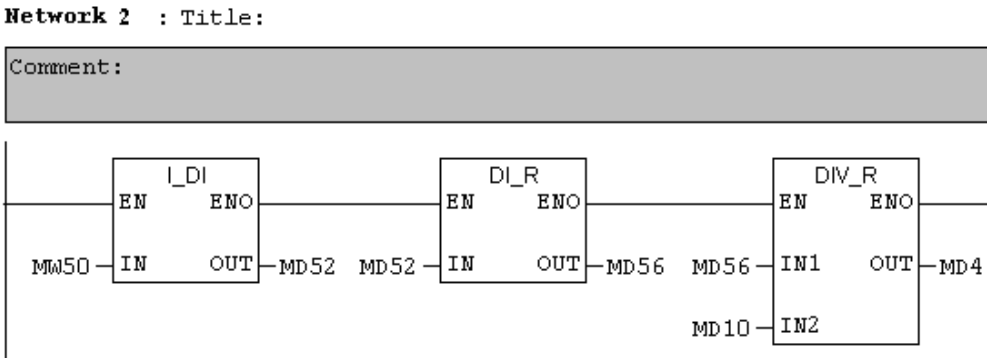


Рисунок 5.1 – LAD-программа реализации арифметического выражения (1)

5.1.3 Загрузите программу в CPU и проверьте правильность её работы. Произведите вычисления с разными значениями, изменяя значения, подаваемые на вход.

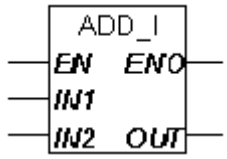
5.1.4 Результаты оформите в таблицу 5.1.

Таблица 5.1

№ блока	Блок № 1			Блок №2			Блок №3		Блок №4		Блок №5		
Имена блоков	ADD_I			MUL_I			I_DI		DI_R		DIV_R		
Входы и выходы	IN1	IN2	OUT	IN1	IN2	OUT	IN1	OUT	IN1	OUT	IN1	IN2	OUT
Значения													
1													
2													
3													

5.1.5 Программирование арифметических операций. Производить операции в контроллерах S7-300 разными типами данных нельзя. Это необходимо учитывать при назначении адресации и при реализации программы.

Таблица 5.2 – Блок сложения

Блок LAD	Параметры	Тип данных	Область памяти	Описание
	EN	BOOL	E, A, M, D, L	Разрешающий вход
	ENO	BOOL	E, A, M, D, L	Разрешающий выход
	IN1	INT	E, A, M, D. L	Первое слагаемое
	IN2	INT	E, A, M, D, L	Второе слагаемое
	OUT	INT	E, A, *M*, D, L	Результат сложения

В этой части работы используются:

- арифметика с фиксированной запятой;
- арифметика с плавающей запятой.

Сложение целых чисел (16 бит) поясняется таблицей 5.2.

Указание:

Здесь показан пример блока сложения. Все остальные операции выполняются так же.

Таблица 5.3– Арифметические Операции на языке STL

	Типы			Описание
Операции	16 битные целые числа	32 битные целые числа	32 битные действительные числа	
	ADD_I	ADD_DI	ADD_R	Сложение чисел
	SUB_ I	SUB_ DI	SUB_R	Вычитание чисел
	MUL_ I	MUL_DI	MUL_R	Умножение чисел
	DIV_I	DIV_DI	DIV_R	Деление чисел
	-	MOD	-	Получение остатка от деления
	-	-	ABS	Абсолютное значение
	-	-	SQR, SQRT	Квадрат и квадратный корень
	-	-	LN	Натуралный логарифм
	-	-	SIN, ASIN, COS, ACOS, TAN, ATAN	Тригонометрические функции
	-	-	EXP	Экспоненциальное распределение

Таблица 5.4– Операции с числами на языке STL

Типы			Описание
16 битные целые числа	32 битные целые числа	32 битные действительные числа	
+I	+D	+R	сложить значения двух аккумуляторов
-I	-D	-R	вычесть значения двух аккумуляторов
*I	*D	*R	умножить значения двух аккумуляторов
/I	/D	/R	разделить значения двух аккумуляторов
L - загрузка младшего слова в аккумулятор из операнда			
T - выгрузить младшее слово аккумулятора в операнд			

