EcoStruxure Machine Expert - HVAC

Руководство по эксплуатации

(локализованная версия документа)

Октябрь 2018 г.

EIO0000003412.00

www.schneider-electric.com

В данном документе приводится общее описание и (или) технические характеристики упомянутых в нем изделий. Данный документ не заменяет собой прочую техническую документацию и не может использоваться для определения эксплуатационной пригодности или надежности рассматриваемых в нем изделий для конкретных вариантов применения, определяемых пользователем. Анализ и оценка пригодности изделий для конкретного применения и всех рисков, связанных с их применением, а также проведение испытаний являются обязанностью пользователя или системного интегратора. Компания Schneider Electric и ее аффилированные или дочерние компании не несут ответственности за неправильное использование приведенной в этом документе информации. Обращайтесь к нам, если у вас есть какие-либо предложения по улучшению, внесению правок или исправлению ошибок в данной публикации.

Вы соглашаетесь не копировать настоящий документ целиком или частично на любой носитель без письменного разрешения компании Schneider Electric, кроме как для личного некоммерческого использования. Также вы соглашаетесь не приводить какие-либо гипертекстовые ссылки на этот документ или его содержание.
Компания Schneider Electric не предоставляет прав или лицензий на личное и некоммерческое использование настоящего документа и его содержимого, за исключением неисключительной лицензии на использование текущей редакции документа в справочных целях на свой собственный страх и риск. Все прочие права защищены.

Во время установки и использования данного продукта следует соблюдать все действующие государственные, региональные, а также местные нормы и правила безопасности. По соображениям безопасности и в целях обеспечения соответствия задокументированным в системе характеристикам ремонт компонентов должен выполняться только производителем.

Если оборудование применяется в среде с повышенными требованиями к технике безопасности, следует соблюдать соответствующие инструкции.

Отказ от использования ПО Schneider Electric или утвержденного ПО с нашим оборудованием может привести к травмам, ущербу или неудовлетворительным операционным результатам.

Несоблюдение этих указаний может привести к увечьям или повреждению оборудования.

© Schneider Electric, 2018. Все права защищены.

Содержание

Информация по технике безопасности 13
Об этом документе 15
Часть 1. Инструкция по началу работы с EcoStruxure Machine Expert – HVAC 19
Глава 1. Инструкция по началу работы с EcoStruxure Machine Expert – HVAC 21
1.1 Введение в EcoStruxure Machine Expert – HVAC 22
Общее описание 22
1.2 Системные требования и поддерживаемые устройства 24
Требования к системе 25
Регистрация ПО EcoStruxure Machine Expert – HVAC 26
Поддерживаемые устройства 27
Принадлежности для подключения и передачи данных 29
Глава 2. Интерфейс программы 31
2.1 Основы пользовательского интерфейса EcoStruxure Machine Expert – HVAC 32
Создание проектов в EcoStruxure Machine Expert – HVAC 33
Написание программ в EcoStruxure Machine Expert – HVAC 34
Навигация в EcoStruxure Machine Expert – HVAC 35
Режимы работы 39
2.2 Настройка рабочего пространства 41
Компоновка 42
Панели инструментов 43
Управление окнами инструментов 44
Управление окнами 45
Полноэкранный режим 46
Параметры программного обеспечения 47
Глава 3. Управление проектами 51
3.1 Управление проектами 52
Создание нового проекта 53
Печать проекта 56
Сохранение проекта 57
Управление существующими проектами 59

Распределение проектов 60
Экспорт в файлы CSV 61
Выбор целевого устройства 62
Сборка 64
Загрузка проекта на целевое устройство 65
ПО Installer 67
Закрытие EcoStruxure Machine Expert – HVAC 68
Часть 2 Конфигурирование 69
Глава 4. Вкладка CONFIGURATION (КОНФИГУРИРОВАНИЕ) 71
4.1 Обзор 72
Обзор окна **CONFIGURATION** (КОНФИГУРИРОВАНИЕ) 73
Панель меню 75
Панель инструментов 79
Строка состояния 80
Глава 5. Управление элементами ресурсов 81
5.1 Обзор 82
Окно Resources (Ресурсы) 83
Поддерживаемые протоколы 85
5.2 Целевое устройство 87
Целевое устройство 87
5.3 Объекты Modbus 88
Объекты Modbus 88
5.4 Меню целевого устройства 92
Меню целевого устройства M171O 93
Меню целевого устройства M171P/M172 95
5.5 Сопоставление входов/выходов 96
Сопоставление входов/выходов 96
5.6 Сигналы тревоги 97
Сигналы тревоги 97
5.7 Веб-сайт 98
Веб-сайт 98
5.8 Шина расширения CAN 101
Шина расширения CAN 102
Использование модуля расширения в качестве ведомого устройства шины расширения CAN 104

Шина расширения CAN для M171P с креплением для скрытого монтажа .. 109
Пользовательское устройство CAN 110
Использование пользовательского устройства CAN 112
Шина расширения CAN (полевая) — виртуальные каналы ведущих устройств 116
5.9 RS485 117
RS485 118
Использование TM171E27I/O в качестве ведомого устройства RS485 119
Обзор объекта **типового устройства Modbus** 120
Сообщения объекта **типового устройства Modbus** 122
Пользовательские устройства Modbus 124
Использование пользовательского устройства Modbus 126
5.10 Ethernet 128
Ethernet 128
5.11 Подключаемые модули 130
Обзор линейки коммуникационных модулей 130
Глава 6. Справочник по технической реализации 133
6.1 Протокол Modbus 134
Обзор 135
Типы данных 136
Коды функций 137
Часть 3. Программирование 139
Глава 7. Вкладка PROGRAMMING (ПРОГРАММИРОВАНИЕ)..... 141
7.1 Обзор 142
Обзор окна Programming (Программирование) 143
Панель меню 147
Панели инструментов 155
Строка состояния 162
Глава 8. Параметры проекта 163
8.1 Параметры проекта 164
Вкладка General (Общее) 165
Вкладка Code Generation (Генерирование кода) 166
Вкладка Build Output (Вывод сборки) 168
Вкладка Debug (Отладка) 170
Вкладка Build Events (События сборки) 171
Вкладка Cross Reference (Перекрестные ссылки) 172

8.2 Работа с библиотеками 174
Диспетчер библиотек 175
Экспорт в библиотеку 177
Импорт из библиотеки или другого источника 178
Обновление имеющихся библиотек 180
Глава 9. Управление элементами проекта 181
9.1 Обзор 182
Окно Project (Проект) 182
9.2 Программные модули 183
Обзор 183
Создание программы 184
Создание функционального блока/функции 185
Редактирование программных модулей 186
Шифрование/дешифрование исходного кода 186
9.3 Переменные 188
Глобальные переменные 188
Локальные переменные 195
Создание нескольких переменных 198
9.4 Задачи 199
Связывание программы с задачей 199
Настройка задач 201
9.5 Производные типы данных 202
Обзор 202
Определение типа 203
Структуры 205
Перечисления 207
Поддиапазоны 209
9.6 Навигация по проекту 211
Обзор 211
Обозреватель объектов 212
Поиск с помощью команды Find in Project (Найти в проекте) 217
9.7 Пользовательское рабочее пространство проекта 219
Обзор 220
Включение пользовательского рабочего пространства в существующий проект 221
Миграция между рабочими пространствами 221

Базовые блоки пользовательского рабочего пространства 221
Операции в пользовательском рабочем пространстве 222
Элементы рабочего пространства и их ограничения 223
Глава 10. Редактирование исходного кода 225
10.1 Обзор 226
Обзор..... 226
10.2 Редактор списков инструкций..... 227
Обзор 227
Функции редактирования..... 228
Ссылка на объекты ПЛК 228
Автоматическое обнаружение ошибок 228
Закладки 229
10.3 Редактор функциональных блок-схем 230
Обзор 230
Создание нового документа FBD 231
Добавление/удаление сетей 231
Маркировка сетей 232
Вставка и соединение блоков 233
Редактирование сетей 235
Изменение свойств блоков 235
Получение информации о блоке..... 235
Автоматическое обнаружение ошибок 235
10.4 Редактор релейных схем 236
Обзор..... 237
Создание нового документа LD 237
Добавление/удаление сетей 238
Маркировка сетей 239
Вставка контактов 240
Вставка катушек 242
Вставка блоков..... 243
Редактирование свойств катушек и контактов 244
Редактирование сетей 244
Изменение свойств блоков..... 245
Получение информации о блоке..... 246
Автоматическое обнаружение ошибок 246
Вставка переменных..... 247

Вставка констант 247
Вставка выражения 248
Комментарии 249
Ветви 250
10.5 Редактор «Структурного текста» 251
Обзор 251
Создание и редактирование объектов ST 252
Функции редактирования 252
Ссылка на объекты ПЛК 252
Автоматическое обнаружение ошибок 252
Закладки 253
10.6 Редактор последовательных функциональных схем 254
Обзор 254
Создание нового документа SFC 254
Вставка нового элемента SFC 255
Присоединение элементов SFC 255
Назначение действия шагу 256
Указание условий перехода 258
Назначение переходу условного кода 259
Указание пункта назначения для прыжка 261
Редактирование сетей SFC 261
10.7 Редактор переменных 262
Обзор 262
Открытие редактора переменных 263
Создание новой переменной 264
Редактирование переменных 264
Удаление переменных 267
Сортировка переменных 267
Копирование переменных 267
Глава 11. Компиляция 269
11.1 Обзор 270
Обзор 270
11.2 Компиляция проекта 271
Обзор 271
Загрузка файла образа 272
11.3 Вывод компилятора 273

Обзор 273
Ошибки компилятора 274
11.4 Компилятор с поддержкой командной строки 276
Обзор 276
Глава 12. Запуск приложения 277
12.1 Обзор 278
Обзор 278
12.2 Настройка связи 279
Обзор 280
Сохранение последнего использованного порта связи 286
12.3 Онлайн-статус 287
Статус соединения 287
Статус проекта 288
12.4 Загрузка приложения 289
Обзор 289
12.5 Моделирование 290
Обзор 290
12.6 Управление выполнением приложения ПЛК 291
Управление выполнением приложения ПЛК 291
Глава 13. Отладка 293
13.1 Обзор 294
Обзор 294
13.2 Окно контрольных значений 295
Обзор 295
Открытие и закрытие окна контрольных значений 296
Добавление элементов в окно контрольных значений 296
Удаление переменной 299
Обновление значений 299
Изменение формата данных 300
Работа со списками контрольных значений 301
Автосохранение списка контрольных значений 302
13.3 Осциллограф 303
Обзор 304
Открытие и закрытие осциллографа 305
Добавление элементов в осциллограф 305
Удаление переменной 309

Выборка переменных 309
Управление сбором и отображением данных 310
Изменение частоты опроса 318
Сохранение и печать графика 318
13.4 Режим редактирования и отладки 321
Обзор 321
13.5 Отладка в режиме реального времени 322
Обзор 322
Моделирование SFC 323
Моделирование LD 324
Моделирование FBD 324
Моделирование IL и ST 325
13.6 Триггеры 326
Окно триггера 326
Отладка с использованием окон триггеров 332
13.7 Графические триггеры 347
Окно графического триггера 347
Отладка с помощью окна графического триггера 354
Глава 14. Справочник по языку 371
14.1 Общие элементы 372
Обзор 372
Базовые элементы 372
Элементарные типы данных 373
Производные типы данных 373
Литералы 376
Переменные 378
Программные модули 382
Операторы и стандартные блоки 385
14.2 Список инструкций (IL) 404
Обзор 404
Синтаксис и семантика 404
Стандартные операторы 406
Вызов функций и функциональных блоков 407
14.3 Функциональная блок-схема (FBD) 409
Обзор 409
Представление линий и блоков 409

Направление движения в сетях 410
Вычисление сетей 410
Элементы управления исполнением 412
14.4 Релейная схема (LD) 414
Обзор 414
Шины питания 415
Соединительные элементы и состояния 415
Контакты 416
Катушки 417
Операторы, функции и функциональные блоки 418
14.5 «Структурный текст» (ST) 419
Обзор 419
Выражения 419
Инструкции в языке ST 421
Присвоения 422
Инструкции вызова функций и функциональных блоков 423
Инструкции выбора 425
Инструкции итерирования 427
14.6 Последовательная функциональная схема (SFC) 430
Обзор 430
Шаги 431
Переходы 433
Правила изменения 434
Управляющие флаги SFC 439
Проверка программного модуля SFC из других программ 440
14.7 Языковые расширения EcoStruxure Machine Expert - HVAC 442
Обзор 442
Макросы 442
Указатели 443
Инструкция ожидания 444
Глава 15. Справочник по ошибкам 445
15.1 Сообщения об ошибках компиляции 446
Обзор 446
Глоссарий 467
Указатель 475

Информация по технике безопасности

Важная информация

ПРИМЕЧАНИЕ

Внимательно прочтите эти инструкции и изучите оборудование перед его установкой, эксплуатацией и обслуживанием. Следующие специальные сообщения могут появляться в технической документации или на оборудовании, чтобы предупредить о возможной опасности или привлечь ваше внимание к информации, которая разъясняет или упрощает процедуру.

Добавление этого символа к знакам безопасности «Опасно» и «Предупреждение» указывает на то, что при несоблюдении инструкций возможно получение травмы в результате поражения электрическим током.

Предупреждение об опасности. Его наличие свидетельствует о потенциальной опасности получения травмы. Чтобы избежать несчастных случаев, в том числе со смертельным исходом, необходимо строго следовать всем указаниям, которые помечены этим знаком.

ОПАСНО

Пометка **ОПАСНО** обозначает опасную ситуацию, которая в отсутствие предупредительных мер неизбежно приведет к несчастному случаю со смертельным исходом или тяжелыми травмами.

ПРЕДУПРЕЖДЕНИЕ

Пометка **ПРЕДУПРЕЖДЕНИЕ** обозначает опасную ситуацию, которая в отсутствие предупредительных мер может привести к несчастному случаю со смертельным исходом или тяжелыми травмами.

ВНИМАНИЕ

Пометка **ВНИМАНИЕ** обозначает опасную ситуацию, которая может привести к получению травмы незначительной или средней тяжести.

