

Программное обеспечение системы автоматического управления электронасосным агрегатом

Руководство по эксплуатации

ООО «РУ-ДРАЙВ ЦИФРОВЫЕ ТЕХНОЛОГИИ»

14.05.2026

1	ВВЕДЕНИЕ	2
2	ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ.....	3
3	ОБЩИЕ СВЕДЕНИЯ.....	3
3.1	Принятые сокращения	3
3.2	Основные термины.....	3
3.3	Структура программного обеспечения.....	6
3.3.1	Программы (PRG)	6
3.3.2	Функциональные блоки (FB)	7
3.3.3	Функции (FC)	7
3.3.4	Списки глобальных переменных и констант(GVL)	7
3.3.5	Пользовательские типы данных(DUT)	8
4	ОПИСАНИЕ БЛОКОВ	8
4.1	Общее графическое представление	8
4.2	Область входов.....	9
4.3	Область выходов.....	9
4.4	Интерфейс блока	9
4.5	Общий принцип работы	10
4.5.1	Режим	10
4.5.2	Состояние	10
4.5.3	Сообщения	11
4.5.4	Конфигурация	11
4.5.5	Команды.....	12
4.5.6	Блокировки	12
4.6	Блок управления маслоустановкой (fb_oilstation).....	13
4.6.1	Назначение.....	13
4.6.2	Функции	13
4.6.3	Таблица входов-выходов.....	14
4.6.4	Принцип работы.....	14
4.6.5	Интерфейс «udt_oilstation».....	15
4.6.6	Переменные интерфейса блока. Списки значений	16
4.6.6.1	Команды «std.in_cmd»	16
4.6.6.2	Аварийные сообщения «std.out_alarm».....	16
4.6.6.3	Предупредительные сообщения «std.out_warning»	16
4.6.6.4	Конфигурация «std.in_config»	17
4.6.6.5	Состояние блока «std.out_state».....	17
4.6.6.6	Состояние объекта «std.out_state_actual».....	17
4.6.7	Пример	18

1 Введение

Программное обеспечение предназначено для реализации системы автоматического управления электронасосным агрегатом. Разработанный алгоритм обеспечивает контроль технологических параметров, предупредительную и аварийную сигнализацию, защитное отключение агрегата, световую сигнализацию состояний механизмов и запорных арматур. Программное обеспечение применяется для работы электронасосного агрегата.

Программное обеспечение может быть собрана под любую платформу, поддерживающую язык ST (Structured Text).

В данном руководстве рассмотрена работа с библиотекой в среде разработки Astra.IDE (https://reglab.ru/programm/group/astra_ide/) для ПЛК REGUL.

2 Функциональные характеристики

Программное обеспечение предназначено для системы автоматического управления электронасосным агрегатом.

Основные функциональные характеристики программного обеспечения:

- контроль технологических параметров;
- защитное отключение агрегата;
- набор глобальных констант для взаимодействия с функциональными блоками из программы пользователя – передачи команд, проверки состояний и режимов, задания конфигурации, проверки аварийных и предупредительных сообщений и т.п.

3 Общие сведения

3.1 Принятые сокращения

PLC (ПЛК) – программируемый логический контроллер

CPU – центральный процессор, главный узел PLC

АРМ – автоматизированное рабочее место

ПТК – программно-технический комплекс

АЦП – аналого-цифровой преобразователь

ЦАП – цифро-аналоговый преобразователь

САУ – система автоматического/автоматизированного управления

АСУ ТП – автоматизированная система управления технологическим процессом

ШУ – шкаф управления

HMI (Human-machine interface) – человеко-машинный интерфейс

3.2 Основные термины

Агрегат, объект - унифицированный механизм или узел механизма, выполняющий определенные функции. Под агрегатом (объектом) в данной системе понимаются её базовые составные части, имеющие возможность независимой работы от остальной системы и выполняющие команды оператора или САУ (такие как насос, задвижка, клапан, датчик, аналоговый выход), состояние которых можно наблюдать в режиме реального времени посредством АРМ.

