

Ru-Drive Object Library

Руководство по эксплуатации

Листов 216

24.11.2022

Оглавление

| | | |
|-----|--|-----|
| 1 | Введение | 2 |
| 2 | Функциональные характеристики | 3 |
| 3 | Общие сведения..... | 3 |
| 3.1 | Принятые сокращения | 3 |
| 3.2 | Основные термины | 3 |
| 3.3 | Структура библиотеки..... | 6 |
| 4 | Описание блоков..... | 18 |
| 6 | Описание вспомогательных типов данных | 215 |
| 7 | Контактная информация производителя..... | 217 |
| 7.1 | Юридическая информация..... | 217 |
| 7.2 | Контактная информация службы технической поддержки | 217 |

1 Введение

Ru-Drive Object Library (далее - библиотека) является кроссплатформенной программной библиотекой и предназначена для быстрой разработки систем управления промышленными установками.

Основой библиотеки является набор алгоритмов, предназначенных для управления промышленным оборудованием: насосными станциями, котельными установками, компрессорными установками, системами водоподготовки и водоочистки, конвейерными системами и т.д.

Библиотека подключается к среде разработки как дополнительный модуль, после чего разработчику становятся доступны функции библиотеки.

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

В данном руководстве рассмотрена работа с библиотекой в среде разработки Codesys (<https://www.codesys.com/>) для контроллеров Овен (<https://owen.ru/>) и Абак (<https://incomsystem.ru/abak-controllers/>).

Перед прочтением данной инструкции рекомендуется ознакомиться с CODESYS V3.5 первый старт https://ftp.owen.ru/CoDeSys3/11_Documentation/03_3.5.11.5/CDSv3.5_FirstStart_v2.1.pdf.

2 Функциональные характеристики

Библиотека предназначена для построения систем управления промышленным оборудованием: насосными станциями, котельными установками, компрессорными установками, системами водоподготовки и водоочистки, конвейерными системами и т.д.

Основные функциональные характеристики библиотеки:

- предоставление разработчику стандартного набора функциональных блоков для управления типовыми элементами/объектами промышленных установок: датчиков, электрических контакторов и ячеек, насосов, дозаторов, устройств плавного пуска, преобразователей частоты, задвижек, клапанов и т.п.;
- реализация стандартного интерфейса типового элемента для упрощения взаимодействия с функциональным блоком из программы пользователя;
- 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

ЧРП – частотно-регулируемый привод

ПЧ – преобразователь частоты

УПП – устройство плавного пуска

PLC (ПЛК) – программируемый логический контроллер

СРУ – 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.

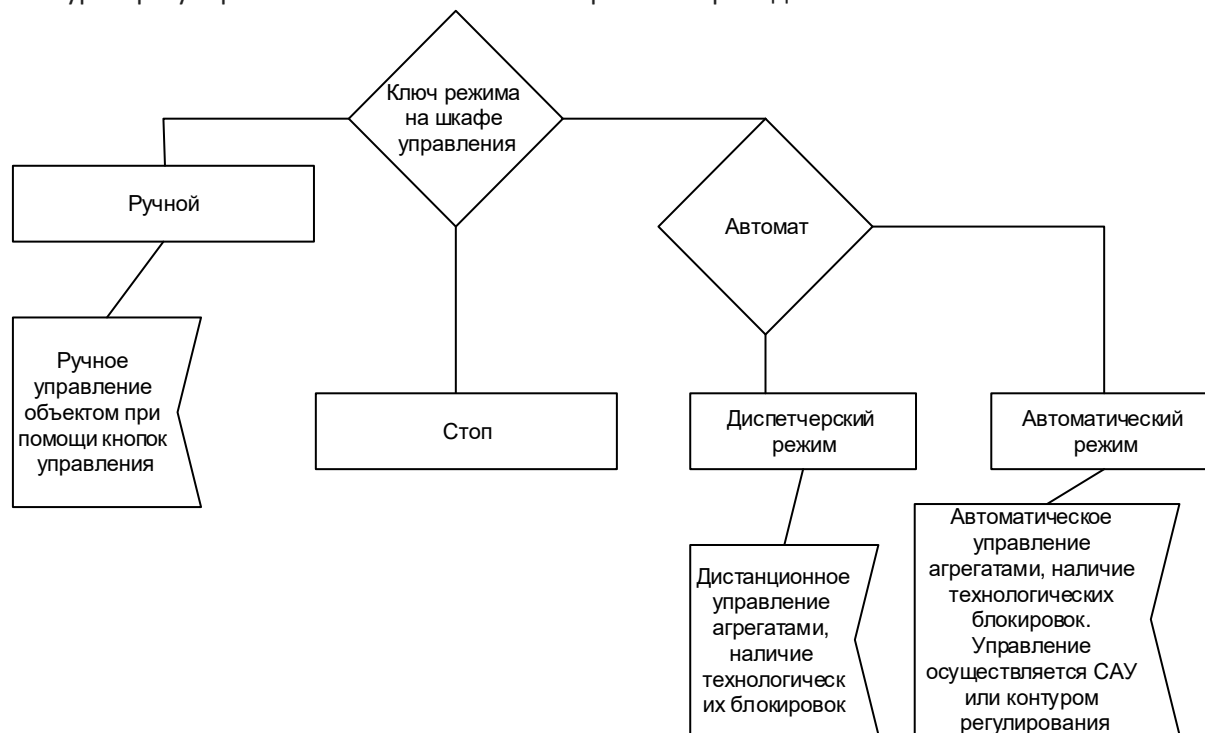
Автоматизированное рабочее место (АРМ) - часть программно-технического комплекса (ПТК), предназначенная для автоматизации технологического процесса. АРМ объединяет программно-аппаратные средства, обеспечивающие взаимодействие человека с компьютером, панелью оператора (любым электронным устройством, обеспечивающим 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.

ЧРП (частотно-регулируемый привод) – система управления частотой вращения ротора асинхронного (или синхронного) электродвигателя. Состоит из собственно электродвигателя и частотного преобразователя.

Частотный преобразователь (преобразователь частоты, ПЧ) – это устройство, состоящее из выпрямителя (моста постоянного тока), преобразующего переменный ток промышленной частоты в постоянный, и инвертора (преобразователя), преобразующего постоянный ток в переменный требуемых частоты и амплитуды.

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

Устройство плавного пуска (УПП) — механическое, электротехническое (электронное) или электромеханическое устройство, используемое для плавного пуска (остановки) электродвигателей с небольшим моментом срабатывания (например, с вентиляторной характеристикой) рабочей машины.

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

Насосный агрегат – система объектов, состоящая из насоса, затвора на всасе и на напоре, набора защитных сигналов (наличие транспортируемой среды, ЭКМ (или AI) минимального и максимального давления после насоса).

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

Насосная станция — комплексная система для перекачки жидкостей из одного места в другое, включает в себя: насосные агрегаты (рабочие и резервные).

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 библиотеки

При объявлении типов, структур, функций и функциональных блоков из библиотеки нужно использовать пространство имен RuDrive.

3.3.1 Функциональные блоки (Function Blocks)

Функциональный блок (ФБ) – это программный объект, значения локальных переменных которого могут сохраняться между вызовами. Все элементы, которым требуется память – счетчики, таймеры, блоки архивации, блоки обмена и т. д. – представляют собой функциональные блоки.

Список и назначение функциональных блоков библиотеки представлены в таблице ниже.

| Название | Краткое описание |
|----------------|--|
| CONT_C | Непрерывный PID-регулятор |
| CONT_S | Шаговый PID-регулятор |
| fb_ai | Обработка аналогового входа |
| fb_ai_2oo3 | Обработка аналогового входа 2 из 3 |
| fb_ao | Обработка аналогового выхода |
| fb_batch_pump | Насос-дозатор |
| fb_contactor | Контактор |
| fb_cont_upp | Блок управления контактор + УПП |
| fb_discrete | Обработка дискретных входов-выходов |
| fb_discrete_v2 | Обработка дискретных входов-выходов. Версия 2. |
| fb_eng_f | Преобразование интерфейса udt_fr к интерфейсу udt_engine |
| fb_eng_c | Преобразование интерфейса udt_contactor к интерфейсу udt_engine |
| fb_eng_u | Преобразование интерфейса udt_upp к интерфейсу udt_engine |
| fb_eng_s | Преобразование интерфейса udt_switch_gear к интерфейсу udt_engine |
| fb_pump_c | Преобразование интерфейса udt_contactor к интерфейсу udt_pump |
| fb_pump_f | Преобразование интерфейса udt_fr к интерфейсу udt_pump |
| fb_eng_cf | Коммутация электродвигателя. Схема подключения: контактор->ПЧ-> (одиночная ветка) |
| fb_eng_c_fc | Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->ПЧ) |
| fb_eng_c_uc | Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->УПП) |
| fb_eng_sc_scfc | Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка->контактор) + (ячейка->контактор->ПЧ->контактор) |
| fb_eng_sf | Коммутация электродвигателя. Схема подключения: ячейка->ПЧ (одиночная ветка). |
| fb_eng_su | Коммутация электродвигателя. Схема подключения: ячейка->УПП (одиночная ветка). |
| fb_eng_s_fs | Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ПЧ ->ячейка) |
| fb_eng_s_sfc | Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка ->ПЧ->контактор) |
| fb_eng_s_sfs | Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка ->ПЧ->ячейка) |
| fb_eng_s_sus | Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка ->УПП->ячейка->) |
| fb_fr | Управление ПЧ |
| fb_motor | Управление двигателем с возможностью реверсивного движения |
| fb_pump | Управление насосным агрегатом |
| fb_rk_cont | Клапан с аналоговым управлением |
| fb_rk_step | Клапан с дискретным управлением |
| fb_rotation | Ротация насосов |
| fb_switch_gear | Управление электрической ячейкой |
| fb_upp | Управление УПП |

| Название | Краткое описание |
|------------------|--|
| fb_valve | Управление задвижкой |
| fb_protection | Технологические защиты. Выдает сигнал аварии на защищаемый контур или техпроцесс |
| fb_flow_counter | Сумматор расхода |
| fb_pulse_counter | Сумматор импульсов |
| fb_pump_sau | Насосная станция |
| fb_queue | Очередь |
| fb_time_counter | Сумматор времени |

3.3.2 Функции (Functions)

Функция – это программный объект, не имеющий памяти для сохранения значений своих переменных между вызовами. Функции используются для арифметических вычислений, масштабирования сигналов, конвертации данных из одного формата в другой, различных проверок (например, на принадлежность значения допустимому диапазону) и т. д.

| Название | Краткое описание |
|----------------------|--|
| fc_cmd | Интерпретатор команд и режимов. Возвращает новую команду и режим после обработки |
| fc_datetime | Получение даты и времени |
| fc_pack_bits_dword | Упаковка битов в двойное слово |
| fc_pack_bits_word | Упаковка битов в слово |
| fc_time | Системный таймер. Возвращает время контроллера с момента старта в секундах |
| fc_unpack_dword_bits | Распаковка битов из двойного слова |
| fc_unpack_word_bits | Распаковка битов из слова |
| fc_set_mode | Установка режима |
| fc_time_operation | Подсчет наработки |

3.3.3 Глобальные константы (Global Constants)

Некоторые значения в приложении всегда должны оставаться неизменными. Вместо использования в коде конкретных чисел лучше объявить константы. Во-первых, это упростит поддержку кода – ведь если «магическое число» изменится, то придется искать в программе все фрагменты, где оно использовалось, и редактировать их. В случае константы – достаточно будет изменить строку объявления константы. Кроме того, константы защищены от записи – даже в случае ошибки программист не сможет изменить их из кода программы.

Раздел содержит глобальные константы для работы с функциональными блоками и функциями библиотеки.

3.3.3.1 Аварийные сообщения блоков (sys_db_alm)

Состав «sys_db_alm» приведен в таблице.

| Название | Тип | Значение | Описание |
|----------|---------------------------|----------|------------|
| none | WORD | 16#0 | Нет аварий |
| ai | <u>sys_db_alm_ai</u> | | |
| ai_2003 | <u>sys_db_alm_ai_2003</u> | | |
| fr | <u>sys_db_alm_fr</u> | | |
| rk_cont | <u>sys_db_alm_rk_cont</u> | | |
| rk_step | <u>sys_db_alm_rk_step</u> | | |
| valve | <u>sys_db_alm_valve</u> | | |

| Название | Тип | Значение | Описание |
|-------------|-------------------------------|----------|----------|
| switch_gear | <u>sys_db_alm_switch_gear</u> | | |
| contactor | <u>sys_db_alm_contactor</u> | | |
| motor | <u>sys_db_alm_motor</u> | | |
| upp | <u>sys_db_alm_upp</u> | | |
| batch_pump | <u>sys_db_alm_batch_pump</u> | | |
| protection | <u>sys_db_alm_protection</u> | | |
| engine | <u>sys_db_alm_engine</u> | | |
| cont_upp | <u>sys_db_alm_cont_upp</u> | | |
| engine_v2 | <u>sys_db_alm_engine_v2</u> | | |

Соответствие типа объекту библиотеки представлено в таблице ниже. Значения констант будут приведены в описании к каждому объекту.

| Название | Объект |
|------------------------|---|
| sys_db_alm_ai | fb_ai. Обработка аналогового входа |
| sys_db_alm_ai_2oo3 | fb_ai_2oo3. Обработка аналогового входа 2 из 3 |
| sys_db_alm_batch_pump | fb_batch_pump. Насос-дозатор |
| sys_db_alm_contactor | fb_contactor. Контактор |
| sys_db_alm_cont_upp | fb_cont_upp. Блок управления контактор + УПП |
| sys_db_alm_engine | fb_eng_cf. Коммутация электродвигателя. Схема подключения: контактор->ПЧ (одиночная ветка) |
| | fb_eng_c_fc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->ПЧ) |
| | fb_eng_c_uc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->УПП) |
| | fb_eng_sc_scfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка->контактор) + (ячейка->контактор->ПЧ->контактор) |
| | fb_eng_sf. Коммутация электродвигателя. Схема подключения: ячейка->ПЧ (одиночная ветка). |
| | fb_eng_su. Коммутация электродвигателя. Схема подключения: ячейка->УПП (одиночная ветка). |
| | fb_eng_s_fs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ПЧ->ячейка) |
| | fb_eng_s_sfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->контактор) |
| | fb_eng_s_sfs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->ячейка) |
| | fb_eng_s_sus. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->УПП->ячейка) |
| sys_db_alm_engine_v2 | fb_engine_v2. Коммутация электродвигателя. Реализует последовательности включений и отключений агрегатов с контролем промежуточных состояний и блокировками |
| sys_db_alm_fr | fb_fr. Управление ПЧ. |
| sys_db_alm_motor | fb_motor. Управление двигателем с возможностью реверсивного движения. |
| sys_db_alm_protection | fb_protection. Технические защиты. Выдает сигнал аварии на защищаемый контур или техпроцесс. |
| sys_db_alm_rk_cont | fb_rk_cont. Клапан с аналоговым управлением |
| sys_db_alm_rk_step | fb_rk_step. Клапан с дискретным управлением |
| sys_db_alm_switch_gear | fb_switch_gear. Управление электрической ячейкой |
| sys_db_alm_upp | fb_upp. Управление УПП |
| sys_db_alm_valve | fb_valve. Управление задвижкой |

3.3.3.2 Предупредительные сообщения блоков (sys_db_wrn)

Состав «sys_db_wrn» приведен в таблице.

| Название | Тип | Значение | Описание |
|-------------|------------------------|----------|--------------------|
| none | WORD | 16#0 | Нет предупреждений |
| test_mode | WORD | 16#8000 | Режим имитации |
| ai | sys_db_wrn_ai | | |
| ai_2oo3 | sys_db_wrn_ai_2oo3 | | |
| fr | sys_db_wrn_fr | | |
| rk_cont | sys_db_wrn_rk_cont | | |
| rk_step | sys_db_wrn_rk_step | | |
| valve | sys_db_wrn_valve | | |
| switch_gear | sys_db_wrn_switch_gear | | |
| contactor | sys_db_wrn_contactor | | |
| motor | sys_db_wrn_motor | | |
| upp | sys_db_wrn_upp | | |
| batch_pump | sys_db_wrn_batch_pump | | |
| engine | sys_db_wrn_engine | | |
| cont_upp | sys_db_wrn_cont_upp | | |
| engine_v2 | sys_db_wrn_engine_v2 | | |

Соответствие типа объекту библиотеки предоставлено в таблице ниже. Значения констант будут приведены в описании к каждому объекту.

| Название | Объект |
|------------------------|---|
| sys_db_wrn_ai | fb_ai. Обработка аналогового входа |
| sys_db_wrn_ai_2oo3 | fb_ai_2oo3. Обработка аналогового входа 2 из 3 |
| sys_db_wrn_batch_pump | fb_batch_pump. Насос-дозатор |
| sys_db_wrn_contactor | fb_contactor. Контактор |
| sys_db_wrn_cont_upp | fb_cont_upp. Блок управления контактор + УПП |
| sys_db_wrn_engine | fb_eng_cf. Коммутация электродвигателя. Схема подключения: контактор->ПЧ (одиночная ветка) |
| | fb_eng_c_fc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->ПЧ) |
| | fb_eng_c_uc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->УПП) |
| | fb_eng_sc_scfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка->контактор) + (ячейка->контактор->ПЧ->контактор) |
| | fb_eng_sf. Коммутация электродвигателя. Схема подключения: ячейка->ПЧ (одиночная ветка). |
| | fb_eng_su. Коммутация электродвигателя. Схема подключения: ячейка->УПП (одиночная ветка). |
| | fb_eng_s_fs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ПЧ->ячейка) |
| | fb_eng_s_sfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->контактор) |
| | fb_eng_s_sfs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->ячейка) |
| | fb_eng_s_sus. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->УПП->ячейка) |
| sys_db_wrn_engine_v2 | fb_engine_v2. Коммутация электродвигателя. Реализует последовательности включений и отключений агрегатов с контролем промежуточных состояний и блокировками |
| sys_db_wrn_fr | fb_fr. Управление ПЧ. |
| sys_db_wrn_motor | fb_motor. Управление двигателем с возможностью реверсивного движения. |
| sys_db_wrn_rk_cont | fb_rk_cont. Клапан с аналоговым управлением |
| sys_db_wrn_rk_step | fb_rk_step. Клапан с дискретным управлением |
| sys_db_wrn_switch_gear | fb_switch_gear. Управление электрической ячейкой |
| sys_db_wrn_upp | fb_upp. Управление УПП |

| Название | Объект |
|------------------|--------------------------------|
| sys_db_wrn_valve | fb_valve. Управление задвижкой |

3.3.3.3 Конфигурация блоков (sys_db_cfg)

Состав «sys_db_cfg» приведен в таблице.

| Название | Тип | Значение | Описание |
|------------------|------------------------|----------|--------------------------------------|
| 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 |
| ai | sys_db_cfg_ai | | |
| ai_2oo3 | sys_db_cfg_ai_2oo3 | | |
| fr | sys_db_cfg_fr | | |
| rk_cont | sys_db_cfg_rk_cont | | |
| rk_step | sys_db_cfg_rk_step | | |
| valve | sys_db_cfg_valve | | |
| upp | sys_db_cfg_upp | | |
| switch_gear | sys_db_cfg_switch_gear | | |
| protection | sys_db_cfg_protection | | |
| engine | sys_db_cfg_engine | | |
| cont_upp | sys_db_cfg_cont_upp | | |
| contactor | sys_db_cfg_contactor | | |

Соответствие типа объекту библиотеки представлено в таблице ниже. Значения констант будут приведены в описании к каждому объекту.

| Название | Объект |
|----------------------|--|
| sys_db_cfg_ai | fb_ai. Обработка аналогового входа |
| sys_db_cfg_ai_2oo3 | fb_ai_2oo3. Обработка аналогового входа 2 из 3 |
| sys_db_cfg_contactor | fb_contactor. Контактор |
| sys_db_cfg_cont_upp | fb_cont_upp. Блок управления контактор + УПП |
| sys_db_cfg_engine | fb_eng_cf. Коммутация электродвигателя. Схема подключения: контактор->ПЧ (одиночная ветка) |
| | fb_eng_c_fc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->ПЧ) |
| | fb_eng_c_uc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->УПП) |
| | fb_eng_sc_scfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка->контактор) + (ячейка->контактор->ПЧ->контактор) |
| | fb_eng_sf. Коммутация электродвигателя. Схема подключения: ячейка->ПЧ (одиночная ветка). |
| | fb_eng_su. Коммутация электродвигателя. Схема подключения: ячейка->УПП (одиночная ветка). |
| | fb_eng_s_fs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ПЧ->ячейка) |
| | fb_eng_s_sfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->контактор) |
| | fb_eng_s_sfs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->ячейка) |
| | fb_eng_s_sus. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->УПП->ячейка) |
| sys_db_cfg_fr | fb_fr. Управление ПЧ. |
| sys_db_cfg_rk_cont | fb_rk_cont. Клапан с аналоговым управлением |
| sys_db_cfg_rk_step | fb_rk_step. Клапан с дискретным управлением |

| Название | Объект |
|------------------------|--|
| sys_db_cfg_switch_gear | fb_switch_gear. Управление электрической ячейкой |
| sys_db_cfg_upp | fb_upp. Управление УПП |
| sys_db_cfg_valve | fb_valve. Управление задвижкой |

3.3.3.4 Команды блоков (sys_db_cmd)

Состав «sys_db_cmd» приведен в таблице.

| 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 |
| fr | sys_db_cmd_fr | | |
| rk_cont | sys_db_cmd_rk_cont | | |
| rk_step | sys_db_cmd_rk_step | | |
| valve | sys_db_cmd_valve | | |
| switch_gear | sys_db_cmd_switch_gear | | |
| contactor | sys_db_cmd_contactor | | |
| motor | sys_db_cmd_motor | | |
| upp | sys_db_cmd_upp | | |
| batch_pump | sys_db_cmd_batch_pump | | |
| protection | sys_db_cmd_protection | | |
| engine | sys_db_cmd_engine | | |
| cont_upp | sys_db_cmd_cont_upp | | |

Соответствие типа объекту библиотеки представлено в таблице ниже. Значения констант будут приведены в описании к каждому объекту.

| Название | Объект |
|------------------------|--|
| sys_db_cmd_contactor | fb_contactor. Контактор |
| sys_db_cmd_cont_upp | fb_cont_upp. Блок управления контактор + УПП |
| sys_db_cmd_engine | fb_eng_cf. Коммутация электродвигателя. Схема подключения: контактор->ПЧ (одиночная ветка) |
| | fb_eng_c_fc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->ПЧ) |
| | fb_eng_c_uc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->УПП) |
| | fb_eng_sc_scfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка->контактор) + (ячейка->контактор->ПЧ->контактор) |
| | fb_eng_sf. Коммутация электродвигателя. Схема подключения: ячейка->ПЧ (одиночная ветка). |
| | fb_eng_su. Коммутация электродвигателя. Схема подключения: ячейка->УПП (одиночная ветка). |
| | fb_eng_s_fs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ПЧ->ячейка) |
| | fb_eng_s_sfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->контактор) |
| | fb_eng_s_sfs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->ячейка) |
| | fb_eng_s_sus. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->УПП->ячейка) |
| sys_db_cmd_fr | fb_fr. Управление ПЧ. |
| sys_db_cmd_rk_cont | fb_rk_cont. Клапан с аналоговым управлением |
| sys_db_cmd_rk_step | fb_rk_step. Клапан с дискретным управлением |
| sys_db_cmd_switch_gear | fb_switch_gear. Управление электрической ячейкой |

| Название | Объект |
|-----------------------|--|
| sys_db_cmd_upp | fb_upp. Управление УПП |
| sys_db_cmd_valve | fb_valve. Управление задвижкой |
| sys_db_cmd_motor | fb_motor. Управление двигателем с возможностью реверсивного движения. |
| sys_db_cmd_batch_pump | fb_batch_pump. Насос-дозатор |
| sys_db_cmd_protection | fb_protection. Технические защиты. Выдает сигнал аварии на защищаемый контур или техпроцесс. |

3.3.3.5 Константы блоков (sys_db_const)

Состав «sys_db_const» приведен в таблице.

| Название | Тип | Значение | Описание |
|-------------|------------------------|----------|-------------------------------------|
| time_sample | REAL | 0.1 | время цикла PLC |
| ai | sys_db_const_ai | | аналоговый вход |
| ao | sys_db_const_ao | | аналоговый выход |
| engine_v2 | sys_db_const_engine_v2 | | блок коммутации электродвигателя v2 |

Соответствие типа объекту библиотеки представлено в таблице ниже. Значения констант будут приведены в описании к каждому объекту.

| Название | Объект |
|------------------------|---|
| sys_db_const_ai | fb_ai. Обработка аналогового входа |
| sys_db_const_ao | fb_ao. Обработка аналогового выхода |
| sys_db_const_engine_v2 | fb_engine_v2. Коммутация электродвигателя. Реализует последовательности включений и отключений агрегатов с контролем промежуточных состояний и блокировками |

3.3.3.6 Состояния блоков (sys_db_hmi_state)

Состав «sys_db_hmi_state» приведен в таблице.

| Название | Тип | Значение | Описание |
|-------------|------------------------------|----------|----------------|
| none | WORD | 16#0 | Неопределенное |
| ai | sys_db_hmi_state_ai | | |
| ai_2oo3 | sys_db_hmi_state_ai_2oo3 | | |
| fr | sys_db_hmi_state_fr | | |
| rk_cont | sys_db_hmi_state_rk_cont | | |
| rk_step | sys_db_hmi_state_rk_step | | |
| valve | sys_db_hmi_state_valve | | |
| switch_gear | sys_db_hmi_state_switch_gear | | |
| contactor | sys_db_hmi_state_contactor | | |
| motor | sys_db_hmi_state_motor | | |
| upp | sys_db_hmi_state_upp | | |
| batch_pump | sys_db_hmi_state_batch_pump | | |
| protection | sys_db_hmi_state_protection | | |
| engine | sys_db_hmi_state_engine | | |
| cont_upp | sys_db_hmi_state_cont_upp | | |
| engine_v2 | sys_db_hmi_state_engine_v2 | | |

Соответствие типа объекту библиотеки представлено в таблице ниже. Значения констант будут приведены в описании к каждому объекту.

| Название | Объект |
|------------------------------|---|
| sys_db_hmi_state_ai | fb_ai. Обработка аналогового входа |
| sys_db_hmi_state_ai_2oo3 | fb_ai_2oo3. Обработка аналогового входа 2 из 3 |
| sys_db_hmi_state_batch_pump | fb_batch_pump. Насос-дозатор |
| sys_db_hmi_state_contactor | fb_contactor. Контактор |
| sys_db_hmi_state_cont_upp | fb_cont_upp. Блок управления контактор + УПП |
| sys_db_hmi_state_engine | fb_eng_cf. Коммутация электродвигателя. Схема подключения: контактор->ПЧ (одиночная ветка) |
| | fb_eng_c_fc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->ПЧ) |
| | fb_eng_c_uc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->УПП) |
| | fb_eng_sc_scfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка->контактор) + (ячейка->контактор->ПЧ->контактор) |
| | fb_eng_sf. Коммутация электродвигателя. Схема подключения: ячейка->ПЧ (одиночная ветка). |
| | fb_eng_su. Коммутация электродвигателя. Схема подключения: ячейка->УПП (одиночная ветка). |
| | fb_eng_s_fs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ПЧ->ячейка) |
| | fb_eng_s_sfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->контактор) |
| | fb_eng_s_sfs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->ячейка) |
| | fb_eng_s_sus. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->УПП->ячейка) |
| sys_db_hmi_state_engine_v2 | fb_engine_v2. Коммутация электродвигателя. Реализует последовательности включений и отключений агрегатов с контролем промежуточных состояний и блокировками |
| sys_db_hmi_state_fr | fb_fr. Управление ПЧ. |
| sys_db_hmi_state_motor | fb_motor. Управление двигателем с возможностью реверсивного движения. |
| sys_db_hmi_state_protection | fb_protection. Технические защиты. Выдает сигнал аварии на защищаемый контур или техпроцесс. |
| sys_db_hmi_state_rk_cont | fb_rk_cont. Клапан с аналоговым управлением |
| sys_db_hmi_state_rk_step | fb_rk_step. Клапан с дискретным управлением |
| sys_db_hmi_state_switch_gear | fb_switch_gear. Управление электрической ячейкой |
| sys_db_hmi_state_upp | fb_upp. Управление УПП |
| sys_db_hmi_state_valve | fb_valve. Управление задвижкой |

3.3.3.7 Фактическое состояние объекта (sys_db_hmi_state_actual)

Состав «sys_db_hmi_state_actual» приведен в таблице.

| Название | Тип | Значение | Описание |
|-------------|-------------------------------------|----------|----------------|
| none | WORD | 16#0 | Неопределенное |
| ai | sys_db_hmi_state_actual_ai | | |
| ai_2oo3 | sys_db_hmi_state_actual_ai_2oo3 | | |
| fr | sys_db_hmi_state_actual_fr | | |
| rk_cont | sys_db_hmi_state_actual_rk_cont | | |
| rk_step | sys_db_hmi_state_actual_rk_step | | |
| valve | sys_db_hmi_state_actual_valve | | |
| switch_gear | sys_db_hmi_state_actual_switch_gear | | |
| contactor | sys_db_hmi_state_actual_contactor | | |
| motor | sys_db_hmi_state_actual_motor | | |
| upp | sys_db_hmi_state_actual_upp | | |
| batch_pump | sys_db_hmi_state_actual_batch_pump | | |
| engine | sys_db_hmi_state_actual_engine | | |

| Название | Тип | Значение | Описание |
|-----------|-----------------------------------|----------|----------|
| cont_upp | sys_db_hmi_state_actual_cont_upp | | |
| engine_v2 | sys_db_hmi_state_actual_engine_v2 | | |

Соответствие типа объекту библиотеки представлено в таблице ниже. Значения констант будут приведены в описании к каждому объекту.

| Название | Объект |
|-------------------------------------|---|
| sys_db_hmi_state_actual_ai | fb_ai. Обработка аналогового входа |
| sys_db_hmi_state_actual_ai_2oo3 | fb_ai_2oo3. Обработка аналогового входа 2 из 3 |
| sys_db_hmi_state_actual_batch_pump | fb_batch_pump. Насос-дозатор |
| sys_db_hmi_state_actual_contactor | fb_contactor. Контактор |
| sys_db_hmi_state_actual_cont_upp | fb_cont_upp. Блок управления контактор + УПП |
| sys_db_hmi_state_actual_engine | fb_eng_cf. Коммутация электродвигателя. Схема подключения: контактор->ПЧ (одиночная ветка) |
| | fb_eng_c_fc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->ПЧ) |
| | fb_eng_c_us. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (контактор) + (контактор->УПП) |
| | fb_eng_sc_sfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка->контактор) + (ячейка->контактор->ПЧ->контактор) |
| | fb_eng_sf. Коммутация электродвигателя. Схема подключения: ячейка->ПЧ (одиночная ветка). |
| | fb_eng_su. Коммутация электродвигателя. Схема подключения: ячейка->УПП (одиночная ветка). |
| | fb_eng_s_fs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ПЧ->ячейка) |
| | fb_eng_s_sfc. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->контактор) |
| | fb_eng_s_sfs. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->ПЧ->ячейка) |
| | fb_eng_s_sus. Коммутация электродвигателя. Схема подключения: 2 параллельные ветки: (ячейка) + (ячейка->УПП->ячейка) |
| sys_db_hmi_state_actual_engine_v2 | fb_engine_v2. Коммутация электродвигателя. Реализует последовательности включений и отключений агрегатов с контролем промежуточных состояний и блокировками |
| sys_db_hmi_state_actual_fr | fb_fr. Управление ПЧ. |
| sys_db_hmi_state_actual_motor | fb_motor. Управление двигателем с возможностью реверсивного движения. |
| sys_db_hmi_state_actual_rk_cont | fb_rk_cont. Клапан с аналоговым управлением |
| sys_db_hmi_state_actual_rk_step | fb_rk_step. Клапан с дискретным управлением |
| sys_db_hmi_state_actual_switch_gear | fb_switch_gear. Управление электрической ячейкой |
| sys_db_hmi_state_actual_upp | fb_upp. Управление УПП |
| sys_db_hmi_state_actual_valve | fb_valve. Управление задвижкой |

3.3.3.8 Константы насосного агрегата (sys_db_pump)

Состав «sys_db_pump» приведен в таблице.

| Название | Тип | Описание |
|------------------|-------------------------------------|---|
| process_state | <u>sys_db_pump_process_state</u> | Шаги процессов пуска и останова. Предназначены для использования разработчиками библиотеки. |
| state | <u>sys_db_pump_state</u> | Состояния. Предназначены для использования разработчиками библиотеки. |
| hmi_state | <u>sys_db_pump_hmi_state</u> | состояния блока для HMI |
| hmi_state_actual | <u>sys_db_pump_hmi_state_actual</u> | Фактические состояния объекта для HMI |
| cmd | <u>sys_db_pump_cmd</u> | команды |
| cfg | <u>sys_db_pump_cfg</u> | конфигурация |

| Название | Тип | Описание |
|----------------|-----------------------------------|----------------|
| alarm | <u>sys_db_pump_alarm</u> | аварии |
| ready_to_start | <u>sys_db_pump_ready_to_start</u> | условия пуска |
| warning | <u>sys_db_pump_warning</u> | предупреждения |

Значения констант будут приведены в описании объекта.

3.3.3.9 Константы насосной станции (*sys_db_pump_sau*)

Состав «sys_db_pump_sau» приведен в таблице.

| Название | Тип | Описание |
|------------------|---|---|
| state | <u>sys_db_pump_sau_state</u> | Состояния. Предназначены для использования разработчиками библиотеки. |
| hmi_state | <u>sys_db_pump_sau_hmi_state</u> | состояния блока для HMI |
| cmd | <u>sys_db_pump_sau_cmd</u> | команды |
| cfg | <u>sys_db_pump_sau_cfg</u> | конфигурация |
| alarm | <u>sys_db_pump_sau_alarm</u> | аварии |
| ready_to_start | <u>sys_db_pump_sau_ready_to_start</u> | условия пуска |
| warning_to_work | <u>sys_db_pump_sau_warning_to_work</u> | предупреждения |
| hmi_state_actual | <u>sys_db_pump_sau_hmi_state_actual</u> | Фактические состояния объекта для HMI |

Значения констант будут приведены в описании объекта.

3.3.3.10 Внутренние состояния блоков (*sys_db_state*)

Данные константы предназначены для использования разработчиками библиотеки.

| Название | Тип |
|-------------|---------------------------------|
| ai | <u>sys_db_state_ai</u> |
| ai_2003 | <u>sys_db_state_ai_2003</u> |
| fr | <u>sys_db_state_fr</u> |
| rk_cont | <u>sys_db_state_rk_cont</u> |
| rk_step | <u>sys_db_state_rk_step</u> |
| valve | <u>sys_db_state_valve</u> |
| switch_gear | <u>sys_db_state_switch_gear</u> |
| contactor | <u>sys_db_state_contactor</u> |
| motor | <u>sys_db_state_motor</u> |
| upp | <u>sys_db_state_upp</u> |
| batch_pump | <u>sys_db_state_batch_pump</u> |
| protection | <u>sys_db_state_protection</u> |
| engine | <u>sys_db_state_engine</u> |
| cont_upp | <u>sys_db_state_cont_upp</u> |
| engine_v2 | <u>sys_db_state_engine_v2</u> |

3.3.3.11 Режимы (*sys_db_mode*)

| Режим | Описание |
|---------------------------|--|
| Ремонт | Ремонт. В данном режиме агрегат находится в ремонте. Его невозможно включить, выключить, открыть, закрыть. Также на агрегат, находящийся в ремонте, не оказывает влияние воздействие САУ и контуров регулирования. |
| Ручной (местный) | Местный режим. В данном режиме агрегат управляется по месту (со шкафа управления). Дистанционное управление и управление от САУ, контуров регулирования невозможно. |
| Дистанционный (диспетчер) | Дистанционный режим. В данном режиме агрегатом управляет оператор, он может агрегат включить, выключить, открыть, закрыть и т.д. САУ и контуры регулирования не оказывают воздействия на агрегат, за исключением технологических блокировок. |
| Автоматический | В данном режиме управление агрегатом осуществляет САУ либо контуры регулирования. Команды диспетчера игнорируются. |

Состав «sys_db_mode» приведен в таблице.

| Название | Тип | Значение | Описание |
|------------|------|----------|-----------------------------|
| none | WORD | 16#0 | Режим не задан |
| reserve | WORD | 16#1 | Резерв |
| repair | WORD | 16#2 | Ремонт |
| manual | WORD | 16#4 | Местный(Ручной) |
| auto | WORD | 16#8 | Автоматический |
| hmi_mask | WORD | 16#FFF0 | Маска диспетчерских режимов |
| station_1 | WORD | 16#10 | Диспетчерский 1 |
| station_2 | WORD | 16#20 | Диспетчерский 2 |
| station_3 | WORD | 16#40 | Диспетчерский 3 |
| station_4 | WORD | 16#80 | Диспетчерский 4 |
| station_5 | WORD | 16#100 | Диспетчерский 5 |
| station_6 | WORD | 16#200 | Диспетчерский 6 |
| station_7 | WORD | 16#400 | Диспетчерский 7 |
| station_8 | WORD | 16#800 | Диспетчерский 8 |
| station_9 | WORD | 16#1000 | Диспетчерский 9 |
| station_10 | WORD | 16#2000 | Диспетчерский 10 |

3.3.4 Типы (Types)

Для взаимодействия блоков с HMI и обмена данными между блоками созданы структуры с набором переменных. Некоторые структуры могут входить в состав других.

Список структур:

- udt_ai_2003,
- udt_analog,
- udt_batch_pump,
- udt_consumer_status,
- udt_cont_upp,
- udt_contactor,
- udt_datetime,
- udt_discrete,
- udt_discrete_v2,
- udt_engine,
- udt_engine_ac,
- udt_engine_v2,
- udt_engine_v2_cfg,
- udt_engine_v2_cfg_line,
- udt_engine_v2_line,
- udt_fr,

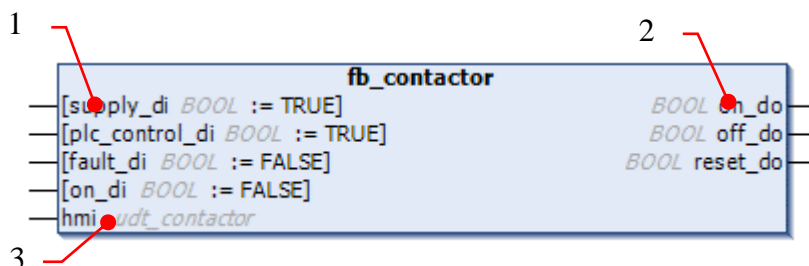
- udt_motor,
- udt_pid,
- udt_pid_settings,
- udt_protection,
- udt_pump,
- udt_pump_sau,
- udt_pump_sau_number,
- udt_pump_state_detail,
- udt_rk_cont,
- udt_rk_step,
- udt_standard,
- udt_switch_gear,
- udt_upp,
- udt_valve.

Использование типов и их состав будут приведены в описании блоков.

4 Описание блоков

В этом разделе для каждого блока будет представлено: подробное описание, примеры использования, список входов-выходов, список и состав используемых типов, описание конфигурации и пример вызова.

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. Список всех структур приведен в п. 3.3.4.

| 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 |

| Название | Тип | Начальное значение | Назначение | Комментарий |
|--------------------|-------|--------------------|------------|--|
| 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. Список режимов приведен в таблице п. 3.3.3.11.

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 (см. п. 3.3.3.4).

4.5.4 Конфигурация

Переменной «in_config» определяется набор функций работы блока. Например, у блока обработки аналогового входа (fb_ai), 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 |
| test_mode | WORD | 16#4000 | Режим имитации Блок не учитывает данные входов и не подает сигналы на выходы. Изменение состояний происходит по подаче команд. 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 |

| Название | Тип | Значение | Описание |
|-------------|------|----------|--------------|
| 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 Обработка аналогового входа (fb_ai)

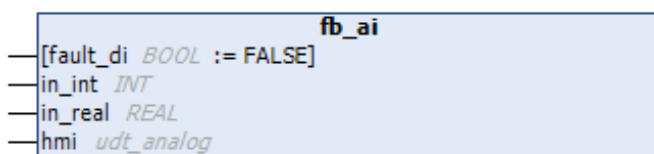
4.6.1 Назначение

Преобразование поступающих значений с модулей аналогового входа либо переменных с других блоков.

4.6.2 Функции

- преобразование входного сигнала в физические единицы,
- преобразование входного сигнала в инженерные единицы,
- сигнализация о достижении преобразованной величины аварийных и предупредительных значений,
- аппроксимация входной величины по линейному закону,
- замещение входной величины введенным вручную значением.

4.6.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------------|--------------------|---|
| Input | fault_di | BOOL | FALSE | неисправность |
| | in_int | INT | | значение на входе модуля (код АЦП) |
| | in_real | REAL | | значение от других блоков (работа в режиме пересчета) |
| Inout | hmi | udt_analog | | интерфейс блока |

4.6.4 Принцип работы

Сигнал (код АЦП), поступающий на вход `in_int` с модуля аналогового ввода, преобразуется по формуле линейной интерполяции в промежуточный сигнал в инженерных единицах (мА, В, Ом). Это значение отражается в переменной `out_x` вход-выходной структуры `hmi`.

Далее значение `out_x` по формуле линейной интерполяции пересчитывается в выходное значение в физических единицах (гр.С, Па, мЗ/ч и т.п.). Это значение отражается в переменной `out_y` вход-выходной структуры `hmi`. Значение переменной `out_y` может быть использовано далее в алгоритмах управления и/или выведено для отображения в графический интерфейс пользователя.

Вход `in_real` используется как альтернатива входу `in_int` для подачи сигналов вещественного типа для обработки на вход блока.

Вход `fault_di` предназначен для перевода блока в состояние «неисправность» установкой этого входа в значение «true».

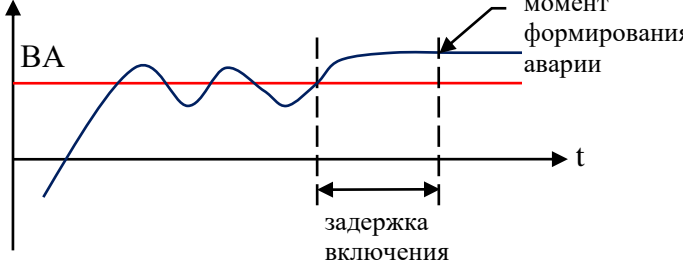
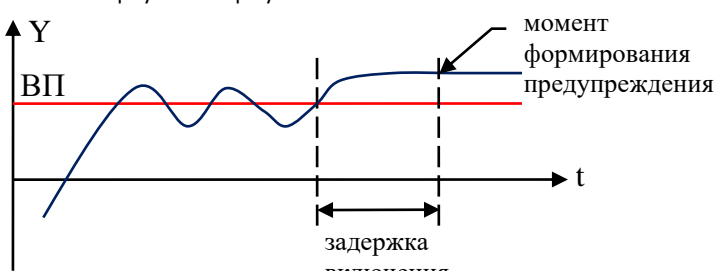
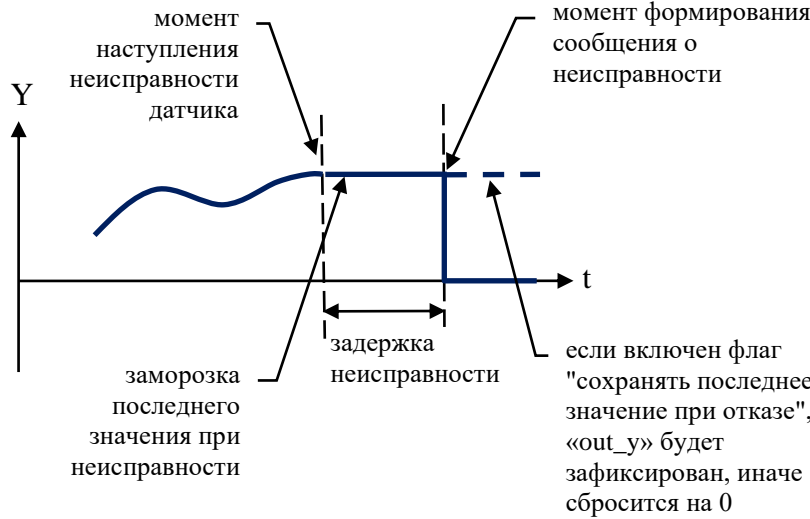
4.6.5 Интерфейс «udt_analog»

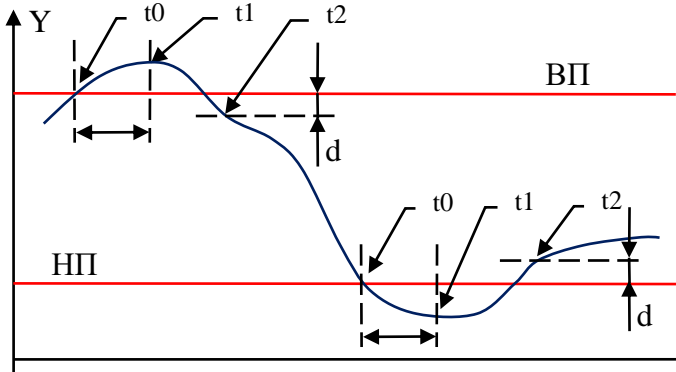
В следующей таблице представлен интерфейс блока, представленный типом данных «udt_analog».

| Название | Тип | Начальное значение | Назначение | Комментарий |
|------------------|---------------------|--------------------|------------|---|
| std | <u>udt_standard</u> | | | Стандартный набор переменных |
| out_y | REAL | | вывод | значение физической величины |
| out_x | REAL | | вывод | преобразованное значение(мА,В) |
| in_x1 | REAL | 4 | ввод | Преобразованное значение(мА,В). Первая калибровочная точка |
| in_x2 | REAL | 20 | ввод | Преобразованное значение(мА,В). Вторая калибровочная точка |
| in_y1 | REAL | 0 | ввод | Значение физической величины. Первая калибровочная точка |
| in_y2 | REAL | 100 | ввод | Значение физической величины. Вторая калибровочная точка |
| in_x_qmin | REAL | 3.5 | ввод | нижний физический предел датчика (отрицательная перегрузка) |
| in_x_qmax | REAL | 20.5 | ввод | верхний физический предел датчика (перегрузка) |
| in_y_wmin | REAL | 0 | ввод | нижняя предупредительная граница |
| in_y_wmax | REAL | 100 | ввод | верхняя предупредительная граница |
| in_y_amin | REAL | 0 | ввод | нижняя аварийная граница |
| in_y_amax | REAL | 100 | ввод | верхняя аварийная граница |
| in_avg_precision | REAL | 0 | ввод | дрейф нуля в инженерных единицах |
| in_avg_time | REAL | 0 | ввод | время усреднения |
| in_man_value | REAL | | ввод | Значение ручного ввода |
| in_time_alm | REAL | 3 | ввод | задержка включения аварий |
| in_time_wrn | REAL | 3 | ввод | задержка включения предупреждений |
| in_time_fault | REAL | 3 | ввод | задержка включения неисправности датчика |
| in_hyst | REAL | 0.1 | ввод | гистерезис на отпускание аварийных и предупредительных сообщений |
| in_block_type | INT | | ввод | Выбор обрабатываемой переменной, 0-AI, 1-AI по второму входу, 2-PIW, 3-IW, для АО не используется |
| in_hw_index | INT | | ввод | адрес аналогового канала в аппаратной конфигурации. (используется только в платформе Siemens) |
| in_sensor | INT | 5 | ввод | Выбор диапазона значений входного электрического сигнала |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |

Таблица описания параметров блока.

| Наименование параметра | Описание |
|------------------------|---|
| in_sensor | Определяет тип выхода датчика (0-20 мА, 4-20 мА и т.п.). Данный параметр влияет на расчет значения X (т.е. пересчет кода АЦП, получаемого непосредственно с модуля входов, в инженерные единицы - ток, напряжение и др. на выходе датчика) и не влияет на установленный в настройках модуля аналоговых входов тип входа. |
| out_x | Измеренное значение выходного параметра датчика (ток, напряжение, сопротивление и т.п.). Рассчитывается по формуле: $X = A + ([\text{код АЦП}] - B) * C / D$, где A, B, C, D - константы, зависящие от типа сенсора (in_sensor). Может подаваться без преобразования, если выбран соответствующий тип сенсора. |
| in_x1 | Задание диапазона измеренных значений на выходе датчика для расчета физической величины, измеряемой датчиком. Обычно задаются значения, указанные непосредственно на корпусе датчика, например, in_x1= 4 мА, in_x2 = 20 мА. |
| in_x2 | |
| in_x_qmin | Минимальное и максимальное значения тока, напряжения и др. на выходе датчика, при выходе за которые система будет считать, что датчик неисправен. Т.е. если out_x < in_x_qmin или out_x > in_x_qmax, датчик в перегрузке. |
| in_x_qmax | |

| Наименование параметра | Описание |
|------------------------|--|
| out_y | <p>Значение физической величины, измеряемой датчиком (температура, давление, электропроводность и т.п.). Рассчитывается по формуле линейной интерполяции:</p> $Y = Y1 + (X-X1)*(Y2-Y1)/(X2-X1),$ <p>где X1,X2 – первая(in_x1) и вторая(in_x2) калибровочные точки в инженерных единицах (мА, В и др.), Y1,Y2 – первая(in_y1) и вторая(in_y2) калибровочные точки в физических единицах (гр.С, бар, кгс/см2 и др.).</p> |
| in_y1 | Задание диапазона реальных измеренных значений физической величины, измеряемой датчиком. |
| in_y2 | Обычно задаются значения, указанные непосредственно на корпусе датчика, например, in_y1 = 0 бар, in_y2 = 16 бар. |
| in_y_wmin | Задание нижнего и верхнего значений физической величины, при выходе за которые формируется предупредительное сообщение. |
| in_y_wmax | |
| in_y_amin | Задание нижнего и верхнего значений физической величины, при выходе за которые формируется аварийное сообщение. |
| in_y_amax | |
| in_time_alm | <p>Время задержки в секундах, по истечении которого формируется авария при выходе значения «out_y» за верхний или нижний аварийный предел, если в течение этого времени сигнал не вернулся в норму.</p>  |
| in_time_wrn | <p>Время задержки в секундах, по истечении которого формируется авария при выходе значения «out_y» за верхний или нижний предупредительный предел, если в течение этого времени сигнал не вернулся в норму.</p>  |
| in_time_fault | <p>Время в секундах, по истечении которого параметр переходит в состояние неисправности, если в течение этого времени сигнал не вернулся в диапазон нижнего и верхнего физического предела исправности ($out_x \geq in_x_qmin$, $out_x \leq in_x_qmax$).</p>  |

| Наименование параметра | Описание |
|------------------------|--|
| in_avg_precision | Отсеивает колебание инженерной единицы возле первой калибровочной точки. |
| in_avg_time | Время усреднения физической величины в секундах по алгоритму скользящего среднего. |
| in_hyst | <p>Параметр, который устанавливает порог отпущения верхнего и нижнего аварийных и предупредительных пределов физической величины датчика. Измеряется в единицах измерения физической величины. Применяется, когда колебания значения физической величины вблизи аварийного или предупредительного порога срабатывания приводят к частому формированию/снятию сообщений.</p>  <p>t0 - момент выхода параметра за аварийное (предупредительное) значение t1 - момент формирования сообщения о выходе значения за аварийные (предупредительные) пределы t2 - момент снятия сообщения о выходе значения за аварийные (предупредительные) пределы d - гистерезис на отпущение (снятие) сообщения</p> |
| in_block_type | Выбор обрабатываемой переменной |
| in_hw_index | Используется только на платформе Siemens |

4.6.6 Переменные интерфейса блока. Списки значений

4.6.6.1 Команды «std.in_cmd»

У данного блока нет индивидуальных команд, а есть только общие команды, которые описаны в 3.3.3.4.

4.6.6.2 Аварийные сообщения «std.out_alarm»

Тип глобальных констант - «sys_db_alm_ai» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текст сообщения/Описание |
|------------------|------|----------|--|
| fault | WORD | 16#1 | «Неисправность». формируется по коду АЦП для платформы Siemens или по входному сигналу «fault_di» |
| overload_h | WORD | 16#2 | «Верхний физический предел (перегрузка)». превышение входного электрического сигнала на модуле ввода PLC выше «in_x_qmax» |
| overload_l | WORD | 16#4 | «Нижний физический предел (отрицательная перегрузка)». входной электрический сигнал на модуле ввода PLC ниже параметра «in_x_qmin» |
| incorrect_calibr | WORD | 16#8 | «Неправильная калибровка». значения первой (in_x1) и второй (in_x2) калибровочной точек в инженерных единицах равны между собой |
| h | WORD | 16#10 | «Верхний аварийный предел». измеряемая физическая величина превысила установленный параметр «in_y_amax» |
| l | WORD | 16#20 | «Нижний аварийный предел». измеряемая физическая величина ниже установленного параметра «in_y_amin» |

4.6.6.3 Предупредительные сообщения «std.out_warning»

Тип глобальных констант - «sys_db_wrn_ai» п. 3.3.3.2. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------|------|----------|---|
| h | WORD | 16#1 | «Верхний предупредительный предел». измеряемая физическая величина превысила установленный параметр «in_y_wmax» |
| l | WORD | 16#2 | «Нижний предупредительный предел». измеряемая физическая величина ниже установленного параметра «in_y_wmin» |
| man_on | WORD | 16#4 | «Ручной ввод». датчик переведен в режим ручного ввода |

4.6.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_ai» п. 3.3.3.3. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------------|------|----------|--|
| save_out | WORD | 16#1 | При включении данного флага параметр «out_y» будет принимать последнее измеренное значение, которое было зафиксировано перед отказом датчика. |
| wrn_h_msg_off | WORD | 16#2 | Отключает формирование сообщения о выходе значения физической величины за верхнюю предупредительную границу. |
| wrn_l_msg_off | WORD | 16#4 | Отключает формирование сообщения о выходе значения физической величины за нижнюю предупредительную границу. |
| negative_value_off | WORD | 16#8 | При включении отсекает получившиеся во время расчета физической величины отрицательные значения (вместо них выводится 0) |
| hold_alarm | WORD | 16#10 | Флаг включает фиксацию аварийных сообщений по выходу значения за аварийные границы. При формировании аварийного сообщения оно не исчезает после возвращения значения в норму, снять сообщение можно только кнопкой «сброс аварии». |
| alm_h_msg_off | WORD | 16#20 | Отключает формирование сообщения о выходе значения физической величины за верхнюю аварийную границу. |
| alm_l_msg_off | WORD | 16#40 | Отключает формирование сообщения о выходе значения физической величины за нижнюю аварийную границу. |
| interp_square | WORD | 16#80 | Флаг включает расчет физической величины по формуле квадратичной интерполяции: $Y = Y1 + (Y2 - Y1) \sqrt{\frac{X - X1}{X2 - X1}}$ |

4.6.6.5 Состояние блока «std.out_state»

Тип глобальных констант - «sys_db_hmi_state_ai» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|------------------|------|----------|-------------------------|
| normal | WORD | 16#1 | Норма |
| warning | WORD | 16#2 | Предупреждение |
| alarm | WORD | 16#4 | Авария |
| fault | WORD | 16#8 | Неисправность |
| overload | WORD | 16#10 | Перегрузка |
| incorrect_calibr | WORD | 16#20 | Неправильная калибровка |

4.6.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_ai» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|---------------|
| normal | WORD | 16#1 | В норме |
| fault | WORD | 16#8 | Неисправность |

4.6.6.7 Тип блока «in_block_type»

Выбор обрабатываемой переменной. Тип глобальных констант - «sys_db_const_ai» п. 3.3.3.5. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------------|-----|----------|--|
| block_type_word | INT | 0 | вход №1 работает с переменными типа int (значение на входе модуля (код АЦП)) |
| block_type_real | INT | 1 | вход №2 работает с переменными типа real (значение от других блоков (работа в режиме пересчета)) |
| block_type_piwi | INT | 2 | вход №3 работает с PIW контроллера (используется только в платформе Siemens) |
| block_type_iw | INT | 3 | вход №4 работает с IW контроллера (используется только в платформе siemens) |

4.6.6.8 Тип сенсора «in_sensor».