ПРИМЕЧАНИЕ

Пометка **ПРИМЕЧАНИЕ** используется для обозначения ситуаций, которые не сопровождаются риском получения травм.

ОБРАТИТЕ ВНИМАНИЕ

К монтажу, эксплуатации и обслуживанию электрического оборудования должен допускаться только квалифицированный персонал. Компания Schneider Electric не несет ответственности за любые последствия, вытекающие из использования этого материала.

Квалифицированный специалист — это человек, обладающий навыками и знаниями в области проектирования, эксплуатации и установки электрооборудования, прошедший обучение технике безопасности, что позволяет ему распознавать связанные с изделием опасные ситуации и избегать их.

Об этом документе

Краткий обзор

Содержание документа

В данном руководстве описывается использование программного обеспечения EcoStruxure™ Machine Expert — HVAC для конфигурирования, программирования и ввода в эксплуатацию приложений для поддерживаемых логических контроллеров.

Примечание о применимости

Информация, приведенная в данном руководстве, применима **только** к продуктам EcoStruxure Machine Expert - HVAC.

Этот документ был обновлен для EcoStruxure Machine Expert - HVAC версии 1.0.

Информация об экологичности и соответствии изделий соответствующим стандартам (RoHS, REACH, PEP, EOLI и т. д.) доступна на веб-сайте *www.schneider-electric.com/green-premium.*

Технические характеристики устройств, описанные в настоящем руководстве, также представлены на веб-сайте. Чтобы получить доступ к этой информации, выполните следующие действия.

Шаг	Действие
1	Откройте главную страницу сайта Schneider Electric *www.schneider-electric.com*.
2	В поле **Поиск** введите номер для заказа изделия или наименование линейки изделий. ● Не используйте символ пробела при указании номера для заказа или наименования. ● Используйте символ звездочки (*) для поиска похожих модулей.
3	Если при поиске вы указываете номер для заказа, выберите раздел **технических описаний** на странице результатов поиска и нажмите на изделие под нужным номером для заказа. Если при поиске вы указываете наименование линейки изделий, выберите **Категории продукции** на странице результатов поиска и выберите нужную линейку продукции.
4	Если при поиске по **продуктам** выпадает несколько результатов, выберите номер для заказа нужного изделия.
5	В зависимости от размеров экрана может потребоваться прокрутка вниз, чтобы увидеть техническое описание.
6	Нажмите **Технические данные продукта**, чтобы сохранить или распечатать техническое описание изделия в формате PDF.

Характеристики, представленные в этом руководстве, должны соответствовать характеристикам, приведенным на сайте. В соответствии с политикой непрерывного развития содержимое подлежит периодическому обновлению и уточнению. Если вы обнаружите различия между информацией, приведенной в руководстве и на веб-сайте, пользоваться в качестве справки следует информацией, представленной на веб-сайте.

Сопутствующие документы

Наименование документа	Номер для заказа
Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized)	EIO0000002032 (ENG)
EcoStruxure Machine Expert HVAC&R Function Library User Guide (Руководство по работе с библиотекой функций EcoStruxure Machine Expert HVAC&R)	EIO0000002057 (ENG)
TM172 Optimized & Performance 7/18 IO Instruction Sheet (Инструкция к TM172 Optimized и Performance на 7 и 18 входов/выходов)	QGH90428
TM172 Performance 28/42 IO Instruction Sheet (Инструкция к TM172 Performance на 28 и 42 входа/выхода)	NHA87740
TM172 Optimized & Performance Isolated 28/42 IO Instruction Sheet (Инструкция к TM172 Optimized и Performance на 28 и 42 входа/выхода с изоляцией)	PHA83703
TM172 Optimized & Performance Expansion 12/28 IO Instruction Sheet (Инструкция к модулю расширения TM172 Optimized и Performance на 12 и 28 входов/выходов)	QGH26895
TM172DCLW••• Display Color Touchscreen Instruction Sheet (Инструкция к цветному сенсорному экрану TM172DCLW•••)	QGH26896
TM172DCLF• Display Color Touchscreen Flush Mounting Instruction Sheet (Инструкция к цветному сенсорному экрану TM172DCLF• скрытого монтажа)	PHA38669

Эти и другие технические документы доступны для загрузки на сайте компании www.schneider-electric.com/en/download.

Информация об изделии

⚠ ПРЕДУПРЕЖДЕНИЕ

ПОТЕРЯ УПРАВЛЕНИЯ

- Разработчик схемы управления должен учитывать возможные режимы отказа цепей управления и для определенных критически важных функций управления предусмотреть средства, обеспечивающие их безопасную работу во время и после возникновения неисправности. Примерами критически важных функций управления являются аварийный останов, останов при выбеге, отключение электропитания и перезапуск.
- Для критически важных функций управления следует предусмотреть разделение или резервирование цепей управления.
- Цепи управления системой могут включать каналы связи. Следует учесть влияние непредвиденных задержек передачи и неисправностей каналов связи.
- Кроме того, необходимо соблюдать все правила предупреждения несчастных случаев и местные правила безопасности[1].
- До ввода в эксплуатацию нужно в отдельном порядке тщательно проверить правильность работы установленного оборудования.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

[1] Дополнительные сведения доступны в последней редакции стандарта NEMA ICS 1.1: Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control (NEMA ICS 1.1 «Руководство по обеспечению безопасности в процессе эксплуатации, монтажа и обслуживания полупроводниковых систем управления»), последней редакции NEMA ICS 7.1: Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems (NEMA ICS 7.1 «Нормативы безопасности конструкции и руководство по выбору, установке и эксплуатации систем электроприводов с регулируемой скоростью») или аналогах этих стандартов в вашем регионе.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Используйте только программное обеспечение, допущенное компанией Schneider Electric для совместного применения с данным оборудованием.
- Обновляйте приложение каждый раз, когда вы изменяете физическую конфигурацию оборудования.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Терминология, заимствованная из стандартов

Технические термины, терминология, символы и соответствующие описания, приведенные в этом руководстве или встречающиеся в или на самих продуктах, как правило, соответствуют терминам и определениям, указанным в международных стандартах.

В сфере систем функциональной безопасности, электроприводов и общей автоматизации к ним могут относиться в числе прочих такие термины, как: *противоаварийная защита, функция противоаварийной защиты, безопасное состояние, отказ, сброс отказа, неисправность, сбой, ошибка, сообщение об ошибке, опасный* и т. д.

В число этих стандартов входят следующие.

Стандарт	Описание
МЭК 61131-2:2007 (ГОСТ IEC 61131-2-2012)	Контроллеры программируемые. Часть 2. Требования к оборудованию и испытания
ИСО 13849-1:2008 (ГОСТ ISO 13849-1-2014)	Безопасность оборудования. Элементы систем управления, связанные с безопасностью. Общие принципы конструирования
МЭК 61496-1:2013 (ГОСТ IEC 61496-1-2016)	Безопасность оборудования. Защитная электрочувствительная аппаратура. Часть 1. Общие требования и испытания
ИСО 12100:2010 (ГОСТ Р 54125-2010)	Безопасность машин. Общие принципы конструирования. Оценка и снижение рисков
МЭК 60204-1:2006 (ГОСТ Р МЭК 60204-1-2007)	Безопасность машин. Электрооборудование машин и механизмов. Часть 1. Общие требования
EN 1088:2008 ИСО 14119:2013	Безопасность машин. Устройства блокировочные, соединенные с предохранительными приспособлениями. Основные принципы проектирования и выбора
ИСО 13850:2006	Безопасность оборудования. Аварийный останов. Принципы конструирования
МЭК 62061:2005 (ГОСТ Р МЭК 62061-2015)	Безопасность оборудования. Функциональная безопасность электрических, электронных и программируемых электронных систем управления, связанных с безопасностью
МЭК 61508-1:2010 (ГОСТ Р МЭК 61508-1-2012)	Функциональная безопасность электрических, электронных, программируемых электронных систем, связанных с безопасностью. Общие требования
МЭК 61508-2:2010 (ГОСТ Р МЭК 61508-2-2012)	Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 2. Требования к системам
МЭК 61508-3:2010 (ГОСТ IEC 61508-3-2018)	Функциональная безопасность электрических, электронных, программируемых электронных систем, связанных с безопасностью. Часть 3. Требования к программному обеспечению
МЭК 61784-3:2008 (ГОСТ Р МЭК 61784-3-2015)	Промышленные сети. Профили. Часть 3. Функциональная безопасность полевых шин. Общие правила и определения профилей

Стандарт	Описание
2006/42/EC	Директива о безопасности машин и оборудования
2014/30/EU	Директива об электромагнитной совместимости
2014/35/EU	Директива по низковольтному электрооборудованию

В дополнение к этому в настоящем документе могут косвенно использоваться термины из других стандартов, например.

Стандарт	Описание
Группа МЭК 60034	Вращающиеся электрические машины
Группа МЭК 61800	Системы силовых электроприводов с регулируемой скоростью
Группа МЭК 61158	Цифровая передача данных для измерения и контроля. Промышленные сети для использования в промышленных системах управления

Наконец, термин *зона эксплуатации* может использоваться в сочетании с описанием конкретных опасностей и определяется как *опасная зона* или *зона риска* согласно *Директиве ЕС о безопасности машин и оборудования (2006/42/EC)* и *стандарту ИСО 12100:2010 (ГОСТ Р 54125-2010).*

ПРИМЕЧАНИЕ. Указанные выше стандарты могут быть применимы или не применимы к конкретным продуктам, указанным в этом документе. Для получения дополнительной информации по поводу применения конкретных стандартов к описанным в данном документе продуктам обратитесь к таблицам характеристик для продуктов с соответствующими номерами для заказа.

Часть 1

Инструкция по началу работы с EcoStruxure Machine Expert - HVAC

Содержание данной части

Эта часть содержит следующие главы.

Глава	Название главы	Страница
1	Инструкция по началу работы с EcoStruxure Machine Expert - HVAC	21
2	Интерфейс программы	31
3	Управление проектами	51

Глава 1

Инструкция по началу работы с EcoStruxure Machine Expert - HVAC

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
1.1	Введение в EcoStruxure Machine Expert - HVAC	22
1.2	Системные требования и поддерживаемые устройства	24

Раздел 1.1
Введение в EcoStruxure Machine Expert - HVAC

Общее описание

Обзор

Средство разработки EcoStruxure Machine Expert - HVAC (TM171SW) позволяет создавать и редактировать программы на языках МЭК 61131-3 (ГОСТ Р МЭК 61131-3) для различных областей применения. Вы можете загрузить EcoStruxure Machine Expert - HVAC *в центре загрузок на веб-сайте Schneider Electric.* Среда ориентирована на разработку приложений в области ОВКВ и охлаждения.

Возможности

EcoStruxure Machine Expert - HVAC является интегрированной средой разработки с поддержкой всех языков, определенных в стандарте МЭК 61131-3 (ГОСТ Р МЭК 61131-3).

EcoStruxure Machine Expert - HVAC предлагает пользователю следующие возможности для разработки приложений.

- Текстовые редакторы исходного кода для таких языков программирования, как:
  - список инструкций (Instruction List, IL) *(см. стр. 404);*
  - структурный текст (Structured Text, ST) *(см. стр. 419).*
- Графические редакторы исходного кода для таких языков программирования, как:
  - язык релейных схем (Ladder Diagram, LD) *(см. стр. 414);*
  - язык функциональных блок-схем (Function Block Diagram, FBD) *(см. стр. 409);*
  - язык последовательных функциональных схем (Sequential Function Chart, SFC) *(см. стр. 430).*
- Компилятор, который транслирует код приложений на языке стандарта МЭК непосредственно в объектный код, что избавляет от необходимости в интерпретаторе и позволяет выполнять программу максимально быстро.
- Коммуникационная система, позволяющая загрузить приложение в целевую среду.
- Средства отладки, начиная с простого в использовании окна контрольных значений и заканчивая более мощными инструментами, позволяющими осуществлять выборку быстро изменяющихся данных непосредственно в целевой среде.

Программный компонент EcoStruxure Machine Expert - HVAC (TM171SW)

Программное обеспечение EcoStruxure Machine Expert - HVAC позволяет:

- создавать и управлять библиотеками, приложениями и процедурами диагностики;
- управлять ранее созданными проектами, выгружать/загружать проекты и изменять параметры устройства с использованием порта связи.

Программное обеспечение EcoStruxure Machine Expert - HVAC содержит четыре вкладки:

- CONFIGURATION (КОНФИГУРИРОВАНИЕ)
- PROGRAMMING (ПРОГРАММИРОВАНИЕ)
- DISPLAY (ОТОБРАЖЕНИЕ)
- COMMISSIONING (ВВОД В ЭКСПЛУАТАЦИЮ)

Дополнительную информацию о вкладках EcoStruxure Machine Expert - HVAC см. в разделе «Вкладки EcoStruxure Machine Expert - HVAC» *(стр. 37)*.

Раздел 1.2

Системные требования и поддерживаемые устройства

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Требования к системе	25
Регистрация ПО EcoStruxure Machine Expert - HVAC	26
Поддерживаемые устройства	27
Принадлежности для подключения и передачи данных	29

Требования к системе

Обзор

Минимальные системные требования к персональному компьютеру, на который устанавливается программное обеспечение EcoStruxure Machine Expert - HVAC:

- Процессор Intel Pentium 1,6 ГГц или выше
- 1 ГБ ОЗУ, предпочтительно 2 ГБ
- 500 МБ пространства на жестком диске для стандартной установки; 1 ГБ дополнительного пространства для установки библиотек и соответствующих приложений
- Разрешение дисплея 800 x 600 пикселей, предпочтительно 1280 x 768 пикселей
- USB-интерфейс для подключения устройств
- 32- или 64-разрядная версия одной из следующих операционных систем:
  - Microsoft Windows 7
  - Microsoft Windows 8
  - Microsoft Windows 8.1
  - Microsoft Windows 10

Регистрация ПО EcoStruxure Machine Expert - HVAC

Обзор

Вы можете использовать программное обеспечение EcoStruxure Machine Expert - HVAC в течение 30 дней. После этого его необходимо будет зарегистрировать. В ходе регистрации вы получите код авторизации для использования программного обеспечения.

