**4 РАЗРАБОТКА ПРОГРАММНОЙ ЧАСТИ**

**АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ**

**4.1 Разработка программы ПЛК**

Программа для ПЛК *SIMATIC S*7-1200 разрабатывается в интегрированной среде разработки *TIA Portal* 15 от компании *Siemens AG*. В *TIA Portal* 15 интегрированы следующие компоненты: *Simatic Step* 7 (инструмент для программирования контроллеров), *WinCC* (инструмент для создания *HMI*), *PLCSIM* (симулятор работы ПЛК). Разрабатываемый проект состоит из программы ПЛК и проекта для графической панели оператора (*HMI*).

Стартовое окно *TIA Portal* 15 обеспечивает доступ к основным элементам проекта. «Проект» представляет собой структуру, которая содержат все программы и данные, необходимые для выполнения задачи автоматизации. Через главное окно можно открыть существующий проект, создать новый проект или произвести преобразование проекта.

В разделе *Devices & network* производится конфигурирование контроллеров, зада­ются модули, н производится их настройка.

В разделе *PLC programming* разрабатывается управляющая программа, формируется структура программы, создаются программные блоки.

В разделе *Visualization* собраны инструменты по разработке человеко-машинных ин­терфейсов (*HMI*) с использованием *Simatic WinCC*.

В разделе *Online & Diagnostics* возможно подключение к программируемому кон­троллеру. управление режимами его работы *CPU*. тестирование работы управляющей про­граммы.

Режим *Project view* позволяет отобразить проект в виде иерархической структуры, причем содержимое и вид окна рабочей области может меняться в зависимости от выпол­няемых действий. Окно содержит рабочую область, в которой производится конфигурирование оборудо­вания контроллера, каталог модулей, отображение свойств устанавливаемых модулей.

Если в структуре проекта будет выбран программный блок, то в окне разработки будет показана программа, а вспомогательные окна будут содержать элементы для формирования программы. Такой подход облегчает переключение окон при разработке проекта. Разработчику всегда доступно дерево проекта, а при редактировании необходимого элемента, дополнительные окна с инструментами редактирования появляются автоматически.

Для создания нового проекта *TIA Portal* 15 необходимо выбрать действие *Create new proje*ct (создать новый проект), после чего указать название проекта, задать каталог на диске, в котором будут храниться файлы проекта, и нажать на экране кнопку *Create*.

Следующий шаг ‒ конфигурация используемого аппаратного оборудования ‒ *Devices & Networks/Configure a device*. Необходимо добавить в проект (*Add new device*) модуль процессора *CPU* 1215*С* *DC/ DC/DC* с номером 6*ES*7 215-1*AG*40-0*XB*0. При этом в поле *Device name* оставляем автоматически предложенное имя (PLC\_1) (рисунок 4.1).

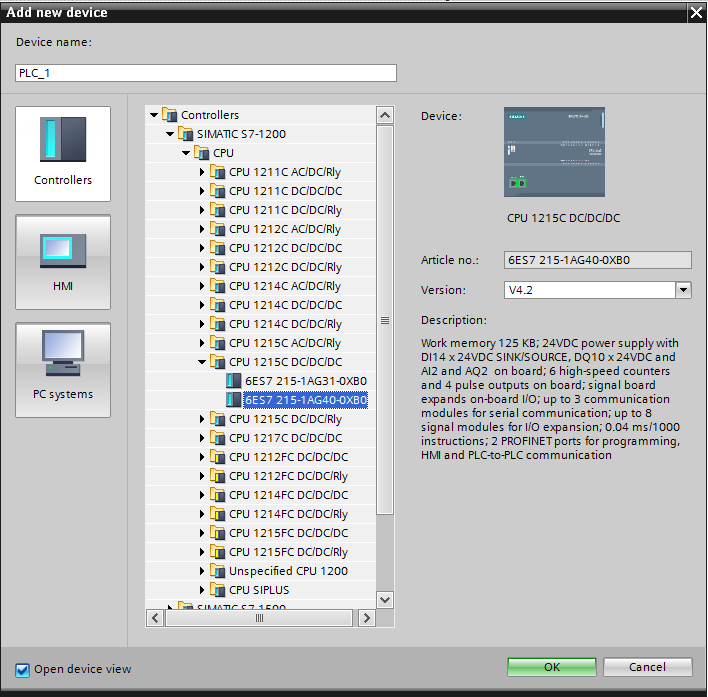


Рисунок 4.1 – Выбор модуля процессора *CPU* 1215*С*

После добавления *CPU* 1215*С* представление проекта автоматически изменится на проектно-ориентированное, в котором в рабочей области окна появится графическое изображение добавленного в проект контроллера. Щелкнув на нем правой кнопкой мыши и выбрав во всплывающем меню пункт *Properties* (свойства), в нижней части экрана в окне инспектора свойств объекта можно просматривать и нужным образом настраивать свойства *CPU*. *CPU* 1215*С* снабжен двумя портами *PROFINET* для обмена данными через сеть *PROFENET*. Для обмена данными через сети *RS*-485 имеется коммуникационный модуль, встраиваемый в *CPU*.

В окне свойств устанавливаются следующие параметры:

* интерфейсы *PROFINET*: установка IP-адресов (PN1: 192.168.0.1)
* *DI*, *DO*, и *AI*: настройка поведения встроенных цифровых и аналоговых входов и выходов;
* время цикла 100 мс.

Далее добавляются из библиотеки модуль ввода аналоговых сигналов *SM*1231 (6ES7 231-4HF32-0XB0). После этого окно конфигурации примет окончательный вид (рисунок 4.2):

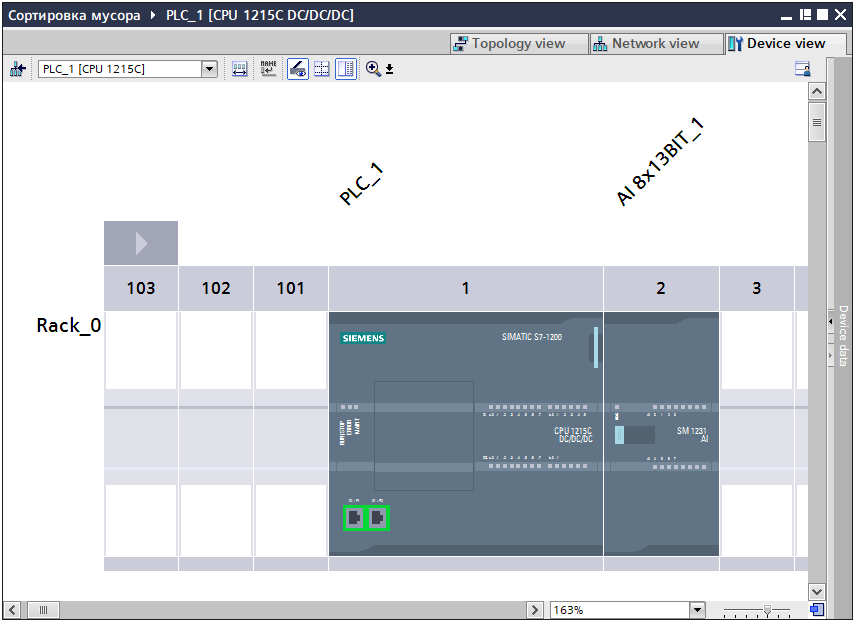


Рисунок 4.2 – Конфигурация контроллера *SIMATIC S*7-1200 для проекта автоматизации в *TIA Portal* 15

