

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

До розрахункового завдання

«Візуалізація роботи механізму»

з дисципліни «Мікропроцесорний електропривод»

Харків НТУ «ХПІ»

2023

Методика применения программного обеспечения фирмы SIEMENS для конфигурирования сетевой коммуникации и средства HMI

После монтажа модулей на профильной шине, их электрического подключения, соединения всех сетевых компонентов необходимо проведение программного конфигурирования аппаратуры, определение параметров ее настройки, а также конфигурирование систем промышленной связи. Кроме того, задачей проектировщика на этом этапе является написание управляющей программы для объекта (механизма, процесса), а также разработка проекта визуализации в соответствии с заданными условиями технологического процесса.

Для решения вышеуказанных задач находит применение промышленное программное обеспечение SIMATIC, а именно стандартное инструментальное средство проектирования STEP7 и средство человеко-машинного интерфейса WinCC flexible. Данные пакеты устанавливаются на персональном компьютере (PC) или специально выпускаемом программаторе (PG).

1 Особенности инсталляции пакетов STEP7 и WinCC flexible

По умолчанию пакет STEP7 содержит стандартный набор языков программирования (LAD, STL и FBD), которых достаточно для решения большинства задач автоматизации. При необходимости устанавливаются дополнительные языки (например, SCL, CFC, GRAPH, HIGRAPH), которые сокращают время проектирования и применяются для решения отдельных задач.

Для тестирования и отладки программ без использования реального оборудования возможна установка программы-имитатора S7-PLCSIM. Начальный этап выбора описанных компонентов установки пакета STEP7 показан на рис.1. Выбор производится путем отметки соответствующего поля.

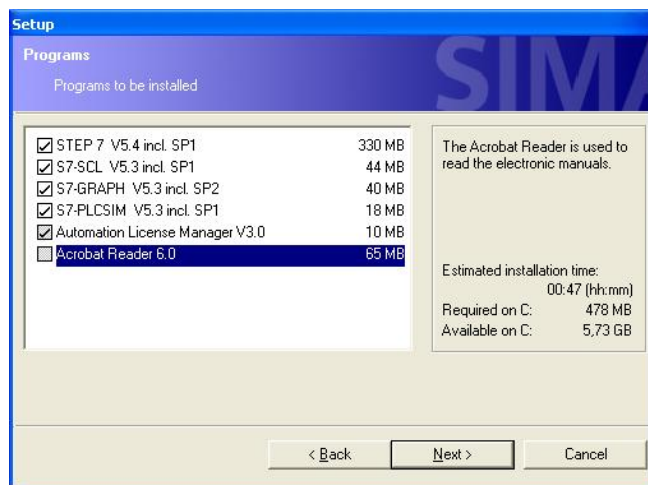


Рисунок 1 – Выбор компонентов

При инсталляции пакета WinCC flexible выберем обычную установку (Custom Installation), а также компонент Runtime/Simulation. WinCC flexible Runtime выполняет функцию симулятора и позволяет проверить работу и отладить проект без наличия оборудования (операторской панели).

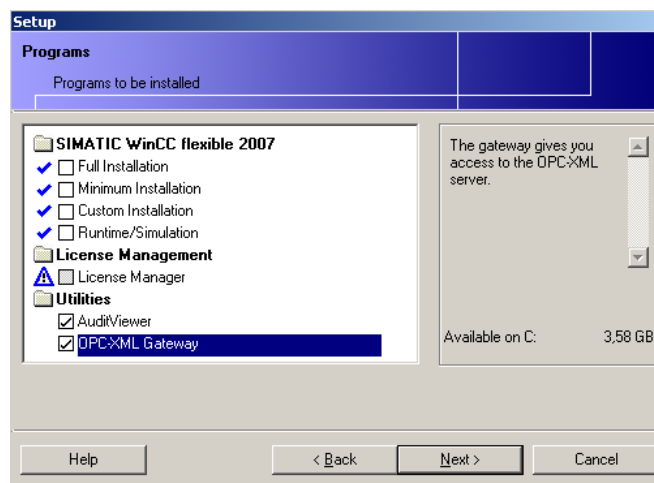


Рисунок 2 – Выбор компонентов

2 Запуск SIMATIC Manager и создание проекта

Работа с пакетом STEP 7 начинается с запуска программы SIMATIC Manager. *SIMATIC Manager* – это основное приложение, программная среда для проектирования и программирования. В нем можно реализовать следующие функции:

- создавать проекты;
- конфигурировать аппаратные средства и назначать им параметры;
- конфигурировать аппаратные сети;
- программировать блоки;
- отлаживать и сдавать в эксплуатацию управляющие программы.

Доступ к различным функциям является объектно-ориентированным, интуитивно понятным и легким для изучения. Окно SIMATIC Manager показано на рис. 2.

3 Структура и окно проекта

Проект – это упорядоченно сохраняемая информация о наборе аппаратуры, предназначенной для решения задач автоматизации, о связях между используемыми устройствами, об управляющих программах пользователя и сами эти программы. Проект похож на папку, в которой все данные хранятся в виде иерархической структуры (иерархии проекта) и доступны в любое время. Отображение этой иерархии в Окне проекта похоже на отображение проводника в Windows. Другой внешний вид имеют только пиктограммы объектов.