Зарегистрировав программное обеспечение EcoStruxure Machine Expert - HVAC, вы сможете получать техническую поддержку и обновления.

Регистрация

Чтобы зарегистрировать EcoStruxure Machine Expert - HVAC в процессе установки, выполните следующие действия.

Шаг	Действие
1	Нажмите кнопку **Register now (Зарегистрировать сейчас)** в верхней части окна **Welcome (Приветствие).**
2	Следуйте указаниям мастера регистрации. Нажмите кнопку **Help (Справка)** для получения более подробной информации.

Чтобы зарегистрировать EcoStruxure Machine Expert - HVAC после установки, выполните следующие действия.

Шаг	Действие
1	Выберите команду **Help (Справка)** → **Registration (Регистрация).**
2	Нажмите кнопку **Activate (Активировать).**
3	Следуйте указаниям мастера регистрации. Нажмите кнопку **Help (Справка)** для получения более подробной информации.

Для просмотра информации о лицензионном ключе, установленном на вашем компьютере, нажмите **About (О программе)** в окне **Welcome (Приветствие).**

Поддерживаемые устройства

Логические контроллеры

Дополнительную информацию о логических контроллерах см. в следующих руководствах по работе с оборудованием.

Номер для заказа	Описание	Руководство по работе с оборудованием
TM171O•••••	Логический контроллер Modicon M171 Optimized	*Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized)*
TM171P•••••	Логический контроллер Modicon M171 Performance	*Modicon M171 Performance Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Performance)*
TM172O•••••	Логический контроллер Modicon M172 Optimized	*Modicon M172 Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M172)*
TM172P••••••	Логический контроллер Modicon M172 Performance	

Модули расширения

Дополнительную информацию о модулях расширения и совместимости с ними см. в следующих руководствах по работе с оборудованием.

Номер для заказа	Описание	Руководство по работе с оборудованием
TM171EO14R	Модуль расширения M171 Optimized на 14 входов/выходов	*Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized)*
TM171EO15R	Модуль расширения M171 Optimized на 15 входов/выходов	*Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171Optimized)*
TM171EO22R	Модуль расширения M171 Optimized на 22 входов/выходов	*Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized)*
TM172E12R	Модуль расширения Modicon M172 Optimized и Performance на 12 входов/выходов	*Modicon M172 Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M172)*
TM172E28R	Модуль расширения Modicon M172 Optimized и Performance на 28 входов/выходов	*Modicon M172 Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M172)*
TM171EP14R	Модуль расширения Modicon M171 Performance на 14 входов/выходов	*Modicon M171 Performance Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Performance)*
TM171EP27R	Модуль расширения Modicon M171 Performance на 27 входов/выходов	

Дисплеи

Дополнительную информацию о дисплеях и их совместимости см. в следующих руководствах по работе с оборудованием.

Номер для заказа	Описание	Руководство по работе с оборудованием
TM171DLED	LED-дисплей для M171 Optimized	*Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized)*
TM171DLCD2U	LCD-дисплей для M171 Optimized	*Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized)*
TM171DWAL2U	Настенный термостат без подсветки для M171 Optimized	*Modicon M171 Optimized Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized)*
TM171DWAL2L	Настенный термостат с подсветкой для M171 Optimized	
TM171DGRP	Графический дисплей для M171 Performance	*Modicon M171 Performance Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M171 Performance)*
TM172DCLFG	Цветной сенсорный выносной дисплей для M172 (скрытый монтаж), белый	*Modicon M172 Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M172)*
TM172DCLFW	Цветной сенсорный выносной дисплей для M172 (скрытый монтаж), серый	
TM172DCLWT	Цветной сенсорный выносной дисплей для M172 (вертикальный монтаж), со встроенным датчиком температуры	*Modicon M172 Logic Controller Hardware Guide (Руководство по работе с оборудованием для логического контроллера Modicon M172)*
TM172DCLWTH	Цветной сенсорный выносной дисплей для M172 (вертикальный монтаж), со встроенными датчиками температуры и влажности	
TM172DCLWTHP	Цветной сенсорный выносной дисплей для M172 (вертикальный монтаж), со встроенными датчиками температуры, влажности и присутствия (ПИК)	

Принадлежности для подключения и передачи данных

Обзор

Для подключения целевого устройства к персональному компьютеру могут потребоваться дополнительные принадлежности.

Подключение к персональному компьютеру позволяет осуществлять отладку, ввод в эксплуатацию, загрузку и выгрузку приложения.

⚠ ПРЕДУПРЕЖДЕНИЕ

АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК КОНТРОЛЛЕРА

- Прежде чем загружать приложение, сначала проверьте состояние машины или процесса.
- Перед загрузкой приложения убедитесь в отсутствии опасностей, способных привести к травмированию людей, находящихся внутри или рядом с машиной или в непосредственной близости от технологического процесса.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Подключение M171O к персональному компьютеру

M171O можно подключить к персональному компьютеру, используя интерфейс USB/TTL-I2C, чтобы:

- работать с программным обеспечением;
- подключиться к целевому устройству и управлять им;
- управлять программирующим флеш-накопителем.

Программирующий флеш-накопитель — это носитель данных, позволяющий выполнить следующие действия.

- Обновление микропрограммы на целевом устройстве.
- Обновление приложения контроллера на целевом устройстве.
- Обновление значений параметров на целевом устройстве.
- Загрузка значений параметров с целевого устройства.

Поставляются следующие соединительные кабели.

- Желтый кабель с разъемами JST — Molex.
- Синий кабель с разъемами JST — JST.
- Удлинитель USB-A/A, 2 м.

Подключение M172 к персональному компьютеру

M172 можно подключить к персональному компьютеру через USB-порт и USB-кабель.

- USB-порт типа Mini-B (УСТРОЙСТВО). Используется для подключения TM172P•••••• / TM172O••••• к персональному компьютеру с помощью USB-кабеля Mini-B/A для отладки, ввода в эксплуатацию, загрузки и выгрузки с помощью EcoStruxure Machine Expert - HVAC.
- USB типа Micro-B (УСТРОЙСТВО). Используется для подключения TM172DCL•••• к персональному компьютеру с помощью USB-кабеля Micro-B/A для отладки, ввода в эксплуатацию, загрузки и выгрузки с помощью EcoStruxure Machine Expert— HVAC.

Также можно использовать USB-порт типа A (ХОСТ) на контроллере для подключения USB-накопителя с целью загрузки приложения.

TM172P•••••• / TM172O••••• также могут получать питание через USB-кабель, однако функции, связанные с отладкой, вводом в эксплуатацию, загрузкой и выгрузкой с помощью EcoStruxure Machine Expert - HVAC, будут ограничены.

Для получения более подробной информации см. *Руководство по работе с оборудованием для логического контроллера Modicon M172*.

Подключение M171P к персональному компьютеру

M171P можно подключить к персональному компьютеру через USB-порт и USB-кабель.

- USB типа A (ХОСТ). Используется для подключения USB-накопителя с целью загрузки приложения или его экспорта, если приложение, запущенное на ПЛК, поддерживает эту функцию.
- USB-порт типа Mini-B (УСТРОЙСТВО). Используется для подключения M171P к персональному компьютеру с помощью USB-кабеля Mini-B/A для

  отладки, ввода в эксплуатацию, загрузки и выгрузки с помощью EcoStruxure Machine Expert - HVAC.

Конвертеры для программирования M171P/M172

Для связи с контроллером можно также использовать адаптер USB/485 TSXCUSB485 с кабелем VW3A83O6D3O.

При наличии последовательного порта RS232 устройство M171P/M172 также можно подключить к персональному компьютеру с помощью адаптера RS485/RS232.

Коммуникационные модули M171P/M172

Широкий спектр коммуникационных модулей позволяет реализовать интеграцию с промышленными системами, системами управления зданиями (BMS) и сетями Ethernet.
ПРИМЕЧАНИЕ. Недоступно для M171P с креплением для скрытого монтажа.
Для получения более подробной информации см. описание коммуникационных модулей *(см. стр. 130)*.

Глава 2
Интерфейс программы

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
2.1	Основы пользовательского интерфейса EcoStruxure Machine Expert - HVAC	32
2.2	Настройка рабочего пространства	41

Раздел 2.1

Основы пользовательского интерфейса EcoStruxure Machine Expert - HVAC

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Создание проектов в EcoStruxure Machine Expert - HVAC	33
Написание программ в EcoStruxure Machine Expert - HVAC	34
Навигация в EcoStruxure Machine Expert - HVAC	35
Режимы работы	39

Создание проектов в EcoStruxure Machine Expert - HVAC

Обзор

EcoStruxure Machine Expert - HVAC представляет собой графический инструмент программирования, предназначенный для конфигурирования, разработки и ввода в эксплуатацию программ для логических контроллеров.

Терминология EcoStruxure Machine Expert - HVAC

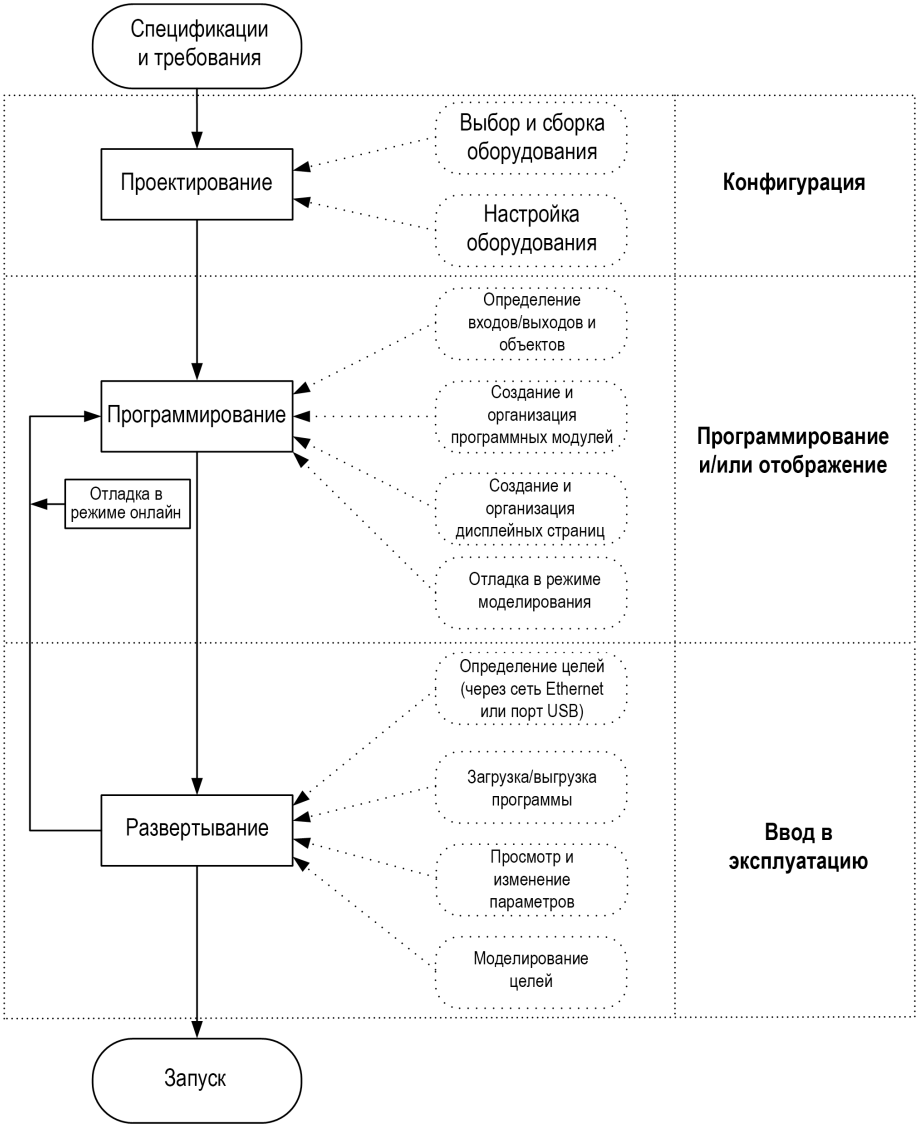
В EcoStruxure Machine Expert - HVAC используются следующие термины.

- **Проект.** Проект EcoStruxure Machine Expert - HVAC содержит:
  - информацию о разработчике и цель проекта;
  - конфигурацию логического контроллера и соответствующих модулей расширения, на которые ориентирован проект;
  - исходный код программы, символы, комментарии, документацию и прочую сопутствующую информацию.
- **Приложение**. Содержит части проекта, которые загружаются в логический контроллер, включая скомпилированную программу и конфигурацию оборудования.
- **Программа**. Скомпилированный исходный код, запускаемый на логическом контроллере.
- **Программный модуль** (POU, Program Organization Unit). Объект многоразового использования, содержащий объявленные переменные и набор инструкций, используемых в программе.
- **Целевое устройство**. Устройство, подключенное к персональному компьютеру.