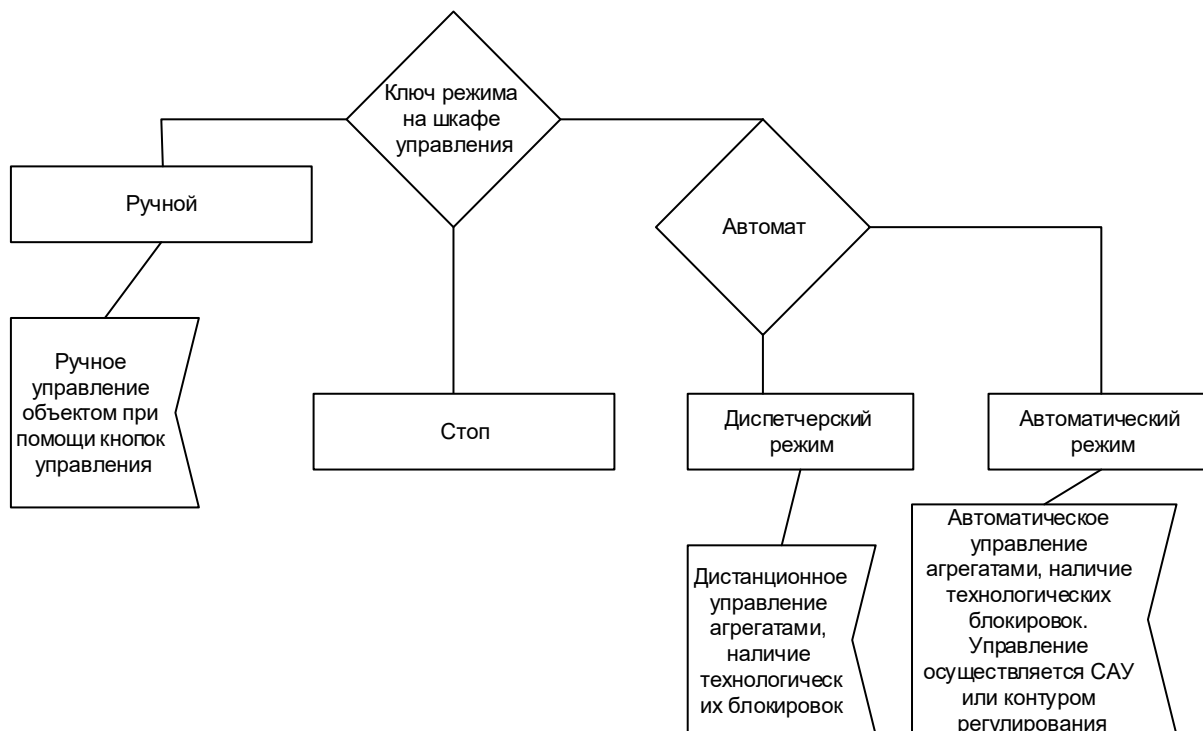
Автоматизированное рабочее место (АРМ) - часть программно-технического комплекса (ПТК), предназначенная для автоматизации технологического процесса. АРМ объединяет программно-аппаратные средства, обеспечивающие взаимодействие человека с компьютером, панелью оператора (любым электронным устройством, обеспечивающим взаимодействие человека с объектом управления), предоставляет возможность ввода информации (через клавиатуру, компьютерную мышь, сканер и пр.) и её вывод на экран монитора, принтер, графопостроитель, звуковую карту, динамики или иные устройства вывода. АРМ является частью АСУ ТП.

Состояние агрегата (объекта) – абстрактный термин, обозначающий множество стабильных значений переменных параметров объекта.

Состояние характеризуется тем, что описывает переменные свойства объекта. Состояние стабильно до тех пор, пока над объектом не будет произведено действие; если над объектом будет произведено некоторое действие, его состояние может измениться.

Например, если мотор остановлен, и оператор нажал кнопку «Пуск», мотор перейдет в состояние «работа», т.е. у объекта произойдет смена состояний «остановлен» -> «работа».

Под **режимом агрегата (объекта)** понимается метод управления исполнительным механизмом, САУ, контуром регулирования. Пояснение понятия режима приведено на схеме ниже.



Режим работы агрегата задается ключом на шкафе управления:

- Ручной (местный) - ручное управление по месту со шкафа;
- Стоп - агрегат отключен (его невозможно включить по месту или дистанционно);
- Автомат - дистанционное управление агрегатом через систему управления;

При положении ключа "автомат" появляется возможность выбора режима кнопками на АРМ оператора - диспетчерский или автоматический. В диспетчерском режиме оператор может непосредственно кнопками на мнемосхеме включить (выключить, открыть, закрыть) требуемый агрегат. В автоматическом режиме управление агрегатом передается контурам управления (регулирования). При запуске соответствующего контура он будет включать/выключать агрегаты согласно заданному алгоритму.

САУ (система автоматического управления) – программный или аппаратный комплекс, а также набор программ или алгоритмов, изменяющих состояние группы агрегатов (объектов) в соответствии с заданным законом управления.

Контур регулирования, контур управления – замкнутая система объектов, выполняющая заданную функцию. Например, регулирование расхода воды с помощью регулирующего клапана посредством обратной связи через датчик расхода воды. Контур регулирования является разновидностью САУ.

Контактор — двухпозиционный электромагнитный аппарат, предназначенный для частых дистанционных включений и выключений силовых электрических цепей в нормальном режиме работы.

HMI (Human-machine interface) – человеко-машинный интерфейс. Широкое понятие, охватывающее инженерные решения, обеспечивающие взаимодействие человека-оператора с управляемыми им машинами.