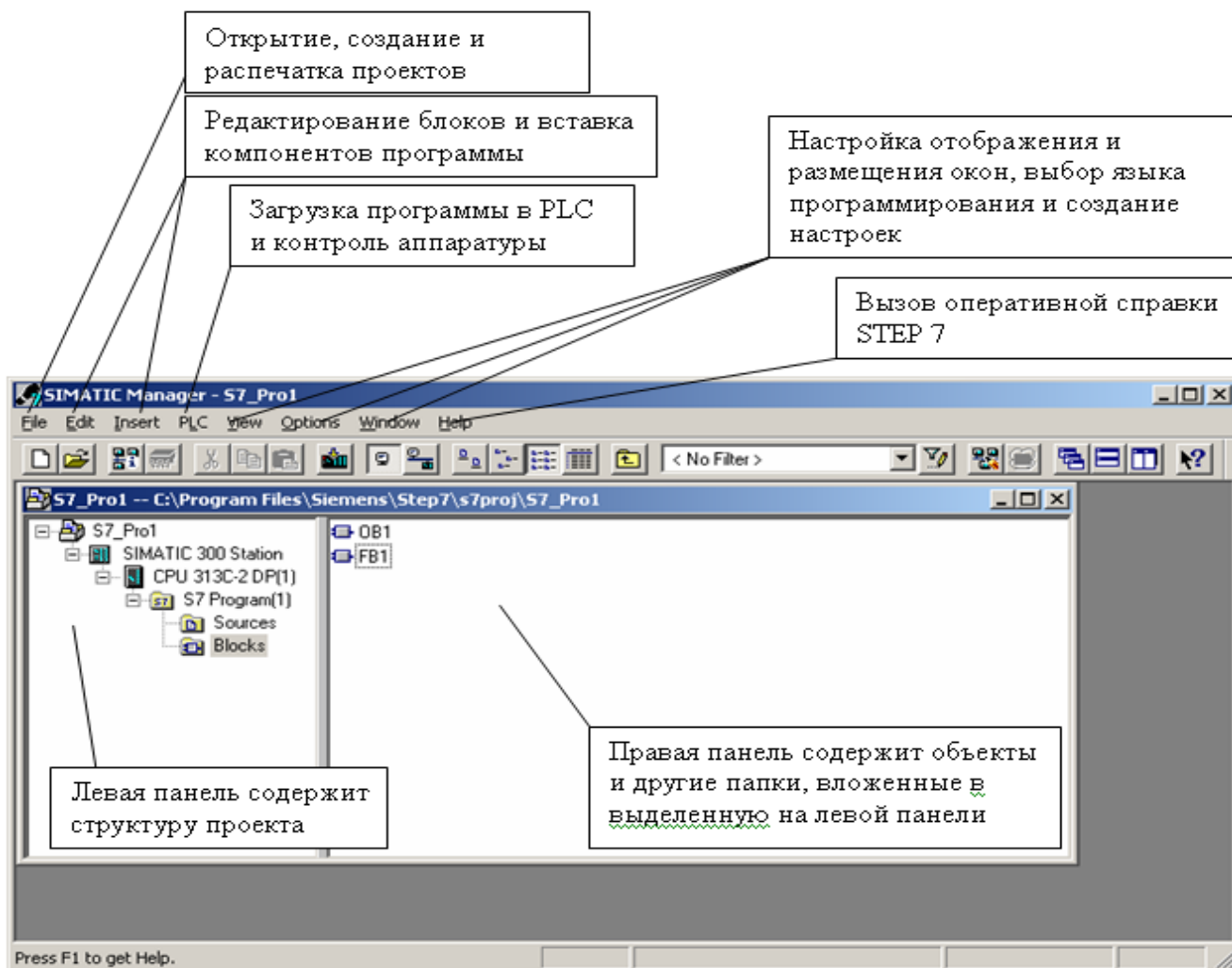


Рисунок 3 – Диалоговое окно SIMATIC Manager

Верхняя часть иерархии проекта структурирована следующим образом:

1-й уровень: проект;

2-й уровень: подсети, станции или программы S7;

3-й уровень: зависит от объекта на уровне 2.

Окно проекта – первичное окно SIMATIC Manager.

Окно проекта разделено на две половины (рис. 3). В левой половине показана древовидная структура проекта. Правая половина показывает объекты, содержащиеся в объекте, выделенном в левой половине (в выбранном представлении – крупные символы, мелкие символы, список или подробности).

Чтобы отобразить полную древовидную структуру проекта, необходимо щелкнуть в левой половине окна на квадратике со знаком плюс. Полученная в результате структура будет выглядеть аналогично структуре, представленной на рис. 3. В данном примере в верхней части иерархии объектов находится объект

"S7_Pro1" как пиктограмма для всего проекта. Она и является названием проекта. Ниже рассмотрим варианты создания нового проекта и их особенности.

4 Порядок создания проекта

Самый легкий путь создания нового проекта – это использование мастера нового проекта. Для открытия мастера используют команду меню *File* → "New Project" Wizard [Файл → Мастер нового проекта]. Мастер напоминает о необходимости ввести требуемые детали в диалоговых окнах, а затем создает и открывает проект.

Новый проект можно создать и вручную, используя также команду меню *File* → *New* [Файл → Новый] в SIMATIC Manager. Этот способ более наглядный и практичный. Он позволяет проектировщику самостоятельно

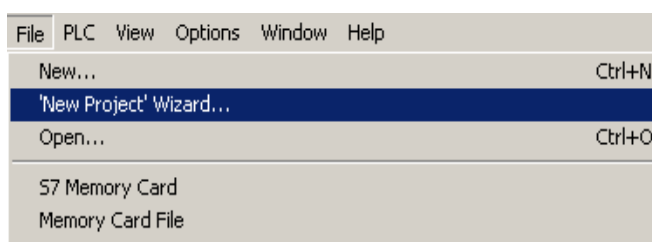


Рисунок 4 – Первый шаг при создании нового проекта

выполнить ключевые этапы создания проекта (например, выбрать необходимую станцию S7-300), которые в первом случае остаются "за кадром". После успешного создания проекта становится доступным его *второй уровень* (уровень подсетей и станций), позволяющий проектировщику выполнить программное конфигурирование аппаратуры, определить ее параметры, а также произвести конфигурирование систем промышленной связи.

5 Порядок конфигурирования и параметрирования оборудования

Под "конфигурированием" оборудования понимается процедура размещения изображений реальных стоек, модулей, устройств децентрализованной периферии в Окне станции (рис. 5).

Под "параметрированием" понимается задание необходимых свойств, назначение параметров выбранным модулям или устройствам из числа допустимых.

Конфигурирование и параметрирование оборудования производится с помощью стандартного инструментального средства *Hardware Configuration* (конфигуратор аппаратуры), которое входит в состав пакета STEP7. Данное приложение также позволяет решать задачи конфигурирования систем промышленной связи, использующих для обмена данными сеть PROFIBUS.

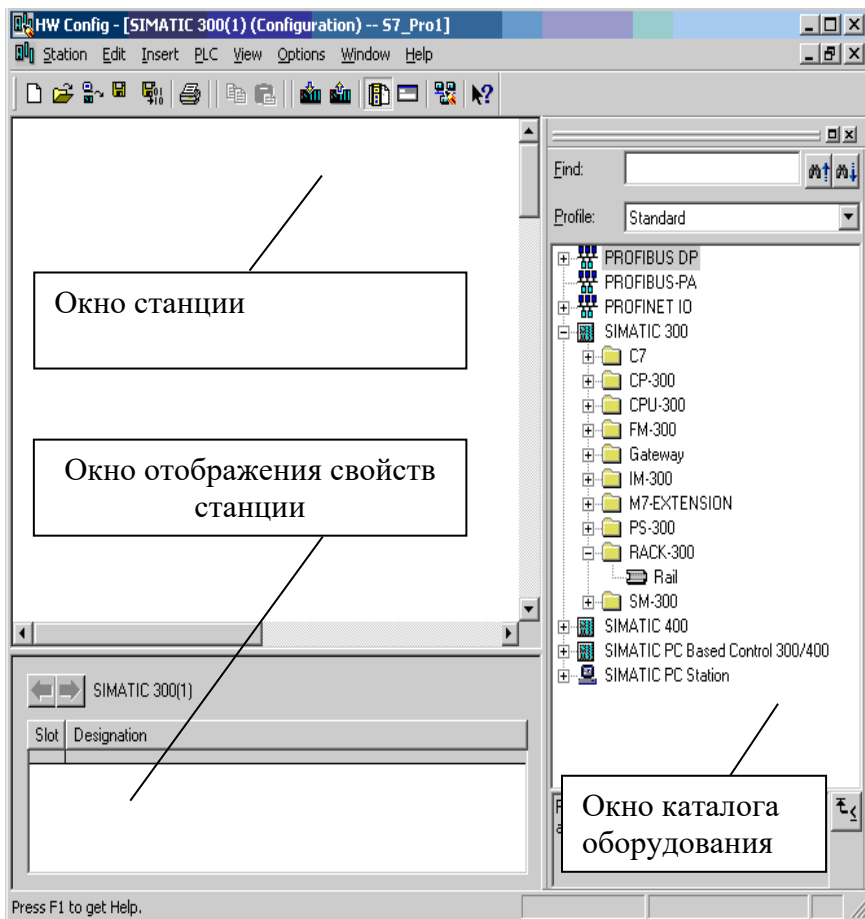




Рисунок 6 – Интерфейсное окно Hardware Configuration

Для вызова Hardware Configuration необходимо в левой половине Окна проекта (рис. 3.10) выделить мышкой имеющуюся станцию Station . После этого в правой половине Окна проекта два раза щелкнуть на появившемся объекте Hardware . Откроется интерфейсное окно Hardware Configuration, представленное на рис. 6.