Далее в проект добавляются сервоприводы *SINAMICS V*90 и панель оператора. Назначение *IP*-адресов подключений по *PROFINET* происходит автоматически. Версия *PROFINET* привода *SINAMICS V*90 оснащена интерфейсом *PROFINET*, который позволяет в режиме реального времени передавать данные пользователя/процесса и диагностическую информацию по одному кабелю. Данное подключение выбрано, как решение с широким набором функции меньшей сложности для нескольких сервоприводов. *SINAMICS V*90 имеет два интерфейса *PROFINET.* Сервоприводы подключаются к *PROFINET CPU* по шинной топологии.

Скриншот окна *Devices & networks* с настройками после добавления сервоприводов и панели оператора будет иметь следующий вид (см. рисунок 4.3).

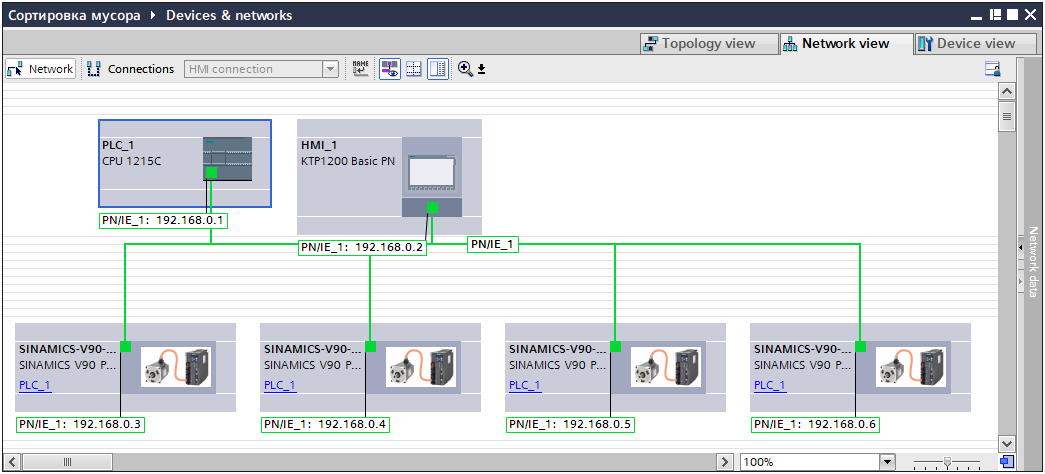


Рисунок 4.3 – Скриншот окна *Devices & networks*

При добавлении устройств автоматически генерируется соединение через *Profinet* и присваивается *IP*-адреса по умолчанию в диапазоне: 192.168.0.2 - 192.168.0.6.

В разрабатываемой программе вызываются и обрабатываются блоки из технологической библиотеки для управления сервоприводами *SINAMICS V*90. Для использования их в проекте необходимо предварительно настроить их конфигурацию при помощи мастера настройки. Для каждого *SINAMICS V*90 используются один независимый экземпляр технологического объекта *Axis*\_1, настаиваемый в *TiaPortal* 15. При этом скорость конвейера будет регулироваться введенными с панели оператора значениями и датчиком скорости.

Окно Мастера настройки технологического объекта *Axis* (см. рисунок 4.4)

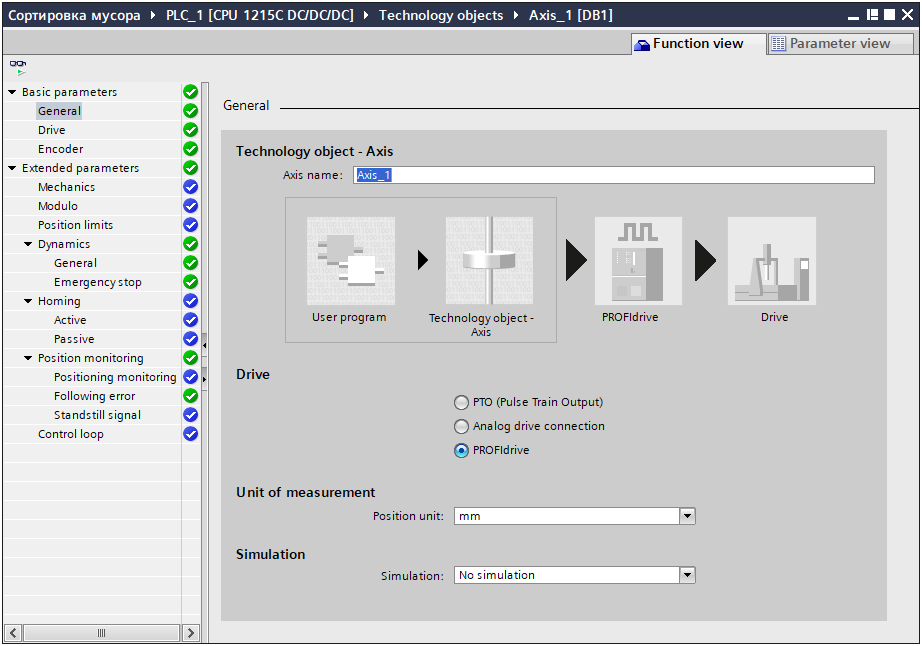


Рисунок 4.4 – Окно Мастера настройки технологического объекта *Axis*

В следующем окне Мастера настройки (см. рисунок 4.5) настраивается конфигурация сервосистемы:

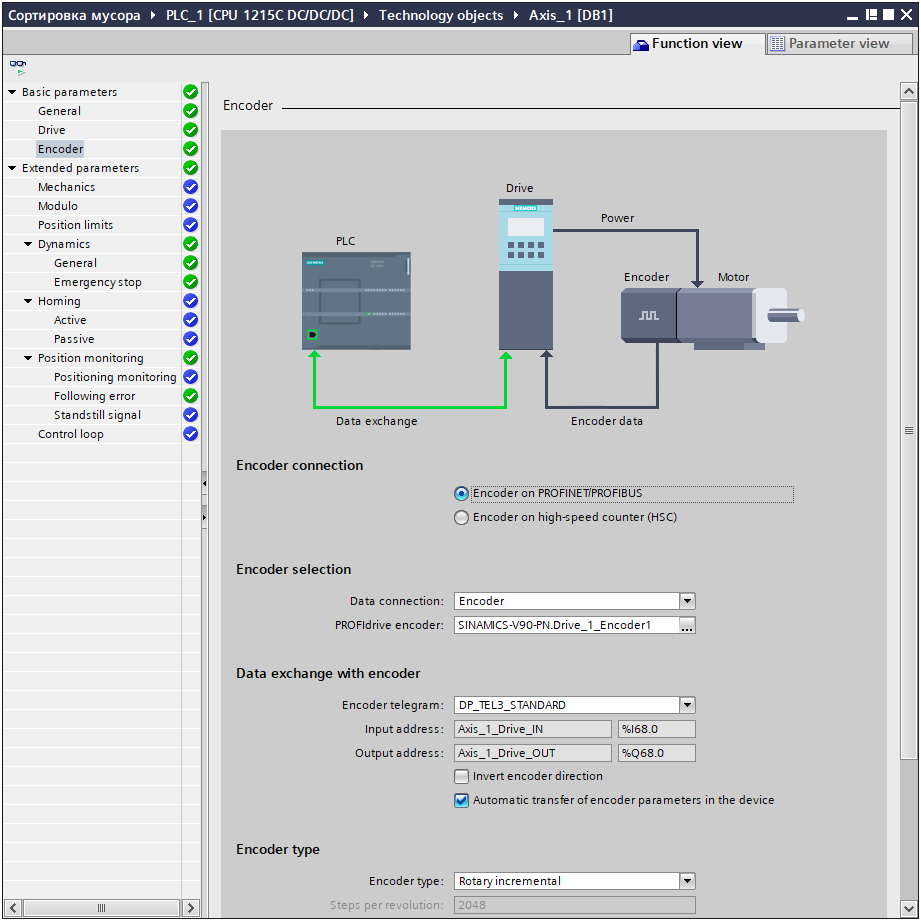


Рисунок 4.5 – Настройка конфигурации сервосистемы

После настроек конфигурации проекта реализуется программный код проекта для ПЛК.

Программа ПЛК по структуре состоит из нескольких блоков. Исполнение программы пользователя *CPU* поддерживает следующие виды блоков, позволяющие создать эффективную структуру пользовательской программы:

* Организационные блоки (*ОВ*) определяют структуру программы. Некоторые *ОВ* имеют предопределенное поведение и стартовые события, но вы можете также создавать *ОВ* со своими собственными стартовыми событиями.
* Функции (*FC*) и функциональные блоки (*FB*) содержат программный код, соответствующий конкретным задачам или комбинациям параметров. Каждая функция и каждый функциональный блок предоставляет в распоряжение набор входных и выходных параметров для совместного использования данных с вызываемым блоком. *FB* использует также связанный с ним блок данных (называемый экземплярным *DB*) для сохранения данных о состоянии во время исполнения, которые могут быть использованы другими блоками в программе.
* Блоки данных (*DB*) хранят данные, которые могут быть использованы программными блоками.

Исполнение программы пользователя начинается одним или несколькими необязательными организационными блоками (*ОВ*), которые после перехода в режим *RUN* обрабатываются один раз, затем следует один или более ОВ программного цикла, которые обрабатываются циклически. ОВ может быть также поставлен в соответствие прерывающем событию, которое может быть стандартным событием или событием-ошибкой; затем он исполняется, когда происходит соответствующее событие.

Функция (*FC*) или функциональный блок (*FB*) − это блок с кодом программы, который может быть вызван из ОВ или из другой функции или другого функционального блока. При этом возможны следующие уровни вложения:

* 16 из циклического *ОВ* или *ОВ* запуска;
* 4 из *ОВ* прерываний с задержкой, *ОВ* циклических прерываний, *ОВ* аппаратных прерываний, *ОВ* ошибок по времени или *ОВ* диагностируемых ошибок *FC* не ставятся в соответствие никакому конкретному блоку данных (*DB*), тогда как *FB* непосредственно связаны с *DB* и используют этот *DB* для передачи параметров и сохранения промежуточных значений и результатов.

Размер пользовательской программы, данных и конфигурации ограничен имеющейся в распоряжении загрузочной памятью и рабочей памятью в *CPU*. В рамках свободной рабочей памяти число поддерживаемых блоков не ограничено. Каждый цикл включает в себя запись выходов, чтение входов, исполнение команд программы пользователя и выполнение обслуживания системы или фоновая обработка. Этот цикл называется также циклом сканирования или просто сканированием. Сигнальная плата, сигнальные и коммуникационные модули обнаруживаются и регистрируются только при запуске.

По структуре программа построена из организационного блока *ОВ*1, блока циклических прерываний *OB*30, стартового блока *OB100*, функциональных блоков масштабирования аналоговых входов, функционального блока оповещения об ошибке, функционального блока управления сервоприводами, функционального блока настроек HMI и функциональных блоков технологической библиотеки для работы с объектами Axis которые вызываются и обрабатываются в организационном блоке *ОВ*1 и в блоке циклических прерываний *OB*30. Главная программа располагается в организационном блоке *ОВ*1, который всегда обрабатывается центральным процессором. Начало пользовательской программы идентично первому сегменту (сети, *network*) в *ОВ*1. По завершению обработки *ОВ*1 (конец программы) *CPU* передает управление операционной системе, и после вызова различных функций операционной системы, таких как обновление образа процесса, центральный процессор снова вызывает *ОВ*1.

Событиями, которые могут вмешиваться в работу программы, являются прерывания *(interrupts*) и ошибки (*errors*). В блоке циклических прерываний *OB*30 с интервалом 10 мс обрабатываются сигналы с аналоговых входов, производиться поддержание давления в пределах 8 бар, отслеживается нажатие кнопки (Stop), срабатывание датчика конвейерной ленты (S1), возникновение ошибки (Fault), отслеживание уровня радиации и запуск функциональных блоков управления двигателями (FB2), HMI (FB3), оповещениями об ошибках (FB4).

Для масштабирования сигналов с датчиков измерения уровня радиации, текущей скорости сервопривода (M1), текущего давления в системе вызывается следующие блоки (см. рисунок 4.6).