Выбор диапазона значений входного электрического сигнала. Тип глобальных констант - «sys_db_const_ai» п. 3.3.3.5. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-------------------|-----|----------|--------------------|
| sensor_10_10v | INT | 0 | -10..+10 V |
| sensor_0_10v | INT | 1 | 0..+10 V |
| sensor_0_5v | INT | 2 | 0..5 V |
| sensor_1_5v | INT | 3 | 1..5 V |
| sensor_0_20ma | INT | 4 | 0..20 mA |
| sensor_4_20ma | INT | 5 | 4..20 mA |
| sensor_rtd_cl | INT | 6 | RTD Climatic |
| sensor_rtd_st | INT | 7 | RTD Standard |
| sensor_no_convert | INT | 8 | без преобразования |

4.6.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_ai для аналогового датчика
    dbs_sensor_1: RuDrive.udt_analog;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_ai для аналогового датчика
    dbi_sensor_1: RuDrive.fb_ai;
```

```
END_VAR
```

```
// вызов экземпляра блока fb_ai для аналогового датчика  
dbi_sensor_1(fault_di := FALSE, in_real := sensor_value, hmi := dbs_sensor_1 );
```

4.7 Обработка аналогового входа 2 из 3 (fb_ai_2oo3)

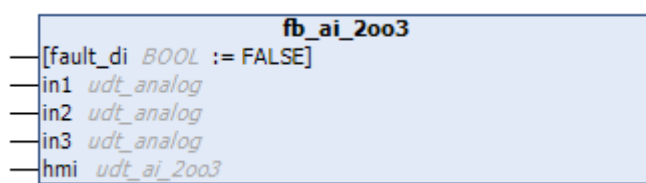
4.7.1 Назначение

Блок предназначен для резервирования каналов измерений датчиков и преобразования этих показаний в одно выходное значение.

4.7.2 Функции

- сигнализация о достижении выбранной величины аварийных и предупредительных значений,
- замещение входной величины введенным вручную значением.

4.7.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|-------------|--------------------|-----------------|
| Input | fault_di | BOOL | FALSE | неисправность |
| | in1 | udt_analog | | датчик 1 |
| | in2 | udt_analog | | датчик 2 |
| | in3 | udt_analog | | датчик 3 |
| Inout | hmi | udt_ai_2oo3 | | интерфейс блока |

4.7.4 Принцип работы

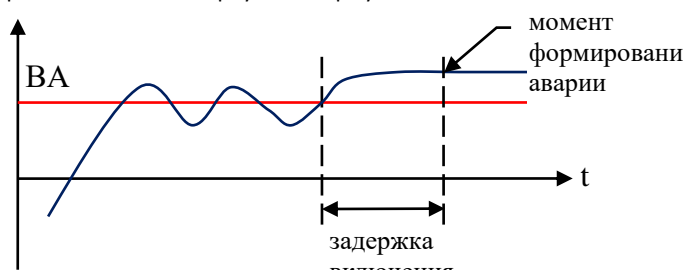
- Объект типа «строенный аналоговый вход» осуществляет резервирование каналов измерений для критически важных параметров. Резервирование осуществляется по схеме два из трех (2oo3). Деграция осуществляется по схеме 2oo3 -> 1oo2 -> 1oo1 (исправны три датчика -> исправны два датчика -> исправен один датчик).
- Выбор датчика, показания которого считаются правильными и принимаются как выходные, производится по мажоритарному алгоритму: значения всех трех датчиков располагаются в порядке возрастания, и в качестве истинного берется центральный элемент.
- Если исправны только два датчика, то за истинное принимаются показания того датчика, выходное значение которого больше второго (или меньше, это зависит от конфигурации, которая задается в параметре in_config структуры hmi).
- На вход блока подаются переменные типа «udt_analog» каждого из трех обрабатываемых датчиков.
- Вход fault_di предназначен для перевода блока в состояние «неисправность» установкой входа в значение «true».

4.7.5 Интерфейс «udt_ai_2oo3»

В следующей таблице представлен интерфейс блока, представленный типом данных «udt_ai_2oo3».

| Название | Тип | Начальное значение | Назначение | Комментарий |
|---------------|--------------|--------------------|------------|--|
| std | udt_standard | | | стандартный набор переменных |
| out_y | REAL | | вывод | значение физической величины |
| out_y1 | REAL | | вывод | значение датчика 1 |
| out_y2 | REAL | | вывод | значение датчика 2 |
| out_y3 | REAL | | вывод | значение датчика 3 |
| in_y_wmin | REAL | 0 | ввод | нижняя предупредительная граница |
| in_y_wmax | REAL | 100 | ввод | верхняя предупредительная граница |
| in_y_amin | REAL | 0 | ввод | нижняя аварийная граница |
| in_y_amax | REAL | 100 | ввод | верхняя аварийная граница |
| in_tolerance | REAL | 0.1 | ввод | допустимое рассогласование |
| in_man_value | REAL | | ввод | значение ручного ввода |
| in_time_alm | REAL | 3 | ввод | задержка включения аварий |
| in_time_wrn | REAL | 3 | ввод | задержка включения предупреждений |
| in_time_fault | REAL | 3 | ввод | задержка включения неисправности датчика |
| in_y1 | REAL | 0 | ввод | максимальное выходное значение |
| in_y2 | REAL | 100 | ввод | минимальное выходное значение |
| in_hyst | REAL | 0.1 | ввод | гистерезис на отпусkanie аварийных и предупредительных сообщений |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |

Таблица описания параметров блока.

| Наименование параметра | Описание |
|------------------------|---|
| out_y1 | Измеренное значение физической величины, полученное с датчиков 1, 2, 3. |
| out_y2 | |
| out_y3 | |
| in_y1 | Задание диапазона значений физической величины для отображения на тренде. Ни на что более эти параметры не влияют. |
| in_y2 | |
| out_y | <p>При исправности всех трех датчиков (схема 2оо3): значения всех трех датчиков располагаются в порядке возрастания, и в качестве истинного берется центральный элемент.</p> <p>При исправности только двух датчиков (схема 1оо2): в качестве истинного берется наименьшее значение из двух. В случае, если установлен флаг "выбрать наибольшее", в качестве истинного берется наибольшее значение из двух.</p> <p>При исправности только одного датчика (схема 1оо1): выходное значение «out_y» равно выходному значению исправного датчика.</p> |
| in_y_wmin | Задание нижнего и верхнего значений физической величины, при выходе за которые формируется предупредительное сообщение. |
| in_y_wmax | |
| in_y_amin | Задание нижнего и верхнего значений физической величины, при выходе за которые формируется аварийное сообщение. |
| in_y_amax | |
| in_time_alm | <p>Время в секундах задержки, по истечении которого формируется авария при выходе значения «out_y» за верхний или нижний аварийный предел, если в течение этого времени сигнал не вернулся в норму.</p>  |

| Наименование параметра | Описание |
|------------------------|---|
| in_time_wrn | <p>Время в секундах задержки, по истечении которого формируется авария при выходе значения «out_y» за верхний или нижний предупредительный предел, если в течение этого времени сигнал не вернулся в норму.</p> <p>момент формирования предупреждения</p> <p>задержка включения</p> |
| in_time_fault | <p>Время в секундах, по истечении которого блок переходит в состояние неисправности при неисправности всех трех датчиков, если в течение этого времени хотя бы один из датчиков не вернулся в исправное состояние.</p> |
| in_tolerance | <p>Если значения одного из двух датчиков различаются со значением основного, выбранным как истинное, на указанную величину или более, формируется сообщение о сильном отличии значения этого датчика от остальных двух. Если значения всех датчиков различаются между собой более, чем на указанный допуск, формируется сообщение "большой разброс показаний между датчиками".</p> |
| in_hyst | <p>Параметр, который устанавливает порог отпускания верхнего и нижнего аварийных и предупредительных пределов физической величины датчика. Измеряется в единицах измерения физической величины. Применяется, когда колебания значения физической величины вблизи аварийного или предупредительного порога срабатывания приводят к частому формированию/снятию сообщений.</p> <p>ВП</p> <p>НП</p> <p>t0 t1 t2</p> <p>d</p> <p>t0 - момент выхода параметра за аварийное (предупредительное) значение. t1 - момент формирования сообщения о выходе значения за аварийные (предупредительные) пределы. t2 - момент снятия сообщения о выходе значения за аварийные (предупредительные) пределы. d - гистерезис на отпускание (снятие) сообщения.</p> |

4.7.6 Переменные интерфейса блока. Списки значений

4.7.6.1 Команды «std.in_cmd»

У данного блока нет индивидуальных команд, а есть только общие команды, которые описаны в 3.3.3.4.

4.7.6.2 Аварийные сообщения «std.out_alarm»

Тип глобальных констант - «sys_db_alm_ai_2oo3» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|----------|------|----------|--|
| fault | WORD | 16#1 | «Неисправность». все три датчика в состоянии «неисправность» или по входному сигналу «fault_di» |
| h | WORD | 16#2 | «Верхний аварийный предел». измеряемая физическая величина превысила установленный параметр «in_y_amax» |
| l | WORD | 16#4 | «Нижний аварийный предел». измеряемая физическая величина ниже установленного параметра «in_y_amin» |

4.7.6.3 Предупредительные сообщения «std.out_warning»

Тип глобальных констант - «sys_db_wrn_ai_2003» п. 3.3.3.2. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|--------------------|------|----------|---|
| h | WORD | 16#1 | «Верхний предупредительный предел». превышено верхнее предупредительное значение измеряемой датчиком физической величины |
| l | WORD | 16#2 | «Нижний предупредительный предел». достигнуто нижнее предупредительное значение измеряемой датчиком физической величины |
| man_on | WORD | 16#4 | «Ручной ввод». датчик был переключен оператором в режим ручного ввода |
| sensor1_diff | WORD | 16#8 | «Датчик 1 отличается от остальных». показание датчика 1 отличается со значением основного, выбранным как истинное, на указанную величину или более |
| sensor2_diff | WORD | 16#10 | «Датчик 2 отличается от остальных». показание датчика 2 отличается со значением основного, выбранным как истинное, на указанную величину или более |
| sensor3_diff | WORD | 16#20 | «Датчик 3 отличается от остальных». показание датчика 3 отличается со значением основного, выбранным как истинное, на указанную величину или более |
| sensors_dispersion | WORD | 16#40 | «Большой разброс показаний между датчиками». значения всех датчиков различаются между собой более, чем на указанный допуск |

4.7.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_ai_2003» п. 3.3.3.3. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|---------------|------|----------|--|
| save_out | WORD | 16#1 | При включении данного флага параметр «out_y» будет принимать последнее измеренное значение, которое было зафиксировано перед отказом датчика. |
| wrn_h_msg_off | WORD | 16#2 | Отключает формирование сообщения о выходе значения физической величины за верхнюю предупредительную границу. |
| wrn_l_msg_off | WORD | 16#4 | Отключает формирование сообщения о выходе значения физической величины за нижнюю предупредительную границу. |
| hold_alarm | WORD | 16#10 | Флаг включает фиксацию аварийных сообщений по выходу значения за аварийные границы. При формировании аварийного сообщения оно не исчезает после возвращения значения в норму, снять сообщение можно только кнопкой «сброс аварии». |
| alm_h_msg_off | WORD | 16#20 | Отключает формирование сообщения о выходе значения физической величины за верхнюю аварийную границу. |

| Название | Тип | Значение | Описание |
|---------------|------|----------|---|
| alm_l_msg_off | WORD | 16#40 | Отключает формирование сообщения о выходе значения физической величины за нижнюю аварийную границу. |
| select_max | WORD | 16#80 | Флаг определяет, какое значение будет взято в качестве выходного (out_y), если исправны только 2 датчика. Если флаг установлен, то будет выбрано наибольшее значение из двух, иначе наименьшее. |

4.7.6.5 Состояние блока «std.out_state»

Тип глобальных констант - «sys_db_hmi_state_ai_2003» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|----------------|
| normal | WORD | 16#1 | Норма |
| warning | WORD | 16#2 | Предупреждение |
| alarm | WORD | 16#4 | Авария |
| fault | WORD | 16#8 | Неисправность |

4.7.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_ai_2003» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|---------------|
| normal | WORD | 16#1 | В норме |
| fault | WORD | 16#8 | Неисправность |

4.7.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
  // объявление интерфейса блока fb_ai для аналогового датчика №1
  dbs_sensor_1: RuDrive.udt_analog;
  // объявление интерфейса блока fb_ai для аналогового датчика №2
  dbs_sensor_2: RuDrive.udt_analog;
  // объявление интерфейса блока fb_ai для аналогового датчика №3
  dbs_sensor_3: RuDrive.udt_analog;
  // объявление интерфейса блока fb_ai_2003 для строенного датчика
  dbs_triple_sensor: RuDrive.udt_ai_2003;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
  // объявление экземпляра блока fb_ai для аналогового датчика №1
  dbi_sensor_1: RuDrive.fb_ai;
  // объявление экземпляра блока fb_ai для аналогового датчика №2
  dbi_sensor_2: RuDrive.fb_ai;
  // объявление экземпляра блока fb_ai для аналогового датчика №3
  dbi_sensor_3: RuDrive.fb_ai;
  // объявление экземпляра блока fb_ai_2003 для строенного датчика
  dbi_triple_sensor: RuDrive.fb_ai_2003;
END_VAR

// вызов экземпляра блока fb_ai для аналогового датчика №1
dbi_sensor_1(fault_di := FALSE, in_real := sensor_1_value, hmi := dbs_sensor_1 );
```

```
// вызов экземпляра блока fb_ai для аналогового датчика №2
dbi_sensor_2(fault_di := FALSE, in_real := sensor_2_value, hmi := dbs_sensor_2 );
// вызов экземпляра блока fb_ai для аналогового датчика №3
dbi_sensor_3(fault_di := FALSE, in_real := sensor_3_value, hmi := dbs_sensor_3 );

// вызов экземпляра блока fb_ai_2003 для строенного датчика
dbi_triple_sensor(
    fault_di:= FALSE,
    in1:= dbs_sensor_1,
    in2:= dbs_sensor_2,
    in3:= dbs_sensor_3,
    hmi:= dbs_triple_sensor);
```

4.8 Обработка аналогового выхода (fb_ao)

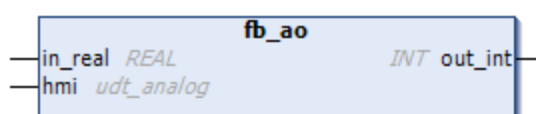
4.8.1 Назначение

Пересчет получаемого сигнала непосредственно в код ЦАП для модуля аналогового вывода и далее подачи сигнала управления на объект.

4.8.2 Функции

- преобразование входного сигнала в код ЦАП,
- замещение входной величины введенным вручную значением.

4.8.3 Таблица входов-выходов



| Область | Название | Тип | Назначение |
|---------|----------|------------|------------------------|
| Input | in_real | REAL | вход от блоков |
| Inout | hmi | udt_analog | интерфейс блока |
| Output | out_int | INT | выход на АЦП модуля АО |

4.8.4 Принцип работы

Значение физической величины, поступающее от других программных блоков на вход `in_real`, интерполируется по линейному закону по двум точкам: $\{X1, X2\}$ и $\{Y1, Y2\}$. Расчетное значение, полученное в результате интерполяции, записывается в переменную `out_x` структуры `hmi`. Обычно это значение электрической величины (мА, В, Ом), которое соответствует сигналу на выходе модуля аналогового вывода.

Далее значение переменной `out_x` пересчитывается в код ЦАП для модуля аналогового вывода по линейному закону по одной из характеристик (0...20 мА, 4...20 мА и т.д.), задаваемых параметром `in_sensor` структуры `hmi`. Расчетное значение записывается в выходную переменную `out_int`.

4.8.5 Интерфейс «udt_analog»

Из-за схожести блоков применяется тип «udt_analog». Состав представлен в п. **Ошибка! Источник с ссылки не найден.**

Таблица описания параметров блока.

| Наименование параметра | Описание |
|------------------------|---|
| in_sensor | Определяет тип выхода модуля ввода-вывода (0-20 мА, 4-20 мА и т.п.), который далее подключается непосредственно на вход исполнительного механизма (клапана, ПЧ и т.п.). Данный параметр влияет на пересчет значения «out_x» (расчетное значение выхода в инженерных единицах - мА, В и др.) в код АЦП |

| Наименование параметра | Описание |
|------------------------|---|
| out_x | Текущее расчетное значение выхода модуля аналогового вывода. Рассчитывается по входному значению «out_y», по формуле линейной интерполяции: $X = X1 + (Y-Y1)*(X2-X1)/(Y2-Y1),$ где X1, X2 - первая и вторая калибровочные точки в инженерных единицах (мА, В и др.), Y1, Y2 - первая и вторая калибровочные точки в физических единицах (Гц, % и др.). Далее, значение X пересчитывается в код АЦП модуля аналогового вывода по формуле: $X = A + ([\text{код ЦАП}] - B) * C / D,$ где A, B, C, D - константы, зависящие от типа выхода (in_sensor). |
| in_x1 | Задание диапазона выходных значений канала модуля ввода-вывода в инженерных единицах (мА, В). Обычно задаются значения конфигурации выхода модуля. |
| in_x2 | |
| out_y | Входной параметр объекта. Поступает для пересчета в код ЦАП для модуля аналогового вывода. |
| in_y1 | Задание диапазона реальных значений физической величины, поступающей на вход блока (Гц, % и др.). Обычно задаются значения, соответствующие реальным физическим параметрам исполнительного механизма. Например, для клапана Y1 = 0 %, Y2 = 100%. |
| in_y2 | |
| in_block_type | Выбор выходной переменной |
| in_hw_index | Используется только на платформе Siemens |

4.8.6 Переменные интерфейса блока. Списки значений

4.8.6.1 Команды «std.in_cmd»

У данного блока нет индивидуальных команд, а есть только общие команды, которые описаны в 3.3.3.4.

4.8.6.2 Аварийные сообщения «std.out_alarm»

Тип глобальных констант - «sys_db_alm_ai» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/ Описание |
|------------------|------|----------|---|
| incorrect_calibr | WORD | 16#8 | «Неправильная калибровка». Значения первой (in_x1) и второй (in_x2) калибровочной точек в инженерных единицах равны между собой |

4.8.6.3 Предупредительные сообщения «std.out_warning»

Тип глобальных констант - «sys_db_wrn_ai» п. 3.3.3.2. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/Описание |
|----------|------|----------|---|
| man_on | WORD | 16#4 | «Ручной ввод». Датчик переведен в режим ручного ввода |

4.8.6.4 Конфигурация «std.in_config»

Данный блок не имеет конфигураций.

4.8.6.5 Состояние блока «std.out_state»

Тип глобальных констант - «sys_db_hmi_state_ai» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|------------------|------|----------|-------------------------|
| incorrect_calibr | WORD | 16#20 | Неправильная калибровка |

4.8.6.6 Состояние объекта «std.out_state_actual»

В данном блоке не используются состояния объекта.

4.8.6.7 Тип блока «in_block_type»

Выбор выходной переменной. Тип глобальных констант - «sys_db_const_ao» п. 3.3.3.5. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------------|-----|----------|--|
| block_type_word | INT | 0 | выход №1 работает с переменными типа int (значение на выходе модуля - код АЦП) |
| block_type_piwi | INT | 2 | выход №3 работает с PQW контроллера (используется только в платформе Siemens) |
| block_type_iw | INT | 3 | выход №4 работает с QW контроллера (используется только в платформе Siemens) |

4.8.6.8 Тип сенсора «in_sensor».

Выбор диапазона значений выходного электрического сигнала. Тип глобальных констант - «sys_db_const_ai» п. 3.3.3.5. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-------------------|-----|----------|--|
| sensor_10_10v | INT | 0 | -10..+10 V |
| sensor_0_10v | INT | 1 | 0..+10 V |
| sensor_0_5v | INT | 2 | 0..5 V |
| sensor_1_5v | INT | 3 | 1..5 V |
| sensor_0_20ma | INT | 4 | 0..20 mA |
| sensor_4_20ma | INT | 5 | 4..20 mA |
| sensor_no_convert | INT | 8 | без преобразования в код ЦАП. Идет преобразование только в инженерные единицы, которые в дальнейшем конвертируются из real в int и подаются на выход «out_int» |

4.8.6.9 Код ЦАП «adc_code»

На выходе блока получаем код ЦАП. Максимальное значение кода ЦАП указано в «sys_db_const_ao» в переменной «adc_code». Значение 0... «adc_code» соответствует значению выхода модуля 0...100%.

4.8.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_ao для аналогового выхода
```

```
    dbs_ao_1: RuDrive.udt_analog;  
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG  
VAR  
    // объявление экземпляра блока fb_ao для аналогового выхода  
    dbi_ao_1: RuDrive.fb_ao;  
END_VAR  
  
// вызов экземпляра блока fb_ao для аналогового выхода  
dbi_ao_1(in_real := input_value, hmi := dbs_ao_1, out_int => output_value);
```

4.9 Насос-дозатор (fb_batch_pump)

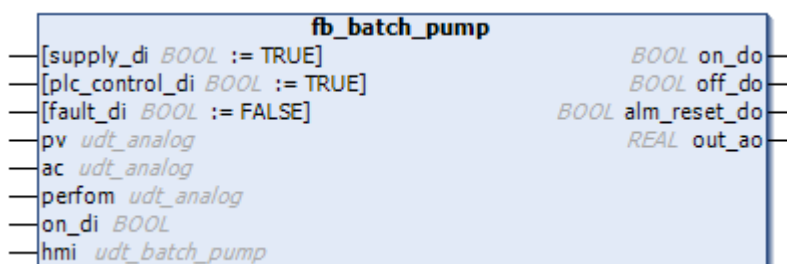
4.9.1 Назначение

Подача точного объема реагента за определенный период времени.

4.9.2 Функции

- ручной ввод производительности,
- расчет производительности в зависимости от переменной процесса и характеристик дозируемого реагента.

4.9.3 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 |
| | fault_di | BOOL | FALSE | неисправность |
| | pv | udt_analog | | переменная процесса |
| | ac | udt_analog | | ток |
| | perform | udt_analog | | производительность |
| | on_di | BOOL | | включен |
| Inout | hmi | udt_batch_pump | | интерфейс блока |
| Output | on_do | BOOL | | включить |
| | off_do | BOOL | | отключить |
| | alm_reset_do | BOOL | | сброс аварии дозатора |
| | out_ao | REAL | | производительность дозатора в % |

4.9.4 Принцип работы

По команде «пуск» блок из состояния «остановлен» перейдет в «включается». На выход «on_do» подается сигнал. При поступлении сигнала «on_di» блок перейдет в состояние «в работе».

Во время работы рассчитывается производительность дозатора по формуле:

$$\text{perform} = \frac{\text{dose} \cdot \text{pv}}{\text{density} \cdot \text{concent}} * 100, \text{ л/ч.}$$

Переменные «perform», «dose», «pv», «density» и «concent» описаны в п 4.9.5.

Затем высчитывается процент от номинальной производительности насоса и подается на выход «out_ao».

По команде «стоп» блок перейдет в состояние «отключается». С выхода «on_do» снимается сигнал, на выход «off_do» - подается. При снятии сигнала «on_di» блок перейдет в состояние «остановлен».

4.9.5 Интерфейс «udt_batch_pump»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-------------|--------------|-------------|------------|---|
| std | udt_standard | | | стандартный набор переменных |
| time_on | REAL | 2 | ввод | время включения. Максимально-допустимое время ожидания (в секундах) прихода сигнала "включен", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_off | REAL | 2 | ввод | время отключения. Максимально-допустимое время ожидания (в секундах) снятия сигнала "включен", по истечении которого объект переходит в аварийное состояние, если сигнал не был снят. |
| time_reset | REAL | 1 | ввод | Время действия (в секундах) импульса сигнала "сброс аварии" на насос-дозатор, который подается при нажатии кнопки "сброс аварии" или при подаче соответствующей команды от САУ. |
| dose | REAL | 1 | ввод | доза дозируемого вещества (г/м3) |
| nom_perform | REAL | 10 | ввод | паспортная производительность дозатора (л/ч) |
| density | REAL | 1000 | ввод | плотность дозируемого вещества (кг/м3) |
| concent | REAL | 100 | ввод | концентрация дозируемого вещества (%) |
| out_min | REAL | 0 | ввод | минимальная производительность дозатора (в %) |
| out_max | REAL | 100 | ввод | максимальная производительность дозатора (в %) |
| perform | REAL | | вывод | расчетная производительность дозатора (л/ч) |
| pv | REAL | | вывод | переменная процесса (расход м3/ч) |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |

4.9.6 Переменные интерфейса блока. Списки значений

4.9.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_batch_pump» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|----------|
| stop | WORD | 16#1 | Стоп |
| start | WORD | 16#2 | Пуск |

4.9.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_batch_pump» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|------------------|------|----------|--|
| start_timeout | WORD | 16#1 | «Превышено время включения». После подачи команды «пуск» запускается время ожидания (time_on) прихода сигнала "включен". По истечении этого времени объект переходит в аварийное состояние, если сигнал не пришел. |
| stop_timeout | WORD | 16#2 | «Превышено время отключения». После подачи команды «стоп» запускается время ожидания (time_off) снятия сигнала "включен". По истечении этого времени объект переходит в аварийное состояние, если сигнал не был снят. |
| unexpected_start | WORD | 16#4 | «Самопроизвольное включение». Поступление сигнала «включен» без подачи команды «пуск». |
| unexpected_stop | WORD | 16#8 | «Самопроизвольное отключение». Снятие сигнала «включен» без подачи сигнала «стоп». |
| fault | WORD | 16#10 | «Авария». Поступление сигнала на вход «неисправность». Данную аварию можно сбросить только после снятия сигнала. |
| power_off | WORD | 16#20 | «Пропадание питания». Снятие входного сигнала «питание подано» во время работы насоса-дозатора. |

4.9.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_batch_pump» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|---|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «питание подано». |

4.9.6.4 Конфигурация «std.in_config»

Данный блок не имеет конфигураций.

4.9.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_batch_pump» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Остановлен |
| started | WORD | 16#4 | В работе |
| stopping | WORD | 16#8 | Отключается |
| starting | WORD | 16#10 | Включается |
| alarm | WORD | 16#20 | Авария |

4.9.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_batch_pump» п. 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |

| Название | Тип | Значение | Описание |
|----------|------|----------|----------|
| started | WORD | 16#4 | Включен |

4.9.7 Пример

Объявление глобальных переменных:

VAR_GLOBAL

```
// объявление интерфейса блока fb_batch_pump для насоса дозатора
dbs_batch_pump_1: RuDrive.udt_batch_pump;
// объявление интерфейса блока fb_ai для переменной процесса насоса-дозатора
dbs_batch_pump_1_pv: RuDrive.udt_analog;
// объявление интерфейса блока fb_ai для сигнала тока насоса-дозатора
dbs_batch_pump_1_ac: RuDrive.udt_analog;
// объявление интерфейса блока fb_ai для сигнала производительности насоса-дозатора
dbs_batch_pump_1_perform: RuDrive.udt_analog;
// объявление интерфейса блока fb_ao для подачи производительности на насос-дозатор
dbs_batch_pump_1_out_ao: RuDrive.udt_analog;
```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM PLC_PRG

VAR

```
// объявление экземпляра блока fb_batch_pump для насоса дозатора
dbi_batch_pump_1 : RuDrive.fb_batch_pump;
// объявление экземпляра блока fb_ai для переменной процесса насоса-дозатора
dbi_batch_pump_1_pv : RuDrive.fb_ai;
// объявление экземпляра блока fb_ai для сигнала тока насоса-дозатора
dbi_batch_pump_1_ac : RuDrive.fb_ai;
// объявление экземпляра блока fb_ai для сигнала производительности насоса-дозатора
dbi_batch_pump_1_perform : RuDrive.fb_ai;
// объявление экземпляра блока fb_ao для подачи производительности на насос-дозатор
dbi_batch_pump_1_out_ao : RuDrive.fb_ao;
```

END_VAR

```
// вызов экземпляра блока fb_ai для сигнала тока насоса-дозатора
dbi_batch_pump_1_ac(fault_di:= FALSE, in_real:= in_real_ac, hmi:= dbs_batch_pump_1_ac);
// вызов экземпляра блока fb_ai для сигнала производительности насоса-дозатора
dbi_batch_pump_1_perform(
    fault_di:= FALSE,
    in_real:= in_real_perform,
    hmi:= dbs_batch_pump_1_perform );
// вызов экземпляра блока fb_ai для переменной процесса насоса-дозатора
dbi_batch_pump_1_pv(fault_di:= FALSE, in_real:= in_real_pv, hmi:= dbs_batch_pump_1_pv);
```

// вызов экземпляра блока fb_batch_pump для насоса дозатора

```
dbi_batch_pump_1(
    supply_di:= input_supply_di,
    plc_control_di:= input_plc_control_di,
    fault_di:= input_fault_di,
    pv:= dbi_batch_pump_1_pv,
    ac:= dbi_batch_pump_1_ac,
    perform:= dbi_batch_pump_1_perform,
    on_di:= input_on_di,
    hmi:= dbs_batch_pump_1,
    on_do=> output_on_do,
    off_do=> output_off_do,
    alm_reset_do=> output_alm_reset_do,
    out_ao=> dbi_batch_pump_1_out_ao.in_real);
```

```
// вызов экземпляра блока fb_ao для подачи производительности на насос-дозатор
dbi_batch_pump_1_out_ao( hmi:= dbs_batch_pump_1_out_ao, out_int=> out_int_value);
```

4.10 Контактор (fb_contactor)

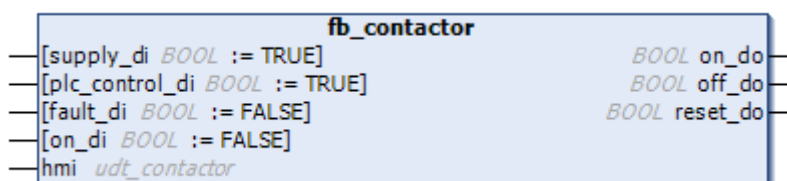
4.10.1 Назначение

Управление элементом, имеющим одно состояние (включено).

4.10.2 Функции

- управление котактором.

4.10.3 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 |
| | fault_di | BOOL | FALSE | неисправность |
| | on_di | BOOL | FALSE | включен |
| Inout | hmi | udt_contactor | | интерфейс блока |
| Output | on_do | BOOL | | включить |
| | off_do | BOOL | | отключить |
| | reset_do | BOOL | | сброс аварии |

4.10.4 Принцип работы

Блок переходит в состояние «остановлен» из состояния «нет питания», когда есть сигнал «питание подано».

По команде «пуск» блок из состояния «остановлен» перейдет в «включается». На выход «on_do» подается сигнал. При поступлении сигнала «on_di» блок перейдет в состояние «в работе».

По команде «стоп» блок перейдет в состояние «отключается». С выхода «on_do» снимается сигнал, на выход «off_do» - подается. При снятии сигнала «on_di» блок перейдет в состояние «остановлен».

4.10.5 Интерфейс «udt_contactor»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|----------|--------------|-------------|------------|------------------------------|
| std | udt_standard | | | стандартный набор переменных |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------|------------|-------------|------------|---|
| time_on | REAL | 2 | ввод | время включения. Максимально-допустимое время ожидания (в секундах) прихода сигнала "включен", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_off | REAL | 2 | ввод | время отключения. Максимально-допустимое время ожидания (в секундах) снятия сигнала "включен", по истечении которого объект переходит в аварийное состояние, если сигнал не был снят. |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |
| time_pulse | REAL | 2 | ввод | время импульса включения/отключения (сек) |
| time_reset | REAL | 2 | ввод | время сброса аварии (сек) |

4.10.6 Переменные интерфейса блока. Списки значений

4.10.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_contactor» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|----------|
| stop | WORD | 16#1 | Стоп |
| start | WORD | 16#2 | Пуск |

4.10.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_contactor» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текст сообщений/Описание |
|------------------|------|----------|---|
| start_timeout | WORD | 16#1 | «Превышено время включения». После подачи команды «пуск» запускается время ожидания (time_on) прихода сигнала "включен". По истечении этого времени объект переходит в аварийное состояние, если сигнал не пришел. |
| stop_timeout | WORD | 16#2 | «Превышено время отключения». После подачи команды «стоп» запускается время ожидания (time_off) снятия сигнала "включен". По истечении этого времени объект переходит в аварийное состояние, если сигнал не был снят. |
| unexpected_start | WORD | 16#4 | «Самопроизвольное включение». Поступление сигнала «включен» без подачи команды «пуск». |
| unexpected_stop | WORD | 16#8 | «Самопроизвольное отключение». Снятие сигнала «включен» без подачи сигнала «стоп». |
| fault | WORD | 16#10 | «Авария». Поступление сигнала на вход «неисправность». Данную аварию можно сбросить только после снятия сигнала. |
| power_off | WORD | 16#20 | «Пропадание питания». Снятие входного сигнала «питание подано» во время работы. |

4.10.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_contactor» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|---|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «питание подано». |

4.10.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_contactor» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|---|
| pulse_signal | WORD | 16#1 | Выходные сигналы «включить» и «отключить» при включении этого флага подаются кратковременно (в виде импульса) при подаче соответствующих команд, а не удерживаются постоянно. |

4.10.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_contactor» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |
| started | WORD | 16#4 | Включен |
| stopping | WORD | 16#8 | Отключается |
| starting | WORD | 16#10 | Включается |
| alarm | WORD | 16#20 | Авария |

4.10.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_contactor» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |
| started | WORD | 16#4 | Включен |

4.10.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_contactor для контактора
    dbs_contactor_1: RuDrive.udt_contactor;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_contactor для контактора
```

```
    dbi_contactor_1 : RuDrive.fb_contactor;  
END_VAR  
  
// вызов экземпляра блока fb_contactor для контактора  
dbi_contactor_1(  
    supply_di:= input_supply_di,  
    plc_control_di:= input_plc_control_di,  
    fault_di:= input_fault_di,  
    on_di:= input_on_di,  
    hmi:= dbs_contactor_1,  
    on_do=> output_on_do,  
    off_do=> output_off_do,  
    reset_do=> output_reset_do);
```

4.11 Контакттор + УПП (fb_cont_upp)

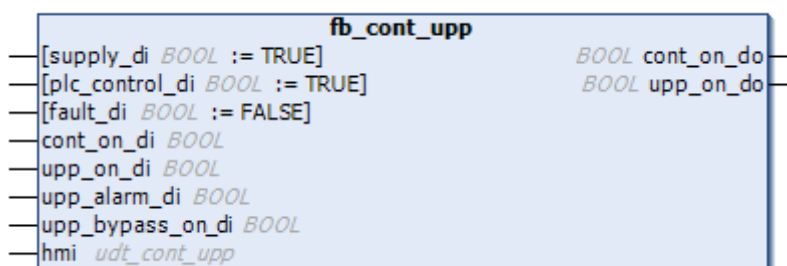
4.11.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации.

4.11.2 Функции

- запуск и останов электродвигателя с помощью УПП, подключаемого к питающей сети контактором.

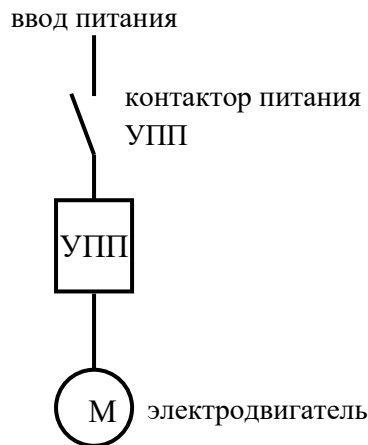
4.11.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|------------------|--------------|--------------------|--|
| Input | supply_di | BOOL | TRUE | схема собрана/питание подано |
| | plc_control_di | BOOL | TRUE | автоматическое управление |
| | fault_di | BOOL | FALSE | Неисправность (не используется) |
| | cont_on_di | BOOL | | контактор включен |
| | upp_on_di | BOOL | | УПП включен |
| | upp_alarm_di | BOOL | | авария УПП |
| | upp_bypass_on_di | BOOL | | сигнал BYPASSED - включения шунтирования |
| Inout | hmi | udt_cont_upp | | интерфейс блока |
| Output | cont_on_do | BOOL | | включить контактор |
| | upp_on_do | BOOL | | включить УПП |

4.11.4 Принцип работы

Схема коммутации показана на рисунке ниже.



Блок переходит в состояние «остановлен» из состояния «нет питания», когда есть сигнал «питание подано».

При подаче команды «пуск» блок переходит в состояние «включение контактора». Подается сигнал «включить контактор».

При поступлении сигнала «контактор включен» блок перейдет в состояние «включение УПП». Подается сигнал «включить УПП».

При поступлении сигнала «УПП включен» и «включения шунтирования» блок перейдет в состояние «работа».

По команде «стоп» блок перейдет в состояние «отключение УПП». Снимается сигнал «включить УПП».

По снятию сигналов «включения шунтирования» и «УПП включен» блок перейдет в состояние «отключение контактора».

Снимается сигнал «включить контактор». По снятию сигнала «контактор включен» блок перейдет в состояние «остановлен».

4.11.5 Интерфейс «udt_cont_upp»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------|--------------|-------------|------------|---|
| std | udt_standard | | | стандартный набор переменных |
| time_sw_on | REAL | 2 | ввод | время включения контактора. Максимально-допустимое время ожидания (в секундах) прихода сигнала "контактор включен", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_upp_on | REAL | 2 | ввод | время включения УПП. Максимально-допустимое время ожидания (в секундах) прихода сигнала "УПП включен", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_bypass_on | REAL | 5 | ввод | время включения шунтирования. Максимально-допустимое время ожидания (в секундах) прихода сигнала "включения шунтирования", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_bypass_of f | REAL | 2 | ввод | время отключения шунтирования. Максимально-допустимое время ожидания (в секундах) снятия сигнала "включения шунтирования", по истечении которого объект переходит в аварийное состояние, если сигнал не был снят. |
| time_upp_off | REAL | 5 | ввод | время отключения УПП. Максимально-допустимое время ожидания (в секундах) снятия сигнала "УПП включен", по истечении которого объект переходит в аварийное состояние, если сигнал не был снят. |
| time_sw_off | REAL | 2 | ввод | время отключения контактора. Максимально-допустимое время ожидания (в секундах) снятия сигнала "контактор включен", по истечении которого объект переходит в аварийное состояние, если сигнал не был снят. |

| Название | Тип | Нач. значе-е | Назначение | Описание |
|----------|------------|--------------|------------|----------------------------|
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |

4.11.6 Переменные интерфейса блока. Списки значений

4.11.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_cont_urr» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|----------|
| stop | WORD | 16#1 | Стоп |
| start | WORD | 16#2 | Пуск |

4.11.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_cont_urr» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|-------------------------|------|----------|---|
| power_off | WORD | 16#1 | «Пропадание питания». Снятие входного сигнала «supply_di» во время работы. |
| sw_start_timeout | WORD | 16#2 | «Превышено время включения контактора». Сигнал "cont_on_di" не пришел в течение времени «time_sw_on» после подачи выхода «cont_on_do». |
| sw_stop_timeout | WORD | 16#4 | «Превышено время отключения контактора». Сигнал "cont_on_di" не ушел в течение времени «time_sw_off» после снятия выхода «cont_on_do». |
| sw_unexpected_start | WORD | 16#8 | «Самопроизвольное включение контактора». Поступление сигнала "cont_on_di" без подачи выхода «cont_on_do». |
| sw_unexpected_stop | WORD | 16#10 | «Самопроизвольное отключение контактора». Пропадание сигнала "cont_on_di" без снятия выхода «cont_on_do». |
| upp_start_timeout | WORD | 16#20 | «Превышено время включения УПП». Сигнал "urr_on_di" не пришел в течение времени «time_urr_on» после подачи выхода «urr_on_do». |
| upp_stop_timeout | WORD | 16#40 | «Превышено время отключения УПП». Сигнал "urr_on_di" не ушел в течение времени «time_urr_off» после снятия выхода «urr_on_do» либо после пропадания входа «urr_bypass_on_di» в состоянии блока «отключение УПП». |
| upp_unexpected_start | WORD | 16#80 | «Самопроизвольное включение УПП». Поступление сигнала "urr_on_di" без подачи выхода «urr_on_do». |
| upp_unexpected_stop | WORD | 16#100 | «Самопроизвольное отключение УПП». Пропадание сигнала "urr_on_di" без снятия выхода «urr_on_do». |
| upp_fault | WORD | 16#200 | «Авария УПП» Поступление сигнала «urr_alarm_di» в состоянии блока «включение УПП» или «работа» |
| bypass_start_timeout | WORD | 16#400 | «Превышено время включения шунта УПП». Сигнал "urr_bypass_on_di" не пришел в течение времени «time_bypass_on» после поступления входа «urr_on_di». |
| bypass_stop_timeout | WORD | 16#800 | «Превышено время отключения шунта УПП». Сигнал "urr_bypass_on_di" не пришел в течение времени «time_bypass_on» после поступления входа «urr_on_di». |
| bypass_unexpected_start | WORD | 16#1000 | «Самопроизвольное включение шунта УПП». Поступление сигнала "urr_bypass_on_di" без подачи выхода «urr_on_do». |
| bypass_unexpected_stop | WORD | 16#2000 | «Самопроизвольное отключение шунта УПП». Пропадание сигнала "urr_bypass_on_di" без снятия выхода «urr_on_do». |

4.11.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_cont_urr» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|--|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |

4.11.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_cont_urr» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-------------------|------|----------|--|
| control_on | WORD | 16#1 | «Контроль включения УПП». Данный флаг включает контроль поступления и снятия сигнала «urr_on_di» и генерацию соответствующих аварийных сообщений. |
| control_bypass_on | WORD | 16#2 | «Контроль шунтирования УПП». Данный флаг включает контроль поступления и снятия сигнала «urr_bypass_on_di» и генерацию соответствующих аварийных сообщений. |

4.11.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_contactor» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|-----------------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Остановлен |
| sw_starting | WORD | 16#4 | Включение контактора |
| upp_starting | WORD | 16#8 | Включение УПП |
| started | WORD | 16#10 | В работе |
| upp_stopping | WORD | 16#20 | Отключение УПП |
| sw_stopping | WORD | 16#40 | Отключение контактора |
| alarm | WORD | 16#80 | Авария |

4.11.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_contactor» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|--|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |
| stopped | WORD | 16#2 | Отключен Есть входной сигнал «supply_di», отсутствуют «urr_on_di» и «urr_bypass_on_di». |
| started | WORD | 16#4 | «Включен». Есть входной сигнал «urr_on_di» или «urr_bypass_on_di». |

4.11.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_cont_upp для контактора с УПП
    dbs_cont_upp_1: RuDrive.udt_cont_upp;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_cont_upp для контактора с УПП
    dbi_cont_upp_1 : RuDrive.fb_cont_upp;
END_VAR

// вызов экземпляра блока fb_cont_upp для контактора с УПП
dbi_cont_upp_1(
    supply_di:= input_supply_di,
    plc_control_di:= input_plc_control_di,
    fault_di:= false,
    cont_on_di:= input_cont_on_di,
    upp_on_di:= input_upp_on_di,
    upp_alarm_di:= input_upp_alarm,
    upp_bypass_on_di:= input_upp_bypass_on_di,
    hmi:= dbs_cont_upp_1,
    cont_on_do=> output_cont_on_do,
    upp_on_do=> output_upp_on_do);
```

4.12 Обработка дискретных входов-выходов (fb_discrete)

4.12.1 Назначение

Обработка дискретных входов-выходов.

4.12.2 Функции

- отключение канала,
- инвертирование,
- задержка поступления сигнала,
- задержка снятия сигнала.

4.12.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|------------------------|---------------------|--------------------|---|
| Input | <code>in_dword</code> | DWORD | | входной DWORD. Массив входов-выходов представленный в виде двойного слова |
| | <code>in_word</code> | WORD | | входной WORD. Массив входов-выходов представленный в виде слова |
| | <code>in_byte</code> | BYTE | | входной BYTE. Массив входов-выходов представленный в виде байта |
| Inout | <code>hmi</code> | <i>udt_discrete</i> | | интерфейс блока |
| Output | <code>out_dword</code> | DWORD | | выходной DWORD. Массив обработанных входов-выходов представленный в виде двойного слова |
| | <code>out_word</code> | WORD | | выходной WORD. Массив обработанных входов-выходов представленный в виде слова |
| | <code>out_byte</code> | BYTE | | выходной BYTE. Массив обработанных входов-выходов представленный в виде байта |

При вызове экземпляра блока задействуется только один из трех входов! Бит входной переменной соответствует значению канала дискретного модуля. Нулевой канал – нулевой бит.

4.12.4 Принцип работы

Из входной переменной формируется массив дискрет на входе преобразований (`in_arr`). Каждый бит обрабатывается согласно выбранным настройкам (инверсия, отключение опроса, задержки прихода/сброса) и записывается в массив дискрет после преобразований (`out_arr`). Из него генерируются выходные переменные.

4.12.5 Интерфейс «*udt_discrete*»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-----------------|--------------------------|-------------|------------|---|
| cfg_invert_arr | ARRAY [0..31] OF BOOL | [32(FALSE)] | ввод | настройка, включающая инвертирование канала. Установленный нулевой бит массива включает инвертирование нулевого канала. |
| cfg_disable_arr | ARRAY [0..31] OF BOOL | [32(FALSE)] | ввод | настройка отключения опроса дискретности. Установленный нулевой бит массива отключает формирование нулевого элемента массива «out_arr». |
| time_on | ARRAY [0..31] OF REAL | [32(0)] | ввод | задержка прихода сигнала в секундах. |
| time_off | ARRAY [0..31] OF REAL | [32(0)] | ввод | задержка сброса сигнала в секундах |
| in_arr | ARRAY [0..31] OF BOOL | [32(FALSE)] | вывод | массив дискрет на входе преобразований |
| out_arr | ARRAY [0..31] OF BOOL | [32(FALSE)] | вывод | массив дискрет после преобразований |

4.12.6 Переменные интерфейса блока. Списки значений

4.12.6.1 Команды «std.in_cmd»

У данного блока нет команд.

4.12.6.2 Аварийные сообщения «std.out_alarm».

У данного блока нет аварийных сообщений.

4.12.6.3 Предупредительные сообщения «std.out_warning».

У данного блока нет предупредительных сообщений.

4.12.6.4 Конфигурация «std.in_config»

Данный блок не имеет констант конфигурации.

4.12.6.5 Состояние блока «std.out_state».

Данный блок не имеет констант состояний.

4.12.6.6 Состояние объекта «std.out_state_actual»

Данный блок не использует константы состояний объекта.

4.12.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
  // объявление интерфейса блока fb_discrete для модуля входов
  dbs_discrete_1: RuDrive.udt_discrete;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
  // объявление экземпляра блока fb_discrete для модуля входов
  dbi_discrete_1 : RuDrive.fb_discrete;
END_VAR
```

```
// вызов экземпляра блока fb_discrete для модуля входов
dbi_discrete_1(
    in_word:= input_word,
    hmi:= dbs_discrete_1,
    out_word=> output_word);
```

4.13 Обработка дискретных входов-выходов версия 2 (fb_discrete_v2)

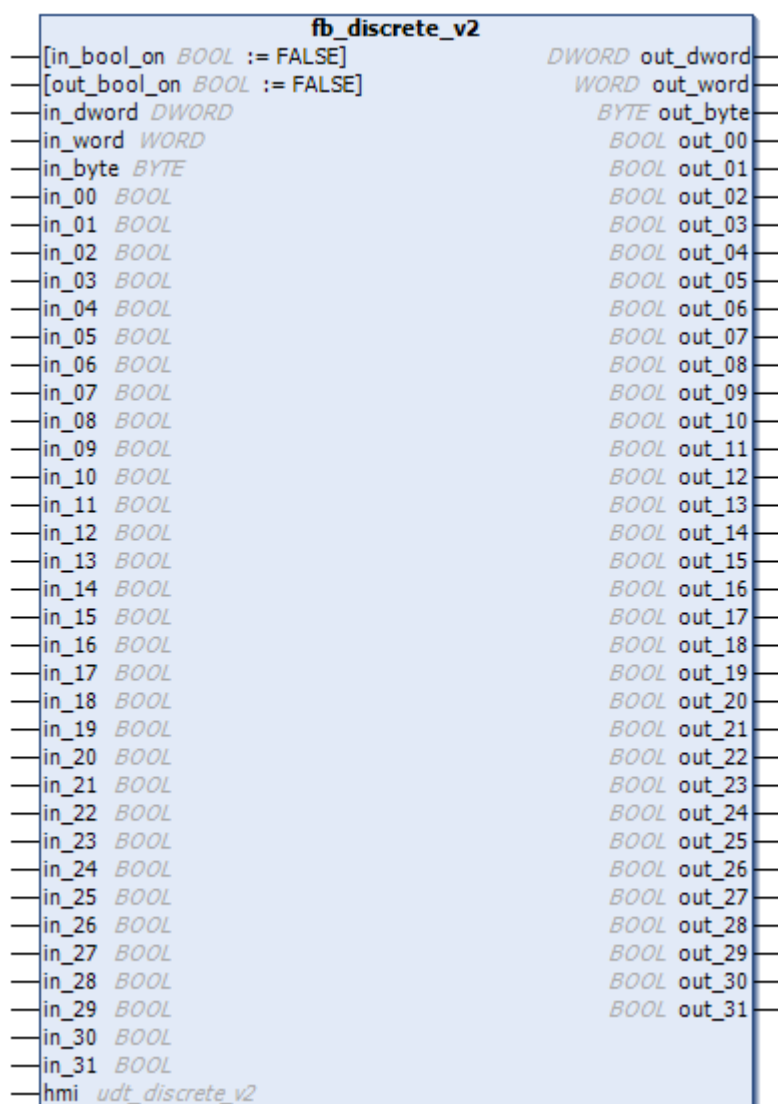
4.13.1 Назначение

Обработка дискретных входов-выходов.

4.13.2 Функции

- отключение канала,
- инвертирование,
- отключение задержек
- задержка поступления сигнала,
- задержка снятия сигнала.

4.13.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|-------------|-----------------|--------------------|------------------------------------|
| Input | in_bool_on | BOOL | FALSE | включение обработки входных битов |
| | out_bool_on | BOOL | FALSE | включение обработки выходных битов |
| | in_dword | DWORD | | входной DWORD |
| | in_word | WORD | | входной WORD |
| | in_byte | BYTE | | входной BYTE |
| | in_00 | BOOL | | входа побитно |
| | in_01 | BOOL | | |
| | in_02 | BOOL | | |
| | in_03 | BOOL | | |
| | in_04 | BOOL | | |
| | in_05 | BOOL | | |
| | in_06 | BOOL | | |
| | in_07 | BOOL | | |
| | in_08 | BOOL | | |
| | in_09 | BOOL | | |
| | in_10 | BOOL | | |
| | in_11 | BOOL | | |
| | in_12 | BOOL | | |
| | in_13 | BOOL | | |
| | in_14 | BOOL | | |
| | in_15 | BOOL | | |
| | in_16 | BOOL | | |
| | in_17 | BOOL | | |
| | in_18 | BOOL | | |
| | in_19 | BOOL | | |
| | in_20 | BOOL | | |
| | in_21 | BOOL | | |
| | in_22 | BOOL | | |
| | in_23 | BOOL | | |
| | in_24 | BOOL | | |
| | in_25 | BOOL | | |
| | in_26 | BOOL | | |
| | in_27 | BOOL | | |
| in_28 | BOOL | | | |
| in_29 | BOOL | | | |
| in_30 | BOOL | | | |
| in_31 | BOOL | | | |
| Inout | hmi | udt_discrete_v2 | | интерфейс блока |
| Output | out_dword | DWORD | | выходной DWORD |
| | out_word | WORD | | выходной WORD |
| | out_byte | BYTE | | выходной BYTE |
| | out_00 | BOOL | | выхода побитно |
| | out_01 | BOOL | | |
| | out_02 | BOOL | | |
| | out_03 | BOOL | | |
| | out_04 | BOOL | | |
| | out_05 | BOOL | | |
| | out_06 | BOOL | | |
| | out_07 | BOOL | | |
| out_08 | BOOL | | | |
| out_09 | BOOL | | | |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--------------------|------------|
| | out_10 | BOOL | | |
| | out_11 | BOOL | | |
| | out_12 | BOOL | | |
| | out_13 | BOOL | | |
| | out_14 | BOOL | | |
| | out_15 | BOOL | | |
| | out_16 | BOOL | | |
| | out_17 | BOOL | | |
| | out_18 | BOOL | | |
| | out_19 | BOOL | | |
| | out_20 | BOOL | | |
| | out_21 | BOOL | | |
| | out_22 | BOOL | | |
| | out_23 | BOOL | | |
| | out_24 | BOOL | | |
| | out_25 | BOOL | | |
| | out_26 | BOOL | | |
| | out_27 | BOOL | | |
| | out_28 | BOOL | | |
| | out_29 | BOOL | | |
| | out_30 | BOOL | | |
| | out_31 | BOOL | | |

При вызове экземпляра блока задействуется только один из трех входов (in_dword, in_word, in_byte).

4.13.4 Принцип работы

При наличии входного сигнала «in_bool_on» переменная интерфейса блока «in» формируется из входов «in_00» .. «in_31». В противном случае – из входной переменной «in_dword», «in_word», «in_byte».

Каждый бит обрабатывается согласно выбранным настройкам (инверсия, отключение опроса, задержки прихода/сброса) и записывается в «out». Генерируются выходные переменные.

При наличии входного сигнала «out_bool_on» формируются все выходы блока. В противном случае – только «out_dword», «out_word», «out_byte».

4.13.5 Интерфейс «udt_discrete_v2»

В следующей таблице представлен интерфейс блока, представленный типом данных «udt_discrete_2».

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------|-------|-------------|------------|--|
| in | DWORD | | вывод | Бит переменной соответствует значению канала дискретного модуля. Нулевой бит – нулевой канал |
| out | DWORD | | вывод | Бит переменной соответствует обработанному значению канала дискретного модуля. Нулевой бит – нулевой канал |
| invert | DWORD | | ввод | настройка, включающая инвертирование канала. Установленный нулевой бит включает инвертирование нулевого канала. |
| disable | DWORD | | ввод | настройка отключения опроса дискреты. Установленный нулевой бит переменной отключает формирование нулевого бита «out». |
| tmr_enable | DWORD | | ввод | слово активации таймеров задержки. Нулевой бит активирует таймеры задержки включения/выключения |

| Название | Тип | Нач. значе-е | Назначение | Описание |
|----------|--------------------------|--------------|------------|--------------------------------------|
| time_on | ARRAY [0..31] OF REAL | [32(0)] | ввод | задержка прихода сигнала в секундах. |
| time_off | ARRAY [0..31] OF REAL | [32(0)] | ввод | задержка сброса сигнала в секундах |

4.13.6 Переменные интерфейса блока. Списки значений

4.13.6.1 Команды «std.in_cmd»

У данного блока нет команд.

4.13.6.2 Аварийные сообщения «std.out_alarm».

У данного блока нет аварийных сообщений.

4.13.6.3 Предупредительные сообщения «std.out_warning».

У данного блока нет предупредительных сообщений.

4.13.6.4 Конфигурация «std.in_config»

Данный блок не имеет констант кофигурации.

4.13.6.5 Состояние блока «std.out_state».

Данный блок не имеет констант состояний.

4.13.6.6 Состояние объекта «std.out_state_actual»

Данный блок не использует константы состояний объекта.

4.13.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
  // объявление интерфейса блока fb_discrete_v2 для модуля входов
  dbs_discrete_v2_1: RuDrive.udt_discrete_v2;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
  // объявление экземпляра блока fb_discrete_v2 для модуля входов
  dbi_discrete_v2_1 : RuDrive.fb_discrete_v2;
END_VAR

// вызов экземпляра блока fb_discrete_v2 для модуля входов
dbi_discrete_v2_1(
  in_bool_on:= FALSE,
  out_bool_on:= FALSE,
  in_dword:= input_dword,
  hmi:= dbs_discrete_v2_1,
  out_dword=> output_dword);
```

4.14 ПЧ (fb_fr)

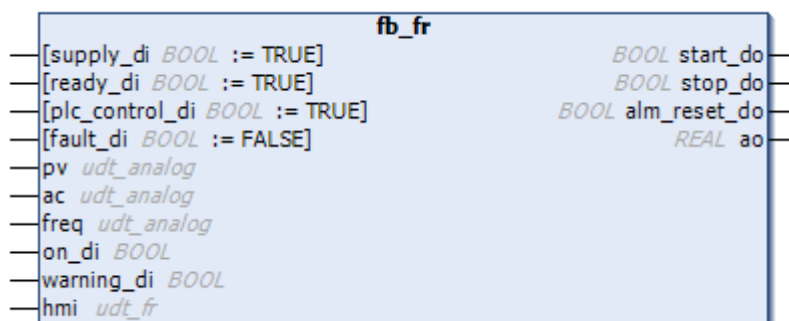
4.14.1 Назначение

Объединяет все объекты, которые напрямую подключаются к преобразователю частоты. Это могут быть насосы, вентиляторы, дозаторы или другие устройства, требующие регулирования скорости вращения.

4.14.2 Функции

- управление ПЧ,
- поддержание переменной процесса в соответствии с заданием,
- ручной ввод частоты вращения электродвигателя.

4.14.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------|------------|--------------------|------------------------------|
| Input | supply_di | BOOL | TRUE | схема собрана/питание подано |
| | ready_di | BOOL | TRUE | готовность |
| | plc_control_di | BOOL | TRUE | автоматическое управление |
| | fault_di | BOOL | FALSE | неисправность |
| | pv | udt_analog | | переменная процесса |
| | ac | udt_analog | | ток |
| | freq | udt_analog | | частота |
| | on_di | BOOL | | включен |
| | warning_di | BOOL | | сигнал предупреждения |
| Inout | hmi | udt_fr | | интерфейс блока |
| Output | start_do | BOOL | | включить |
| | stop_do | BOOL | | выключить |
| | alm_reset_do | BOOL | | сброс аварии |
| | ao | REAL | | задание частоты |

4.14.4 Принцип работы

При наличии питания «supply_di» блок перейдет в состояние «ожидание готовности». Происходит проверка наличия сигнала «ready_di» и отсутствие «fault_di» и переход в «остановлен». По команде «пуск» сменится состояние на «ожидание включения», подается выход «start_do».

Для перехода в «запуск» необходим сигнал «on_di». В состоянии «запуск» происходит набор минимальной частоты и переход блока в работу.

Команда «стоп» меняет состояние на «останов». Частота вращения понижается до минимальной, затем снимается выход «start_do» и подается «stop_do».

По отсутствию сигнала «on_di», блок будет остановлен.

4.14.5 Интерфейс «udt_fr»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|---------------|-------------|------------|---|
| std | udt_standard | | | стандартный набор переменных |
| pid | udt_pid | | | набор параметров регулятора |
| in_ac_settings | udt_engine_ac | | | настройки токов для передачи от двигателя |
| out_ac | REAL | | вывод | ток, А. |
| freq_low | REAL | 5 | ввод | минимальная частота вращения двигателя, Гц |
| freq_hi | REAL | 50 | ввод | максимальная частота вращения двигателя, Гц |
| freq_rising | REAL | 5 | ввод | скорость нарастания частоты, Гц/сек |
| freq_lowering | REAL | 5 | ввод | скорость понижения частоты, Гц/сек |
| freq_overload_lowering | REAL | 5 | ввод | скорость понижения частоты, Гц/сек, при перегрузке по току |
| freq_deadband | REAL | 1 | ввод | допустимое отклонение частоты при разгоне и торможении |
| time_on | REAL | 1 | ввод | время включения. Максимально-допустимое время ожидания (в секундах) прихода сигнала "on_di", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_off | REAL | 20 | ввод | время отключения. Максимально-допустимое время ожидания (в секундах) снятия сигнала "on_di", по истечении которого объект переходит в аварийное состояние, если сигнал не ушел. |
| time_reset | REAL | 1 | ввод | длительность импульса сигнала сброса аварии «alm_reset_do» |
| time_freq_min_wait | REAL | 10 | ввод | время ожидания набора минимальной частоты, сек |
| time_pulse | REAL | 1 | ввод | длительность импульса включения/выключения (сек) при импульсной подаче сигналов «start_do» и «stop_do» |
| out_freq | REAL | | ввод | частота, Гц |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |
| min_freq_run | REAL | 2 | ввод | частота, больше которой считается, что ЧРП запущен, Гц |
| time_supply | REAL | 10 | ввод | время ожидания возврата питания, сек |
| time_restart | REAL | 10 | ввод | время ожидания готовности после возврата питания, сек |

4.14.6 Переменные интерфейса блока. Списки значений

4.14.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_fr» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-----------------|
| stop | WORD | 16#1 | Стоп |
| start | WORD | 16#2 | Пуск |
| stop_fast | WORD | 16#4 | Быстрый останов |

| Название | Тип | Значение | Описание |
|----------|------|----------|---|
| man_up | WORD | 16#8 | При включенном ручном управлении выходом регулятора, увеличивает задание управляющего воздействия на величину «pid.in_step» |
| man_down | WORD | 16#10 | При включенном ручном управлении выходом регулятора, уменьшает задание управляющего воздействия на величину «pid.in_step» |

4.14.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_fr» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|------------------|------|----------|--|
| start_timeout | WORD | 16#1 | «Превышено время включения». Сигнал "on_di" не пришел в течение времени «time_on» после подачи выхода «start_do». |
| stop_timeout | WORD | 16#2 | «Превышено время отключения». Сигнал "on_di" не ушел в течение времени «time_off» после снятия выхода «start_do» и подачи «stop_do». |
| unexpected_start | WORD | 16#4 | «Самопроизвольное включение». Поступление сигнала «on_di» без подачи команды «пуск». |
| unexpected_stop | WORD | 16#8 | «Самопроизвольное отключение». Пропадание сигнала «on_di» без подачи команды «стоп». |
| fault | WORD | 16#10 | «Авария». Поступление сигнала на вход «fault_di». Данную аварию можно сбросить только после снятия сигнала. |
| power_off | WORD | 16#20 | «Пропадание питания». Снятие входного сигнала «supply_di» во время работы. |
| no_load | WORD | 16#40 | Минимальный ток при максимальной частоте |
| min_freq | WORD | 16#80 | Превышено время ожидания набора минимальной частоты |
| ready_timeout | WORD | 16#100 | Превышено время ожидания готовности после возврата питания |

4.14.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_fr» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|---|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |
| pv_bad | WORD | 16#2 | «Недостовверное значение регулируемого параметра» Состояние объекта (out_state_actual) - неисправность |
| freq_bad | WORD | 16#4 | «Недостовверное значение датчика частоты» Состояние объекта (out_state_actual) - неисправность |
| not_ready | WORD | 16#8 | «Нет готовности». Отсутствие входного сигнала «ready_di». |
| fault | WORD | 16#10 | «Авария» не используется |
| man_on | WORD | 16#20 | Ручное задание частоты |
| ac_max | WORD | 16#40 | Достигнут максимальный ток двигателя |
| warning | WORD | 16#80 | Наличие сигнала предупреждения «warning_di» |

4.14.6.4 Конфигурация «std.in_cfg»

Тип глобальных констант - «sys_db_cfg_fr» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|------------------|------|----------|---|
| inverse | WORD | 16#1 | При включении данного флага задание SP и значение процесса PV регулятора меняются местами, т.е. рассогласование вычисляется как $dX = PV - SP$ (вместо обычного $dX = SP - PV$). При этом имеет место увеличение выходного значения регулятора при увеличении переменной процесса (регулируемого параметра). |
| control_ac | WORD | 16#2 | При включенном флаге задействуется контроль параметров двигателя, указанных в «in_ac_settings». |
| freq_sensor | WORD | 16#4 | При включенном флаге производится контроль значения частоты «freq». Если задание частоты не совпадает с текущим значением в пределах погрешности, заданной «freq_deadband», производится останов ЧРП. Контроль производится только в процессе разгона и торможения. |
| control_run_freq | WORD | 16#8 | Позволяет в качестве сигнала «on_di» использовать показания датчика частоты «freq». Если показания датчика частоты после пуска достигают минимального значения частоты «min_freq_run», считается, что ЧРП в работе. |
| pulse_signal | WORD | 16#10 | Выходные сигналы «start_do» и «stop_do» при включении этого флага подаются кратковременно (в виде импульса) при подаче соответствующих команд, а не удерживаются постоянно. |
| auto_restart | WORD | 16#20 | Возобновление работы после возврата питания «supply_di» |

4.14.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_fr» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|---------------|------|----------|---|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Остановлен |
| started | WORD | 16#4 | В работе |
| alarm | WORD | 16#8 | Авария |
| starting | WORD | 16#10 | Запуск |
| stopping | WORD | 16#20 | Останов |
| regulation_l | WORD | 16#40 | Регулирование, мин. частота Во время работы, задание частоты от регулятора \leq «freq_low» |
| regulation_h | WORD | 16#80 | Регулирование, макс. частота Во время работы, задание частоты от регулятора \geq «freq_hi» |
| starting_wait | WORD | 16#100 | Ожидание включения |
| ready_waiting | WORD | 16#200 | Ожидание готовности |
| fr_no_supply | WORD | 16#400 | Ожидание возврата питания |

4.14.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_fr» п. 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |
| started | WORD | 16#4 | Включен |

4.14.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
  // объявление интерфейса блока fb_fr для ПЧ
  dbs_fr_1: RuDrive.udt_fr;
  // объявление интерфейса блока fb_ai для переменной процесса ПЧ
  dbs_fr_1_pv: RuDrive.udt_analog;
  // объявление интерфейса блока fb_ai для сигнала тока ПЧ
  dbs_fr_1_ac: RuDrive.udt_analog;
  // объявление интерфейса блока fb_ai для сигнала частоты
  dbs_fr_1_freq: RuDrive.udt_analog;
  // объявление интерфейса блока fb_ao для задания частоты на ПЧ
  dbs_fr_1_ao: RuDrive.udt_analog;

END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
  // объявление экземпляра блока fb_fr для ПЧ
  dbi_fr_1 : RuDrive.fb_fr;
  // объявление экземпляра блока fb_ai для переменной процесса ПЧ
  dbi_fr_1_pv : RuDrive.fb_ai;
  // объявление экземпляра блока fb_ai для сигнала тока ПЧ
  dbi_fr_1_ac : RuDrive.fb_ai;
  // объявление экземпляра блока fb_ai для сигнала частоты
  dbi_fr_1_freq : RuDrive.fb_ai;
  // объявление экземпляра блока fb_ao для задания частоты на ПЧ
  dbi_fr_1_ao : RuDrive.fb_ao;

END_VAR

// вызов экземпляра блока fb_ai для для сигнала тока ПЧ
dbi_fr_1_ac(fault_di:= FALSE, in_real:= in_real_ac, hmi:= dbs_fr_1_ac);
// вызов экземпляра блока fb_ai для сигнала частоты
dbi_fr_1_freq(
  fault_di:= FALSE,
  in_real:= in_real_freq,
  hmi:= dbs_fr_1_freq);
// вызов экземпляра блока fb_ai для переменной процесса ПЧ
dbi_fr_1_pv(fault_di:= FALSE, in_real:= in_real_pv, hmi:= dbs_fr_1_pv);

// вызов экземпляра блока fb_fr для ПЧ
dbi_fr_1(
  supply_di:= input_supply_di,
  ready_di:= input_ready_di,
  plc_control_di:= input_plc_control_di,
  fault_di:= input_fault_di,
  pv:= dbs_fr_1_pv,
  ac:= dbi_fr_1_ac,
  freq:= dbi_fr_1_freq,
  on_di:= input_on_di,
  warning_di:= input_warning_di,
  hmi:= dbs_fr_1,
  start_do=> output_start_do,
  stop_do=> output_stop_do,
  alm_reset_do=> output_alm_reset_do,
  ao=> dbi_fr_1_ao.in_real);

// вызов экземпляра блока fb_ao для задания частоты на ПЧ
dbi_fr_1_ao (hmi:= dbs_fr_1_ao, out_int=> out_int_value);
```

4.15 Мотор (fb_motor)

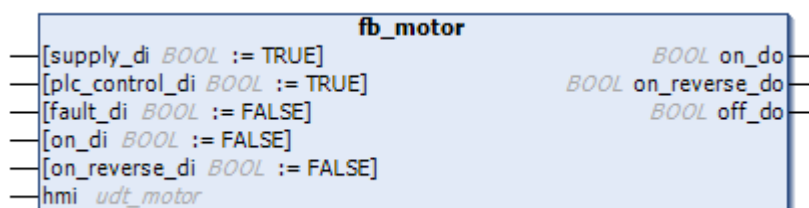
4.15.1 Назначение

Включение и выключение электродвигателя, изменение направления вращения.