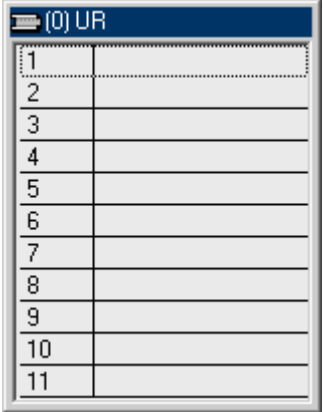
Для конфигурирования систем управления используют два окна:

1. Окно станции (слева), в котором размещается стойка с оборудованием;
2. Окно Hardware Catalog [Каталог аппаратуры] (справа), из которого выбираются необходимые названия аппаратных компонентов, например, стойка (RACK), сигнальные модули (SM), функциональные модули (FM), коммуникационные процессоры (CP) и т.д., соответствующие реально смонтированному оборудованию.

5.1 Конфигурирование контроллера

Прежде всего, необходимо разместить в левом Окне станции пустую стойку, в которую будет помещаться выбранное оборудование контроллера. Для этого в каталоге выбираются соответствующая центральная стойка (для S7-300 это профильная шина *RACK-300* → *Rail*) и перетаскиваются в левое окно. Стойка появляется в виде небольшой конфигурационной таблицы в верхней части Окна станции. Строки этой таблицы называются слотами. В нижней части Окна станции дополнительно появляется подробное описание стойки с ее данными, например, заказными номерами, адресом многоточечного интерфейса MPI, адресами входов/выходов.

Пример выбранной центральной стойки (профильной шины) без (до) заполнения ее модулями показан на рис. 7. Теперь можно заполнить слоты профильной шины необходимым оборудованием в соответствии с правилами для слотов. Так, центральный процессор CPU S7-300 может быть установлен лишь во втором слоте (то есть, с номером 2), блок питания PS (Power Supply) – только в первом, в третий слот устанавливается интерфейсный модуль IM для стоек расширения ввода-вывода (при отсутствии модуля, слот остается пустым).



[0] UR	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Рисунок 7 –
Профильная шина до
конфигурирования
PLC

Проведем конфигурирование имеющегося оборудования стенда с учетом его электрической и функциональной схем, описанных ранее. Заполнение стойки начнем с первого слота.

Слот №1 – заполняется блоком питания PS 307 2A центрального процессора (заказной номер модуля 6ES7307-1BA00-0AA0).

Слот №2 – заполняется центральным процессором CPU 313C-2DP контроллера (заказной номер модуля 6ES7313-6CF03-0AB0).

Слот №3 – остается пустым, так как интерфейсный модуль IM не используется в составе PLC.

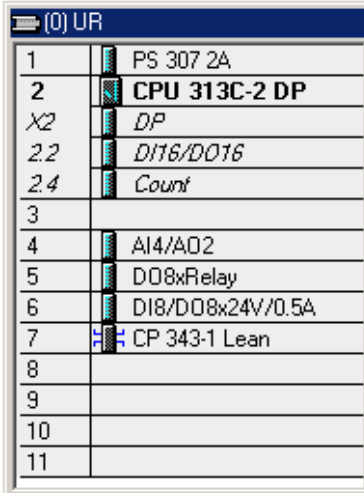
Слот №4 – заполняется модулем ввода-вывода аналоговых сигналов SM 334 AI4/AO2 (заказной номер модуля 6ES7334-0KE00-0AB0).

Слот №5 – заполняется модулем вывода дискретных сигналов (релейные выходы) SM 322 (заказной номер модуля 6ES7322-5HF00-0AB0).

Слот №6 – формально должен заполняться имитационным модулем входных и выходных дискретных сигналов SM 374 (заказной номер модуля 6ES7374-2XH01-0AA0). Поскольку данный модуль не включен во встроенный каталог оборудования пакета STEP7, то для корректности используем модуль ввода-вывода дискретных сигналов SM 323 DI8/DO8 (заказной номер модуля 6ES7323-1BH00-0AA0), что подробнее пояснено в описании модуля SM 374.

Слот №7 – заполняется коммуникационным процессором CP 343-1 LEAN для осуществления связи с контроллером по сети Ethernet/Profinet (заказной номер модуля 6GK7343-1CX10-0XE0).

После такого проведения конфигурирования стойки профильная шина принимает следующий вид (рис. 8). В данном случае она имеет резерв для возможного заполнения 8, 9, 10, 11 слотов дополнительными модулями. Дополнительные стойки расширения ввода-вывода не использованы, поэтому слот №3 остается пустым. Далее на данном этапе можно выбрать (задать) необходимые параметры модулей, то есть, провести параметрирование (например, задать начальные адреса) путем вхождения в разделы их свойств за счет двойного щелчка на каждом из наименований устройств в слотах профильной шины.



Slot	Module
1	PS 307 2A
2	CPU 313C-2 DP
2.2	DP
2.4	Count
3	
4	AI4/AO2
5	DO8xRelay
6	DI8/DO8x24V/0.5A
7	CP 343-1 Lean
8	
9	
10	
11	

Рисунок 8 –
Профильная шина после
конфигурирования PLC

5.2 Конфигурирование децентрализованной периферии

Для конфигурирования децентрализованной периферии, а именно станции удаленного ввода-вывода ET 200M и модуля связи DP/AS-I 20E, вначале необходимо активировать (создать) подсеть PROFIBUS-DP. Активация

выполняется за счет двойного щелчка мышкой на master-устройстве DP (в нашем случае CPU 313C-2DP со встроенным интерфейсом DP) в открывшемся окне свойств модуля. После этого выбрать указанные модули (для ET 200M интерфейсный модуль IM153-1) из встроенного в STEP7 каталога оборудования и мышкой подключить их к появившейся шине PROFIBUS. Результат проведенных действий показан на рис. 9.

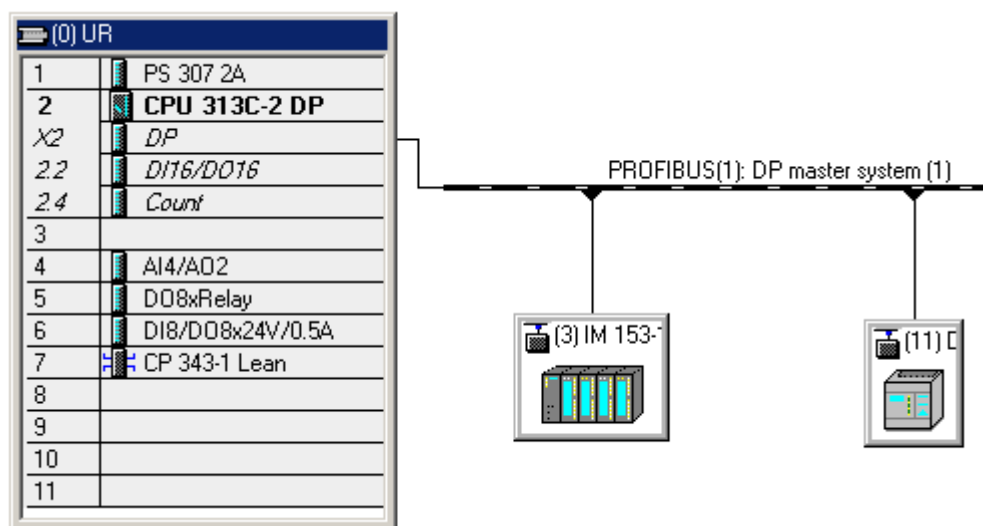


Рисунок 9 – Конфигурация контроллера и децентрализованной периферии в Окне станции

Проведем конфигурирование станции удаленного ввода-вывода ET 200M (ее составных частей). Для этого надо щелкнуть мышкой на пиктограмме станции и заполнить всплывшее окно с профильной шиной, показанной на рис. 3.16. Структура заполнения слотов аналогична рассмотренной для центральной стойки. Однако, имеются некоторые отличия.

Слот №1 – обычно резервируется для установки блока питания интерфейсного модуля IM153-1. Однако в нашем случае интерфейсный модуль подключен к блоку питания, который не установлен на профильной шине. Поэтому первый слот оставляем пустым.