5.3 Чтение и нормирование аналоговых сигналов

Рассмотрим на примере чтение значения с датчика уровня. Допустим, что пределы измерения уровня находятся в значениях от 100 мм до 300 мм. При этом датчик выдает измеренные значения в пределах от 0 до 10 вольт. Процесс нормирования включает в себя процесс приведения любого измеренного значения $X_{изм}$. фактическому значению $X_{фак}$. по следующей формуле:

$$Y_{фак} = Y_{фак.мин} + (X_{изм} - X_{изм.мин}) \times \frac{Y_{фак.макс} - Y_{фак.мин}}{X_{изм.макс} - X_{изм.мин}} \quad (2)$$

Допустим, датчик уровня выдал сигнал 6 вольт. Необходимо узнать какому значению уровня это показание соответствует. Используя формулу 2 получим:

$$X_{\text{фак}} = 100 + (6 - 0) \cdot \frac{300 - 100}{10 - 0} = 200 \text{ мм}$$

В программном обеспечении TIAPortal существует библиотека готовых стандартных функций. Для чтения аналогового сигнала используются функции SCALE и для вывода UNSCALE.

Функция линейного масштабирования входного сигнала SCALE назначает каждому значению входного аналогового сигнала определенное выходное значение, то есть нормирует входной сигнал в соответствии с диапазоном.

Сигналы функции:

IN – входная переменная в безразмерных величинах;

HI_LIM – максимальное измеренное значение по напряжению или току;

LO_LIM– минимальное измеренное значение по напряжению или току;

BIPOLAR- дискретный сигнал установки биполярного (TRUE) или униполярного (FALSE) сигнала;

OUT- выходная переменная по напряжению (в вольтах) или по току (в амперах).

Выходное значение рассчитывается контроллером по формуле:

OUT=[((FLOAT(IN) – K1)/(K2-K1))·(HI_LIM-LO_LIM)]+LO_LIM; (3)

Функция линейного масштабирования UNSCALE назначает каждому значению входного сигнала определенное выходное аналоговое значение, то есть нормирует выходной сигнал.

Сигналы функции:

IN – входная переменная в безразмерных величинах;

HI_LIM – максимальное выходное значение по напряжению или току;

LO_LIM– минимальное выходное значение по напряжению или току;

BIPOLAR- дискретный сигнал установки биполярного (TRUE) или униполярного (FALSE) сигнала;

OUT- выходная переменная по напряжению (в вольтах) или по току (в амперах).

Выходное значение рассчитывается контроллером по формуле:

$$OUT=[((IN – LO_LIM)/(HI_LIM-LO_LIM))\cdot(K2-K1)]+ K1. \quad (4)$$

Как видно, формулы (4) и (5) идентичны формуле (2). А коэффициенты К1 и К2 определяются форматом представления числа в контроллере.

Задача студента лишь исследовать действие формул (4) и (5), а также составить независимую программу нормирования сигнала конкретной физической величины 2.

5.2.1 Выполните нормирование датчика уровня с помощью блока SCALE, согласно формулы 3.

5.2.2 Исследуйте процесс нормирования аналогового сигнала.

5.2.3 По заданию преподавателя и в качестве самотестирования выполните следующие задачи:

Задание по варианту:

В1. Реализовать с помощью функции SCALE датчик температуры, работающий в диапазоне температур 30 -150 градусов. При выходе за пределы диапазона нормальных температур (50-100 градусов) включать индикацию L1 и L2.

В2. Реализовать с помощью функции SCALE датчик уровня в баке, работающий в диапазоне температур 200-1300 мм. При выходе за пределы диапазона уровня (500-1000 мм) включать запускать закачивающий насос или выкачивающий насос.

В3. Реализовать с помощью функции SCALE датчик давления, работающий в диапазоне температур 10-130 кПа. При выходе за пределы диапазона нормальных температур (86-118 кПа) включать индикацию L1 и L2.

5.2.4 Результаты оформите в отчете к лабораторной работе.