4.15.2 Функции

- запуск и останов электродвигателя,
- выбор направления вращения.

4.15.3 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 |
| | fault_di | BOOL | FALSE | неисправность |
| | on_di | BOOL | FALSE | включен в прямом направлении |
| | on_reverse_di | BOOL | FALSE | включен в обратном направлении |
| Inout | hmi | udt_motor | | интерфейс блока |
| Output | on_do | BOOL | | включить в прямом направлении |
| | on_reverse_do | BOOL | | включить в обратном направлении |
| | off_do | BOOL | | отключить |

4.15.4 Принцип работы

Блок переходит в состояние «остановлен» из состояния «нет питания», когда есть сигнал «supply_di».

По команде «пуск» блок из состояния «остановлен» перейдет в «включается». На выход «on_do» подается сигнал. При поступлении сигнала «on_di» блок перейдет в состояние «включен». Либо по команде «Пуск в обратном направлении» блок из состояния «остановлен» перейдет в «включается в обратном направлении». На выход «on_reverse_do» подается сигнал. При поступлении сигнала «on_reverse_di» блок перейдет в состояние «включен в обратном направлении».

По команде «стоп» блок перейдет в состояние «отключается» либо «отключается в обратном направлении». С выхода «on_do» либо «on_reverse_do» снимается сигнал, на выход «off_do» - подается. При снятии сигнала «on_di» либо «on_reverse_di» блок перейдет в состояние «остановлен».

4.15.5 Интерфейс «udt_motor»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|----------|--------------|-------------|------------|---|
| std | udt_standard | | | стандартный набор переменных |
| time_on | REAL | 2 | ввод | время включения. Максимально-допустимое время ожидания (в секундах) прихода сигнала "on_di" или «on_reverse_di», по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_off | REAL | 2 | ввод | время отключения. Максимально-допустимое время ожидания (в секундах) снятия сигнала "on_di" или «on_reverse_di», по истечении которого объект переходит в аварийное состояние, если сигнал не ушел. |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |

4.15.6 Переменные интерфейса блока. Списки значений

4.15.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_motor» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|---------------|------|----------|-----------------------------|
| stop | WORD | 16#1 | Стоп |
| start | WORD | 16#2 | Пуск |
| start_reverse | WORD | 16#4 | Пуск в обратном направлении |

4.15.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_motor» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|--------------------------|------|----------|--|
| start_timeout | WORD | 16#1 | «Превышено время включения». Сигнал "on_di" не пришел в течение времени «time_on» после подачи команды «Пуск» |
| stop_timeout | WORD | 16#2 | «Превышено время отключения». Сигнал "on_di" не ушел в течение времени «time_off» после подачи команды «Стоп». |
| unexpected_start | WORD | 16#4 | «Самопроизвольное включение». Поступление сигнала "on_di" без подачи команды «Пуск» |
| unexpected_stop | WORD | 16#8 | «Самопроизвольное отключение». Снятие сигнала "on_di" без подачи команды «Стоп» |
| fault | WORD | 16#10 | «Авария». Поступление сигнала на вход «fault_di». Данную аварию можно сбросить только после снятия сигнала. |
| power_off | WORD | 16#20 | «Пропадание питания». Снятие входного сигнала «supply_di» во время работы. |
| start_timeout_reverse | WORD | 16#40 | «Превышено время включения в обратном направлении». Сигнал «on_revers_di» не пришел в течение времени «time_on» после подачи команды «Пуск в обратном направлении» |
| stop_timeout_reverse | WORD | 16#80 | «Превышено время отключения в обратном направлении». Сигнал «on_revers_di» не ушел в течение времени «time_off» после подачи команды «Стоп». |
| unexpected_start_reverse | WORD | 16#100 | «Самопроизвольное включение в обратном направлении». Поступление сигнала «on_revers_di» без подачи команды «Пуск в обратном направлении» |
| unexpected_stop_reverse | WORD | 16#200 | «Самопроизвольное отключение в обратном направлении». Снятие сигнала «on_revers_di» без подачи команды «Стоп» |

| Название | Тип | Значение | Текс сообщений/Описание |
|-------------------------|------|----------|---|
| start_and_start_reverse | WORD | 16#400 | «Одновременное включение в прямом и обратном направлении» Поступление сигналов "on_di" и «on_revers_di» одновременно |

4.15.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_motor» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|--|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |

4.15.6.4 Конфигурация «std.in_config»

Данный блок не имеет конфигураций.

4.15.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_motor» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|------------------|------|----------|------------------------------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |
| started | WORD | 16#4 | Включен |
| stopping | WORD | 16#8 | Отключается |
| starting | WORD | 16#10 | Включается |
| alarm | WORD | 16#20 | Авария |
| started_reverse | WORD | 16#40 | Включен в обратном направлении |
| stopping_reverse | WORD | 16#80 | Отключается в обратном направлении |
| starting_reverse | WORD | 16#100 | Включается в обратном направлении |

4.15.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_motor» п. 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------------|------|----------|--------------------------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |
| started | WORD | 16#4 | Включен |
| started_reverse | WORD | 16#40 | Включен в обратном направлении |

4.15.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_motor для электропривода с реверсным вращением
    dbs_motor_1: RuDrive.udt_motor;
```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_motor для электропривода с реверсным вращением
    dbi_motor_1: RuDrive.fb_motor;
END_VAR

// вызов экземпляра блока fb_motor для электропривода с реверсным вращением
dbi_motor_1(
    supply_di:= input_supply_di,
    plc_control_di:= input_plc_control_di,
    fault_di:= input_fault_di,
    on_di:= input_on_di,
    on_reverse_di:= input_on_reverse_di,
    hmi:= dbs_motor_1,
    on_do=> output_on_do,
    on_reverse_do=> output_on_reverse_do,
    off_do=> output_off_do);
```

4.16 Клапан с аналоговым управлением (fb_rk_cont)

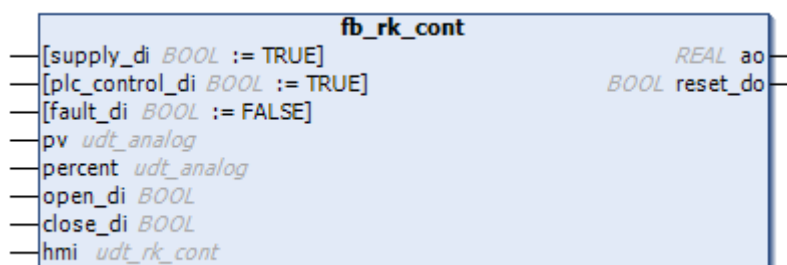
4.16.1 Назначение

Поддержание переменной процесса путем открытия-закрытия клапана.

4.16.2 Функции

- поддержание переменной процесса в соответствии с заданием,
- открытие клапана на определенный процент.

4.16.3 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 |
| | fault_di | BOOL | FALSE | неисправность |
| | pv | udt_analog | | переменная процесса |
| | percent | udt_analog | | процент открытия клапана |
| | open_di | BOOL | | концевик открытого положения |
| | close_di | BOOL | | концевик закрытого положения |
| Inout | hmi | udt_rk_cont | | интерфейс блока |
| Output | ao | REAL | | выход регулятора на управление клапаном |
| | reset_do | BOOL | | сброс аварии |

4.16.4 Принцип работы

Блок по наличию питания «supply_di» в зависимости от сигналов «open_di» и «close_di» из состояния «нет питания» переходит в одно из «открыт», «закрыт» или «промежуточное».

Команда «pid» переведет блок в состояние регулирования. Клапан своим положением будет поддерживать переменную процесса согласно уставке.

По команде «стоп» в зависимости от сигналов «open_di» и «close_di» из состояния «регулирование» блок переходит в одно из «открыт», «закрыт» или «промежуточное».

4.16.5 Интерфейс «udt_rk_cont»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|--------------------|--------------|-------------|------------|---|
| std | udt_standard | | | стандартный набор переменных |
| pid | udt_pid | | | набор параметров регулятора |
| out_percent | REAL | | вывод | положение клапана, % |
| percent_min | REAL | 0 | ввод | минимальный процент открытия |
| percent_max | REAL | 100 | ввод | максимальный процент открытия |
| position_diff | REAL | 1 | ввод | допустимое рассогласование заданной и текущей позиции |
| time_motion | REAL | 60 | ввод | время хода клапана, сек время, за которое клапан переходит из одного крайнего положения в другое |
| time_leave | REAL | 20 | ввод | время схода с концевого выключателя, сек |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |
| delta_limit_switch | REAL | 1 | ввод | дельта погрешности состояний открытия и закрытия по указателю положения |
| time_reset | REAL | 2 | ввод | время сброса аварии, сек |

4.16.6 Переменные интерфейса блока. Списки значений

4.16.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_rk_cont» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|---|
| stop | WORD | 16#1 | Стоп |
| close | WORD | 16#2 | Закрыть |
| open | WORD | 16#4 | Открыть |
| pid | WORD | 16#8 | Регулирование |
| man_up | WORD | 16#10 | При включенном ручном управлении выходом регулятора, увеличивает задание управляющего воздействия на величину «pid.in_step» |
| man_down | WORD | 16#20 | При включенном ручном управлении выходом регулятора, уменьшает задание управляющего воздействия на величину «pid.in_step» |

4.16.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_rk_cont» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текст сообщений/Описание |
|------------------|------|----------|---|
| sw_open_timeout | WORD | 16#1 | «Превышено время схода с КО». Сигнал "open_di" не ушел в течение времени «time_leave» после подачи команды «Закрыть» |
| sw_close_timeout | WORD | 16#2 | «Превышено время схода с КЗ». Сигнал "close_di" не ушел в течение времени «time_leave» после подачи команды «Открыть» |
| open_timeout | WORD | 16#4 | «Превышено время открытия». После подачи команды «Открыть» в течении времени «time_motion» не пришел сигнал "open_di". |
| close_timeout | WORD | 16#8 | «Превышено время закрытия». После подачи команды «Закрыть» в течении времени «time_motion» не пришел сигнал "close_di". |
| unexpected_open | WORD | 16#10 | «Самопроизвольное срабатывание КО». Во время закрытия клапана пришел сигнал «open_di» |

| Название | Тип | Значение | Текс сообщений/Описание |
|------------------|------|----------|---|
| unexpected_close | WORD | 16#20 | «Самопроизвольное срабатывание КЗ». Во время открытия клапана пришел сигнал «close_di» |
| open_close | WORD | 16#40 | «Неопределенное состояние». Одновременное поступление сигналов «open_di» и «close_di». |
| fault | WORD | 16#80 | «Авария». Поступление сигнала на вход «fault_di». Данную аварию можно сбросить только после снятия сигнала. |
| jamming | WORD | 16#100 | «Заклинивание». значение сигнала положения «percent» не соответствует значению выхода регулятора в пределах допуска |

4.16.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_rk_cont» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|---|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |
| pv_bad | WORD | 16#2 | «Недостовверное значение регулируемого параметра» Состояние объекта (out_state_actual) - неисправность |

4.16.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_rk_cont» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-------------------------|------|----------|---|
| inverse | WORD | 16#1 | При включении данного флага задание SP и значение процесса PV регулятора меняются местами, т.е. рассогласование вычисляется как $dX = PV - SP$ (вместо обычного $dX = SP - PV$). При этом имеет место увеличение выходного значения регулятора при увеличении переменной процесса (регулируемого параметра). |
| pos_sensor | WORD | 16#2 | Наличие указателя положения |
| block_open_close | WORD | 16#4 | Ограничить положение (мин./макс.) Открытие-закрытие ограничивается «percent_max» и «percent_min» |
| limit_switch_open_close | WORD | 16#8 | Определение открытия/закрытия по концевикам |
| pos_sensor_open_close | WORD | 16#10 | Определение открытия/закрытия по указателю положения |

4.16.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_rk_cont» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|--|
| power_off | WORD | 16#1 | Нет питания |
| closed | WORD | 16#2 | Закрит |
| opened | WORD | 16#4 | Открыт |
| intermediate | WORD | 16#8 | Промежуточное |
| alarm | WORD | 16#10 | Авария |
| regulation | WORD | 16#20 | Регулирование |
| regulation_l | WORD | 16#40 | «Регулирование, закрыт» Во время регулирования пришел сигнал «close_di» |

| Название | Тип | Значение | Описание |
|--------------|------|----------|---|
| regulation_h | WORD | 16#80 | «Регулирование, открыт» Во время регулирования пришел сигнал «open_di» |

4.16.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_rk_cont» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|---------------|
| power_off | WORD | 16#1 | Нет питания |
| closed | WORD | 16#2 | Закрыт |
| opened | WORD | 16#4 | Открыт |
| intermediate | WORD | 16#8 | Промежуточное |

4.16.7 Пример

Объявление глобальных переменных:

VAR_GLOBAL

```
// объявление интерфейса блока fb_rk_cont для клапана с аналоговым управлением
dbs_rk_cont_1: RuDrive.udt_rk_cont;
// объявление интерфейса блока fb_ai для переменной процесса клапана
dbs_rk_cont_1_pv: RuDrive.udt_analog;
// объявление интерфейса блока fb_ai для сигнала положения клапана
dbs_rk_cont_1_percent: RuDrive.udt_analog;
// объявление интерфейса блока fb_ao для задания положения клапана
dbs_rk_cont_1_ao: RuDrive.udt_analog;
```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM PLC_PRG

VAR

```
// объявление экземпляра блока fb_rk_cont для клапана с аналоговым управлением
dbi_rk_cont_1: RuDrive.fb_rk_cont;
// объявление экземпляра блока fb_ai для переменной процесса клапана
dbi_rk_cont_1_pv: RuDrive.fb_ai;
// объявление экземпляра блока fb_ai для сигнала положения клапана
dbi_rk_cont_1_percent: RuDrive.fb_ai;
// объявление экземпляра блока fb_ao для задания положения клапана
dbi_rk_cont_1_ao: RuDrive.fb_ao;
```

END_VAR

```
// вызов экземпляра блока fb_ai для сигнала положения клапана
```

```
dbi_rk_cont_1_percent(
    fault_di:= FALSE,
    in_real:= in_real_percent,
    hmi:= dbs_rk_cont_1_percent);
// вызов экземпляра блока fb_ai для переменной процесса клапана
dbi_rk_cont_1_pv(fault_di:= FALSE, in_real:= in_real_pv, hmi:= dbs_rk_cont_1_pv);
```

```
// вызов экземпляра блока fb_rk_cont для клапана с аналоговым управлением
```

```
dbi_rk_cont_1(
    supply_di:= input_supply_di,
    plc_control_di:= input_plc_control_di,
    fault_di:= input_fault_di,
    pv:= dbs_rk_cont_1_pv,
    percent:= dbs_rk_cont_1_percent,
    open_di:= input_open_di,
    close_di:= input_close_di,
    hmi:= dbs_rk_cont_1,
    ao=> dbi_rk_cont_1_ao.in_real,
```

```
reset_do=> output_reset_do);  
  
// вызов экземпляра блока fb_ao для задания положения клапана  
dbi_rk_cont_1_ao(hmi:= dbs_rk_cont_1_ao, out_int=> out_int_value);
```

4.17 Клапан с дискретным управлением (fb_rk_step)

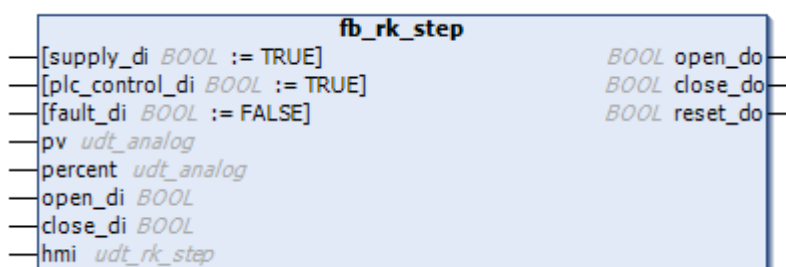
4.17.1 Назначение

Поддержание переменной процесса путем открытия-закрытия клапана.

4.17.2 Функции

- поддержание переменной процесса в соответствии с заданием,
- открытие клапана на определенный процент.

4.17.3 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 |
| | fault_di | BOOL | FALSE | неисправность |
| | pv | udt_analog | | переменная процесса |
| | percent | udt_analog | | процент открытия клапана |
| | open_di | BOOL | | концевик открытого положения |
| | close_di | BOOL | | концевик закрытого положения |
| Inout | hmi | udt_rk_step | | интерфейс блока |
| Output | open_do | BOOL | | открывать |
| | close_do | BOOL | | закрывать |
| | reset_do | BOOL | | сброс аварии |

4.17.4 Принцип работы

Блок по наличию питания «supply_di» в зависимости от сигналов «open_di» и «close_di» из состояния «нет питания» переходит в одно из «открыт», «закрыт» или «промежуточное».

Команда «pid» переведет блок в состояние регулирования. Клапан своим положением будет поддерживать переменную процесса согласно уставке.

По команде «стоп» в зависимости от сигналов «open_di» и «close_di» из состояния «регулирование» блок переходит в одно из «открыт», «закрыт» или «промежуточное».

4.17.5 Интерфейс «udt_rk_step»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|--------------------|--------------|-------------|------------|---|
| std | udt_standard | | | стандартный набор переменных |
| pid | udt_pid | | | набор параметров регулятора |
| out_percent | REAL | | вывод | положение клапана, % |
| percent_min | REAL | 0 | ввод | минимальный процент открытия |
| percent_max | REAL | 100 | ввод | максимальный процент открытия |
| time_motion | REAL | 60 | ввод | время хода клапана, сек время, за которое клапан переходит из одного крайнего положения в другое |
| time_leave | REAL | 20 | ввод | время схода с концевого выключателя, сек |
| time_pause | REAL | 2 | ввод | время паузы между импульсами позиционирования, сек во время ручного ввода положения |
| time_pulse_min | REAL | 3 | ввод | минимальное время импульса регулятора, сек во время регулирования |
| time_break_min | REAL | 3 | ввод | минимальное время паузы между импульсами регулятора, сек во время регулирования |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |
| delta_limit_switch | REAL | 1 | ввод | дельта погрешности состояний открытия и закрытия по указателю положения |
| time_reset | REAL | 2 | ввод | время сброса аварии |

4.17.6 Переменные интерфейса блока. Списки значений

4.17.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_rk_step» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|---|
| stop | WORD | 16#1 | Стоп |
| close | WORD | 16#2 | Закрыть |
| open | WORD | 16#4 | Открыть |
| pid | WORD | 16#8 | Регулирование |
| man_up | WORD | 16#10 | При включенном ручном управлении выходом регулятора, увеличивает задание управляющего воздействия на величину «pid.in_step» |
| man_down | WORD | 16#20 | При включенном ручном управлении выходом регулятора, уменьшает задание управляющего воздействия на величину «pid.in_step» |

4.17.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_rk_step» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|------------------|------|----------|---|
| sw_open_timeout | WORD | 16#1 | «Превышено время схода с КО». Сигнал "open_di" не ушел в течение времени «time_leave» после подачи команды «Закрыть» |
| sw_close_timeout | WORD | 16#2 | «Превышено время схода с КЗ». Сигнал "close_di" не ушел в течение времени «time_leave» после подачи команды «Открыть» |
| open_timeout | WORD | 16#4 | «Превышено время открытия». |

| Название | Тип | Значение | Текс сообщений/Описание |
|------------------|------|----------|---|
| | | | После подачи команды «Открыть» в течении времени «time_motion» не пришел сигнал "open_di". |
| close_timeout | WORD | 16#8 | «Превышено время закрытия». После подачи команды «Закрыть» в течении времени «time_motion» не пришел сигнал "close_di". |
| unexpected_open | WORD | 16#10 | «Самопроизвольное срабатывание КО». Во время закрытия клапана пришел сигнал «open_di» |
| unexpected_close | WORD | 16#20 | «Самопроизвольное срабатывание КЗ». Во время открытия клапана пришел сигнал «close_di» |
| open_close | WORD | 16#40 | «Неопределенное состояние». Одновременное поступление сигналов «open_di» и «close_di». |
| fault | WORD | 16#80 | «Авария». Поступление сигнала на вход «fault_di». Данную аварию можно сбросить только после снятия сигнала. |

4.17.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_rk_step» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|---|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |
| pv_bad | WORD | 16#2 | «Недостовверное значение регулируемого параметра» Состояние объекта (out_state_actual) - неисправность |

4.17.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_rk_step» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-------------------------|------|----------|---|
| inverse | WORD | 16#1 | При включении данного флага задание SP и значение процесса PV регулятора меняются местами, т.е. рассогласование вычисляется как $dX = PV - SP$ (вместо обычного $dX = SP - PV$). При этом имеет место увеличение выходного значения регулятора при увеличении переменной процесса (регулируемого параметра). |
| pos_sensor | WORD | 16#2 | Наличие указателя положения |
| block_open_close | WORD | 16#4 | Ограничить положение (мин./макс.) Открытие-закрытие ограничивается «percent_max» и «percent_min» |
| limit_switch_open_close | WORD | 16#8 | Определение открытия/закрытия по концевикам |
| pos_sensor_open_close | WORD | 16#10 | Определение открытия/закрытия по указателю положения |

4.17.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_rk_step» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|---------------|
| power_off | WORD | 16#1 | Нет питания |
| closed | WORD | 16#2 | Закрыт |
| opened | WORD | 16#4 | Открыт |
| intermediate | WORD | 16#8 | Промежуточное |

| Название | Тип | Значение | Описание |
|--------------|------|----------|--|
| alarm | WORD | 16#10 | Авария |
| regulation | WORD | 16#20 | Регулирование |
| regulation_l | WORD | 16#40 | «Регулирование, закрыт» Во время регулирования пришел сигнал «close_di» |
| regulation_h | WORD | 16#80 | «Регулирование, открыт» Во время регулирования пришел сигнал «open_di» |
| opening | WORD | 16#100 | «Открывается» При наличии выхода «open_do» |
| closing | WORD | 16#200 | «Закрывается» При наличии выхода «close_do» |

4.17.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_rk_step» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|---------------|
| power_off | WORD | 16#1 | Нет питания |
| closed | WORD | 16#2 | Закрыт |
| opened | WORD | 16#4 | Открыт |
| intermediate | WORD | 16#8 | Промежуточное |

4.17.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_rk_step для клапана с дискретным управлением
    dbs_rk_step_1: RuDrive.udt_rk_step;
    // объявление интерфейса блока fb_ai для переменной процесса клапана
    dbs_rk_step_1_pv: RuDrive.udt_analog;
    // объявление интерфейса блока fb_ai для сигнала положения клапана
    dbs_rk_step_1_percent: RuDrive.udt_analog;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_rk_step для клапана с дискретным управлением
    dbi_rk_step_1: RuDrive.fb_rk_step;
    // объявление экземпляра блока fb_ai для переменной процесса клапана
    dbi_rk_step_1_pv: RuDrive.fb_ai;
    // объявление экземпляра блока fb_ai для сигнала положения клапана
    dbi_rk_step_1_percent: RuDrive.fb_ai;
END_VAR

// вызов экземпляра блока fb_ai для сигнала положения клапана
dbi_rk_step_1_percent(
    fault_di:= FALSE,
    in_real:= in_real_percent,
    hmi:= dbs_rk_step_1_percent);
// вызов экземпляра блока fb_ai для переменной процесса клапана
dbi_rk_step_1_pv(fault_di:= FALSE, in_real:= in_real_pv, hmi:= dbs_rk_step_1_pv);

// вызов экземпляра блока fb_rk_step для клапана с дискретным управлением
dbi_rk_step_1(
    supply_di:= input_supply_di,
    plc_control_di:= input_plc_control_di,
    fault_di:= input_fault_di,
    pv:= dbs_rk_step_1_pv,
```

```
percent:= dbs_rk_step_1_percent,  
open_di:= input_open_di,  
close_di:= input_close_di,  
hmi:= dbs_rk_step_1,  
open_do=> output_open_do,  
close_do=> output_close_do,  
reset_do=> output_reset_do);
```

4.18 Ячейка (fb_switch_gear)

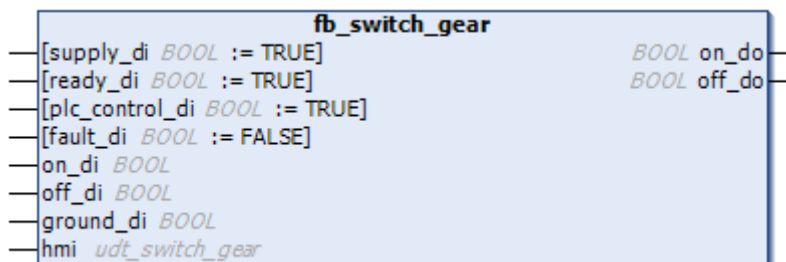
4.18.1 Назначение

Управление коммутацией электрической ячейки.

4.18.2 Функции

- коммутация электрической ячейки.

4.18.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|-----------|----------------|-----------------|--------------------|------------------------------|
| Input | supply_di | BOOL | TRUE | схема собрана/питание подано |
| | ready_di | BOOL | TRUE | готовность |
| | plc_control_di | BOOL | TRUE | автоматическое управление |
| | fault_di | BOOL | FALSE | неисправность |
| | on_di | BOOL | | включена |
| | off_di | BOOL | | отключена |
| ground_di | BOOL | | заземлена | |
| Inout | hmi | udt_switch_gear | | интерфейс блока |
| Output | on_do | BOOL | | включить |
| | off_do | BOOL | | отключить |

4.18.4 Принцип работы

Блок переходит в состояние «ожидание готовности» из состояния «нет питания», когда есть сигнал «supply_di».

Проверяется наличие готовности «ready_di», отсутствие неисправности «fault_di», сигнала отключена «off_di» и переход в «отключена».

Команда «включить» запускает процедуру включения, импульсом «time_pulse» подается выход «on_do». Ожидается поступление «on_do».

По команде «stop_delay» процедура отключения начнется с задержкой «time_off_delay», «stop» - незамедлительно. Блок перейдет в состояние «отключается». Ожидается снятие сигнала «on_di» и поступление «off_di» для перехода в «отключена».

Команда «reset_alarm» в состоянии «авария» по мимо сброс внутренних аварий блока импульсом подает выход «off_do».

4.18.5 Интерфейс «udt_switch_gear»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|----------------|--------------|-------------|------------|---|
| std | udt_standard | | | стандартный набор переменных |
| time_on | REAL | 2 | ввод | время включения. Максимально-допустимое время ожидания (в секундах) прихода сигнала "on_di", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_off | REAL | 2 | ввод | время отключения. Максимально-допустимое время ожидания (в секундах) прихода сигнала "off_di", по истечении которого объект переходит в аварийное состояние, если сигнал не ушел. |
| time_pulse | REAL | 2 | ввод | время импульса включения/отключения, сек |
| time_off_delay | REAL | 120 | ввод | задержка отключения ячейки, сек |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |

4.18.6 Переменные интерфейса блока. Списки значений

4.18.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_switch_gear» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|------------|------|----------|-----------------------|
| stop | WORD | 16#1 | Отключить |
| start | WORD | 16#2 | Включить |
| stop_delay | WORD | 16#4 | Отключить с задержкой |

4.18.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_switch_gear» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|------------------|------|----------|--|
| start_timeout | WORD | 16#1 | «Превышено время включения». Сигнал "on_di" не пришел в течение времени «time_on» после подачи команды «Включить» |
| stop_timeout | WORD | 16#2 | «Превышено время отключения». Сигнал "on_di" не ушел или «off_di» не пришел в течение времени «time_off» после подачи команды «Отключить». |
| unexpected_start | WORD | 16#4 | «Самопроизвольное включение». Поступление сигнала "on_di" без подачи команды «Включить» |
| unexpected_stop | WORD | 16#8 | «Самопроизвольное отключение». Поступление сигнала "off_di" без подачи команды «Отключить» |
| fault | WORD | 16#10 | «Авария». Поступление сигнала на вход «fault_di». Данную аварию можно сбросить только после снятия сигнала. |
| ground | WORD | 16#20 | «Включено заземление». Поступление сигнала «ground_di» в состояниях «включена», «отключается» и «включается» |

4.18.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_switch_gear» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|---|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |
| not_ready | WORD | 16#2 | «Нет готовности» Отсутствие входного сигнала «ready_di». |
| no_start | WORD | 16#4 | «Нет сигнала включена». В состоянии «включена» пропал сигнал «on_di» |
| no_stop | WORD | 16#8 | «Нет сигнала отключена» В состоянии «отключена» или «нет питания» ушел сигнал «off_di» |
| fault | WORD | 16#10 | «Неисправность устройства защиты» По сигналу «fault_di» |
| ground | WORD | 16#20 | «Включено заземление» Поступление сигнала «ground_di» |

4.18.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_switch_gear» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|---|
| fault_alarm | WORD | 16#1 | «Разрешить отключение по сигналу авария» Флаг разрешает переход блока в состояние аварии с подачей сигнала на отключение ячейки в случае, если придет сигнал «авария ячейки» во время работы. |
| ground_alarm | WORD | 16#2 | «Разрешить отключение по сигналу заземление» Флаг разрешает переход блока в состояние аварии с подачей сигнала на отключение ячейки в случае, если придет сигнал «ячейка заземлена» во время работы. |

4.18.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_switch_gear» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|---------------|------|----------|---------------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключена |
| started | WORD | 16#4 | Включена |
| stopping | WORD | 16#8 | Отключается |
| starting | WORD | 16#10 | Включается |
| alarm | WORD | 16#20 | Авария |
| ready_waiting | WORD | 16#40 | Ожидание готовности |
| grounded | WORD | 16#80 | Заземлено |

4.18.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_motor» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключена |
| started | WORD | 16#4 | Включена |

4.18.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_switch_gear для электрической ячейки
    dbs_switch_gear_1: RuDrive.udt_switch_gear;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_switch_gear для электрической ячейки
    dbi_switch_gear_1: RuDrive.fb_switch_gear;
END_VAR

// вызов экземпляра блока fb_switch_gear для электрической ячейки
dbi_switch_gear_1(
    supply_di:= input_supply_di,
    ready_di:= input_ready_di,
    plc_control_di:= input_plc_control_di,
    fault_di:= input_fault_di,
    on_di:= input_on_di,
    off_di:= input_off_di,
    ground_di:= input_ground_di,
    hmi:= dbs_switch_gear_1,
    on_do=> output_on_do,
    off_do=> output_off_do);
```

4.19 УПП (fb_урр)

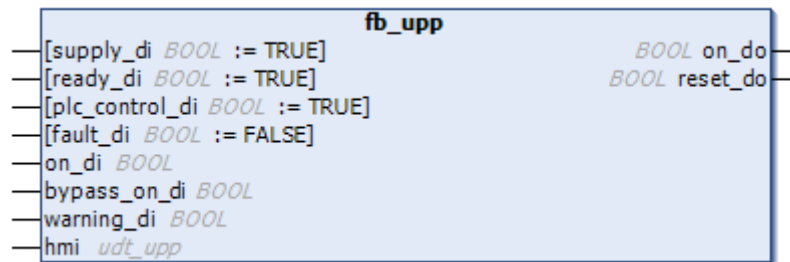
4.19.1 Назначение

Включение и выключение электродвигателя, подключенного к УПП.

4.19.2 Функции

- запуск и останов электродвигателя, подключенного к УПП,
- контроль подключения шунтирования в УПП.

4.19.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------|---------|--------------------|--|
| Input | supply_di | BOOL | TRUE | схема собрана/питание подано |
| | ready_di | BOOL | TRUE | готовность |
| | plc_control_di | BOOL | TRUE | автоматическое управление |
| | fault_di | BOOL | FALSE | неисправность |
| | on_di | BOOL | | включен |
| | bypass_on_di | BOOL | | сигнал BYPASSED - включения шунтирования |
| | warning_di | BOOL | | сигнал предупреждения |
| Inout | hmi | udt_урр | | интерфейс блока |
| Output | on_do | BOOL | | включить |
| | reset_do | BOOL | | сброс аварии |

4.19.4 Принцип работы

Блок переходит в состояние «ожидание готовности» из состояния «нет питания», когда есть сигнал «supply_di».

Проверяется наличие готовности «ready_di», отсутствие неисправности «fault_di» и переход в «отключен».

По команде «пуск» блок из состояния «остановлен» перейдет в «включается». На выход «on_do» подается сигнал. При поступлении «on_di» и «bypass_on_di» блок перейдет в состояние «в работе».

По команде «стоп» блок перейдет в состояние «отключается». С выхода «on_do» снимается сигнал. При снятии сигнала «on_di» и «bypass_on_di» блок перейдет в состояние «остановлен».

4.19.5 Интерфейс «udt_урр»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------|--------------|-------------|------------|--|
| std | udt_standard | | | стандартный набор переменных |
| time_on | REAL | 2 | ввод | время включения. Максимально-допустимое время ожидания (в секундах) прихода сигнала "on_di", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_off | REAL | 2 | ввод | время отключения. Максимально-допустимое время ожидания (в секундах) снятия сигнала "on_di", по истечении которого объект переходит в аварийное состояние, если сигнал не ушел. |
| time_accel | REAL | 5 | ввод | время разгона (для контроля шунтирования). Максимально-допустимое время ожидания (в секундах) прихода сигнала "bypass_on_di", по истечении которого объект переходит в аварийное состояние, если сигнал не пришел. |
| time_reset | REAL | 2 | ввод | длительность импульса сигнала сброса аварии «reset_do» |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |

4.19.6 Переменные интерфейса блока. Списки значений

4.19.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_urr» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|----------|
| stop | WORD | 16#1 | Стоп |
| start | WORD | 16#2 | Пуск |

4.19.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_urr» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|-------------------------|------|----------|--|
| start_timeout | WORD | 16#1 | «Превышено время включения». Сигнал "on_di" не пришел в течение времени «time_on» после подачи команды «Пуск» |
| stop_timeout | WORD | 16#2 | «Превышено время отключения». Сигнал "on_di" не ушел в течение времени «time_off» после подачи команды «Стоп». |
| unexpected_start | WORD | 16#4 | «Самопроизвольное включение». Поступление сигнала "on_di" без подачи команды «Пуск» |
| unexpected_stop | WORD | 16#8 | «Самопроизвольное отключение». Снятие сигнала "on_di" без подачи команды «Стоп» |
| fault | WORD | 16#10 | «Авария». Поступление сигнала на вход «fault_di». Данную аварию можно сбросить только после снятия сигнала. |
| power_off | WORD | 16#20 | «Пропадание питания». Снятие входного сигнала «supply_di» во время работы. |
| start_timeout_by_pass | WORD | 16#40 | «Превышено время включения шунта». Сигнал "bypass_on_di" не пришел в течение времени «time_accel» после поступления входа «on_di». |
| unexpected_start_bypass | WORD | 16#80 | «Самопроизвольное включение шунта». Поступление сигнала "bypass_on_di" без подачи команды «Пуск» |
| unexpected_stop_bypass | WORD | 16#100 | «Самопроизвольное отключение шунта». Пропадание сигнала "bypass_on_di" без подачи команды «Стоп» |

4.19.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_urr» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|--|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |
| not_ready | WORD | 16#2 | «Нет готовности». Отсутствие входного сигнала «ready_di». |
| fault | WORD | 16#4 | «Авария» По сигналу «fault_di» |
| warning | WORD | 16#8 | Наличие сигнала предупреждения «warning_di» |

4.19.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_urr» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-------------------|------|----------|--|
| control_bypass_on | WORD | 16#1 | «Контроль шунтирования». Включает контроль сигнала «bypass_on_di» |

4.19.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_urr» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|---------------|------|----------|---------------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |
| started | WORD | 16#4 | Включен |
| stopping | WORD | 16#8 | Отключается |
| starting | WORD | 16#10 | Включается |
| alarm | WORD | 16#20 | Авария |
| ready_waiting | WORD | 16#40 | Ожидание готовности |

4.19.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_urr» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-----------|------|----------|-------------|
| power_off | WORD | 16#1 | Нет питания |
| stopped | WORD | 16#2 | Отключен |
| started | WORD | 16#4 | Включен |

4.19.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
  // объявление интерфейса блока fb_урр для электропривода с УПП
  dbs_урр_1: RuDrive.udt_урр;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
  // объявление экземпляра блока fb_урр для электропривода с УПП
  dbi_урр_1: RuDrive.fb_урр;
END_VAR

// вызов экземпляра блока fb_урр для электропривода с УПП
dbi_урр_1(
  supply_di:= input_supply_di,
  ready_di:= input_ready_di,
  plc_control_di:= input_plc_control_di,
  fault_di:= input_fault_di,
  on_di:= input_on_di,
  bypass_on_di:= input_bypass_on_di,
  warning_di:= input_warning_di,
  hmi:= dbs_урр_1,
  on_do=> output_on_do,
  reset_do=> output_reset_do);
```

4.20 Задвижка (fb_valve)

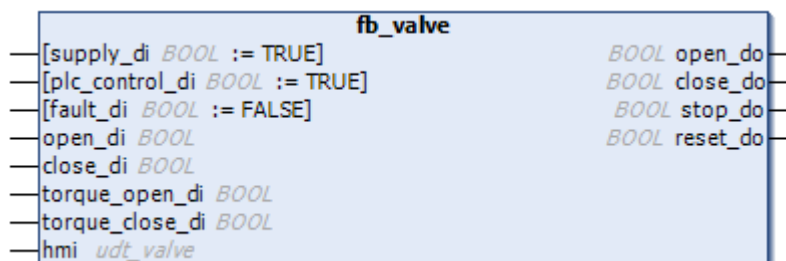
4.20.1 Назначение

Управление задвижкой с дискретным непрерывным управлением.

4.20.2 Функции

- управление задвижкой.

4.20.3 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 |
| | fault_di | BOOL | FALSE | неисправность |
| | open_di | BOOL | | концевик открытого положения |
| | close_di | BOOL | | концевик закрытого положения |
| | torque_open_di | BOOL | | моментный выключатель открытия |
| | torque_close_di | BOOL | | моментный выключатель закрытия |
| Inout | hmi | udt_valve | | интерфейс блока |
| Output | open_do | BOOL | | команда на открытие |
| | close_do | BOOL | | команда на закрытие |
| | stop_do | BOOL | | команда на останов движения |
| | reset_do | BOOL | | сброс аварии |

4.20.4 Принцип работы

Блок по наличию питания «supply_di» в зависимости от сигналов «open_di» и «close_di» из состояния «нет питания» переходит в одно из «открыт», «закрыт» или «промежуточное».

Команды «open» и «close» вызывают переход в состояние «открывается» или «закрывается». Подаются соответствующие выходные сигналы «open_do» либо «close_do».

Процесс открытия/закрытия завершается по приходу сигналов с концевых выключателей. Устанавливается состояние «открыта»/«закрыта».

По команде «стоп» снимаются выходы «open_do», «close_do» и подается «stop_do».

4.20.5 Интерфейс «udt_valve»

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

| Название | Тип | Нач. знач-е | Назначен ие | Описание |
|------------------|--------------|-------------|-------------|--|
| std | udt_standard | | | стандартный набор переменных |
| time_open | REAL | 60 | ввод | время открытия, сек. Время, за которое задвижка переходит из положения закрыта в положение открыта |
| time_close | REAL | 60 | ввод | время закрытия, сек. Время, за которое задвижка переходит из положения открыта в положение закрыта |
| time_open_leave | REAL | 20 | ввод | время схода с концевого выключателя открытого положения, сек |
| time_close_leave | REAL | 20 | ввод | время схода с концевого выключателя закрытого положения, сек |
| time_pulse | REAL | 1 | ввод | время импульса выхода «open_do»/«close_do», сек. |
| tag_name | STRING(16) | '' | ввод-вывод | диспетчерское наименование |
| time_reset | REAL | 2 | ввод | длительность импульса сигнала сброса аварии «reset_do», сек |

4.20.6 Переменные интерфейса блока. Списки значений

4.20.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_valve» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|----------|------|----------|----------|
| stop | WORD | 16#1 | Стоп |
| close | WORD | 16#2 | Закрыть |
| open | WORD | 16#4 | Открыть |

4.20.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_valve» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщений/Описание |
|------------------|------|----------|---|
| sw_open_timeout | WORD | 16#1 | «Превышено время схода с КО». Сигнал "open_di" не ушел в течение времени «time_open_leave» после подачи команды «Закрыть» |
| sw_close_timeout | WORD | 16#2 | «Превышено время схода с КЗ». Сигнал "close_di" не ушел в течение времени «time_close_leave» после подачи команды «Открыть» |
| open_timeout | WORD | 16#4 | «Превышено время открытия». После подачи команды «Открыть» в течении времени «time_open» не пришел сигнал "open_di". |
| close_timeout | WORD | 16#8 | «Превышено время закрытия». После подачи команды «Закрыть» в течении времени «time_close» не пришел сигнал "close_di". |
| unexpected_open | WORD | 16#10 | «Самопроизвольное срабатывание КО». Во время закрытия задвижки пришел сигнал «open_di» |
| unexpected_close | WORD | 16#20 | «Самопроизвольное срабатывание КЗ». Во время открытия задвижки пришел сигнал «close_di» |
| open_close | WORD | 16#40 | «Неопределенное состояние». Одновременное поступление сигналов «open_di» и «close_di». |
| fault | WORD | 16#80 | «Авария». Поступление сигнала на вход «fault_di». Данную аварию можно сбросить только после снятия сигнала. |

| Название | Тип | Значение | Текс сообщений/Описание |
|--------------|------|----------|--|
| torque_open | WORD | 16#100 | Поступление сигнала с моментного выключателя открытия «torque_open_di». |
| torque_close | WORD | 16#200 | Поступление сигнала с моментного выключателя закрытия «torque_close_di». |
| power_off | WORD | 16#400 | «Пропадание питания». Снятие входного сигнала «supply_di» во время движения задвижки. |

4.20.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_valve» п. 3.3.3.2. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текс сообщения/ Описание |
|-----------|------|----------|--|
| power_off | WORD | 16#1 | «Нет питания». Отсутствие входного сигнала «supply_di». |

4.20.6.4 Конфигурация «std.in_config»

Тип глобальных констант - «sys_db_cfg_valve» п.3.3.3.3. Ее возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|-------------------|------|----------|---|
| stop_in_manual | WORD | 16#1 | «Останов при переключении в местный режим» При переключении в местный режим подается сигнал «stop_do» |
| no_open_sw | WORD | 16#2 | «Отсутствии концевика открытия» Состояние «открыта» будет достигаться по истечении таймера «time_open» |
| no_close_sw | WORD | 16#4 | «Отсутствии концевика закрытия» Состояние «закрыта» будет достигаться по истечении таймера «time_close» |
| pulse_signal | WORD | 16#8 | «Импульсная логика управляющих сигналов» Выходные сигналы «open_do» и «close_do» при включении этого флага подаются кратковременно (в виде импульса) при подаче соответствующих команд, а не удерживаются постоянно. |
| hold_signal_open | WORD | 16#10 | «Удерживать управляющий сигнал в положении открыто» При достижении состояния «открыта» выход «open_do» не снимается |
| hold_signal_close | WORD | 16#20 | «Удерживать управляющий сигнал в положении закрыто» При достижении состояния «закрыта» выход «close_do» не снимается |

4.20.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_valve» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|---------------|
| power_off | WORD | 16#1 | Нет питания |
| closed | WORD | 16#2 | Закрыта |
| opened | WORD | 16#4 | Открыта |
| intermediate | WORD | 16#8 | Промежуточное |
| alarm | WORD | 16#10 | Авария |
| opening | WORD | 16#20 | Открывается |
| closing | WORD | 16#40 | Закрывается |

4.20.6.6 Состояние объекта «std.out_state_actual»

Тип глобальных констант - «sys_db_hmi_state_actual_valve» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|---------------|
| power_off | WORD | 16#1 | Нет питания |
| closed | WORD | 16#2 | Закрыта |
| opened | WORD | 16#4 | Открыта |
| intermediate | WORD | 16#8 | Промежуточное |

4.20.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока fb_valve для задвижки
    dbs_valve_1: RuDrive.udt_valve;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока fb_valve для задвижки
    dbi_valve_1: RuDrive.fb_valve;
END_VAR

// вызов экземпляра блока fb_valve для для задвижки
dbi_valve_1(
    supply_di:= input_supply_di,
    plc_control_di:= input_plc_control_di,
    fault_di:= input_fault_di,
    open_di:= input_open_di,
    close_di:= input_close_di,
    torque_open_di:= input_torque_open_di,
    torque_close_di:= input_torque_close_di,
    hmi:= dbs_valve_1,
    open_do=> output_open_do,
    close_do=> output_close_do,
    stop_do=> output_stop_do,
    reset_do=> output_reset_do);
```

4.21 Технологическая защита (fb_protection)

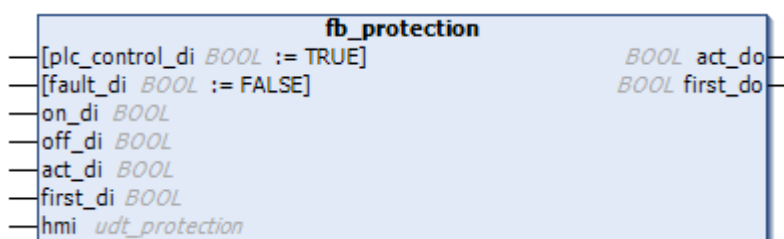
4.21.1 Назначение

Предназначена для подачи сигнала аварии (отключения) на защищаемый контур или техпроцесс.

4.21.2 Функции

- ввод работу и вывод из работы защиты как по внешним условиям, так и по команде оператора.
- определение первой сработавшей защиты (при наличии нескольких взаимодействующих блоков защит).
- возможность теста защиты.
- автоматическое снятие аварии.

4.21.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------|----------------|--------------------|---|
| Input | plc_control_di | BOOL | TRUE | не используется |
| | fault_di | BOOL | FALSE | не используется |
| | on_di | BOOL | | ввод защиты в авт. режиме, импульс или удержание (конфигурация) |
| | off_di | BOOL | | вывод защиты в авт. режиме, импульс |
| | act_di | BOOL | | условие срабатывания защиты |
| | first_di | BOOL | | наличие первой сработавшей защиты (сигнал first_do от других защит) |
| Inout | hmi | udt_protection | | обмен данными с верхним уровнем |
| Output | act_do | BOOL | | срабатывание защиты (сигнал в САУ) |
| | first_do | BOOL | | защита сработала первой (для подачи на first_di других защит) |

4.21.4 Принцип работы

Блок защиты вырабатывает сигнал аварии на своем выходе «act_do», если вход «act_di» (условие срабатывания) принимает значение «TRUE». Срабатывание защиты происходит не сразу, а через время задержки, которое задается параметром «time_countdown» в интерфейсе блока.

Для того, чтобы защита реагировала на условие срабатывания, подаваемое на вход «act_di», защиту нужно ввести в работу.

В случае, если выбран режим «автомат», ввод защиты производится подачей и удержанием значения «TRUE» на вход «on_di», вывод – подачей значения «FALSE».

Возможен альтернативный вариант ввода и вывода защиты импульсными сигналами на входы «on_di» и «off_di». Для его включения нужно установить бит конфигурации «pulse_signal».

Если защита в режиме «диспетчер», то ввод и вывод защиты выполняется по команде оператора (см. раздел «команды»).

Защита имеет два типа срабатывания – «на сигнал» и «в работу», которые задаются параметром «action_type» в интерфейсе блока. Если установлен тип срабатывания «на сигнал», то защита при срабатывании будет только формировать аварийное сообщение в «std.out_alarm», но выход «act_do» установлен не будет. Если тип срабатывания – «в работу», то при срабатывании защиты будет также установлен выход «act_do».

Чтобы определить, какая из защит сработала первой при совместном использовании нескольких защит, предусмотрены вход «first_di» и выход «first_do». Выход «first_do» каждой из защит нужно подать на входы «first_di» остальных. Если защит более двух, то объединить подаваемые сигналы логическим «или».

Время срабатывания защиты выводится в структуру «datetime» интерфейса блока.

Если необходимо проверить срабатывание защиты, не активируя вход «act_di» (т.е. при отсутствии условия срабатывания), нужно подать команду «test» (тестирование защиты). При подаче этой команды защита срабатывает полноценно, т.е. формируется аварийное сообщение в «std.out_alarm» и устанавливается выход «act_do». После истечения времени «time_test» (время действия тестового сигнала) защита возвращается в исходное состояние, аварийное сообщение снимается, выход «act_do» устанавливается в «FALSE».

Если необходимо, чтобы после срабатывания защиты в обычном режиме (не по команде «test») авария защиты снималась автоматически, нужно установить бит конфигурации «alarm_auto_reset». Снятие аварии будет происходить после того, как будет отсчитано время, задаваемое параметром «time_auto_reset».

4.21.5 Интерфейс «udt_protection»

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-----------------------|--------------|-------------|------------|---|
| std | udt_standard | | ввод-вывод | стандартный набор переменных |
| datetime | udt_datetime | | вывод | Время срабатывания. В структуру выводится время срабатывания защиты. |
| time_countdown | REAL | 10 | ввод | Время обратного отсчета до срабатывания, сек. Время, через которое срабатывает защита после активации входа «act_di». |
| time_test | REAL | 10 | ввод | Время действия тестового сигнала защиты, сек. Время, в течение которого защита находится в состоянии срабатывания после прихода команды «test». |
| time_auto_reset | REAL | 120 | ввод | Время автоматического снятия аварии, сек. Время, после истечения которого снимается состояние срабатывания защиты. Работает только при условии, что бит конфигурации «alarm_auto_reset» установлен. |
| time_countdown_remain | REAL | | вывод | Осталось до срабатывания, сек. Таймер обратного отсчета до срабатывания защиты после активации входа «act_di». |
| act_di | BOOL | | вывод | Состояние условия срабатывания защиты (вход). Сюда передается состояние входа «act_di». |
| first_do | BOOL | | вывод | Защита сработала первой. Сюда передается состояние выхода «first_do». |
| action_type | BOOL | | ввод | Тип срабатывания: 0 - на сигнал, 1 - в работу. «На сигнал» - защита срабатывает с выводом только аварийного сообщения. «В работу» - защита срабатывает с выводом аварийного сообщения и установкой выхода «act_do». |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|----------|------------|-------------|------------|-----------------------------|
| reserve | BOOL | | - | Резерв. |
| tag_name | STRING(16) | '' | ввод-вывод | Диспетчерское наименование. |

4.21.6 Переменные интерфейса блока. Списки значений

4.21.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_protection» п. 3.3.3.4. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текст сообщений |
|----------|------|----------|-------------------------|
| stop | WORD | 16#1 | Вывод защиты из работы. |
| start | WORD | 16#2 | Ввод защиты в работу. |
| test | WORD | 16#4 | Тестирование защиты. |

4.21.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_protection» п. 3.3.3.1. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Текст сообщения/описание |
|----------|------|----------|--------------------------|
| fault | WORD | 16#1 | Срабатывание защиты. |

4.21.6.3 Предупредительные сообщения «std.out_warning».

Данный блок не имеет предупредительных сообщений.

4.21.6.4 Конфигурация «std.in_config»

Тип глобальных констант – «sys_db_cfg_protection» п. 3.3.3.3. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Описание |
|--------------|------|----------|--|
| pulse_signal | WORD | 16#1 | Импульсная логика сигналов ввода защиты. Если бит установлен, то ввод защиты в режиме «автомат» выполняется по входу «on_di» (импульс), а вывод – по входу «off_di» (импульс). |

4.21.6.5 Состояние блока «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_protection» п. 3.3.3.6. Возможные значения представлены в таблице ниже.

| Название | Тип | Значение | Комментарий |
|----------|------|----------|---|
| off | WORD | 16#1 | Выведена. Защита была выведена из работы оператором либо входами «on_di», «off_di». |
| on | WORD | 16#2 | Введена. Защита была введена в работу оператором либо входом «on_di». |

| Название | Тип | Значение | Комментарий |
|----------|------|----------|---|
| operated | WORD | 16#4 | Сработала. Был установлен вход «act_di» и истекло время до срабатывания защиты. |
| test | WORD | 16#8 | Тест. Была подана команда «test». |

4.21.6.6 Состояние объекта «std.out_state_actual»

Блок не имеет состояний данного типа.

4.21.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока защиты
    dbs_protection:          RuDrive.udt_protection;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
VAR_GLOBAL
    // объявление интерфейса блока защиты
    dbs_protection:          RuDrive.udt_protection;
END_VAR

// вызов экземпляра блока
dbi_protection(
    plc_control_di:= TRUE,
    fault_di:= FALSE,
    on_di:= on_condition_di,
    off_di:= FALSE,
    act_di:= activation_condition_di,
    first_di:= FALSE,
    hmi:= dbs_protection,
    act_do=> activation_do );
```

4.22 Сумматор расхода (fb_flow_counter)

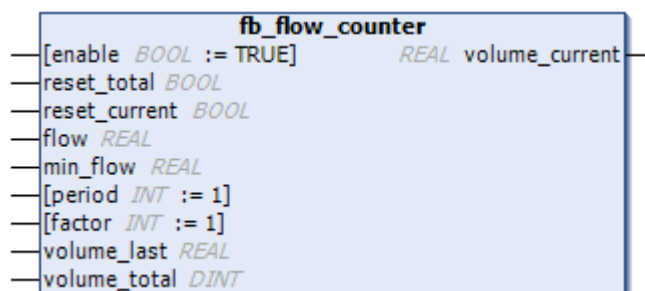
4.22.1 Назначение

Подсчет суммарного расхода на основе мгновенного значения расхода.

4.22.2 Функции

- подсчет суммарного расхода за заданный период времени.
- подсчет суммарного общего расхода.

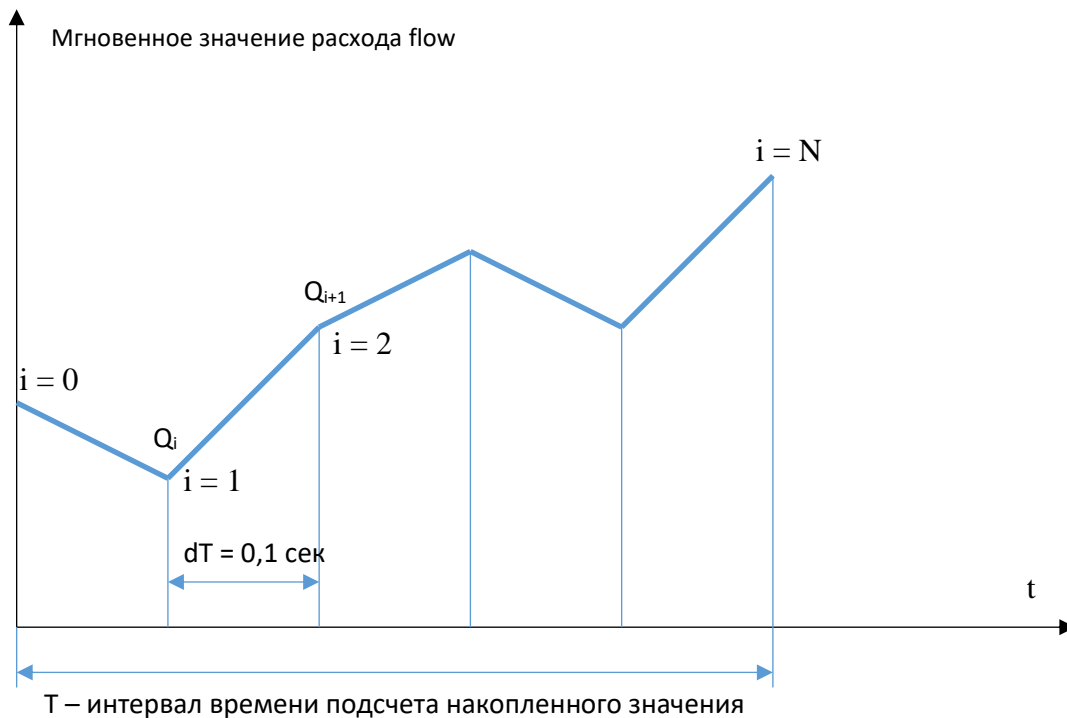
4.22.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------|------|--------------------|--|
| Input | enable | BOOL | TRUE | Разрешение подсчета расхода. Если TRUE, разрешает подсчет расхода. |
| | reset_total | BOOL | | Сброс общего расхода. Если TRUE, сбрасывает общий расход «volume_total» к значению 0. |
| | reset_current | BOOL | | Сброс текущего расхода. Если TRUE, сбрасывает текущий расход «volume_current» к значению 0. |
| | flow | REAL | | Текущий расход. Вход для подачи мгновенного значения расхода (например, с датчика). |
| | min_flow | REAL | | Минимальный значимый расход. Если мгновенное значение расхода flow ниже этого значения, то суммирование не производится. |
| | period | INT | 1 | Период подсчета суммарного расхода. Текущее накапливаемое значение отображается «volume_current». После завершения периода записывается в «volume_last», при этом «volume_current» обнуляется и начинает считать заново. 0 - раз в сек, 1 - в мин, 2 - в час, 3 - в сутки, 4 - в месяц. |
| | factor | INT | 1 | Множитель периода. Позволяет увеличить значение периода подсчета «volume_last» в необходимое кол-во раз. |
| Output | volume_current | REAL | | Суммарный текущий расход за период. Значение обновляется по мере накопления. При истечении периода сбрасывается в 0. |
| Inout | volume_last | REAL | | Суммарный расход за последний указанный период. После завершения периода подсчета суммарного расхода значение из «volume_current» записывается сюда. «volume_current» обнуляется и начинает считать заново. |
| | volume_total | DINT | | Суммарный общий расход за все время работы блока. |

4.22.4 Принцип работы

Подсчет суммарного расхода по его мгновенному значению выполняется по методике, приведенной ниже. Единицы измерения суммарного расхода те же, в которых на вход блока подается мгновенное значение расхода.



Суммарный расход за интервал времени T :

$$Q_{\Sigma} = \frac{dT}{3600} * \sum_{i=0}^{N-1} \frac{Q_{i+1} + Q_i}{2},$$

где N – число точек измерения текущего значения, $N = \frac{T}{dT} + 1$,

dT – период измерения мгновенного значения расхода (=период вызова блока (0,1 сек)).

4.22.5 Пример

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    dbi_flow_counter:   RuDrive.fb_flow_counter;
    sum_flow_current:   REAL;
    sum_flow_period:    REAL;
    sum_flow_total:     DINT;
END_VAR

dbi_flow_counter(
    enable:= TRUE,
    reset_total:= FALSE,
    reset_current:= FALSE,
    flow:= flow_value,
    min_flow:= 0.1,
    period:= 2,
    factor:= 1,
```

```
volume_current=> sum_flow_current,  
volume_last:= sum_flow_period,  
volume_total:= sum_flow_total);
```

4.23 Счетчик импульсов (fb_pulse_counter)

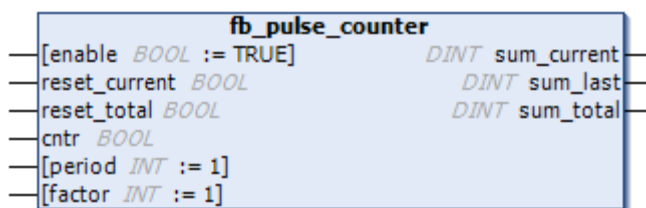
4.23.1 Назначение

Подсчет числа импульсов, подаваемых на счетный вход.