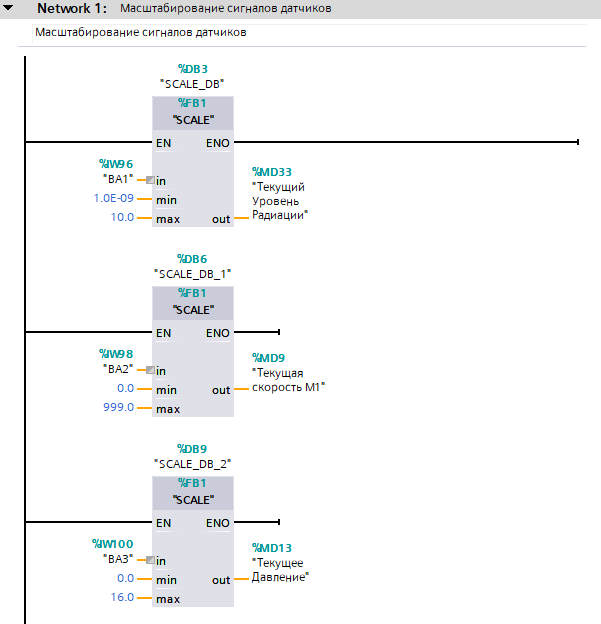


Рисунок 4.6 – Функциональные блоки масштабирования сигналов с датчиков уровня радиации, текущей скорости сервопривода (M1), текущего давления соответственно (*Network 1)*

В программе используется функция масштабирования *SCALЕ*. Она имеет следующий код:

#*out* := #in \* (#*max* - # *min*) / 27648 + # *min*.

Функция *SCALE* принимает целое значение (*in*) и преобразует его в вещественную величину в технических единицах (единицах измерения), масштабированную между нижним и верхним пределом (*min* и *max*). Результат записывается в *out*. Для датчиков с выходным сигналом 4…20 мА в программе используются измеренные и масштабированные значения в соответствующих единицах измерения и рабочем диапазоне. Например, *Уровень радиации* типа *real* на выходе блока при 4 мА на входе имеет значение 10 нЗв/ч. Максимальное значение на выходе *Уровень радиации* = 10 Зв/ч соответствует входному сигналу датчика 20 мА. Аналогично выполняется обработка для остальных аналоговых входов.

Для поддержания давления в системе на уровне в 8 бар используется следующая схема (см. рисунок 4.7).

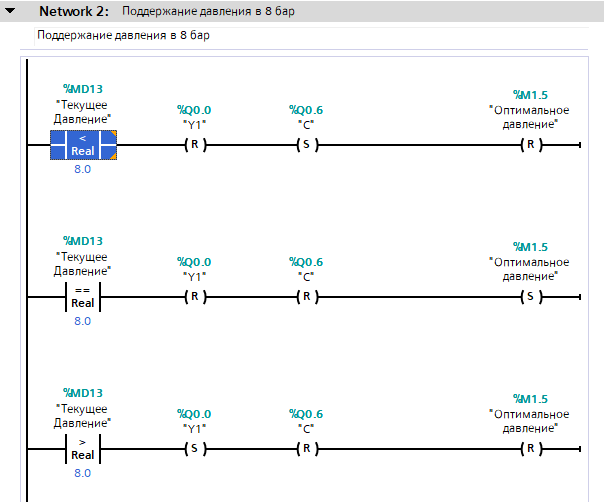


Рисунок 4.7 – Поддержание давления в системе на уровне 8 бар (*Network 2)*

При включении контроллера, запускается компрессор и накачивает сжатый воздух в систему, если давление превысит восемь бар компрессор выключается и включается клапан сброса давления, если давление в системе упадет меньше восьми бар включается компрессор и выключается клапан сброса давления.

Для оповещения о превышении допустимого уровня радиации используется следующая схема (см. рисунок 4.8).

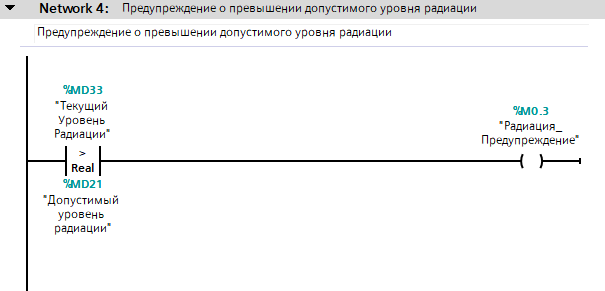


Рисунок 4.8 – Оповещение о превышении допустимого уровня радиации (*Network 4)*

Если значение с датчика радиации превышает допустимый уровень радиации активируется предупреждение.

В блоке *ОВ*1 реализуется вся логика. Согласно разработанному алгоритму, обрабатываются состояния дискретных входов ПЛК, переменных, и команд, поступающих с панели оператора. Программа реализована на языке *LAD*.

После загрузки ПЛК в программе один раз вызывается блок *OB100,* затем циклически вызывается блок *OB*30 и в блоке *ОВ*1 реализуется управление работой установки сортировки отходов.

Для установки значений по умолчанию переменным (ШАГ, Допустимый уровень радиации, установить скорость M1) используется команда MOVE, которая присваивает переменным на выходе (OUT1) значения, введенные на вход (IN) (см. рисунок 4.9).

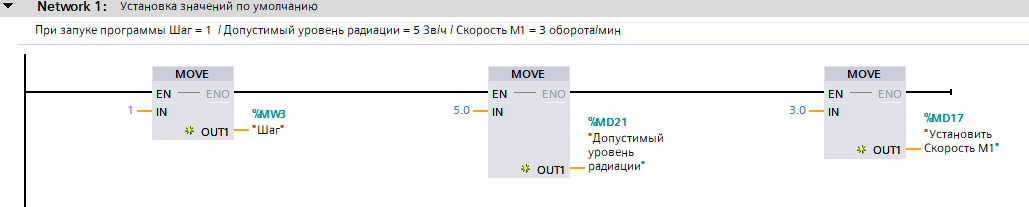


Рисунок 4.9 – Установка значений по умолчанию переменных (*Network 1)*

Для запуска сортировки отходов программа проверяет условия после ввода команды *Start* c сенсорной панели оператора (см. рисунок 4.10).

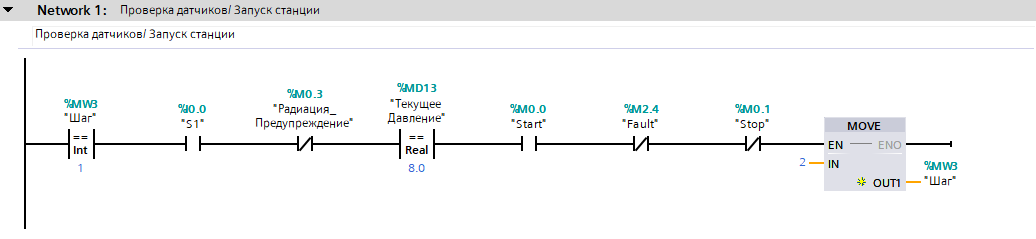


Рисунок 4.10 – Проверка датчиков и отсутствия ошибки перед запуском программы (*Network 1)*

Условия запуска – номер шага равен единице, лента конвейера натянута, радиация в допустимых пределах, давление в системе 8 бар, кнопка стоп не нажата и отсутствуют неисправности (Fault). С HMI на ПЛК поступает команда и устанавливается бит (Start). Переменной (ШАГ) присваивается значение два, и программа переходит на следующий шаг.

Когда переменная (ШАГ) равняется двум, производиться запуск станции разрыва пакетов, сепаратора, сервоприводов (M1, M2, M3, M4) (см. рисунок 4.11).