5.4 Контрольные вопросы

1. Какие типы данных присваиваются переменным для работы с арифметическими уравнениями и вычислениями?
2. Какие арифметические операции расположены в разделе Math Operations?
3. Для чего необходимы операции конвертации?
4. Особенности конвертации из одного типа данных в другой?
5. В чем разница между 32 битными действительными числами и 32 битными целыми числами?
6. Как производится чтение аналогового сигнала?
7. Как производиться нормирование аналогового сигнала?
8. Входные сигналы функции SCALE?

9. Как задается диапазон масштабирования аналогового сигнала?

10. В каком формате отображается нормированное значение на выходе из блока SCALE?

Лабораторная работа № 6. Программирование функций и функциональных блоков

Цель работы: получить навыки работы с программными блоками. Изучить функциональные блоки и функции. Разработать функцию, состоящую из нескольких подсистем.

6.1 Задание к лабораторной работе

1. Изучить возможности программных блоков и их виды.
2. Заполнить таблицу символов и таблицу переменных в программном обеспечении TIAPortal.
3. Программировать функции чтения и нормирования FC1, и вывода FC2 аналогового сигнала в виде функций.
4. Программировать основную программу в организационном блоке OB1.
5. Сделать выводы. Заполнить отчет.

6.2 Подсистемы нормирования и чтения аналогового сигнала

6.2.1 Разделим систему на следующие подсистемы:

1. Подсистема чтения и нормирования аналогового сигнала.
2. Подсистема записи аналогового сигнала.

6.2.2 Составим таблицу с перечнем всех необходимых символов с назначением адресов для каждого блока. Таблица сигналов для каждой подсистемы приведена в таблице 6.1.

Таблица 6.1 – Сигналы подсистемы

Обозначение	Статус параметра	Символ в таблице символов	Тип переменной	Комментарии
1 Подсистема чтения аналогового сигнала				
Символы в основной программе OB1				
Символы функции FC1				
AN	Вход функции	#AN	INT	Любой входной аналоговый сигнал с модуля ввода – формальный параметр

Продолжение таблицы 6.1

H_AN	Внутренний параметр функции	#H_AN	REAL	Верхнее предельное значение напряжения датчика – формальный параметр
L_AN	Внутренний параметр функции	#L_AN	REAL	Нижнее предельное значение напряжения датчика – формальный параметр
OUT_AN	Выход функции	#OUT_AN	REAL	Измеренное значение в вольтах – формальный параметр
POL_AN	Вход функции	#POL_AN	BOOL	Установить униполярный (FALSE) или биполярный вид сигнала (TRUE) – формальный параметр

2 Подсистема нормирования сигнала					
Символы в функции FC 2					
XFMAX1	300.0	Константа	#XFMAX1	REAL	Максимальное значение фактической переменной
XFMIN1	100.0	Константа	#XFMIN1	REAL	Минимальное значение фактической переменной
XIZMAX1	10.0	Константа	#XIZMAX1	REAL	Максимальное значение измеренной переменной
XIZMIN1	0.0	Константа	#XIZMIN1	REAL	Минимальное значение измеренной переменной
XF1	MD24	Выход функции	#XF1	REAL	Значение фактической переменной

6.2.3 Изучите алгоритмическую схему каждой функциональной подсистемы отдельно. Подсистема чтения аналогового сигнала представлена на рисунке 6.1. Подсистема нормирования аналогового сигнала представлена на рисунке 6.2.