4.23.2 Функции

- подсчет числа импульсов за заданный период времени.
- подсчет общего числа импульсов за все время работы.

4.23.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|---------------|------|--------------------|--|
| Input | enable | BOOL | TRUE | Разрешение подсчета импульсов. Если TRUE, разрешает подсчет числа импульсов. |
| | reset_total | BOOL | | Сброс общего числа импульсов. Если TRUE, сбрасывает общий счетчик «sum_total» к значению 0. |
| | reset_current | BOOL | | Сброс текущего числа импульсов. Если TRUE, сбрасывает текущий счетчик «sum_current» к значению 0. |
| | cntr | BOOL | | Счетный вход. Вход для подачи импульсного сигнала (например, с датчика), для которого требуется подсчет числа импульсов. Длительность импульса должна быть больше, чем период вызова блока, иначе импульс не сможет быть зафиксирован. |
| | period | INT | 1 | Период подсчета числа импульсов. Текущее накапливаемое значение отображается в «sum_current». После завершения периода записывается в «sum_last», при этом «sum_current» обнуляется и начинает считать заново. 0 - раз в сек, 1 - в мин, 2 - в час, 3 - в сутки, 4 - в месяц. |
| | factor | INT | 1 | Множитель периода. Позволяет увеличить значение периода подсчета «sum_last» в необходимое кол-во раз. |
| Output | sum_current | DINT | | Число импульсов за период. Значение обновляется по мере накопления. При истечении периода переписывается в «sum_last», затем сбрасывается в 0 и начинает отсчет заново. |
| | sum_last | DINT | | Число импульсов за последний указанный период. После завершения периода подсчета значение из «sum_current» записывается сюда. «sum_current» обнуляется и начинает считать заново. |
| | sum_total | DINT | | Суммарное число импульсов за все время работы блока. |

4.23.4 Пример

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    dbi_pulse_counter: RuDrive.fb_pulse_counter;
    sum_current:       DINT;
    sum_period:        DINT;
    sum_total:         DINT;
END_VAR

dbi_pulse_counter(
    enable:= TRUE,
    reset_total:= FALSE,
    reset_current:= FALSE,
    cntr:= pulse_value,
    period:= 2,
    factor:= 1,
    sum_current=> sum_current,
    sum_last=> sum_period,
    sum_total=> sum_total);
```

4.24 Счетчик времени (fb_time_counter)

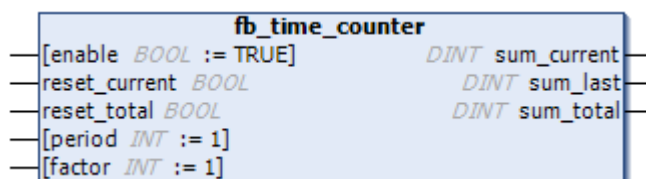
4.24.1 Назначение

Подсчет суммарного времени нахождения в активном состоянии (TRUE) сигнала, подаваемого на вход блока (enable). Блок может быть использован для подсчета времени работы агрегатов и т.п.

4.24.2 Функции

- подсчет времени работы за заданный промежуток времени.
- подсчет общего времени работы (за все время работы блока).

4.24.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|---------------|------|--------------------|---|
| Input | enable | BOOL | TRUE | Разрешение счета. Если TRUE, включает подсчет времени. Минимальная длительность активного состояния входа не должна быть меньше периода вызова блока (0,1 сек по умолчанию). Импульсы меньшей длительности не будут зафиксированы. |
| | reset_total | BOOL | | Сброс общего времени работы. Если TRUE, сбрасывает общий счетчик «sum_total» к значению 0. |
| | reset_current | BOOL | | Сброс текущего времени работы. Если TRUE, сбрасывает текущий счетчик «sum_current» к значению 0. |
| | period | INT | 1 | Период подсчета времени работы. Текущее накапливаемое значение отображается в «sum_current». После завершения периода записывается в «sum_last», при этом «sum_current» обнуляется и начинает считать заново. 0 - раз в сек, 1 - в мин, 2 - в час, 3 - в сутки, 4 - в месяц. |
| | factor | INT | 1 | Множитель периода. Позволяет увеличить значение периода подсчета «sum_last» в необходимое кол-во раз. |
| Output | sum_current | DINT | | Время работы за период, сек. Значение обновляется по мере накопления. При истечении периода переписывается в «sum_last», затем сбрасывается в 0 и начинает отсчет заново. |
| | sum_last | DINT | | Время работы, сек, за последний указанный период. После завершения периода подсчета значение из «sum_current» записывается сюда. «sum_current» обнуляется и начинает считать заново. |
| | sum_total | DINT | | Общее время работы за все время работы блока, сек. |

4.24.4 Пример

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    dbi_pulse_counter: RuDrive.fb_time_counter;
    sum_current:      DINT;
    sum_period:       DINT;
    sum_total:        DINT;
END_VAR

dbi_time_counter(
    enable:= TRUE,
    reset_total:= FALSE,
    reset_current:= FALSE,
    period:= 2,
    factor:= 1,
    sum_current=> sum_current,
    sum_last=> sum_period,
    sum_total=> sum_total);
```

4.25 Очередь (fb_queue)

4.25.1 Назначение

Блок предназначен для выполнения функции арбитра разделяемого ресурса для других блоков/алгоритмов, которые выступают потребителями этого ресурса.

4.25.2 Функции

- выдача разрешений занять ресурс потребителю по запросу.
- постановка потребителей, желающих занять ресурс, в очередь, если ресурс в данный момент занят.
- максимальный размер очереди – 10 потребителей.

4.25.3 Таблица входов-выходов

| fb_queue | |
|--------------------------|---|
| resource_busy_ext | BOOL |
| resource_busy_ext_enable | BOOL |
| consumer_status | ARRAY[1..MAX_CONSUMER] OF udt_consumer_status |
| max_consumer_one_time | INT := 1 |
| consumer_status_out | ARRAY[1..MAX_CONSUMER] OF udt_consumer_status |
| queue | ARRAY[1..MAX_CONSUMER] OF INT |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|--------------------------|--|--------------------|--|
| Input | resource_busy_ext | BOOL | | условие занятости общего ресурса (внешнее) |
| | resource_busy_ext_enable | BOOL | | включить использование внешнего условия занятости ресурса |
| | consumer_status | ARRAY [1..MAX_CONSUMER] OF udt_consumer_status | | обмен состояниями с потребителем ресурса |
| | max_consumer_one_time | INT | 1 | макс. число потребителей, которые могут одновременно занять ресурс |
| Output | consumer_status_out | ARRAY [1..MAX_CONSUMER] OF udt_consumer_status | | обмен состояниями с потребителем ресурса |
| | queue | ARRAY [1..MAX_CONSUMER] OF INT | | очередь потребителей |

Тип данных «udt_consumer_status»

| Название | Тип | Нач. значение | Назначение | Комментарий |
|----------|------|---------------|------------|--|
| free | BOOL | | вывод | Ресурс свободен. Статус для потребителя. |
| request | BOOL | | ввод | Запрос разрешения занять ресурс. Команда от потребителя. |
| permit | BOOL | | вывод | Разрешение занять ресурс. Статус для потребителя. |
| busy | BOOL | | ввод | Ресурс занят. Команда от потребителя. |

4.25.4 Принцип работы

Данный блок выполняет функцию арбитра некоторого общего для других алгоритмов ресурса, который они могут использовать. Эти алгоритмы являются потребителями этого ресурса.

Блок принимает запросы от потребителей на возможность использования ресурса. В ответ блок выдает статус, свободен ли ресурс и разрешено ли его занять. Обмен данными с самим ресурсом в блоке не предусмотрен.

На вход блока «consumer_status» передается массив значений типа «udt_consumer_status». Каждая ячейка массива соответствует номеру конкретного потребителя (N – номер потребителя). Потребитель должен заполнить следующие поля:

- «consumer_status[N].request»;
- «consumer_status[N].busy».

Состояние ресурса каждый потребитель получает из выходного массива «consumer_status_out». Каждая ячейка массива соответствует номеру конкретного потребителя. Из этой структуры данных каждый потребитель получает следующие поля статусов:

- «consumer_status_out[N].free»;
- «consumer_status_out[N].permit».

Перед тем, как занять ресурс, потребитель должен проверить, свободен ли ресурс. Для этого предназначено выходное поле «consumer_status_out[N].free». Если его значение равно «TRUE», то ресурс свободен, «FALSE» - занят.

Если ресурс свободен, потребитель должен дать запрос разрешения занять ресурс. Для этого ему нужно установить входное поле «consumer_status[N].request» в значение «TRUE».

В случае, если ресурс занят, потребитель может встать в очередь и ждать, когда он освободится. Для этого ему также нужно установить поле «consumer_status[N].request» в значение «TRUE». При этом потребитель будет поставлен в очередь, и в конец массива «queue» будет добавлен номер потребителя (в очереди уже может кто-то стоять).

Затем потребитель должен дождаться, когда блок очереди выдаст разрешение на занятие ресурса, установив поле «consumer_status_out[N].permit» в значение «TRUE». Если потребитель стоит в очереди, то сначала будет обслужен потребитель, который стоит в очереди раньше.

Потребитель может принять это разрешение, сообщив блоку очереди, что он занял ресурс. Для этого потребитель должен установить поле «consumer_status[N].busy» в значение «TRUE». После этого потребитель может начать использовать ресурс. Пока ресурс используется, поле «consumer_status[N].busy» должно удерживаться в «TRUE».

Если потребителю больше не требуется ресурс, он должен его освободить, установив поле «consumer_status[N].busy» в значение «FALSE». В случае, если в очереди на ресурс стояли другие потребители, то первому в очереди будет дано разрешение занять ресурс. Если же больше нет потребителей, которым требуется ресурс, тогда поле «consumer_status_out[N].free» будет установлено в значение «TRUE».

Входной параметр «max_consumer_one_time» задает количество потребителей, которые могут занять ресурс одновременно.

Входной параметр «resource_busy_ext_enable» (если он «TRUE») разрешает использовать внешнее условие занятости ресурса, т.е. это условие будет формировать не сам блок, а внешний алгоритм. В этом случае само условие занятости нужно подать на вход «resource_busy_ext».

4.25.5 Пример

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    dbi_queue: RuDrive.fb_queue;
END_VAR

// команды от потребителей
dbi_queue.consumer_status[1].request := consumer[1].request;
dbi_queue.consumer_status[1].busy := consumer[1].busy;

dbi_queue.consumer_status[2].request := consumer[2].request;
dbi_queue.consumer_status[2].busy := consumer[2].busy;

dbi_queue.consumer_status[3].request := consumer[3].request;
dbi_queue.consumer_status[3].busy := consumer[3].busy;
```

```
// вызов экземпляра блока очереди
dbi_queue(
    resource_busy_ext:= FALSE,
    resource_busy_ext_enable:= FALSE,
    max_consumer_one_time:= 1 );

// статусы для потребителей
dbi_queue.consumer_status_out[1].free := consumer[1].free;
dbi_queue.consumer_status_out[1].permit := consumer[1].permit;

dbi_queue.consumer_status_out[2].free := consumer[2].free;
dbi_queue.consumer_status_out[2].permit := consumer[2].permit;

dbi_queue.consumer_status_out[3].free := consumer[3].free;
dbi_queue.consumer_status_out[3].permit := consumer[3].permit;
```

4.26 Блок ротации (fb_rotation)

4.26.1 Назначение

Блок предназначен для расчета параметров очередности запуска и останова агрегатов по наработке среди группы агрегатов.

4.26.2 Функции

- вычисление наработки каждого агрегата в часах.
- вычисление номера агрегата, который должен быть включен следующим, по минимальной наработке среди неработающих агрегатов.
- вычисление номера агрегата, который должен быть отключен следующим, по максимальной наработке среди работающих агрегатов.
- выдача сигнала о необходимости ротации одного из агрегатов.
- максимальное кол-во агрегатов – 5.

4.26.3 Таблица входов-выходов

| Область | Название | Тип | Начальное значение | Назначение |
|---------|-------------------|----------------------------|--------------------|---|
| Input | ready | ARRAY [0..3, 1..5] OF BOOL | | готовность агрегата N |
| | run | ARRAY [0..3, 1..5] OF BOOL | | агрегат N в работе |
| | rotation_interval | DINT | | интервал ротации, ч |
| Output | time_operation | ARRAY [1..5] OF DINT | | наработка агрегата N, ч |
| | next_start | ARRAY [0..3] OF INT | | следующий для включения агрегат по наработке |
| | next_stop | ARRAY [0..3] OF INT | | следующий для остановки агрегат по наработке |
| | rotation_required | ARRAY [0..3] OF BOOL | | требуется ротация работающего насоса (по наработке) |

4.26.4 Принцип работы

Блок выполняет расчет следующих параметров:

- наработка каждого агрегата «time_operation[N]» в часах;
- номер следующего для включения агрегата по наработке «next_start[G]»;
- номер следующего для отключения агрегата по наработке «next_stop[G]»;
- сигнал о необходимости ротации агрегатов «rotation_required».

Здесь N – номер агрегата, 1...5.

G – схема включения, если под агрегатом понимается насосный агрегат из данной библиотеки. Схема включения используется только блоком насосной станции «fb_pump_sau», и не обязательна к применению при использовании данного блока отдельно в других алгоритмах. В этом случае значение G можно задавать равным 0.

Есть следующие варианты схема включения:

- 0 – любая схема,
- 1 – от ПЧ,
- 2 – от УПП,
- 3 – от сети.

Разделение по схемам включения может потребоваться, если необходимо отдельно рассчитывать номера для включения и отключения агрегатов по наработке для разных схем.

Блок выполняет расчет параметров для каждой из схем включения отдельно, кроме времени наработки «time_operation», которое подсчитывается всегда, если агрегат в работе по какой-либо из схем.

На вход «ready» необходимо подать статусы готовности агрегатов (для каждого агрегата N): «TRUE» - если агрегат готов к включению, «FALSE» - если нет.

На вход «run» нужно подать статус работы агрегатов (для каждого агрегата N): «TRUE» - если агрегат в работе, «FALSE» - если нет.

Первый индекс массивов «ready» и «run» - номер схемы включения (можно задать 0), второй индекс – номер агрегата.

На вход «rotation_interval» подается значение времени в часах, через которое требуется переход с одного агрегата на другой, т.е. отключение работающего агрегата и включение вместо него одного из неработающих.

Наработка агрегата «time_operation[N]»

Подсчет наработки агрегата активируется, когда на вход «run» приходит сигнал, что агрегат в работе. От схемы включения подсчет наработки не зависит – подсчет ведется всегда, когда хотя бы для одной из схем включения состояние «работа» агрегата равно «TRUE».

Например, если «run[0, 1] = TRUE» или «run[3, 1] = TRUE», значение «time_operation[1]» для 1-го агрегата будет увеличиваться.

Следующий для включения агрегат по наработке «next_start[G]»

Значение этого параметра рассчитывается следующим образом: среди агрегатов, которые не в работе и готовы к пуску, находится агрегат с минимальной наработкой. Найденное значение записывается на выход «next_start». Расчет ведется для каждой из схем включения агрегатов отдельно.

Следующий для отключения агрегат по наработке «next_stop[G]»

Значение этого параметра рассчитывается следующим образом: среди агрегатов, которые в работе, находится агрегат с максимальной наработкой. Найденное значение записывается на выход «next_stop». Расчет ведется для каждой из схем включения (G) агрегатов отдельно.

Требуется ротация насоса «rotation_required»

Значение этого выхода устанавливается в «TRUE», если у одного или нескольких насосов, находящихся в работе, время наработки «time_operation» превысило значение интервала ротации «rotation_interval». При этом выход «next_stop» будет содержать номер насоса, который нужно отключить, а «next_start» - номер насоса, который нужно включить.

4.26.5 Пример

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    dbi_rotation:           RuDrive.fb_rotation;
    time_operation:        ARRAY[1..5] OF DINT;
    next_start:            INT;
    next_stop:             INT;
    rotation_required:     BOOL;
END_VAR

// передача состояния готовности агрегатов
dbi_rotation.ready[0, 1] := pump_1_ready;    // насос 1
dbi_rotation.ready[0, 2] := pump_2_ready;    // насос 2
dbi_rotation.ready[0, 3] := pump_3_ready;    // насос 3

// передача состояния "работа" агрегатов
dbi_rotation.run[0, 1] := pump_1_run;        // насос 1
dbi_rotation.run[0, 2] := pump_2_run;        // насос 2
```

```
dbi_rotation.run[0, 3] := pump_3_run;           // насос 3

// вызов экземпляра блока ротации
dbi_rotation(
    rotation_interval:= 24,
    time_operation=>time_operation );

// далее используем результаты расчета

// номер агрегата, который можно включить
next_start := dbi_rotation.next_start[0];
// номер агрегата, который можно отключить
next_stop := dbi_rotation.next_stop[0];
// флаг "требуется ротация"
rotation_required := dbi_rotation.rotation_required[0];
```

4.27 Непрерывный ПИД-регулятор (CONT_C)

4.27.1 Назначение

Регулятор непрерывного действия предназначен для управления технологическими процессами с непрерывными входными и выходными переменными.

4.27.2 Функции

- независимое включение и отключение P, I, D составляющих.
- масштабирование регулируемой переменной (переменной процесса).
- ручное задание управляющего воздействия (выхода регулятора).
- масштабирование выхода регулятора.
- задание верхнего и нижнего предела выхода регулятора.
- внешнее возмущающее воздействие.

4.27.3 Таблица входов-выходов

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--------------------|--|
| Input | COM_RST | BOOL | | Полный рестарт. Если вход установлен, производится сброс накопленных значений пропорциональной, интегральной, дифференциальной составляющих, обнуление выхода регулятора. |
| | MAN_ON | BOOL | TRUE | Включить ручной режим. Если этот вход установлен, то контур управления разрывается. Значение, заданное вручную в параметре «MAN», устанавливается в качестве управляющего. |
| | PVPER_ON | BOOL | | Чтение входной переменной с периферии включить. Если значение переменной процесса должно считываться с модуля аналогового ввода (периферийного входа), то на вход «PV_PER» должно быть подано необработанное значение аналогового сигнала с периферийного входа, а вход «PVPER_ON» должен быть установлен. Параметр используется только для платформы Siemens. |
| | P_SEL | BOOL | TRUE | Включить пропорциональную составляющую. Управляющее воздействие складывается из P (пропорциональной), I (интегральной), D (дифференциальной) составляющих. Если вход установлен, то расчет пропорциональной составляющей выполняется, иначе она приравнивается к нулю. |
| | I_SEL | BOOL | TRUE | Включить интегральную составляющую. Управляющее воздействие складывается из P (пропорциональной), I (интегральной), D (дифференциальной) составляющих. Если вход установлен, то расчет интегральной составляющей выполняется, иначе она приравнивается к нулю. |
| | INT_HOLD | BOOL | | Удержание интегральной составляющей. Выход интегратора может быть "заморожен" установкой входа "INT_HOLD". |
| | I_ITL_ON | BOOL | | Инициализировать интегральную составляющую. Если вход установлен, значение интегральной составляющей становится равным значению входа «I_ITL_VAL». |
| | D_SEL | BOOL | TRUE | Дифференциальную составляющую включить. Управляющее воздействие складывается из P (пропорциональной), I (интегральной), D (дифференциальной) составляющих. |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--------------------|--|
| | | | | Если вход установлен, то расчет дифференциальной составляющей выполняется, иначе она приравнивается к нулю. |
| | CYCLE | TIME | TIME#1s0ms | Время выполнения блока. Время между вызовами блока должно быть постоянным. Вход «CYCLE» задает время между вызовами блока. |
| | SP_INT | REAL | 0 | Внутреннее задание. Вход «Внутреннее значение уставки» служит для установки заданного значения регулятора. |
| | PV_IN | REAL | 0 | Входная переменная. Значение регулируемой переменной (переменной процесса). |
| | PV_PER | WORD | 0 | Входная переменная (периферия). Этот вход предназначен для ввода значения переменной процесса напрямую с аналогового входа модуля ввода, если установлен вход «PVPER_ON». Значение с модуля ввода внутри блока преобразуется к диапазону значений 0...100 (%), а затем масштабируется с помощью коэффициентов «PV_FAC» и «PV_OFF». Значение переменной процесса рассчитывается по следующей формуле: $PV = (PV_PER/27648)*100*PV_FAC + PV_OFF$. Параметр используется только для платформы Siemens. |
| | MAN | REAL | 0 | Ручное задание выхода (управляющего воздействия). Если вход «MAN_ON» установлен, то значение входа «MAN» передается напрямую на выход регулятора «LMN». |
| | GAIN | REAL | 2 | Коэффициент пропорциональности. Вход задает коэффициент усиления регулятора. |
| | TI | TIME | TIME#20s0ms | Время интегрирования. Определяет временную характеристику интегратора. |
| | TD | TIME | TIME#10s0ms | Время дифференцирования. Определяет временную характеристику дифференцирующего звена. |
| | TM_LAG | TIME | TIME#2s0ms | Время действия дифференциальной составляющей. Время запаздывания дифференциального воздействия. Алгоритм дифф. составляющей содержит запаздывание, которое может быть задано входом «TM_LAG». |
| | DEADB_W | REAL | 0 | Ширина зоны нечувствительности. Если рассогласование по модулю меньше зоны нечувствительности ($ SP-PV < DEADB_W$), то оно считается равным нулю, и регулятор на него не реагирует, т.е. не изменяет управляющего воздействия. |
| | LMN_HLM | REAL | 100 | Верхний предел выходного сигнала. Ограничивает значение управляющего воздействия сверху данным значением. |
| | LMN_LLM | REAL | 0 | Нижний предел выходного сигнала. Ограничивает значение управляющего воздействия снизу данным значением. |
| | PV_FAC | REAL | 1 | Коэффициент для корректировки входной переменной (множитель). Используется для масштабирования входного сигнала, если задействован ввод регулируемой переменной с периферии (вход «PVPER_ON» установлен). Значение с периферийного аналогового входа «PV_PER», преобразованное к диапазону значений 0...100 (%), умножается на этот коэффициент. Параметр используется только для платформы Siemens. |
| | PV_OFF | REAL | 0 | Коэффициент для корректировки входной переменной (смещение). Используется для масштабирования входного сигнала, если задействован ввод регулируемой переменной с периферии (вход «PVPER_ON» установлен). Значение с периферийного аналогового входа «PV_PER», преобразованное к диапазону значений 0...100 (%) и |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--------------------|--|
| | | | | умноженное на значение входа «PV_FAC», складывается с этим коэффициентом. Параметр используется только для платформы Siemens. |
| | LMN_FAC | REAL | 1 | Коэффициент для корректировки выходной переменной (множитель). Используется для масштабирования выходного сигнала регулятора. Значение управляющего воздействия, изменяющееся в диапазоне [LMN_LLM, LMN_HLM], умножается на «LMN_FAC». |
| | LMN_OFF | REAL | 0 | Коэффициент для корректировки выходной переменной (смещение). Используется для масштабирования выходного сигнала регулятора. Значение управляющего воздействия, изменяющееся в диапазоне [LMN_LLM, LMN_HLM], умножается на «LMN_FAC», а затем складывается с «LMN_OFF». |
| | I_ITLVAL | REAL | 0 | Начальное значение интегральной составляющей. Если вход «I_ITL_ON» установлен, значение интегральной составляющей становится равным значению входа «I_ITL_VAL». |
| | DISV | REAL | 0 | Возмущающая переменная. Значение возмущающего воздействия добавляется к управляющему воздействию для упреждающего регулирования. |
| Output | LMN | REAL | | Выходное значение регулятора. Значение управляющего воздействия для подачи на объект управления. |
| | LMN_PER | WORD | | Выходное значение (периферия). Значение управляющего воздействия, пересчитанное в условные единицы для подачи напрямую на периферийный модуль аналогового вывода. $LMN_PER = (LMN/100)*27648$. Параметр используется только для платформы Siemens. |
| | QLMN_HLM | BOOL | | Верхний предел выхода достигнут. Выход устанавливается в «TRUE», если значение выхода «LMN» достигло верхнего предела «LMN_HLM». |
| | QLMN_LLM | BOOL | | Нижний предел выхода достигнут. Выход устанавливается в «TRUE», если значение выхода «LMN» достигло нижнего предела «LMN_LLM». |
| | LMN_P | REAL | | Пропорциональная составляющая. Расчетное значение пропорциональной составляющей. |
| | LMN_I | REAL | | Интегральная составляющая. Расчетное значение интегральной составляющей. |
| | LMN_D | REAL | | Дифференциальная составляющая. Расчетное значение дифференциальной составляющей. |
| | PV | REAL | | Входная переменная. Расчетное значение переменной процесса. Если вход «PVPER_ON» снят, то $PV = PV_IN$. Если вход «PVPER_ON» установлен, то $PV = (PV_PER/27648)*100*PV_FAC + PV_OFF$. |
| | ER | REAL | | Сигнал рассогласования. Вычисляется по следующим формулам: Если $(SP_INT - PV_IN) < (-DEADB_W)$, то $ER = (SP_INT - PV_IN) + DEADB_W$, Если $(SP_INT - PV_IN) > DEADB_W$, то $ER = (SP_INT - PV_IN) - DEADB_W$, иначе $ER = 0$. |

4.27.4 Принцип работы

Непрерывный ПИД-регулятор воздействует на управляемый объект или процесс выходной переменной «LMN», которая формируется на основе вычлененного рассогласования между заданным «SP_INT» и фактическим «PV_IN» значениями регулируемой переменной (переменной процесса).

Дополнительно предусмотрен вход возмущающего воздействия «DISV», который напрямую воздействует на выходную переменную LMN. Это позволяет сразу выйти на нужное предустановленное значение управляющего воздействия, а точный выход на задание уже отработать P, I, D-составляющими.

У регулятора также предусмотрен ручной режим, который позволяет управлять его выходным значением «LMN» напрямую. Для этого нужно установить вход «MAN_ON» в значение «TRUE», а на вход «MAN» подать желаемое выходное значение регулятора.

Если у регулируемой переменной в установившемся режиме имеются колебания относительно точки равновесия, то для того, чтобы исключить паразитные изменения управляющего воздействия, можно задать необходимое значение зоны нечувствительности «DEADB_W».

Для того, чтобы ограничить диапазон изменения выходного управляющего сигнала снизу и сверху, предусмотрены входы «LMN_LLM» и «LMN_HLM».

Также значение управляющего воздействия может быть смасштабировано по линейной характеристике с помощью коэффициентов «LMN_FAC» (множитель) и «LMN_OFFSET» (смещение).

Для контроля работы ПИД-регулятора пропорциональная, интегральная и дифференциальная составляющие, рассчитываемые внутри блока, выведены на выходы «LMN_P», «LMN_I», «LMN_D».

Также на выход блока выведены рассогласование «ER» и регулируемая переменная «PV».

4.27.5 Пример

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    dbi_pid:      RuDrive.CONT_C;      // экземпляр блока регулятора
    man_on:       BOOL;                // включить ручное задание выхода
    man:          REAL;                // значение ручного задания выхода
    pv:          REAL;                // значение регулируемой переменной
    sp:          REAL;                // задание регулируемой переменной
    lmn:         REAL;                // управляющее воздействие
END_VAR

// задание регулятора
sp := 10.0;

// вызов экземпляра блока регулятора
dbi_pid(
    MAN_ON:= man_on,
    CYCLE:=T#100MS,
    SP_INT:= sp,
    PV_IN:= pv,
    MAN:= man,
    GAIN:= 1.0,
    TI:= T#10S,
    TD:= T#10S,
    TM_LAG:= T#10S,
    DEADB_W:= 0.1,
    LMN=> lmn);
```

4.28 Шаговый ПИ-регулятор (CONT_S)

4.28.1 Назначение

Шаговый регулятор предназначен для управления технологическими процессами с помощью дискретных управляющих сигналов.

4.28.2 Функции

- масштабирование регулируемой переменной (переменной процесса).
- ручное управление выходными сигналами.
- внешнее возмущающее воздействие.

4.28.3 Таблица входов-выходов

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--|--|
| Input | COM_RST | BOOL | | Полный рестарт. Если вход установлен, производится сброс накопленных значений пропорциональной, интегральной составляющих, обнуление выхода регулятора. |
| | LMNR_HS | BOOL | | Верхняя граница сигнала обратной связи по положению. Если вход установлен, это означает, что исполнительный механизм достиг крайнего верхнего положения, дальнейшее движение вверх невозможно. |
| | LMNR_LS | BOOL | | Нижняя граница сигнала обратной связи по положению. Если вход установлен, это означает, что исполнительный механизм достиг крайнего нижнего положения, дальнейшее движение вниз невозможно. |
| | LMNS_ON | BOOL | | Включить ручное управление. ПИ-регулятор отключается, выходные дискретные сигналы управляются с помощью входов «LMNUP» и «LMNDN». |
| | LMNUP | BOOL | | Ручной сигнал управления «движение вверх» на исполнительный механизм. |
| | LMNDN | BOOL | | Ручной сигнал управления «движение вниз» на исполнительный механизм. |
| | PVPER_ON | BOOL | | Включить чтение входной переменной с периферии. Если значение переменной процесса должно считываться с модуля аналогового ввода (периферийного входа), то на вход «PV_PER» должно быть подано необработанное значение аналогового сигнала с периферийного входа, а вход «PVPER_ON» должен быть установлен. Параметр используется только для платформы Siemens. |
| | CYCLE | TIME | TIME#1s0ms | Время выполнения блока. Время между вызовами блока должно быть постоянным. Вход «CYCLE» задает время между вызовами блока. |
| | SP_INT | REAL | 0 | Внутреннее задание. Вход «Внутреннее значение уставки» служит для установки заданного значения регулятора. |
| | PV_IN | REAL | 0 | Входная переменная. Значение регулируемой переменной (переменной процесса). |
| PV_PER | WORD | 0 | Входная переменная (периферия). Этот вход предназначен для ввода значения переменной процесса напрямую с аналогового входа модуля ввода, если установлен вход «PVPER_ON». Значение с модуля ввода внутри блока преобразуется к диапазону значений 0...100 (%), а затем масштабируется с помощью коэффициентов «PV_FAC» и «PV_OFF». | |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--------------------|---|
| | | | | Значение переменной процесса рассчитывается по следующей формуле: $PV = (PV_PER/27648)*100*PV_FAC + PV_OFF$. Параметр используется только для платформы Siemens. |
| | GAIN | REAL | 2 | Коэффициент пропорциональности. Вход задает коэффициент усиления регулятора. |
| | TI | TIME | TIME#20s0ms | Время интегрирования. Определяет временную характеристику интегратора. |
| | DEADB_W | REAL | 0 | Ширина зоны нечувствительности. Если рассогласование по модулю меньше зоны нечувствительности ($ SP-PV < DEADB_W$), то оно считается равным нулю, и регулятор на него не реагирует, т.е. не изменяет управляющего воздействия. |
| | LMN_HLM | REAL | 100 | Верхняя граница обратной связи по положению (при наличии аналогового сигнала обратной связи). Положение исполнительного механизма, при котором перестает подаваться управляющий сигнал «движение вверх» («QLMNUP»). |
| | LMN_LLM | REAL | 0 | Нижняя граница обратной связи по положению (при наличии аналогового сигнала обратной связи). Положение исполнительного механизма, при котором перестает подаваться управляющий сигнал «движение вниз» («QLMNDN»). |
| | PV_FAC | REAL | 1 | Коэффициент для корректировки входной переменной (множитель). Используется для масштабирования входного сигнала, если задействован ввод регулируемой переменной с периферии (вход «PVPER_ON» установлен). Значение с периферийного аналогового входа «PV_PER», преобразованное к диапазону значений 0...100 (%), умножается на этот коэффициент. Параметр используется только для платформы Siemens. |
| | PV_OFF | REAL | 0 | Коэффициент для корректировки входной переменной (смещение). Используется для масштабирования входного сигнала, если задействован ввод регулируемой переменной с периферии (вход «PVPER_ON» установлен). Значение с периферийного аналогового входа «PV_PER», преобразованное к диапазону значений 0...100 (%) и умноженное на значение входа «PV_FAC», складывается с этим коэффициентом. Параметр используется только для платформы Siemens. |
| | PULSE_TM | TIME | TIME#3s0ms | Минимальное время импульса. Минимальная продолжительность импульса управляющих воздействий на дискретные выходы «QLMNUP» или «QLMNDN» при регулировании. |
| | BREAK_TM | TIME | TIME#3s0ms | Минимальное время паузы. Минимальное время паузы между импульсами управляющих воздействий на дискретные выходы «QLMNUP» или «QLMNDN» при регулировании. |
| | MTR_TM | TIME | TIME#30s0ms | Время хода исполнительного механизма. Время, требуемое исполнительному механизму для движения от крайнего нижнего до крайнего верхнего положения. |
| | DISV | REAL | 0 | Возмущающая переменная. Значение возмущающего воздействия добавляется к управляющему воздействию для упреждающего регулирования. |
| | LMNRS_ON | BOOL | | Включить обратную связь по положению. Если вход установлен, то при работе клапана учитывается значение положения исполнительного механизма, подаваемое на вход «LMNRSVAL». |
| | LMNRSVAL | REAL | | Обратная связь по положению исполнительного механизма. Обратная связь включается входом «LMNRS_ON». |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--------------------|---|
| Output | QLMNUP | BOOL | | Управляющий дискретный сигнал «движение вверх» на исполнительный механизм. |
| | QLMNDN | BOOL | | Управляющий дискретный сигнал «движение вниз» на исполнительный механизм. |
| | PV | REAL | | На выход выводится расчетное значение переменной процесса. Если вход «PVPER_ON» снят, то $PV = PV_IN$. Если вход «PVPER_ON» установлен, то $PV = (PV_PER/27648)*100*PV_FAC + PV_OFF$. |
| | ER | REAL | | На выход выводится сигнал рассогласования. Вычисляется по следующим формулам: Если $(SP_INT - PV_IN) < (-DEADB_W)$, то $ER = (SP_INT - PV_IN) + DEADB_W$, Если $(SP_INT - PV_IN) > DEADB_W$, то $ER = (SP_INT - PV_IN) - DEADB_W$, иначе $ER = 0$. |

4.28.4 Принцип работы

Шаговый ПИ-регулятор воздействует на управляемый объект или процесс выходными дискретными сигналами «QLMNUP», «QLMNDN», которые формируются на основе вычисленного рассогласования между заданным «SP_INT» и фактическим «PV_IN» значениями регулируемой переменной (переменной процесса).

Дополнительно предусмотрен вход возмущающего воздействия «DISV», который напрямую воздействует на выходную переменную LMN. Это позволяет сразу выйти на нужное предустановленное значение управляющего воздействия, а точный выход на задание уже отработать P, I-составляющими.

У регулятора также предусмотрен ручной режим, который позволяет управлять его выходными дискретными сигналами напрямую. Для этого нужно установить вход «LMNS_ON» в значение «TRUE», а на входы «LMNUP», «LMNDN» подавать команды на движение вверх и движение вниз исполнительного механизма.

Если у регулируемой переменной в установившемся режиме имеются колебания относительно точки равновесия, то для того, чтобы исключить формирование паразитных сигналов управляющих воздействий, можно задать необходимое значение зоны нечувствительности «DEADB_W».

Для того, чтобы ограничить диапазон изменения положения исполнительного механизма (при наличии сигнала обратной связи по положению) снизу и сверху, предусмотрены входы «LMN_LLM» и «LMN_HLM».

Для контроля работы блока на его выходы выведены значения рассогласования «ER» и регулируемой переменной «PV».

4.28.5 Пример

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    dbi_pid: RuDrive.CONT_S; // экземпляр блока регулятора
    man_on:  BOOL;           // включить ручное задание выхода
    pv:      REAL;          // значение регулируемой переменной
    sp:      REAL;          // задание регулируемой переменной
    lmnr_hs: BOOL;          // достигнуто верхнее положение исполнительного
механизма
    lmnr_ls: BOOL;          // достигнуто нижнее положение исполнительного механизма
    lmnup:   BOOL;          // команда в ручном режиме - движение вверх
    lmnndn:  BOOL;          // команда в ручном режиме - движение вниз
    qlmnup:  BOOL;          // сигнал управления - движение вверх
    qlmnndn: BOOL;          // сигнал управления - движение вниз
END_VAR
```

```
// вызов экземпляра блока регулятора
dbi_pid(
    CYCLE:=T#100MS,
    SP_INT:= sp,
    PV_IN:= pv,
    GAIN:= 1.0,
    TI:= T#10S,
    DEADB W:= 0.1,
    LMNS_ON:=man_on,
    LMNUP:= lmnup,
    LMNDN:= lmnndn,
    LMNR_HS:= lmnr_hs,
    LMNR_LS:= lmnr_ls,
    QLMNUP=> qlmnup,
    QLMNDN=> qlmndn);
```

4.29 Коммутация электродвигателя «Контактор->ПЧ» (fb_eng_cf), «Ячейка->ПЧ» (fb_eng_sf)

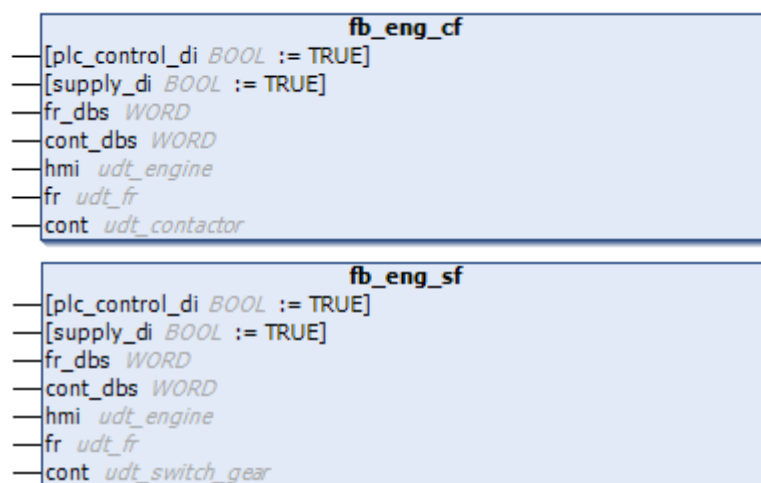
4.29.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации. Описание работы блока приведено для блока fb_eng_cf, но полностью справедливо для блока fb_eng_sf, где блок контактора (fb_contactor) заменен блоком ячейки (fb_switch_gear).

4.29.2 Функции

- запуск и останов электродвигателя с помощью преобразователя частоты, подключаемого к питающей сети контактором (ячейкой).
- возможность регулирования скорости электродвигателя с помощью ПИД-регулятора или вводом ручного задания.

4.29.3 Таблица входов-выходов



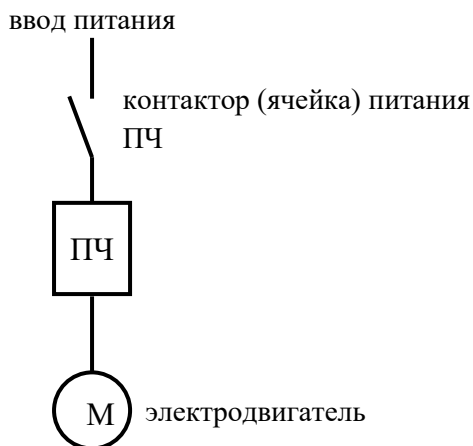
| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------|---------------|--------------------|---|
| Input | plc_control_di | BOOL | TRUE | состояние переключателя режима ручной/автомат (FALSE - ручной, TRUE - автомат) |
| | supply_di | BOOL | TRUE | наличие напряжения на вводе |
| | fr_dbs | WORD | | номера DB объектов fr и contactor (switch_gear). Используется только для платформы Siemens для корректного отображения данных на панели оператора. |
| | cont_dbs | WORD | | |
| Inout | fr | udt_fr | | интерфейс блока ПЧ |
| | cont | udt_contactor | | интерфейс блока контактора (ячейки) питания ПЧ |
| | hmi | udt_engine | | интерфейс блока |

4.29.4 Принцип работы

Блок включает и выключает электродвигатель, а также регулирует его скорость в процессе работы, управляя блоками контактора (cont) и ПЧ (fr).

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_contactor (контактор) и fb_fr (ПЧ).

Схема коммутации показана на рисунке ниже.



Блок переходит в состояние «готовность», когда блоки контактора и ПЧ находятся в состоянии готовности и в режиме «автомат». При подаче команды «пуск» блок сначала включает контактор питания ПЧ (в свою очередь, подавая команду «пуск» на блок управления контактором), а затем включает ПЧ (подавая команду «пуск» на блок управления ПЧ).

Регулировать частоту вращения электродвигателя можно как вручную, подавая фиксированное задание частоты (скорости) через интерфейсную структуру hmi, так и с помощью ПИД-регулятора, встроенного в блок ПЧ.

4.29.5 Интерфейс блока (udt_engine)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|--------------|----------------------|-------------|------------|--|
| std | <u>udt_standard</u> | | ввод-вывод | стандартный набор переменных |
| out_ac | REAL | | вывод | Ток двигателя. Значение передается с блока ПЧ. |
| out_freq | REAL | | вывод | Частота двигателя. Значение передается с блока ПЧ. |
| out_speed | REAL | | вывод | Частота вращения двигателя, об/мин. Расчетное значение на основе текущей частоты ПЧ (out_freq) и максимальной частоты вращения (max_speed): $out_speed = (out_freq/50)*max_speed$. |
| ready_net | WORD | | вывод | Готовность к пуску от сети. Не используется для данного блока. |
| ready_fr | WORD | | вывод | Готовность к пуску от ПЧ. Бит 0 – общая готовность от ПЧ. Если бит взведен, блок готов к пуску. Если нет, смотри причину в следующих битах. Бит 1 – есть питание контактора Бит 2 – контактор не в аварии Бит 3 – ПЧ не в аварии Бит 4 – нет сигнала аварии с ПЧ Бит 5 – контактор в режиме "Автомат" Бит 6 – ПЧ в режиме "Автомат" |
| pid | <u>udt_pid</u> | | ввод-вывод | Набор параметров регулятора. Если блок в работе, этот набор параметров передается в блок ПЧ. |
| ac_settings | <u>udt_engine_ac</u> | | ввод | Настройки токов для передачи в ПЧ. Если блок в работе, этот набор параметров передается в блок ПЧ. |
| type_connect | INT | | вывод | Тип подключения двигателя (=1024). Служебный параметр для использования блоком насосной станции (fb_pump_sau). |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|------|-------------|------------|--|
| freq_low | REAL | 5 | ввод | Минимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_hi | REAL | 50 | ввод | максимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_rising | REAL | 5 | ввод | Скорость нарастания частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_lowering | REAL | 5 | ввод | Скорость понижения частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_overload_lowering | REAL | 5 | ввод | Скорость понижения частоты при перегрузке по току. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_deadband | REAL | 1 | ввод | Допустимое отклонение частоты при разгоне и торможении. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_freq_min_wait | REAL | 10 | ввод | Время ожидания набора минимальной частоты, сек. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_supply | REAL | 10 | ввод | Время ожидания возвращения напряжения на вводе, сек. При пропадании напряжения на вводе (supply_di = FALSE) блок удерживает состояние «в работе». Если напряжение на вводе вернулось (supply_di = TRUE) в течение этого времени, осуществляется перезапуск контактора и ПЧ. Если напряжение не вернулось, блок переходит в состояние аварии. |
| time_ready_fr | REAL | 5 | ввод | Время ожидания готовности ПЧ, сек. Если сигнал готовности ПЧ не пришел в течение этого времени после включения контактора питания, блок переходит в состояние аварии. |
| time_stop_cont_fr | REAL | 60 | ввод | Время таймаута отключения ячейки питания ПЧ, сек. Задержка отключения контактора питания ПЧ после подачи команды останова на блок. |
| max_speed | INT | 3000 | ввод | Максимальная частота вращения двигателя, об/мин. Используется для расчета текущей частоты вращения out_speed. |

4.29.6 Переменные интерфейса блока. Списки значений

4.29.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_engine» п. 3.3.3.4. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|------------------|------|----------|--|
| stop | WORD | 16#1 | Стоп. |
| net_start | WORD | 16#2 | Не используется. |
| fr_start | WORD | 16#4 | Пуск. |
| reconnect_fr_net | WORD | 16#8 | Не используется. |
| stop_fast | WORD | 16#10 | Быстрый останов. Сначала останавливает ПЧ без плавного снижения частоты. На ПЧ сразу передается задание частоты = 0. Затем выключает контактор питания ПЧ. |
| stop_alarm | WORD | 16#20 | Аварийный останов. Подаются команды на отключение контактора и быстрый останов ПЧ одновременно. |

4.29.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_engine» п. 3.3.3.1. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|---|
| net_out_s1 | WORD | 16#1 | «Авария контактора(ячейки) сети». Не используется. |
| fr_in_s1 | WORD | 16#2 | «Авария контактора(ячейки) питания ПЧ». Блок контактора питания ПЧ перешел в состояние «авария». |
| fr_out_s1 | WORD | 16#4 | «Авария контактора(ячейки) ПЧ». Не используется. |
| fr | WORD | 16#8 | «Авария ПЧ». Блок ПЧ перешел в состояние «авария». |
| fr_no_supply | WORD | 16#10 | «Пропало напряжение на вводе». Вход supply_di отсутствовал в течение времени time_supply (см. интерфейс блока). |
| fr_not_ready | WORD | 16#20 | «Нет готовности ПЧ». Сигнал готовности ПЧ не пришел в течение времени time_ready_fr (см. интерфейс блока) после включения контактора питания ПЧ. |
| in_s1 | WORD | 16#40 | «Авария общего контактора(ячейки) питания». Не используется. |
| fr_busy | WORD | 16#80 | «ПЧ занят». Не используется. |

4.29.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_engine» п. 3.3.3.2. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------------|------|----------|--|
| net_out_s1_not_ready | WORD | 16#1 | «Нет готовности контактора(ячейки) сети». Не используется. |
| fr_in_s1_not_ready | WORD | 16#2 | «Нет готовности контактора(ячейки) питания ПЧ». Отсутствует состояние готовности блока контактора питания ПЧ. |
| fr_out_s1_not_ready | WORD | 16#4 | «Нет готовности контактора(ячейки) ПЧ». Не используется. |
| fr_not_ready | WORD | 16#8 | «Нет готовности ПЧ». Отсутствует состояние готовности блока ПЧ. |
| net_out_s1_not_auto | WORD | 16#10 | «Контактор(ячейка) сети не в режиме автомат». Не используется. |
| fr_in_s1_not_auto | WORD | 16#20 | «Контактор(ячейка) питания ПЧ не в режиме автомат». Режим блока контактора питания ПЧ – не «автомат». |
| fr_out_s1_not_auto | WORD | 16#40 | «Контактор(ячейка) ПЧ не в режиме автомат». Не используется. |
| fr_not_auto | WORD | 16#80 | «ПЧ не в режиме автомат». Режим блока ПЧ – не «автомат». |
| fr_no_supply | WORD | 16#100 | «Пропало напряжение на вводе». Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. интерфейс блока). |
| fr_wrn | WORD | 16#200 | «Предупреждение ПЧ». От блока ПЧ пришел сигнал «Предупреждение». |
| fr_alm | WORD | 16#400 | «Авария ПЧ». Блок ПЧ перешел в состояние аварии. |

4.29.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_cfg_engine» п. 3.3.3.3. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|----------------|------|----------|---|
| auto_reconnect | WORD | 16#1 | Автоматическое переключение ПЧ/УПП на сеть после разгона. Не используется. |

4.29.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_engine» п. 3.3.3.6. Возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|----------------|------|--------------------|--|
| not_ready | WORD | 16#1 | Ожидание готовности. Нет готовности одного из подчиненных блоков к пуску. |
| stopped | WORD | 16#2 | Готовность. Все подчиненные блоки готовы к пуску. |
| net_starting | WORD | 16#4 | Запуск от сети. Не используется. |
| net_started | WORD | 16#8 | В работе от сети. Не используется. |
| net_stopping | WORD | 16#10 | Останов от сети. Не используется. |
| fr_starting | WORD | 16#20 | Запуск от ПЧ. Идет процесс запуска электродвигателя. Включается контактор питания ПЧ, затем включается ПЧ. |
| fr_started | WORD | 16#40 | В работе от ПЧ. Электродвигатель в работе – контактор питания ПЧ включен, ПЧ в работе. |
| fr_stopping | WORD | 16#80 | Останов от ПЧ. Идет процесс останова электродвигателя. Отключается ПЧ с плавным снижением частоты до нуля, затем отключается контактор питания ПЧ. |
| fr_net | WORD | 16#100 | Переключение ПЧ->сеть. Не используется. |
| alarm | WORD | 16#200 | Авария. Один из подчиненных блоков перешел в состояние аварии или отключился во время работы. |
| net_stopped | WORD | 16#400 | Готовность к запуску от сети. Не используется. |
| fr_stopped | WORD | 16#800 | Готовность к запуску от ПЧ. Контактор питания ПЧ выключен и готов к пуску, ПЧ выключен и готов к пуску. |
| fr_no_supply | WORD | 16#1000 | Пропало напряжение на вводе. Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник ссылки не найден.). |
| alarm_stopping | WORD | 16#2000 | Аварийный останов. Подана команда аварийного останова. Отключаются ПЧ и контактор питания ПЧ. Команды на их отключение подаются одновременно. |

4.29.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_hmi_state_actual_engine» п. 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|---|
| power_off | WORD | 16#1 | Нет питания. ПЧ или контактор питания ПЧ находятся в состоянии «Нет питания» (std.out_state_actual = power_off). |
| stopped | WORD | 16#2 | Отключен. |

| Название | Тип | Начальное значение | Описание |
|----------|------|--------------------|---|
| | | | ПЧ и контактор питания ПЧ находятся в состоянии «Отключен» (std.out_state_actual = stopped). |
| started | WORD | 16#4 | Включен. ПЧ и контактор питания ПЧ находятся в состоянии «Включен» (std.out_state_actual = started). |

4.29.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока контактор
    dbs_contactor: RuDrive.udt_contactor;
    // объявление интерфейса блока ПЧ
    dbs_fr: RuDrive.udt_fr;
    // объявление интерфейса блока коммутации электродвигателя "Контактор->ПЧ"
    dbs_engine: RuDrive.udt_engine;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока контактор
    dbi_contactor: RuDrive.fb_contactor;
    // объявление экземпляра блока ПЧ
    dbi_fr: RuDrive.fb_fr;
    // объявление экземпляра блока коммутации электродвигателя "Контактор->ПЧ"
    dbi_engine: RuDrive.fb_eng_cf;
END_VAR

// вызов экземпляра блока контактор
dbi_contactor(
    supply_di := input_contactor_supply_di,
    plc_control_di := input_contactor_plc_control_di,
    fault_di := input_contactor_fault_di,
    on_di := input_contactor_on_di,
    hmi := dbs_contactor,
    on_do => output_contactor_on_do,
    off_do => output_contactor_off_do,
    reset_do => output_contactor_reset_do );

// вызов экземпляра блока ПЧ
dbi_fr(
    supply_di := input_fr_supply_di,
    ready_di := input_fr_ready_di,
    plc_control_di := input_fr_plc_control_di,
    fault_di := input_fr_fault_di,
    on_di := input_fr_on_di,
    hmi := dbs_fr,
    start_do => output_fr_start_do,
    stop_do => output_fr_stop_do,
    alm_reset_do => output_fr_alm_reset_do,
    ao => output_fr_ao);

// вызов экземпляра блока коммутации электродвигателя "Контактор->ПЧ"
dbi_engine(
    plc_control_di := input_engine_plc_control_di,
    supply_di := input_engine_supply_di,
    fr_dbs := 0,
    cont_dbs := 0,
    hmi := dbs_engine,
    fr := dbs_fr,
    cont := dbs_contactor );
```

4.30 Коммутация электродвигателя «Ячейка->УПП» (fb_eng_su)

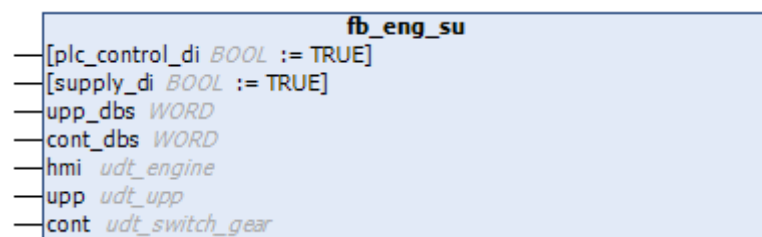
4.30.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации.

4.30.2 Функции

- запуск и останов электродвигателя с помощью УПП, подключаемого к питающей сети ячейкой.

4.30.3 Таблица входов-выходов



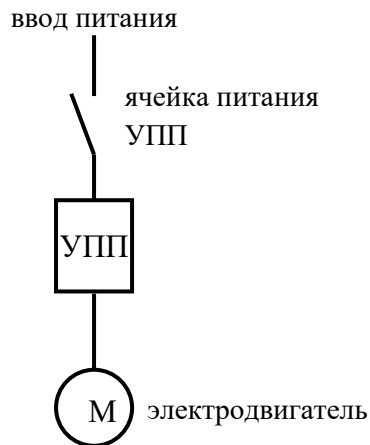
| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------|-----------------|--------------------|---|
| Input | plc_control_di | BOOL | TRUE | состояние переключателя режима ручной/автомат (FALSE - ручной, TRUE - автомат) |
| | supply_di | BOOL | TRUE | наличие напряжения на вводе |
| | upp_dbs | WORD | | номера DB объектов upp и switch_gear. |
| | cont_dbs | WORD | | Используется только для платформы Siemens для корректного отображения данных на панели оператора. |
| Inout | upp | udt_upp | | интерфейс блока ПЧ |
| | cont | udt_switch_gear | | интерфейс блока ячейки питания УПП |
| | hmi | udt_engine | | интерфейс блока |

4.30.4 Принцип работы

Блок включает и выключает электродвигатель, управляя блоками ячейки и УПП.

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_switch_gear (ячейка) и fb_upp (УПП).

Схема коммутации показана на рисунке ниже.



Блок переходит в состояние «готовность», когда блоки ячейки и УПП находятся в состоянии готовности и в режиме «автомат». При подаче команды «пуск» блок сначала включает ячейку питания УПП (в свою очередь, подавая команду «пуск» на блок управления ячейкой), а затем включает УПП (подавая команду «пуск» на блок управления УПП).

4.30.5 Интерфейс блока (udt_engine)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|----------------------|-------------|------------|---|
| std | <u>udt_standard</u> | | ввод-вывод | стандартный набор переменных |
| out_ac | REAL | | вывод | Ток двигателя. Значение передается с блока ПЧ. |
| out_freq | REAL | | вывод | Не используется. |
| out_speed | REAL | | вывод | Не используется. |
| ready_net | WORD | | вывод | Не используется. |
| ready_fr | WORD | | вывод | Готовность к пуску от УПП. Бит 0 – общая готовность от УПП. Если бит взведен, блок готов к пуску. Если нет, смотри причину в следующих битах. Бит 1 – есть питание ячейки Бит 2 – ячейка не в аварии Бит 3 – УПП не в аварии Бит 4 – ячейка в режиме "автомат" Бит 5 – УПП в режиме "автомат" |
| pid | <u>udt_pid</u> | | ввод-вывод | Не используется. |
| ac_settings | <u>udt_engine_ac</u> | | ввод | Не используется. |
| type_connect | INT | | вывод | Тип подключения двигателя. Служебный параметр для использования другими блоками. |
| freq_low | REAL | 5 | ввод | Не используется. |
| freq_hi | REAL | 50 | ввод | Не используется. |
| freq_rising | REAL | 5 | ввод | Не используется. |
| freq_lowering | REAL | 5 | ввод | Не используется. |
| freq_overload_lowering | REAL | 5 | ввод | Не используется. |
| freq_deadband | REAL | 1 | ввод | Не используется. |
| time_freq_min_wait | REAL | 10 | ввод | Не используется. |
| time_supply | REAL | 10 | ввод | Время ожидания возвращения напряжения на вводе, сек. При пропадании напряжения на вводе (supply_di = FALSE) блок удерживает состояние «в работе». Если напряжение на вводе вернулось (supply_di = |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-------------------|------|-------------|------------|--|
| | | | | TRUE) в течение этого времени, осуществляется перезапуск УПП (ячейка при остается включенной). Если напряжение не вернулось, блок переходит в состояние аварии. |
| time_ready_fr | REAL | 5 | ввод | Время ожидания готовности УПП, сек. Если сигнал готовности не пришел в течение этого времени после включения ячейки питания, блок переходит в состояние аварии. |
| time_stop_cont_fr | REAL | 60 | ввод | Время таймаута отключения ячейки питания УПП, сек. Задержка отключения ячейки питания УПП после подачи команды останова на блок. |
| max_speed | INT | 3000 | ввод | Не используется. |

4.30.6 Переменные интерфейса блока. Списки значений

4.30.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_engine» п. 3.3.3.4. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|------------------|------|----------|--|
| stop | WORD | 16#1 | Стоп. |
| net_start | WORD | 16#2 | Не используется. |
| fr_start | WORD | 16#4 | Пуск. |
| reconnect_fr_net | WORD | 16#8 | Не используется. |
| stop_fast | WORD | 16#10 | Быстрый останов. Сначала останавливает УПП. Затем выключает ячейку питания УПП. |
| stop_alarm | WORD | 16#20 | Аварийный останов. Подаются команды на отключение ячейки и останов УПП одновременно. |

4.30.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_engine» п. 3.3.3.1. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|--|
| net_out_s1 | WORD | 16#1 | «Авария контактора(ячейки) сети». Не используется. |
| fr_in_s1 | WORD | 16#2 | «Авария контактора(ячейки) питания УПП». Блок ячейки питания УПП перешел в состояние «авария». |
| fr_out_s1 | WORD | 16#4 | «Авария контактора(ячейки) УПП». Не используется. |
| fr | WORD | 16#8 | «Авария УПП». Блок УПП перешел в состояние «авария». |
| fr_no_supply | WORD | 16#10 | «Пропало напряжение на вводе». Вход «suprly_di» отсутствовал в течение времени «time_supply» (см. интерфейс блока). |
| fr_not_ready | WORD | 16#20 | «Нет готовности УПП». Сигнал готовности УПП не пришел в течение времени «time_ready_fr» (см. интерфейс блока) после включения ячейки питания УПП. |
| in_s1 | WORD | 16#40 | «Авария общего контактора(ячейки) питания». Не используется. |
| fr_busy | WORD | 16#80 | «УПП занят». Не используется. |

4.30.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_engine» п. 3.3.3.2. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------------|------|----------|--|
| net_out_s1_not_ready | WORD | 16#1 | «Нет готовности контактора(ячейки) сети». Не используется. |
| fr_in_s1_not_ready | WORD | 16#2 | «Нет готовности контактора(ячейки) питания УПП». Отсутствует состояние готовности блока ячейки питания УПП. |
| fr_out_s1_not_ready | WORD | 16#4 | «Нет готовности контактора(ячейки) УПП». Не используется. |
| fr_not_ready | WORD | 16#8 | «Нет готовности УПП». Отсутствует состояние готовности блока УПП. |
| net_out_s1_not_auto | WORD | 16#10 | «Контактор(ячейка) сети не в режиме автомат». Не используется. |
| fr_in_s1_not_auto | WORD | 16#20 | «Контактор(ячейка) питания УПП не в режиме автомат». Режим блока контактора питания УПП – не «автомат». |
| fr_out_s1_not_auto | WORD | 16#40 | «Контактор(ячейка) УПП не в режиме автомат». Не используется. |
| fr_not_auto | WORD | 16#80 | «УПП не в режиме автомат». Режим блока УПП – не «автомат». |
| fr_no_supply | WORD | 16#100 | «Пропало напряжение на вводе». Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. интерфейс блока). |
| fr_wrn | WORD | 16#200 | «Предупреждение УПП». Не используется. |
| fr_alm | WORD | 16#400 | «Авария УПП». Не используется. |

4.30.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_cfg_engine» п. 3.3.3.3. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|----------------|------|----------|---|
| auto_reconnect | WORD | 16#1 | Автоматическое переключение ПЧ/УПП на сеть после разгона. Не используется. |

4.30.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_engine» п. 3.3.3.6. Возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|--------------|------|--------------------|--|
| not_ready | WORD | 16#1 | Ожидание готовности. Нет готовности одного из подчиненных блоков к пуску. |
| stopped | WORD | 16#2 | Готовность. Все подчиненные блоки готовы к пуску. |
| net_starting | WORD | 16#4 | Запуск от сети. Не используется. |
| net_started | WORD | 16#8 | В работе от сети. Не используется. |
| net_stopping | WORD | 16#10 | Останов от сети. Не используется. |

| Название | Тип | Начальное значение | Описание |
|----------------|------|--------------------|--|
| fr_starting | WORD | 16#20 | Запуск от УПП. Идет процесс запуска электродвигателя. Включается ячейка питания УПП, затем включается УПП. |
| fr_started | WORD | 16#40 | В работе от УПП. Электродвигатель в работе – ячейка питания УПП включена, УПП в работе. |
| fr_stopping | WORD | 16#80 | Останов от УПП. Идет процесс останова электродвигателя. Отключается УПП, затем отключается ячейка питания УПП. |
| fr_net | WORD | 16#100 | Переключение УПП->сеть. Не используется. |
| alarm | WORD | 16#200 | Авария. Один из подчиненных блоков перешел в состояние аварии или отключился во время работы. |
| net_stopped | WORD | 16#400 | Готовность к запуску от сети. Не используется. |
| fr_stopped | WORD | 16#800 | Готовность к запуску от УПП. Ячейка питания УПП выключена и готова к пуску, УПП выключен и готов к пуску. |
| fr_no_supply | WORD | 16#1000 | Пропало напряжение на вводе. Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник ссылки не найден.). |
| alarm_stopping | WORD | 16#2000 | Аварийный останов. Подана команда аварийного останова. Отключаются УПП и ячейка питания УПП. Команды на их отключение подаются одновременно. |

4.30.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_hmi_state_actual_engine» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|--|
| power_off | WORD | 16#1 | Нет питания. УПП или ячейка питания УПП находятся в состоянии «Нет питания» (std.out_state_actual = power_off). |
| stopped | WORD | 16#2 | Отключен. УПП и ячейка питания УПП находятся в состоянии «Отключен» (std.out_state_actual = stopped). |
| started | WORD | 16#4 | Включен. УПП и ячейка питания УПП находятся в состоянии «Включен» (std.out_state_actual = started). |

4.30.6.7 Пример

Объявление глобальных переменных:

```

VAR_GLOBAL
  // объявление интерфейса блока ячейка
  dbs_switch_gear: RuDrive.udt_switch_gear;
  // объявление интерфейса блока УПП
  dbs_upp: RuDrive.udt_upp;
  // объявление интерфейса блока коммутации электродвигателя "Ячейка->УПП"
  dbs_engine: RuDrive.udt_engine;
END_VAR

```

Вызов в циклической подпрограмме (цикл 100 мсек):

```

PROGRAM PLC_PRG
VAR
  // объявление экземпляра блока ячейка

```

```

    dbi_switch_gear:    RuDrive.fb_switch_gear;
    // объявление экземпляра блока УПП
    dbi_upp:            RuDrive.fb_upp;
    // объявление экземпляра блока коммутации электродвигателя "Ячейка->УПП"
    dbi_engine:        RuDrive.fb_eng_su;
END_VAR

// вызов экземпляра блока ячейка
dbi_switch_gear(
    supply_di := input_switch_gear_supply_di,
    plc_control_di := input_switch_gear_plc_control_di,
    fault_di := input_switch_gear_fault_di,
    ready_di := input_switch_gear_ready_di,
    on_di := input_switch_gear_on_di,
    off_di := input_switch_gear_off_di,
    hmi := dbs_switch_gear,
    on_do => output_switch_gear_on_do,
    off_do => output_switch_gear_off_do );

// вызов экземпляра блока УПП
dbi_upp(
    supply_di := input_upp_supply_di,
    ready_di := input_upp_ready_di,
    plc_control_di := input_upp_plc_control_di,
    fault_di := input_upp_fault_di,
    on_di := input_upp_on_di,
    hmi := dbs_upp,
    on_do => output_upp_on_do,
    reset_do => output_upp_reset_do);

// вызов экземпляра блока коммутации электродвигателя "Ячейка->УПП"
dbi_engine(
    plc_control_di := input_engine_plc_control_di,
    supply_di := input_engine_supply_di,
    upp_dbs := 0,
    cont_dbs := 0,
    hmi := dbs_engine,
    upp := dbs_upp,
    cont := dbs_switch_gear );

```

4.31 Коммутация электродвигателя «Контактор + ПЧ->Контактор» (fb_eng_c_fc), «Ячейка + ПЧ->Ячейка» (fb_eng_s_fs)

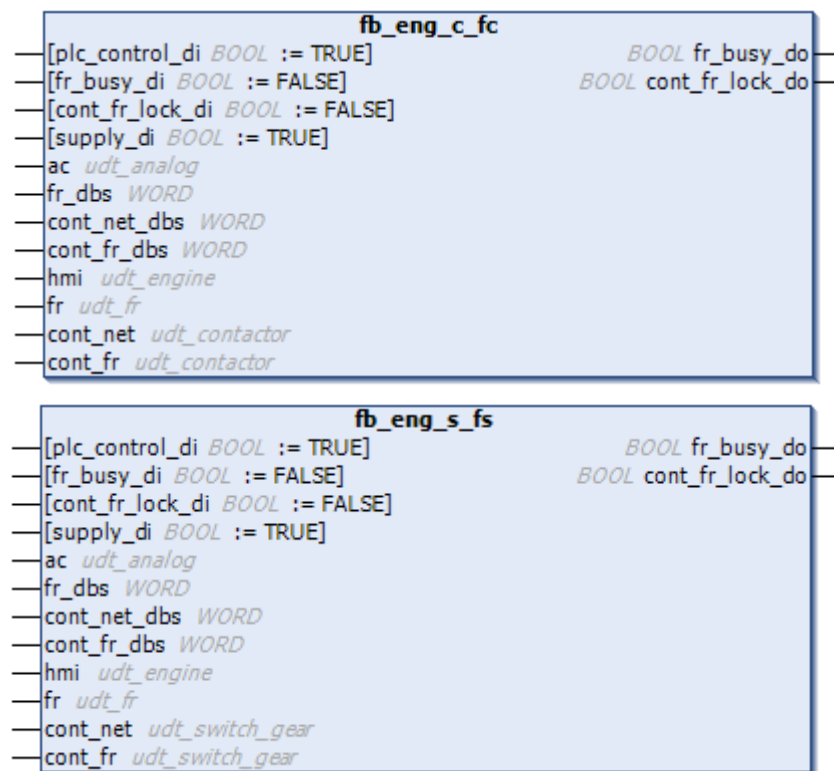
4.31.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации. Описание работы блока приведено для блока fb_eng_c_fc, но полностью справедливо для блока fb_eng_s_fs, где блок контактора (fb_contactor) заменен блоком ячейки (fb_switch_gear).

4.31.2 Функции

- запуск и останов электродвигателя с помощью преобразователя частоты и прямым пуском от сети.
- переключение с ПЧ на сеть.
- возможность регулирования скорости электродвигателя с помощью ПИД-регулятора или внешним заданием частоты.
- возобновление работы от ПЧ при пропадании и восстановлении питания.

4.31.3 Таблица входов-выходов

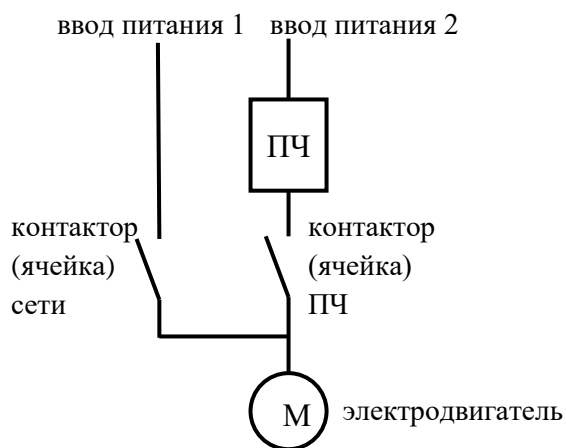


| Область | Название | Тип | Начальное значение | Назначение |
|---------|-----------------|------|--------------------|--|
| Input | plc_control_di | BOOL | TRUE | Состояние переключателя режима ручной/автомат (FALSE - ручной, TRUE - автомат) |
| | fr_busy_di | BOOL | FALSE | ПЧ занят (сигнал от других блоков) |
| | cont_fr_lock_di | BOOL | FALSE | Запрет включения контактора ПЧ в любом режиме (сигнал от других блоков) |
| | supply_di | BOOL | TRUE | Наличие напряжения на вводе питания ПЧ |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|-----------------|---------------|--------------------|---|
| | ac | udt_analog | | Ток двигателя при работе от сети. Вход для передачи интерфейса аналогового входа, который измеряет ток двигателя. |
| | fr_dbs | WORD | | Номера DB объектов. Используется только для платформы Siemens для корректного отображения данных на панели оператора. |
| | cont_net_dbs | WORD | | |
| | cont_fr_dbs | WORD | | |
| Inout | hmi | udt_engine | | интерфейс блока |
| | fr | udt_fr | | интерфейс блока ПЧ |
| | cont_net | udt_contactor | | интерфейс блока контактора сети |
| | cont_fr | udt_contactor | | интерфейс блока контактора ПЧ |
| Output | fr_busy_do | BOOL | FALSE | ПЧ занят (сигнал другим блокам) |
| | cont_fr_lock_do | BOOL | FALSE | Запрет включения контактора ПЧ в любом режиме (сигнал другим блокам) |

4.31.4 Принцип работы

Схема коммутации показана на рисунке.



Блок может выполнять следующие операции:

- запуск электродвигателя от сети (с помощью контактора сети);
- запуск электродвигателя от ПЧ (с помощью контактора ПЧ и ПЧ);
- переключение электродвигателя с ПЧ на сеть на ходу (во время работы от ПЧ);
- регулирование частоты вращения электродвигателя как вручную подачей фиксированного задания частоты через интерфейсную структуру hmi, так и с помощью ПИД-регулятора, встроенного в блок ПЧ;
- возобновление работы от ПЧ при пропадании и восстановлении питания.

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_contactor (контактор сети и контактор ПЧ) и fb_fr (ПЧ).

Готовность к пуску

Блок имеет несколько состояний готовности:

- «готовность к пуску от сети»;
- «готовность к пуску от ПЧ»;
- «готовность к пуску».

«Готовность к пуску от сети» означает, что контактор сети в состоянии готовности и в режиме «автомат», а также контактор ПЧ разомкнут. При этом другие объекты (ПЧ или контактор ПЧ) либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску от ПЧ» означает, что ПЧ и контактор ПЧ в состоянии готовности и в режиме «автомат», а также контактор сети разомкнут. При этом контактор сети либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску» означает, что все объекты (контактор сети, контактор ПЧ, ПЧ) находятся в состоянии готовности и в режиме «автомат».

Работа от сети

Для запуска электродвигателя от сети необходимо, чтобы блок был либо в состоянии «готовность к пуску от сети», либо «готовность к пуску».

При подаче команды «пуск от сети» (start_net) будет включен контактор сети, блок перейдет в состояние «работа от сети».

При подаче команды «стоп» (stop) контактор сети будет отключен, блок перейдет в состояние «остановлен».

Работа от ПЧ

Для запуска электродвигателя от ПЧ необходимо, чтобы блок был либо в состоянии «готовность к пуску от ПЧ», либо «готовность к пуску».

При подаче команды «пуск от ПЧ» (start_fr) будет включен контактор ПЧ, затем будет включен ПЧ, блок перейдет в состояние «работа от ПЧ».

При подаче команды «стоп» (stop) ПЧ будет плавно остановлен со снижением скорости до нуля, затем контактор ПЧ будет отключен, блок перейдет в состояние «остановлен».

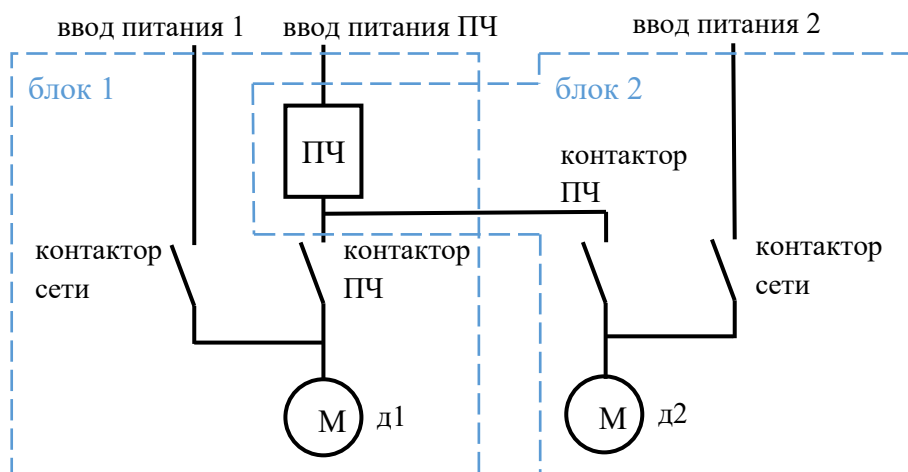
Переключение с ПЧ на сеть

Для переключения электродвигателя с ПЧ на сеть во время работы от ПЧ нужно подать команду «reconnect_net_fr». Контакттор ПЧ будет отключен (без выключения ПЧ), и сразу включен контактор сети. Электродвигатель продолжит работу от сети. После этого будет подана команда на останов ПЧ. Эта функция не рекомендуется к использованию, поскольку при переключении нет синхронизации фаз ПЧ и сети, переключение производится ударно.

Совместное использование нескольких блоков. Один ПЧ на несколько электродвигателей

На рисунке показана схема включения двух электродвигателей, которые могут каждый независимо включаться прямым пуском от сети, но имеют один общий ПЧ. Электродвигателей может быть и больше, чем два.

Пунктирной линией выделены объекты, которые относятся к каждому из блоков коммутации электродвигателей. В данном случае ПЧ общий на оба блока.



Для того, чтобы блоки коммутации могли корректно управлять разделяемым ресурсом – ПЧ, предусмотрены следующие входные и выходные сигналы:

- «fr_busy_di»: вход; сигнал от других блоков коммутации, что в данный момент ПЧ занят ими;
- «fr_busy_do»: выход; сигнал другим блокам коммутации, что в данный момент ПЧ занят данным блоком.
- «cont_fr_lock_di»: вход; сигнал от других блоков коммутации, что требуется запретить запуск контактора ПЧ, поскольку включен контактор ПЧ другого блока.

- «cont_fr_lock_do»: выход; сигнал другим блокам коммутации, чтобы запретили запуск своих контакторов ПЧ, поскольку включен контактор ПЧ данного блока.

Таким образом, при вызове блоков в программе нужно на вход «fr_busy_di» подать сигналы «fr_busy_do» от других блоков, и на вход «cont_fr_lock_di» подать сигналы «cont_fr_lock_do» от других блоков. Если ко входу нужно привязать более, чем один сигнал, объединить их логическим «или».

Возобновление работы электродвигателя от ПЧ при пропадании и восстановлении питания

Блок поддерживает функцию возобновления работы электродвигателя от ПЧ при пропадании и последующем восстановлении питания. Сигнал наличия напряжения на вводе питания ПЧ нужно привязать ко входу «supply_di» (сигнал может быть взят с реле контроля фаз, например).

Если во время работы двигателя от ПЧ питание пропало, а затем восстановилось в течение заданного времени, будет произведено повторное включение контактора ПЧ и запуск ПЧ.

4.31.5 Интерфейс блока (udt_engine)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|--------------|----------------------|-------------|------------|--|
| std | udt_standard | | ввод-вывод | стандартный набор переменных |
| out_ac | REAL | | вывод | Ток двигателя. Значение передается с блока ПЧ в состоянии «Работа от ПЧ». В состоянии «Работа от сети» значение передается со входа «ас». |
| out_freq | REAL | | вывод | Частота двигателя. Значение передается с блока ПЧ. |
| out_speed | REAL | | вывод | Частота вращения двигателя, об/мин. Расчетное значение на основе текущей частоты ПЧ (out_freq) и максимальной частоты вращения (max_speed): $out_speed = (out_freq/50)*max_speed$. |
| ready_net | WORD | | вывод | Готовность к пуску от сети. Бит 0 – общая готовность к пуску от сети. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – контактор сети отключен. Бит 2 – контактор ПЧ отключен. Бит 3 – контактор сети в режиме «автомат». |
| ready_fr | WORD | | вывод | Готовность к пуску от ПЧ. Бит 0 – общая готовность от ПЧ. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – контактор сети отключен. Бит 2 – контактор ПЧ отключен. Бит 3 – ПЧ отключен (готов к запуску). Бит 4 – контактор ПЧ в режиме «автомат». Бит 5 – ПЧ в режиме «автомат». Бит 6 – ПЧ не занят. Бит 7 – Нет запрета включения контактора ПЧ (от других блоков). |
| pid | <u>udt_pid</u> | | ввод-вывод | Набор параметров регулятора. Если блок в работе, этот набор параметров передается в блок ПЧ. |
| ac_settings | <u>udt_engine_ac</u> | | ввод | Настройки токов для передачи в ПЧ. Если блок в работе, этот набор параметров передается в блок ПЧ. |
| type_connect | INT | | вывод | Тип подключения двигателя. |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|------|-------------|------------|---|
| | | | | Служебный параметр для использования другими блоками. |
| freq_low | REAL | 5 | ввод | Минимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_hi | REAL | 50 | ввод | Максимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_rising | REAL | 5 | ввод | Скорость нарастания частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_lowering | REAL | 5 | ввод | Скорость понижения частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_overload_lowering | REAL | 5 | ввод | Скорость понижения частоты при перегрузке по току. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_deadband | REAL | 1 | ввод | Допустимое отклонение частоты при разгоне и торможении. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_freq_min_wait | REAL | 10 | ввод | Время ожидания набора минимальной частоты, сек. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_supply | REAL | 10 | ввод | Время ожидания возвращения напряжения на вводе, сек. При пропадании напряжения на вводе питания ПЧ (supply_di = FALSE) блок удерживает состояние «в работе». Если напряжение на вводе вернулось (supply_di = TRUE) в течение этого времени, осуществляется перезапуск контактора ПЧ и ПЧ. Если напряжение не вернулось, блок переходит в состояние аварии. |
| time_ready_fr | REAL | 5 | ввод | Время ожидания готовности ПЧ, сек. Не используется для данного блока. |
| time_stop_cont_fr | REAL | 60 | ввод | Время таймаута отключения ячейки питания ПЧ, сек. Не используется для данного блока. |
| max_speed | INT | 3000 | ввод | Максимальная частота вращения двигателя, об/мин. Используется для расчета текущей частоты вращения out_speed. |

4.31.6 Переменные интерфейса блока. Списки значений

4.31.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_engine» п. 3.3.3.4. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|------------------|------|----------|--|
| stop | WORD | 16#1 | Стоп. |
| net_start | WORD | 16#2 | Пуск от сети. |
| fr_start | WORD | 16#4 | Пуск от ПЧ. |
| reconnect_fr_net | WORD | 16#8 | Переключение с ПЧ на сеть. |
| stop_fast | WORD | 16#10 | Быстрый останов. Сначала останавливает ПЧ без плавного снижения частоты. На ПЧ сразу передается задание частоты = 0. Затем выключает контактор ПЧ. |
| stop_alarm | WORD | 16#20 | Аварийный останов. Подаются команды на отключение контактора ПЧ и быстрый останов ПЧ одновременно. |

4.31.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_engine» п. 3.3.3.1. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|---|
| net_out_s1 | WORD | 16#1 | «Авария контактора(ячейки) сети». Блок контактора сети перешел в состояние «авария». |
| fr_in_s1 | WORD | 16#2 | «Авария контактора(ячейки) питания ПЧ». Не используется. |
| fr_out_s1 | WORD | 16#4 | «Авария контактора(ячейки) ПЧ». Блок контактора ПЧ перешел в состояние «авария». |
| fr | WORD | 16#8 | «Авария ПЧ». Блок ПЧ перешел в состояние «авария». |
| fr_no_supply | WORD | 16#10 | «Пропало напряжение на вводе питания ПЧ». Вход supply_di отсутствовал в течение времени time_supply (см. интерфейс блока). |
| fr_not_ready | WORD | 16#20 | «Нет готовности ПЧ». Не используется. |
| in_s1 | WORD | 16#40 | «Авария общего контактора(ячейки) питания». Не используется. |
| fr_busy | WORD | 16#80 | «ПЧ занят». Не используется. |

4.31.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_engine» п. 3.3.3.2. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------------|------|----------|---|
| net_out_s1_not_ready | WORD | 16#1 | «Нет готовности контактора(ячейки) сети». Отсутствует состояние готовности блока контактора сети. |
| fr_in_s1_not_ready | WORD | 16#2 | «Нет готовности контактора(ячейки) питания ПЧ». Не используется. |
| fr_out_s1_not_ready | WORD | 16#4 | «Нет готовности контактора(ячейки) ПЧ». Отсутствует состояние готовности блока контактора ПЧ. |
| fr_not_ready | WORD | 16#8 | «Нет готовности ПЧ». Отсутствует состояние готовности блока ПЧ. |
| net_out_s1_not_auto | WORD | 16#10 | «Контактор(ячейка) сети не в режиме автомат». Режим блока контактора сети – не «автомат». |
| fr_in_s1_not_auto | WORD | 16#20 | «Контактор(ячейка) питания ПЧ не в режиме автомат». Не используется. |
| fr_out_s1_not_auto | WORD | 16#40 | «Контактор(ячейка) ПЧ не в режиме автомат». Режим блока контактора ПЧ – не «автомат». |
| fr_not_auto | WORD | 16#80 | «ПЧ не в режиме автомат». Режим блока ПЧ – не «автомат». |
| fr_no_supply | WORD | 16#100 | «Пропало напряжение на вводе питания ПЧ». Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник ссылки не найден.). |
| fr_wrn | WORD | 16#200 | «Предупреждение ПЧ». От блока ПЧ пришел сигнал «Предупреждение». |
| fr_alm | WORD | 16#400 | «Авария ПЧ». Блок ПЧ перешел в состояние аварии. |

4.31.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_cfg_engine» п. 3.3.3.3. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|----------------|------|----------|---|
| auto_reconnect | WORD | 16#1 | Автоматическое переключение ПЧ/УПП на сеть после разгона. Не используется. |

4.31.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_engine» п. 3.3.3.6. Возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|----------------|------|--------------------|---|
| not_ready | WORD | 16#1 | Ожидание готовности. Нет готовности одного из подчиненных блоков к пуску. |
| stopped | WORD | 16#2 | Готовность. Все подчиненные блоки готовы к пуску. |
| net_starting | WORD | 16#4 | Запуск от сети. Идет процесс запуска электродвигателя от сети. Включается контактор сети. |
| net_started | WORD | 16#8 | В работе от сети. Контактор сети включен, электродвигатель работает от сети. |
| net_stopping | WORD | 16#10 | Останов от сети. Идет процесс останова электродвигателя от сети. Отключается контактор сети. |
| fr_starting | WORD | 16#20 | Запуск от ПЧ. Идет процесс запуска электродвигателя от ПЧ. Включается контактор ПЧ, затем включается ПЧ. |
| fr_started | WORD | 16#40 | В работе от ПЧ. Электродвигатель в работе – контактор ПЧ включен, ПЧ в работе. |
| fr_stopping | WORD | 16#80 | Останов от ПЧ. Идет процесс останова электродвигателя от ПЧ. Отключается ПЧ с плавным снижением частоты до нуля, затем отключается контактор ПЧ. |
| fr_net | WORD | 16#100 | Переключение ПЧ->сеть. Идет процесс переключения электродвигателя с ПЧ на сеть. Отключается контактор ПЧ, включается контактор сети. Отключается ПЧ. Затем блок переходит в состояние «В работе от сети». |
| alarm | WORD | 16#200 | Авария. Один из подчиненных блоков перешел в состояние аварии или отключился во время работы. |
| net_stopped | WORD | 16#400 | Готовность к запуску от сети. Контактор сети отключен и готов к пуску в режиме «автомат». Контактор ПЧ отключен. |
| fr_stopped | WORD | 16#800 | Готовность к запуску от ПЧ. Контактор питания ПЧ выключен и готов к пуску, ПЧ выключен и готов к пуску. |
| fr_no_supply | WORD | 16#1000 | Пропало напряжение на вводе. Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник ссылки не найден.). |
| alarm_stopping | WORD | 16#2000 | Аварийный останов. Подана команда аварийного останова. Отключаются ПЧ и контактор ПЧ при работе двигателя от ПЧ (команды на их отключение подаются одновременно) или контактор сети при работе двигателя от сети. |

4.31.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_hmi_state_actual_engine» п. 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|--------------|
| power_off | WORD | 16#1 | Нет питания. |

| Название | Тип | Начальное значение | Описание |
|----------|------|--------------------|--|
| | | | ПЧ или контактор ПЧ находятся в состоянии «Нет питания» (std.out_state_actual = power_off), а также контактор сети находится в состоянии «Нет питания» (std.out_state_actual = power_off). |
| stopped | WORD | 16#2 | Отключен. ПЧ и контактор ПЧ находятся в состоянии «Отключен» (std.out_state_actual = stopped) или контактор сети находится в состоянии «Отключен» (std.out_state_actual = stopped). |
| started | WORD | 16#4 | Включен. ПЧ и контактор ПЧ находятся в состоянии «Включен» (std.out_state_actual = started) или контактор сети в состоянии «Включен» (std.out_state_actual = started). |

4.31.7 Пример 1. Один электродвигатель.

Объявление глобальных переменных:

VAR_GLOBAL

```
// объявление интерфейса блока контактор сети
dbs_cont_net: RuDrive.udt_contactor;
// объявление интерфейса блока контактор ПЧ
dbs_cont_fr: RuDrive.udt_contactor;
// объявление интерфейса блока ПЧ
dbs_fr: RuDrive.udt_fr;
// объявление интерфейса блока коммутации электродвигателя "Контактор + ПЧ-
>Контактор"
dbs_engine: RuDrive.udt_engine;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM PLC_PRG

VAR

```
// объявление экземпляра блока контактор сети
dbi_cont_net: RuDrive.fb_contactor;
// объявление экземпляра блока контактор ПЧ
dbi_cont_fr: RuDrive.fb_contactor;
// объявление экземпляра блока ПЧ
dbi_fr: RuDrive.fb_fr;
// объявление экземпляра блока коммутации электродвигателя "Контактор + ПЧ-
>Контактор"
dbi_engine: RuDrive.fb_eng_c_fc;
END_VAR
```

// вызов экземпляра блока контактор ПЧ

```
dbi_cont_fr(
  supply_di := input_cont_fr_supply_di,
  plc_control_di := input_cont_fr_plc_control_di,
  fault_di := input_cont_fr_fault_di,
  on_di := input_cont_fr_on_di,
  hmi := dbs_cont_fr,
  on_do => output_cont_fr_on_do,
  off_do => output_cont_fr_off_do,
  reset_do => output_cont_fr_reset_do );
```

// вызов экземпляра блока контактор сети

```
dbi_cont_net(
  supply_di := input_cont_net_supply_di,
  plc_control_di := input_cont_net_plc_control_di,
  fault_di := input_cont_net_fault_di,
  on_di := input_cont_net_on_di,
  hmi := dbs_cont_net,
  on_do => output_cont_net_on_do,
  off_do => output_cont_net_off_do,
```

```

        reset_do => output_cont_net_reset_do );

// вызов экземпляра блока ПЧ
dbi_fr(
    supply_di := input_fr_supply_di,
    ready_di := input_fr_ready_di,
    plc_control_di := input_fr_plc_control_di,
    fault_di := input_fr_fault_di,
    on_di := input_fr_on_di,
    hmi := dbs_fr,
    start_do => output_fr_start_do,
    stop_do => output_fr_stop_do,
    alm_reset_do => output_fr_alm_reset_do,
    ao => output_fr_ao);

// вызов экземпляра блока коммутации электродвигателя "Контактор + ПЧ->Контактор"
dbi_engine(
    plc_control_di := input_engine_plc_control_di,
    supply_di := input_engine_supply_di,
    hmi := dbs_engine,
    fr := dbs_fr,
    cont_net := dbs_cont_net,
    cont_fr := dbs_cont_fr );

```

4.31.8 Пример 2. Два электродвигателя с одним общим ПЧ.

Объявление глобальных переменных:

```

VAR_GLOBAL
    // объявление экземпляра блока контактор ПЧ
    dbs_cont_net_1: RuDrive.udt_contactor;
    dbs_cont_net_2: RuDrive.udt_contactor;
    // объявление экземпляра блока контактор сети
    dbs_cont_fr_1: RuDrive.udt_contactor;
    dbs_cont_fr_2: RuDrive.udt_contactor;
    // объявление экземпляра блока ПЧ
    dbs_fr: RuDrive.udt_fr;
    // объявление экземпляра блока коммутации электродвигателя "Контактор + ПЧ-
    >Контактор"
    dbs_engine_1: RuDrive.udt_engine;
    dbs_engine_2: RuDrive.udt_engine;
END_VAR

```

Вызов в циклической подпрограмме (цикл 100 мсек):

```

PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока контактор ПЧ
    dbi_cont_net_1: RuDrive.fb_contactor;
    dbi_cont_net_2: RuDrive.fb_contactor;
    // объявление экземпляра блока контактор сети
    dbi_cont_fr_1: RuDrive.fb_contactor;
    dbi_cont_fr_2: RuDrive.fb_contactor;
    // объявление экземпляра блока ПЧ
    dbi_fr: RuDrive.fb_fr;
    // объявление экземпляра блока коммутации электродвигателя "Контактор + ПЧ-
    >Контактор"
    dbi_engine_1: RuDrive.fb_eng_c_fc;
    dbi_engine_2: RuDrive.fb_eng_c_fc;
END_VAR

// вызов экземпляра блока контактор ПЧ
dbi_cont_fr_1(
    supply_di := input_cont_fr_1_supply_di,
    plc_control_di := input_cont_fr_1_plc_control_di,
    fault_di := input_cont_fr_1_fault_di,
    on_di := input_cont_fr_1_on_di,
    hmi := dbs_cont_fr_1,
    on_do => output_cont_fr_1_on_do,

```

```

    off_do => output_cont_fr_1_off_do,
    reset_do => output_cont_fr_1_reset_do );

dbi_cont_fr_2(
    supply_di := input_cont_fr_2_supply_di,
    plc_control_di := input_cont_fr_2_plc_control_di,
    fault_di := input_cont_fr_2_fault_di,
    on_di := input_cont_fr_2_on_di,
    hmi := dbs_cont_fr_2,
    on_do => output_cont_fr_2_on_do,
    off_do => output_cont_fr_2_off_do,
    reset_do => output_cont_fr_2_reset_do );

// вызов экземпляра блока контактор сети
dbi_cont_net_1(
    supply_di := input_cont_net_1_supply_di,
    plc_control_di := input_cont_net_1_plc_control_di,
    fault_di := input_cont_net_1_fault_di,
    on_di := input_cont_net_1_on_di,
    hmi := dbs_cont_net_1,
    on_do => output_cont_net_1_on_do,
    off_do => output_cont_net_1_off_do,
    reset_do => output_cont_net_1_reset_do );

dbi_cont_net_2(
    supply_di := input_cont_net_2_supply_di,
    plc_control_di := input_cont_net_2_plc_control_di,
    fault_di := input_cont_net_2_fault_di,
    on_di := input_cont_net_2_on_di,
    hmi := dbs_cont_net_2,
    on_do => output_cont_net_2_on_do,
    off_do => output_cont_net_2_off_do,
    reset_do => output_cont_net_2_reset_do );

// вызов экземпляра блока ПЧ
dbi_fr(
    supply_di := input_fr_supply_di,
    ready_di := input_fr_ready_di,
    plc_control_di := input_fr_plc_control_di,
    fault_di := input_fr_fault_di,
    on_di := input_fr_on_di,
    hmi := dbs_fr,
    start_do => output_fr_start_do,
    stop_do => output_fr_stop_do,
    alm_reset_do => output_fr_alm_reset_do,
    ao => output_fr_ao );

// вызов экземпляра блока коммутации электродвигателя "Контактор + ПЧ->Контактор"
dbi_engine_1(
    plc_control_di := input_engine_1_plc_control_di,
    supply_di := input_engine_1_supply_di,
    fr_busy_di := dbi_engine_2.fr_busy_do,
    cont_fr_lock_di := dbi_engine_2.cont_fr_lock_do,
    hmi := dbs_engine_1,
    fr := dbs_fr,
    cont_net := dbs_cont_net_1,
    cont_fr := dbs_cont_fr_1 );

dbi_engine_2(
    plc_control_di := input_engine_2_plc_control_di,
    supply_di := input_engine_2_supply_di,
    fr_busy_di := dbi_engine_1.fr_busy_do,
    cont_fr_lock_di := dbi_engine_1.cont_fr_lock_do,
    hmi := dbs_engine_2,
    fr := dbs_fr,
    cont_net := dbs_cont_net_2,
    cont_fr := dbs_cont_fr_2 );

```


4.32 Коммутация электродвигателя «Контактор + УПП->Контактор» (fb_eng_c_uc)

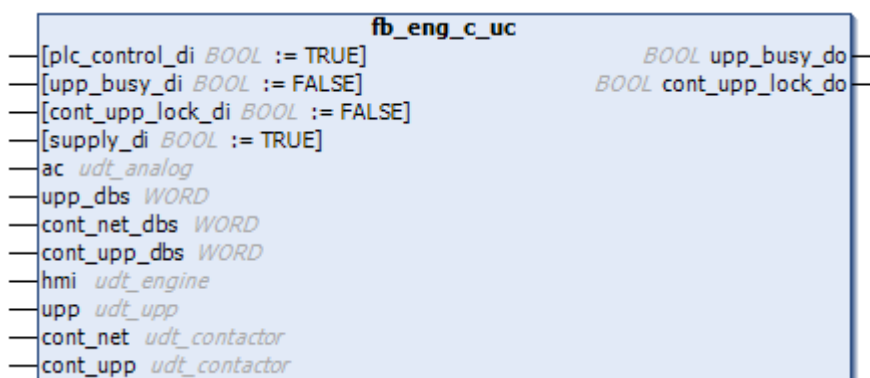
4.32.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации.

4.32.2 Функции

- запуск и останов электродвигателя с помощью устройства плавного пуска (УПП) и прямым пуском от сети.
- переключение с УПП на сеть.
- возобновление работы от УПП при пропадании и восстановлении питания.

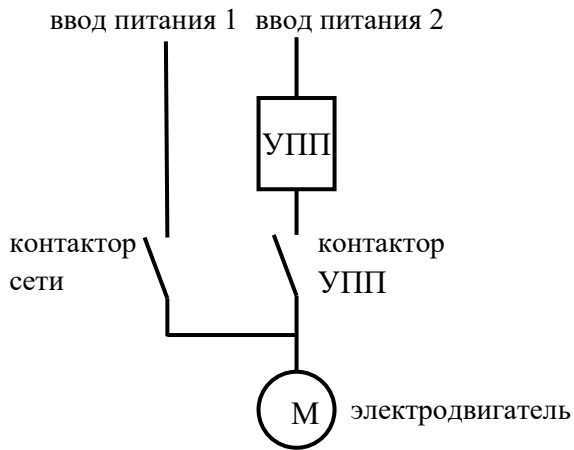
4.32.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|-------------|------------------|---------------|--------------------|--|
| Input | plc_control_di | BOOL | TRUE | Состояние переключателя режима ручной/автомат (FALSE - ручной, TRUE - автомат) |
| | upp_busy_di | BOOL | FALSE | УПП занят (сигнал от других блоков) |
| | cont_upp_lock_di | BOOL | FALSE | Запрет включения контактора УПП в любом режиме (сигнал от других блоков) |
| | supply_di | BOOL | TRUE | Наличие напряжения на вводе питания УПП |
| | ac | udt_analog | | Ток двигателя при работе от сети. Вход для передачи экземпляра интерфейса блока аналогового входа, который измеряет ток двигателя. |
| | upp_dbs | WORD | | Номера DB объектов. Используется только для платформы Siemens для корректного отображения данных на панели оператора. |
| | cont_net_dbs | WORD | | |
| cont_fr_dbs | WORD | | | |
| Inout | hmi | udt_engine | | интерфейс блока |
| | upp | udt_upp | | интерфейс блока УПП |
| | cont_net | udt_contactor | | интерфейс блока контактора сети |
| | cont_fr | udt_contactor | | интерфейс блока контактора УПП |
| Output | upp_busy_do | BOOL | FALSE | УПП занят (сигнал другим блокам) |
| | cont_upp_lock_do | BOOL | FALSE | Запрет включения контактора УПП в любом режиме (сигнал другим блокам) |

4.32.4 Принцип работы

Схема коммутации показана на рисунке.



Блок может выполнять следующие операции:

- запуск электродвигателя от сети (с помощью контактора сети);
- запуск электродвигателя от УПП (с помощью контактора УПП и УПП);
- переключение электродвигателя с УПП на сеть (после разгона УПП);
- возобновление работы от УПП при пропадании и восстановлении питания.

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_contactor (контактор сети и контактор УПП) и fb_fr (ПЧ).

Готовность к пуску

Блок имеет несколько состояний готовности:

- «готовность к пуску от сети»;
- «готовность к пуску от УПП»;
- «готовность к пуску».

«Готовность к пуску от сети» означает, что контактор сети в состоянии готовности и в режиме «автомат», а также контактор УПП разомкнут. При этом другие объекты (УПП или контактор УПП) либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску от УПП» означает, что УПП и контактор УПП в состоянии готовности и в режиме «автомат», а также контактор сети разомкнут. При этом контактор сети либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску» означает, что все объекты (контактор сети, контактор УПП, УПП) находятся в состоянии готовности и в режиме «автомат».

Работа от сети

Для запуска электродвигателя от сети необходимо, чтобы блок был либо в состоянии «готовность к пуску от сети», либо «готовность к пуску».

При подаче команды «пуск от сети» (start_net) будет включен контактор сети, блок перейдет в состояние «работа от сети».

При подаче команды «стоп» (stop) контактор сети будет отключен, блок перейдет в состояние «остановлен».

Работа от УПП

Для запуска электродвигателя от УПП необходимо, чтобы блок был либо в состоянии «готовность к пуску от УПП», либо «готовность к пуску».

При подаче команды «пуск от УПП» (start_fr) будет включен контактор УПП, затем будет включен УПП, блок перейдет в состояние «работа от УПП».

При подаче команды «стоп» (stop) УПП будет плавно остановлен со снижением скорости до нуля, затем контактор УПП будет отключен, блок перейдет в состояние «остановлен».

Переключение с УПП на сеть

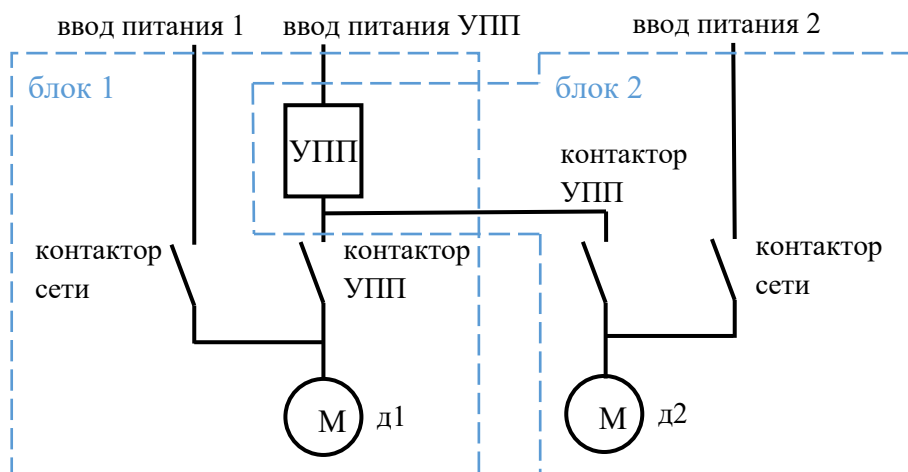
Для переключения электродвигателя с УПП на сеть во время работы от УПП нужно подать команду «reconnect_net_fr». Контакттор УПП будет отключен (без выключения УПП), и сразу включен контакттор сети. Электродвигатель продолжит работу от сети. После этого будет подана команда на останов УПП.

Желательно, чтобы для питания контакттора сети и УПП использовался один ввод питания, тогда переключение на сеть будет безударным.

Совместное использование нескольких блоков. Один УПП на несколько электродвигателей

На рисунке показана схема включения двух электродвигателей, которые могут каждый независимо включаться прямым пуском от сети, но имеют один общий УПП. Электродвигателей может быть и больше, чем два.

Пунктирной линией выделены объекты, которые относятся к каждому из блоков коммутации электродвигателей. В данном случае УПП общий на оба блока.



Для того, чтобы блоки коммутации могли корректно управлять разделяемым ресурсом – УПП, предусмотрены следующие входные и выходные сигналы:

- «urr_busy_di»: вход; сигнал от других блоков коммутации, что в данный момент УПП занят ими;
- «urr_busy_do»: выход; сигнал другим блокам коммутации, что в данный момент УПП занят данным блоком.
- «cont_urr_lock_di»: вход; сигнал от других блоков коммутации, что требуется запретить запуск контакттора УПП, поскольку включен контакттор УПП другого блока.
- «cont_urr_lock_do»: выход; сигнал другим блокам коммутации, чтобы запретили запуск своих контактторов УПП, поскольку включен контакттор УПП данного блока.

Таким образом, при вызове блоков в программе нужно на вход «urr_busy_di» подать сигналы «urr_busy_do» от других блоков, и на вход «cont_urr_lock_di» подать сигналы «cont_urr_lock_do» от других блоков. Если ко входу нужно привязать более, чем один сигнал, их нужно объединить логическим «или».

Возобновление работы электродвигателя от УПП при пропадании и восстановлении питания

Блок поддерживает функцию возобновления работы электродвигателя от УПП при пропадании и последующем восстановлении питания. Сигнал наличия напряжения на вводе питания УПП нужно привязать ко входу «supply_di» (сигнал может быть взят с реле контроля фаз, например).

Если во время работы двигателя от УПП питание пропало, а затем восстановилось в течение заданного времени, будет произведено повторное включение контакттора УПП и запуск УПП.

4.32.5 Интерфейс блока (udt_engine)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|----------------------|-------------|------------|--|
| std | <u>udt_standard</u> | | ввод-вывод | стандартный набор переменных |
| out_ac | REAL | | вывод | Ток двигателя. Значение передается со входа «ас». |
| out_freq | REAL | | вывод | Частота двигателя. Выводится фиксированное значение 50 Гц, когда электродвигатель в работе от сети или УПП, иначе выводится 0. |
| out_speed | REAL | | вывод | Частота вращения двигателя, об/мин. Расчетное значение на основе текущей частоты (out_freq) и максимальной частоты вращения (max_speed): $out_speed = (out_freq/50)*max_speed$. |
| ready_net | WORD | | вывод | Готовность к пуску от сети. Бит 0 – общая готовность к пуску от сети. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – контактор сети отключен. Бит 2 – контактор УПП отключен. Бит 3 – контактор сети в режиме «автомат». |
| ready_fr | WORD | | вывод | Готовность к пуску от УПП. Бит 0 – общая готовность от УПП. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – контактор сети отключен. Бит 2 – контактор УПП отключен. Бит 3 – УПП отключен (готов к запуску). Бит 4 – контактор УПП в режиме «автомат». Бит 5 – УПП в режиме «автомат». Бит 6 – УПП не занят. Бит 7 – Нет запрета включения контактора УПП (от других блоков). |
| pid | <u>udt_pid</u> | | ввод-вывод | Не используется. |
| ac_settings | <u>udt_engine_ac</u> | | ввод | Не используется. |
| type_connect | INT | | вывод | Тип подключения двигателя. Служебный параметр для использования другими блоками. |
| freq_low | REAL | 5 | ввод | Не используется. |
| freq_hi | REAL | 50 | ввод | Не используется. |
| freq_rising | REAL | 5 | ввод | Не используется. |
| freq_lowering | REAL | 5 | ввод | Не используется. |
| freq_overload_lowering | REAL | 5 | ввод | Не используется. |
| freq_deadband | REAL | 1 | ввод | Не используется. |
| time_freq_min_wait | REAL | 10 | ввод | Не используется. |
| time_supply | REAL | 10 | ввод | Время ожидания возвращения напряжения на вводе, сек. При пропадании напряжения на вводе питания УПП (supply_di = FALSE) блок удерживает состояние «в работе». Если напряжение на вводе вернулось (supply_di = TRUE) в течение этого времени, осуществляется перезапуск контактора УПП и УПП. Если напряжение не вернулось, блок переходит в состояние аварии. |
| time_ready_fr | REAL | 5 | ввод | Не используется. |
| time_stop_cont_fr | REAL | 60 | ввод | Не используется. |
| max_speed | INT | 3000 | ввод | Максимальная частота вращения двигателя, об/мин. Используется для расчета текущей частоты вращения out_speed. |

4.32.6 Переменные интерфейса блока. Списки значений

4.32.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_engine» п. 3.3.3.4. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|------------------|------|----------|---|
| stop | WORD | 16#1 | Стоп. |
| net_start | WORD | 16#2 | Пуск от сети. |
| fr_start | WORD | 16#4 | Пуск от УПП. |
| reconnect_fr_net | WORD | 16#8 | Переключение с УПП на сеть. |
| stop_fast | WORD | 16#10 | Быстрый останов. Сначала отключает УПП. Затем отключает контактор УПП. |
| stop_alarm | WORD | 16#20 | Аварийный останов. Подаются команды на отключение контактора УПП и отключение УПП одновременно. |

4.32.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_engine» п. 3.3.3.1. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|---|
| net_out_s1 | WORD | 16#1 | «Авария контактора(ячейки) сети». Блок контактора сети перешел в состояние «авария». |
| fr_in_s1 | WORD | 16#2 | «Авария контактора(ячейки) питания УПП». Не используется. |
| fr_out_s1 | WORD | 16#4 | «Авария контактора(ячейки) УПП». Блок контактора УПП перешел в состояние «авария». |
| fr | WORD | 16#8 | «Авария УПП». Блок УПП перешел в состояние «авария». |
| fr_no_supply | WORD | 16#10 | «Пропало напряжение на вводе питания УПП». Вход supply_di отсутствовал в течение времени time_supply (см. интерфейс блока). |
| fr_not_ready | WORD | 16#20 | «Нет готовности УПП». |
| in_s1 | WORD | 16#40 | «Авария общего контактора(ячейки) питания». Не используется. |
| fr_busy | WORD | 16#80 | «УПП занят». Сообщение возникает при попытке включения контактора УПП (или если контактор УПП уже включен), если с других блоков приходит сигнал запрета его включения (cont_urr_lock_di = TRUE). |

4.32.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_engine» п. 3.3.3.2. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------------|------|----------|---|
| net_out_s1_not_ready | WORD | 16#1 | «Нет готовности контактора(ячейки) сети». Отсутствует состояние готовности блока контактора сети. |
| fr_in_s1_not_ready | WORD | 16#2 | «Нет готовности контактора(ячейки) питания УПП». Не используется. |
| fr_out_s1_not_ready | WORD | 16#4 | «Нет готовности контактора(ячейки) УПП». Отсутствует состояние готовности блока контактора УПП. |
| fr_not_ready | WORD | 16#8 | «Нет готовности УПП». Отсутствует состояние готовности блока УПП. |

| Название | Тип | Значение | Текст сообщения/Описание |
|---------------------|------|----------|---|
| net_out_s1_not_auto | WORD | 16#10 | «Контактор(ячейка) сети не в режиме автомат». Режим блока контактора сети – не «автомат». |
| fr_in_s1_not_auto | WORD | 16#20 | «Контактор(ячейка) питания УПП не в режиме автомат». Не используется. |
| fr_out_s1_not_auto | WORD | 16#40 | «Контактор(ячейка) УПП не в режиме автомат». Режим блока контактора УПП – не «автомат». |
| fr_not_auto | WORD | 16#80 | «УПП не в режиме автомат». Режим блока УПП – не «автомат». |
| fr_no_supply | WORD | 16#100 | «Пропало напряжение на вводе питания УПП». Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник с ссылки не найден.). |
| fr_wrn | WORD | 16#200 | «Предупреждение УПП». От блока УПП пришел сигнал «Предупреждение». |
| fr_alm | WORD | 16#400 | «Авария УПП». Блок УПП перешел в состояние аварии. |

4.32.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_cfg_engine» п. 3.3.3.3. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|----------------|------|----------|---|
| auto_reconnect | WORD | 16#1 | Автоматическое переключение УПП на сеть после разгона. Если флаг включен, то после разгона УПП (прихода сигнала «работа» от УПП) будет инициировано переключение с УПП на сеть. |

4.32.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_engine» п. 3.3.3.6. Возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|--------------|------|--------------------|---|
| not_ready | WORD | 16#1 | Ожидание готовности. Нет готовности одного из подчиненных блоков к пуску. |
| stopped | WORD | 16#2 | Готовность. Все подчиненные блоки готовы к пуску. |
| net_starting | WORD | 16#4 | Запуск от сети. Идет процесс запуска электродвигателя от сети. Включается контактор сети. |
| net_started | WORD | 16#8 | В работе от сети. Контактор сети включен, электродвигатель работает от сети. |
| net_stopping | WORD | 16#10 | Останов от сети. Идет процесс останова электродвигателя от сети. Отключается контактор сети. |
| fr_starting | WORD | 16#20 | Запуск от УПП. Идет процесс запуска электродвигателя от УПП. Включается контактор УПП, затем включается УПП. |
| fr_started | WORD | 16#40 | В работе от УПП. Электродвигатель в работе – контактор УПП включен, УПП в работе. |
| fr_stopping | WORD | 16#80 | Останов от УПП. Идет процесс останова электродвигателя от УПП. Отключается УПП, затем отключается контактор УПП. |
| fr_net | WORD | 16#100 | Переключение УПП->сеть. Идет процесс переключения электродвигателя с УПП на сеть. Отключается контактор УПП, включается контактор сети. Отключается УПП. Затем блок переходит в состояние «В работе от сети». |
| alarm | WORD | 16#200 | Авария. |

| Название | Тип | Начальное значение | Описание |
|----------------|------|--------------------|--|
| | | | Один из подчиненных блоков перешел в состояние аварии или отключился во время работы. |
| net_stopped | WORD | 16#400 | Готовность к запуску от сети. Контактор сети отключен и готов к пуску в режиме «автомат». Контактор УПП отключен. |
| fr_stopped | WORD | 16#800 | Готовность к запуску от УПП. Контактор питания УПП выключен и готов к пуску, УПП выключен и готов к пуску. |
| fr_no_supply | WORD | 16#1000 | Пропало напряжение на вводе. Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник ссылки не найден.). |
| alarm_stopping | WORD | 16#2000 | Аварийный останов. Подана команда аварийного останова. Отключаются УПП и контактор УПП при работе двигателя от УПП (команды на их отключение подаются одновременно) или контактор сети при работе двигателя от сети. |

4.32.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_hmi_state_actual_engine» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|--|
| power_off | WORD | 16#1 | Нет питания. УПП или контактор УПП находятся в состоянии «Нет питания» (std.out_state_actual = power_off), а также контактор сети находится в состоянии «Нет питания» (std.out_state_actual = power_off). |
| stopped | WORD | 16#2 | Отключен. УПП и контактор УПП находятся в состоянии «Отключен» (std.out_state_actual = stopped) или контактор сети находится в состоянии «Отключен» (std.out_state_actual = stopped). |
| started | WORD | 16#4 | Включен. УПП и контактор УПП находятся в состоянии «Включен» (std.out_state_actual = started) или контактор сети в состоянии «Включен» (std.out_state_actual = started). |

4.32.7 Пример 1. Один электродвигатель.

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока контактор сети
    dbs_cont_net:      RuDrive.udt_contactor;
    // объявление интерфейса блока контактор УПП
    dbs_cont_upp:     RuDrive.udt_contactor;
    // объявление интерфейса блока УПП
    dbs_upp:          RuDrive.udt_upp;
    // объявление интерфейса блока коммутации электродвигателя "Контактор + УПП-
    >Контактор"
    dbs_engine:      RuDrive.udt_engine;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока контактор сети
```

```

dbi_cont_net:      RuDrive.fb_contactor;
// объявление экземпляра блока контактор УПП
dbi_cont_upp:     RuDrive.fb_contactor;
// объявление экземпляра блока УПП
dbi_upp:          RuDrive.fb_upp;
// объявление экземпляра блока коммутации электродвигателя "Контактор + УПП-
>Контактор"
dbi_engine:       RuDrive.fb_eng_c_uc;
END_VAR

// вызов экземпляра блока контактор УПП
dbi_cont_upp(
  supply_di := input_cont_upp_supply_di,
  plc_control_di := input_cont_upp_plc_control_di,
  fault_di := input_cont_upp_fault_di,
  on_di := input_cont_upp_on_di,
  hmi := dbs_cont_upp,
  on_do => output_cont_upp_on_do,
  off_do => output_cont_upp_off_do,
  reset_do => output_cont_upp_reset_do );

// вызов экземпляра блока контактор сети
dbi_cont_net(
  supply_di := input_cont_net_supply_di,
  plc_control_di := input_cont_net_plc_control_di,
  fault_di := input_cont_net_fault_di,
  on_di := input_cont_net_on_di,
  hmi := dbs_cont_net,
  on_do => output_cont_net_on_do,
  off_do => output_cont_net_off_do,
  reset_do => output_cont_net_reset_do );

// вызов экземпляра блока УПП
dbi_upp(
  supply_di := input_upp_supply_di,
  ready_di := input_upp_ready_di,
  plc_control_di := input_upp_plc_control_di,
  fault_di := input_upp_fault_di,
  on_di := input_upp_on_di,
  hmi := dbs_upp,
  on_do => output_upp_on_do,
  reset_do => output_upp_reset_do);

// вызов экземпляра блока коммутации электродвигателя "Контактор + УПП->Контактор"
dbi_engine(
  plc_control_di := input_engine_plc_control_di,
  supply_di := input_engine_supply_di,
  hmi := dbs_engine,
  upp := dbs_upp,
  cont_net := dbs_cont_net,
  cont_upp := dbs_cont_upp );

```

4.32.8 Пример 2. Два электродвигателя с одним общим УПП.

Объявление глобальных переменных:

```

VAR_GLOBAL
  // объявление экземпляра блока контактор сети
  dbs_cont_net_1: RuDrive.udt_contactor;
  dbs_cont_net_2: RuDrive.udt_contactor;
  // объявление экземпляра блока контактор УПП
  dbs_cont_upp_1: RuDrive.udt_contactor;
  dbs_cont_upp_2: RuDrive.udt_contactor;
  // объявление экземпляра блока УПП
  dbs_upp: RuDrive.udt_upp;
  // объявление экземпляра блока коммутации электродвигателя "Контактор + УПП-
>Контактор"
  dbs_engine_1: RuDrive.udt_engine;
  dbs_engine_2: RuDrive.udt_engine;

```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM PLC_PRG

VAR

```
// объявление экземпляра блока контактор сети
dbi_cont_net_1:   RuDrive.fb_contactor;
dbi_cont_net_2:   RuDrive.fb_contactor;
// объявление экземпляра блока контактор УПП
dbi_cont_upp_1:   RuDrive.fb_contactor;
dbi_cont_upp_2:   RuDrive.fb_contactor;
// объявление экземпляра блока УПП
dbi_fr:           RuDrive.fb_fr;
// объявление экземпляра блока коммутации электродвигателя "Контактор + УПП-
>Контактор"
dbi_engine_1:     RuDrive.fb_eng_c_fc;
dbi_engine_2:     RuDrive.fb_eng_c_fc;
```

END_VAR

// вызов экземпляра блока контактор УПП

```
dbi_cont_upp_1(
  supply_di := input_cont_upp_1_supply_di,
  plc_control_di := input_cont_upp_1_plc_control_di,
  fault_di := input_cont_upp_1_fault_di,
  on_di := input_cont_upp_1_on_di,
  hmi := dbs_cont_upp_1,
  on_do => output_cont_upp_1_on_do,
  off_do => output_cont_upp_1_off_do,
  reset_do => output_cont_upp_1_reset_do );
```

```
dbi_cont_upp_2(
  supply_di := input_cont_upp_2_supply_di,
  plc_control_di := input_cont_upp_2_plc_control_di,
  fault_di := input_cont_upp_2_fault_di,
  on_di := input_cont_upp_2_on_di,
  hmi := dbs_cont_upp_2,
  on_do => output_cont_upp_2_on_do,
  off_do => output_cont_upp_2_off_do,
  reset_do => output_cont_upp_2_reset_do );
```

// вызов экземпляра блока контактор сети

```
dbi_cont_net_1(
  supply_di := input_cont_net_1_supply_di,
  plc_control_di := input_cont_net_1_plc_control_di,
  fault_di := input_cont_net_1_fault_di,
  on_di := input_cont_net_1_on_di,
  hmi := dbs_cont_net_1,
  on_do => output_cont_net_1_on_do,
  off_do => output_cont_net_1_off_do,
  reset_do => output_cont_net_1_reset_do );
```

```
dbi_cont_net_2(
  supply_di := input_cont_net_2_supply_di,
  plc_control_di := input_cont_net_2_plc_control_di,
  fault_di := input_cont_net_2_fault_di,
  on_di := input_cont_net_2_on_di,
  hmi := dbs_cont_net_2,
  on_do => output_cont_net_2_on_do,
  off_do => output_cont_net_2_off_do,
  reset_do => output_cont_net_2_reset_do );
```

// вызов экземпляра блока УПП

```
dbi_upp(
  supply_di := input_upp_supply_di,
  ready_di := input_upp_ready_di,
  plc_control_di := input_upp_plc_control_di,
  fault_di := input_upp_fault_di,
  on_di := input_upp_on_di,
  hmi := dbs_upp,
  on_do => output_upp_on_do,
  reset_do => output_upp_reset_do);
```