Слот №2 – заполняем интерфейсным модулем IM153-1 для ET 200M (заказной номер модуля 6ES7153-1AA03-0XB0).

Слот №3 – остается пустым, так как в стенде интерфейсный модуль IM для дополнительных стоек расширения ввода-вывода не используется.

Слот №4 – заполняем имитационным модулем входных и выходных дискретных сигналов SM 374 (заказной номер модуля 6ES7374-2XH01-0AA0) аналогично слоту №6 для рассмотренной центральной стойки.

Слот №5 – заполняем модулем вывода дискретных сигналов (релейные выходы) SM 322 (заказной номер модуля 6ES7322-5HF00-0AB0).

В результате получим заполненную оборудованием профильную шину для ET 200M (рис. 10). Слоты № 6, 7, 8, 9, 10, 11 зарезервированы для установки дополнительных модулей в случае модернизации стенда.

Slot	Module	Order Number	I Address	Q Address	Comment
1					
2	IM 153-1	6ES7 153-1AA03-0XB0	1021*		
3					
4	DI8/DO8x24V/0.5A	6ES7 323-1BH00-0AA0	0	0	
5	DO8xRelay	6ES7 322-5HF00-0AB0		2	
6					
7					
8					
9					
10					
11					

5.3 Конфигурирование модуля связи

Конфигурирование

модуля связи DP/AS-I 20E

производится путем

заполнения конфигураци-

онной таблицы для

Рисунок 10 – Профильная шина для ET 200M


ведомых устройств AS-интерфейса из каталога аппаратуры. Количество ведомых AS-I устройств не должно превышать 31. Пример такой конфигурационной таблицы приведен на рис. 11.

AS-i addr.	Module	I Address	Q Address	IO code	ID code	ID1 code	ID2 code	Comment
	ETER DP/AS-i	1019*						
	DP/AS-i	9..40	9..40					
1A	DP/AS-i Link 20E							

Рисунок 11 – Конфигурационная таблица для AS-I устройств

Более подробное описание работы с AS-интерфейсом содержится в подразделах 3.3.3 (пункт 3.3.3.3) и 3.3.4.

5.4 Назначение Profibus-адресов устройствам конфигурируемой схемы

Завершающим этапом процесса программного конфигурирования оборудования является назначение каждому из абонентов сетей MPI, PROFIBUS, ETHERNET его собственного уникального (не повторяющегося) адреса. Для решения этой задачи находит применение стандартное инструментальное средство *Net Pro* (конфигуратор связей). Вызов приложения Net Pro производится нажатием кнопки  на панели инструментов активного окна Hardware Configuration. Адрес абонента задается в интерфейсном окне свойств каждого модуля.

На этом этапе добавим станцию HMI в созданный проект STEP7, подключим ее к шине PROFIBUS и назначим соответствующий адрес абонента. Операция производится вручную аналогично введению станции S7-300. В ходе добавления, как и прежде, необходимо выбрать тип оборудования. В нашем случае это операторская панель OP 177B color PN/DP. В итоге станция HMI появится в иерархической структуре проекта (рис. 12), а также в поле проектирования связей приложения Net Pro (рис. 13).

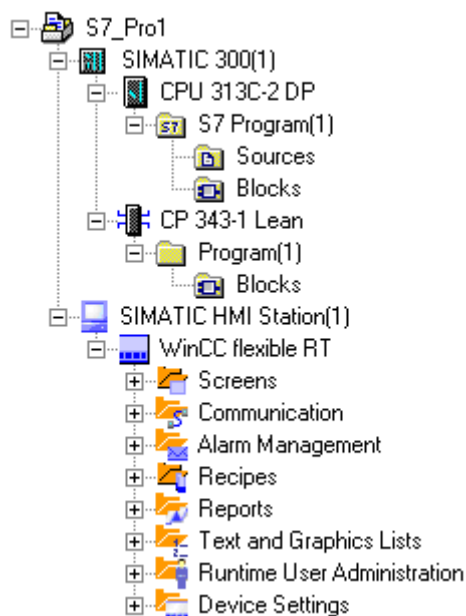


Рисунок 12 – Иерархия проекта

Теперь назначим адреса шины PROFIBUS абонентам этой сети следующим образом:

1. Персональный компьютер – адрес "0";
2. Центральный процессор CPU – адрес "2";
3. Станция ET 200M – адрес "3";
4. Операторская панель – адрес "4";
5. Модуль связи DP/AS-I 20E – адрес "11".

Фрагмент интерфейсного окна приложения Net Pro с отображением сетевой конфигурации оборудования станда приведен на рис. 13.

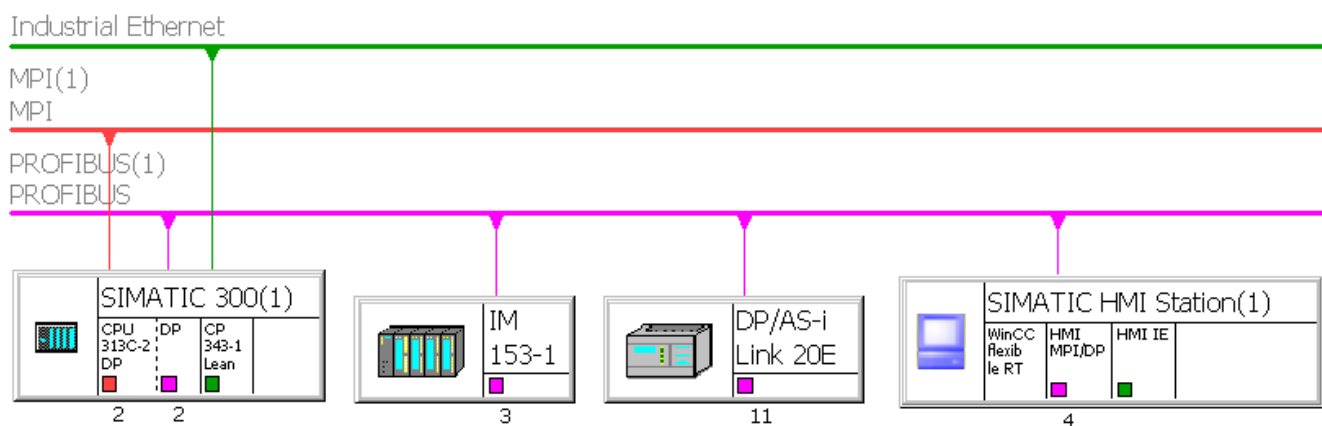


Рисунок 13 – Отображение сетевой конфигурации оборудования в Net Pro

Так, подсети MPI и PROFIBUS в окне приложения создаются в ходе конфигурирования центрального процессора CPU 313C-2DP. Подсеть Industrial Ethernet создадим в разделе свойств коммуникационного процессора CP 343-1 LEAN также при его конфигурировании. Вхождение в окно свойств каждого модуля выполняется двойным щелчком на соответствующем его изображении в окне Net Pro. Здесь же можно подключить (отключить) выбранный модуль к указанным шинам.

Подключение персонального компьютера к шине PROFIBUS реализуется через PC USB адаптер (заказной номер 6ES7972-0CB20-0XA0). Данная связь не является обязательной для конфигурирования, поэтому на рис. 13 она не показана.

Кроме этого, для устройств ET 200M, OP 177B, DP/AS-I 20E необходимо установить соответствующие физические адреса PROFIBUS ("3", "4", "11") непосредственно в самих модулях, то есть за счет переключения тумблеров (или кнопок) на этих модулях. Более подробно техника задания адресов описана в подразделах 3.3.4, 3.3.6, 3.3.7, 3.3.8.