Написание программ в EcoStruxure Machine Expert - HVAC

Введение

На следующей диаграмме представлены стандартные этапы разработки проекта в EcoStruxure Machine Expert - HVAC:



Навигация в EcoStruxure Machine Expert - HVAC

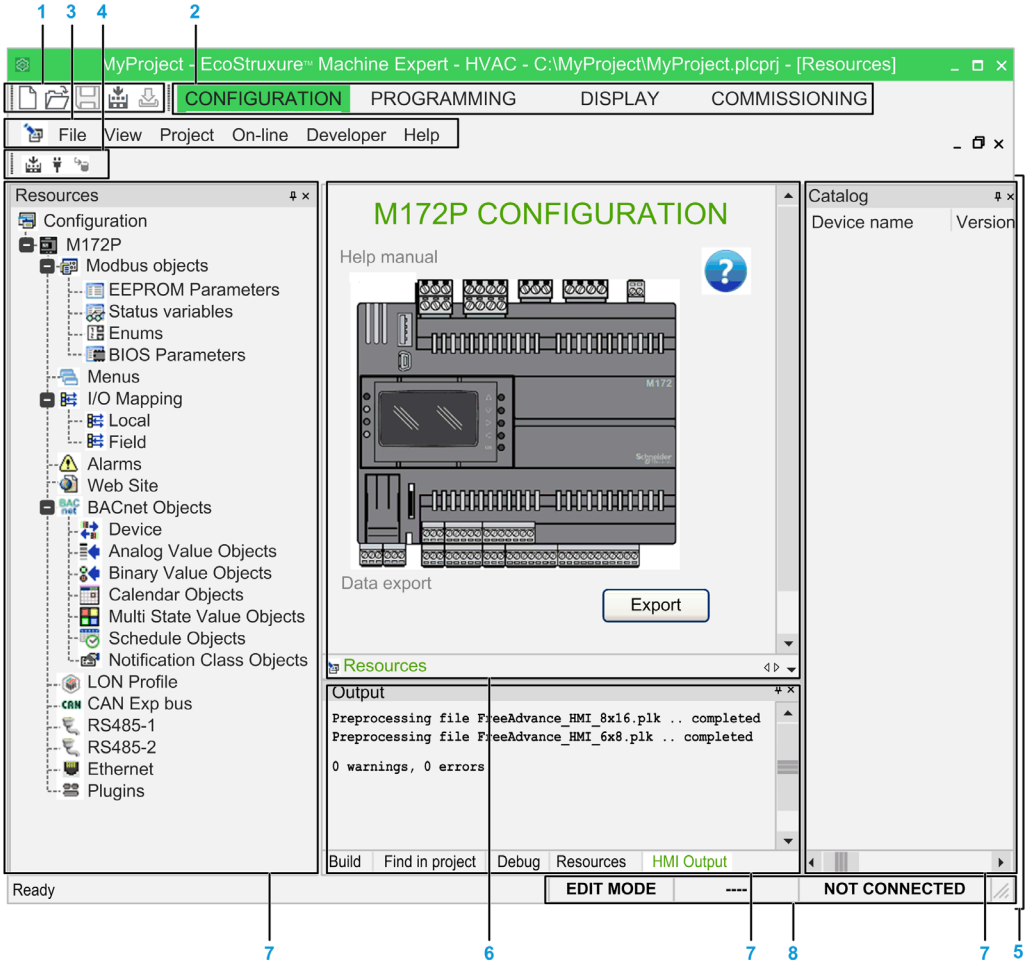
Окно Welcome (Приветствие)

Окно **Welcome (Приветствие)** всегда отображается при запуске EcoStruxure Machine Expert - HVAC. С помощью этого окна можно выбрать и открыть существующий проект или создать новый *(см. стр. 53)*.

Главное окно

Главное окно EcoStruxure Machine Expert - HVAC предоставляет доступ к меню и командам, окнам и панелям инструментов.

На рисунке ниже представлено главное окно EcoStruxure Machine Expert - HVAC:



Область	Описание
1	Панель инструментов проекта *(см. стр. 36)*
2	Вкладки *(см. стр. 37)*
3	Панель меню (содержимое панелей меню зависит от выбранной вкладки, *см. стр. 37)*
4	Панели инструментов *(см. стр. 37)*
5	Рабочее пространство *(см. стр. 41)*
6	Окно редактора
7	Окна инструментов *(см. стр. 38)*
8	Строка состояния *(см. стр. 162)*

Панель инструментов проекта

Панель инструментов проекта расположена в левой верхней области главного окна и обеспечивает доступ к основным командам с помощью значков.

Значок	Описание
	New project (Новый проект): создает новый проект, любой открытый проект должен быть при этом закрыт. Если проект не сохранен, в диалоговом окне выводится запрос на сохранение проекта. Для получения дополнительной информации см. раздел «Создание нового проекта» *(стр. 53)*.
	Open project (Открыть проект): открывает проект, сохраненный на компьютере. Для получения дополнительной информации см. раздел «Открытие существующего проекта» *(стр. 59)*.
	Save project (Сохранить проект): сохраняет изменения в текущем проекте. Для получения дополнительной информации см. раздел «Сохранение проекта» *(стр. 57)*.
	Build all (Собрать все): компилирует ваш проект. Для получения дополнительной информации см. раздел «Сборка» *(стр. 64)*.
	Download all (Загрузить все): загружает на целевое устройство. Для получения дополнительной информации см. раздел «Загрузка и выгрузка приложений» *(стр. 65)*.

Вкладки EcoStruxure Machine Expert - HVAC

EcoStruxure Machine Expert - HVAC предоставляет четыре среды разработки для программирования контроллеров:

- CONFIGURATION (КОНФИГУРИРОВАНИЕ) *(см. стр. 69)*
  Для создания сетей. Является точкой входа для программного обеспечения.
- PROGRAMMING (ПРОГРАММИРОВАНИЕ) *(см. стр. 139)*
  Для создания и управления библиотеками, приложениями контроллера и процедурой диагностики.

  Включает в себя **режим моделирования** для разработчиков программного обеспечения, в котором можно запускать и тестировать приложения без необходимости физического подключения логического контроллера и модулей расширения.
- DISPLAY (ОТОБРАЖЕНИЕ)
  Для создания графического интерфейса, отображаемого на встроенных и выносных дисплеях.
- COMMISSIONING (ВВОД В ЭКСПЛУАТАЦИЮ)
  Для загрузки проекта в целевое устройство (логический контроллер) и изменения параметров устройства через последовательный порт.

Панели инструментов

EcoStruxure Machine Expert - HVAC содержит несколько панелей инструментов для каждой вкладки.

Панели инструментов	Вкладки
Main (Главная)	Все
Project (Проект)	PROGRAMMING (ПРОГРАММИРОВАНИЕ) *(см. стр. 155)*
Debug (Отладка)	
FBD	
SFC	
LD	
Network (Сеть)	
Configuration (Конфигурирование)	CONFIGURATION (КОНФИГУРИРОВАНИЕ) *(см. стр. 79)*
HMI Page (Страница ЧМИ)	DISPLAY (ОТОБРАЖЕНИЕ)
HMI Project (Проект ЧМИ)	
HMI Profiles (Профили ЧМИ)	
Commissioning (Ввод в эксплуатацию)	COMMISSIONING (ВВОД В ЭКСПЛУАТАЦИЮ)

Сведения об управлении панелями инструментов см. в разделе «Управление панелями инструментов» *(стр. 43)*.

Окна инструментов

EcoStruxure Machine Expert - HVAC содержит несколько окон инструментов для каждой вкладки.

Окна инструментов	Вкладки
Local variables (Локальные переменные)	CONFIGURATION (КОНФИГУРИРОВАНИЕ) *(см. стр. 69)*
Project (Проект)	PROGRAMMING (ПРОГРАММИРОВАНИЕ) *(см. стр. 139)*
Watch (Контрольные значения)	
Properties Window (Окно свойств)	
Oscilloscope (Осциллограф)	
PLC run-time status (Статус работы ПЛК)	
Operators and blocks (Операторы и блоки)	
Дерево библиотеки (Дерево библиотеки)	
Output (Вывод)	CONFIGURATION (КОНФИГУРИРОВАНИЕ) *(см. стр. 69)* PROGRAMMING (ПРОГРАММИРОВАНИЕ) *(см. стр. 139)*
Cross Reference (Перекрестные ссылки)	
Resources (Ресурсы)	CONFIGURATION (КОНФИГУРИРОВАНИЕ) *(см. стр. 69)*
Catalog (Каталог)	
HMI Project (Проект ЧМИ)	DISPLAY (ОТОБРАЖЕНИЕ)
HMI Properties (Свойства ЧМИ)	
HMI Actions (Действия ЧМИ)	
HMI Vars and Parameters (Переменные и параметры ЧМИ)	
HMI Templates (Шаблоны ЧМИ)	
Commissioning (Ввод в эксплуатацию)	COMMISSIONING (ВВОД В ЭКСПЛУАТАЦИЮ)
Commissioning Watch (Контрольные значения для ввода в эксплуатацию)	
Commissioning Oscilloscope (Осциллограф для ввода в эксплуатацию)	

Сведения об управлении окнами инструментов см. в разделе «Управление окнами инструментов» *(стр. 44)*.

Режимы работы

Введение

В различных режимах работы доступны средства для разработки, отладки, мониторинга и изменения приложения при отключенном или подключенном к EcoStruxure Machine Expert - HVAC контроллере.

Доступны следующие режимы:

- Офлайн-режим
- Онлайн-режим
- Режим моделирования

Офлайн-режим

EcoStruxure Machine Expert - HVAC работает в офлайн-режиме, если физическое подключение к логическому контроллеру не установлено.

При использовании EcoStruxure Machine Expert - HVAC в офлайн-режиме сначала выполняется настройка под целевые аппаратные компоненты, а затем разрабатывается приложение.

Онлайн-режим

EcoStruxure Machine Expert - HVAC работает в онлайн-режиме, если логический контроллер физически подключен к персональному компьютеру.

В онлайн-режиме вы можете загрузить свое приложение на логический контроллер (загрузка и выгрузка приложения недоступны в режиме моделирования). После этого EcoStruxure Machine Expert - HVAC синхронизирует приложение в памяти компьютера с версией, хранящейся в логическом контроллере, позволяя отслеживать, контролировать и изменять приложение.

⚠ ПРЕДУПРЕЖДЕНИЕ

АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК КОНТРОЛЛЕРА

- Прежде чем загружать приложение, сначала проверьте состояние машины или процесса.
- Перед загрузкой приложения убедитесь в отсутствии опасностей, способных привести к травмированию людей, находящихся внутри или рядом с машиной или в непосредственной близости от технологического процесса.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Определенные элементы программы можно изменять в онлайн-режиме. Например, можно добавлять или удалять ступеньки, а также изменять значения определенных параметров функционального блока.

ПРИМЕЧАНИЕ. Возможность изменения программы в онлайн-режиме зависит от предустановленной конфигурации. Для получения дополнительной информации см. разделы «Отладка в режиме реального времени» *(стр. 322)* и «Триггеры» *(стр. 326)*.

Режим моделирования

EcoStruxure Machine Expert - HVAC работает в режиме моделирования, когда соединение устанавливается с виртуальным логическим контроллером. В режиме моделирования физическое соединение с логическим контроллером не устанавливается; вместо этого EcoStruxure Machine Expert - HVAC моделирует соединение с логическим контроллером и модулями расширения для запуска и тестирования программы.

Для получения дополнительной информации см. раздел «Моделирование» *(стр. 290)*.

Раздел 2.2
Настройка рабочего пространства

Обзор

В этом разделе рассматривается работа с элементами пользовательского интерфейса программного обеспечения. Вы можете настроить интегрированную среду программного обеспечения таким образом, чтобы она наилучшим образом соответствовала конкретно вашему процессу разработки.

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Компоновка	42
Панели инструментов	43
Управление окнами инструментов	44
Управление окнами	45
Полноэкранный режим	46
Параметры программного обеспечения	47

Компоновка

Обзор

Компоновка рабочего пространства программного обеспечения настраивается свободно, в соответствии с вашими потребностями.

Конфигурация компоновки сохраняется при выходе из приложения. Внесенные вами изменения будут сохранены по завершении сеанса работы.

Сброс компоновки

Чтобы сбросить параметры компоновки до значений по умолчанию, соответствующих стандартной компоновке, выполните следующие действия.

Шаг	Действие
1	Нажмите **File (Файл)** → **Options… (Параметры…)**. Откроется диалоговое окно **Program options (Параметры программы)**.
2	В области **Tool windows (Окна инструментов)** нажмите кнопку **Reset bars positions (Сбросить позиции панелей)**. Откроется новое диалоговое окно.
3	Нажмите кнопку **OK**.
4	В диалоговом окне **Program options (Параметры программы)** нажмите кнопку **OK**.
5	На панели инструментов проекта нажмите значок **Save project (Сохранить проект)**.
6	Нажмите **File (Файл)** → **Exit (Выход)**.
7	Перезапустите EcoStruxure Machine Expert - HVAC.

ПРИМЕЧАНИЕ. При сбросе компоновки сбрасываются настройки интерфейса на всех вкладках.

Панели инструментов

Обзор

Отображаются следующие панели инструментов:

Для получения дополнительной информации см. раздел «Панели инструментов» *(стр. 155)*.

Скрытие/отображение

Чтобы отобразить или скрыть панель инструментов, выполните следующие действия.

Шаг	Действие
1	Нажмите **View (Вид)** → **Toolbars (Панели инструментов)** и выберите нужную. Или щелкните правой кнопкой мыши в области панели инструментов.
2	Во всплывающем окне отображается список доступных панелей инструментов: ☑ Main ☑ Project ☑ Debug ☑ FBD Bar ☑ SFC Bar ☑ LD bar Network Configuration HMI Page HMI Project HMI Profils Commissioning
3	Щелкните по имени панели инструментов, которую необходимо отобразить/скрыть.

Перемещение

Чтобы переместить панель инструментов, щелкните по ее левому краю и перетащите на новое место.

Управление окнами инструментов

Отображение/скрытие окон инструментов

Чтобы отобразить/скрыть окно инструментов, выполните следующие действия.

Шаг	Действие
1	Нажмите **View (Вид)** → **Tool windows (Окна инструментов)** или щелкните правой кнопкой мыши в области окна инструментов. Откроется всплывающее окно.
2	Выберите панель инструментов, чтобы отобразить ее или скрыть.

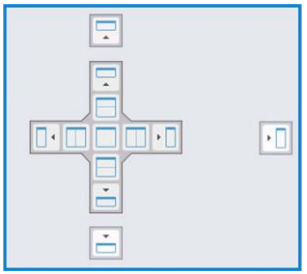
Отстыковка окон инструментов

Чтобы отстыковать окно инструментов от объекта по умолчанию, щелкните по его заголовку и перетащите в новое место.

Чтобы вернуть окно инструментов в последнее место стыковки, дважды щелкните по заголовку окна.

Пристыковка окон инструментов

Чтобы пристыковать окно инструментов к другому объекту, выполните следующие действия.

Шаг	Действие
1	Щелкните по заголовку окна инструментов и переместите курсор. На экране отобразится направляющий крест, переместите курсор на нужную часть направляющего креста. Соответствующая часть будет затемнена. 
2	Отпустите кнопку мыши.