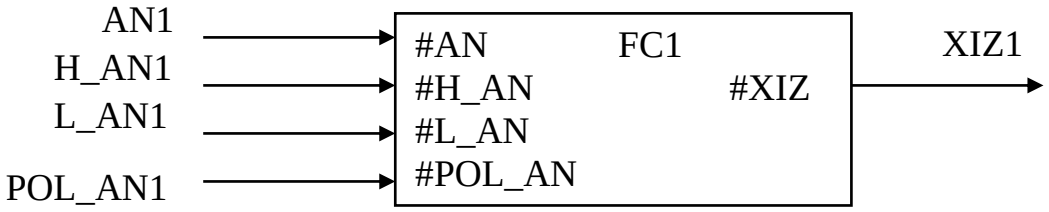


Рисунок 6.1 - Подсистема чтения аналогового сигнала

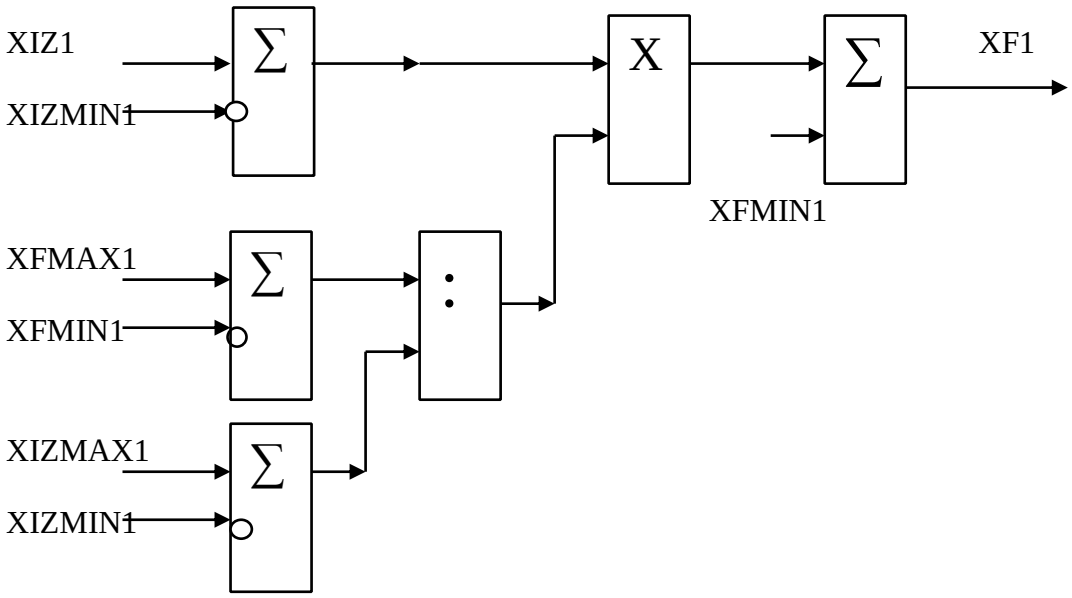


Рисунок 6.2 - Подсистема нормирования сигнала

6.2.4 При программировании используйте те же обозначения, что и в таблице 6.1 и на рисунках 6.1 - 6.2. Программу нужно выполнить в соответствии с алгоритмической схемой.

При составлении подобного рода программ рекомендуется соблюдать следующую последовательность действий:

- составить таблицу символов;
- программировать функции, начиная с таблицы описания блока;
- программировать организационный блок;
- составить таблицу переменных.

6.2.5 Составить программу, используя все программные блоки. Основная программа должна быть записана в виде организационного блока OB1 в соответствии с приведенной алгоритмической схемой. Она предназначена для вызова блоков FC1, FC2. В первом варианте программы реализуются формулы (2) и (3). По заданию преподавателя следует

продолжить организационный блок нормируя различные физические величины (температуру, расход, напряжение и т.д.).

6.2.6 Составить таблицу символов в соответствии с программой. Она отражает относительные адреса переменных и соответствующие им абсолютные адреса. Таблица составлена с учетом всех подсистем, предназначена для удобства чтения программы. Таблица переменных предназначена для управления входными переменными и контроля выходных переменных. Заполнение таблицы следует начать с заполнения необходимых адресов входов и выходов, символы, при наличии таблицы символов и типы данных появляются автоматически.

6.2.7 Проверить выполнение составленной программы вызвав блок SCALE. Если выходные значения двух блоков, совпадают, то программа выполнена верно.

6.2.8 Составление отчета. Внести в отчет программу и результаты реализации подсистем.