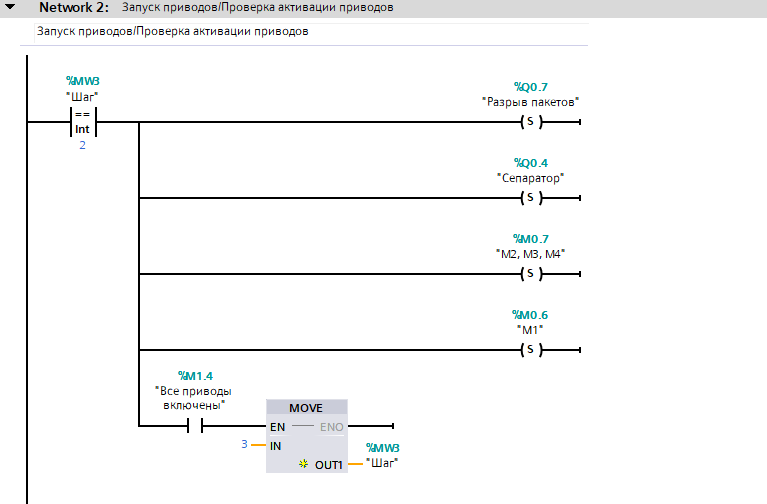


Рисунок 4.11 – Запуск приводов станции сортировки отходов и проверка их активации (*Network 2)*

Проверка активации приводов проводится путем получения обратного сигнала с реле приводов, если хотя бы один привод выключиться после второго шага, программа предупредит о наличии ошибки (см. рисунок 4.12).

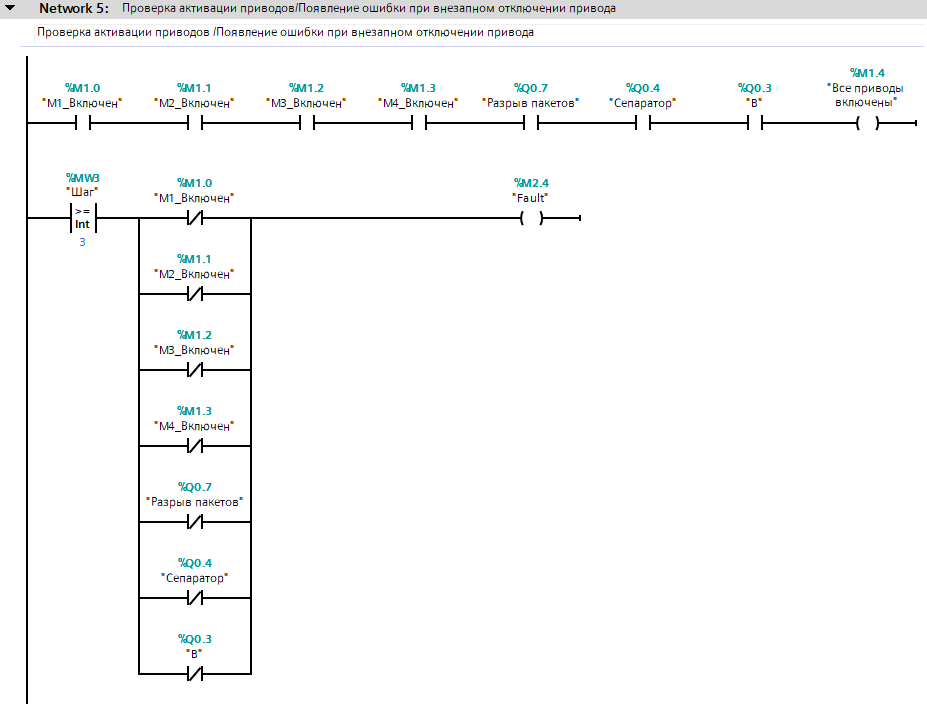


Рисунок 4.12 – Проверка активации приводов станции сортировки отходов (*Network 5)*

Как только все приводы были успешно включены, переменной (ШАГ) присваивается значение три, и программа переходит на следующий шаг.

Когда переменная (ШАГ) равняется трём, происходит проверка достижения оптимальной скорости вращения сервопривода M1 и последующее включение света на рабочих группах (H1, H2) и вентилятора охлаждения ламп (В) (см. рисунок 4.13).

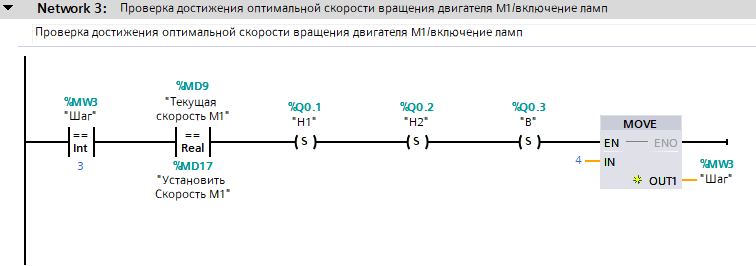


Рисунок 4.13 – Проверка достижения оптимальной скорости сервоприводом M1 и последующее включение ламп и вентилятора (*Network 3)*

Переменной (ШАГ) присваивается значение четыре, и программа переходит на следующий шаг.

Когда переменная (ШАГ) равняется четырём, программа запускает ПО камеры, оператор настраивает параметры сортировки отходов (см. рисунок 4.14).

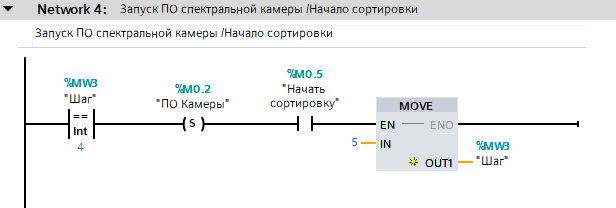


Рисунок 4.14 – Запуск программного обеспечения камеры и запуск начала сортировки (*Network 4)*

После настройки параметров сортировки отходов оператор нажимает на кнопку (Начать сортировку), на бит (M0.5) посылается положительный фронт, переменной (ШАГ) присваивается значение пять и начинается процесс сортировки отходов.

В программе перевод сервоприводов в активное состояние заключается в следующем. Для каждого из объектов *Axis\_1*, в *Motors* вызываются блоки активации оси сервопривода “*MC\_Power*”. На рисунке 4.15 приведен блок активации сервопривода *M*1.