Автоматическое отображение/скрытие окон инструментов

Чтобы автоматически скрывать окно инструментов, нажмите на булавку в правом верхнем углу окна инструментов.

Окно инструментов свернется во вкладку в левом верхнем углу главного окна.

Чтобы снова отобразить окно инструментов, щелкните по вкладке.

Чтобы переключить окно инструментов из режима автоматического скрытия в режим обычной стыковки, повторно нажмите на булавку при открытом окне инструментов.

Управление окнами

Обзор

EcoStruxure Machine Expert - HVAC позволяет переключаться между открытыми окнами редактора исходного кода.

Вкладка Document (Документ)

Для переключения между открытыми редакторами щелкните по заголовку программы на вкладке, расположенной под окном программирования.

Меню Window (Окно)

Меню **Window (Окно)** позволяет:

- отобразить открытые программы в виде каскада;
- отобразить открытые программы в виде плиток;
- расположить значки свернутых документов в левом нижнем углу окна редактора;
- закрыть открытые программы;
- переключаться между открытыми в данный момент программами, щелкая по заголовкам программ.

Полноэкранный режим

Обзор

Чтобы включить или выключить полноэкранный режим, нажмите **View (Вид)** → **Full screen (Полноэкранный режим)** (или нажмите Ctrl+U).

В полноэкранном режиме редактор исходного кода разворачивается на всю рабочую область, облегчая составление программ с помощью графических языков программирования.

Параметры программного обеспечения

Обзор

EcoStruxure Machine Expert - HVAC позволяет настроить некоторые параметры программного обеспечения. Чтобы отобразить опции диалогового окна, нажмите **File (Файл)** → **Options… (Параметры…)**.

Общие

Вкладка General (Общее) позволяет выполнить следующие настройки:

- **Visual Theme (Тема оформления)**
  В списке **Color theme (Цветовая тема)** вы можете выбирать между двумя темами: **Standard (Стандартная)** и **Dark (Темная)**.
- **Save options (Параметры сохранения)**
  - **Autosave (Автосохранение)**: если флажок **Autosave (Автосохранение)** установлен, программное обеспечение периодически выполняет сохранение всего проекта. Вы можете указать периодичность выполнения этой задачи, указав количество минут между двумя автоматическими сохранениями в окне **Interval (min) (Интервал [мин.])**.
  - **Max previous version to keep (Макс. кол-во предыдущих версий)**: указывает максимальное количество копий проекта, которые должны быть заархивированы и сохранены в папке **PreviousVersions**.
- **Output window (Окно вывода)**
  Вы можете указать семейство и размер шрифта, используемого в окне вывода.
- **Communication (Связь)**
  Если установлен флажок **Use last port (Использовать последний порт)**, то в качестве порта по умолчанию будет выбран последний используемый порт.
- **Tooltip (Подсказка)**
  Если установлен флажок **Enable tooltip on editors (Включить всплывающие подсказки в редакторах)**, при наведении курсора на какой-либо символ в редакторе будут появляться небольшие окошки с информацией.
- **Tool windows (Окна инструментов)**
  Вы можете указать семейство и размер шрифта, используемого в окне инструментов.

  Reset bars positions (Сбросить позиции панелей): компоновка, позиции и размеры пристыкованных панелей в среде разработки сбрасываются до значений по умолчанию. Чтобы изменения вступили в силу, программное обеспечение должно быть перезапущено.
- **Source editors options (Параметры редакторов исходного кода)**
  Если установлен флажок **ST - LD: Auto declaration of variables (ST - LD: автоматическое объявление переменных)**, то в программах на языке «Структурный текст» (ST) и языке релейных схем (LD) переменные объявляются автоматически.

Graphic Editor (Графический редактор)

Эта панель позволяет редактировать свойства редакторов исходного кода для языков LD, FBD и SFC.

Вы можете указать семейство и размер шрифта, используемого в графическом редакторе.

Вы также можете изменять цвета графического объекта.

Text Editors (Текстовые редакторы)

Вы можете указать семейство и размер шрифта, используемого как в редакторе кода, так и редакторе переменных.

Язык

Чтобы изменить язык среды, выполните следующие действия.

Шаг	Действие
1	Выберите язык из списка, отображаемого на этой панели.
2	Нажмите кнопку **Select (Выбрать)**.
3	Нажмите кнопку **OK** для подтверждения.
4	Чтобы изменения вступили в силу, перезапустите программное обеспечение.

Custom Tools (Пользовательские инструменты)

В меню **Custom tools (Пользовательские инструменты)** можно добавить до 16 команд. Эти команды можно ассоциировать с любой программой, имеющейся в вашей операционной системе. Вы также можете указать аргументы для любой команды, добавленной в меню **Custom tools (Пользовательские инструменты)**.
Чтобы добавить инструмент в меню **Custom tools (Пользовательские инструменты)**, выполните следующие действия.

Шаг	Действие
1	В поле **Command (Команда)** введите полный путь к файлу программы, которую вы хотите использовать в качестве инструмента. Кроме того, вы можете выбрать файл программы, нажав кнопку обзора.
2	В поле **Arguments (Аргументы)** введите аргументы (если таковые имеются), которые будут переданы исполняемой команде, указанной на первом этапе. Их необходимо отделять пробелом.
3	В поле **Menu string (Текст в меню)** введите имя для добавляемого инструмента. Этот текст отображается в меню **Tools (Инструменты)**.
4	Нажмите кнопку **Add (Добавить)**, чтобы вставить новую команду в соответствующее меню.
5	Нажмите кнопку **OK** для подтверждения или кнопку **Cancel (Отмена)** для выхода.

Например, чтобы добавить **Калькулятор Windows** в меню **Custom tools (Пользовательские инструменты)**, выполните следующие действия.

Шаг	Действие
1	Заполните поля диалогового окна, как показано на рисунке.
2	Нажмите кнопку **Add (Добавить)**. Имя, которое вы дали этому инструменту (в данном примере **Calc**), теперь отображается в меню **Custom tools (Пользовательские инструменты)**.

Merge (Объединение)

Если установлен флажок **Enable Merge (Включить объединение)**, вы можете настроить следующие параметры:

- **Identical name (Идентичное имя)**
  - **Objects with different types (Объекты разных типов)**
  - **Object with same type (not variables) (Объекты того же типа [не переменные])**
  - **Variables (Переменные)**
- **Check address (Проверка адреса)**
  - **Overlapped (Наложение)**
  - **Copy\Paste mapped variable (Копировать/вставить сопоставленную переменную)**

Для получения дополнительной информации см. раздел «Функция объединения» *(стр. 179)* .

Глава 3
Управление проектами

Раздел 3.1
Управление проектами

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Создание нового проекта	53
Печать проекта	56
Сохранение проекта	57
Управление существующими проектами	59
Распределение проектов	60
Экспорт в файлы CSV	61
Выбор целевого устройства	62
Сборка	64
Загрузка проекта на целевое устройство	65
ПО Installer	67
Закрытие EcoStruxure Machine Expert - HVAC	68

Создание нового проекта

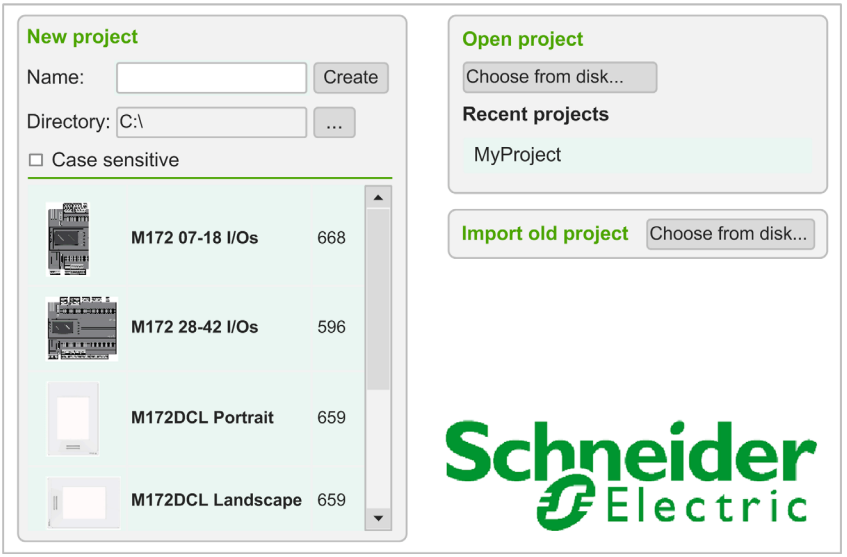
Обзор

Существует два способа создать новый проект:

- В окне Welcome (Приветствие) *(см. стр. 53)*
- В окне New project (Новый проект) *(см. стр. 55)*

Окно Welcome (Приветствие)

При запуске EcoStruxure Machine Expert - HVAC появляется окно **Welcome (Приветствие):**



Окно **Welcome (Приветствие)** разделено на три области:

- **New project (Новый проект)**
- **Open project (Открыть проект)**
- **Import old project (Импортировать старый проект)**

Чтобы создать новый проект, выполните следующие действия.

Шаг	Действие
1	В поле **Name (Название)** введите название нового проекта. Вводимый текст также является именем папки, в которой находятся файлы проекта. Название можно изменить позже, см. раздел «Параметры проекта» *(стр. 164)*.
2	В поле **Directory (Каталог)** указывается расположение этой папки по умолчанию. Нажмите кнопку обзора, чтобы выбрать другую папку.
3	В списке устройств щелкните на целевом устройстве, на котором запускается проект. **ПРИМЕЧАНИЕ.** Доступные целевые устройства перечислены в разделе «Поддерживаемые устройства» *(стр. 27)*.
4	Если установлен флажок **Case-sensitive (Чувствительность к регистру)**, то исходный код проекта будет чувствителен к регистру. Этот параметр можно изменить позже, см. раздел «Параметры проекта» *(стр. 164)*. **ПРИМЕЧАНИЕ.** Этот параметр не соответствует стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3).
5	Нажмите кнопку **Create (Создать)**.

Чтобы открыть существующий проект, используйте одну из двух процедур:

- Нажмите кнопку **Choose from disk… (Выбрать с диска…)**.

  Результат: откроется диалоговое окно, в котором можно будет перейти в директорию, содержащую проект, и выбрать соответствующий файл проекта.
- В списке **Recent projects (Последние проекты)** дважды щелкните по имени проекта.

Чтобы импортировать старый проект:

- Нажмите кнопку **Choose from disk… (Выбрать с диска…)**.

  Результат: откроется диалоговое окно, в котором можно будет перейти в директорию, содержащую проект, и выбрать соответствующий файл проекта.

  «Старым» считается проект, созданный с помощью SoMachine HVAC. EcoStruxure Machine Expert - HVAC выполнит конвертацию старой программы. Тем не менее, возможны несовместимости между SoMachine HVAC и EcoStruxure Machine Expert - HVAC.

⚠ ПРЕДУПРЕЖДЕНИЕ

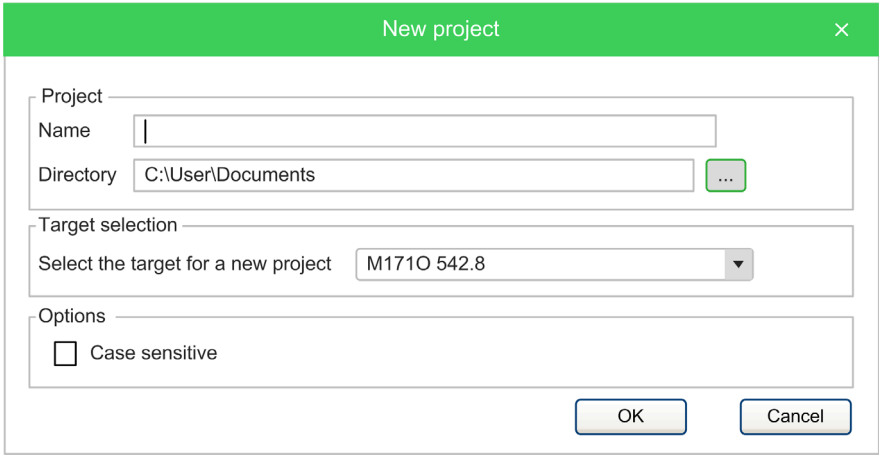
НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Всегда проверяйте, работает ли ваша прикладная программа так же, как и до конвертации. В ней должны применяться корректные значения конфигурации, параметры, значения параметров, функции и функциональные блоки.
- Измените приложение таким образом, чтобы оно работало, как и раньше.
- Тщательно протестируйте и проверьте новую скомпилированную версию, прежде чем вводить ваше приложение в эксплуатацию.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Окно New Project (Новый проект)

Нажмите **File (Файл)** → **New project (Новый проект)** или значок **File (Файл)** на панели инструментов проекта, чтобы открыть окно **New project (Новый проект)**:



ПРИМЕЧАНИЕ. Если у вас уже есть открытый проект, появится диалоговое окно, в котором будет предложено сохранить текущий проект.

Чтобы создать новый проект, выполните следующие действия.

Шаг	Действие
1	В поле **Name (Название)** введите название нового проекта. Вводимый текст также является именем папки, в которой находятся файлы проекта. Название можно изменить позже, см. раздел «Параметры проекта» *(стр. 164)*.
2	В поле **Directory (Каталог)** указывается расположение этой папки по умолчанию. Нажмите кнопку ... **Browse (Обзор)**, чтобы выбрать другую папку.
3	В списке **Select the target for a new project (Выберите целевое устройство для нового проекта)** щелкните целевое устройство, на котором запускается проект. **ПРИМЕЧАНИЕ.** Доступные целевые устройства перечислены в разделе «Поддерживаемые устройства» *(стр. 27)*.
4	Если установлен флажок **Case-sensitive (Чувствительность к регистру)**, то исходный код проекта будет чувствителен к регистру. Этот параметр можно изменить позже, см. раздел «Параметры проекта» *(стр. 164)*. **ПРИМЕЧАНИЕ.** Этот параметр не соответствует стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3).
5	Нажмите кнопку **OK**.

Печать проекта

Печать проекта

EcoStruxure Machine Expert - HVAC позволяет распечатывать данные проекта, такие как программы и переменные.
Чтобы распечатать проект, выполните следующие действия.