```
// вызов экземпляра блока коммутации электродвигателя "Контактор + УПП->Контактор"
dbi_engine_1(
    plc_control_di := input_engine_1_plc_control_di,
    supply_di := input_engine_1_supply_di,
    fr_busy_di := dbi_engine_2.fr_busy_do,
    cont_upp_lock_di := dbi_engine_2.cont_upp_lock_do,
    hmi := dbs_engine_1,
    fr := dbs_fr,
    cont_net := dbs_cont_net_1,
    cont_upp := dbs_cont_upp_1 );

dbi_engine_2(
    plc_control_di := input_engine_2_plc_control_di,
    supply_di := input_engine_2_supply_di,
    fr_busy_di := dbi_engine_1.fr_busy_do,
    cont_upp_lock_di := dbi_engine_1.cont_upp_lock_do,
    hmi := dbs_engine_2,
    fr := dbs_fr,
    cont_net := dbs_cont_net_2,
    cont_upp := dbs_cont_upp_2 );
```

4.33 Коммутация электродвигателя «Ячейка + Ячейка->ПЧ->Ячейка» (fb_eng_s_sfs), «Ячейка + Ячейка->ПЧ->Контактора» (fb_eng_s_sfc)

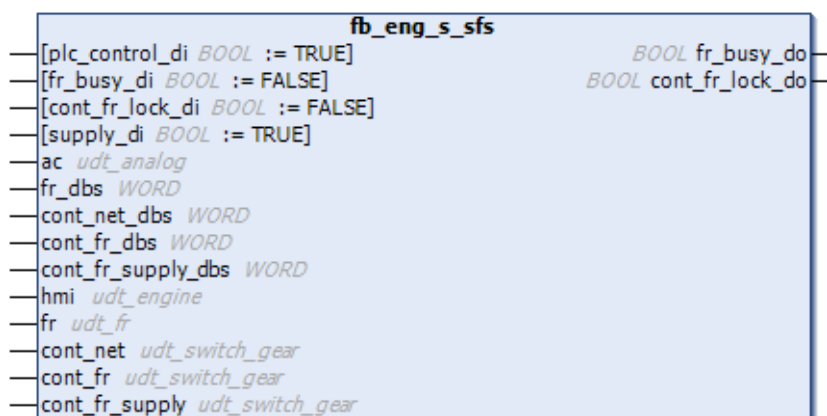
4.33.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации. Описание работы блока приведено для блока fb_eng_s_sfs, но полностью справедливо для блока fb_eng_s_sfc, где блок ячейки ПЧ заменен блоком контактора (fb_contactor).

4.33.2 Функции

- запуск и останов электродвигателя с помощью преобразователя частоты и прямым пуском от сети.
- переключение с ПЧ на сеть.
- возможность регулирования скорости электродвигателя с помощью ПИД-регулятора или внешним заданием частоты.
- возобновление работы от ПЧ при пропадании и восстановлении питания.

4.33.3 Таблица входов-выходов

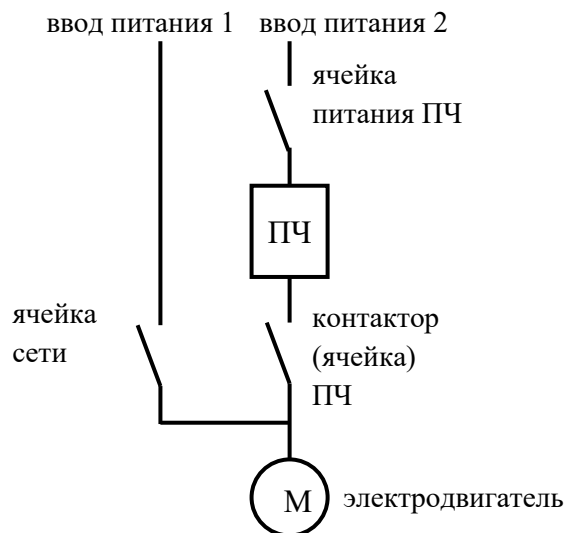


| Область | Название | Тип | Начальное значение | Назначение |
|--------------------|-----------------|-----------------|--------------------|---|
| Input | plc_control_di | BOOL | TRUE | Состояние переключателя режима ручной/автомат (FALSE - ручной, TRUE - автомат) |
| | fr_busy_di | BOOL | FALSE | ПЧ занят (сигнал от других блоков) |
| | cont_fr_lock_di | BOOL | FALSE | Запрет включения контактора ПЧ в любом режиме (сигнал от других блоков) |
| | supply_di | BOOL | TRUE | Наличие напряжения на вводе питания ПЧ |
| | ac | udt_analog | | Ток двигателя при работе от сети. Вход для передачи интерфейса аналогового входа, который измеряет ток двигателя. |
| | fr_dbs | WORD | | Номера DB объектов. Используется только для платформы Siemens для корректного отображения данных на панели оператора. |
| | cont_net_dbs | WORD | | |
| | cont_fr_dbs | WORD | | |
| cont_fr_supply_dbs | WORD | | | |
| Inout | hmi | udt_engine | | интерфейс блока |
| | fr | udt_fr | | интерфейс блока ПЧ |
| | cont_net | udt_switch_gear | | интерфейс блока ячейки сети |
| | cont_fr | udt_switch_gear | | интерфейс блока ячейки (контактора) ПЧ |
| | cont_fr_supply | udt_switch_gear | | интерфейс блока ячейки питания ПЧ |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|-----------------|------|--------------------|--|
| Output | fr_busy_do | BOOL | FALSE | ПЧ занят (сигнал другим блокам) |
| | cont_fr_lock_do | BOOL | FALSE | Запрет включения контактора ПЧ в любом режиме (сигнал другим блокам) |

4.33.4 Принцип работы

Схема коммутации показана на рисунке.



Блок может выполнять следующие операции:

- запуск электродвигателя от сети (с помощью ячейки сети);
- запуск электродвигателя от ПЧ (с помощью ячейки питания ПЧ, ПЧ и ячейки ПЧ);
- переключение электродвигателя с ПЧ на сеть на ходу (во время работы от ПЧ);
- регулирование частоты вращения электродвигателя как вручную подачей фиксированного задания частоты через интерфейсную структуру hmi, так и с помощью ПИД-регулятора, встроенного в блок ПЧ;
- возобновление работы от ПЧ при пропадании и восстановлении питания.

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_switch_gear (ячейка сети, ячейка питания ПЧ, ячейка ПЧ) и fb_fr (ПЧ).

Готовность к пуску

Блок имеет несколько состояний готовности:

- «готовность к пуску от сети»;
- «готовность к пуску от ПЧ»;
- «готовность к пуску».

«Готовность к пуску от сети» означает, что ячейка сети в состоянии готовности и в режиме «автомат», а также ячейка ПЧ отключена. При этом другие объекты (ячейка питания ПЧ, ПЧ или ячейка ПЧ) либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску от ПЧ» означает, ячейка питания ПЧ, ПЧ и ячейка ПЧ в состоянии готовности и в режиме «автомат», а также ячейка сети отключена. При этом ячейка сети либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску» означает, что все объекты (ячейка сети, ячейка питания ПЧ, ПЧ, ячейка ПЧ) находятся в состоянии готовности и в режиме «автомат».

Работа от сети

Для запуска электродвигателя от сети необходимо, чтобы блок был либо в состоянии «готовность к пуску от сети», либо «готовность к пуску».

При подаче команды «пуск от сети» (start_net) будет включена ячейка сети, блок перейдет в состояние «работа от сети».

При подаче команды «стоп» (stop) ячейка сети будет отключена, блок перейдет в состояние «остановлен».

Работа от ПЧ

Для запуска электродвигателя от ПЧ необходимо, чтобы блок был либо в состоянии «готовность к пуску от ПЧ», либо «готовность к пуску».

При подаче команды «пуск от ПЧ» (start_fr) будут выполнены следующие действия:

- включение ячейки питания ПЧ;
- ожидание готовности ПЧ;
- включение ячейки ПЧ (подключение электродвигателя к выходу ПЧ);
- включение ПЧ.

Блок перейдет в состояние «работа от ПЧ».

При подаче команды «стоп» (stop) будут выполнены следующие действия:

- плавный останов ПЧ со снижением скорости вращения электродвигателя до нуля;
- отключение ячейки ПЧ (отключение электродвигателя от выхода ПЧ);
- отключение ячейки питания ПЧ.

Блок перейдет в состояние «готовность к пуску от ПЧ» либо «готовность к пуску».

Переключение с ПЧ на сеть

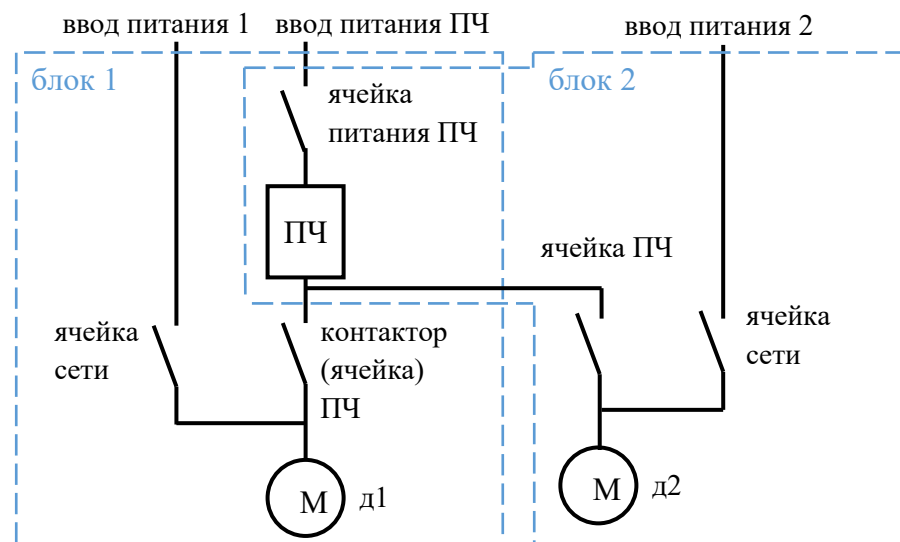
Для переключения электродвигателя с ПЧ на сеть во время работы от ПЧ нужно подать команду «reconnect_fr_net». Ячейка ПЧ будет отключена (без выключения ПЧ), и сразу включена ячейка сети. Электродвигатель продолжит работу от сети. После этого будет подана команда на останов ПЧ.

Эта функция не рекомендуется к использованию, поскольку при переключении нет синхронизации фаз напряжения ПЧ и сети, переключение производится ударно.

Совместное использование нескольких блоков. Один ПЧ на несколько электродвигателей

На рисунке показана схема включения двух электродвигателей, которые могут каждый независимо включаться прямым пуском от сети, но имеют один общий ПЧ и его ячейку питания. Электродвигателей может быть и больше, чем два.

Пунктирной линией выделены объекты, которые относятся к каждому из блоков коммутации электродвигателей. В данном случае ячейка питания ПЧ и ПЧ общие на оба блока.



Для того, чтобы блоки коммутации могли корректно управлять разделяемыми ресурсами – ПЧ и его ячейкой питания,

предусмотрены следующие входные и выходные сигналы:

- «fr_busy_di»: вход; сигнал от других блоков коммутации, что в данный момент ПЧ и ячейка питания ПЧ заняты ими;

- «fr_busy_do»: выход; сигнал другим блокам коммутации, что в данный момент ПЧ и ячейка питания ПЧ заняты данным блоком.

- «cont_fr_lock_di»: вход; сигнал от других блоков коммутации, что требуется запретить запуск ячейки ПЧ, поскольку включена ячейка ПЧ другого блока.

- «cont_fr_lock_do»: выход; сигнал другим блокам коммутации, чтобы запретили запуск своих ячеек ПЧ, поскольку включена ячейка ПЧ данного блока.

Таким образом, при вызове блоков в программе нужно на вход «fr_busy_di» подать сигналы «fr_busy_do» от других блоков, и на вход «cont_fr_lock_di» подать сигналы «cont_fr_lock_do» от других блоков. Если ко входу нужно привязать более, чем один сигнал, объединить их логическим «или».

Возобновление работы электродвигателя от ПЧ при пропадании и восстановлении питания

Блок поддерживает функцию возобновления работы электродвигателя от ПЧ при пропадании и последующем восстановлении питания. Сигнал наличия напряжения на вводе питания ПЧ нужно привязать ко входу «suprply_di» (сигнал может быть взят с реле контроля фаз, например).

Если во время работы двигателя от ПЧ питание пропало, а затем восстановилось в течение заданного времени, будет произведено повторное включение ячейки (контактора) ПЧ и запуск ПЧ.

4.33.5 Интерфейс блока (udt_engine)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-----------|---------------------|-------------|------------|--|
| std | <u>udt_standard</u> | | ввод-вывод | стандартный набор переменных |
| out_ac | REAL | | вывод | Ток двигателя. Значение передается с блока ПЧ в состоянии «Работа от ПЧ». В состоянии «Работа от сети» значение передается со входа «ас». |
| out_freq | REAL | | вывод | Частота двигателя. Значение передается с блока ПЧ. |
| out_speed | REAL | | вывод | Частота вращения двигателя, об/мин. Расчетное значение на основе текущей частоты ПЧ (out_freq) и максимальной частоты вращения (max_speed): $out_speed = (out_freq/50)*max_speed$. |
| ready_net | WORD | | вывод | Готовность к пуску от сети. Бит 0 – общая готовность к пуску от сети. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – ячейка сети отключена. Бит 2 – ячейка ПЧ отключена. Бит 3 – ячейка сети в режиме «автомат». |
| ready_fr | WORD | | вывод | Готовность к пуску от ПЧ. Бит 0 – общая готовность от ПЧ. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – ячейка сети отключена. Бит 2 – ячейка ПЧ отключена. Бит 3 – ячейка ПЧ в режиме «автомат». Бит 4 – ПЧ в режиме «автомат». Бит 5 – ПЧ не занят. Бит 6 – Нет запрета включения ячейки ПЧ (от других блоков). Бит 7 – Ячейка питания ПЧ готова (отключена или включена) |
| pid | <u>udt_pid</u> | | ввод-вывод | Набор параметров регулятора. Если блок в работе, этот набор параметров передается в блок ПЧ. |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|----------------------|-------------|------------|--|
| ac_settings | <u>udt_engine_ac</u> | | ввод | Настройки токов для передачи в ПЧ. Если блок в работе, этот набор параметров передается в блок ПЧ. |
| type_connect | INT | | вывод | Тип подключения двигателя. Служебный параметр для использования другими блоками. |
| freq_low | REAL | 5 | ввод | Минимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_hi | REAL | 50 | ввод | Максимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_rising | REAL | 5 | ввод | Скорость нарастания частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_lowering | REAL | 5 | ввод | Скорость понижения частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_overload_lowering | REAL | 5 | ввод | Скорость понижения частоты при перегрузке по току. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_deadband | REAL | 1 | ввод | Допустимое отклонение частоты при разгоне и торможении. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_freq_min_wait | REAL | 10 | ввод | Время ожидания набора минимальной частоты, сек. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_supply | REAL | 10 | ввод | Время ожидания возвращения напряжения на вводе, сек. При пропадании напряжения на вводе питания ПЧ (supply_di = FALSE) блок удерживает состояние «в работе». Если напряжение на вводе вернулось (supply_di = TRUE) в течение этого времени, осуществляется перезапуск ячейки ПЧ и ПЧ. Если напряжение не вернулось, блок переходит в состояние аварии. |
| time_ready_fr | REAL | 5 | ввод | Время ожидания готовности ПЧ, сек. Если сигнал готовности ПЧ не пришел в течение этого времени после включения ячейки питания, блок переходит в состояние аварии. |
| time_stop_cont_fr | REAL | 60 | ввод | Время таймаута отключения ячейки питания ПЧ, сек. Задержка отключения ячейки питания ПЧ после подачи команды останова на блок. |
| max_speed | INT | 3000 | ввод | Максимальная частота вращения двигателя, об/мин. Используется для расчета текущей частоты вращения out_speed. |

4.33.6 Переменные интерфейса блока. Списки значений

4.33.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_engine» п. 3.3.3.4. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|------------------|------|----------|---|
| stop | WORD | 16#1 | Стоп. |
| net_start | WORD | 16#2 | Пуск от сети. |
| fr_start | WORD | 16#4 | Пуск от ПЧ. |
| reconnect_fr_net | WORD | 16#8 | Переключение с ПЧ на сеть. |
| stop_fast | WORD | 16#10 | Быстрый останов. Сначала останавливает ПЧ без плавного снижения частоты. На ПЧ сразу передается задание |

| Название | Тип | Значение | Описание |
|------------|------|----------|---|
| | | | частоты = 0. Затем выключает ячейку ПЧ и ячейку питания ПЧ. |
| stop_alarm | WORD | 16#20 | Аварийный останов. Подаются команды на отключение ячейки ПЧ, ячейки питания ПЧ и быстрый останов ПЧ одновременно. |

4.33.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_engine» п. 3.3.3.1. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|--|
| net_out_s1 | WORD | 16#1 | «Авария контактора(ячейки) сети». Блок ячейки сети перешел в состояние «авария». |
| fr_in_s1 | WORD | 16#2 | «Авария контактора(ячейки) питания ПЧ». Блок ячейки питания ПЧ перешел в состояние «авария». |
| fr_out_s1 | WORD | 16#4 | «Авария контактора(ячейки) ПЧ». Блок ячейки ПЧ перешел в состояние «авария». |
| fr | WORD | 16#8 | «Авария ПЧ». Блок ПЧ перешел в состояние «авария». |
| fr_no_supply | WORD | 16#10 | «Пропало напряжение на вводе питания ПЧ». Вход supply_di отсутствовал в течение времени time_supply (см. интерфейс блока). |
| fr_not_ready | WORD | 16#20 | «Нет готовности ПЧ». ПЧ не пришел в состояние готовности после включения ячейки питания в течение времени, заданного параметром «time_ready_fr» (см. интерфейс блока). |
| in_s1 | WORD | 16#40 | «Авария общего контактора(ячейки) питания». Не используется. |
| fr_busy | WORD | 16#80 | «ПЧ занят». Не используется. |

4.33.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_engine» п. 3.3.3.2. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------------|------|----------|---|
| net_out_s1_not_ready | WORD | 16#1 | «Нет готовности контактора(ячейки) сети». Отсутствует состояние готовности блока ячейки сети. |
| fr_in_s1_not_ready | WORD | 16#2 | «Нет готовности контактора(ячейки) питания ПЧ». Отсутствует состояние готовности блока ячейки питания ПЧ. |
| fr_out_s1_not_ready | WORD | 16#4 | «Нет готовности контактора(ячейки) ПЧ». Отсутствует состояние готовности блока ячейки ПЧ. |
| fr_not_ready | WORD | 16#8 | «Нет готовности ПЧ». Отсутствует состояние готовности блока ПЧ. |
| net_out_s1_not_auto | WORD | 16#10 | «Контактор(ячейка) сети не в режиме автомат». Режим блока ячейки сети – не «автомат». |
| fr_in_s1_not_auto | WORD | 16#20 | «Контактор(ячейка) питания ПЧ не в режиме автомат». Режим блока ячейки питания ПЧ – не «автомат». |
| fr_out_s1_not_auto | WORD | 16#40 | «Контактор(ячейка) ПЧ не в режиме автомат». Режим блока ячейки ПЧ – не «автомат». |
| fr_not_auto | WORD | 16#80 | «ПЧ не в режиме автомат». Режим блока ПЧ – не «автомат». |
| fr_no_supply | WORD | 16#100 | «Пропало напряжение на вводе питания ПЧ». Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник с сылки не найден.). |

| Название | Тип | Значение | Текст сообщения/Описание |
|----------|------|----------|---|
| fr_wrn | WORD | 16#200 | «Предупреждение ПЧ». От блока ПЧ пришел сигнал «Предупреждение». |
| fr_alm | WORD | 16#400 | «Авария ПЧ». Блок ПЧ перешел в состояние аварии. |

4.33.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_cfg_engine» п. 3.3.3.3. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|----------------|------|----------|---|
| auto_reconnect | WORD | 16#1 | Автоматическое переключение ПЧ/УПП на сеть после разгона. Не используется. |

4.33.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_engine» п. 3.3.3.6. Возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|----------------|------|--------------------|--|
| not_ready | WORD | 16#1 | Ожидание готовности. Нет готовности одного из подчиненных блоков к пуску. |
| stopped | WORD | 16#2 | Готовность. Все подчиненные блоки готовы к пуску. |
| net_starting | WORD | 16#4 | Запуск от сети. Идет процесс запуска электродвигателя от сети. Включается ячейка сети. |
| net_started | WORD | 16#8 | В работе от сети. Ячейка сети включена, электродвигатель работает от сети. |
| net_stopping | WORD | 16#10 | Останов от сети. Идет процесс останова электродвигателя от сети. Отключается ячейка сети. |
| fr_starting | WORD | 16#20 | Запуск от ПЧ. Идет процесс запуска электродвигателя от ПЧ. Включается ячейка питания ПЧ, затем включается ячейка ПЧ. Далее включается ПЧ. |
| fr_started | WORD | 16#40 | В работе от ПЧ. Электродвигатель в работе – ячейка ПЧ включена, ПЧ в работе. |
| fr_stopping | WORD | 16#80 | Останов от ПЧ. Идет процесс останова электродвигателя от ПЧ. Отключается ПЧ с плавным снижением частоты до нуля, затем отключается ячейка ПЧ, далее отключается ячейка питания ПЧ. |
| fr_net | WORD | 16#100 | Переключение ПЧ->сеть. Идет процесс переключения электродвигателя с ПЧ на сеть. Отключается ячейка ПЧ, включается ячейка сети. Отключается ПЧ, затем ячейка питания ПЧ. Затем блок переходит в состояние «В работе от сети». |
| alarm | WORD | 16#200 | Авария. Один из подчиненных блоков перешел в состояние аварии или отключился во время работы. |
| net_stopped | WORD | 16#400 | Готовность к запуску от сети. Ячейка сети отключена и готова к пуску в режиме «автомат». Ячейка ПЧ отключена. |
| fr_stopped | WORD | 16#800 | Готовность к запуску от ПЧ. Ячейка питания ПЧ отключена и готова к пуску, ПЧ выключен и готов к пуску. |
| fr_no_supply | WORD | 16#1000 | Пропало напряжение на вводе. Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник ссылки не найден.). |
| alarm_stopping | WORD | 16#2000 | Аварийный останов. Подана команда аварийного останова. |

| Название | Тип | Начальное значение | Описание |
|----------|-----|--------------------|--|
| | | | Отключаются ПЧ, ячейка ПЧ, ячейка питания ПЧ при работе двигателя от ПЧ (команды на их отключение подаются одновременно) или ячейка сети при работе двигателя от сети. |

4.33.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_hmi_state_actual_engine» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|--|
| power_off | WORD | 16#1 | Нет питания. ПЧ или ячейка ПЧ находятся в состоянии «Нет питания» (std.out_state_actual = power_off), а также ячейка сети находится в состоянии «Нет питания» (std.out_state_actual = power_off). |
| stopped | WORD | 16#2 | Отключен. ячейка питания ПЧ , ПЧ, ячейка ПЧ находятся в состоянии «Отключен» (std.out_state_actual = stopped) или ячейка сети находится в состоянии «Отключен» (std.out_state_actual = stopped). |
| started | WORD | 16#4 | Включен. ячейка питания ПЧ, ПЧ, ячейка ПЧ находятся в состоянии «Включен» (std.out_state_actual = started) или ячейка сети в состоянии «Включен» (std.out_state_actual = started). |

4.33.7 Пример 1. Один электродвигатель.

Объявление глобальных переменных:

```
VAR_GLOBAL
// объявление интерфейса блока ячейка сети
dbs_cont_net: RuDrive.udt_switch_gear;
// объявление интерфейса блока ячейка питания ПЧ
dbs_cont_fr_supply: RuDrive.udt_switch_gear;
// объявление интерфейса блока ячейка ПЧ
dbs_cont_fr: RuDrive.udt_switch_gear;
// объявление интерфейса блока ПЧ
dbs_fr: RuDrive.udt_fr;
// объявление интерфейса блока коммутации электродвигателя
dbs_engine: RuDrive.udt_engine;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM POU
VAR
// объявление экземпляра блока ячейка сети
dbi_cont_net: RuDrive.fb_switch_gear;
// объявление экземпляра блока ячейка питания ПЧ
dbi_cont_fr_supply: RuDrive.fb_switch_gear;
// объявление экземпляра блока ячейка ПЧ
dbi_cont_fr: RuDrive.fb_switch_gear;
// объявление экземпляра блока ПЧ
dbi_fr: RuDrive.fb_fr;
// объявление экземпляра блока коммутации электродвигателя
dbi_engine: RuDrive.fb_eng_s_sfs;
END_VAR

// вызов экземпляра блока ячейка сети
dbi_cont_net(
```

```

supply_di := input_cont_net_supply_di,
ready_di := input_cont_net_ready_di,
plc_control_di := input_cont_net_plc_control_di,
fault_di := input_cont_net_fault_di,
on_di := input_cont_net_on_di,
off_di := input_cont_net_off_di,
hmi := dbs_cont_net,
on_do => output_cont_net_on_do,
off_do => output_cont_net_off_do);

// вызов экземпляра блока ячейка питания ПЧ
dbi_cont_fr_supply(
supply_di := input_cont_fr_supply_supply_di,
ready_di := input_cont_fr_supply_ready_di,
plc_control_di := input_cont_fr_supply_plc_control_di,
fault_di := input_cont_fr_supply_fault_di,
on_di := input_cont_fr_supply_on_di,
off_di := input_cont_fr_supply_off_di,
hmi := dbs_cont_fr_supply,
on_do => output_cont_fr_supply_on_do,
off_do => output_cont_fr_supply_off_do);

// вызов экземпляра блока ячейка ПЧ
dbi_cont_fr(
supply_di := input_cont_fr_supply_di,
ready_di := input_cont_fr_ready_di,
plc_control_di := input_cont_fr_plc_control_di,
fault_di := input_cont_fr_fault_di,
on_di := input_cont_fr_on_di,
off_di := input_cont_fr_off_di,
hmi := dbs_cont_fr,
on_do => output_cont_fr_on_do,
off_do => output_cont_fr_off_do);

// вызов экземпляра блока ПЧ
dbi_fr(
supply_di := input_fr_supply_di,
ready_di := input_fr_ready_di,
plc_control_di := input_fr_plc_control_di,
fault_di := input_fr_fault_di,
on_di := input_fr_on_di,
hmi := dbs_fr,
start_do => output_fr_start_do,
stop_do => output_fr_stop_do,
alm_reset_do => output_fr_alm_reset_do,
ao => output_fr_ao);

// вызов экземпляра блока коммутации электродвигателя "Ячейка + Ячейка->ПЧ->Ячейка"
dbi_engine(
plc_control_di := input_engine_plc_control_di,
supply_di := input_engine_supply_di,
hmi := dbs_engine,
fr := dbs_fr,
cont_net := dbs_cont_net,
cont_fr_supply := dbs_cont_fr_supply,
cont_fr := dbs_cont_fr );

```

4.33.8 Пример 2. Два электродвигателя с одним общим ПЧ и общей ячейкой питания ПЧ.

Объявление глобальных переменных:

```

VAR_GLOBAL
// объявление интерфейса блока ячейка сети
dbs_cont_net_1:          RuDrive.udt_switch_gear;
dbs_cont_net_2:          RuDrive.udt_switch_gear;
// объявление интерфейса блока ячейка питания ПЧ
dbs_cont_fr_supply: RuDrive.udt_switch_gear;
// объявление интерфейса блока ячейка ПЧ
dbs_cont_fr_1:          RuDrive.udt_switch_gear;

```

```

dbs_cont_fr_2:          RuDrive.udt_switch_gear;
// объявление интерфейса блока ПЧ
dbs_fr:                RuDrive.udt_fr;
// объявление интерфейса блока коммутации электродвигателя
dbs_engine_1:         RuDrive.udt_engine;
dbs_engine_2:         RuDrive.udt_engine;

```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM PLC_PRG

VAR

```

// объявление экземпляра блока ячейка сети
dbi_cont_net_1:       RuDrive.fb_switch_gear;
dbi_cont_net_2:       RuDrive.fb_switch_gear;
// объявление экземпляра блока ячейка питания ПЧ
dbi_cont_fr_supply:   RuDrive.fb_switch_gear;
// объявление экземпляра блока ячейка ПЧ
dbi_cont_fr_1:        RuDrive.fb_switch_gear;
dbi_cont_fr_2:        RuDrive.fb_switch_gear;
// объявление экземпляра блока ПЧ
dbi_fr:               RuDrive.fb_fr;
// объявление экземпляра блока коммутации электродвигателя
dbi_engine_1:         RuDrive.fb_eng_s_sfs;
dbi_engine_2:         RuDrive.fb_eng_s_sfs;

```

END_VAR

// вызов экземпляра блока ячейка сети

```

dbi_cont_net_1(
  supply_di := input_cont_net_1_supply_di,
  ready_di := input_cont_net_1_ready_di,
  plc_control_di := input_cont_net_1_plc_control_di,
  fault_di := input_cont_net_1_fault_di,
  on_di := input_cont_net_1_on_di,
  off_di := input_cont_net_1_off_di,
  hmi := dbs_cont_net_1,
  on_do => output_cont_net_1_on_do,
  off_do => output_cont_net_1_off_do);

```

```

dbi_cont_net_2(
  supply_di := input_cont_net_2_supply_di,
  ready_di := input_cont_net_2_ready_di,
  plc_control_di := input_cont_net_2_plc_control_di,
  fault_di := input_cont_net_2_fault_di,
  on_di := input_cont_net_2_on_di,
  off_di := input_cont_net_2_off_di,
  hmi := dbs_cont_net_2,
  on_do => output_cont_net_2_on_do,
  off_do => output_cont_net_2_off_do);

```

// вызов экземпляра блока ячейка питания ПЧ

```

dbi_cont_fr_supply(
  supply_di := input_cont_fr_supply_supply_di,
  ready_di := input_cont_fr_supply_ready_di,
  plc_control_di := input_cont_fr_supply_plc_control_di,
  fault_di := input_cont_fr_supply_fault_di,
  on_di := input_cont_fr_supply_on_di,
  off_di := input_cont_fr_supply_off_di,
  hmi := dbs_cont_fr_supply,
  on_do => output_cont_fr_supply_on_do,
  off_do => output_cont_fr_supply_off_do);

```

// вызов экземпляра блока ячейка ПЧ

```

dbi_cont_fr_1(
  supply_di := input_cont_fr_1_supply_di,
  ready_di := input_cont_fr_1_ready_di,
  plc_control_di := input_cont_fr_1_plc_control_di,
  fault_di := input_cont_fr_1_fault_di,
  on_di := input_cont_fr_1_on_di,
  off_di := input_cont_fr_1_off_di,
  hmi := dbs_cont_fr_1,
  on_do => output_cont_fr_1_on_do,

```

```

    off_do => output_cont_fr_1_off_do);

dbi_cont_fr_2(
    supply_di := input_cont_fr_2_supply_di,
    ready_di := input_cont_fr_2_ready_di,
    plc_control_di := input_cont_fr_2_plc_control_di,
    fault_di := input_cont_fr_2_fault_di,
    on_di := input_cont_fr_2_on_di,
    off_di := input_cont_fr_2_off_di,
    hmi := dbs_cont_fr_2,
    on_do => output_cont_fr_2_on_do,
    off_do => output_cont_fr_2_off_do);

// вызов экземпляра блока ПЧ
dbi_fr(
    supply_di := input_fr_supply_di,
    ready_di := input_fr_ready_di,
    plc_control_di := input_fr_plc_control_di,
    fault_di := input_fr_fault_di,
    on_di := input_fr_on_di,
    hmi := dbs_fr,
    start_do => output_fr_start_do,
    stop_do => output_fr_stop_do,
    alm_reset_do => output_fr_alm_reset_do,
    ao => output_fr_ao);

// вызов экземпляра блока коммутации электродвигателя "Ячейка + Ячейка->ПЧ->Ячейка"
dbi_engine_1(
    plc_control_di := input_engine_1_plc_control_di,
    supply_di := input_engine_1_supply_di,
    fr_busy_di := dbi_engine_2.fr_busy_do,
    cont_fr_lock_di := dbi_engine_2.cont_fr_lock_do,
    hmi := dbs_engine_1,
    fr := dbs_fr,
    cont_net := dbs_cont_net_1,
    cont_fr_supply := dbs_cont_fr_supply,
    cont_fr := dbs_cont_fr_1 );

dbi_engine_2(
    plc_control_di := input_engine_2_plc_control_di,
    supply_di := input_engine_2_supply_di,
    fr_busy_di := dbi_engine_1.fr_busy_do,
    cont_fr_lock_di := dbi_engine_1.cont_fr_lock_do,
    hmi := dbs_engine_2,
    fr := dbs_fr,
    cont_net := dbs_cont_net_1,
    cont_fr_supply := dbs_cont_fr_supply,
    cont_fr := dbs_cont_fr_1 );

```

4.34 Коммутация электродвигателя «Ячейка + Ячейка->УПП->Ячейка» (fb_eng_s_sus)

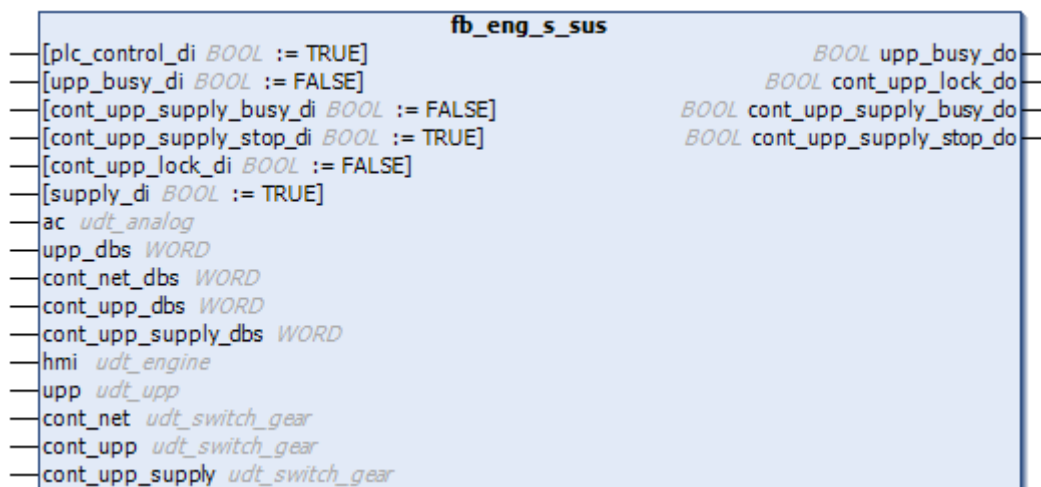
4.34.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации.

4.34.2 Функции

- запуск и останов электродвигателя с помощью УПП и прямым пуском от сети.
- переключение с УПП на сеть.
- возобновление работы от УПП при пропадании и восстановлении питания.

4.34.3 Таблица входов-выходов

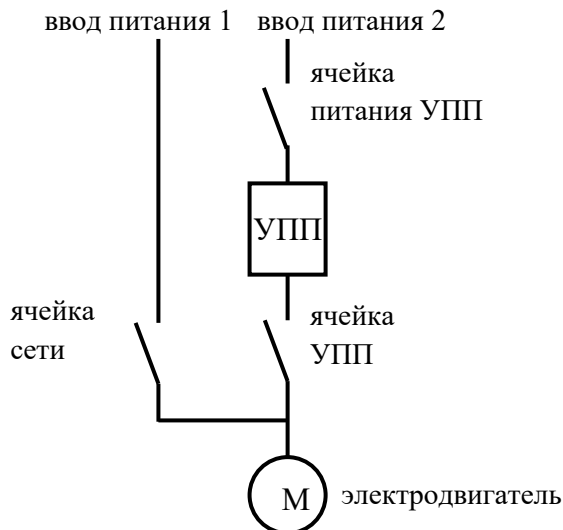


| Область | Название | Тип | Начальное значение | Назначение |
|---------------------|-------------------------|-----------------|--------------------|---|
| Input | plc_control_di | BOOL | TRUE | Состояние переключателя режима ручной/автомат (FALSE - ручной, TRUE - автомат) |
| | upp_busy_di | BOOL | FALSE | УПП занят (сигнал от других блоков) |
| | cont_upp_supply_busy_di | BOOL | FALSE | Ячейка питания УПП занята (если питание УПП от одной ячейки) |
| | cont_upp_supply_stop_di | BOOL | TRUE | Соседняя ячейка питания УПП выключена (если УПП питается от нескольких ячеек) |
| | cont_upp_lock_di | BOOL | FALSE | Запрет включения ячейки УПП в любом режиме (сигнал от других блоков) |
| | supply_di | BOOL | TRUE | Наличие напряжения на вводе питания УПП |
| | ac | udt_analog | | Ток двигателя при работе от сети. Вход для передачи интерфейса аналогового входа, который измеряет ток двигателя. |
| | upp_dbs | WORD | | Номера DB объектов. Используется только для платформы Siemens для корректного отображения данных на панели оператора. |
| | cont_net_dbs | WORD | | |
| | cont_upp_dbs | WORD | | |
| cont_upp_supply_dbs | WORD | | | |
| Inout | hmi | udt_engine | | интерфейс блока |
| | upp | udt_upp | | интерфейс блока УПП |
| | cont_net | udt_switch_gear | | интерфейс блока ячейки сети |
| | cont_upp | udt_switch_gear | | интерфейс блока ячейки УПП |
| | cont_upp_supply | udt_switch_gear | | интерфейс блока ячейки питания УПП |
| Output | upp_busy_do | BOOL | FALSE | УПП занят (сигнал другим блокам) |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|-------------------------|------|--------------------|--|
| | cont_upp_lock_do | BOOL | FALSE | Запрет включения ячейки УПП в любом режиме (сигнал другим блокам) |
| | cont_upp_supply_busy_do | BOOL | FALSE | Ячейка питания УПП занята (если питание УПП от одной ячейки) |
| | cont_upp_supply_stop_do | BOOL | FALSE | Ячейка питания УПП отключена (блокировка запуска других ячеек питания УПП) |

4.34.4 Принцип работы

Схема коммутации показана на рисунке.



Блок может выполнять следующие операции:

- запуск электродвигателя от сети (с помощью ячейки сети);
- запуск электродвигателя от УПП (с помощью ячейки питания УПП, УПП и ячейки УПП);
- переключение электродвигателя с УПП на сеть на ходу (во время работы от УПП);
- возобновление работы от УПП при пропадании и восстановлении питания.

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_switch_gear (ячейка сети, ячейка питания УПП, ячейка УПП) и fb_fr (УПП).

Готовность к пуску

Блок имеет несколько состояний готовности:

- «готовность к пуску от сети»;
- «готовность к пуску от УПП»;
- «готовность к пуску».

«Готовность к пуску от сети» означает, что ячейка сети в состоянии готовности и в режиме «автомат», а также ячейка УПП отключена. При этом другие объекты (ячейка питания УПП, УПП или ячейка УПП) либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску от УПП» означает, ячейка питания УПП, УПП и ячейка УПП в состоянии готовности и в режиме «автомат», а также ячейка сети отключена. При этом ячейка сети либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску» означает, что все объекты (ячейка сети, ячейка питания УПП, УПП, ячейка УПП) находятся в состоянии готовности и в режиме «автомат».

Работа от сети

Для запуска электродвигателя от сети необходимо, чтобы блок был либо в состоянии «готовность к пуску от сети», либо «готовность к пуску».

При подаче команды «пуск от сети» (start_net) будет включена ячейка сети, блок перейдет в состояние «работа от сети».

При подаче команды «стоп» (stop) ячейка сети будет отключена, блок перейдет в состояние «остановлен».

Работа от УПП

Для запуска электродвигателя от УПП необходимо, чтобы блок был либо в состоянии «готовность к пуску от УПП», либо «готовность к пуску».

При подаче команды «пуск от УПП» (start_fr) будут выполнены следующие действия:

- включение ячейки питания УПП;
- ожидание готовности УПП;
- включение ячейки УПП (подключение электродвигателя к выходу УПП);
- включение УПП.

Блок перейдет в состояние «работа от УПП».

При подаче команды «стоп» (stop) будут выполнены следующие действия:

- останов УПП;
- отключение ячейки УПП (отключение электродвигателя от выхода УПП);
- отключение ячейки питания УПП.

Блок перейдет в состояние «готовность к пуску от УПП» либо «готовность к пуску».

Переключение с УПП на сеть

Для переключения электродвигателя с УПП на сеть во время работы от УПП нужно подать команду «reconnect_fr_net». Ячейка УПП будет отключена (без выключения УПП), и сразу включена ячейка сети. Электродвигатель продолжит работу от сети. После этого будет подана команда на останов УПП. Желательно, чтобы для питания ячейки сети и УПП использовался один ввод питания, тогда переключение на сеть будет безударным.

Совместное использование нескольких блоков. Один УПП на несколько электродвигателей

На рисунке ниже показаны схемы включения двух электродвигателей, которые могут каждый независимо включаться прямым пуском от сети. Электродвигателей может быть и больше, чем два. Пунктирной линией выделены объекты, которые относятся к каждому из блоков коммутации электродвигателей. На первой схеме ячейка питания УПП и УПП общие на оба блока. На второй – общим является только УПП, а ячейки питания независимые и подключены к разным вводам.

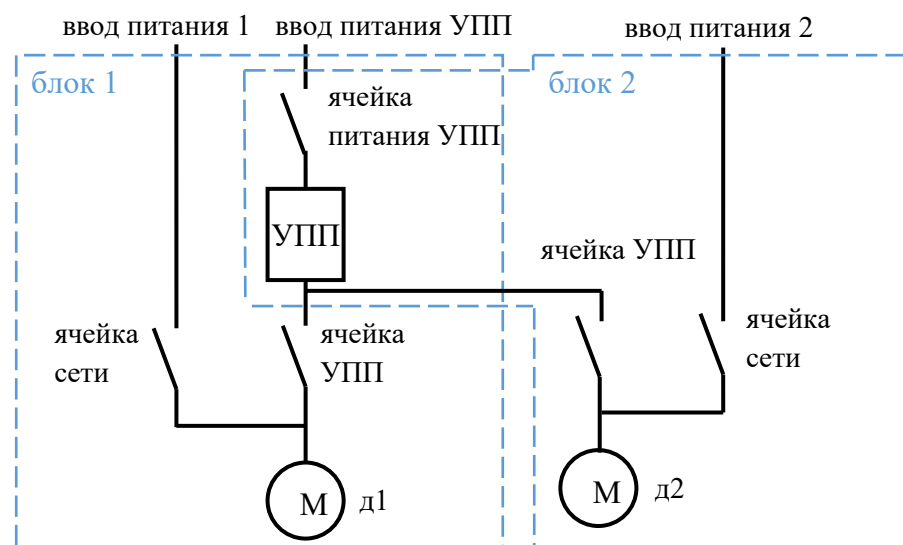


Схема включения двух электродвигателей с общими УПП и ячейкой питания УПП

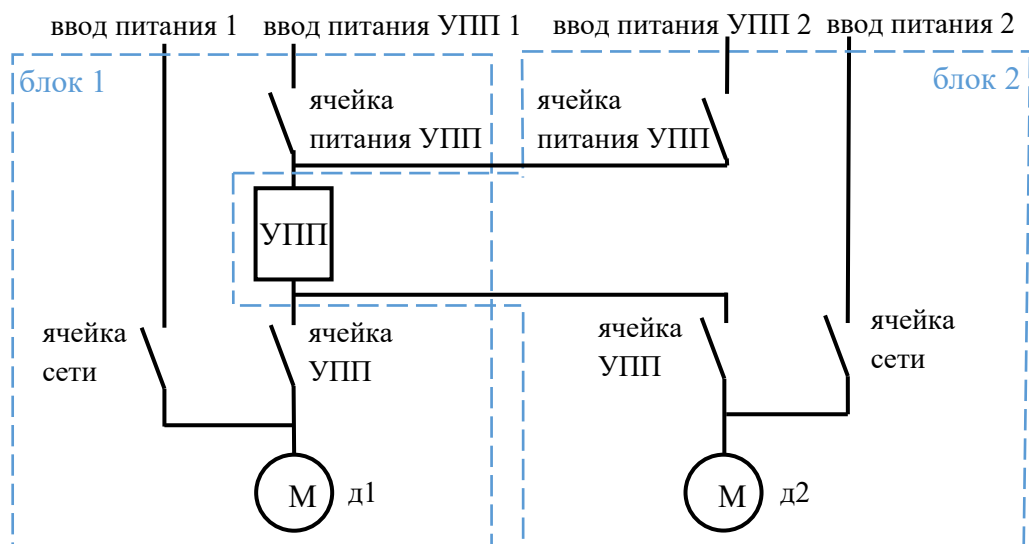


Схема включения двух электродвигателей с общим УПП и отдельными ячейками питания УПП

Для того, чтобы блоки коммутации могли корректно управлять разделяемыми ресурсами – УПП и его ячейкой питания,

предусмотрены следующие входные и выходные сигналы:

- «fr_busy_di»: вход; сигнал от других блоков коммутации, что в данный момент УПП занят ими;
- «fr_busy_do»: выход; сигнал другим блокам коммутации, что в данный момент УПП занят данным блоком.
- «cont_fr_lock_di»: вход; сигнал от других блоков коммутации, что требуется запретить запуск ячейки УПП, поскольку включена ячейка УПП другого блока.
- «cont_fr_lock_do»: выход; сигнал другим блокам коммутации, чтобы запретили запуск своих ячеек УПП, поскольку включена ячейка УПП данного блока.
- «cont_urr_supply_busy_di»: вход; сигнал от других блоков коммутации, что в данный момент ячейка питания УПП занята ими; вход нужно использовать, когда ячейка питания УПП общая на все блоки;
- «cont_urr_supply_busy_do»: выход; сигнал другим блокам коммутации, что в данный момент ячейка питания УПП занята данным блоком; выход нужно использовать, когда ячейка питания УПП общая на все блоки;
- «cont_urr_supply_stop_di»: вход; сигнал от других блоков коммутации, что в данный момент соседняя ячейка питания УПП выключена; вход нужно использовать, когда у каждого блока своя ячейка питания УПП; в этом случае сигналы «cont_urr_supply_busy_di» и «cont_urr_supply_busy_do» не используются;
- «cont_urr_supply_stop_do»: выход; сигнал другим блокам коммутации, что в данный момент ячейка питания УПП данного блока выключена; выход нужно использовать, когда у каждого блока своя ячейка питания УПП; в этом случае сигналы «cont_urr_supply_busy_di» и «cont_urr_supply_busy_do» не используются.

Таким образом, при вызове блоков в программе нужно:

- на вход «fr_busy_di» подать сигналы «fr_busy_do» от других блоков;
- на вход «cont_fr_lock_di» подать сигналы «cont_fr_lock_do» от других блоков;
- на вход «cont_urr_supply_busy_di» подать сигналы «cont_urr_supply_busy_do» от других блоков, если ячейка питания УПП общая; иначе не использовать;
- на вход «cont_urr_supply_stop_di» подать сигналы «cont_urr_supply_stop_do» от других блоков, если ячейка питания УПП отдельная на каждый блок; иначе не использовать.

Если ко входу нужно привязать более, чем один сигнал, нужно объединить их логическим «или».

Возобновление работы электродвигателя от УПП при пропадании и восстановлении питания

Блок поддерживает функцию возобновления работы электродвигателя от УПП при пропадании и последующем восстановлении питания. Сигнал наличия напряжения на вводе питания УПП нужно

привязать ко входу «suprly_di» (сигнал может быть взят с реле контроля фаз, например).
Если во время работы двигателя от УПП питание пропало, а затем восстановилось в течение заданного времени, будет произведено повторное включение ячейки УПП и запуск УПП.

4.34.5 Интерфейс блока (udt_engine)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|----------------------|-------------|------------|---|
| std | <u>udt_standard</u> | | ввод-вывод | стандартный набор переменных |
| out_ac | REAL | | вывод | Ток двигателя. Значение передается со входа «ас». |
| out_freq | REAL | | вывод | Частота двигателя. Выводится фиксированное значение 50 Гц, когда электродвигатель в работе от сети или УПП, иначе выводится 0. |
| out_speed | REAL | | вывод | Частота вращения двигателя, об/мин. Расчетное значение на основе текущей частоты (out_freq) и максимальной частоты вращения (max_speed): $out_speed = (out_freq/50)*max_speed$. |
| ready_net | WORD | | вывод | Готовность к пуску от сети. Бит 0 – общая готовность к пуску от сети. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – ячейка сети отключена. Бит 2 – ячейка УПП отключена. Бит 3 – ячейка сети в режиме «автомат». |
| ready_fr | WORD | | вывод | Готовность к пуску от ПЧ. Бит 0 – общая готовность от ПЧ. Если бит взведен, блок готов к пуску. Если нет, то нужно смотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – ячейка сети отключена. Бит 2 – ячейка УПП отключена. Бит 3 – ячейка УПП в режиме «автомат». Бит 4 – УПП в режиме «автомат». Бит 5 – УПП не занят. Бит 6 – Ячейка питания УПП не занята. Бит 7 – Нет запрета включения ячейки УПП (от других блоков). Бит 8 – Ячейка питания УПП готова (отключена или включена) |
| pid | <u>udt_pid</u> | | ввод-вывод | Не используется. |
| ac_settings | <u>udt_engine_ac</u> | | ввод | Не используется. |
| type_connect | INT | | вывод | Тип подключения двигателя. Служебный параметр для использования другими блоками. |
| freq_low | REAL | 5 | ввод | Не используется. |
| freq_hi | REAL | 50 | ввод | Не используется. |
| freq_rising | REAL | 5 | ввод | Не используется. |
| freq_lowering | REAL | 5 | ввод | Не используется. |
| freq_overload_lowering | REAL | 5 | ввод | Не используется. |
| freq_deadband | REAL | 1 | ввод | Не используется. |
| time_freq_min_wait | REAL | 10 | ввод | Не используется. |
| time_supply | REAL | 10 | ввод | Время ожидания возвращения напряжения на вводе питания УПП, сек. При пропадании напряжения на вводе питания УПП (suprly_di = FALSE) блок удерживает состояние «в работе». Если напряжение на вводе |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-------------------|------|-------------|------------|--|
| | | | | вернулось (supply_di = TRUE) в течение этого времени, осуществляется перезапуск ячейки УПП и УПП. Если напряжение не вернулось, блок переходит в состояние аварии. |
| time_ready_fr | REAL | 5 | ввод | Время ожидания готовности УПП, сек. Если сигнал готовности УПП не пришел в течение этого времени после включения ячейки питания, блок переходит в состояние аварии. |
| time_stop_cont_fr | REAL | 60 | ввод | Время таймаута отключения ячейки питания УПП, сек. Задержка отключения ячейки питания УПП после подачи команды останова на блок. |
| max_speed | INT | 3000 | ввод | Максимальная частота вращения двигателя, об/мин. Используется для расчета текущей частоты вращения out_speed. |

4.34.6 Переменные интерфейса блока. Списки значений

4.34.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_engine» п. 3.3.3.4. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|------------------|------|----------|--|
| stop | WORD | 16#1 | Стоп. |
| net_start | WORD | 16#2 | Пуск от сети. |
| fr_start | WORD | 16#4 | Пуск от УПП. |
| reconnect_fr_net | WORD | 16#8 | Переключение с УПП на сеть. |
| stop_fast | WORD | 16#10 | Быстрый останов. Сначала останавливает УПП. Затем выключает ячейку УПП и ячейку питания УПП. |
| stop_alarm | WORD | 16#20 | Аварийный останов. Подаются команды на отключение ячейки УПП, ячейки питания УПП и останов УПП одновременно. |

4.34.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_engine» п. 3.3.3.1. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|---|
| net_out_s1 | WORD | 16#1 | «Авария контактора(ячейки) сети». Блок ячейки сети перешел в состояние «авария». |
| fr_in_s1 | WORD | 16#2 | «Авария контактора(ячейки) питания УПП». Блок ячейки питания УПП перешел в состояние «авария». |
| fr_out_s1 | WORD | 16#4 | «Авария контактора(ячейки) УПП». Блок ячейки УПП перешел в состояние «авария». |
| fr | WORD | 16#8 | «Авария УПП». Блок УПП перешел в состояние «авария». |
| fr_no_supply | WORD | 16#10 | «Пропало напряжение на вводе питания УПП». Вход supply_di отсутствовал в течение времени time_supply (см. интерфейс блока). |
| fr_not_ready | WORD | 16#20 | «Нет готовности УПП». УПП не пришел в состояние готовности после включения ячейки питания в течение времени, заданного параметром «time_ready_fr» (см. интерфейс блока). |

| Название | Тип | Значение | Текст сообщения/Описание |
|----------|------|----------|---|
| in_s1 | WORD | 16#40 | «Авария общего контактора(ячейки) питания». Не используется. |
| fr_busy | WORD | 16#80 | «УПП занят». Не используется. |

4.34.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_engine» п. 3.3.3.2. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------------|------|----------|--|
| net_out_s1_not_ready | WORD | 16#1 | «Нет готовности контактора(ячейки) сети». Отсутствует состояние готовности блока ячейки сети. |
| fr_in_s1_not_ready | WORD | 16#2 | «Нет готовности контактора(ячейки) питания УПП». Отсутствует состояние готовности блока ячейки питания УПП. |
| fr_out_s1_not_ready | WORD | 16#4 | «Нет готовности контактора(ячейки) УПП». Отсутствует состояние готовности блока ячейки УПП. |
| fr_not_ready | WORD | 16#8 | «Нет готовности УПП». Отсутствует состояние готовности блока УПП. |
| net_out_s1_not_auto | WORD | 16#10 | «Контактор(ячейка) сети не в режиме автомат». Режим блока ячейки сети – не «автомат». |
| fr_in_s1_not_auto | WORD | 16#20 | «Контактор(ячейка) питания УПП не в режиме автомат». Режим блока ячейки питания УПП – не «автомат». |
| fr_out_s1_not_auto | WORD | 16#40 | «Контактор(ячейка) УПП не в режиме автомат». Режим блока ячейки УПП – не «автомат». |
| fr_not_auto | WORD | 16#80 | «УПП не в режиме автомат». Режим блока УПП – не «автомат». |
| fr_no_supply | WORD | 16#100 | «Пропало напряжение на вводе питания УПП». Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. таблицу п. Ошибка! Источник с ссылки не найден.). |
| fr_wrn | WORD | 16#200 | «Предупреждение УПП». От блока УПП пришел сигнал «Предупреждение». |
| fr_alm | WORD | 16#400 | «Авария УПП». Блок УПП перешел в состояние аварии. |

4.34.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_cfg_engine» п. 3.3.3.3. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|----------------|------|----------|---|
| auto_reconnect | WORD | 16#1 | Автоматическое переключение ПЧ/УПП на сеть после разгона. Если флаг включен, то после разгона УПП (прихода сигнала «работа» от УПП) будет инициировано переключение с УПП на сеть. |

4.34.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_engine» п. 3.3.3.6. Возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|----------------------|
| not_ready | WORD | 16#1 | Ожидание готовности. |

| Название | Тип | Начальное значение | Описание |
|----------------|------|--------------------|---|
| | | | Нет готовности одного из подчиненных блоков к пуску. |
| stopped | WORD | 16#2 | Готовность. Все подчиненные блоки готовы к пуску. |
| net_starting | WORD | 16#4 | Запуск от сети. Идет процесс запуска электродвигателя от сети. Включается ячейка сети. |
| net_started | WORD | 16#8 | В работе от сети. Ячейка сети включена, электродвигатель работает от сети. |
| net_stopping | WORD | 16#10 | Останов от сети. Идет процесс останова электродвигателя от сети. Отключается ячейка сети. |
| fr_starting | WORD | 16#20 | Запуск от УПП. Идет процесс запуска электродвигателя от УПП. Включается ячейка питания УПП, затем включается ячейка УПП. Далее включается УПП. |
| fr_started | WORD | 16#40 | В работе от УПП. Электродвигатель в работе – ячейка питания УПП включена, ячейка УПП включена, УПП в работе. |
| fr_stopping | WORD | 16#80 | Останов от УПП. Идет процесс останова электродвигателя от УПП. Отключается УПП, затем отключается ячейка УПП, далее отключается ячейка питания УПП. |
| fr_net | WORD | 16#100 | Переключение УПП->сеть. Идет процесс переключения электродвигателя с УПП на сеть. Отключается ячейка УПП, включается ячейка сети. Отключается УПП, затем ячейка питания УПП. Затем блок переходит в состояние «В работе от сети». |
| alarm | WORD | 16#200 | Авария. Один из подчиненных блоков перешел в состояние аварии или отключился во время работы. |
| net_stopped | WORD | 16#400 | Готовность к запуску от сети. Ячейка сети отключена и готова к пуску в режиме «автомат». Ячейка УПП отключена. |
| fr_stopped | WORD | 16#800 | Готовность к запуску от УПП. Ячейка питания УПП отключена и готова к пуску или включена, УПП выключен и готов к пуску, ячейка УПП выключена и готова к пуску. Все объекты в режиме «автомат». |
| fr_no_supply | WORD | 16#1000 | Пропало напряжение на вводе. Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. описание интерфейса блока). |
| alarm_stopping | WORD | 16#2000 | Аварийный останов. Подана команда аварийного останова. Отключаются УПП, ячейка УПП, ячейка питания УПП при работе двигателя от УПП (команды на их отключение подаются одновременно) или ячейка сети при работе двигателя от сети. |

4.34.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_hmi_state_actual_engine» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|--|
| power_off | WORD | 16#1 | Нет питания. ячейка питания УПП, УПП или ячейка УПП находятся в состоянии «Нет питания» (std.out_state_actual = power_off), а также ячейка сети находится в состоянии «Нет питания» (std.out_state_actual = power_off). |
| stopped | WORD | 16#2 | Отключен. ячейка питания УПП, УПП, ячейка УПП находятся в состоянии «Отключен» (std.out_state_actual = stopped) или ячейка сети находится в состоянии «Отключен» (std.out_state_actual = stopped). |
| started | WORD | 16#4 | Включен. ячейка питания УПП, УПП, ячейка УПП находятся в состоянии «Включен» (std.out_state_actual = started) |

| Название | Тип | Начальное значение | Описание |
|----------|-----|--------------------|---|
| | | | или ячейка сети в состоянии «Включен» (std.out_state_actual = started). |

4.34.7 Пример 1. Один электродвигатель.

Объявление глобальных переменных:

VAR_GLOBAL

```
// объявление интерфейса блока ячейка сети
dbs_cont_net: RuDrive.udt_switch_gear;
// объявление интерфейса блока ячейка питания УПП
dbs_cont_fr_supply: RuDrive.udt_switch_gear;
// объявление интерфейса блока ячейка ПЧ
dbs_cont_fr: RuDrive.udt_switch_gear;
// объявление интерфейса блока УПП
dbs_upp: RuDrive.udt_upp;
// объявление интерфейса блока коммутации электродвигателя
dbs_engine: RuDrive.udt_engine;
```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM POU

VAR

```
// объявление экземпляра блока ячейка сети
dbi_cont_net: RuDrive.fb_switch_gear;
// объявление экземпляра блока ячейка питания УПП
dbi_cont_fr_supply: RuDrive.fb_switch_gear;
// объявление экземпляра блока ячейка УПП
dbi_cont_fr: RuDrive.fb_switch_gear;
// объявление экземпляра блока УПП
dbi_upp: RuDrive.fb_upp;
// объявление экземпляра блока коммутации электродвигателя
dbi_engine: RuDrive.fb_eng_s_sus;
```

END_VAR

// вызов экземпляра блока ячейка сети

```
dbi_cont_net(
    supply_di := input_cont_net_supply_di,
    ready_di := input_cont_net_ready_di,
    plc_control_di := input_cont_net_plc_control_di,
    fault_di := input_cont_net_fault_di,
    on_di := input_cont_net_on_di,
    off_di := input_cont_net_off_di,
    hmi := dbs_cont_net,
    on_do => output_cont_net_on_do,
    off_do => output_cont_net_off_do);
```

// вызов экземпляра блока ячейка питания УПП

```
dbi_cont_fr_supply(
    supply_di := input_cont_fr_supply_supply_di,
    ready_di := input_cont_fr_supply_ready_di,
    plc_control_di := input_cont_fr_supply_plc_control_di,
    fault_di := input_cont_fr_supply_fault_di,
    on_di := input_cont_fr_supply_on_di,
    off_di := input_cont_fr_supply_off_di,
    hmi := dbs_cont_fr_supply,
    on_do => output_cont_fr_supply_on_do,
    off_do => output_cont_fr_supply_off_do);
```

// вызов экземпляра блока ячейка УПП

```
dbi_cont_fr(
    supply_di := input_cont_fr_supply_di,
    ready_di := input_cont_fr_ready_di,
    plc_control_di := input_cont_fr_plc_control_di,
    fault_di := input_cont_fr_fault_di,
    on_di := input_cont_fr_on_di,
```

```

off_di := input_cont_fr_off_di,
hmi := dbs_cont_fr,
on_do => output_cont_fr_on_do,
off_do => output_cont_fr_off_do);

// вызов экземпляра блока УПП
dbi_upp(
    supply_di := input_upp_supply_di,
    ready_di := input_upp_ready_di,
    plc_control_di := input_upp_plc_control_di,
    fault_di := input_upp_fault_di,
    on_di := input_upp_on_di,
    hmi := dbs_upp,
    on_do => output_upp_on_do,
    reset_do => output_upp_reset_do);

// вызов экземпляра блока коммутации электродвигателя "Ячейка + Ячейка->УПП->Ячейка"
dbi_engine(
    plc_control_di := input_engine_plc_control_di,
    supply_di := input_engine_supply_di,
    hmi := dbs_engine,
    fr := dbs_upp,
    cont_net := dbs_cont_net,
    cont_fr_supply := dbs_cont_fr_supply,
    cont_fr := dbs_cont_fr );

```

4.34.8 Пример 2. Два электродвигателя с одним общим УПП и общей ячейкой питания УПП.

Объявление глобальных переменных:

```

VAR_GLOBAL
    // объявление интерфейса блока ячейка сети
    dbs_cont_net_1:      RuDrive.udt_switch_gear;
    dbs_cont_net_2:      RuDrive.udt_switch_gear;
    // объявление интерфейса блока ячейка питания УПП
    dbs_cont_fr_supply:  RuDrive.udt_switch_gear;
    // объявление интерфейса блока ячейка УПП
    dbs_cont_fr_1:      RuDrive.udt_switch_gear;
    dbs_cont_fr_2:      RuDrive.udt_switch_gear;
    // объявление интерфейса блока УПП
    dbs_upp:            RuDrive.udt_upp;
    // объявление интерфейса блока коммутации электродвигателя
    dbs_engine_1:      RuDrive.udt_engine;
    dbs_engine_2:      RuDrive.udt_engine;
END_VAR

```

Вызов в циклической подпрограмме (цикл 100 мсек):

```

PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока ячейка сети
    dbi_cont_net_1:      RuDrive.fb_switch_gear;
    dbi_cont_net_2:      RuDrive.fb_switch_gear;
    // объявление экземпляра блока ячейка питания УПП
    dbi_cont_fr_supply:  RuDrive.fb_switch_gear;
    // объявление экземпляра блока ячейка УПП
    dbi_cont_fr_1:      RuDrive.fb_switch_gear;
    dbi_cont_fr_2:      RuDrive.fb_switch_gear;
    // объявление экземпляра блока УПП
    dbi_upp:            RuDrive.fb_upp;
    // объявление экземпляра блока коммутации электродвигателя
    dbi_engine_1:      RuDrive.fb_eng_s_sus;
    dbi_engine_2:      RuDrive.fb_eng_s_sus;
END_VAR

// вызовы экземпляров подчиненных блоков для простоты не показаны

// вызов экземпляра блока коммутации электродвигателя "Ячейка + Ячейка->УПП->Ячейка"

```

```

dbi_engine_1(
    plc_control_di := input_engine_1_plc_control_di,
    supply_di := input_engine_1_supply_di,
    fr_busy_di := dbi_engine_2.fr_busy_do,
    cont_fr_lock_di := dbi_engine_2.cont_fr_lock_do,
    cont_fr_supply_busy_di := dbi_engine_2.cont_fr_supply_busy_do,
    hmi := dbs_engine_1,
    upp := dbs_upp,
    cont_net := dbs_cont_net_1,
    cont_fr_supply := dbs_cont_fr_supply,
    cont_fr := dbs_cont_fr_1 );

dbi_engine_2(
    plc_control_di := input_engine_2_plc_control_di,
    supply_di := input_engine_2_supply_di,
    fr_busy_di := dbi_engine_1.fr_busy_do,
    cont_fr_lock_di := dbi_engine_1.cont_fr_lock_do,
    cont_fr_supply_busy_di := dbi_engine_1.cont_fr_supply_busy_do,
    hmi := dbs_engine_2,
    upp := dbs_upp,
    cont_net := dbs_cont_net_1,
    cont_fr_supply := dbs_cont_fr_supply,
    cont_fr := dbs_cont_fr_1 );

```

4.34.9 Пример 3. Два электродвигателя с одним общим УПП и отдельными ячейками питания УПП.

Объявление глобальных переменных:

```

VAR_GLOBAL
    // объявление интерфейса блока ячейка сети
    dbs_cont_net_1: RuDrive.udt_switch_gear;
    dbs_cont_net_2: RuDrive.udt_switch_gear;
    // объявление интерфейса блока ячейка питания УПП
    dbs_cont_fr_supply_1: RuDrive.udt_switch_gear;
    dbs_cont_fr_supply_2: RuDrive.udt_switch_gear;
    // объявление интерфейса блока ячейка УПП
    dbs_cont_fr_1: RuDrive.udt_switch_gear;
    dbs_cont_fr_2: RuDrive.udt_switch_gear;
    // объявление интерфейса блока УПП
    dbs_upp: RuDrive.udt_upp;
    // объявление интерфейса блока коммутации электродвигателя
    dbs_engine_1: RuDrive.udt_engine;
    dbs_engine_2: RuDrive.udt_engine;
END_VAR

```

Вызов в циклической подпрограмме (цикл 100 мсек):

```

PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока ячейка сети
    dbi_cont_net_1: RuDrive.fb_switch_gear;
    dbi_cont_net_2: RuDrive.fb_switch_gear;
    // объявление экземпляра блока ячейка питания УПП
    dbi_cont_fr_supply_1: RuDrive.fb_switch_gear;
    dbi_cont_fr_supply_2: RuDrive.fb_switch_gear;
    // объявление экземпляра блока ячейка УПП
    dbi_cont_fr_1: RuDrive.fb_switch_gear;
    dbi_cont_fr_2: RuDrive.fb_switch_gear;
    // объявление экземпляра блока УПП
    dbi_upp: RuDrive.fb_upp;
    // объявление экземпляра блока коммутации электродвигателя
    dbi_engine_1: RuDrive.fb_eng_s_sus;
    dbi_engine_2: RuDrive.fb_eng_s_sus;
END_VAR

// вызовы экземпляров подчиненных блоков для простоты не показаны

```

```

// вызов экземпляра блока коммутации электродвигателя "Ячейка + Ячейка->УПП->Ячейка"
dbi_engine_1(
    plc_control_di := input_engine_1_plc_control_di,
    supply_di := input_engine_1_supply_di,
    fr_busy_di := dbi_engine_2.fr_busy_do,
    cont_fr_lock_di := dbi_engine_2.cont_fr_lock_do,
    cont_fr_supply_stop_di := dbi_engine_2.cont_fr_supply_stop_do
    hmi := dbs_engine_1,
    upp := dbs_upp,
    cont_net := dbs_cont_net_1,
    cont_fr_supply := dbs_cont_fr_supply,
    cont_fr := dbs_cont_fr_1 );

dbi_engine_2(
    plc_control_di := input_engine_2_plc_control_di,
    supply_di := input_engine_2_supply_di,
    fr_busy_di := dbi_engine_1.fr_busy_do,
    cont_fr_lock_di := dbi_engine_1.cont_fr_lock_do,
    cont_fr_supply_stop_di := dbi_engine_1.cont_fr_supply_stop_do
    hmi := dbs_engine_2,
    upp := dbs_upp,
    cont_net := dbs_cont_net_1,
    cont_fr_supply := dbs_cont_fr_supply,
    cont_fr := dbs_cont_fr_1 );

```

4.35 Коммутация электродвигателя «Ячейка->Контактор + Ячейка->Контактор->ПЧ->Ячейка» (fb_eng_sc_scfs)

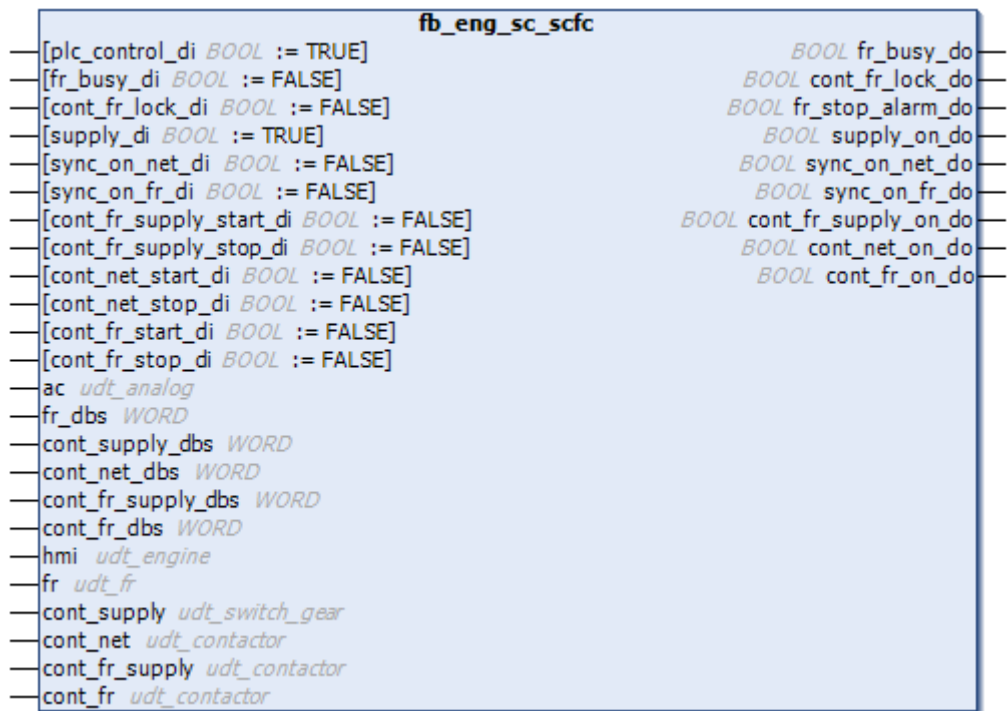
4.35.1 Назначение

Включение и выключение электродвигателя согласно заданной электрической схеме коммутации.

4.35.2 Функции

- запуск и останов электродвигателя с помощью преобразователя частоты и прямым пуском от сети.
- переключение с ПЧ на сеть с синхронизацией фаз и уровней напряжения на выходе ПЧ и сети.
- возможность регулирования скорости электродвигателя с помощью ПИД-регулятора или внешним заданием частоты.
- возобновление работы от ПЧ при пропадании и восстановлении питания.

4.35.3 Таблица входов-выходов

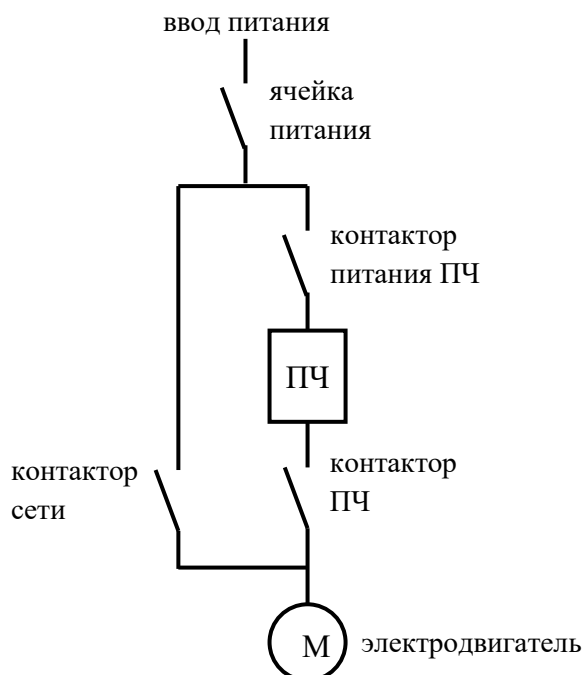


| Область | Название | Тип | Начальное значение | Назначение |
|--------------------|-------------------------|-----------------|--------------------|---|
| Input | plc_control_di | BOOL | TRUE | Состояние переключателя режима ручной/автомат (FALSE - ручной, TRUE - автомат) |
| | fr_busy_di | BOOL | FALSE | ПЧ занят (сигнал от других блоков) |
| | cont_fr_lock_di | BOOL | FALSE | Запрет включения контактора ПЧ в любом режиме (сигнал от других блоков) |
| | supply_di | BOOL | TRUE | Наличие напряжения на вводе питания ПЧ |
| | sync_on_net_di | BOOL | FALSE | Идет процесс синхронизации на сеть (сигнал от других блоков) |
| | sync_on_fr_di | BOOL | FALSE | Не используется |
| | cont_fr_supply_start_di | BOOL | FALSE | Не используется |
| | cont_fr_supply_stop_di | BOOL | FALSE | Не используется |
| | cont_net_start_di | BOOL | FALSE | Команда от ПЧ включить контактор сети (для ПЧ с поддержкой синхронизации на сеть) |
| | cont_net_stop_di | BOOL | FALSE | Не используется |
| | cont_fr_start_di | BOOL | FALSE | Не используется |
| | cont_fr_stop_di | BOOL | FALSE | Команда от ПЧ отключить контактор ПЧ (для ПЧ с поддержкой синхронизации на сеть) |
| | ac | udt_analog | | Ток двигателя при работе от сети. Вход для передачи интерфейса аналогового входа, который измеряет ток двигателя. |
| | fr_dbs | WORD | | Номера DB объектов. Используется только для платформы Siemens для корректного отображения данных на панели оператора. |
| | cont_supply_dbs | WORD | | |
| cont_fr_supply_dbs | WORD | | | |
| cont_net_dbs | WORD | | | |
| cont_fr_dbs | WORD | | | |
| Inout | hmi | udt_engine | | интерфейс блока |
| | fr | udt_fr | | интерфейс блока ПЧ |
| | cont_supply | udt_switch_gear | | интерфейс блока общей ячейки питания (от ПЧ и сети) |
| | cont_net | udt_contactor | | интерфейс блока контактора сети |
| | cont_fr_supply | udt_contactor | | интерфейс блока контактора питания ПЧ |
| | cont_fr | udt_contactor | | интерфейс блока контактора ПЧ |
| Output | fr_busy_do | BOOL | FALSE | ПЧ занят (сигнал другим блокам) |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------------|------|--------------------|---|
| | cont_fr_lock_do | BOOL | FALSE | Запрет включения контактора ПЧ в любом режиме (сигнал другим блокам) |
| | fr_stop_alarm_do | BOOL | FALSE | Не используется |
| | supply_on_do | BOOL | FALSE | Не используется |
| | sync_on_net_do | BOOL | FALSE | Идет процесс синхронизации на сеть (сигнал для других блоков, а также для подачи команды запуска синхронизации на ПЧ) |
| | sync_on_fr_do | BOOL | FALSE | Не используется |
| | cont_fr_supply_on_do | BOOL | FALSE | Не используется |
| | cont_net_on_do | BOOL | FALSE | Не используется |
| | cont_fr_on_do | BOOL | FALSE | Не используется |

4.35.4 Принцип работы

Схема коммутации показана на рисунке.



Блок может выполнять следующие операции:

- запуск электродвигателя от сети (с помощью контактора сети и общей ячейки питания);
- запуск электродвигателя от ПЧ (с помощью ячейки питания, контактора питания ПЧ, ПЧ и контактора ПЧ);
- переключение электродвигателя с ПЧ на сеть (с синхронизацией фаз и напряжений сети и выхода ПЧ);
- регулирование частоты вращения электродвигателя как вручную подачей фиксированного задания частоты через интерфейсную структуру hmi, так и с помощью ПИД-регулятора, встроенного в блок ПЧ;
- возобновление работы от ПЧ при пропадании и восстановлении питания.

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_switch_gear (ячейка питания), fb_contactor (контактор сети, контактор питания ПЧ, контактор ПЧ) и fb_fr (ПЧ).

Готовность к пуску

Блок имеет несколько состояний готовности:

- «готовность к пуску от сети»;
- «готовность к пуску от ПЧ»;
- «готовность к пуску».

«Готовность к пуску от сети» означает, что контактор сети в состоянии готовности и в режиме «автомат», ячейка отключена и в режиме «автомат», контактор ПЧ отключен. При этом другие объекты

(контактор питания ПЧ, ПЧ или контактор ПЧ) либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску от ПЧ» означает, что ячейка питания, контактор питания ПЧ, ПЧ и контактор ПЧ в состоянии готовности и в режиме «автомат», а также контактор сети отключен. При этом контактор сети либо не в режиме «автомат», либо не в состоянии готовности.

«Готовность к пуску» означает, что все объекты (ячейка питания, контактор сети, контактор питания ПЧ, ПЧ, контактор ПЧ) находятся в состоянии готовности и в режиме «автомат».

Работа от сети

Для запуска электродвигателя от сети необходимо, чтобы блок был либо в состоянии «готовность к пуску от сети», либо «готовность к пуску».

При подаче команды «пуск от сети» (start_net) будут включены ячейка питания, контактор сети, блок перейдет в состояние «работа от сети».

При подаче команды «стоп» (stop) контактор сети и ячейка питания будут отключены, блок перейдет в состояние «остановлен».

Работа от ПЧ

Для запуска электродвигателя от ПЧ необходимо, чтобы блок был либо в состоянии «готовность к пуску от ПЧ», либо «готовность к пуску».

При подаче команды «пуск от ПЧ» (start_fr) будут выполнены следующие действия:

- включение ячейки питания;
- включение контактора питания ПЧ;
- ожидание готовности ПЧ;
- включение контактора ПЧ (подключение электродвигателя к выходу ПЧ);
- включение ПЧ.

Блок перейдет в состояние «работа от ПЧ».

При подаче команды «стоп» (stop) будут выполнены следующие действия:

- плавный останов ПЧ со снижением скорости вращения электродвигателя до нуля;
- отключение контактора ПЧ (отключение электродвигателя от выхода ПЧ);
- отключение контактора питания ПЧ;
- отключение общей ячейки питания.

Блок перейдет в состояние «готовность к пуску от ПЧ» либо «готовность к пуску».

Переключение с ПЧ на сеть

Для переключения электродвигателя с ПЧ на сеть во время работы от ПЧ нужно подать команду «reconnect_fr_net».

Начнется процесс синхронизации фаз и напряжений сети и выхода ПЧ. Выход блока «sync_on_net_do» будет установлен в значение «TRUE». Это сигнал для ПЧ начать процесс синхронизации фаз и напряжений своего выхода с сетью.

Когда синхронизация будет достигнута, ПЧ выдаст сигнал о достижении синхронизации, который придет на вход «sync_on_net_di». Далее ПЧ выдаст сигнал отключения контактора ПЧ, который придет на вход «cont_fr_stop_di».

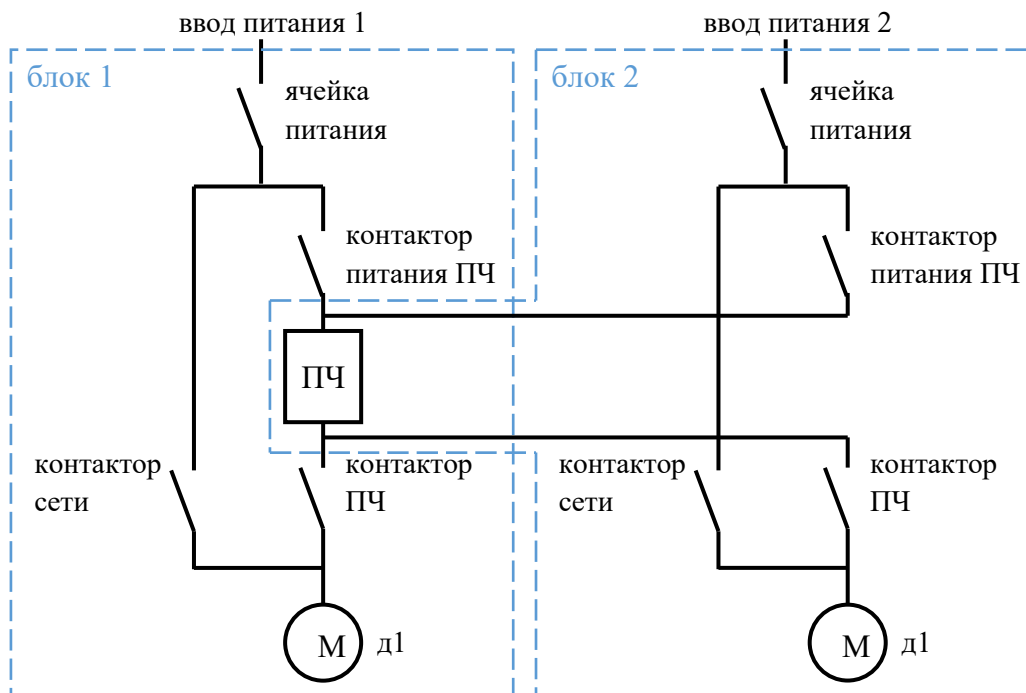
Как только входы «sync_on_net_di» и «cont_fr_stop_di» установятся в значение «TRUE», будет отключен контактор ПЧ и включен контактор сети. Электродвигатель перейдет на работу от сети.

Далее будет выполнено отключение ПЧ и отключение контактора питания ПЧ. Блок перейдет в состояние «работа от сети».

Совместное использование нескольких блоков. Один ПЧ на несколько электродвигателей

На рисунке показана схема включения двух электродвигателей, которые могут каждый независимо включаться прямым пуском от сети, но имеют один общий ПЧ. Электродвигателей может быть и больше, чем два.

Пунктирной линией выделены объекты, которые относятся к каждому из блоков коммутации электродвигателей. В данном случае ПЧ общий на оба блока.



Для того, чтобы блоки коммутации могли корректно управлять разделяемым ресурсом – ПЧ, предусмотрены следующие входные и выходные сигналы:

- «fr_busy_di»: вход; сигнал от других блоков коммутации, что в данный момент ПЧ занят ими;
- «fr_busy_do»: выход; сигнал другим блокам коммутации, что в данный момент ПЧ занят данным блоком.
- «cont_fr_lock_di»: вход; сигнал от других блоков коммутации, что требуется запретить запуск контактора ПЧ, поскольку включен контактор ПЧ другого блока.
- «cont_fr_lock_do»: выход; сигнал другим блокам коммутации, чтобы запретили запуск своих контакторов ПЧ, поскольку включен контактор ПЧ данного блока.

Таким образом, при вызове блоков в программе нужно на вход «fr_busy_di» подать сигналы «fr_busy_do» от других блоков, и на вход «cont_fr_lock_di» подать сигналы «cont_fr_lock_do» от других блоков. Если ко входу нужно привязать более, чем один сигнал, нужно объединить их логическим «или».

Возобновление работы электродвигателя от ПЧ при пропадании и восстановлении питания

Блок поддерживает функцию возобновления работы электродвигателя от ПЧ при пропадании и последующем восстановлении питания. Сигнал наличия напряжения на вводе питания ПЧ нужно привязать ко входу «supply_di» (сигнал может быть взят с реле контроля фаз, например).

Если во время работы двигателя от ПЧ питание пропало, а затем восстановилось в течение заданного времени, будет произведено повторное включение контактора питания ПЧ, контактора ПЧ и запуск ПЧ.

4.35.5 Интерфейс блока (udt_engine)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|----------|---------------------|-------------|------------|---|
| std | <u>udt_standard</u> | | ввод-вывод | стандартный набор переменных |
| out_ac | REAL | | вывод | Ток двигателя. Значение передается с блока ПЧ в состоянии «Работа от ПЧ». В состоянии «Работа от сети» значение передается со входа «ас». |
| out_freq | REAL | | вывод | Частота двигателя. Значение передается с блока ПЧ. |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|------------------------|----------------------|-------------|------------|---|
| out_speed | REAL | | вывод | Частота вращения двигателя, об/мин. Расчетное значение на основе текущей частоты ПЧ (out_freq) и максимальной частоты вращения (max_speed): $out_speed = (out_freq/50)*max_speed$. |
| ready_net | WORD | | вывод | Готовность к пуску от сети. Бит 0 – общая готовность к пуску от сети. Если бит взведен, блок готов к пуску. Если нет, то нужно посмотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – ячейка питания готова (отключена или включена). Бит 2 – контактор сети отключен. Бит 3 – контактор ПЧ отключен. Бит 2 – контактор питания ПЧ отключен. Бит 3 – контактор сети в режиме «автомат». |
| ready_fr | WORD | | вывод | Готовность к пуску от ПЧ. Бит 0 – общая готовность от ПЧ. Если бит взведен, блок готов к пуску. Если нет, то нужно посмотреть причину отсутствия готовности в следующих битах. Если готовности по какому-то условию нет, то бит будет сброшен. Бит 1 – ячейка питания готова (отключена или включена). Бит 2 – контактор сети отключен. Бит 3 – контактор ПЧ отключен. Бит 4 – контактор питания ПЧ отключен. Бит 5 – контактор ПЧ в режиме «автомат». Бит 6 – контактор питания ПЧ в режиме «автомат». Бит 7 – ПЧ в режиме «автомат». Бит 8 – ПЧ не занят. Бит 9 – Нет запрета включения контактора ПЧ (от других блоков). |
| pid | <u>udt_pid</u> | | ввод-вывод | Набор параметров регулятора. Если блок в работе, этот набор параметров передается в блок ПЧ. |
| ac_settings | <u>udt_engine_ac</u> | | ввод | Настройки токов для передачи в ПЧ. Если блок в работе, этот набор параметров передается в блок ПЧ. |
| type_connect | INT | | вывод | Тип подключения двигателя. Служебный параметр для использования другими блоками. |
| freq_low | REAL | 5 | ввод | Минимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_hi | REAL | 50 | ввод | Максимальная частота. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_rising | REAL | 5 | ввод | Скорость нарастания частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_lowering | REAL | 5 | ввод | Скорость понижения частоты. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_overload_lowering | REAL | 5 | ввод | Скорость понижения частоты при перегрузке по току. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| freq_deadband | REAL | 1 | ввод | Допустимое отклонение частоты при разгоне и торможении. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_freq_min_wait | REAL | 10 | ввод | Время ожидания набора минимальной частоты, сек. Значение для передачи в блок ПЧ. См. описание блока fb_fr. |
| time_supply | REAL | 10 | ввод | Время ожидания возвращения напряжения на вводе, сек. При пропадании напряжения на вводе питания ПЧ (supply_di = FALSE) блок удерживает состояние «в работе». Если напряжение на вводе вернулось |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-------------------|------|-------------|------------|--|
| | | | | (supply_di = TRUE) в течение этого времени, осуществляется перезапуск контактора питания ПЧ, контактора ПЧ и ПЧ. Если напряжение не вернулось, блок переходит в состояние аварии. |
| time_ready_fr | REAL | 5 | ввод | Время ожидания готовности ПЧ, сек. Если сигнал готовности ПЧ не пришел в течение этого времени после включения контактора питания, блок переходит в состояние аварии. |
| time_stop_cont_fr | REAL | 60 | ввод | Не используется. |
| max_speed | INT | 3000 | ввод | Максимальная частота вращения двигателя, об/мин. Используется для расчета текущей частоты вращения out_speed. |

4.35.6 Переменные интерфейса блока. Списки значений

4.35.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_cmd_engine» п. 3.3.3.4. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|------------------|------|----------|--|
| stop | WORD | 16#1 | Стоп. |
| net_start | WORD | 16#2 | Пуск от сети. |
| fr_start | WORD | 16#4 | Пуск от ПЧ. |
| reconnect_fr_net | WORD | 16#8 | Переключение с ПЧ на сеть. |
| stop_fast | WORD | 16#10 | Быстрый останов. Сначала останавливает ПЧ без плавного снижения частоты. На ПЧ сразу передается задание частоты = 0. Затем выключает контактор ПЧ, контактор питания ПЧ, ячейку питания. |
| stop_alarm | WORD | 16#20 | Аварийный останов. Подаются команды на отключение ячейки питания, контактора ПЧ, контактора питания ПЧ и быстрый останов ПЧ одновременно. |

4.35.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_alm_engine» п. 3.3.3.1. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|---|
| net_out_s1 | WORD | 16#1 | «Авария контактора(ячейки) сети». Блок контактора сети перешел в состояние «авария». |
| fr_in_s1 | WORD | 16#2 | «Авария контактора(ячейки) питания ПЧ». Блок контактора питания ПЧ перешел в состояние «авария». |
| fr_out_s1 | WORD | 16#4 | «Авария контактора(ячейки) ПЧ». Блок контактора ПЧ перешел в состояние «авария». |
| fr | WORD | 16#8 | «Авария ПЧ». Блок ПЧ перешел в состояние «авария». |
| fr_no_supply | WORD | 16#10 | «Пропало напряжение на вводе питания ПЧ». Вход supply_di отсутствовал в течение времени time_supply (см. интерфейс блока). |
| fr_not_ready | WORD | 16#20 | «Нет готовности ПЧ». ПЧ не пришел в состояние готовности после включения контактора питания в течение времени, заданного параметром «time_ready_fr» (см. интерфейс блока). |

| Название | Тип | Значение | Текст сообщения/Описание |
|----------|------|----------|--|
| in_s1 | WORD | 16#40 | «Авария общего контактора(ячейки) питания». Блок ячейки питания перешел в состояние «авария». |
| fr_busy | WORD | 16#80 | «ПЧ занят». Не используется. |

4.35.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_wrn_engine» п. 3.3.3.2. Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------------|------|----------|---|
| net_out_s1_not_ready | WORD | 16#1 | «Нет готовности контактора(ячейки) сети». Отсутствует состояние готовности блока контактора сети. |
| fr_in_s1_not_ready | WORD | 16#2 | «Нет готовности контактора(ячейки) питания ПЧ». Отсутствует состояние готовности блока контактора питания ПЧ. |
| fr_out_s1_not_ready | WORD | 16#4 | «Нет готовности контактора(ячейки) ПЧ». Отсутствует состояние готовности блока контактора ПЧ. |
| fr_not_ready | WORD | 16#8 | «Нет готовности ПЧ». Отсутствует состояние готовности блока ПЧ. |
| net_out_s1_not_auto | WORD | 16#10 | «Контактор(ячейка) сети не в режиме автомат». Режим блока контактора сети – не «автомат». |
| fr_in_s1_not_auto | WORD | 16#20 | «Контактор(ячейка) питания ПЧ не в режиме автомат». Режим блока контактора питания ПЧ – не «автомат». |
| fr_out_s1_not_auto | WORD | 16#40 | «Контактор(ячейка) ПЧ не в режиме автомат». Режим блока контактора ПЧ – не «автомат». |
| fr_not_auto | WORD | 16#80 | «ПЧ не в режиме автомат». Режим блока ПЧ – не «автомат». |
| fr_no_supply | WORD | 16#100 | «Пропало напряжение на вводе питания ПЧ». Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. интерфейс блока). |
| fr_wrn | WORD | 16#200 | «Предупреждение ПЧ». От блока ПЧ пришел сигнал «Предупреждение». |
| fr_alm | WORD | 16#400 | «Авария ПЧ». Блок ПЧ перешел в состояние аварии. |

4.35.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_cfg_engine» п. 3.3.3.3. Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|----------------|------|----------|---|
| auto_reconnect | WORD | 16#1 | Автоматическое переключение ПЧ/УПП на сеть после разгона. Не используется. |

4.35.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_hmi_state_engine» п. 3.3.3.6. Возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|--|
| not_ready | WORD | 16#1 | Ожидание готовности. Нет готовности одного из подчиненных блоков к пуску. |
| stopped | WORD | 16#2 | Готовность. Все подчиненные блоки готовы к пуску. |

| Название | Тип | Начальное значение | Описание |
|----------------|------|--------------------|---|
| net_starting | WORD | 16#4 | Запуск от сети. Идет процесс запуска электродвигателя от сети. Включаются ячейка питания и контактор сети. |
| net_started | WORD | 16#8 | В работе от сети. Ячейка питания и контактор сети включены, электродвигатель работает от сети. |
| net_stopping | WORD | 16#10 | Останов от сети. Идет процесс останова электродвигателя от сети. Отключается контактор сети и ячейка питания. |
| fr_starting | WORD | 16#20 | Запуск от ПЧ. Идет процесс запуска электродвигателя от ПЧ. Включаются ячейка питания, контактор питания ПЧ, контактор ПЧ. Далее включается ПЧ. |
| fr_started | WORD | 16#40 | В работе от ПЧ. Электродвигатель в работе – ячейка питания включена, контактор питания ПЧ включен, контактор ПЧ включен, ПЧ в работе. |
| fr_stopping | WORD | 16#80 | Останов от ПЧ. Идет процесс останова электродвигателя от ПЧ. Отключается ПЧ с плавным снижением частоты до нуля, затем отключается контактор ПЧ, далее отключается контактор питания ПЧ и ячейка питания. |
| fr_net | WORD | 16#100 | Переключение ПЧ->сеть. Идет процесс переключения электродвигателя с ПЧ на сеть. Отключается контактор ПЧ, включается контактор сети. Отключается ПЧ, затем контактор питания ПЧ. Затем блок переходит в состояние «В работе от сети». |
| alarm | WORD | 16#200 | Авария. Один из подчиненных блоков перешел в состояние аварии или отключился во время работы. |
| net_stopped | WORD | 16#400 | Готовность к запуску от сети. Контактор сети отключен и готов к пуску в режиме «автомат». Ячейка питания либо отключена и в режиме «автомат», либо включена. Контактор ПЧ отключен. |
| fr_stopped | WORD | 16#800 | Готовность к запуску от ПЧ. - ячейка питания либо отключена и в режиме «автомат», либо включена; - контактор питания ПЧ отключен и в режиме «автомат»; - ПЧ отключен и готов к пуску; - контактор ПЧ отключен и в режиме «автомат». |
| fr_no_supply | WORD | 16#1000 | Пропало напряжение на вводе. Вход supply_di принял значение FALSE, идет отсчет времени time_supply (см. интерфейс блока). |
| alarm_stopping | WORD | 16#2000 | Аварийный останов. Подана команда аварийного останова. Отключаются ПЧ, контактор ПЧ, контактор питания ПЧ, ячейка питания при работе двигателя от ПЧ (команды на их отключение подаются одновременно) или контактор сети и ячейка питания при работе двигателя от сети. |

4.35.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_hmi_state_actual_engine» п 3.3.3.7. Возможные значения представлены в таблице ниже.

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|---|
| power_off | WORD | 16#1 | Нет питания. ПЧ, контактор ПЧ, контактор питания ПЧ или ячейка питания находятся в состоянии «Нет питания» (std.out_state_actual = power_off), а также контактор сети или ячейка питания находятся в состоянии «Нет питания» (std.out_state_actual = power_off). |
| stopped | WORD | 16#2 | Отключен. |

| Название | Тип | Начальное значение | Описание |
|----------|------|--------------------|--|
| | | | ячейка питания, контактор питания ПЧ, ПЧ, контактор ПЧ находятся в состоянии «Отключен» (std.out_state_actual = stopped) или ячейка питания и ячейка сети находятся в состоянии «Отключен» (std.out_state_actual = stopped). |
| started | WORD | 16#4 | Включен. ячейка питания, конатктор питания ПЧ, ПЧ, контактор ПЧ находятся в состоянии «Включен» (std.out_state_actual = started) или ячейка питания и контактор сети в состоянии «Включен» (std.out_state_actual = started). |

4.35.7 Пример 1. Один электродвигатель.

Объявление глобальных переменных:

VAR_GLOBAL

```
// объявление интерфейса блока ячейка питания
dbs_cont_supply: RuDrive.udt_switch_gear;
// объявление интерфейса блока контактор сети
dbs_cont_net: RuDrive.udt_contactor;
// объявление интерфейса блока контактор питания ПЧ
dbs_cont_fr_supply: RuDrive.udt_contactor;
// объявление интерфейса блока контактор ПЧ
dbs_cont_fr: RuDrive.udt_contactor;
// объявление интерфейса блока ПЧ
dbs_fr: RuDrive.udt_fr;
// объявление интерфейса блока коммутации электродвигателя
dbs_engine: RuDrive.udt_engine;
```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM POU

VAR

```
// объявление экземпляра блока ячейка питания
dbi_cont_supply: RuDrive.fb_switch_gear;
// объявление экземпляра блока контактор сети
dbi_cont_net: RuDrive.fb_contactor;
// объявление экземпляра блока контактор питания ПЧ
dbi_cont_fr_supply: RuDrive.fb_contactor;
// объявление экземпляра блока контактор ПЧ
dbi_cont_fr: RuDrive.fb_contactor;
// объявление экземпляра блока ПЧ
dbi_fr: RuDrive.fb_fr;
// объявление экземпляра блока коммутации электродвигателя
dbi_engine: RuDrive.fb_eng_sc_scfc;
```

END_VAR

// вызовы экземпляров подчиненных блоков для простоты не показаны

// вызов экземпляра блока коммутации электродвигателя «Ячейка->Контактор + Ячейка->Контактор->ПЧ->Ячейка»

```
dbi_engine(
  plc_control_di := input_engine_plc_control_di,
  supply_di := input_engine_supply_di
  sync_on_net_di := input_engine_sync_on_net_di, // От ПЧ. Есть синхронизация
  sync_on_net_do => output_engine_sync_on_net_do, // К ПЧ. Пуск синхронизации
  cont_fr_stop_di := input_engine_cont_fr_stop_di, // От ПЧ. Отключить контактор ПЧ
  hmi := dbs_engine,
  fr := dbs_fr,
  cont_supply := dbs_cont_supply,
  cont_net := dbs_cont_net,
  cont_fr_supply := dbs_cont_fr_supply,
  cont_fr := dbs_cont_fr );
```

4.35.8 Пример 2. Два электродвигателя с одним общим ПЧ.

Объявление глобальных переменных:

VAR_GLOBAL

```
// объявление интерфейса блока ячейка питания
dbs_cont_supply_1: RuDrive.udt_switch_gear;
dbs_cont_supply_2: RuDrive.udt_switch_gear;
// объявление интерфейса блока контактор сети
dbs_cont_net_1: RuDrive.udt_contactor;
dbs_cont_net_2: RuDrive.udt_contactor;
// объявление интерфейса блока контактор питания ПЧ
dbs_cont_fr_supply_1: RuDrive.udt_contactor;
dbs_cont_fr_supply_2: RuDrive.udt_contactor;
// объявление интерфейса блока контактор ПЧ
dbs_cont_fr_1: RuDrive.udt_contactor;
dbs_cont_fr_2: RuDrive.udt_contactor;
// объявление интерфейса блока ПЧ
dbs_fr: RuDrive.udt_fr;
// объявление интерфейса блока коммутации электродвигателя
dbs_engine_1: RuDrive.udt_engine;
dbs_engine_2: RuDrive.udt_engine;
```

END_VAR

Вызов в циклической подпрограмме (цикл 100 мсек):

PROGRAM PLC_PRG

VAR

```
// объявление экземпляра блока ячейка питания
dbi_cont_supply_1: RuDrive.fb_switch_gear;
dbi_cont_supply_2: RuDrive.fb_switch_gear;
// объявление экземпляра блока контактор сети
dbi_cont_net_1: RuDrive.fb_contactor;
dbi_cont_net_2: RuDrive.fb_contactor;
// объявление экземпляра блока контактор питания ПЧ
dbi_cont_fr_supply_1: RuDrive.fb_contactor;
dbi_cont_fr_supply_2: RuDrive.fb_contactor;
// объявление экземпляра блока контактор ПЧ
dbi_cont_fr_1: RuDrive.fb_contactor;
dbi_cont_fr_2: RuDrive.fb_contactor;
// объявление экземпляра блока ПЧ
dbi_fr: RuDrive.fb_fr;
// объявление экземпляра блока коммутации электродвигателя
dbi_engine_1: RuDrive.fb_eng_sc_scfc;
dbi_engine_2: RuDrive.fb_eng_sc_scfc;
```

END_VAR

// вызовы экземпляров подчиненных блоков для простоты не показаны

// вызов экземпляра блока коммутации электродвигателя «Ячейка->Контактор + Ячейка->Контактор->ПЧ->Ячейка»

```
dbi_engine_1(
  plc_control_di := input_engine_1_plc_control_di,
  supply_di := input_engine_1_supply_di
  sync_on_net_di := input_fr_sync_on_net_di, // От ПЧ. Есть синхронизация
  sync_on_net_do => output_fr_sync_on_net_do, // К ПЧ. Пуск синхронизации
  cont_fr_stop_di := input_fr_cont_fr_stop_di, // От ПЧ. Откл контактор ПЧ
  fr_busy_di := dbi_engine_2.fr_busy_do,
  cont_fr_lock_di := dbi_engine_2.cont_fr_lock_do,
  hmi := dbs_engine_1,
  fr := dbs_fr,
  cont_supply := dbs_cont_supply_1,
  cont_net := dbs_cont_net_1,
  cont_fr_supply := dbs_cont_fr_supply_1,
  cont_fr := dbs_cont_fr_1 );
```

```
dbi_engine_2(
  plc_control_di := input_engine_2_plc_control_di,
```

```
supply_di := input_engine_2_supply_di
sync_on_net_di := input_fr_sync_on_net_di, // От ПЧ. Есть синхронизация
sync_on_net_do => output_fr_sync_on_net_do, // К ПЧ. Пуск синхронизации
cont_fr_stop_di := input_fr_cont_fr_stop_di, // От ПЧ. Откл контактор ПЧ
fr_busy_di := dbi_engine_1.fr_busy_do,
cont_fr_lock_di := dbi_engine_1.cont_fr_lock_do,
hmi := dbs_engine_2,
fr := dbs_fr,
cont_supply := dbs_cont_supply_2,
cont_net := dbs_cont_net_2,
cont_fr_supply := dbs_cont_fr_supply_2,
cont_fr := dbs_cont_fr_2 );
```

4.36 Преобразование интерфейса блоков контактор, ячейка, ПЧ, УПП к интерфейсу блока коммутации электродвигателя (fb_eng_c, fb_eng_s, fb_eng_f, fb_eng_u)

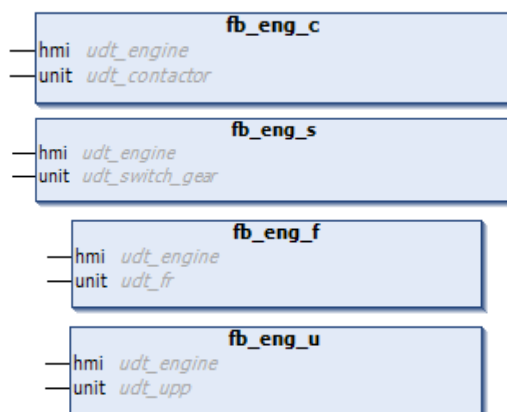
4.36.1 Назначение

Данная категория блоков является вспомогательной и предназначена для преобразования интерфейса блоков fb_contactor, fb_switch_gear, fb_fr, fb_урр к интерфейсу блоков коммутации электродвигателя (udt_engine).

4.36.2 Функции

- представление простых блоков для вышестоящих алгоритмов как блоков коммутации электродвигателя (например, для блока fb_rump).
- упрощение взаимодействия вышестоящих алгоритмов с простыми блоками (контактор, ячейка, ПЧ, УПП). Взаимодействие осуществляется через единый интерфейс (udt_engine).

4.36.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|---|--------------------|--|
| Inout | hmi | udt_engine | | интерфейс блока коммутации электродвигателя |
| | unit | udt_contactor (udt_switch_gear, udt_fr, udt_урр) | | интерфейс блока контактора (ячейки, ПЧ, УПП) |

4.36.4 Принцип работы

Блок преобразует следующие поля структуры hmi блоков контактора, ячейки, ПЧ, УПП к значениям, соответствующим блоку коммутации электродвигателя:

- режим; «std.out_mode»; из исходного блока (unit) режим записывается в блок коммутации (hmi) напрямую, без преобразования (hmi.std.out_mode := unit.std.out_mode).
- команда; «std.in_sau_cmd»;
- состояние; «std.out_state»;
- фактическое состояние; «std.out_state_actual»; из исходного блока (unit) фактическое состояние записывается в блок коммутации (hmi) напрямую, без преобразования (hmi.std.out_state_actual := unit.std.out_state_actual).

Таблица преобразования команд и состояний блоков приведена ниже.

| Контактор (fb_contactor) | Ячейка (fb_switch_gear) | ПЧ (fb_fr) | УПП (fb_upp) | Блок коммутации электродвигателя |
|---------------------------------|--|--|----------------------------|----------------------------------|
| Состояние unit.std.out_state => | | | | Состояние hmi.std.out_state |
| power_off | power_off ready_waiting grounded | power_off ready_waiting | power_off ready_waiting | not_ready |
| stopped | stopped | stopped | stopped | net_stopped |
| started | started | started regulation_ regulation_h fr_no_supply | started | net_started |
| stopping | stopping | stopping | stopping | net_stopping |
| starting | starting | starting starting_wait | starting | net_starting |
| alarm | alarm | alarm | alarm | alarm |
| Команда unit.std.in_sau_cmd | | | | <= Команда hmi.std.in_sau_cmd |
| stop | stop | stop | stop | stop |
| stop | stop | stop_fast | stop | stop_fast stop_alarm |
| start | start | start | start | net_start |
| reset_alarm | reset_alarm | reset_alarm | reset_alarm | reset_alarm |

4.36.5 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока контактор
    dbs_contactor:      RuDrive.udt_contactor;
    // объявление интерфейса блока коммутации электродвигателя
    dbs_engine:        RuDrive.udt_engine;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM POU
VAR
    // объявление экземпляра блока контактор
    dbi_contactor:      RuDrive.fb_contactor;
    // объявление экземпляра блока коммутации электродвигателя
    dbi_engine:        RuDrive.fb_eng_c;
END_VAR

// вызов экземпляра блока контактор
dbi_contactor(
    supply_di := input_contactor_supply_di,
    plc_control_di := input_contactor_plc_control_di,
    fault_di := input_contactor_fault_di,
    on_di := input_contactor_on_di,
    hmi := dbs_contactor,
    on_do => output_contactor_on_do,
    off_do => output_contactor_off_do,
    reset_do => output_contactor_reset_do);

// вызов блока преобразования интерфейса
dbi_engine(
    hmi := dbs_engine,
    unit := dbs_contactor);
```

4.37 Насосный агрегат (fb_pump)

4.37.1 Назначение

Данный блок предназначен для согласованного управления по заданному алгоритму следующими объектами:

- насосом (блоком коммутации электродвигателя);
- затвором на всасе насоса;
- затвором на напоре насоса.

Дополнительно в блок могут быть заведены следующие сигналы:

- датчик давления на всасе (для контроля сухого хода); датчик может быть дискретным (ЭКМ) или аналоговым;
- датчик давления на напоре (для контроля максимального и минимального выходного давления); датчик может быть дискретным (ЭКМ) или аналоговым;
- сигнал от технологической защиты (для останова насосного агрегата).

4.37.2 Функции

- запуск и останов насоса с открытием и закрытием затворов на всасе и напоре.
- защита насоса от сухого хода.
- защита насоса по максимальному и минимальному давлению на напоре.
- останов насоса по сигналу от технологической защиты.
- регулирование скорости вращения насоса (при наличии ПЧ в составе блока коммутации электродвигателя).

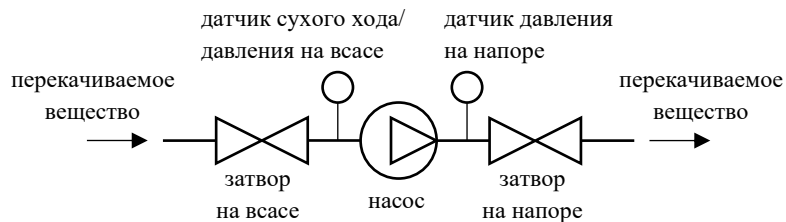
4.37.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|--------------|------------|--------------------|--|
| Input | protect | BOOL | | технологическая защита |
| | dry | BOOL | | минимальное давление на всасе (сухой ход) |
| | pressure_min | BOOL | | минимальное давление после насоса |
| | pressure_max | BOOL | | максимальное давление после насоса |
| | dry_ai | udt_analog | | датчик сухого хода/наличия давления на всасе |
| | p_ai | udt_analog | | датчик давления после насоса |
| Inout | hmi | udt_pump | | интерфейс блока |
| | valve_in | udt_valve | | интерфейс блока затвора на всасе |
| | valve_out | udt_valve | | интерфейс блока затвора на напоре |
| | engine | udt_engine | | интерфейс блока коммутации электродвигателя (насоса) |

4.37.4 Принцип работы

Технологическая схема насосного агрегата показана на рисунке.



Блок может выполнять следующие операции:

- запуск насоса;
- останов насоса;
- регулирование скорости вращения насоса (ПИД-регулятор, ручное задание оператором, внешнее задание от других блоков);
- отключение насоса по срабатыванию технологических защит;
- отключение насоса по сигналу сухого хода;
- отключение насоса по минимальному и максимальному давлению на напоре.

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_valve (затворы на всасе и напоре) и блок коммутации электродвигателя (любой блок fb_eng_[X] в зависимости от требуемой схемы коммутации электродвигателя насоса).

Готовность к пуску

Блок имеет несколько состояний готовности:

- «готовность к пуску от сети»;
- «готовность к пуску от ПЧ»;
- «готовность к пуску».

Для того, чтобы блок перешел в одно из этих состояний готовности, необходимо выполнение следующих условий:

1. Затвор на всасе в режиме «автомат», не в состояниях «авария» или «нет питания»; условие проверяется, если разрешено управление затвором в конфигурации (в параметре «std.in_config» интерфейса блока).
2. Затвор на напоре в режиме «автомат», не в состояниях «авария» или «нет питания»; условие проверяется, если разрешено управление затвором в конфигурации (в параметре «std.in_config» интерфейса блока).
3. Насос в режиме «автомат» и в одном из состояний:
 - «готовность к пуску от сети»;
 - «готовность к пуску от ПЧ»;
 - «готовность к пуску».

Состояние готовности блока насосного агрегата повторяет состояние готовности насоса, если остальные условия выполнены.

4. Нет срабатывания технологической защиты.
5. Нет срабатывания сухого хода; условие проверяется, если разрешено битом конфигурации в параметре «std.in_config» интерфейса блока.
6. Затвор на всасе не закрыт; условие проверяется, если не разрешено управление затвором в конфигурации (в параметре «std.in_config» интерфейса блока).

Запуск насосного агрегата

Запуск осуществляется в следующей последовательности:

1. Закрытие затвора на напоре; перед пуском затвор полностью закрывается, если он не закрыт; необходимость закрытия задается битом конфигурации в параметре «std.in_config» интерфейса блока.
2. Открытие затвора на всасе;
3. Приоткрытие затвора на напоре; время, в течение которого действует команда открытия, задается параметром «time_preopen» в интерфейсе блока;

4. Запуск насоса от сети или от ПЧ. При запуске от ПЧ запуск идет на фиксированной частоте (параметр «f_start» в интерфейсе блока);
5. Полное открытие затвора на напоре; ПЧ продолжает работать на частоте «f_start» до полного открытия; открытие затвора может идти в прерывистом режиме – импульс открытия/пауза, что бывает необходимо для некоторых процессов (замедление открытия задвижки); время импульса и паузы задается в настройках параметрами «time_pulse_open», «time_break_open» (см. интерфейс блока);
6. Переход блока в состояние «работа»; ПЧ переходит в режим регулирования частоты или работы от ручного или внешнего задания.

Работа насосного агрегата

Во время работы насосного агрегата контролируются следующие параметры:

- исправность насоса; если насос перейдет в состояние «авария», «нет питания» или «остановлен», насосный агрегат будет остановлен;
- срабатывание технологической защиты; если вход «protect» примет значение «TRUE», насосный агрегат будет остановлен;
- срабатывание защиты от сухого хода; защита может работать как по дискретному входу «dry», так и по аналоговому входу «dry_ai»; во втором случае защита срабатывает при формировании сообщения о достижении нижней аварийной границы аналогового входа; выбор входа типа входа («dry» или «dry_ai») зависит от конфигурации («std.in_config» интерфейса блока);
- срабатывание защит по минимальному и максимальному давлению на выходе насоса; защита может работать как по дискретным входам «pressure_min», «pressure_max», так и по аналоговому входу «p_ai»; во втором случае защита срабатывает при формировании сообщения о достижении нижней аварийной (минимальное давление) или верхней аварийной границы (максимальное давление) аналогового входа; выбор входа типа входа («pressure_min/ pressure_max» или «p_ai») зависит от конфигурации («std.in_config» интерфейса блока);

Останов насосного агрегата

Останов осуществляется в следующей последовательности:

1. Призакрытие затвора на напоре перед остановом; время, в течение которого действует команда закрытия, задается параметром «time_preclose» в интерфейсе блока; Если насос работает от ПЧ, частота ПЧ снижается до значения «f_start»;
2. Останов насоса. При работе от ПЧ выполняется плавный останов со снижением скорости до нуля;
3. Закрытие затвора на напоре; необходимость закрытия задается битом конфигурации в параметре «std.in_config» интерфейса блока.
4. Переход блока в состояние «готов к пуску».

4.37.5 Интерфейс блока (udt_pump)

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

| Название | Тип | Нач. знач-е | Назначение | Описание |
|----------------|--------------|-------------|------------|---|
| std | udt_standard | | ввод-вывод | Стандартный набор переменных |
| ready_to_start | WORD | | вывод | Условия готовности к пуску. Для готовности достаточно, чтобы один из битов 4 или 11 был установлен (насос готов к пуску либо от сети, либо от ПЧ). Остальные биты должны быть установлены все, кроме неиспользуемых. Бит 0 – затвор на всасе насоса в режиме «автомат». Бит 1 – затвор на напоре насоса в режиме «автомат». Бит 2 – насос в режиме «автомат». Бит 3 – затвор на всасе исправен (нет аварий, есть питание). Бит 3 – затвор на напоре исправен (нет аварий, есть питание). |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|-------------------|-----------------------|-------------|------------|---|
| | | | | Бит 4 – насос готов к пуску от сети. Бит 5 – не используется. Бит 6 – нет срабатывания технологических защит. Бит 7 – нет срабатывания защиты от сухого хода. Бит 8 – затвор на всасе приоткрыт (или в режиме «автомат», есть питание, нет аварий). Бит 9 – не используется. Бит 10 – не используется. Бит 11 – насос готов к пуску от ПЧ. |
| state_detail | udt_pump_state_detail | | вывод | детализированное состояние блока. Описание структуры см. таблицу ниже. |
| pid | udt_pid | | ввод-вывод | Набор параметров регулятора. См. п. Ошибка! Источник ссылки не найден.. |
| time_dry | REAL | 10 | ввод | Задержка по сухому ходу, сек. Время задержки срабатывания защиты по сухому ходу после установки входа «dry» в значение «TRUE» или срабатывания нижней аварийной границы аналогового входа «dry_ai». |
| time_pressure_min | REAL | 10 | ввод | Задержка по минимальному давлению, сек. Время задержки срабатывания защиты по минимальному давлению после установки входа «pressure_min» в значение «TRUE» или срабатывания нижней аварийной границы аналогового входа «p_ai». |
| time_pressure_max | REAL | 10 | ввод | Задержка по максимальному давлению, сек. Время задержки срабатывания защиты по максимальному давлению после установки входа «pressure_min» в значение «TRUE» или срабатывания верхней аварийной границы аналогового входа «p_ai». |
| pre_open | REAL | 0 | ввод | Время приоткрытия затвора на напоре перед пуском, сек. |
| pre_close | REAL | 0 | ввод | Время призакрытия затвора на напоре перед остановом, сек. |
| p_out_min | REAL | 0.5 | ввод | Минимальное давление после насоса. Уставка защиты по минимальному давлению. |
| p_out_max | REAL | 10 | ввод | Максимальное давление после насоса. Уставка защиты по максимальному давлению. |
| p_out_max_close | REAL | 8 | ввод | Максимальное рабочее давление перед остановом насоса. Давление по показаниям аналогового входа «p_ai», при превышении которого закрытие затвора на напоре прекращается, и насос останавливается при текущем положении затвора. |
| f_start | REAL | 5 | ввод | Выходная частота ПЧ перед открытием/закрытием затвора на напоре. При пуске насосного агрегата ПЧ работает на данном фиксированном значении частоты до полного открытия затвора на напоре. Данное значение также передается в качестве фиксированного задания частоты ПЧ при останове насосного агрегата. |
| type_connect | INT | | ввод | Тип подключения двигателя. Служебный параметр для использования другими блоками. |
| tag_name | STRING(16) | '' | ввод-вывод | Диспетчерское наименование. Параметр для вывода и отображения технологического наименования объекта на мнемосхеме. |
| time_pulse_open | REAL | 0 | ввод | Время импульса открытия затвора на напоре, сек. Используется при импульсном открытии затвора на напоре. Если время импульса = 0, то импульсное открытие отключено, затвор открывается как обычно. |
| time_break_open | REAL | 0 | ввод | Время паузы между импульсами открытия затвора на напоре, сек. |

| Название | Тип | Нач. знач-е | Назначение | Описание |
|----------|-----|-------------|------------|---|
| | | | | Используется при импульсном открытии затвора на напоре. Если время паузы = 0, то импульсное открытие отключено, затвор открывается как обычно. |

4.37.6 Переменные интерфейса блока. Списки значений

4.37.6.1 Команды «std.in_cmd»

Тип глобальных констант - «sys_db_pump_cmd» п. **Ошибка! Источник ссылки не найден.** Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|-----------|------|----------|---|
| stop | WORD | 16#1 | Стоп. |
| stop_fast | WORD | 16#2 | Быстрый останов. Останавливает насосный агрегат без плавного снижения частоты. На ПЧ (если он есть) сразу передается задание частоты = 0. |
| start_net | WORD | 16#4 | Пуск от сети. |
| start_fr | WORD | 16#8 | Пуск от ПЧ. |

4.37.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_pump_alarm» п. **Ошибка! Источник ссылки не найден.** Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|--------------|------|----------|---|
| valve_in | WORD | 16#1 | «Авария затвора на всасе». Затвор на всасе перешел в состояние «авария». |
| valve_out | WORD | 16#2 | «Авария затвора на напоре». Затвор на напоре перешел в состояние «авария». |
| pump | WORD | 16#4 | «Авария насоса». Насос (блок коммутации электродвигателя) перешел в состояние «авария». |
| pressure_min | WORD | 16#8 | «Низкое давление на напоре». Во время работы насосного агрегата давление на напоре снизилось до минимального значения. Вход «pressure_min» принял значения «TRUE» либо сработала нижняя аварийная граница аналогового входа «p_ai». |
| pressure_max | WORD | 16#10 | «Высокое давление на напоре». Во время работы насосного агрегата давление на напоре увеличилось до максимального значения. Вход «pressure_max» принял значения «TRUE» либо сработала верхняя аварийная граница аналогового входа «p_ai». |
| dry | WORD | 16#20 | «Сухой ход». Во время работы насосного агрегата давление на всасе снизилось до минимального значения. Вход «dry» принял значения «TRUE» либо сработала нижняя аварийная граница аналогового входа «dry_ai». |
| protection | WORD | 16#40 | «Срабатывание защиты». Вход «protect» принял значения «TRUE» (сработала технологическая защита). |

4.37.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_pump_warning» п. **Ошибка! Источник ссылки не найден.** Возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|-----------------|------|----------|--|
| valve_in_mode | WORD | 16#1 | «Затвор на всасе не в режиме автомат». |
| valve_out_mode | WORD | 16#2 | «Затвор на напоре не в режиме автомат». |
| pump_mode | WORD | 16#4 | «Насос не в режиме автомат». |
| valve_in_alarm | WORD | 16#8 | «Авария затвора на всасе». Затвор на всасе перешел в состояние «авария». |
| valve_out_alarm | WORD | 16#10 | «Авария затвора на напоре». Затвор на напоре перешел в состояние «авария». |
| pump_alarm | WORD | 16#20 | «Авария насоса». Насос (блок коммутации электродвигателя) перешел в состояние «авария». |
| protection | WORD | 16#40 | «Срабатывание защиты». Вход «protect» принял значение «TRUE» (сработала технологическая защита). |
| dry | WORD | 16#80 | «Сухой ход». Во время работы насосного агрегата давление на всасе снизилось до минимального значения. Вход «dry» принял значение «TRUE» либо сработала нижняя аварийная граница аналогового входа «dry_ai». Если блок в работе, идет отсчет времени до срабатывания аварии («time_dry»). |
| pressure_min | WORD | 16#100 | «Низкое давление на напоре». Во время работы насосного агрегата давление на напоре снизилось до минимального значения. Вход «pressure_min» принял значени «TRUE» либо сработала нижняя аварийная граница аналогового входа «p_ai». Если насос в работе, идет отсчет времени до срабатывания аварии («time_pressure_min»). |
| pressure_max | WORD | 16#200 | «Высокое давление на напоре». Во время работы насосного агрегата давление на напоре увеличилось до максимального значения. Вход «pressure_max» принял значени «TRUE» либо сработала верхняя аварийная граница аналогового входа «p_ai». Если насос в работе, идет отсчет времени до срабатывания аварии («time_pressure_max»). |
| valve_in_close | WORD | 16#400 | «Затвор на всасе закрыт». Нет готовности к пуску насосного агрегата, поскольку затвор на всасе закрыт. Перед пуском он должен быть приоткрыт. |

4.37.6.4 Конфигурация «std.in_config».

Тип глобальных констант - «sys_db_pump_cfg» п. **Ошибка! Источник ссылки не найден..** Возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|---------------------|------|----------|---|
| stop_min_pressure | WORD | 16#1 | Отключение насосного агрегата при минимальном давлении на напоре. Если данный бит конфигурации установлен, он разрешает работу защиты по минимальному давлению на напоре. Защита срабатывает по входу «pressure_min» (дискретный вход) или по срабатыванию нижней аварийной границы аналогового входа «p_ai». Выбор входа осуществляется битом конфигурации «p_out_type». |
| stop_max_pressure | WORD | 16#2 | Отключение насосного агрегата при максимальном давлении на напоре. Если данный бит конфигурации установлен, он разрешает работу защиты по максимальному давлению на напоре. Защита срабатывает по входу «pressure_max» (дискретный вход) или по срабатыванию верхней аварийной границы аналогового входа «p_ai». Выбор входа осуществляется битом конфигурации «p_out_type». |
| dry_protection_type | WORD | 16#4 | Датчик сухого хода (ОТКЛ - дискретный сигнал, ВКЛ - аналоговый сигнал). |

| Название | Тип | Значение | Описание |
|-------------------------|------|----------|--|
| | | | Если данный бит конфигурации установлен, то защита от сухого хода срабатывает по дискретному входу «dry», иначе она срабатывает по нижней аварийной границе аналогового входа «dry_ai». |
| dry_protection_on | WORD | 16#8 | Контроль сухого хода (наличия среды). Если данный бит конфигурации установлен, то защита от сухого хода активна. |
| p_out_type | WORD | 16#10 | Тип датчика давления на напоре (ОТКЛ – дискретный вход, ВКЛ - аналоговый вход). Если данный бит конфигурации установлен, то защиты по минимальному и максимальному давлению работают по аналоговому входу «p_ai», иначе по дискретным входам «pressure_min» и «pressure_max». |
| valve_in_control | WORD | 16#20 | Управление затвором на всасе. Если данный бит конфигурации установлен, разрешается управление затвором на всасе при запуске и останове насосного агрегата. Если бит снят, управление затвором не выполняется, контроль его аварий также не производится. |
| ignore_valve_alarm | WORD | 16#40 | Игнорирование аварии затворов. Если данный бит конфигурации установлен, то при переходе затвора на всасе или напоре в состояние «аварии» в процессе запуска или останова насосного агрегата его аварии игнорируются, процесс запуска или останова продолжается дальше. Работает только при включенных битах конфигурации «valve_in_control» и/или «valve_out_control». |
| valve_out_control | WORD | 16#80 | Управление затвором на напоре. Если данный бит конфигурации установлен, разрешается управление затвором на напоре при запуске и останове насосного агрегата. Если бит снят, управление затвором не выполняется, контроль его аварий также не производится. |
| stop_valve_p_out | WORD | 16#100 | Контроль максимального рабочего давления перед остановом насоса. Если данный бит конфигурации установлен, то в процессе призакрытия затвора на напоре перед остановом насоса контролируется давление на напоре по аналоговому входу «p_ai». Если значение давления превысит заданное параметром «p_out_max_close» (см. интерфейс блока), то закрытие затвора прекратится и произойдет переход к останову насоса. |
| valve_in_open_control | WORD | 16#200 | Контроль приоткрытого затвора на всасе. Если данный бит конфигурации установлен, то при формировании готовности насосного агрегата к пуску проверяется, чтобы затвор на всасе был приоткрыт. Условие работает, если отключено управление затвором на всасе «valve_in_control» (бит конфигурации снят). |
| valve_out_open_ready_fr | WORD | 16#400 | Разрешение запуска на открытый затвор на напоре для ПЧ. Если данный бит конфигурации установлен, то при запуске насоса закрытие затвора на напоре не выполняется, идет сразу запуск насоса. |
| valve_out_open_pressure | WORD | 16#800 | Открытие затвора на напоре при достижении минимального давления. Если бит конфигурации установлен, то в процессе запуска насоса на шаге открытия затвора на напоре команда открытия на затвор подается не сразу, а только при достижении минимального давления. Условие достижения мин. давления – дискретный вход «pressure_min» принимает значение «FALSE» (если бит конфигурации «p_out_type» снят) или значение аналогового входа «p_ai» превышает значение «p_out_min» (если бит конфигурации «p_out_type» установлен). |

4.37.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_pump_hmi_state» п. **Ошибка! Источник ссылки не найден..** В возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|------------------------|-------|--------------------|---|
| stop_not_ready | DWORD | 16#1 | Отключен, нет готовности к пуску. Не выполнены условия готовности насосного агрегата к пуску (см. параметр «ready_to_start» в интерфейсе блока). |
| stop_ready_net | DWORD | 16#2 | Отключен, готов к пуску от сети. Выполнены условия готовности насосного агрегата к пуску от сети (см. параметр «ready_to_start» в интерфейсе блока). |
| stop_ready_fr | DWORD | 16#4 | Отключен, готов к пуску от ПЧ. Выполнены условия готовности насосного агрегата к пуску от ПЧ (см. параметр «ready_to_start» в интерфейсе блока). |
| stop_ready | DWORD | 16#8 | Отключен, готов к пуску. Выполнены условия готовности насосного агрегата к пуску от сети и ПЧ (см. параметр «ready_to_start» в интерфейсе блока). |
| valve_out_close_before | DWORD | 16#10 | Затвор на напоре закрывается перед запуском насоса от сети/ПЧ. |
| valve_in_open_net | DWORD | 16#20 | Затвор на всасе открывается перед запуском насоса от сети. |
| valve_out_close_after | DWORD | 16#40 | Затвор на напоре закрывается после останова насоса. |
| valve_out_open_net | DWORD | 16#80 | Затвор на напоре открывается перед запуском насоса от сети. |
| starting_net | DWORD | 16#100 | Насос включается от сети. |
| starting_fr | DWORD | 16#200 | Насос включается от ПЧ. |
| stopping | DWORD | 16#400 | Насос останавливается. |
| started_net | DWORD | 16#800 | Насосный агрегат в работе от сети. |
| started_fr | DWORD | 16#1000 | Насосный агрегат в работе от ПЧ. |
| alarm | DWORD | 16#2000 | Аварийный останов. Выполняется аварийный останов насосного агрегата. Подаются команды закрытия на затворы, команда аварийного останова на насос. |
| valve_out_preopen_net | DWORD | 16#4000 | Приоткрытие затвора на напоре перед стартом от сети. |
| valve_out_preclose | DWORD | 16#8000 | Призакрытие затвора на напоре перед остановом. |
| valve_in_open_fr | DWORD | 16#10000 | Затвор на всасе открывается перед запуском насоса от ПЧ. |
| valve_out_open_fr | DWORD | 16#20000 | Затвор на напоре открывается после запуска насоса от ПЧ. |
| valve_out_preopen_fr | DWORD | 16#40000 | Приоткрытие затвора на напоре перед запуском насоса от ПЧ. |

4.37.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_pump_hmi_state_actual» п. **Ошибка! Источник ссылки не найден..** В возможные значения представлены в таблице ниже.