Клапан — механизм, предназначенный для открытия, закрытия или регулирования потока чего-либо.

Цех – совокупность технологических участков, выполняющих технологические операции для определенного производственного процесса.

Технологический участок – совокупность технологических узлов, объединенных процессом выполнения автономной технологической операции/набора операций.

Технологический узел - набор датчиков, исполнительных механизмов, агрегатов, объединенных процессом выполнения автономной технологической функции.

3.3 Структура программного обеспечения

В программном обеспечении выделяются три ключевых элемента организации кода и данных:

1. POU (Program Organization Unit) – программная организационная единица. Является контейнером для исходного кода, реализующего алгоритмы управления. Существует три типа POU:
 - PRG (Программа) – основная исполняемая единица, описывающая глобальное поведение системы.
 - FB (Функциональный блок) – тип POU с внутренней памятью, сохраняющий своё состояние между вызовами.
 - FC (Функция) – тип POU без внутренней памяти, возвращающий единственное значение.
2. GVL (Global Variable List) – список глобальных переменных. Обеспечивает централизованное хранение переменных, видимых из всех POU проекта. Используется для обмена данными между различными программными компонентами. GVL может быть двух видов:
 - Стандартный GVL – переменные доступны для чтения и записи из любого места проекта.
 - Константный GVL – все переменные объявлены с атрибутом только для чтения, что исключает их случайное изменение и гарантирует неизменность конфигурационных параметров.
3. DUT (Data Unit Type) – определяемый пользователем тип данных. Служит для создания собственных типов данных на основе стандартных. Позволяет описывать структуры (STRUCT), перечисления (ENUM) и псевдонимы (ALIAS), повышая читаемость и структурированность проекта.

3.3.1 Программы (PRG)

В составе программного обеспечения реализован набор программ (PRG), которые по своему назначению могут быть разделены на несколько функциональных групп:

- Обработка ввода-вывода – fc_ai, fc_di, fc_do;
- Управление объектами и регулирование – fc_as, fc_objects, fc_object_index;
- Технологические защиты – fc_protection;
- Работа с энергонезависимой памятью – fc_init_dbs, fc_save_dbs_to_retain;
- Системные сервисы – fc_main, fc_time_sync.

Подробное описание каждой программы приведено в таблице ниже.

Программа	Назначение
fc_ai	Вызов экземпляров функциональных блоков обработки аналоговых входов
fc_di	Вызов экземпляров функциональных блоков обработки дискретных входов
fc_do	Вызов экземпляров функциональных блоков обработки дискретных выходов
fc_as	Вызов экземпляров функциональных блоков контуров регулирования (управление маслоустановкой)
fc_objects	Вызов экземпляров функциональных блоков объектов управления (здвижки, контакторы и т.д.)
fc_object_index	Индексное обращение к переменным объектов управления и контуров регулирования
fc_protection	Вызов экземпляров функциональных блоков технологических защит
fc_init_dbs	Инициализация переменных пользовательских типов данных из энергонезависимой памяти ПЛК
fc_save_dbs_to_retain	Сохранение текущих состояний переменных пользовательских типов данных (объектов управления и контуров регулирования) в энергонезависимую память ПЛК
fc_main	Является точкой входа в программное обеспечение ПЛК, реализует циклический вызов остальных программ в соответствии с заданной последовательностью
fc_time_sync	Настройка и синхронизация системного времени ПЛК

3.3.2 Функциональные блоки (FB)

В программном обеспечении используются следующие типы функциональных блоков (FB):

Тип функционального блока	Назначение
fb_oilstation	Управление маслоустановкой
fb_ai	Обработка аналоговых входов
fb_valve	Управление задвижкой
fb_contactor	Управление контактором
fb_discrete_v2	Обработка дискретных сигналов
fb_protection	Реализация технологических защит

Типы fb_ai, fb_valve, fb_contactor, fb_discrete_v2, fb_protection предоставлены библиотекой Ru-Drive Object Library. Тип fb_oilstation разработан специально для данного программного обеспечения.

3.3.3 Функции (FC)

В программном обеспечении не реализовано собственных функций (FC). При необходимости выполнения вспомогательных вычислений используются стандартные функции, предоставляемые библиотекой Ru-Drive Object Library (например, математические, логические и преобразующие функции).

3.3.4 Списки глобальных переменных и констант(GVL)

В программном обеспечении глобальные переменные организованы в виде списков GVL (Global Variable List), которые обеспечивают централизованное хранение данных и доступ к ним из любых POU проекта. GVL подразделяются на стандартные (переменные доступны для чтения и записи) и константные (переменные доступны только для чтения, что гарантирует неизменность настроек и конфигурационных параметров).