Шаг	Действие
1	На вкладке **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** нажмите **File (Файл)** → **Print Project (Печать проекта)**.
2	В поле **Name (Имя)** выберите принтер для печати проекта.
3	Нажмите кнопку **OK**.

Печать текущего рабочего окна

EcoStruxure Machine Expert - HVAC позволяет распечатывать только текущее рабочее окно. Чтобы распечатать текущее рабочее окно, выполните следующие действия.

Шаг	Действие
1	На вкладке **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** нажмите **File (Файл)** → **Print… (Печать…)**.
2	В поле **Name (Имя)** выберите принтер для печати проекта.
3	Нажмите кнопку **OK**.

Предпросмотр перед печатью

Чтобы просмотреть документ, выводимый на печать, нажмите **File (Файл)** → **Print preview (Предпросмотр перед печатью)**. Результат выводится в текущем рабочем окне.

Сохранение проекта

Обзор

Проекты EcoStruxure Machine Expert - HVAC можно сохранять в виде файлов на локальном компьютере или на сервере. Такие файлы имеют расширение *.plcprj или *.ppjs и содержат:

- исходный код программы;
- текущую конфигурацию оборудования;
- настройки и параметры проекта EcoStruxure Machine Expert - HVAC.

Сохранение проекта

Чтобы сохранить проект:

- Нажмите **Save project (Сохранить проект)** на панели инструментов проекта.
- Нажмите **File (Файл)** → **Save project (Сохранить проект)**.

Сохранение проекта как

Чтобы сохранить проект под другим именем, в другом формате или в другой папке, выполните следующие действия.

Шаг	Действие
1	Нажмите **File (Файл)** → **Save project As… (Сохранить проект как…)**.
2	Введите новое имя файла проекта.
3	Выберите новую папку, в которой будет храниться файл проекта.
4	Выберите новый формат *(см. стр. 60)* для файла проекта.
5	Нажмите кнопку **OK**.

Автосохранение

EcoStruxure Machine Expert - HVAC включает в себя функцию **AutoSave (Автосохранение)**, которая периодически сохраняет ваш проект во время работы над ним.

Функция **AutoSave** сохраняет данные в отдельной папке под названием **Backup**, расположенной в том же месте, что и папка проекта.

Если вы регулярно сохраняете или закрываете открытый файл, то связанный с ним файл автосохранения удаляется, если только команда сохранения не отменяется или не завершается ошибкой. В этом случае файл сохраняется. При повторном открытии проекта, для которого имеется соответствующий файл автосохранения, отображается диалоговое окно **AutoSave Backup (Автоматически сохраненная резервная копия)**. Вы можете открыть автоматически сохраненный проект или последнюю вручную сохраненную версию.

Вы можете указать интервал времени (в минутах) между сохранениями. По умолчанию функция **AutoSave** запускается каждую минуту. Для получения дополнительной информации см. раздел «Параметры сохранения» *(стр. 47)*.

Резервные копии

EcoStruxure Machine Expert - HVAC включает в себя функцию резервного копирования предыдущей версии проекта, над которым вы работаете.

При явном сохранении проекта EcoStruxure Machine Expert - HVAC сохраняет текущую версию (до сохранения) проекта в папке **PreviousVersions**, расположенной в том же месте, что и папка проекта.

Вы можете задать максимальное количество резервных копий, которые будут храниться на вашем компьютере. По умолчанию это значение равно 10. Чтобы отключить эту функцию, установите 0. Для получения дополнительной информации см. раздел «Параметры сохранения» *(стр. 47)*.

Управление существующими проектами

Открытие существующего проекта

Чтобы открыть существующий проект, нажмите **File (Файл) → Open project (Открыть проект)**. Вы также можете открыть существующий проект в окне **Welcome (Приветствие)** (если ни один проект не открыт).

Откроется диалоговое окно, в котором можно будет перейти в директорию, содержащую проект, и выбрать соответствующий файл проекта.

Редактирование проекта

Чтобы изменить элемент проекта, выполните следующие действия.

Шаг	Действие
1	Найдите элемент в древовидной структуре окна инструмента.
2	Дважды щелкните по имени, чтобы открыть его. **Результат:** открывается редактор для объектов данного типа. Например, при двойном щелчке по имени программного модуля проекта открывается соответствующий редактор исходного кода; при двойном щелчке по имени глобальной переменной открывается редактор переменных.

EcoStruxure Machine Expert - HVAC не позволяет изменить элементы проекта, если выполняется хотя бы одно из следующих условий:

- EcoStruxure Machine Expert - HVAC находится в режиме отладки.
- Элемент является объектом подключенной библиотеки (при этом вы можете изменять объект, импортированный из библиотеки).
- Проект открыт в режиме только для чтения (просмотр проекта).

Закрытие проекта

Чтобы закрыть проект, нажмите **File (Файл) → Close project (Закрыть проект)** или закройте программу.

В обоих случаях при наличии несохраненных изменений EcoStruxure Machine Expert - HVAC предлагает пользователю выбрать между их сохранением и удалением.
После этого открывается окно **Welcome (Приветствие)** *(см. стр. 53)*, и вы можете запустить новый сеанс работы.

Распределение проектов

Обзор

Чтобы поделиться проектом с другим разработчиком, отправьте ему копию одного или нескольких файлов проекта или распространяемый модуль исходного кода (RSM), сгенерированный EcoStruxure Machine Expert - HVAC.

В первом случае количество пересылаемых файлов для совместного использования зависит от формата файла проекта:

- Одиночный файл проекта ПЛК (расширение файла `.ppjs`): сам файл проекта содержит всю информацию, необходимую для запуска приложения (предполагается, что получатель-разработчик имеет в своем распоряжении соответствующее целевое устройство). Он включает в себя модули с исходным кодом, так что для совместного использования проекта потребуется только файл `.ppjs`.
- Комплексный файл проекта ПЛК (расширение файла `.ppjx` или `.ppj`): файл проекта содержит только ссылки на модули исходного кода, составляющие проект, которые находятся в каталоге проекта в виде отдельных файлов. Для совместного использования проекта требуется весь каталог.
- Полный файл проекта ПЛК в формате XML (`.plcprj`): файл проекта полностью сгенерирован на языке XML. Информация, содержащаяся в файле проекта, и его поведение совпадают с файлом, имеющим расширение `.ppjs`.

Чтобы сгенерировать распространяемый модуль исходного кода (RSM), нажмите **Project (Проект) → Generate redistributable source module (Сгенерировать распространяемый модуль исходного кода)**.

EcoStruxure Machine Expert - HVAC отображает имя файла RSM и предлагает защитить файл паролем. Для защиты файла необходимо ввести пароль.

Преимущества формата RSM:

- Он закодирован в бинарном формате, поэтому может быть прочитан только лицами, использующими EcoStruxure Machine Expert - HVAC.
- При открытии файла в EcoStruxure Machine Expert - HVAC может потребоваться пароль.
- Минимальный размер за счет бинарного формата.

Экспорт в файлы CSV

Обзор

EcoStruxure Machine Expert - HVAC позволяет экспортировать параметры и переменные в файлы формата **.csv**, которые, в свою очередь, могут использоваться для обмена информацией и разработки документации, поставляемой вместе с продуктом.

Экспорт данных

Чтобы экспортировать данные, выполните следующие действия.

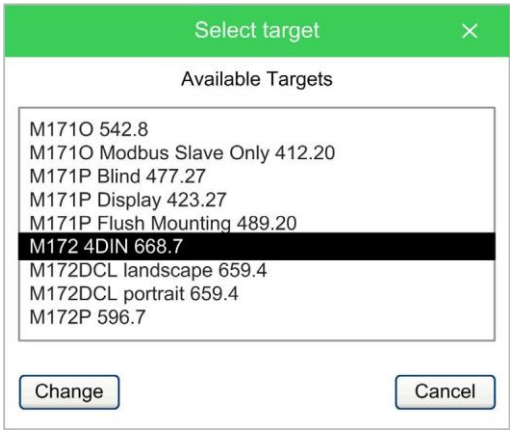
Шаг	Действие
1	Нажмите на вкладку **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**.
2	В окне **Resources (Ресурсы)** нажмите на целевое устройство.
3	В основном поле нажмите кнопку **Export (Экспорт)**.
4	В окне **Data Export (Экспорт данных)** выберите необходимые данные и нажмите кнопку **OK**.
5	В диалоговом окне **Save As (Сохранить как)** укажите имя вашего файла и нажмите кнопку **Save (Сохранить)**.
6	В диалоговом окне **Export succeeded (Экспорт выполнен успешно)** нажмите кнопку **OK**.

Выбор целевого устройства

Обзор

Возможно, вам потребуется адаптировать приложение ПЛК под новое целевое устройство, если оно отличается от того, для которого вы изначально писали код.

Чтобы адаптировать проект приложения под новое целевое устройство, выполните следующие действия.

Шаг	Действие
1	На вкладке **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** нажмите **Project (Проект)** → **Select target… (Выбрать целевое устройство…)**. Откроется следующее диалоговое окно: 
2	Выберите одно из целевых устройств, перечисленных в окне.
3	Нажмите **Change (Изменить)**, чтобы подтвердить свой выбор, или **Cancel (Отменить)**, чтобы отменить.
4	Если вы подтверждаете, нажмите кнопку **Yes (Да)**, чтобы завершить преобразование, или **No (Нет)**, чтобы отменить. Если вы нажмете кнопку **Yes (Да)**, то EcoStruxure Machine Expert - HVAC обновит проект для работы с новым целевым устройством. При этом также создается резервная копия файлов проекта в подкаталоге внутри каталога проекта. Поэтому вы можете откатить операцию вручную (то есть с помощью проводника Windows), заменив файлы проекта на резервную копию.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Всегда проверяйте, работает ли ваша прикладная программа так же, как и до конвертации. В ней должны применяться корректные значения конфигурации, параметры, значения параметров, функции и функциональные блоки.
- Измените приложение таким образом, чтобы оно работало, как и раньше.
- Тщательно протестируйте и проверьте новую скомпилированную версию, прежде чем вводить ваше приложение в эксплуатацию.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Сборка

Обзор

Чтобы скомпилировать проект, нажмите **Build All (Собрать все)** на панели инструментов проекта.
Результат компиляции отображается в окне **Output (Вывод)** *(см. стр. 146)*.

Загрузка проекта на целевое устройство

Обзор

В зависимости от контроллера проект можно загрузить различными способами.

Ниже приведена таблица соответствия контроллеров и возможных типов соединения.

Контроллер	TTL-порт		Порт USB A	Порт USB Mini-B	Порт RS485 Modbus	Ethernet
	Кабель для программирования TM171ADMI	Программирующий флеш-накопитель TM171AMFK	Накопитель USB A	Кабель USB A — USB Mini-B	Адаптер USB — RS485	Кабель Ethernet
M171O	✓	✓	–	–	–	–
M171P	–	–	✓	✓	✓	✓ (1)
M172O	–	–	–	✓	✓	✓ (1)
M172P	–	–	✓	✓	✓	✓
(1) К контроллеру должен быть подключен коммуникационный модуль Ethernet, за исключением контроллера M171P с креплением для скрытого монтажа, который имеет встроенный модуль связи Ethernet.						

Загрузка проекта контроллера на целевое устройство

Чтобы загрузить проект на целевое устройство, выполните следующие действия.

Шаг	Действие
1	Настройка связи с помощью **Device Link Manager Config**: ● перейдите в меню **On-line (Онлайн) → Set up communication… (Настройка связи…)** на вкладке **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** или **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**; ● или в меню **Target (Целевое устройство) → Communication settings (Настройки связи)** на вкладке **COMMISSIONING (ВВОД В ЭКСПЛУАТАЦИЮ)** . Свойства можно просмотреть и отредактировать, нажав на пункт **Properties (Свойства)**. Протокол должен быть активирован заранее. Для получения дополнительной информации см. раздел «Настройка связи» *(стр. 279)*.
2	Подключите целевое устройство физически к компьютеру. Дополнительную информацию о подключении оборудования см. в соответствующих руководствах по работе с оборудованием.
3	Подключите целевое устройство с помощью меню **On-line (Онлайн) → Connect (Подключить)** на вкладке **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** или **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**. Для получения дополнительной информации см. раздел «Онлайн-статус» *(стр. 287)*.
4	Загрузите проект, перейдя на вкладку **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** или **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**: ● выберите **On-line (Онлайн) → Download code (Загрузить код)**; ● или нажмите F5. Также можно использовать кнопку **Download all (Загрузить все)** на панели инструментов проекта.
5	Следуйте инструкциям в диалоговых окнах.

В случае отключения питания устройства, сбоя в подаче электроэнергии или обрыва связи во время переноса приложения устройство может перестать работать. В случае обрыва связи или отключения электричества повторите передачу. При отключении электричества или обрыве связи во время обновления микропрограммы, а также при использовании недействительной микропрограммы устройство перестанет работать. В этом случае используйте правильную микропрограмму и повторите попытку обновления.

⚠ ПРЕДУПРЕЖДЕНИЕ

АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК КОНТРОЛЛЕРА

- Прежде чем загружать приложение, сначала проверьте состояние машины или процесса.
- Перед загрузкой приложения убедитесь в отсутствии опасностей, способных привести к травмированию людей, находящихся внутри или рядом с машиной или в непосредственной близости от технологического процесса.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

ПРИМЕЧАНИЕ

НЕРАБОТОСПОСОБНОЕ ОБОРУДОВАНИЕ

- Не прерывайте передачу прикладной программы или процедуру обновления микропрограммы после запуска процедуры.
- Передайте файл заново, если процедура была прервана по каким-либо причинам.
- Не пытайтесь вводить устройство (логический контроллер, контроллер движения, контроллер ЧМИ или преобразователь частоты) в эксплуатацию до тех пор, пока передача файлов не будет успешно завершена.

Несоблюдение этих указаний может привести к повреждению оборудования.

ПО Installer

Описание

Автономное программное обеспечение **Installer** поставляется в комплекте с EcoStruxure Machine Expert - HVAC.

ПО **Installer** позволяет:

- управлять техническим обслуживанием систем путем загрузки проектов;
- управлять конфигурацией связанных контроллеров;
- конфигурировать устройства;
- изменять параметры BIOS.