Таким образом, выполнено конфигурирование оборудования центральной стойки станции S7-300, станции удаленного ввода-вывода ET 200M, а также модуля связи DP/AS-I 20E для дальнейшего подключения ведомых AS-I устройств. Разработку управляющей программы и проекта визуализации рассмотрим на следующем примере.

6 Разработка проекта автоматизации работы насосов

Постановка задачи на проектирование

Пусть имеется бак, объемом $V_{\max}=1000$ л, предназначенный для наполнения жидкостью (рис. 14). Наполнение обеспечивает НАСОС №1, приводимый в действие асинхронным двигателем М1. Опустошение обеспечивает НАСОС №2, приводимый в действие асинхронным двигателем М2. Ручное управление насосами должно производиться дистанционно (по сети) с экрана операторской панели. Включение насоса №1 должно выполняться путем перемещения тумблера "НАПОР" в положение "Включено", а выключение – в положение "Выключено". Управление насосом №2 должно выполняться аналогично с помощью тумблера "РАСХОД".

Одновременно могут работать оба насоса при нахождении уровня жидкости в пределах: $0 < V_{\text{тек}} < 1000$. Но как только бак заполнится полностью должен автоматически отключиться двигатель М1, а при полном опустошении бака должен автоматически отключиться двигатель М2. Текущий уровень жидкости в баке должен отображаться на экране операторской панели.

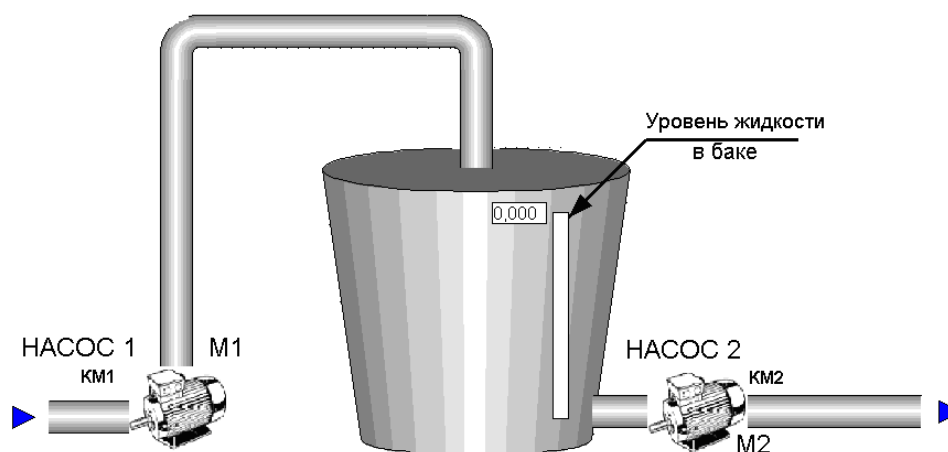


Рисунок 14 – Иллюстрация задачи автоматизации

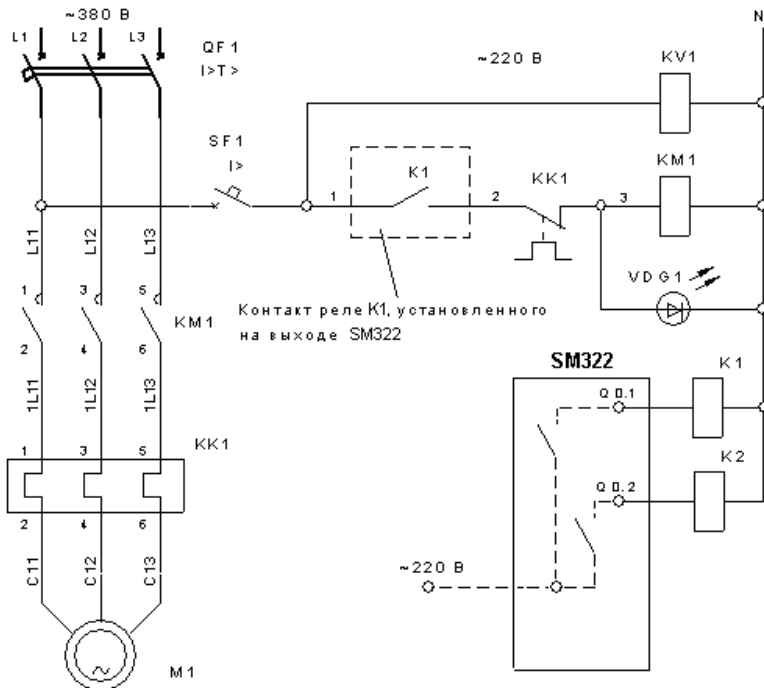
Для решения поставленной задачи (при ранее сконфигурированных PLC, периферии и др.) необходимо:

1. Составить электрическую схему подключения электрооборудования насосов к модулям контроллера SIMATIC S7-300;
2. Провести параметрирование задействованных модулей;

3. Разработать управляющую программу для автоматизации насосов;
4. Разработать проект визуализации для операторской панели в соответствии с поставленной задачей.

6.1 Подключение электрооборудования насосов к модулям контроллера

На рис. 15 представлена принципиальная электрическая схема включения электропривода насоса №1.



Асинхронный двигатель М1 с короткозамкнутым ротором питается трехфазным переменным напряжением 380В. Включение и отключение двигателя осуществляется пускателем КМ1. Защита двигателя от перегрузки осуществляется с помощью теплового реле КК1. Функции защиты от перегрузки и короткого замыкания, а также включение и отключение схемы

Рисунок 15 – Принципиальная электрическая схема насоса №1

от питающей сети выполняет автоматический выключатель QF1. Функции включения, отключения и защиты схемы управления от короткого замыкания выполняет автоматический выключатель SF1. Реле KV1 предназначено для контроля напряжения в цепи управления. Светодиод VDG1 сигнализирует о том, работает насос или отключен. Включение двигателя М1 насоса осуществляется путем замыкания контакта промежуточного реле К1, питающая катушка которого подключена к модулю релейных выходов SM322 станции ET 200М, а именно к клемме с адресом Q 0.1. Можно было бы не добавлять реле К1 в электрическую схему, но тогда на месте контакта К1 должен был бы находиться внутренний контакт модуля SM322.

Аналогичную схему и описание имеет электропривод насоса №2. Питающая катушка промежуточного реле K2 в его схеме подключена к модулю релейных выходов SM322 станции ET 200M к клемме с адресом Q 0.2. Модуль SM322 имеет 8 релейных выходов и с его помощью можно управлять группой из 8 насосов. Поскольку в схеме задействовано только два выхода, то остальные на рис. 15 не показаны.

6.2 Параметрирование модуля SM322

Перед написанием управляющей программы необходимо задать начальный адрес модуля SM322 в разделе его свойств приложения Hardware Configuration. Для удобства начальный адрес выберем равным нулю, тогда адресация выходов модуля будет следующей: Q0.0, Q0.1, Q0.2, Q0.3 и тд.

6.3 Разработка управляющей программы для автоматизации работы насосов

С использованием пакета STEP 7, выполним программную реализацию заданной логики управления работой электроприводов насосов. Управляющую программу составим на языке STL непосредственно в организационном блоке OB1 или создадим новую функцию.

Поскольку управление насосами осуществляет контроллер SIMATIC S7-300, а элементы индикации и ручного управления рабочим процессом находятся на экране операторской панели, то для реализации обмена данными между этими устройствами необходимо использование блоков данных DB в составе PLC.