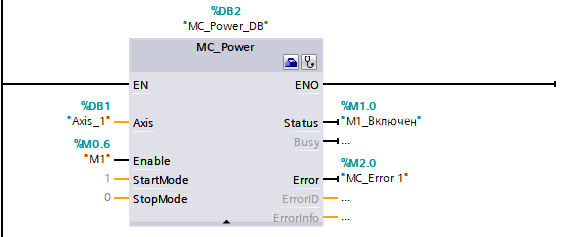


Рисунок 4.15 – Блок активации оси сервопривода *M*1

Аналогичный блок вызывается для всех сервоприводов M2, M3, M4. Перед перемещением оси она должна быть активирована. Если на вход *Enable* блока “MC\_Power“ подается сигнал "*TRUE*", то устанавливается спараметрированный в конфигурации технологического объекта " *Axis\_1*" профиль и сервопривод переходит в режим готовности. Вход „режима останова" указывает, должна ли ось при деактивации быть остановлена со сконфигурированным "аварийным остановом" и после отключена (0) или она должна быть остановлена сразу же (1). На выход "Состояние" от сервопривода поступает подтверждение о готовности, значение заносится в бит (M1.0). Ошибки при работе отображаются на выходе *Error*, значение заносится в бит (M2.0). При наличии квитируемой ошибки, она должна быть сброшена через положительный фронт входа *Execute* на блоке *MC\_Reset* (см. рисунок 4.16).

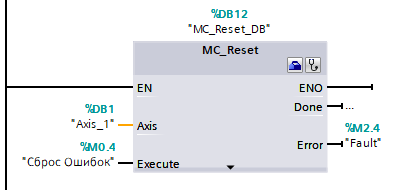


Рисунок 4.16 – Блок *МС\_Reset*

При возникновении ошибки срабатывает определенный бит, отвечающий за свою ошибку в функциональном блоке Alarms (см. рисунок 4.17).

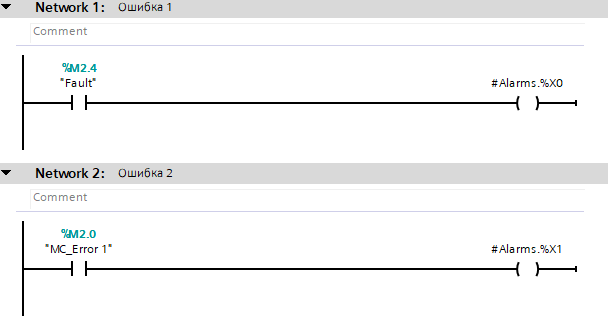


Рисунок 4.17 – Привязка событий к определенной ошибке в блоке Alarms

Этот бит заносит сообщение об ошибке во вкладку *Журнала* на HMI панели. Эта ошибка регистрируется в (Alarm Logs), в определенной папке флеш накопителя HMI панели (см. рисунок 4.18).

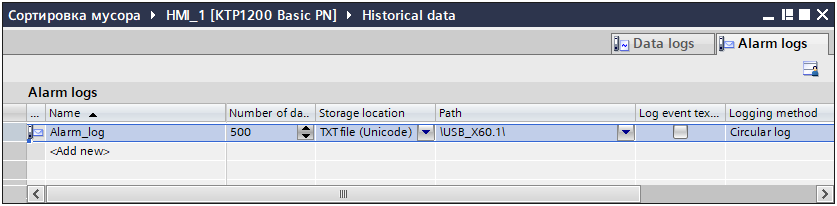


Рисунок 4.18 – Лог файл ошибок

Управление скоростью конвейера (M1) осуществляется в блоке “*MC-MoveVelocity”* который представлен на рисунке 4.19.

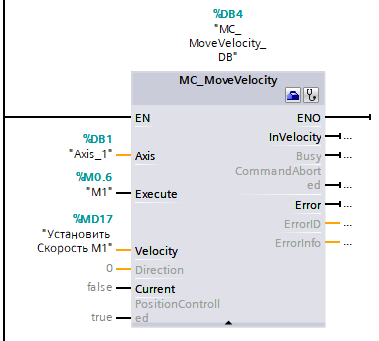


Рисунок 4.19 – Блок *MC-MoveVelocity*

Сохранение внесенных параметров в окне *Настройки* HMI панели реализовано по следующей схеме (см. рисунок 4.20).

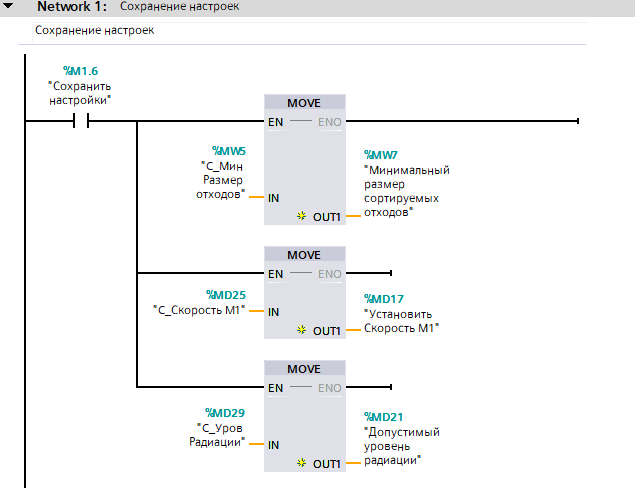


Рисунок 4.20 − Сохранение внесенных через панель HMI параметров

Оператор вносит значение параметра в поле ввода, это значение заносится во временную переменную и только после нажатия на кнопку *Сохранить* они заносятся в необходимую переменную.

Сервоприводы *M*1, *M*2, *M3, M4* отключаются при сбросе бита *Power\_Enable* и процесс завершается отключением сепаратора, ламп рабочих групп, вентилятора, компрессора, остановкой станции разрыва пакетов. Код завершения процесса сортировки отходов приводиться на рисунке 4.21.

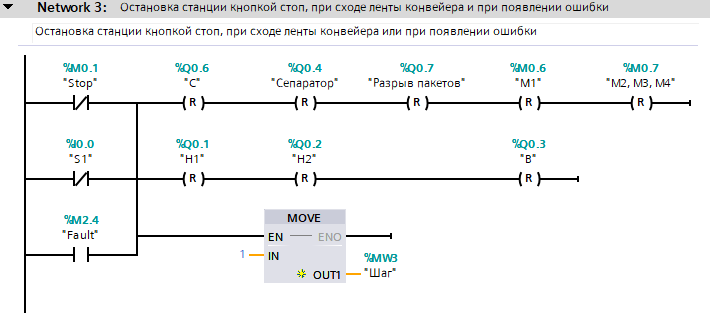


Рисунок 4.21 − Код завершения процесса сортировки отходов (*Network 3)*

Полный код программы на языке *LAD* приводиться в приложении.

**4.2 Разработка проекта для панели оператора**

Создание проекта для панели оператора *SIMATIC HMI KTP*1200 *Basic* *color PN* начинается с ее предварительной настройки при помощи *HMI Device Wizard*. На первом шаге задается интерфейс связи с ПЛК (см. рисунок 4.22).