ПО **Installer** предназначено для технического обслуживания. Оно не позволяет изменять код проекта. Для получения более подробной информации обратитесь к онлайн-справке ПО **Installer**.

ПРИМЕЧАНИЕ. Под привязкой в данном случае понимается связь и обмен переменными между логическими контроллерами.

Закрытие EcoStruxure Machine Expert - HVAC

Обзор

Для выхода из EcoStruxure Machine Expert - HVAC нажмите кнопку **Close (Закрыть)** в правом верхнем углу окна EcoStruxure Machine Expert - HVAC.

Вы также можете нажать кнопку **Exit (Выход)** в окне **Welcome (Приветствие).**

Часть 2
Конфигурирование

Содержание данной части

Эта часть содержит следующие главы.

Глава	Название главы	Страница
4	Вкладка CONFIGURATION (КОНФИГУРИРОВАНИЕ)	71
5	Управление элементами ресурсов	81
6	Справочник по технической реализации	133

Глава 4
Вкладка CONFIGURATION (КОНФИГУРИРОВАНИЕ)

Раздел 4.1
Обзор

Содержание этого раздела

В этом разделе описаны следующие темы.

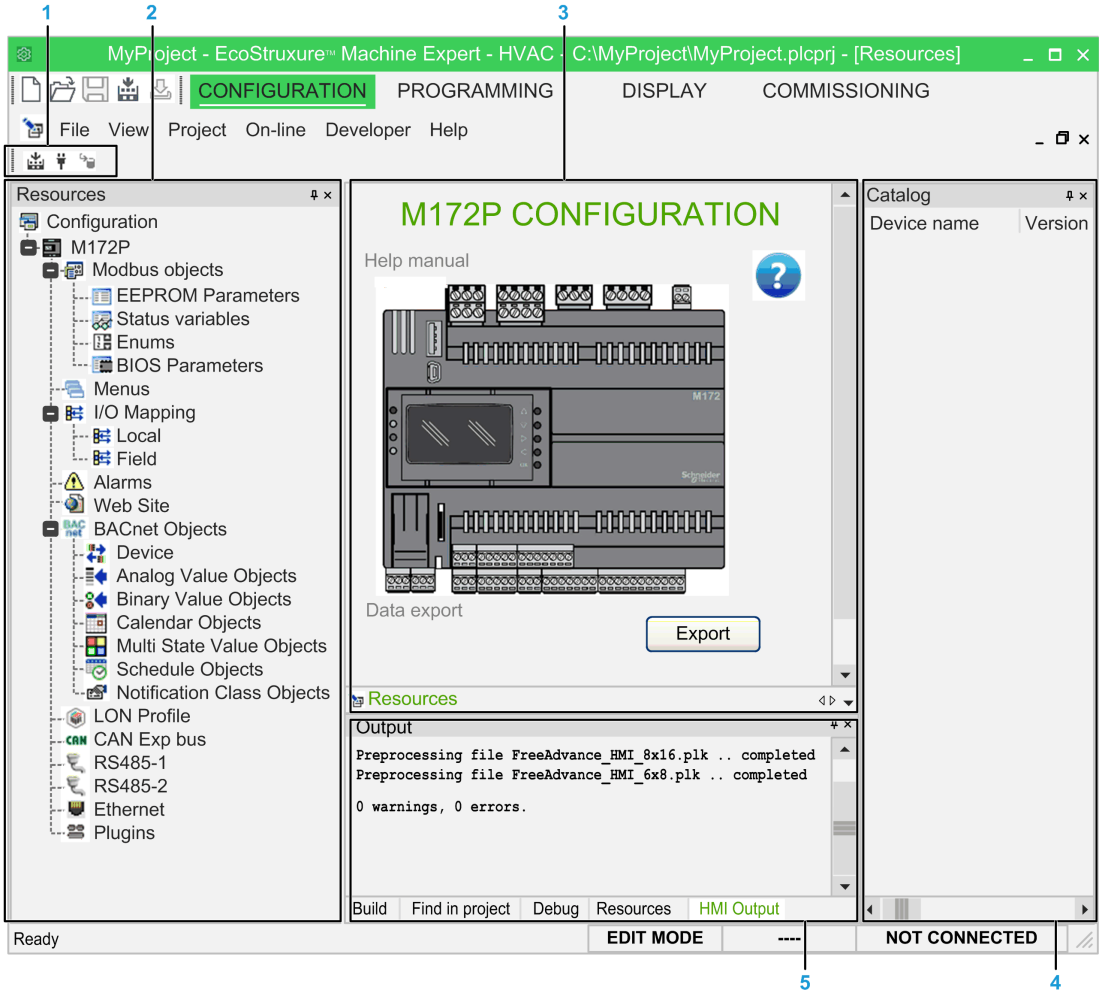
Тема	Страница
Обзор окна **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**	73
Панель меню	75
Панель инструментов	79
Строка состояния	80

Обзор окна CONFIGURATION (КОНФИГУРИРОВАНИЕ)

Общее описание

Окно **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** является точкой входа при разработке проектов.

На рисунке ниже показано окно **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** с настройками по умолчанию:



Позиция	Описание	
1	Панель инструментов	На этой панели инструменты показаны в виде значков. Для получения дополнительной информации см. раздел «Панели инструментов» *(стр. 79)*.
2	Окно **Resources (Ресурсы)**	В этом окне отображаются настраиваемые параметры устройства. Для получения дополнительной информации см. раздел «Содержимое окна Resources (Ресурсы)» *(стр. 83)*.
3	Окно редактора	Данное окно позволяет редактировать содержимое выделенного элемента в окне **Resources (Ресурсы)**.
4	Окно **Catalog (Каталог)**	В этом окне отображаются устройства, доступные в каталоге. **ПРИМЕЧАНИЕ.** Видимость устройств изменяется динамически в зависимости от выбора (например, модули связи).
5	Окно **Output (Вывод)**	В данном окне отображаются сообщения, связанные с разработкой проекта (открытие файла, ошибки чтения/записи, состояние подключения к устройству и т. д.). **ПРИМЕЧАНИЕ.** Подключение к устройству также отображается в строке состояния *(см. стр. 80)*. Для получения дополнительной информации см. раздел «Загрузка и выгрузка приложений» *(стр. 65)*.

Панель меню

Обзор

Панель меню состоит из нескольких меню:

- File (Файл) *(см. стр. 75)*
- View (Вид) *(см. стр. 76)*
- Project (Проект) *(см. стр. 77)*
- On-line (Онлайн)*(см. стр. 77)*
- Developer (Разработчик) *(см. стр. 77)*
- Help (Справка) *(см. стр. 78)*

Меню File (Файл)

Это меню предоставляет доступ к функциям управления проектом.

Команда	Значок	Клавиша	Описание
New project (Новый проект)		–	Создает новый проект *(см. стр. 53)*.
Open project (Открыть проект)		Ctrl+O	Открывает существующий проект *(см. стр. 59)*.
Save project (Сохранить проект)		Ctrl+S	Сохраняет текущий открытый проект *(см. стр. 57)*.
Save project as… (Сохранить проект как…)	–	–	Сохраняет текущий открытый проект, позволяя указать новое имя, местоположение и расширение *(см. стр. 57)*.
Close project (Закрыть проект)	–	–	Закрывает открытый проект *(см. стр. 59)*.
Options (Параметры)	–	–	Открывает диалоговое окно Program options (Параметры программы) *(см. стр. 47)*.
Print… (Печать…)		Ctrl+P	Распечатывает документ, открытый в текущем активном окне *(см. стр. 56)*.
Print preview (Предпросмотр перед печатью)		–	Создает предварительный просмотр документа, открытого в текущем активном окне, готовый к печати.
Printer setup… (Настройки принтера…)	–	–	Открывает диалоговое окно Printer setup (Настройки принтера).
..recent.. (недавние)	–	–	Выводит список недавно открытых файлов проектов.
Exit (Выход)	–	–	Закрывает EcoStruxure Machine Expert - HVAC.

Меню View (Вид)

Это меню предоставляет доступ к функциям для выбора элементов, которые должны отображаться в рабочем пространстве.

Команда	Значок	Клавиша	Описание
Панели инструментов	–	–	См. раздел «Панели инструментов» *(см. стр. 43)*.
Main (Главная)	–	–	Отображает или скрывает панель инструментов **Main (Главная)**.
Project (Проект)	–	–	Отображает или скрывает панель **Project (Проект)**.
Debug (Отладка)	–	–	Отображает или скрывает панель инструментов **Debug (Отладка)**.
FBD Bar (Панель FBD)	–	–	Отображает или скрывает панель инструментов **FBD**.
SFC Bar (Панель SFC)	–	–	Отображает или скрывает панель инструментов **SFC**.
LD Bar (Панель LD)	–	–	Отображает или скрывает панель инструментов **LD** .
Network (Сеть)	–	–	Отображает или скрывает панель инструментов **Network (Сеть)**.
Configuration (Конфигурирование)	–	–	Отображает или скрывает панель инструментов **Configuration (Конфигурирование)**.
HMI Page (Страница ЧМИ)	–	–	Отображает или скрывает панель инструментов **HMI Page (Страница ЧМИ)**.
HMI Project (Проект ЧМИ)	–	–	Отображает или скрывает панель инструментов **HMI Project (Проект ЧМИ)**.
HMI Profiles (Профили ЧМИ)	–	–	Отображает или скрывает панель инструментов **HMI Profiles (Профили ЧМИ)**.
Commissioning (Ввод в эксплуатацию)	–	–	Отображает или скрывает панель инструментов **Commissioning (Ввод в эксплуатацию)**.
Окна инструментов	–	–	См. раздел «Управление окнами инструментов» *(стр. 44)*.
Local variables (Локальные переменные)	–	–	Отображает или скрывает окно **Local variables (Локальные переменные)**.
Project (Проект)	–	–	Отображает или скрывает окно **Project (Проект)**.
Watch (Контрольные значения)	–	Ctrl+T	Отображает или скрывает окно **Watch (Контрольные значения)**.
Properties Window (Окно свойств)	–	Ctrl+W	Отображает или скрывает окно **Properties Window (Окно свойств)**.
Oscilloscope (Осциллограф)	–	–	Отображает или скрывает окно **Oscilloscope (Осциллограф)**.
PLC run-time status (Статус работы ПЛК)	–	–	Отображает или скрывает панель **PLC run-time status (Статус работы ПЛК)**.
Operators and blocks (Операторы и блоки)	–	–	Отображает или скрывает окно **Operators and blocks (Операторы и блоки)**.
Дерево библиотеки (Дерево библиотеки)	–	–	Отображает или скрывает окно **Library Tree (Дерево библиотеки)**.
Output (Выход)	–	–	Отображает или скрывает окно **Output (Вывод)**.
Cross Reference (Перекрестные ссылки)	–	–	Отображает или скрывает окно **Cross Reference (Перекрестные ссылки)**.
Resources (Ресурсы)	–	–	Отображает или скрывает окно **Resources (Ресурсы)**.
Catalog (Каталог)	–	–	Отображает или скрывает окно **Catalog (Каталог)**.
HMI Project (Проект ЧМИ)	–	–	Отображает или скрывает окно **HMI Project (Проект ЧМИ)**.
HMI Properties (Свойства ЧМИ)	–	–	Отображает или скрывает окно **HMI Properties (Свойства ЧМИ)**.
HMI Actions (Действия ЧМИ)	–	–	Отображает или скрывает окно **HMI Actions (Действия ЧМИ)**.
HMI Vars and Parameters (Переменные и параметры ЧМИ)	–	–	Отображает или скрывает окно **HMI Vars and Parameters (Переменные и параметры ЧМИ)**.
HMI Templates (Шаблоны ЧМИ)	–	–	Отображает или скрывает окно **HMI Templates (Шаблоны ЧМИ)**.
Commissioning (Ввод в эксплуатацию)	–	–	Отображает или скрывает окно **Commissioning (Ввод в эксплуатацию)**.

Команда	Значок	Клавиша	Описание
Commissioning Watch (Контрольные значения для ввода в эсплуатацию)	–	–	Отображает или скрывает окно **Commissioning Watch (Контрольные значения для ввода в эксплуатацию)**.
Commissioning Oscilloscope (Осциллограф для ввода в эксплуатацию)	–	–	Отображает или скрывает окно **Commissioning Oscilloscope (Осциллограф для ввода в эксплуатацию)**.
Full screen (Полный экран)		Ctrl+U	Разворачивает текущее активное окно документа на весь экран *(см. стр. 46)*. (Нажмите **Esc** для выхода из этого режима.)

Меню Project (Проект)

Данное меню предоставляет доступ к функциям, позволяющим компилировать проект и управлять целевым устройством.

Команда	Значок	Клавиша	Описание
Compile (Компилировать)		F7	Запускает компилятор *(см. стр. 64)*.
Select target… (Выбрать целевое устройство…)	–	–	Позволяет выбрать новое целевое устройство для проекта *(см. стр. 62)*.
Refresh current target (Обновить текущее целевое устройство)	–	–	Позволяет обновить целевой файл для той же версии целевого устройства.

Меню On-line (Онлайн)

Это меню предоставляет доступ к функциям обмена данными с целевым устройством.

Команда	Значок	Клавиша	Описание
Set up communication… (Настройка связи…)	–	–	Позволяет настроить параметры соединения с целевым устройством *(см. стр. 280)*.
Connect (Подключить)		–	Запускает обмен данными с устройством *(см. стр. 280)*.
Download code (Загрузить код)		F5	Загрузка конфигурации и приложения ПЛК для проекта на устройство *(см. стр. 289)*.

Меню Developer (Разработчик)

Это меню предоставляет доступ к функциям для совместного использования проекта с другими разработчиками.

Команда	Значок	Клавиша	Описание
Import EDS (Импорт EDS)	–	–	Импорт файла EDS (Electronic Data Sheet, электронное техническое описание) *(см. стр. 110)*.
Run Modbus Custom Editor (Запустить редактор Modbus Custom Editor)	–	–	Запускает редактор Modbus Custom Editor *(см. стр. 124)*.
Data export (Экспорт данных)	–	–	Экспорт данных в файл CSV *(см. стр. 61)*.