Стандартные GVL:

GVL	Назначение
GVL_AI	Переменные, подключённые к входам аналоговых модулей
GVL_AI_DIAGN	Диагностика каналов модулей аналогового ввода
GVL_DBI	Экземпляры функциональных блоков объектов управления, контуров регулирования
GVL_DBS	Переменные пользовательских типов данных (объекты управления, контуры регулирования)
GVL_DBS_RETAIN	Переменные пользовательских типов данных, хранящиеся в энергонезависимой памяти (RETAIN)
GVL_DI	Переменные, подключённые к каналам модулей дискретного ввода
GVL_DIM	Значения входов дискретных модулей после обработки блоком обработки дискретных сигналов
GVL_DI_DIAGN	Диагностика каналов модулей дискретного ввода
GVL_DO	Переменные, подключённые к каналам модулей дискретного вывода
GVL_DOM	Дискретные выходные сигналы, сформированные алгоритмами перед отправкой на блок обработки дискретных выходов
GVL_DO_DIAGN	Диагностика каналов модулей дискретного вывода
GVL_OBJECT_INDEX	Переменные для индексного обращения к объектам управления

Константные GVL:

GVL	Назначение
sys_db_alm_oilstation	Аварийные сообщения блока управления маслоустановкой
sys_db_cfg_oilstation	Конфигурация блока управления маслоустановкой
sys_db_cmd_oilstation	Команды блока управления маслоустановкой
sys_db_const_oilstation	Константы блока управления маслоустановкой

GVL	Назначение
sys_db_hmi_state_actual_oilstation	Фактическое состояние маслоустановки (для HMI)
sys_db_hmi_state_oilstation	Состояния блока управления маслоустановкой (для HMI)
sys_db_mode_oilstation	Режимы работы блока управления маслоустановкой
sys_db_state_oilstation	Внутренние состояния блока управления маслоустановкой
sys_db_wrn_oilstation	Предупредительные сообщения блока управления маслоустановкой

Константные GVL для библиотечных блоков управления объектами объявлены внутри библиотеки Ru-Drive Object Library и не требуют отдельного описания в данном разделе.

3.3.5 Пользовательские типы данных(DUT)

Для хранения и структурирования данных, связанных с управлением маслоустановкой, в программном обеспечении определён следующий пользовательский тип данных (DUT).

Тип DUT	Назначение
udt_oilstation	Тип данных для блока управления маслоустановкой

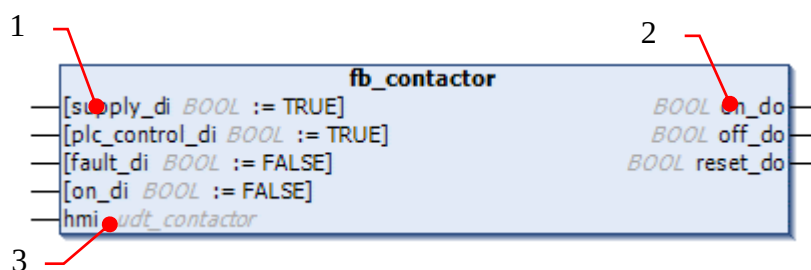
Типы данных, используемые библиотечными функциональными блоками (объекты управления, обработка сигналов и т.д.), инкапсулированы в библиотеке Ru-Drive Object Library и не представлены в явном виде в структуре программного обеспечения.

4 Описание блоков

В этом разделе для блока маслоустановки будут представлены: подробное описание, примеры использования, список входов и выходов, перечень используемых типов данных, описание конфигурации и пример вызова.

Описание библиотечных блоков приведено в документации к библиотеке Ru-Drive Object Library и не требует отдельного рассмотрения в данном разделе.

4.1 Общее графическое представление



1. Область входов
2. Область выходов
3. Вход-выходная структура (hmi)

4.2 Область входов

Набор сигналов от датчиков, показывающий действительное состояние объекта, узла, технологического участка. В некоторых случаях сигналы от других блоков.

В зависимости от этих сигналов блок может менять свои состояния. Подробнее описано в п. 4.5.2.

4.3 Область выходов

Сигналы для управления исполнительными механизмами объекта, узла, технологического участка. В некоторых случаях – данные для другого блока.

4.4 Интерфейс блока

Набор переменных для взаимодействия блоков между собой и HMI. Состоит из набора стандартных переменных (udt_standard) и собственного набора переменных, характерных для данного блока. «Udt_standard» может быть не у каждого блока.