Создадим блок данных DB1 для элементов управления насосами. Для этого в правой панели диалогового окна SIMATIC Manager (рис. 3.10) вставим соответствующий блок и откроем его. В результате появится таблица, показанная на рис. 16. Заполним ее, впечатав названия переменных, их типы и начальные значения, а для удобства дальнейшего использования запишем комментарии.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	MotorON	BOOL	FALSE	Вкл.
+0.1	Napor	BOOL	FALSE	Вкл.напора
+0.2	Rashod	BOOL	FALSE	Вкл.расхода
=2.0		END_STRUCT		

Рисунок 16 – Блок данных DB1 элементов управления насосами

В первую ячейку памяти с адресом DB1.DBX0.0 (на рис. 16 обозначена +0.0) запишем бит MotorON разрешения управления насосами (может не использоваться). В ячейку памяти с адресом DB1.DBX0.1 запишем бит Napor включения насоса №1 – НАПОР. В ячейку с адресом DB1.DBX0.2 запишем бит Rashod включения насоса №2 – РАСХОД. Эти биты как булевы переменные в ходе работы системы будут принимать два фиксированных значения: или логический нуль или логическую единицу. Их начальное значение – false, то есть нуль.

Аналогичным образом создадим блок данных DB2 для элементов индикации работы насосов.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Q00	BOOL	FALSE	Сигнализация вкл.
+0.1	Q01	BOOL	FALSE	Сигнализация + Вкл.напора
+0.2	Q02	BOOL	FALSE	Сигнализация + Вкл.расхода
+2.0	LEVEL	REAL	0.000000e+000	Уровень жидкости
+6.0	LEVEL1	REAL	0.000000e+000	Уровень жидкости промеж.
=10.0		END_STRUCT		

Рисунок 17 – Блок данных DB2 элементов индикации

В ячейку памяти с адресом DB2.DBX0.0 запишем бит сигнализации разрешения управления насосами (может не использоваться). В ячейки с адресами DB2.DBX0.1 и DB2.DBX0.2 запишем биты Q01 и Q02 сигнализации состояния (включен/выключен) насосов. Их названия соответствуют названиям клемм блока SM322, показанным на рис. 17. В ячейку с адресом DB2.DBX2.0 запишем переменную LEVEL в формате Real – вещественную переменную, соответствующую значению уровня жидкости в баке. В эту ячейку в ходе работы системы будет записываться текущее значение уровня жидкости в формате числа

с плавающей точкой. Ее начальное значение зададим нулевым. Переменную LEVEL1 можно задать аналогично.

Для написания управляющей программы автоматизации работы насосов используем следующие команды языка STL:

"A" – команда логического И;

"L" и "T" – команды загрузки и передачи;

Команды загрузки (L) и передачи (T) позволяют программировать обмен информацией между модулями ввода или вывода и областями памяти CPU или собственно между областями памяти. CPU выполняет эти команды в каждом цикле как безусловные команды, то есть результат логической операции (RLO) на них не влияет.

Способ обмена: команды L и T производят обмен информацией через аккумулятор. Команда L записывает (загружает) содержимое своего исходного адреса в аккумулятор 1, сдвигая всю уже содержащуюся там информацию в аккумулятор 2. Старое содержимое аккумулятора 2 заменяется. Команда T копирует содержимое аккумулятора 1 и записывает его в соответствующую целевую память.

"ТАК" – команда обмена содержимого аккумуляторов 1 и 2;

"+R" и "-R" – арифметические команды сложения и вычитания вещественных чисел, содержащихся в аккумуляторах 1 и 2;

Так, команда сложения ("+R") складывает содержимое аккумулятора 1 с содержимым аккумулятора 2 и результат сохраняет в аккумуляторе 1. Команда вычитания ("-R") вычитает содержимое аккумулятора 1 из содержимого аккумулятора 2 и результат сохраняет в аккумуляторе 1.

">R" и "<R" – команды сравнения вещественных чисел;

Для применения команд сравнения необходимо загрузить числовые значения в аккумуляторы 1 и 2. Команда "Сравнить" сравнивает значение в аккумуляторе 1 со значением в аккумуляторе 2. Результатом сравнения является двоичная цифра, то есть 1 или 0:

– 1 указывает, что результат сравнения является истиной;

– 0 указывает, что результат сравнения является ложью.

Этот результат сохраняется в бите результата логической операции (RLO) и его можно использовать в программе для дальнейшей обработки.

"JU" – команда безусловного перехода;

Команда "Перейти безусловно" (JU) прерывает в программе нормальную последовательность логического управления и заставляет программу перейти на метку (операнд команды JU). Метка отмечает точку, с которой программа должна продолжить свою работу. Переход выполняется независимо от каких-либо условий.

"JCN" – команда перехода, если результат логической операции RLO =0;

Для написания собственно программы, прежде всего, войдем в интерфейсное окно приложения LAD/STL/FBD за счет двойного щелчка на изображении организационного блока OB1 или созданной функции в правой

```
//Расход
A   DB1.DBX   0.2
JCN a1
L   DB2.DBBD  2
L   1.000000e-001
-R
L   0.000000e+000
>R
JCN a2
ТАК
JU  a2
a1: L   DB2.DBBD  2
    T   DB2.DBBD  6
a2: T   DB2.DBBD  6
//Напор
A   DB1.DBX   0.1
JCN a3
L   DB2.DBBD  6
L   1.000000e-001
+R
L   1.000000e+003
<R
JCN a4
ТАК
JU  a4
a3: L   DB2.DBBD  6
    T   DB2.DBBD  2
a4: T   DB2.DBBD  2
```

Рисунок 18 – Программа на языке STL

панели диалогового окна SIMATIC Manager (рис. 3.10). В открывшемся окне составим программу, имитирующую наполнение и опустошение бака в соответствии с рис. 18 и управляющую этим процессом.

Процесс наполнения емкости начинается с момента записи "1" в ячейку DB1.DBX0.1 и происходит (имитируется в данной программе) за счет прибавления величины 0,1 к текущему значению уровня жидкости (используется команда "+R"). Поскольку CPU выполняет программу циклически, то с каждым циклом значение переменной, соответствующей уровню жидкости в баке, будет увеличиваться. Двигатель M1 будет работать до тех пор, пока не нарушится условие $V_{тек} < 1000$.

Опустошение емкости происходит за счет вычитания величины 0,1 из текущего значению уровня жидкости (используется команда "-R"). Минимальный уровень контролируется командой сравнения "<R". Двигатель M2 будет работать, пока не нарушится условие $V_{тек} > 0$.

6.4 Разработка проекта визуализации для операторской панели

Проект визуализации для операторской панели выполним в пакете WinCC flexible. Рабочее пространство данного пакета содержит изображение операторской панели и состоит из областей (элементов), приведенных на рис. 19. Как и в STEP7, доступ к отдельным приложениям осуществляется из *окна проекта* путем выделения необходимого подраздела.