6.3 Контрольные вопросы

1. Как создаётся функция?
2. Как происходит обмен информацией между основной программой и функцией?
3. Почему функцию называют блоком без параметров?
4. Как обозначены в функциях формальные и фактические параметры?
5. Сколько входных и выходных данных может содержать функция?
6. Какие соответствия должны быть между фактическими и формальными переменными?
7. Перечислите типы данных использованных в программе.
8. Как преобразовать функцию так, чтобы её работа включалась по дискретному сигналу с пятого входа?
9. Что отражают переменные К1 и К2 в формулах (4) и (5)?
10. Изменится ли коэффициент нормирования сигнала, если изменить диапазон измерения, без изменения диапазона соответствующего напряжения?

Лабораторная работа № 7. Визуализация технологических процессов. Основы работы со SCADA системами

Цель работы: получить навыки работы с человеко-машинным интерфейсом. Познакомиться с созданием проекта для визуализации и диспетчеризации.

7.1 Задание к лабораторной работе

1. Создать проект с HMI панелью.
2. Произвести конфигурацию и настройку связи между PLC и HMI.
3. Реализовать программу согласно заданию.
4. Установить соединение и проверить выполнение программы.

5. Сделать выводы. Заполнить отчет.

7.2 Конфигурирование HMI устройства

7.2.1 Создайте проект.

7.2.2 Выполните задание, написав программу по заданию, приведенному ниже.

Задание: в операторской комнате расположены три кнопки включения освещения кнопка К1, К2 и К2. По нажатию любой из кнопок включаются три лампы освещения Л1, Л2 и Л3.

7.2.3 Составить таблицу символов в соответствии с заданием.

7.2.4 Написать программу в организационном блоке, проверить и загрузить в контроллер.

7.2.5 Проверить правильность выполнения программы.

7.2.6 Добавить новое устройство HMI, нажав *Add new device* расположенного на дереве проектов. Из всплывающего окна с тремя типами устройств выберите HMI устройство, как показано на рисунке 7.1. Выберите Simatic Basic Panel, 7’’KTP700 Basic DP. В данном случае DP означает тип интерфейса (MPI). Окно вывода устройств отображает информацию о панели. Нажмите ОК.

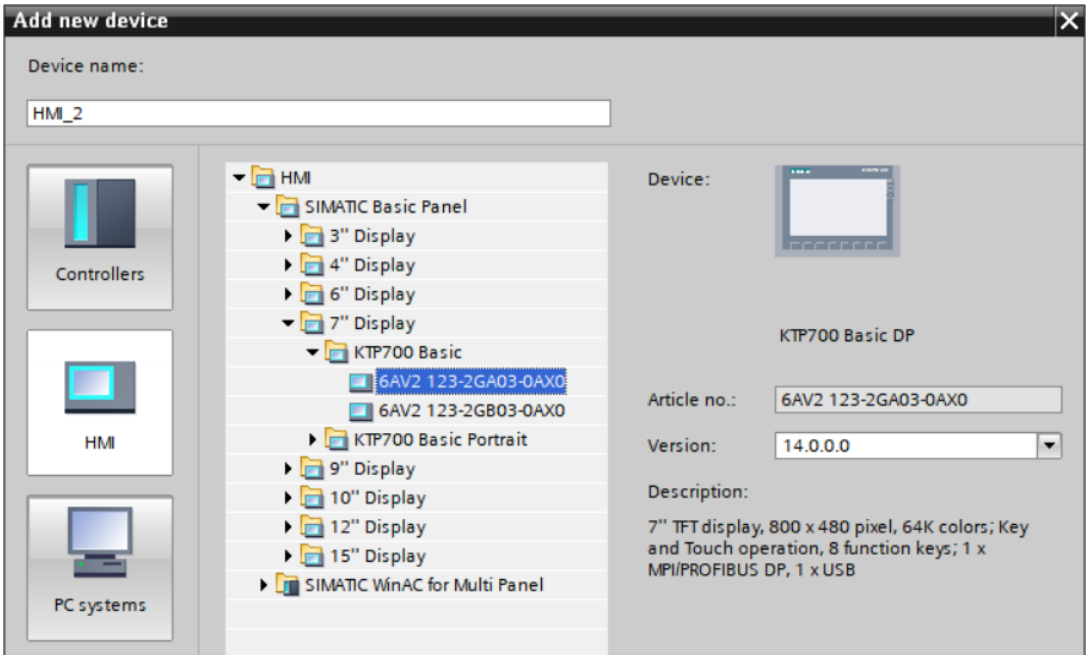


Рисунок 7.1 – HMI устройство

7.2.7 Далее выйдет окно настройки устройства (рисунок 7.2). Данное окно позволит пошагово установить соединение между устройствами. Во вкладке *Browse* выберите ЦПУ с которым хотите установить соединение. TIAPortal автоматически создает соединение. После нажмите *Next* и продолжите настройку устройства.