Название	Тип	Начальное значение	Назначение	Комментарий
in_cmd	DWORD		ввод	задание режима (HI WORD), команды (LO WORD)
in_config	WORD		ввод	задание конфигурации
in_alarm	WORD		ввод	генерирование аварии в наладочном режиме
out_mode	DWORD	16#8	вывод	текущий режим
out_state	DWORD		вывод	текущее состояние блока
out_state_actual	DWORD		вывод	текущее состояние объекта
out_alarm	WORD		вывод	аварийные сообщения
out_warning	WORD		вывод	предупредительные сообщения
out_locked	DWORD		вывод	заблокированные режимы (HI WORD) и команды (LO WORD)
out_cmd	DWORD		вывод	команда высшего приоритета (HI WORD) и обычная команда (LO WORD)
out_time_operation	DINT		вывод	наработка в часах
in_sau_cmd	WORD		ввод	команда от САУ
in_sau_cmd_hi	WORD		ввод	команда от САУ с наивысшим приоритетом
in_cmd_lock	WORD		ввод	блокировки команд
in_mode_lock	WORD		ввод	блокировки режимов

- in_cmd – для подачи команд и смены режима. Команды можно подавать непрерывно и однократно, а смену режима только однократно.
- in_config – задает конфигурацию блока. Можно подавать непрерывно и однократно.
- in_alarm – подается необходимая авария для отладки в тестовом режиме. Записывается однократно
- out_mode – отображает текущий режим блока
- out_state – текущее состояние блока
- out_state_actual – текущее состояние объекта
- out_alarm – переменная, биты которой соответствуют определенным авариям блока; список возможных аварий представлен в описании
- out_warning – переменная, биты которой соответствуют определенным предупреждениям блока; список возможных предупреждений представлен в описании
- out_locked – старшее слово (HI WORD) содержит режимы, которые заблокированы для использования в текущий момент; младшее слово (LO WORD) – заблокированные команды
- out_cmd – содержит текущую команду

- out_time_operation – наработка блока в часах
- in_sau_cmd – переменная для подачи команды на блок от вышестоящего алгоритма управления. Блок должен находиться в режиме автомат. Можно подавать непрерывно и однократно.
- in_sau_cmd_hi – переменная для подачи команды с наивысшим приоритетом на блок от вышестоящего алгоритма управления. Блок должен находиться в режиме автомат. Можно подавать непрерывно и однократно.
- in_cmd_lock – переменная для подачи команд, которые будут заблокированы; необходимо слать непрерывно. Необходимо подавать непрерывно.
- in_mode_lock – переменная для подачи режимов, которые будут заблокированы; необходимо слать непрерывно. Необходимо подавать непрерывно.

4.5 Общий принцип работы

4.5.1 Режим

Под режимом понимается метод управления исполнительным механизмом, САУ, контуром регулирования.

Различают следующие режимы:

- ручной(местный) – объект, как правило, управляется вручную по месту или со шкафа управления с помощью переключателей, кнопок и т.п. В программный блок приходят только сигналы фактического состояния объекта (включен, выключен, открыт, закрыт и т.п.);
- диспетчерский – объект управляется через экземпляр функционального блока с помощью команд, поступивших в переменную «in_cmd» от оператора, формируются состояния блока и объекта;
- автоматический – управление через экземпляр функционального блока с помощью команд, поступивших в переменную «in_sau_cmd» от САУ, формируются состояния блока и объекта.

У большинства блоков присутствует вход «plc_control_di». При отсутствии данного сигнала блок переходит в режим ручной(местный). По наличию – появляется возможность выбора диспетчерского/автоматического режима.

Смена режима происходит с помощью подачи константы на переменную «in_cmd». Константа формируется смещением на шестнадцать бит глобальной константы из типа «sys_db_mode». Например, для переключения блока в режим «ремонт» возьмем константу «sys_db_mode.repair» равную 16#2. Осуществив смещение на шестнадцать бит, получим - 16#2000. Это значение нужно записать в «in_cmd».

Текущий режим отображается в переменной «out_mode». Значения переменной описываются константами в типе «sys_db_mode». Режимы идентичны для всех блоков, где они используются.

4.5.2 Состояние

Общий принцип работы функционального блока – переход по состояниям происходит зависимости от входов блока, режима, команд и внутренней логики работы блока.

В большинстве блоков используется три разновидности состояний:

- Внутреннее - «state». Используется разработчиками для реализации алгоритма блока.

- Состояние блока (out_state) – переменная для передачи внутреннего состояния для алгоритмов верхнего уровня и устройств HMI.
- Состояние объекта – «out_state_actual». Фактическое состояние объекта, которое определяется только значениями входных сигналов.

Например, задвижка может находиться в логическом состоянии «авария», «открывается», «закрывается» по переменной out_state. При этом ее фактическое состояние out_state_actual, определяемое концевыми выключателями положения, будет «открыто», «промежуточное» или «закрыто».

Таблицы с константами состояний будут представлены в описании к каждому блоку.

Большинству блоков присущи состояния такие как «нет готовности (питания)», «готовность», «работа» и «авария».

В состоянии «нет готовности» проверяются условия для готовности к работе объекта. Если все условия выполнены, то блок перейдет в состояние «готовность».