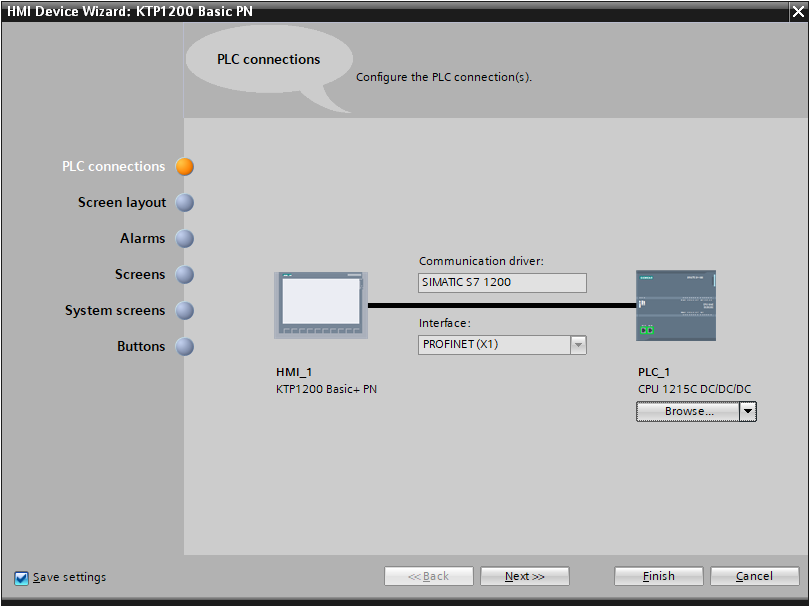


Рисунок 4.22 – Настройка соединения с ПЛК

Для подключения *SIMATIC HMI KTP*1200 *Basic color PN* используется интерфейс *Profinet(X1)*.

При создании графического интерфейса *HMI Device Wizard* предлагает создать заранее шаблон с необходимыми элементами управления. Данном случае выбирается отображение в верхней части экрана заголовка и даты/времени. Задается также разрешение экрана 640x480.

По структуре проект будет состоять из четырех экранов: Главный, Настройки, Мнемосхема, Журнал. Они выполняют служебную функцию для наладки и сервисного обслуживания системы. Окно задания экранной навигации (*Screen navigation*) представлено на рисунке 4.23.

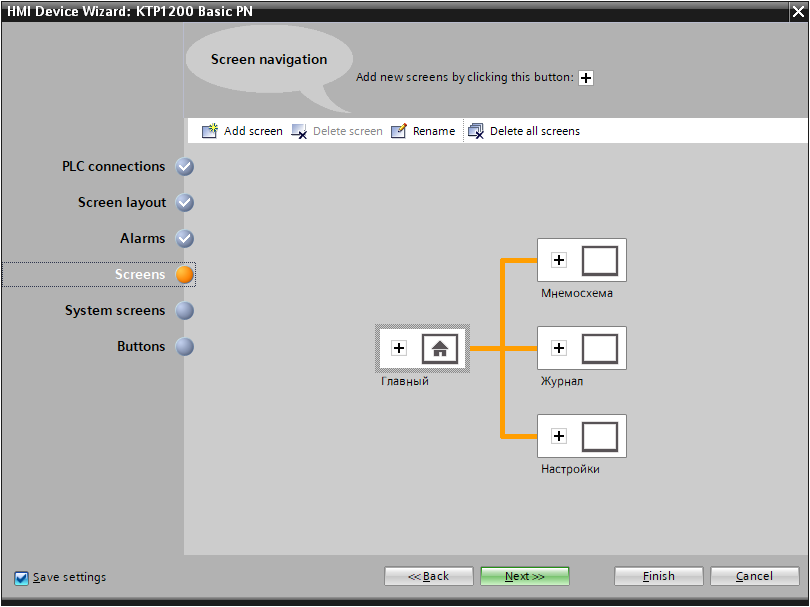


Рисунок 4.23 – Окно *Screen navigation*

Связь между тегами в проекте *WinCC* для *HMI* и данными из ПЛК организуется с помощью коммуникационных драйверов или каналов. Каналы собирают информацию о запрашиваемых компонентами системы исполнения значениях процесса, читают значения тегов процесса из систем автоматизации, и, если необходимо, записывают новые значения в компоненты системы автоматизации. Выполняется составление перечня тегов (*HMI Tags*) для связи программы контроллера с объектами экранов *HMI* панели. Каждому тегу из проекта *WinCC* ставиться в соответствие параметр из блока *PLC tags*. Функция управления архивами в *WinCC* (англ. *Archive Management*) позволяет архивировать значения процесса и сообщения с целью документирования данных об определенных рабочих и аварийных состояниях.

В разделе *HMI Tags* в таблице *Default tag table* создается список тегов для визуализации и управления технологическим объектом.

Внешний вид сформированного окна *HMI tags* для проекта панели оператора представлен на рисунке 4.24.