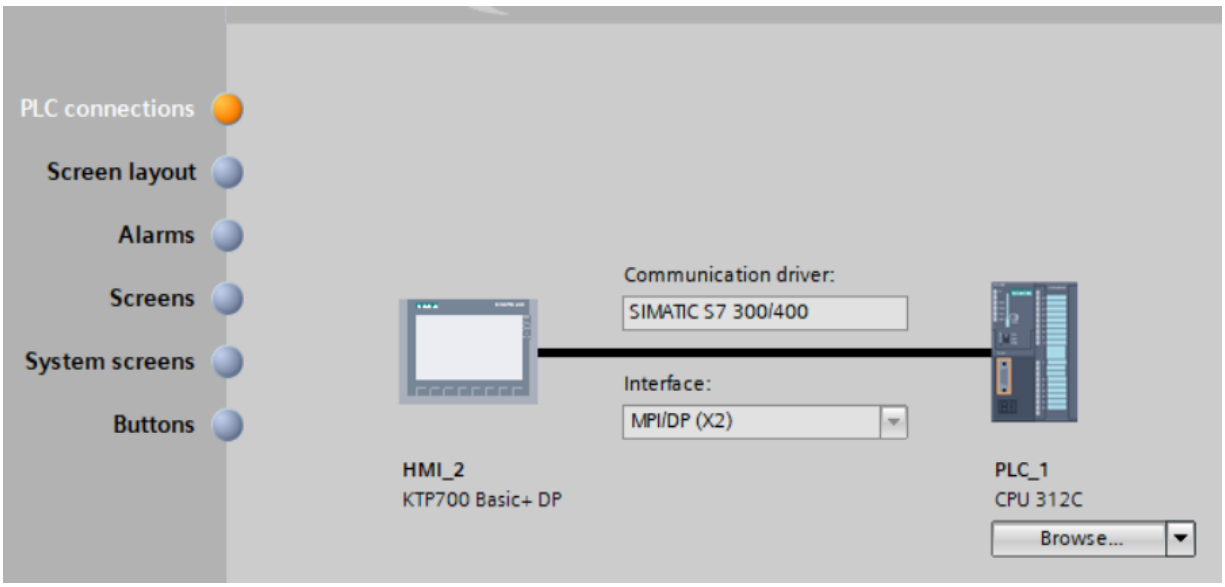


Рисунок 7.2 – Окно настройки связи

7.2.3 В следующем окне настроек выберите цвет и заголовки экрана (рисунок 7.3).