При нахождении в «готовности» блок ожидает поступления команд для смены своего состояния. Также происходит мониторинг условий готовности к работе. При их отсутствии будет осуществлен переход в «нет готовности».

При возникновении нештатных ситуаций во время работы, блок перейдет в состояние «авария».

4.5.3 Сообщения

Для более глубокого взаимодействия блока с оператором предусмотрена генерация сообщений. Сообщение – представление информации в различной форме (текст, речь, изображение и т.п.).

У большинства блоков имеются предупредительные (out_warning) и аварийные (out_alarm) сообщения. Каждому биту соответствует свое сообщение. Таблицы с константами будут представлены в описании к каждому блоку.

Причины неготовности блока, а также различные сообщения, не влияющие на работу блока, отображаются в «out_warning». Если одна из причин перестает действовать, то соответствующий бит слова предупреждений будет сброшен.

Аварии фиксируются в «out_alarm». Для сброса зафиксированной аварии и перехода в состояние «готовности» необходимо подать команду сброса аварии.

4.5.4 Конфигурация

Переменной «in_config» определяется набор функций работы блока. Например, у блока обработки аналогового входа, можно отключить формирование сообщений о выходе значения физической величины за предупредительные пределы.

У всех объектов имеются как индивидуальные, так и общие биты конфигурации. Константы индивидуальных конфигураций даны для каждого блока отдельно. Общие - представлены в таблице ниже:

Название	Тип	Значение	Описание
alarm_auto_reset	WORD	16#2000	Сброс аварий при пересбросе питания

Название	Тип	Значение	Описание
test_mode	WORD	16#4000	Режим имитации Блок не учитывает данные входов и не подает сигналы на выходы. Изменение состояний происходит по подаче команды. Используется для отладки алгоритмов.
show_state	WORD	16#8000	Показывать состояние в режиме ремонт

4.5.5 Команды

Существуют 2 вида команд:

1. Обычные команды;
2. Команды с наивысшим приоритетом.

Обычные команды подаются на вход «in_cmd» от HMI вместе с режимом объекта (один из диспетчерских режимов). Если режим у команды не совпадает с режимом объекта, то команда выполняться не будет. Также обычная команда подается из контура управления данным объектом на вход «in_sau_cmd». В данном случае, чтобы команда выполнялась, у объекта необходим автоматический режим.

Команды с наивысшим приоритетом подаются на вход «in_sau_cmd_hi» и выполняются в автоматическом и любых диспетчерских режимах. Эти команды необходимы для отработки технологических защит.

У всех объектов имеются как индивидуальные, так и общие команды. Индивидуальные команды даны для каждого блока отдельно. Общие команды представлены в таблице ниже:

Название	Тип	Значение	Описание
none	WORD	16#0	Нет команды
reset_operation_time	WORD	16#4000	Сброс наработки
reset_alarm	WORD	16#8000	Сброс аварии

4.5.6 Блокировки

Блоки имеют возможность блокировать одну или несколько команд. Для блокировки одной команды необходимо значение команды подавать на вход «in_cmd_lock». Для блокировки нескольких команд подается сумма значений (объединение по логическому «И») на этот же вход. При этом объект не меняет своего состояния. Например, при блокировке команды включения контактора у включенного контактора, контактор остается включенным. Но при отключении контактора и последующей попытке включить его контактор уже не включится.

Так же возможно блокировать установку одного или нескольких режимов. Для блокировки режима необходимо одно из значений типа «sys_db_mode» подать на вход «in_mode_lock». Для блокировки нескольких режимов подается сумма значений (объединение по логическому «И») на этот же вход. При этом, если уже заблокированный режим установлен, то никакого переключения на другой режим производиться не будет.

4.6 Блок управления маслоустановкой (fb_oilstation)

4.6.1 Назначение

Предназначен для автоматического управления работой маслосистемы насосного агрегата. Блок обеспечивает:

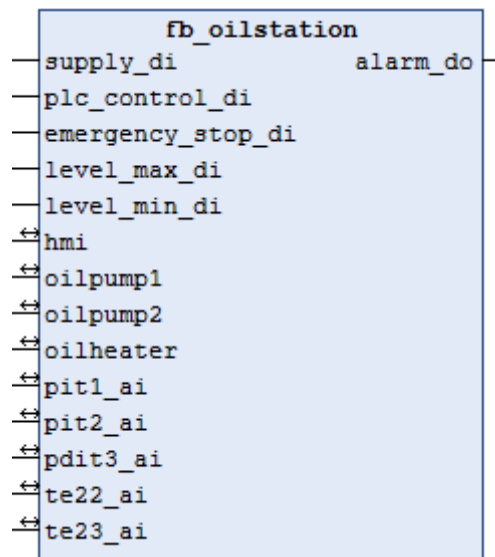
- управление пуском и остановом рабочего и резервного маслонасосов;
- управление электронагревателем масла в маслобаке;
- контроль давления масла на выходе насосов и в конце масляной линии;
- автоматическое включение резервного маслонасоса (ABP) при неисправности рабочего;
- контроль температуры масла и управление охлаждением;
- формирование сигнализации (предупредительной и аварийной) при отклонении параметров.