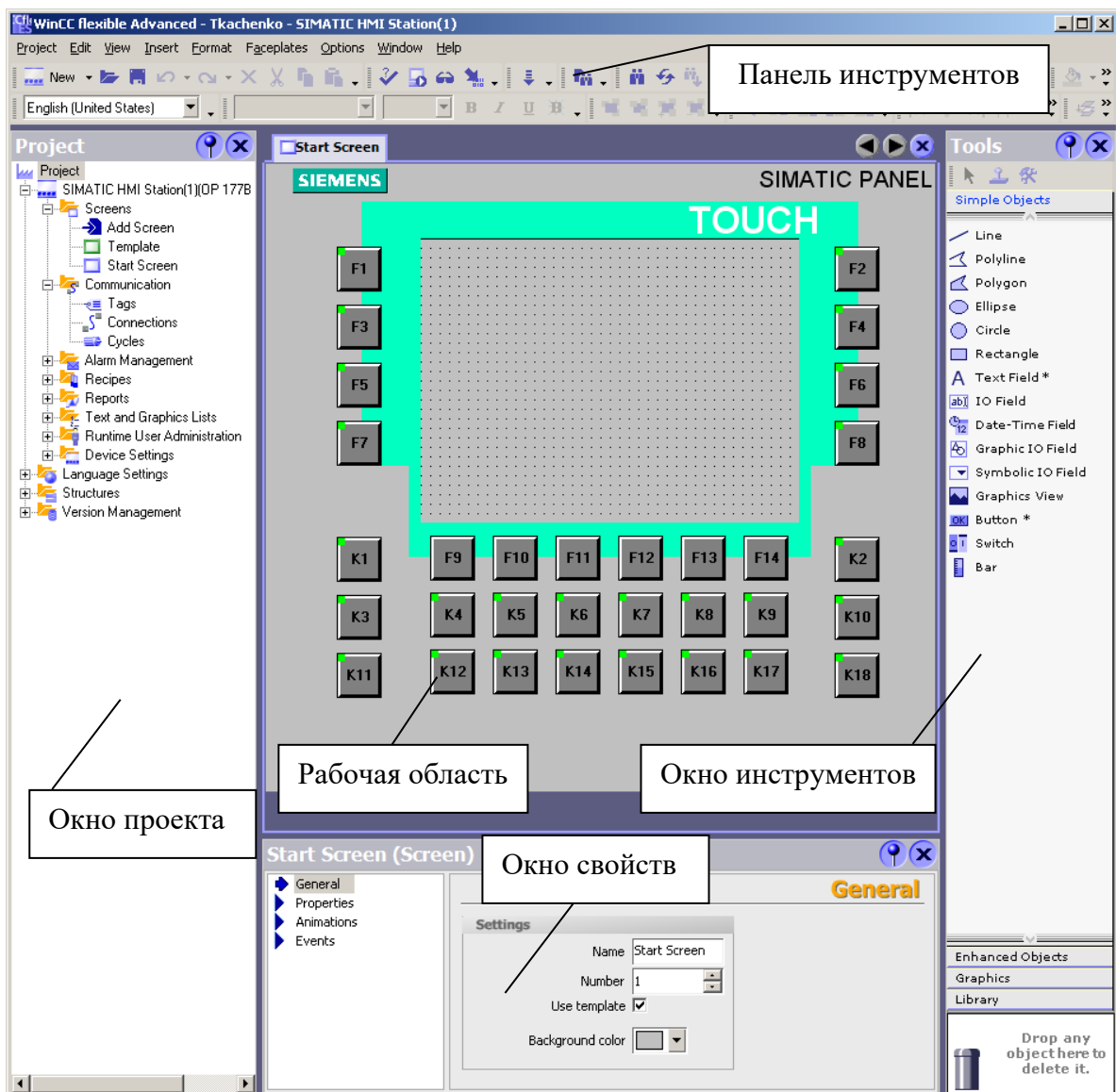


Рисунок 19 – Рабочее пространство пакета WinCC flexible

Для выполнения проекта визуализации необходимо пройти следующие этапы:

1. Создать новый проект;
2. Спроектировать соединение между станциями HMI и S7-300;
3. Определить теги;
4. Создать экран процесса для управления механизмами.

Создание проекта: поскольку станция HMI типа OP 177B была включена в проект STEP7 в процессе конфигурирования оборудования, то проект для WinCC flexible был создан автоматически под тем же именем. Поэтому его создавать не надо, а следует вызвать для доработки.

Проектирование соединения: прежде всего, спроектируем соединение между станцией S7-300 и операторской панелью. Для этого в окне проекта выберем элемент *Connections*. В открывшемся контекстном меню, показанном на рис. 20, выберем новое соединение, например *Connection_1* и активируем его, выбрав *On*. Из столбца *Communication driver* выберем драйвер связи, соответствующий используемому контроллеру, в данном случае SIMATIC S7 300/400. Далее в колонке *Partner* выберем тип станции, с которой операторская панель будет производить связь, то есть центральный процессор CPU 313C-2DP, и тип узла подключения – DP. В результате получим параметры соединения, приведенные на рис. 20.

Name	Active	Communication driver	Station	Partner	Node	Online
Connection_1	On	SIMATIC S7 300/400	\NASOS_2PUS...	CPU 313C-2 DP	DP	On

Рисунок 20 – Проектирование соединений

Схематическое представление коммуникационных партнеров и детальные параметры спроектированного соединения доступны в окне *Parameters* (рис. 21), которое становится доступным при проектировании описанного соединения и отображается в окне свойств станции HMI.

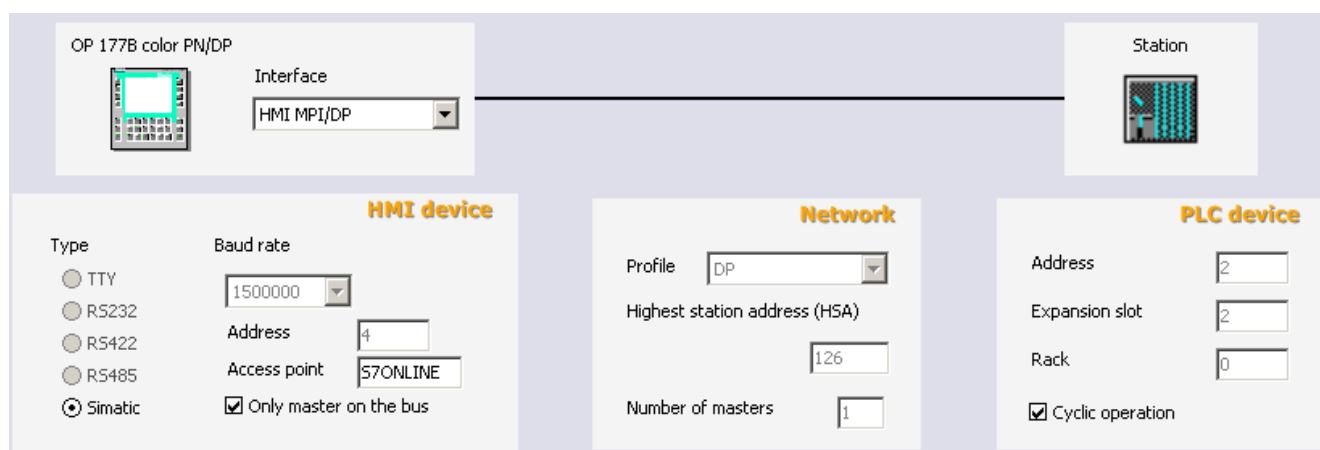


Рисунок 21 – Параметры соединения

Определение тегов: для передачи данных из области памяти контроллера в WinCC flexible и назад необходимо определить внешние теги – переменные процесса. Для этого в окне проекта выберем элемент *Tags*. В открывшемся редакторе тегов создадим теги включения насосов и сигнализации их работы, а также тег уровня жидкости в баке. Их названия должны совпадать с названиями переменных, заданными на рис. 16, рис. 17. Окно редактора и созданные в нем теги показаны на рис. 22.

Name	Connection	Data type	Symbol	Address	Array count	Acquisition cycle	Comment
DB1.Napor	Connection_1	Bool	Napor	DB 1 DBX 0.1	1	100 ms	Вкл.напора
DB1.Rashod	Connection_1	Bool	Rashod	DB 1 DBX 0.2	1	100 ms	Вкл.расхода
DB2.LEVEL	Connection_1	Real	LEVEL	DB 2 DBD 2	1	100 ms	Уровень жидкости
DB2.Q01	Connection_1	Bool	Q01	DB 2 DBX 0.1	1	100 ms	Сигнализация + Вкл.напора
DB2.Q02	Connection_1	Bool	Q02	DB 2 DBX 0.2	1	100 ms	Сигнализация + Вкл.расхода

Рисунок 22 – Окно редактора тегов

Так в меню *Name* вводят имена каждого из тегов с учетом названия блока данных PLC (например, DB1.Napor), в меню *Connection* выбирается спроектированное ранее соединение Connection_1. *Data type* определяет тип значений, которые будут храниться в теге (например, Bool или Real в соответствии с рис. 16, рис. 17). В контекстных меню *Symbol* и *Address* высвечивается информация (в частности, имя ячейки памяти и ее адрес) о созданных ранее блоках данных DB1 и DB2, которую можно использовать для задания имени тегов. Это упрощает процедуру определения тегов и исключает ошибки при вводе. *Acquisition cycle* определяет промежуток времени (зададим 100

мс.), через который устройство НМІ будет считывать значения процесса из внешних тегов. В меню *Comment* для каждого тега запишем комментарии.