Фактическое состояние насосного агрегата полностью копируется из состояния подчиненного блока коммутации электродвигателя (hmi.std.out_state_actual = engine.std.out_state_actual).

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|-----------------|
| power_off | WORD | 16#1 | Нет готовности. |
| stopped | WORD | 16#2 | Остановлен. |
| started | WORD | 16#4 | Работа. |

4.37.7 Пример

Объявление глобальных переменных:

`VAR_GLOBAL`

```

// объявление интерфейса блока затвор на всасе
dbs_valve_in:      RuDrive.udt_valve;
// объявление интерфейса блока затвор на напоре
dbs_valve_out:    RuDrive.udt_valve;
// объявление интерфейса блока коммутации электродвигателя
dbs_engine:      RuDrive.udt_engine;
// объявление интерфейса блока насосный агрегат
dbs_pump:        RuDrive.udt_pump;
END_VAR

```

Вызов в циклической подпрограмме (цикл 100 мсек):

```

PROGRAM PLC_PRG
VAR
    // объявление экземпляра блока затвор на всасе
    dbi_valve_in:      RuDrive.fb_valve;
    // объявление экземпляра блока затвор на напоре
    dbi_valve_out:    RuDrive.fb_valve;
    // объявление экземпляра блока коммутации электродвигателя (может быть любой
fb_eng_[X] в зависимости от необходимой схемы)
    dbi_engine:      RuDrive.fb_eng_c_fc;
    // объявление экземпляра блока насосный агрегат
    dbi_pump:        RuDrive.fb_pump;
END_VAR

// вызов экземпляра блока затвор на всасе
dbi_valve_in(
    supply_di := valve_in_supply_di,
    plc_control_di := valve_in_plc_control_di,
    fault_di := valve_in_fault_di,
    open_di := valve_in_open_di,
    close_di := valve_in_close_di,
    hmi := dbs_valve_in,
    open_do => valve_in_open_do,
    close_do => valve_in_open_do );

// вызов экземпляра блока затвор на напоре
dbi_valve_out(
    supply_di := valve_out_supply_di,
    plc_control_di := valve_out_plc_control_di,
    fault_di := valve_out_fault_di,
    open_di := valve_out_open_di,
    close_di := valve_out_close_di,
    hmi := dbs_valve_out,
    open_do => valve_out_open_do,
    close_do => valve_out_open_do );

// вызов экземпляра блока коммутации электродвигателя "Контактор + ПЧ->Контактор"
// вызовы входящих в его состав блоков для простоты не приведены
dbi_engine(
    plc_control_di := input_engine_plc_control_di,
    supply_di := input_engine_supply_di,
    hmi := dbs_engine,
    fr := dbs_fr,
    cont_net := dbs_cont_net,
    cont_fr := dbs_cont_fr );

// вызов экземпляра блока насосного агрегата
dbi_pump(
    protect := pump_protect,
    dry := pump_dry,
    pressure_min := pump_pressure_min,
    pressure_max := pump_pressure_max,
    dry_ai := pump_dry_ai,
    p_ai := pump_p_ai,
    hmi := dbs_pump,
    valve_in := dbs_valve_in,
    valve_out := dbs_valve_out,
    engine := dbs_engine );

```

4.38 Преобразование интерфейса блоков контакторов, ПЧ к интерфейсу блока насосного агрегата (fb_pump_c, fb_pump_f)

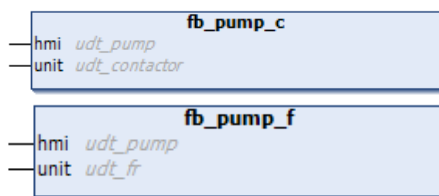
4.38.1 Назначение

Данная категория блоков является вспомогательной и предназначена для преобразования интерфейса блоков fb_contactor (udt_conactor), fb_fr (udt_fr) к интерфейсу блока насосного агрегата (udt_pump).

4.38.2 Функции

- представление простых блоков для вышестоящих алгоритмов как блока насосного агрегата (например, для блока fb_pump_sau).
- упрощение взаимодействия вышестоящих алгоритмов с простыми блоками (контактор, ПЧ). Взаимодействие осуществляется через единый интерфейс (udt_pump).

4.38.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|---------------------------|--------------------|------------------------------------|
| Inout | hmi | udt_pump | | интерфейс блока насосного агрегата |
| | unit | udt_contactor (udt_fr) | | интерфейс блока контактора (ПЧ) |

4.38.4 Принцип работы

Блок преобразует следующие поля структуры hmi блоков контактора, ПЧ к значениям, соответствующим блоку насосного агрегата:

- режим; «std.out_mode»; из исходного блока (unit) режим записывается в блок коммутации (hmi) напрямую, без преобразования (hmi.std.out_mode := unit.std.out_mode).
- команда; «std.in_sau_cmd»;
- состояние; «std.out_state»;
- фактическое состояние; «std.out_state_actual»; из исходного блока (unit) фактическое состояние записывается в блок коммутации (hmi) напрямую, без преобразования (hmi.std.out_state_actual := unit.std.out_state_actual).

Таблица преобразования команд и состояний блоков приведена ниже.

| Контактор (fb_contactor) | | ПЧ (fb_fr) | | Блок насосного агрегата |
|-----------------------------|--|---|--|------------------------------------|
| | | Состояние unit.std.out_state => | | Состояние hmi.std.out_state |
| power_off | | power_off ready_waiting | | stop_not_ready |
| stopped | | - | | stop_ready_net |
| - | | stopped | | stop_ready_fr |
| started | | - | | started_net |
| - | | started regulation_l | | started_fr |

| Контактор (fb_contactor) | | ПЧ (fb_fr) | | Блок насосного агрегата |
|------------------------------------|--|------------------------------|--|---|
| | | regulation_h fr_no_supply | | |
| stopping | | stopping | | stopping |
| starting | | - | | starting_net |
| - | | starting starting_wait | | starting_fr |
| alarm | | alarm | | alarm |
| Команда unit.std.in_sau_cmd | | | | <= Команда hmi.std.in_sau_cmd |
| stop | | stop | | stop |
| stop | | stop_fast | | stop_fast |
| start | | - | | start_net |
| - | | start | | start_fr |
| reset_alarm | | reset_alarm | | reset_alarm |

4.38.5 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
    // объявление интерфейса блока контактор
    dbs_contactor: RuDrive.udt_contactor;
    // объявление интерфейса блока насосного агрегата
    dbs_engine: RuDrive.udt_pump;
END_VAR
```

Вызов в циклической подпрограмме (цикл 100 мсек):

```
PROGRAM POU
VAR
    // объявление экземпляра блока контактор
    dbi_contactor: RuDrive.fb_contactor;
    // объявление экземпляра блока преобразования интерфейса
    dbi_pump: RuDrive.fb_pump_c;
END_VAR

// вызов экземпляра блока контактор
dbi_contactor(
    supply_di := input_contactor_supply_di,
    plc_control_di := input_contactor_plc_control_di,
    fault_di := input_contactor_fault_di,
    on_di := input_contactor_on_di,
    hmi := dbs_contactor,
    on_do => output_contactor_on_do,
    off_do => output_contactor_off_do,
    reset_do => output_contactor_reset_do);

// вызов блока преобразования интерфейса
dbi_pump(
    hmi := dbs_pump,
    unit := dbs_contactor);
```

4.39 Насосная станция (fb_pump_sau)

4.39.1 Назначение

Данный блок предназначен для согласованного управления набором блоков типа «насосный агрегат» (fb_pump), которое позволяет регулировать производительность станции подключением и отключением насосных агрегатов, а также обеспечивать включение резервного насосного агрегата при отключении работающего.

4.39.2 Функции

- регулирование производительности насосной станции как изменением скорости вращения работающих от ПЧ агрегатов, так и подключением/отключением дополнительных насосных агрегатов.
- автоматический ввод резерва (АВР) – включение резервного агрегата при аварийном отключении работающего.
- автоматическая/ручная ротация насосных агрегатов – периодическое отключение работающего насосного агрегата и включение резервного.

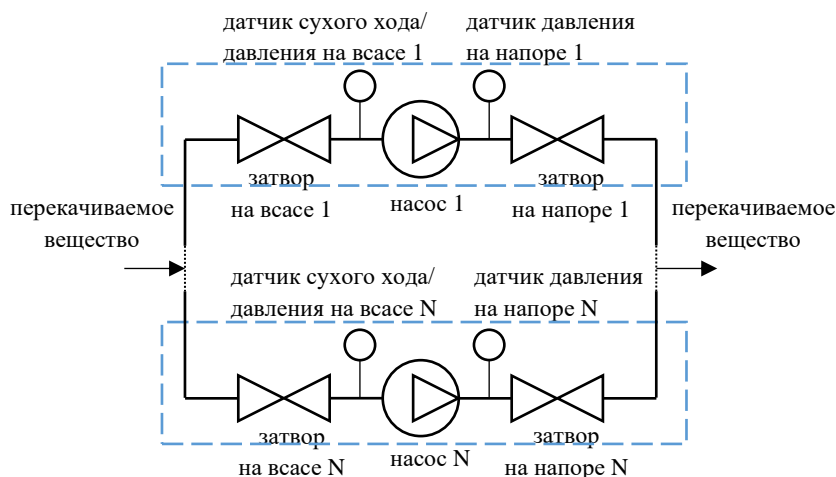
4.39.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------|--------------|--------------------|----------------------------------|
| Input | ai_1 | udt_analog | | регулируемый параметр 1 |
| | ai_2 | udt_analog | | регулируемый параметр 2 (резерв) |
| | alarm_stop_btn | BOOL | | кнопка аварийного останова |
| Inout | hmi | udt_pump_sau | | интерфейс блока |

4.39.4 Принцип работы

Технологическая схема насосной станции показана на рисунке.



Блок насосной станции может управлять от 1 до 5 блоками насосных агрегатов. В составе насосной станции могут быть как насосы с ПЧ, так и насосы с прямым пуском от сети или УПП. Их порядок и тип включения могут быть произвольными.

Если в составе станции есть хотя бы один насос с ПЧ, то станция может обеспечить плавное регулирование производительности.

Для подачи регулируемого параметра на вход блока насосной станции предусмотрены аналоговые входы «ai_1» (основной) и «ai_2» (резервный). Выбор входа осуществляется битом конфигурации в параметре «std.in_config».

Дискретный вход «alarm_stop_btn» предназначен для подключения к блоку сигнала от кнопки аварийного останова станции (или другого сигнала останова, например, от общестанционных технологических защит).

Блок может выполнять следующие операции:

- регулирование производительности насосной станции изменением скорости вращения насоса, если в составе станции есть хотя бы один насос с ПЧ; встроенный ПИД-регулятор осуществляет формирование задания производительности для насосов, используя в качестве обратной связи вход «ai_1» (или «ai_2»);
- регулирование производительности насосной станции включением и отключением насосных агрегатов;
- автоматический ввод резерва (АВР) – включение резервного насосного агрегата при аварийном отключении работающего;
- автоматическая/ручная ротация насосных агрегатов - периодическое отключение работающего насосного агрегата и включение резервного;
- регулирование параметра, изменение которого связано с производительностью насосной станции (например, давление/расход на входе или выходе станции).

Для работы блока необходимы предварительно объявленные и вызываемые в циклической программе блоки fb_pump (насосный агрегат) и требуемые для их работы блоки.

Готовность к пуску

Станция придет в состояние «готовность» при следующих условиях:

- число готовых к пуску насосных агрегатов («готов к пуску», «готов к пуску от сети», «готов к пуску от ПЧ») больше или равно значению параметра «pump_number_min» (минимальное число одновременно работающих агрегатов, см. интерфейс блока);
- вход «alarm_stop_btn» имеет значение «FALSE» (нет сигнала аварийного останова).

Запуск станции

Запуск станции после подачи команды «пуск» выполняется в следующей последовательности:

1. Выбор насосного агрегата для запуска в зависимости от настроек. Если запуск агрегатов по наработке выключен (бит «time_operation_enable» в «std.in_config»), то выбирается агрегат с наименьшим приоритетом, иначе выбирается агрегат с наименьшей наработкой.
2. На выбранный агрегат подается команда «пуск» и ожидается его переход в состояние «работа».
3. Если минимальное число работающих агрегатов достигнуто (параметр «pump_number_min»), то насосная станция переходит в состояние «работа». Если нет, то возвращается к п.1.

Работа станции

В состоянии «работа» насосная станция выполняет следующие действия:

1. Подключает дополнительные и отключает лишние насосные агрегаты, если это разрешено в конфигурации (бит «load_unload_enable» в «std.in_config»).

Условия подключения дополнительного агрегата:

- текущее значение регулируемого параметра (pv) меньше задания (sp) на величину, задаваемую параметром «delta_sp_pv_switch_on»; это условие становится обратным, если в конфигурации включена инверсия регулятора (бит «pid_inverse» в «std.in_config»);

- все насосы, работающие от ПЧ, достигли максимального значения частоты.

Если условия выполнены, то после выдержки времени «time_switch_on» подключается дополнительный насосный агрегат.

Условия отключения лишнего агрегата:

- текущее значение регулируемого параметра (pv) больше задания (sp) на величину, задаваемую параметром «delta_sp_pv_switch_off»; это условие становится обратным, если в конфигурации включена инверсия регулятора (бит «pid_inverse» в «std.in_config»);
- один из насосов, работающих от ПЧ, достиг минимального значения частоты.

Если условия выполнены, то после выдержки времени «time_switch_off» отключается лишний насосный агрегат.

Если в составе станции нет насосных агрегатов с ПЧ, то условие достижения минимальной/максимальной частоты игнорируется.

2. Включает резервный насосный агрегат при аварийном отключении одного из работающих.
3. Выполняет ротацию насосных агрегатов либо при подаче команды ротации, либо автоматически по таймеру. Чтобы включить автоматическую ротацию, нужно взвести бит «auto_rotation_enable» в слове конфигурации «std.in_config».
4. При подаче значение «TRUE» на вход «alarm_stop_btn» останавливает все насосные агрегаты.
5. Если в процессе регулирования производительности регулируемый параметр (выбранный аналоговый вход «ai_1» или «ai_2») переходит в состояние «неисправность», выход регулятора «замораживается» на текущем значении. Регулирование продолжится при возврате аналогового входа в исправное состояние.

Останов станции

Останов станции после подачи команды «стоп» осуществляется в следующей последовательности:

1. Выбор насосного агрегата для останова в зависимости от настроек. Если бит «time_operation_enable» в «std.in_config» снят, то выбирается агрегат с наибольшим приоритетом, иначе выбирается агрегат с наибольшей наработкой.
2. На выбранный агрегат подается команда «стоп» и ожидается его переход в состояние «готов к пуску» (или «авария», или «нет готовности»).
3. Если все насосные агрегаты остановлены, то насосная станция переходит в состояние «готовность». Если нет, то возвращается к п.1.

4.39.5 Интерфейс блока (udt_pump_sau)

В следующей таблице приведен интерфейс блока, представленный типом данных «udt_pump_sau».

| Название | Тип | Нач. знач-е | Описание |
|-----------------|---------------------|-----------------|--|
| std | udt_standard | | Стандартный набор переменных |
| pid | udt_pid | | Набор параметров регулятора. См. п. Ошибка! Источник ссылки не найден.. |
| pump_number_min | INT | | Минимальное кол-во одновременно работающих агрегатов. Параметр указывает кол-во агрегатов, которые должны быть запущены в работу. Если число готовых к пуску агрегатов меньше этого значения, станция не перейдет в состояние готовности к пуску. |
| pump_number_max | INT | | Максимальное кол-во одновременно работающих агрегатов. Параметр ограничивает кол-во агрегатов, которые могут быть запущены в процессе регулирования производительности станции при подключении новых агрегатов. |
| pump_priority | ARRAY [1..5] OF INT | [1, 2, 3, 4, 5] | Приоритет насосного агрегата. Приоритет указывает порядок запуска насосных агрегатов. Агрегаты запускаются в порядке возрастания приоритета. Первым запускается агрегат с наименьшим приоритетом. Этот набор параметров игнорируется, если включена работа по наработке, т.е. взведен бит «time_operation_enable» в слове «std.in_config». |

| Название | Тип | Нач. знач-е | Описание |
|--------------------------|----------------------------|-------------|--|
| pump_time_operation | ARRAY [1..5] OF DINT | | Наработка насосного агрегата, ч. Используется для выбора насосного агрегата для включения и отключения, если включена работа по наработке (взведен бит «time_operation_enable» в слове «std.in_config»). Также используется для автоматической ротации насосных агрегатов. Если общее время работы агрегата превысило значение «time_rotation», то этот агрегат отключается, и вместо него включается резервный. Чтобы включить автоматическую ротацию, нужно установить бит «auto_rotation_enable» в слове «std.in_config». |
| pump_ready | ARRAY [1..2, 1..5] OF BOOL | | Готовность к пуску насосного агрегата. Массив значений, который содержит статус готовности каждого агрегата к пуску от сети, УПП или от ПЧ (TRUE – есть готовность, FALSE - нет). Первый индекс задает линию, по которой будет включаться блок коммутации электродвигателя (линия 1 или 2), второй – номер насосного агрегата (1...5). Обычно линия 1 – это пуск от ПЧ/УПП, а линия 2 – пуск от сети. Некоторые блоки коммутации имеют только одну линию (только от сети, только от УПП или только от ПЧ). |
| pump_run | ARRAY [1..2, 1..5] OF BOOL | | Насосный агрегат в работе. Массив значений, который содержит статус работы каждого агрегата от сети или от ПЧ (TRUE – агрегат в работе, FALSE - нет). Первый индекс задает линию включения (линия 1 или 2), второй – номер насосного агрегата (1...5). Обычно линия 1 – это ПЧ/УПП, а линия 2 – от сети. Некоторые блоки коммутации имеют только одну линию (только от сети, только от УПП или только от ПЧ). |
| pump_rotation_off_enable | ARRAY [1..5] OF BOOL | | Разрешено отключить при ротации. Массив значений, который содержит статус разрешения отключения насосного агрегата при ротации (TRUE – агрегат в работе, можно отключить, FALSE - нет). Индекс задает номер насосного агрегата (1...5). |
| pump_rotation_on_enable | ARRAY [1..2, 1..5] OF BOOL | | Разрешено включить при ротации на замену. Массив значений, который содержит статус разрешения включения от сети или от ПЧ/УПП для каждого агрегата (TRUE – агрегат можно включить, FALSE - нет). Первый индекс определяет линию включения (линия 1 или 2), второй – номер насосного агрегата (1...5). Обычно линия 1 – это ПЧ/УПП, а линия 2 – от сети. Некоторые блоки коммутации имеют только одну линию (только от сети, только от УПП или только от ПЧ). |
| pump_rot_off_selected | ARRAY [1..5] OF BOOL | | Выбранный для отключения при ротации агрегат. Индекс задает номер насосного агрегата (1...5). Значение элемента массива с заданным индексом равно TRUE, если агрегат с данным номером выбран для отключения. Выбор агрегата осуществляется параметром «pump_rotation_off_select». |
| pump_rot_on_selected | ARRAY [1..2, 1..5] OF BOOL | | Выбранный для включения при ротации агрегат. Массив значений, в одну из ячеек которого необходимо записать разрешение на включение от сети или от ПЧ/УПП для выбранного агрегата (TRUE - включить, FALSE - нет). Первый индекс определяет линию включения (линия 1 или 2), второй – номер насосного агрегата (1...5). Обычно линия 1 – это ПЧ/УПП, а линия 2 – от сети. Список разрешенных для включения агрегатов определяется параметром «pump_rotation_on_enable». |
| pump_rotation_off_select | INT | | Выбор агрегата для отключения при ротации. |

| Название | Тип | Нач. знач-е | Описание |
|-------------------------|---------------------|-------------|---|
| | | | В параметр нужно записать номер насосного агрегата (1...5), который будет отключен при подаче команды ротации. |
| pump_rotation_on_select | udt_pump_sau_number | | Выбор агрегата для включения при ротации (см. п. 4.39.5.1.). В параметр нужно записать номер линии (1 или 2) и номер насоса (1...5), который нужно включить. Обычно линия 1 – это пуск от ПЧ/УПП, а линия 2 – пуск от сети. |
| next_to_stop | INT | | Следующий для останова агрегат (при разгрузке). Номер агрегата, который будет остановлен следующим при разгрузке (т.е. при отключении лишнего агрегата в процессе регулирования) или останове насосной станции. |
| next_to_start | INT | | Следующий для старта агрегат (при загрузке). Номер агрегата, который будет включен следующим при загрузке (т.е. при подключении дополнительного агрегата в процессе регулирования) или запуске насосной станции. |
| next_to_stop_auto_rot | ARRAY [0..3] OF INT | | Следующий для останова агрегат при автоматической ротации по наработке. Номер агрегата, который будет остановлен при автоматической ротации, т.е. при смене работающего насосного агрегата. |
| next_to_start_auto_rot | ARRAY [0..3] OF INT | | Следующий для старта агрегат при автоматической ротации по наработке. Номер агрегата, который будет запущен при автоматической ротации, т.е. при смене работающего насосного агрегата. |
| ready_to_start | BOOL | | Готовность станции к пуску. TRUE, если станция готова к пуску. |
| ready_to_rotation | BOOL | | Готовность к ротации выбранных насосов. TRUE, если есть готовность. |
| time_rotation | REAL | 168 | Время наработки до автоматической ротации, ч. Общее время работы насосного агрегата (по данным «pump_time_operation»), при достижении которого произойдет автоматическое переключение на другой насосный агрегат. Чтобы активировать данный режим, нужно установить флаг «auto_rotation_enable» в «std.in_config». |
| delta_sp_pv_switch_on | REAL | 0.3 | Недостаток регулирующего параметра, при котором подключается агрегат. Если значение регулируемого параметра (pv) меньше задания (sp) на эту величину, то включается таймер отсчета времени до подключения дополнительного насосного агрегата («tim_switch_on»). По истечении времени таймера подключается дополнительный агрегат. Для работы этой функции нужно разрешить загрузку/разгрузку станции («load_unload_enable» в «std.in_config»). |
| delta_sp_pv_switch_off | REAL | 0.5 | Избыток регулирующего параметра, при котором отключается агрегат. Если значение регулируемого параметра (pv) больше задания (sp) на эту величину, то включается таймер отсчета времени до отключения лишнего насосного агрегата («tim_switch_off»). По истечении времени таймера отключается лишний агрегат. Для работы этой функции нужно разрешить загрузку/разгрузку станции («load_unload_enable» в «std.in_config»). |
| time_switch_on | REAL | 60 | Время выдержки для подключения агрегата, сек. См. параметр «delta_sp_pv_switch_on». |
| time_switch_off | REAL | 60 | Время выдержки для отключения агрегата, сек. |

| Название | Тип | Нач. знач-е | Описание |
|-------------------|-----------------------------|-------------|--|
| | | | См. параметр «delta_sp_pv_switch_off». |
| f_low | REAL | 5 | Минимальная частота регулятора, Гц. Определяет минимальное значение задания частоты, подаваемого на насосные агрегаты. |
| f_hi | REAL | 50 | Максимальная частота регулятора, Гц. Определяет максимальное значение задания частоты, подаваемого на насосные агрегаты. |
| f_ramp_on | REAL | 5 | Скорость разгона при подключении агрегата с ПЧ, Гц/с. Определяет быстроту повышения частоты дополнительно подключенного агрегата с ПЧ при загрузке станции. Частота повышается до тех пор, пока не выровняется с частотой агрегата, находящегося в режиме регулирования. |
| f_ramp_off | REAL | 5 | Скорость останова при отключении агрегата с ПЧ, Гц/с. Определяет быстроту снижения частоты лишнего отключаемого агрегата с ПЧ при разгрузке станции. |
| f_switch_on | REAL | 5 | Частота ПЧ при подключении агрегата, Гц. Параметр определяет, до какого значения будет снижена частота агрегата с ПЧ при подключении дополнительного агрегата. Необходимо для компенсации избыточной производительности станции, когда подключается дополнительный насос от сети, при этом насос с ПЧ работает на максимальной частоте. Чтобы активировать режим, нужно включить флаг «change_freq_enable» в «std.in_config». |
| f_switch_off | REAL | 50 | Частота ПЧ при отключении агрегата, Гц. Параметр определяет, до какого значения будет увеличена частота агрегата с ПЧ при отключении лишнего агрегата. Необходимо для компенсации сильного снижения производительности станции, когда отключается лишний насос от сети, при этом насос с ПЧ работает на минимальной частоте. Чтобы активировать режим, нужно включить флаг «change_freq_enable» в «std.in_config». |
| f_setup_max_time | REAL | 20 | Максимальное время установки задания частоты, сек. Не используется. |
| f_setup_deviation | REAL | 2 | Допустимое отклонение при установке заданной частоты, Гц. Не используется. |
| reserve1 | REAL | 0 | резерв |
| reserve2 | REAL | 0 | резерв |
| pump | ARRAY [1..5] OF udt_pump | | Насосные агрегаты. В этот параметр в программе передаются интерфейсы блоков насосных агрегатов перед вызовом блока насосной станции (см. пример). |

4.39.5.1 Структура udt_pump_sau_number

Тип данных для задания номера линии и номера насоса для блока fb_pump_sau.

| Название | Тип | Нач. значение | Назначение | Комментарий |
|-------------|-----|---------------|------------|-----------------------|
| line_number | INT | | ввод | номер линии (1 или 2) |
| pump_number | INT | | ввод | номер насоса (1...5) |

4.39.6 Переменные интерфейса блока. Списки значений

4.39.6.1 Команды («std.in_cmd», «std.in_sau_cmd», «std.in_sau_cmd_hi»)

Тип глобальных констант - «sys_db_pump_sau_cmd» п. **Ошибка! Источник ссылки не найден..** В возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|-------------|------|----------|---|
| stop | WORD | 16#1 | Стоп. Останавливает станцию. Останов выполняется поочередно – сначала последний включенный агрегат, затем предпоследний и так до тех пор, пока все агрегаты не будут остановлены. |
| alarm_stop | WORD | 16#2 | Аварийный стоп. Останавливает станцию. На все насосные агрегаты одновременно подается команда аварийного останова. Это приводит к немедленному закрытию затворов насосных агрегатов и останову насосов. |
| start | WORD | 16#4 | Пуск. Запускает станцию. |
| rotation | WORD | 16#8 | Ротация насосных агрегатов. Команда выполняет замену одного работающего агрегата другим. Выбор останавливаемого и запускаемого агрегатов задается параметрами «pump_rotation_off_select» и «pump_rotation_on_select» (см. интерфейс блока). |
| man_up | WORD | 16#10 | Увеличить частоту. Увеличивает выходное значение частоты регулятора насосной станции на значение «pid.in_step» (см. интерфейс блока). У регулятора для этого должен быть включен режим ручного задания частоты (pid.in_man_lm_n_on = TRUE или pid.in_sau_lm_n_on = TRUE). |
| man_down | WORD | 16#20 | Уменьшить частоту. Уменьшает выходное значение частоты регулятора насосной станции на значение «pid.in_step» (см. интерфейс блока). У регулятора для этого должен быть включен режим ручного задания частоты (pid.in_man_lm_n_on = TRUE или pid.in_sau_lm_n_on = TRUE). |
| reset_alarm | WORD | 16#8000 | Сброс аварии. Сбрасывает аварийные сообщения и аварийное состояние насосной станции и насосных агрегатов. |

4.39.6.2 Аварийные сообщения «std.out_alarm».

Тип глобальных констант - «sys_db_pump_sau_alarm» п. **Ошибка! Источник ссылки не найден..** В возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|----------------|------|----------|---|
| starting_fault | WORD | 16#1 | Ни один насос не включился. Не удалось включить хотя бы один насос в процессе пуска. |
| alarm_stop | WORD | 16#2 | Нажата кнопка аварийного останова. На вход «alarm_stop_btn» пришло значение TRUE. |

4.39.6.3 Предупредительные сообщения «std.out_warning».

Тип глобальных констант - «sys_db_pump_sau_warning» п. **Ошибка! Источник ссылки не найден..** В возможные значения представлены в таблице.

| Название | Тип | Значение | Текст сообщения/Описание |
|------------|------|----------|--|
| ai_fault | WORD | 16#1 | Нет опорного датчика. Зафиксирована неисправность выбранного аналогового входа («ai_1» или «ai_2»). |
| alarm_stop | WORD | 16#2 | Нажата кнопка аварийного останова. На вход «alarm_stop_btn» пришло значение TRUE. |

4.39.6.4 Конфигурация «std.in_cfg».

Тип глобальных констант - «sys_db_pump_sau_cfg» п. **Ошибка! Источник ссылки не найден.** В возможные значения представлены в таблице.

| Название | Тип | Значение | Описание |
|-------------------------|------|----------|---|
| auto_rotation_enable | WORD | 16#1 | Разрешить автоматическую ротацию. Бит конфигурации разрешает автоматический переход с одного насоса на другой при достижении времени наработки насоса значения «time_rotation» (см. интерфейс блока). |
| time_operation_enable | WORD | 16#2 | Подключение и отключение насосных агрегатов по наработке. Бит конфигурации разрешает выбор насосного агрегата по наработке при пуске насосной станции и при подключении и отключении агрегатов в процессе регулирования производительности. |
| change_freq_enable | WORD | 16#4 | Разрешить изменение частоты ПЧ при подключении/отключении агрегата. Бит конфигурации разрешает изменение частоты регулирующего насоса с ПЧ при подключении/отключении дополнительных агрегатов (обычно от сети) для компенсации избыточного повышения/снижения производительности станции. Значения частот задаются параметрами «f_switch_on», «f_switch_off». |
| pid_inverse | WORD | 16#8 | Инвертировать регулятор. Меняет местами задание и регулируемый параметр при подаче на вход внутреннего ПИД-регулятора блока. Если бит конфигурации включен, то при увеличении значения переменной процесса производительность станции увеличивается, и наоборот. |
| step_by_step_fr_connect | WORD | 16#10 | Каскадное подключение регулируемых агрегатов. В том случае, когда в составе насосной станции есть несколько агрегатов с ПЧ, этот бит конфигурации позволяет разгонять/останавливать и подключать/отключать эти насосные агрегаты по очереди. Когда очередной агрегат достиг максимальной частоты, подключается следующий и регулирует производительность. При достижении им максимальной частоты подключается следующий и т.д. То же самое при разгрузке станции – сначала снижает частоту до минимума последний включенный агрегат, отключается, затем снижает частоту следующий и т.д. |
| select_ai_reserve | WORD | 16#20 | Выбрать резервный опорный датчик. Если бит включен, то ПИД-регулятор станции работает со входом «ai_2» в качестве переменной процесса (pv). |
| load_unload_enable | WORD | 16#40 | Разрешить загрузку/разгрузку станции по регулируемому параметру. Бит конфигурации разрешает подключение дополнительных агрегатов и отключение лишних агрегатов в процесс регулирования производительности станции. |
| stop_pump_by_rotation | WORD | 16#80 | При ротации сначала отключать основной насос. Если бит конфигурации включен, то при ротации сначала отключается работающий насос, и только затем включается резервный. Если бит выключен, то сначала включается резервный насос, и только затем отключается работающий. |

4.39.6.5 Состояние объекта «std.out_state».

Тип глобальных констант - «sys_db_pump_sau_hmi_state» п. **Ошибка! Источник ссылки не найден..** В возможные значения представлены в таблице.

| Название | Тип | Начальное значение | Описание |
|------------------|------|--------------------|--|
| not_ready | WORD | 16#1 | Нет готовности. Нет готовности станции к пуску. |
| ready | WORD | 16#2 | Готовность. Станция готова к пуску. |
| starting | WORD | 16#4 | Запуск. Идет процесс запуска насосной станции. |
| started | WORD | 16#8 | Работа. Насосная станция в работе. |
| stopping | WORD | 16#10 | Останов. Идет процесс останова насосной станции. |
| alarm | WORD | 16#20 | Авария. Насосная станция в состоянии аварии. |
| start_pump | WORD | 16#40 | Запуск насосного агрегата. Идет процесс пуска. Запускается очередной насосный агрегат. |
| stop_pump | WORD | 16#80 | Останов насосного агрегата. Идет процесс останова. Останавливается очередной насосный агрегат. |
| reserve_up | WORD | 16#100 | Включение технологического резерва. Идет процесс подключения дополнительного насосного агрегата в процессе регулирования производительности насосной станции. |
| reserve_down | WORD | 16#200 | Отключение технологического резерва. Идет процесс отключения лишнего насосного агрегата в процессе регулирования производительности насосной станции. |
| alarm_reserve_up | WORD | 16#400 | Включение аварийного резерва. Идет процесс подключения резервного насосного агрегата вместо аварийно отключившегося. |
| rotation | WORD | 16#800 | Ротация. Идет процесс перехода с одного насоса на другой. Была подана команда ротации на блок, либо переход инициирован автоматически (автоматическая ротация). |

4.39.6.6 Состояние объекта «std.out_state_actual».

Тип глобальных констант - «sys_db_pump_sau_hmi_state_actual» п. **Ошибка! Источник ссылки не найден..** Возможные значения представлены в таблице ниже.

Фактическое состояние насосного агрегата полностью копируется из состояния подчиненного блока коммутации электродвигателя (hmi.std.out_state_actual = engine.std.out_state_actual).

| Название | Тип | Начальное значение | Описание |
|-----------|------|--------------------|--------------------------------------|
| power_off | WORD | 16#1 | Нет готовности станции к пуску. |
| stopped | WORD | 16#2 | Станция остановлена, готова к пуску. |
| started | WORD | 16#4 | Станция в работе. |

4.39.7 Пример

Объявление глобальных переменных:

```
VAR_GLOBAL
// объявление интерфейса блоков насосный агрегат
dbs_pump_1:          RuDrive.udt_pump;
```

```

    dbs_pump_2:      RuDrive.udt_pump;
    dbs_pump_3:      RuDrive.udt_pump;
    // объявление интерфейса блока насосная станция
    dbs_pump_station: RuDrive.udt_pump_sau;
END_VAR

```

Вызов в циклической подпрограмме (цикл 100 мсек):

```

PROGRAM PLC_PRG
VAR
    // объявление экземпляров блоков насосный агрегат
    dbi_pump_1:      RuDrive.fb_pump;
    dbi_pump_1:      RuDrive.fb_pump;
    dbi_pump_1:      RuDrive.fb_pump;
    // объявление экземпляра блока насосная станция
    dbi_pump_station: RuDrive.fb_pump_sau;
END_VAR

// вызовы экземпляров блоков, входящих в состав насосных агрегатов, для простоты не
// показаны

// вызов экземпляра блока насосного агрегата
dbi_pump_1(
    protect := pump_1_protect,
    dry := pump_1_dry,
    pressure_min := pump_1_pressure_min,
    pressure_max := pump_1_pressure_max,
    dry_ai := pump_1_dry_ai,
    p_ai := pump_1_p_ai,
    hmi := dbs_pump_1,
    valve_in := dbs_valve_1_in,
    valve_out := dbs_valve_1_out,
    engine := dbs_engine_1 );

// вызов экземпляра блока насосного агрегата
dbi_pump_2(
    protect := pump_2_protect,
    dry := pump_2_dry,
    pressure_min := pump_2_pressure_min,
    pressure_max := pump_2_pressure_max,
    dry_ai := pump_2_dry_ai,
    p_ai := pump_2_p_ai,
    hmi := dbs_pump_2,
    valve_in := dbs_valve_2_in,
    valve_out := dbs_valve_2_out,
    engine := dbs_engine_2 );

// вызов экземпляра блока насосного агрегата
dbi_pump_3(
    protect := pump_3_protect,
    dry := pump_3_dry,
    pressure_min := pump_3_pressure_min,
    pressure_max := pump_3_pressure_max,
    dry_ai := pump_3_dry_ai,
    p_ai := pump_3_p_ai,
    hmi := dbs_pump_3,
    valve_in := dbs_valve_3_in,
    valve_out := dbs_valve_3_out,
    engine := dbs_engine_3 );

// вызов экземпляра блока насосной станции

dbs_pump_station.pump[1] := dbs_pump_1;
dbs_pump_station.pump[2] := dbs_pump_2;
dbs_pump_station.pump[3] := dbs_pump_3;

dbi_pump_station(
    ai_1 := pv_ai,
    alarm_stop_btn := pump_station_alarm_stop,
    hmi := dbs_pump_station );

```


5 Описание функций

5.1 Интерпретатор команд и режимов (fc_cmd)

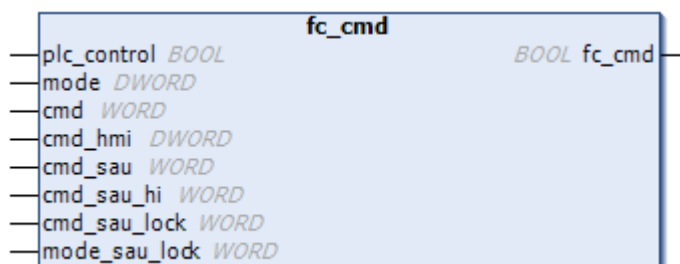
5.1.1 Назначение

Вспомогательная функция для обработки команд и режимов, подаваемых на функциональный блок. Функция обычно используется внутри других функциональных блоков, в качестве входных параметров принимая часть параметров структуры «udt_standard». Может быть использована при написании собственного функционального блока.

5.1.2 Функции

- изменение режима блока.
- фильтрация команд в зависимости от режима.

5.1.3 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|-------------|-------|--------------------|---|
| Return | fc_cmd | BOOL | | Возвращаемое значение функции. Не используется. Равно 0. |
| Input | plc_control | BOOL | | Вход выбора режима управления: местный (plc_control = «FALSE») или дистанционный (plc_control = «TRUE»). |
| Inout | mode | DWORD | | Текущий режим блока. Вычисляется на основе своего предыдущего значения, значения входа «plc_control» и поступающих команд. |
| | cmd | WORD | | Текущая команда. Команда после фильтрации для дальнейшего использования в алгоритмах. |
| | cmd_hmi | DWORD | | Команда управления от HMI (оператора). Будет выполнена (т.е. передана на выход «cmd») только в режиме «диспетчер». Исключение – команды сброса аварии и сброса наработки. После выполнения функции значение команды сбрасывается в 0. |
| | cmd_sau | WORD | | Команда управления от САУ. Будет выполнена (т.е. передана на выход «cmd») только в режиме «автомат». Исключение – команды сброса аварии и сброса наработки. После выполнения функции значение переменной сбрасывается в 0. |
| | cmd_sau_hi | WORD | | Команда управления от САУ (с наивысшим приоритетом). Данная команда будет выполнена в режиме «автомат» и «диспетчер». |

| Область | Название | Тип | Начальное значение | Назначение |
|---------|---------------|------|--------------------|---|
| | | | | Блокировки команд на высокоприоритетные команды не действуют. После выполнения функции значение переменной сбрасывается в 0. |
| | cmd_sau_lock | WORD | | Блокировки команд управления от САУ. Установленные в слове биты, соответствующие номерам команд из «sys_db_cmd», блокируют прохождение этих команд на выход «cmd». После выполнения функции значение переменной сбрасывается в 0, поэтому для того, чтобы блокировка была активной, значение переменной нужно удерживать (т.е. присваивать в каждом цикле перед вызовом функции). |
| | mode_sau_lock | WORD | | Блокировки режимов от САУ. Установленные в слове биты, соответствующие номерам режимов из «sys_db_mode», блокируют установку этих режимов в «mode». После выполнения функции значение переменной сбрасывается в 0, поэтому для того, чтобы блокировка была активной, значение переменной нужно удерживать (т.е. присваивать в каждом цикле перед вызовом функции). |

5.1.4 Принцип работы

Функция выполняет две основные операции:

1. Изменение режима «mode» в зависимости от состояния входа «plc_control» и команды на изменение режима, подаваемой на вход «cmd_hmi».

Если новый режим входит в список заблокированных (соответствующий номеру режима бит установлен в «mode_sau_lock»), то команда на изменение режима не выполнится. Список возможных режимов хранится в структуре «sys_db_mode» (см. п. 3.3.3.11).

2. Фильтрация команды, подаваемой на входы «cmd_hmi», «cmd_sau» или «cmd_sau_hi» и ее передача на выход «cmd» для дальнейшего использования в алгоритме блока. Если команда входит в список заблокированных (соответствующий номеру команды бит установлен в «cmd_sau_lock»), то команда не пройдет через фильтр на выход «cmd» (на выходе будет 0).

Таблица изменения режимов. Результат работы функции в зависимости от входных данных

| Вход режима «plc_control» | Команда «cmd_hmi» | Начальное значение текущего режима «mode» (до вызова функции) | Расчетное значение текущего режима «mode» (после вызова функции) |
|---------------------------|---|---|--|
| FALSE | 16#0000XXXX (нет команды на смену режима) | 16#0002 (ремонт) | 16#0002 (ремонт) |
| | | 16#0004 (местный) | 16#0004 (местный) |
| | | 16#0008 (автомат) | 16#0004 (местный) |
| | | 16#00100000 («диспетчер 1»).. 16#80000000 («диспетчер 12») | 16#0004 (местный) |
| | 16#00020000 (в ремонт) | 16#0004 (местный) | 16#0002 (ремонт) |
| | 16#00080000 (в автомат) | 16#0002 (ремонт) | 16#0004 (местный) |
| TRUE | 16#0000XXXX (нет команды на смену режима) | 16#0002 (ремонт) | 16#0002 (ремонт) |
| | | 16#0004 (местный) | 16#0008 (автомат) |
| | | 16#0008 (автомат) | 16#0008 (автомат) |
| | | 16#00100000 («диспетчер 1»).. 16#80000000 («диспетчер 12») | 16#00100000 («диспетчер 1»).. 16#80000000 («диспетчер 12») |

Таблица преобразования команд «cmd_hmi». Результат работы функции в зависимости от входных данных

| Команда «cmd_hmi» (YYYY – любой режим, XXXX – любая команда) | Текущий режим «mode» | Расчетное значение команды «cmd» (после вызова функции) |
|--|---|--|
| 16#YYYYXXXX | 16#0002 (ремонт) | 16#0000 |
| 16#YYYYXXXX | 16#0004 (местный) | 16#0000 |
| 16#YYYYXXXX | 16#0008 (автомат) | 16#0000 |
| 16#0010XXXX... 16#8000XXXX | 16#0010 («диспетчер 1»).. 16#8000 («диспетчер 12») | 16#XXXX. Если режим «mode» совпал с режимом команды YYYY, то команда проходит, иначе «cmd» = 16#0000. |

Исключением являются общие команды блоков: команда сброса аварии «reset_alarm» и команда сброса анарботки «reset_time_operation». Они проходят на выход «cmd» во всех режимах.

Таблица преобразования команд «cmd_sau». Результат работы функции в зависимости от входных данных

| Команда «cmd_sau» (XXXX – любая команда) | Текущий режим «mode» | Расчетное значение команды «cmd» (после вызова функции) |
|--|---|---|
| 16#XXXX | 16#0002 (ремонт) | 16#0000 |
| 16#XXXX | 16#0004 (местный) | 16#0000 |
| 16#XXXX | 16#0008 (автомат) | 16#XXXX |
| 16#XXXX | 16#0010 («диспетчер 1»).. 16#8000 («диспетчер 12») | 16#0000 |

Исключением являются общие команды блоков: команда сброса аварии «reset_alarm» и команда сброса анарботки «reset_time_operation». Они проходят на выход «cmd» во всех режимах.

Таблица преобразования команд «cmd_sau_hi». Результат работы функции в зависимости от входных данных

| Команда «cmd_sau_hi» (XXXX – любая команда) | Текущий режим «mode» | Расчетное значение команды «cmd» (после вызова функции) |
|---|---|---|
| 16#XXXX | 16#0002 (ремонт) | 16#XXXX |
| 16#XXXX | 16#0004 (местный) | 16#XXXX |
| 16#XXXX | 16#0008 (автомат) | 16#XXXX |
| 16#XXXX | 16#0010 («диспетчер 1»).. 16#8000 («диспетчер 12») | 16#XXXX |

Исключением является команда сброса аварии «reset_alarm». Она проходит на выход «cmd» во всех режимах.

5.2 Установка режима (fc_set_mode)

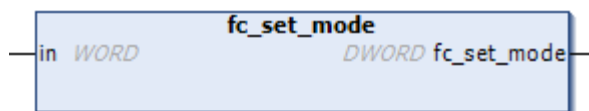
5.2.1 Назначение

Функция предназначена для преобразования переменной типа WORD, содержащей режим объекта согласно списку констант в структуре sys_db_mode, к типу DWORD, в котором хранится смещенное на 16 бит влево значение режима. Это значение можно непосредственно подавать на вход «std.in_cmd» любого из функциональных блоков для задания его режима.

Также функцию можно использовать для формирования команды на функциональный блок в режимк «диспетчер», например, так:

```
fc_set_mode(sys_db_mode.station_1) OR sys_db_cmd.contactor.start
```

5.2.2 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|-------------|-------|--------------------|------------|
| Return | fc_set_mode | DWORD | Return | |
| Input | in | WORD | Input | |

5.3 Получение даты-времени (fc_datetime)

5.3.1 Назначение

Функция предназначена для чтения текущих даты и времени контроллера (часов реального времени).

5.3.2 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|-------------|------|--------------------|--------------------------------|
| Return | fc_datetime | BOOL | | Не используется. Возвращает 0. |
| Inout | year | INT | | год |
| | month | INT | | месяц |
| | day | INT | | день |
| | hour | INT | | час |
| | minute | INT | | минута |
| | second | INT | | секунда |
| | ms | INT | | миллисекунда |

5.4 Системный таймер (fc_time)

5.4.1 Назначение

Функция предназначена для чтения системного таймера контроллера (время с последнего включения в секундах).

5.4.2 Таблица входов-выходов



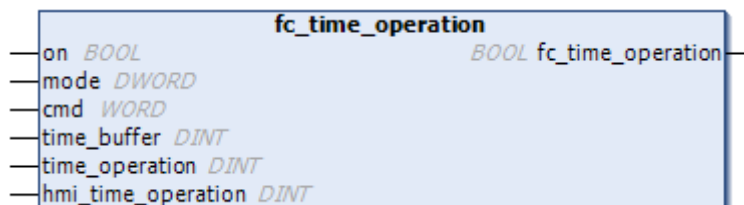
| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------|------|--------------------|--|
| Return | fc_time | DINT | | Время с последнего включения в секундах. |

5.5 Подсчет наработки (fc_time_operation)

5.5.1 Назначение

Функция предназначена для подсчета времени наработки агрегата (исполнительного механизма) в часах.

5.5.2 Таблица входов-выходов



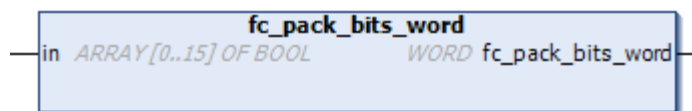
| Область | Название | Тип | Начальное значение | Назначение |
|---------|--------------------|-------|--------------------|---|
| Return | fc_time_operation | BOOL | | Не используется. Возвращает 0. |
| Input | on | BOOL | | Если «on=TRUE» и режим агрегата «mode» не ремонт, включается подсчет наработки. |
| | mode | DWORD | | Текущий режим агрегата. Если значение режима не равно «sys_db_mode.repair» (т.е. не ремонт), и вход «on=TRUE», включается подсчет наработки. |
| | cmd | WORD | | Текущая команда. Вход предназначен для сброса наработки («time_operation» и «hmi_time_operation») командой «sys_db_cmd.reset_time_operation». |
| Inout | time_buffer | DINT | | Буферная переменная. |
| | time_operation | DINT | | Наработка в секундах. Если значение достигло 3600 (1 час), обнуляется и начинает считать заново. К значению «hmi_time_operation» прибавляется один час. |
| | hmi_time_operation | DINT | | Суммарная наработка в часах. |

5.6 Упаковка битов в слово (fc_pack_bits_word)

5.6.1 Назначение

Функция предназначена для преобразования массива из 16 переменных типа BOOL в слово типа WORD.

5.6.2 Таблица входов-выходов



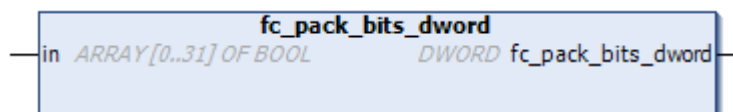
| Область | Название | Тип | Начальное значение | Назначение |
|---------|-------------------|-----------------------|--------------------|---|
| Return | fc_pack_bits_word | WORD | | Возвращаемое значение функции - выходное слово. Младший элемент входного массива «in» соответствует младшему биту в преобразованном слове, старший элемент – старшему биту. |
| Input | in | ARRAY [0..15] OF BOOL | | Входные биты для преобразования. |

5.7 Упаковка битов в двойное слово (fc_pack_bits_dword)

5.7.1 Назначение

Функция предназначена для преобразования массива из 32 переменных типа BOOL в двойное слово типа DWORD.

5.7.2 Таблица входов-выходов



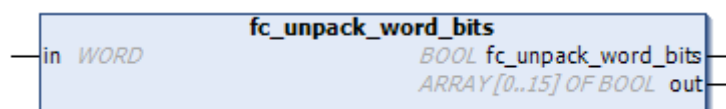
| Область | Название | Тип | Начальное значение | Назначение |
|---------|--------------------|-----------------------|--------------------|---|
| Return | fc_pack_bits_dword | DWORD | | Возвращаемое значение функции - выходное двойное слово. Младший элемент входного массива «in» соответствует младшему биту в преобразованном слове, старший элемент – старшему биту. |
| Input | in | ARRAY [0..31] OF BOOL | | Входные биты для преобразования. |

5.8 Распаковка битов из слова (fc_unpack_word_bits)

5.8.1 Назначение

Функция предназначена для преобразования слова типа WORD в массив из 16 переменных типа BOOL.

5.8.2 Таблица входов-выходов



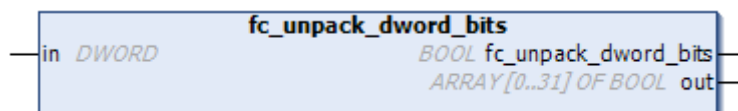
| Область | Название | Тип | Начальное значение | Назначение |
|---------|---------------------|-----------------------|--------------------|--|
| Return | fc_unpack_word_bits | BOOL | | Возвращаемое значение. Не используется, равно 0. |
| Input | in | WORD | | Входное слово для преобразования. |
| Output | out | ARRAY [0..15] OF BOOL | | Выходные биты после преобразования входного слова. Младший элемент массива соответствует младшему биту в слове, старший элемент – старшему биту. |

5.9 Распаковка битов из двойного слова (fc_unpack_dword_bits)

5.9.1 Назначение

Функция предназначена для преобразования двойного слова типа DWORD в массив из 32 переменных типа BOOL.

5.9.2 Таблица входов-выходов



| Область | Название | Тип | Начальное значение | Назначение |
|---------|----------------------|-----------------------|--------------------|--|
| Return | fc_unpack_dword_bits | BOOL | | Возвращаемое значение. Не используется, равно 0. |
| Input | in | DWORD | | Входное двойное слово для преобразования. |
| Output | out | ARRAY [0..31] OF BOOL | | Выходные биты после преобразования входного слова. Младший элемент массива соответствует младшему биту в слове, старший элемент – старшему биту. |

6 Описание вспомогательных типов данных

В разделе приведено описание типов данных, которые используются в функциях и функциональных блоках, например, как составная часть интерфейса блока.

6.1 Параметры ПИД-регулятора (udt_pid)

Тип данных описывает входные и выходные параметры ПИД-регулятора, встроенного в функциональный блок. С помощью этих параметров можно управлять регулятором, а также изменять его коэффициенты регулирования.

| Название | Тип | Нач. значение | Назначение | Комментарий |
|---------------|------|---------------|------------|--|
| in_man_sp | REAL | | ввод | Уставка регулятора в дистанции. Параметр предназначен для ввода значения задания ПИД-регулятора, когда блок, в который встроен регулятор, находится в режиме «диспетчер», и ПИД-регулятор включен (у блока отключен режим «ручной ввод», in_man_lmn_on = FALSE). |
| in_man_lmn | REAL | | ввод | Задание управляющего воздействия в дистанции. Параметр предназначен для ввода выходного значения ПИД-регулятора, когда блок, в который встроен регулятор, находится в режиме «диспетчер», и ПИД-регулятор выключен (у блока включен режим «ручной ввод», in_man_lmn_on = TRUE). |
| in_sau_sp | REAL | | ввод | Уставка регулятора в авто режиме от САУ. Параметр предназначен для передачи значения задания ПИД-регулятору от других программных блоков, когда блок, в который встроен регулятор, находится в режиме «автомат», и ПИД-регулятор включен (у блока отключен режим «ручной ввод», in_sau_lmn_on = FALSE). |
| in_sau_lmn | REAL | | ввод | Задание управляющего воздействия от САУ. Параметр предназначен для передачи выходного значения ПИД-регулятору от других программных блоков, когда блок, в который встроен регулятор, находится в режиме «автомат», и ПИД-регулятор выключен (у блока включен режим «ручной ввод», in_sau_lmn_on = TRUE). |
| in_man_lmn_on | BOOL | | ввод | Включить ручное управление выходом регулятора в дистанции. Параметр предназначен для включения ручного задания выхода регулятора, когда блок, в который встроен регулятор, находится в режиме «диспетчер». Если установить параметр в значение TRUE, то значение, записанное в переменную in_man_lmn, будет передаваться на выход регулятора. При установке параметра в FALSE регулятор включится и начнет регулирование. |
| in_sau_lmn_on | BOOL | | ввод | Включить ручное управление выходом регулятора от САУ. Параметр предназначен для включения ручного задания выхода регулятора, когда блок, в который встроен регулятор, находится в режиме «автомат». Если установить параметр в значение TRUE, то значение, записанное в переменную in_sau_lmn, будет передаваться на выход регулятора. При установке параметра в FALSE регулятор включится и начнет регулирование. |
| in_reserve | INT | | | заглушка для выравнивания по 4 байта |
| in_step | REAL | 0.1 | ввод | Шаг изменения выхода регулятора при ручном задании выхода командами «увеличить» и «уменьшить». Параметр определяет, на какую величину будет изменяться выход регулятора при однократной подаче команды «увеличить» или «уменьшить», когда регулятор находится в |

| Название | Тип | Нач. значение | Назначение | Комментарий |
|----------|-------------------------|---------------|------------|--|
| | | | | режиме ручного задания выхода (т.е. in_man_lmн_on = TRUE или in_sau_lmн_on = TRUE). |
| out_sp | REAL | | вывод | Текущая уставка. Фактическое значение задания регулируемой переменной, которое ПИД-регулятор должен достичь. |
| out_pv | REAL | | вывод | Переменная процесса. Фактическое значение регулируемой переменной. |
| out_lmн | REAL | | вывод | Выход регулятора. Выходное значение регулятора, которое поступает на выход блока и далее на аналоговый выход управления регулирующим клапаном, ПЧ или другим устройством. |
| settings | <u>udt_pid_settings</u> | | ввод-вывод | Коэффициенты ПИД-регулятора. См. п. 6.2. |

6.2 Коэффициенты ПИД-регулятора (udt_pid_settings)

Тип данных описывает коэффициенты ПИД-регулятора.

| Название | Тип | Нач. значение | Назначение | Комментарий |
|-------------|------|---------------|------------|--|
| in_gain | REAL | 1 | ввод | Пропорциональный коэффициент |
| in_ti | REAL | 20 | ввод | Время интегрирования, сек |
| in_td | REAL | 0 | ввод | Время дифференцирования, сек |
| in_tm_lag | REAL | 0 | ввод | Время затухания дифференциальной составляющей, сек |
| in_deadband | REAL | 0 | ввод | Зона нечувствительности |

6.3 Параметры двигателя (udt_engine_ac)

Тип данных описывает параметры двигателя, используемые в интерфейсах блока ПЧ (udt_fr) и коммутации электродвигателей (udt_engine).

| Название | Тип | Нач. значение | Назначение | Комментарий |
|------------|------|---------------|------------|--|
| i_xx | REAL | 0 | ввод | Ток холостого хода $(0,05 - 0,1) \cdot I_{ном}$. Параметр предназначен для защиты двигателя, работающего от ПЧ, если ток двигателя ниже или равен току холостого хода при максимальной частоте вращения (предположительно, обрыв муфты). |
| i_nom | REAL | 0 | ввод | Номинальный ток. Параметр предназначен для блокировки повышения частоты вращения двигателя, работающего от ПЧ, если ток двигателя достиг номинального значения. |
| i_overload | REAL | 0 | ввод | Ток перегрузки $(1,2 \cdot I_{ном})$. Параметр предназначен для снижения частоты вращения двигателя, работающего от ПЧ, если ток двигателя достиг тока перегрузки. Частота ПЧ снижается до тех пор, пока ток не достигнет номинального значения. |

6.4 Дата-время

Тип данных описывает структуру хранения даты и времени, применяемую в некоторых блоках.

| Название | Тип | Нач. значение | Назначение | Комментарий |
|----------|-----|---------------|------------|-------------|
| year | INT | | вывод | год |

| Название | Тип | Нач. значение | Назначение | Комментарий |
|----------|-----|---------------|------------|--------------|
| month | INT | | вывод | месяц |
| day | INT | | вывод | день |
| hour | INT | | вывод | час |
| minute | INT | | вывод | минута |
| second | INT | | вывод | секунда |
| ms | INT | | вывод | миллисекунда |

7 КОНТАКТНАЯ ИНФОРМАЦИЯ ПРОИЗВОДИТЕЛЯ

7.1 Юридическая информация

Информация о юридическом лице компании:

Название компании: ООО «РУ-ДРАЙВ ЦИФРОВЫЕ ТЕХНОЛОГИИ»

Юр. адрес: 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.

ОГРН: 1221600086751

ИНН: 1650418125

7.2 Контактная информация службы технической поддержки

Связаться со специалистами службы технической поддержки можно одним из следующих способов:

Сайт: <https://ru-drive.it>

Телефон: 8 800 555-70-30

Email: digital@ru-drive.com

Фактический адрес размещения инфраструктуры разработки: 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.

Фактический адрес размещения разработчиков: 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.

Фактический адрес размещения службы поддержки: 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.