4.6.2 Функции

Блок реализует следующие функции:

- контроль уровня масла – пуск рабочего маслонасоса разрешён при уровне масла в маслобаке больше минимального, но меньше максимального;
- управление электронагревателем – включение нагревателя при температуре масла $<30\text{ }^{\circ}\text{C}$ и отключение при достижении $\geq 35\text{ }^{\circ}\text{C}$;
- формирование готовности к пуску – при давлении в конце масляной линии $\geq 0,07\text{ МПа}$ после включения рабочего насоса формируется сигнал готовности насосного агрегата;
- контроль набора давления при пуске – после включения магнитного пускателя давление на выходе рабочего насоса должно достичь $>0,18\text{ МПа}$ в течение 5 секунд. При успехе – насос считается включённым;
- автоматическое резервирование (ABP) – если за 5 секунд давление не набрано – насос признаётся неисправным, выполняется ABP на резервный насос с выдачей предупредительной сигнализации;
- аварийная сигнализация – при неисправности обоих маслонасосов формируется сигнал «Авария маслонасосов»;
- контроль падения давления в работе – при снижении давления на выходе рабочего насоса $<0,18\text{ МПа}$ с выдержкой 2 секунды выполняется ABP; через 1 секунду после ABP неисправный насос отключается;
- управление охлаждением масла – в рабочем режиме поддерживается температура масла после маслоохладителя в диапазоне $35\text{--}45\text{ }^{\circ}\text{C}$ посредством подачи охлаждающей воды;
- авария по температуре – при температуре масла после маслоохладителя $\geq 50\text{ }^{\circ}\text{C}$ формируется сигнал «Высокая температура масла».

4.6.3 Таблица входов-выходов



Область	Название	Тип	Начальное значение	Назначение
Input	supply_di	BOOL	TRUE	схема собрана/питание подано
	plc_control_di	BOOL	TRUE	автоматическое управление
	emergency_stop_di	BOOL	FALSE	аварийный останов
	level_max_di	BOOL	FALSE	LE1a: уровень масла >= Max
	level_min_di	BOOL	FALSE	LE1b: уровень масла <= Min
Inout	hmi	udt_oilstation		интерфейс блока
	oilpump1	udt_contactor		интерфейс маслососа H1
	oilpump2	udt_contactor		интерфейс маслососа H2
	oilheater	udt_contactor		нагреватель масла в маслобаке
	pit1_ai	udt_analog		датчик PIT1
	pit2_ai	udt_analog		датчик PIT2
	pdit3_ai	udt_analog		датчик PDIT3
	te22_ai	udt_analog		датчик TE22
	te23_ai	udt_analog		датчик TE23
Output	alarm_do	BOOL		общая авария

4.6.4 Принцип работы

Блок реализует управление маслоустановкой на основе конечного автомата. Логика блока определяется текущим состоянием, командами с верхнего уровня, сигналами датчиков и диагностикой исполнительных устройств.

Управление нагревателем выполняется независимо от конечного автомата. Нагреватель включается при температуре масла в баке ниже уставки `t_oil_heater_on`, нормальном уровне и отсутствии аварийного останова. Выключается нагреватель при достижении верхней уставки `t_oil_heater_off`, падении уровня или аварийном останове.

В состоянии `S_NOT_READY` блок проверяет условия готовности: уровень масла в норме, температура не ниже минимальной, оба насоса исправны и остановлены. При выполнении условий блок переходит в состояние `S_STOPPED`. В `S_STOPPED` блок ожидает команду `start`. По её получению запускается текущий рабочий насос, и блок переходит в `S_OILPUMP_STARTING`.

В S_OILPUMP_STARTING запускается таймер контроля набора давления. Если за заданное время давление достигает уставки p_start_min, блок переходит в S_RUNNING. Если нет — насос помечается как неисправный, и при наличии исправного резерва выполняется автоматическое резервирование (ABP). При неисправности обоих насосов фиксируется авария.

В S_RUNNING блок контролирует давление рабочего насоса. При падении давления ниже уставки p_work_min запускается таймер; если давление не восстанавливается — инициируется ABP. Команда stop переводит блок в S_STOPPED. В этом состоянии также может выполняться ротация насосов — смена рабочего и резервного насосов без останова маслоустановки.

При ABP происходит смена ролей насосов, запуск нового рабочего насоса и отключение неисправного после заданной задержки. При ротации запускается резервный насос; при успешном наборе давления старый насос останавливается, и роли меняются. При неудаче запустившийся насос останавливается и помечается как неисправный.