Создание экрана процесса: завершающим этапом создания проекта визуализации является разработка дизайна экрана процесса (то есть удобного изображения управляемых устройств и управляющих "кнопок") путем добавления и редактирования элементов, необходимых для отображения. Разработка дизайна экрана выполняется в редакторе экранов. Для создания нового экрана необходимо выбрать элемент *Add Screen* в разделе *Screens* Окна проекта. В результате откроется рабочая область с новым экраном, показанная на рис. 23.

Теперь из Окна инструментов (рис. 19) можно выбрать необходимые графические объекты (для отображения технологического процесса и оперативного влияния на него) и переместить их в область с экраном.

Для включения насоса №1 выберем ползунковый переключатель Switch, представленный на рис. 23. Этот объект поместим в левую часть экрана и зададим его свойства (рис. 24). В первую очередь определим его связь с тегом DB1.Napor включения насоса №1. Далее укажем маркировку НАПОР, то есть, название переключателя, высвечиваемое на экране, а также текстовые обозначения включенного (1) и выключенного (0) состояний. Они же являются числами, пересылаемыми в PLC при "ручном" переключении переключателя.



Рисунок 23 – Переключатель Switch

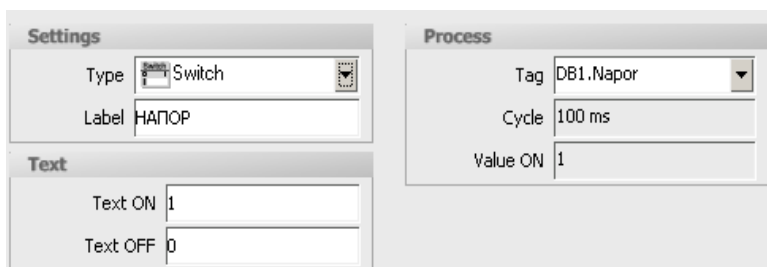


Рисунок 24 – Свойства объекта Switch

Таким образом, после перемещения ползунка вправо, в ячейку памяти с адресом DB1.DBX0.1 будет записана логическая единица и насос №1 "включится". В исходном

положении ползунка насос будет "выключен". Для включения насоса №2 используем аналогичный переключатель и разместим его в правой части экрана.

Аналогичным образом выполним его связь с тегом DB1.Rashod и определим свойства объекта.

Для отображения текущего уровня жидкости в баке используем столбиковый индикатор Bar, представленный на рис. 25.

Выберем этот объект из Окна инструментов, поместим его в центральную часть экрана и зададим свойства (рис. 26). Отображаемое значение индикатора свяжем с тегом уровня жидкости DB2.LEVEL. Определим минимальное отображаемое значение уровня жидкости в баке на уровне 0, а максимальное значение – на уровне 1000.

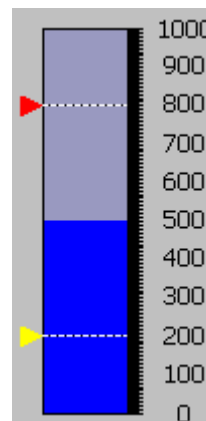


Рисунок 25 –
Столбиковый индикатор
Bar

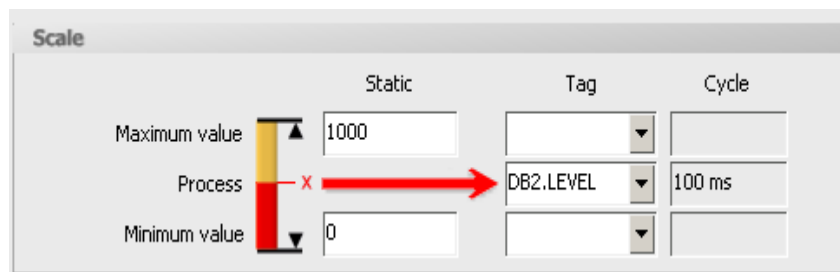


Рисунок 26 – Свойства объекта Bar

Поскольку при создании всех тегов время цикла

опроса было задано 100 мс., то в рассмотренных графических объектах эта величина вводится автоматически по умолчанию. Для отображения значений даты и времени поместим в верхнюю правую часть экрана соответствующее поле Date-Time Field и настроим его, задав текущее значение даты и времени. Таким образом, для визуализации работы насосов получим экран процесса, представленный на рис. 27.

Теперь необходимо загрузить конфигурацию оборудования и управляющую программу для насосов в программируемый контроллер (выбрав *PLC* → *Download* в диалоговом окне SIMATIC Manager, показанном на рис. 3.10), а проект визуализации в операторскую панель (выбрав *Project* → *Transfer* → *Transfer Settings* → *Transfer* в диалоговом окне пакета WinCC flexible, показанном на рис. 19).

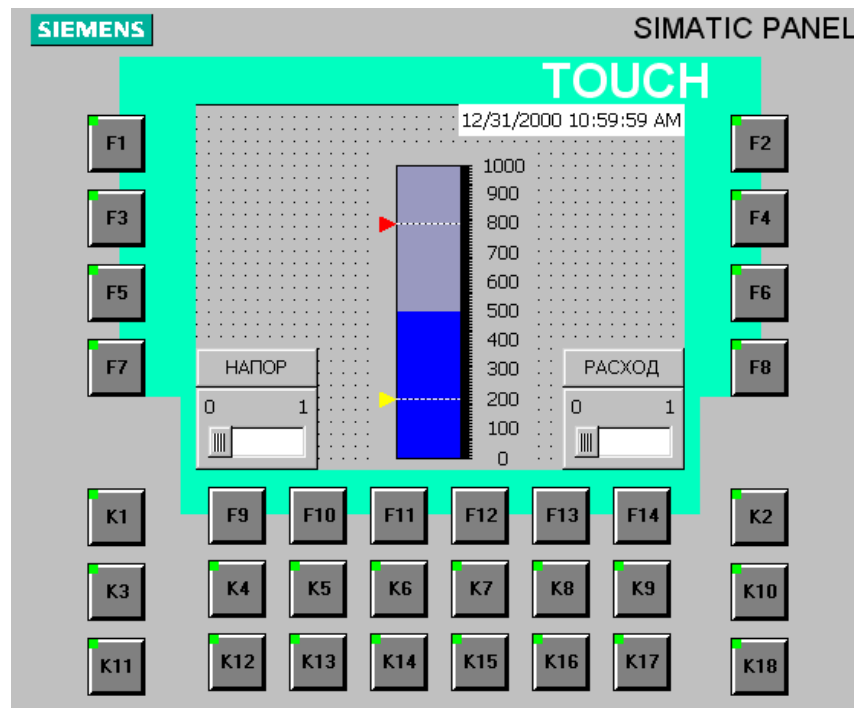


Рисунок 27 – Экран процесса

Снимок экрана функционирующей операторской панели, отображающей в реальном времени состояние бака в соответствии с набранными элементами визуализации и управления показан ниже.



Рисунок 28 – Экран функционирующей операторской панели