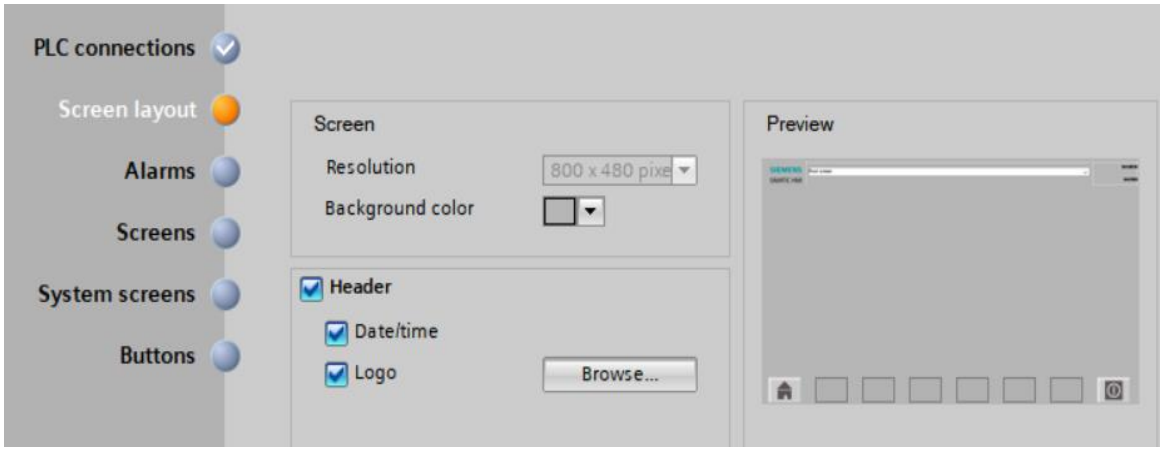


Рисунок 7.2 – Настройка экрана

7.2.4 В следующем окне настроек, необходимо убрать отметки со всех свойств. Это свойства отображения и вывода предупреждений и сигналов тревоги.

7.2.5 Следующая настройка определяет количество окно и переходы между ними. Для нашего проекта достаточно одного окна.

7.2.6 Далее нужно отключить все системные окна. Системные окна позволяют выводить системную информацию. Информация о системе, режиме системы или о правах доступа.

7.2.7 Последнее окно настройки предназначены для добавления базовых кнопок. Добавьте кнопку выхода из RUNTime «Exit» и кнопку «Start screen». Завершите настройку нажав кнопку Finish.

7.2.8 В дереве проекта появилось устройство HMI. Раскройте вкладку HMI. В разделе *Connections* проверьте соединение и его свойства.

7.2.9 Для внесения переменных в таблицу откройте раздел HMItags. Перенесите переменные из таблицы PLCtagsвHMItags, просто скопировав переменные.

7.2.10 В разделе Screen окно Rootscreen является главным окном. Перейдите на главное окно. На карте задач справа от рабочего окна, расположены элементы для визуализации. Добавьте кнопку Button, выбрав в разделе *Elements,* и перенести элемент кнопки на рабочую область. Перенесите элемент окружность. Как показано на рисунке 7.3. Внизу во вкладке *Properties>>General>>Label* измените название кнопки по заданию.