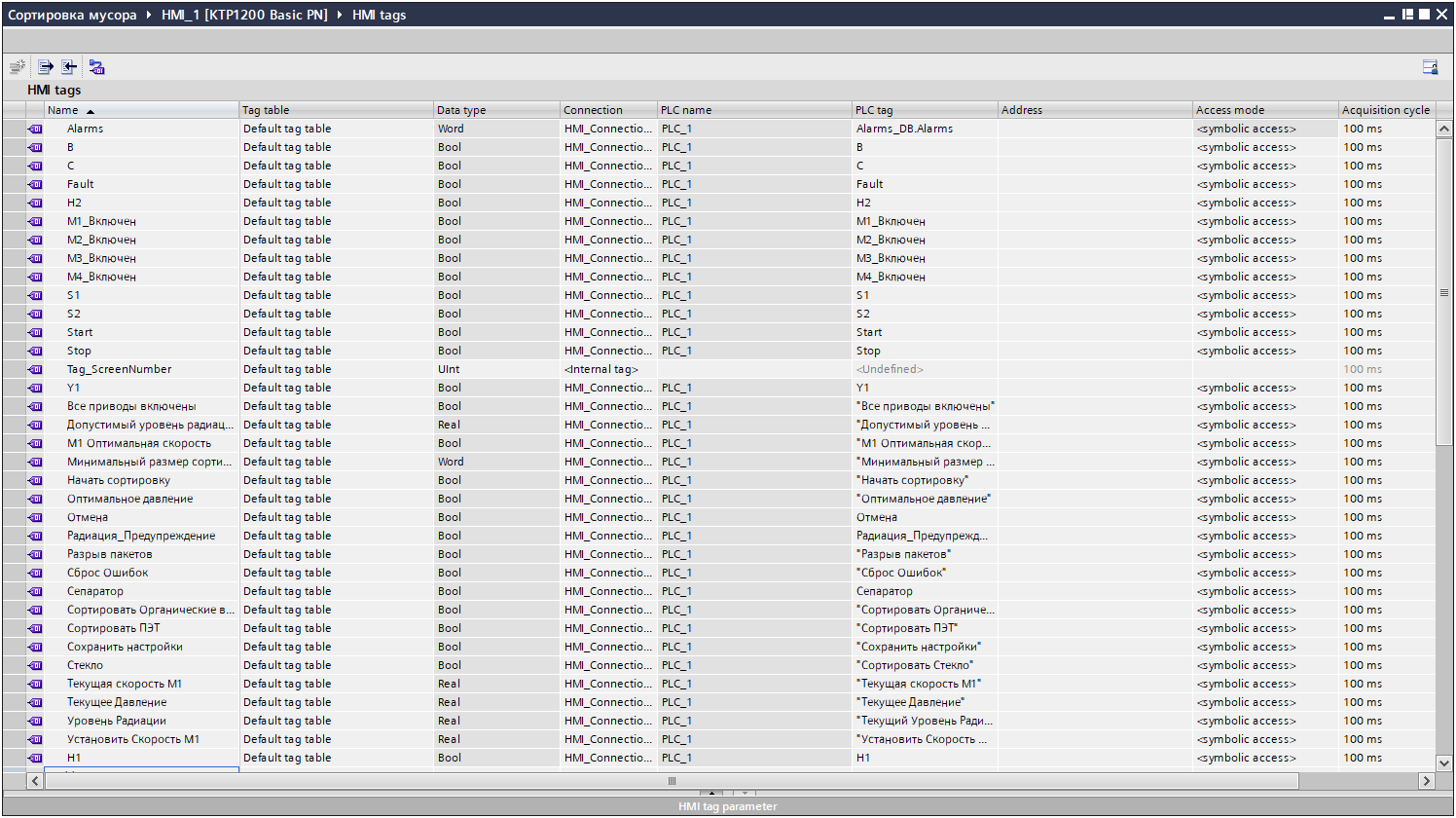


Рисунок 4.24 – Окно *HMI tags*

Наименования тегов отражают их технологическое назначение и указывают на связь с контроллером. В таблице тегов указывается наименование переменной, тип данных, тип соединения с контроллером, контроллер, соответствие *HMI* тега переменной контроллера, время обновления, комментарии и т.д.

Для контроля и управления параметрами технологического объекта и отображения текущего состояния переменных, необходимо все объекты окон визуализации *HMI* панели связать с программой контроллера. Связь объектов интерфейса с программой контроллера осуществляется с помощью настройки свойств элементов «*Object properties*». Каждому активному элементу окна присваивается необходимый тег, функция, анимация и т.д. для отображения состояния системы автоматизации в реальном времени.

В графическом редакторе *WinCC* на главном экране формируются элементы управления интерфейса пользователя. Используются стандартные компоненты: кнопки, поля ввода/вывода, выпадающие списки, индикаторы, геометрические фигуры. Внизу главного экрана располагается панель навигации для перехода на служебные экраны:

* Мнемосхема;
* Журнал;
* Настройки.

Внешний вид главного экрана представлен на рисунке 4.25.



Рисунок 4.25 – Главный экран панели оператора

Внешний вид экрана Мнемосхема представлен на рисунке 4.26.

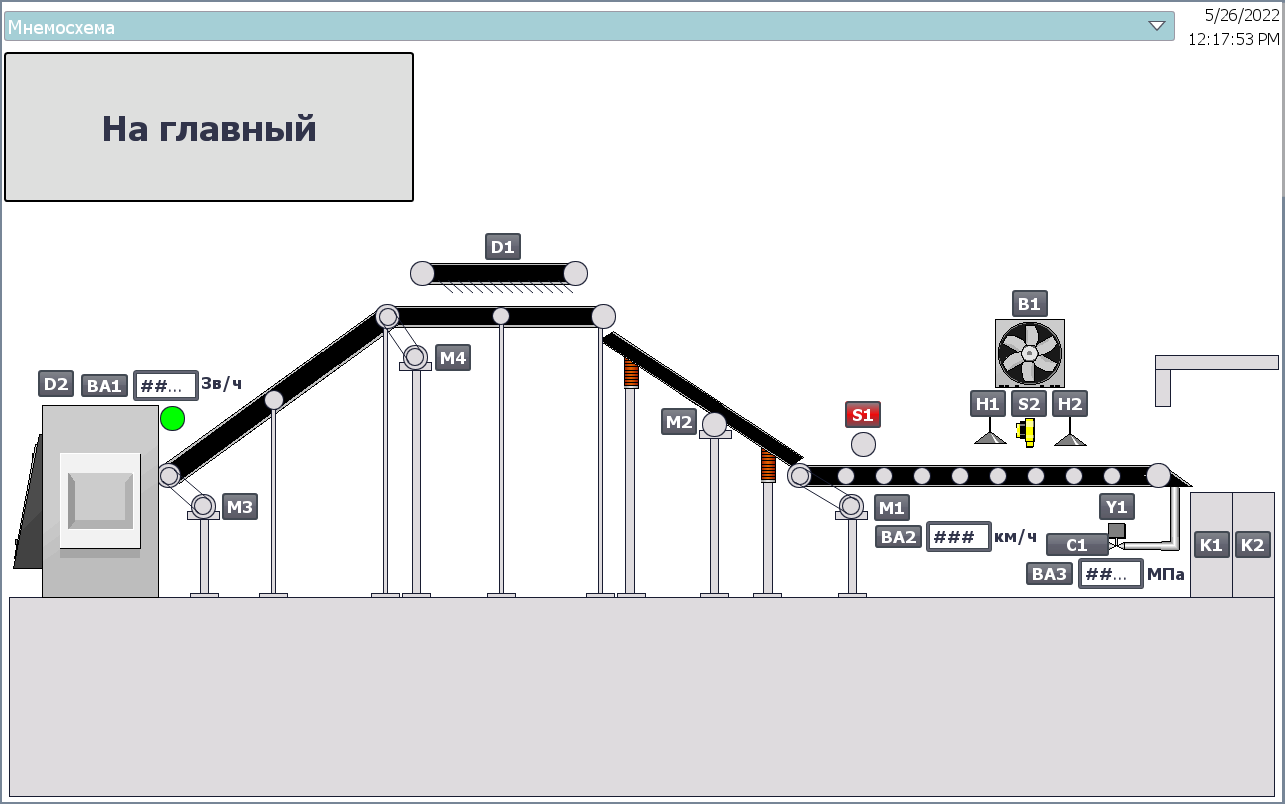


Рисунок 4.26 – Экран Мнемосхема

На мнемосхеме в режиме реального времени выводятся в полях вывода значения аналоговых параметров. Состояния срабатывания датчиков отображаются зелёным цветом.

Таким образом, в разделе разработана программная часть проекта. Для реализации использована интегрированная среда разработки *TIA Portal* 15*.* Средствами данной среды реализована программа ПЛК *SIMATIC S*7-1200 c использованием интегрированных технологических библиотек регулирования и позиционирования сервоприводов. Для создания программы использовался язык релейных схем *LAD*. Для сенсорной панели оператора был разработан проект, состоящий из главного экрана и трех служебных экранов.