В аварийном состоянии S_ALARM выход alarm_do устанавливается в TRUE, все команды на насосы блокируются. Выход из аварии возможен только по команде reset_alarm при отсутствии активного аварийного останова.

Блок формирует предупредительные сообщения: высокая температура масла, неисправность конкретного насоса, отклонение уровня, наладочный режим. Аварийные сообщения формируются при неисправности обоих насосов, аварийном останове и авариях подчинённых блоков насосов.

Блок поддерживает автоматический, диспетчерский, местный и ремонтный режимы. В местном и ремонтном режимах конечный автомат отключается. Блок также ведёт учёт наработки маслоустановки и предоставляет вектор готовности для верхнего уровня.

4.6.5 Интерфейс «udt_oilstation»

В следующей таблице представлен интерфейс блока, представленный типом данных «udt_oilstation».

Название	Тип	Нач. знач-е	Назначение	Описание
std	udt_standard			стандартный набор переменных
time_pressure_start	REAL	5.0	ввод	время набора давления при пуске, с
time_pressure_drop	REAL	2.0	ввод	время снижения давления в работе, с
time_avr_delay	REAL	1.0	ввод	задержка отключения неисправного маслонасоса после ABP, с
p_start_min	REAL	0.18	ввод	мин. давление на выходе маслонасоса при пуске, МПа
p_work_min	REAL	0.18	ввод	мин. давление на выходе маслонасоса в работе, МПа
t_oil_heater_on	REAL	30.0	ввод	температура включения нагревателя, °C
t_oil_heater_off	REAL	35.0	ввод	температура отключения нагревателя, °C
ready	WORD	16#0000	вывод	флаги готовности маслоустановки к запуску
tag_name	STRING[16]		ввод-вывод	диспетчерское наименование

4.6.6 Переменные интерфейса блока. Списки значений

4.6.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_oilstation». Возможные значения представлены в таблице ниже.

Название	Тип	Значение	Описание
none	WORD	16#0	Нет команды
stop	WORD	16#1	Стоп
start	WORD	16#2	Пуск
reset_oilpump_fault	WORD	16#4	Сброс неисправности маслонасоса
reset_operation_time	WORD	16#4000	Сброс наработки
reset_alarm	WORD	16#8000	Сброс аварии

4.6.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_oilstation». Возможные значения представлены в таблице ниже.

Название	Тип	Значение	Текс сообщений
none	WORD	16#0	Нет аварий
both_oilpump_fault	WORD	16#1	Авария маслонасосов (оба неисправны)
emergency_stop	WORD	16#2	Аварийный останов
power_off	WORD	16#4	Пропадание питания

4.6.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_oilstation». Ее возможные значения представлены в таблице ниже.

Название	Тип	Значение	Текс сообщения
none	WORD	16#0	Нет предупреждения
oilpump1_fault	WORD	16#1	Неисправность маслонасоса Н1
oilpump2_fault	WORD	16#2	Неисправность маслонасоса Н2
oiltemp_high	WORD	16#4	Высокая температура масла после охладителя
oillevel_min	WORD	16#8	Уровень масла ниже минимума
oillevel_max	WORD	16#10	Уровень масла выше максимума
test_mode	WORD	16#20	Режим имитации

4.6.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_oilstation». Ее возможные значения представлены в таблице ниже.

Название	Тип	Значение	Описание
oilpump2_working	WORD	16#1	Рабочий маслонасос H2(маслонасос H1 в резерве)
alarm_auto_reset	WORD	16#2000	Сброс аварий при пересбросе питания
test_mode	WORD	16#4000	Режим имитации
show_state	WORD	16#8000	Показывать состояние в режиме ремонт

4.6.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_oilstation». Возможные значения представлены в таблице ниже.

Название	Тип	Значение	Описание
none	WORD	16#0	Неопределенное
not_ready	WORD	16#1	Не готова
stopped	WORD	16#2	Остановлена
oilpump_starting	WORD	16#4	Пуск
running	WORD	16#8	В работе
alarm	WORD	16#10	Авария
avr	WORD	16#20	АВР
oilpump_rotation	WORD	16#40	Ротация маслонасосов

4.6.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_oilstation». Возможные значения представлены в таблице ниже.

Название	Тип	Значение	Описание
none	WORD	16#0	Неопределенное
not_ready	WORD	16#1	Не готова
stopped	WORD	16#2	Остановлена
running	WORD	16#4	В работе

4.6.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_oilstation для маслоустановки
    dbs_oilstation_1: udt_oilstation;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_oilstation для маслоустановки
    dbi_oilstation_1 : fb_oilstation;
END_VAR

// вызов экземпляра блока fb_oilstation для маслоустановки
dbi_oilstation_1(hmi := dbs_oilstation);
```