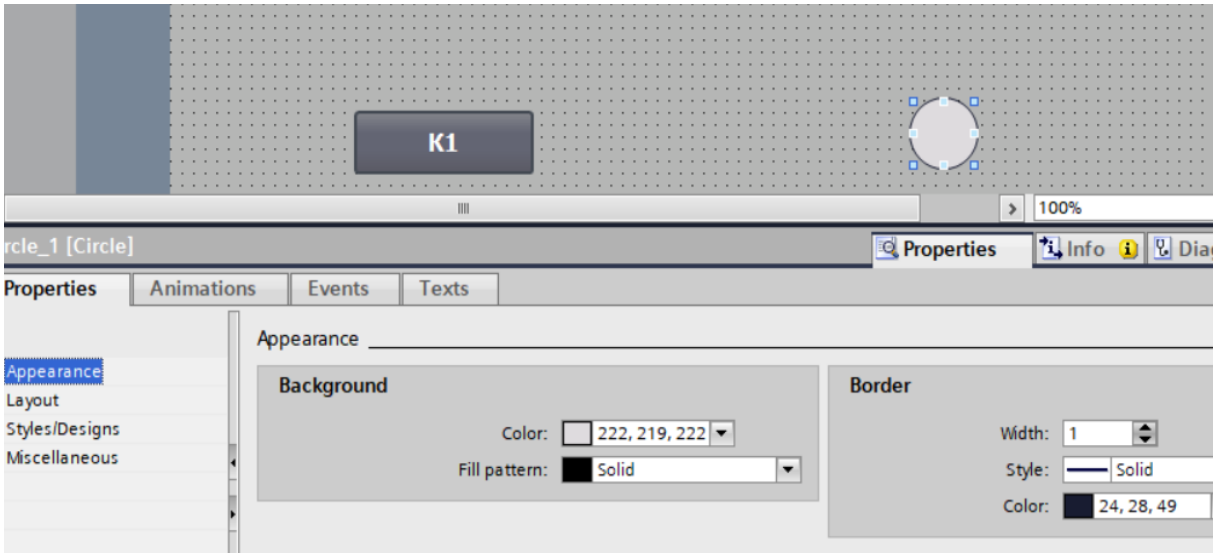


Рисунок 7.3 – Окно визуализации

7.2.11 Необходимо присвоить кнопке событие, а именно привязать кнопку к переменной. Для этого необходимо во вкладке Events>>Click>>Add Function выбрать событие которое позволит осуществлять изменение сигнала. В открывшемся меню выберите во вкладке Edit Bits>>Invert Bit в строке Tag выберите переменную кнопку К1. При однократном нажатии кнопки будет происходить изменение состояния сигнала.

7.2.12 Для анимации лампочки для визуализации задачи, необходимо кликнуть на элемент окружности, и во вкладке *Animations>>Display>>Add New Animations*. Выберите тип анимации Appearance, который позволяет работать с анимацией связанное с изменением цвета.

7.2.13 Привяжите тег Л1 к данному элементу, ниже в таблице в столбце Range необходимо указать цвет переключения сигнала. При выключенной лампе Л1 указать значение «0», в столбце Background Color выбрать цвет для выключенной лампы. Далее указать значение «1» и выбрать цвет включенной лампы.

7.2.14 Добавьте остальные кнопки К2 и К3, и элементы окружности, имитирующие сигнал лампы. И согласно пункту п.7.2.12 и п.7.2.13 осуществите привязку переменных к кнопкам и окружностям.

7.2.15 После настройки окна панели необходимо запустить окно в режиме RunTime. Для этого на панели инструментов нажмите кнопку Start Simulation. После компиляции откроется окно панели.

7.2.16 Произведите проверку программы. Исследуйте процессы, происходящие при нажатии кнопок.

7.2.17 Составьте отчет. Занесите все результаты в отчет лабораторной работы.

7.3 Контрольные вопросы

1. Что такое человеко-машинный интерфейс?
2. Какие устройства доступны в TIAPortal?
3. Особенности конфигурации HMIпанелей?
4. В чем разница PLCtagsи HMItags?
5. Как производится настройка окна панели?
6. Какие основные компоненты при работе с конфигурации?
7. Какие элементы визуализации расположены во вкладке Elements?
8. Какие события можно присвоить кнопке управления?
9. Как осуществляется анимация элементов HMI?
10. Каким образом производиться привязка переменных к элементам визуализации?

Список литературы

1. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования/Под ред. проф. В.П. Дьяконова. – М.: СОЛОН - Пресс, 2004. – 256 с.

2. Романчик А.Л., Рудакова Л.Н. Автоматизация теплоэнергетических процессов. Учебное пособие. Алматы: АИЭС, 1994. – 72 с.

3. Э. Парр Программируемые контроллеры: руководство для инженера / пер. с англ. – М.: БИНОМ, 2007. – 516 с.

4. А.А. Копесбаева. Промышленные контроллеры. Конспект лекций для студентов специальности 5В071600 – Приборостроение - Алматы: АУЭС, 2015. - 45 с.

5. А.А. Копесбаева, Е.С. Ким. Средства автоматизации технологического процесса ТЭК. Методические указания по выполнению курсовой работы для студентов специальности 5В071600 – Приборостроение. - Алматы: АУЭС, 2016. - 23 с.

сводный план 2020 г., поз. 67

Елена Сергеевна Ким
Акшолпан Ауелбековна Копесбаева

МИКРОПРОЦЕССОРНЫЕ КОМПЛЕКСЫ В СИСТЕМЕ УПРАВЛЕНИЯ

Методические указания к выполнению лабораторных работ для студентов специальности
5В070200 – Автоматизация и управление

Редактор: Мусабаева З.Т.
Специалист по стандартизации : Данько Е.Т.

Подписано в печать __. __. __.
Тираж _50_ экз.
Объем 2,6уч.-изд. л.

Формат 60х84 1/16
Бумага типографская №1
Заказ ___. Цена _1300__ тенге

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи имени Гумарбека Даукеева»
050013 Алматы, ул. Байтурсынова, 126