Меню Help (Справка)

Это меню предоставляет доступ к документации на оборудование или некоторым специфическим функциям EcoStruxure Machine Expert - HVAC.

Команда	Значок	Клавиша	Описание
Index (Указатель)	–	–	Выводит список **Help keywords (Ключевые слова справки)** и открывает соответствующий раздел.
Context (Контекст)	–	F1	Контекстно-зависимая справка. Открывает раздел, посвященный открытому в настоящее время окну.
Registration (Регистрация)	–	–	Регистрация ПО *(см. стр. 26)*.
About… (О программе…)	–	–	Сведения об авторах и версии ПО.

Панель инструментов

Введение

Панель инструментов расположена в верхней части окна EcoStruxure Machine Expert - HVAC и обеспечивает доступ к часто используемым функциям.

Общие сведения о панелях инструментов см. в разделе «Панели инструментов» *(стр. 43)*.

Панель инструментов Configuration (Конфигурирование)

Панель инструментов **Configuration (Конфигурирование)** содержит следующие кнопки.

Значок	Клавиша	Описание
	–	**Компиляция** Запускает компилятор.
	–	**Подключение к целевому устройству** Запускает обмен данными с устройством.
	–	**Загрузка кода** Загрузка конфигурации и приложения ПЛК проекта на устройство.

Строка состояния

Обзор

Строка состояния расположена в нижней правой части окна EcoStruxure Machine Expert - HVAC. Она служит для отображения состояния связи и статуса приложения, выполняемого в данный момент на целевом устройстве.

EDIT MODE	SOURCE OK	CONNECTED

Дополнительную информацию см. в следующих разделах:

- Режим редактирования и отладки *(см. стр. 321)*
- Статус соединения *(см. стр. 287)*
- Статус приложения *(см. стр. 288)*

Глава 5
Управление элементами ресурсов

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
5.1	Обзор	82
5.2	Целевое устройство	87
5.3	Объекты Modbus	88
5.4	Меню целевого устройства	92
5.5	Сопоставление входов/выходов	96
5.6	Сигналы тревоги	97
5.7	Веб-сайт	98
5.8	Шина расширения CAN	101
5.9	RS485	117
5.10	Ethernet	128
5.11	Подключаемые модули	130

Раздел 5.1
Обзор

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Окно Resources (Ресурсы)	83
Поддерживаемые протоколы	85

Окно Resources (Ресурсы)

Обзор

Окно **Resources (Ресурсы)** позволяет сконфигурировать устройство:

- Определить параметры и переменные.
- Создать и настроить встроенный веб-сайт (только для M171P и M172).
- Настроить аппаратную структуру проекта.
- Настроить протоколы передачи данных.

Содержимое окна Resources (Ресурсы)

Окно **Resources (Ресурсы)** содержит следующие элементы.

Элемент	Значок	Описание
Целевое устройство *(см. стр. 87)*		Выводит изображение целевого устройства и позволяет настроить некоторые параметры.
Объекты Modbus *(см. стр. 88)*		Позволяет настроить **параметры EEPROM** (энергонезависимой памяти) и переменные состояния, которые затем можно будет использовать в коде приложения. Позволяет настроить **параметры EEPROM** (энергонезависимой памяти) и переменные состояния, которые затем можно отобразить на целевом устройстве и считать с помощью протоколов Modbus (RTU или TCP) или CAN.
Меню *(см. стр. 92)*		Управляет меню, в которых можно группировать объекты Modbus, доступные на вкладке **COMMISSIONING (ВВОД В ЭКСПЛУАТАЦИЮ)**.
Сопоставление входов/выходов *(см. стр. 96)*		Позволяет определить связи между переменными и физическими входами/выходами целевого устройства.
Сигналы тревоги *(см. стр. 97)*		Позволяет определить переменные сигналов тревоги, статус которых должен обрабатываться разработчиком.
Веб-сайт *(см. стр. 98)*		Позволяет определить страницы веб-сайта для мониторинга устройства с помощью веб-браузера.
Объекты BACnet		Позволяет настроить объекты BACnet.
Профиль LON		Позволяет настроить протокол LonWorks.
Шина CAN *(см. стр. 101)*		Позволяет настроить шину расширения CAN.
RS485-1 *(см. стр. 117)*		Позволяет настроить первый порт RS485.
RS485-2 *(см. стр. 117)*		Позволяет настроить второй порт RS485.

Элемент	Значок	Описание
Ведущее устройство Modbus RTU *(см. стр. 117)*		Позволяет настроить порт RS485. Действительно только для ведущего/ведомого устройства Modbus M171O.
Ethernet *(см. стр. 130)*		Позволяет настроить порт Ethernet.
Подключаемые модули *(см. стр. 130)*		Позволяет настроить протоколы с помощью коммуникационных модулей.
Справка		Отображает справку по светодиодной индикации для разработчиков. Только для M171O.

ПРИМЕЧАНИЕ. Содержание окна **Resources (Ресурсы)** зависит от выбранного устройства.

Согласование конфигурации программного и аппаратного обеспечения

Встроенные в контроллер входы/выходы не зависят от тех входов/выходов, которые можно добавить посредством модуля расширения входов/выходов. Важно, чтобы логическая конфигурация входов/выходов внутри вашей программы соответствовала физической конфигурации входов/выходов вашей установки. Если вы добавляете или удаляете какие-либо физические входы/выходы, ведущие к шине расширения входов/выходов или из нее, вам необходимо обновить конфигурацию вашего приложения. Это также относится к любым устройствам полевой шины, которые могут присутствовать в вашей установке. В противном случае существует вероятность того, что шина расширения или полевая шина перестанут функционировать, в то время как встроенные входы/выходы, которые могут присутствовать в контроллере, продолжат работать.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ
Обновляйте конфигурацию программы после каждого добавления или удаления любых устройств полевой шины и входов/выходов модуля расширения на шине ввода/вывода вашего контролера.
Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Шина расширения

Ваше приложение должно контролировать состояние шины и состояния ошибок модулей на шине, а также принимать соответствующие этим состояниям меры с учетом поставленных задач.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Учитывайте при оценке риска вероятность нарушения связи между логическим контроллером и любыми модулями расширения входов/выходов.
- Отслеживайте состояние шины расширения входов/выходов с помощью соответствующих системных слов и принимайте соответствующие меры согласно выполненной оценке рисков.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Поддерживаемые протоколы

Обзор

Каждое устройство имеет следующие ресурсы, которые отображаются как узлы целевого устройства. Выберите **Mode (Режим)** и добавьте устройство из каталога.

Целевое устройство	Шина связи	Описание
M171P	Шина расширения CAN	Встроенная Для модуля расширения входов/выходов и выносного дисплея
	RS485	Встроенный Modbus RTU (ведущее/ведомое устройство)
	Ethernet	Дополнительно с коммуникационным модулем: Modbus TCP (сервер), BACnet IP (сервер), HTTP
	Подключаемые модули	Опциональные модули доступны по отдельности
M171P с креплением для скрытого монтажа	Шина расширения CAN	Встроенная Для модуля расширения входов/выходов
	RS485	Встроенный Modbus RTU (ведущее/ведомое устройство) или BACnet MS/TP (сервер)
	Ethernet	Встроенный Modbus TCP (клиент/сервер), BACnet IP (сервер), FTP, HTTP
M172O	Шина расширения CAN	Встроенная Для модуля расширения входов/выходов и выносного дисплея
	RS485-1	Встроенный Modbus RTU (только ведомое устройство) или BACnet MS/TP (сервер)
	RS485-2	Встроенный Modbus RTU (ведущее/ведомое устройство) или BACnet MS/TP (сервер)
	Ethernet	Дополнительно с коммуникационным модулем: Modbus TCP (клиент/сервер), BACnet IP (сервер), FTP, HTTP
	Подключаемые модули	Опциональные модули доступны по отдельности
M172P	Шина расширения CAN	Встроенная Для модуля расширения входов/выходов и выносного дисплея
	RS485-1	Встроенный Modbus RTU (только ведомое устройство) или BACnet MS/TP (сервер)
	RS485-2	Встроенный Modbus RTU (ведущее/ведомое устройство) или BACnet MS/TP (сервер)
	Ethernet	Встроенный Modbus TCP (клиент/сервер), BACnet IP (сервер), FTP, HTTP
	Подключаемые модули	Опциональные модули доступны по отдельности

ПРИМЕЧАНИЕ. В окне **Catalog (Каталог)** отображаются устройства, которые можно добавить (путем перетаскивания) к соответствующему протоколу.

ПРИМЕЧАНИЕ. По протоколу RS485 можно также подключать универсальные устройства Modbus.

Предоставление страниц ЧМИ

M172P поддерживает функцию удаленного ЧМИ, благодаря которой его страницы можно загружать и отображать на вкладке **DISPLAY (ОТОБРАЖЕНИЕ)** при подключении к дисплеям M171P.

Эта функция не поддерживается M171P в исполнении для скрытого монтажа. Подключенные устройства не могут загружать страницы ЧМИ с устройства M171P с креплением для скрытого монтажа.

Раздел 5.2
Целевое устройство

Целевое устройство

Обзор

В окне **Resources (Ресурсы)** дважды щелкните по заголовку проекта, чтобы отобразить окно редактора.
Окно редактора содержит изображение целевого устройства и позволяет получить доступ к некоторым настройкам.

Конфигурирование M171

В окне редактора вы можете:

- Задать значение параметра, отображаемого на главном дисплее в состоянии простоя, выбрав значение в окне **Fundamental state display (Отображение фундаментального состояния)**.
  ПРИМЕЧАНИЕ. Доступно только для M171O.
- Задать время выполнения проекта в миллисекундах (мс).
  По умолчанию это значение составляет 100 мс. Допустимый диапазон: 20–100 мс.

  ПРИМЕЧАНИЕ. Доступно только для M171O.
- Экспортировать параметры и переменные в формате **.csv**.
  Для получения дополнительной информации см. раздел «Экспорт в файлы CSV» *(стр. 61)* .
- Обратитесь к руководству по работе с оборудованием для данного устройства,

  нажав на значок .

Конфигурирование M172

В окне редактора вы можете:

- Экспортировать параметры и переменные в формате **.csv**.
  Для получения дополнительной информации см. раздел «Экспорт в файлы CSV» *(стр. 61)* .
- Обратитесь к руководству по работе с оборудованием для данного устройства,

  нажав на значок .

Раздел 5.3
Объекты Modbus

Объекты Modbus

Обзор

Объекты Modbus позволяют определять **параметры EEPROM** (энергонезависимой памяти) и **переменные состояния**, которые могут отображаться на целевом устройстве и считываться с помощью протокола Modbus *(см. стр. 134)* .

Параметры EEPROM (энергонезависимой памяти) и **переменные состояния** можно использовать в коде приложения. Они отображаются в окне **Project (Проект)**: **Project (Проект)** → **Aux Variables (Вспом. переменные)** → **Global Shared (Общие глобальные)**.

Добавить или удалить параметры и переменные, используя кнопки **Add (Добавить)** и **Remove (Удалить)**, можно так же, как и в случае с переменными в окне **Project (Проект)**.
Кнопка **Recalc (Пересчитать)** позволяет пересчитать адреса выбранных строк.

Параметры EEPROM

Пункт **EEPROM Parameters (Параметры EEPROM)** позволяет создавать переменные, которые разработчик намерен хранить в энергонезависимой памяти даже при выключенном устройстве.
Подробнее о столбцах окна редактора см. в разделе «Переменные состояния» *(стр. 90)*.

В этой таблице представлены столбцы окна редактора.

Столбец	Описание
IPA	Заранее присвоенный индекс.
Address (Адрес)	Адрес ресурса Modbus.
Name (Имя)	Имя ресурса, которое может быть использовано разработчиком в приложении контроллера.
Display label (Отображаемая метка)	Имя, отображаемое в меню приложения целевого устройства M171O (семисегментный дисплей на четыре знака).
Device type (Тип устройства)	Тип данных, отображаемых на целевом устройстве.
Application type (Тип приложения)	Тип данных, используемых в приложении контроллера.
Size (Размер)	Имеет значение только в случае типа STRING. Размер (длина) строки. По умолчанию = 31 символ (макс.).
Read only (Только для чтения)	Включает/выключает редактирование переменных состояния.
Default value (Значение по умолчанию)	Значение объекта по умолчанию.
Min (Мин.)	Минимальное значение объекта.
Max (Макс.)	Максимальное значение объекта.

Столбец	Описание
Scale (Масштаб)	Коэффициенты пересчета между **Device type (Тип устройства)** и **Application type (Тип приложения)**. Application Type (Тип приложения) = Scale (Масштаб) x Device Type (Тип устройства) + Offset (Смещение).
Offset (Смещение)	
Unit (Единица)	Единица измерения типа устройства, отображаемая на устройстве и, если доступна, в виде значка на целевом устройстве.
Format (Формат)	Формат отображения для значения по умолчанию/мин. значения/макс. значения. Например, отображение XXX.Y целого числа с десятичной точкой.
Access Level (Уровень доступа)	Этот столбец не применяется к M171P/M172. См. раздел «Видимость ресурсов меню» *(стр. 94)*.
Description (Описание)	Любой текст.
Note (Примечание)	

ПРИМЕЧАНИЕ. Столбцы можно скрыть или показать, щелкнув правой кнопкой мыши их название и выбрав команду **Hide column (Скрыть столбец)** или **Show columns (Показать столбцы)**.

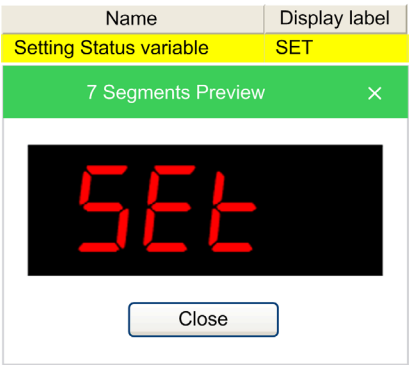
Энергонезависимая память рассчитана как минимум на 100 000 циклов записи. Использование энергонезависимой памяти для циклической операции записи может привести к быстрому исчерпанию ресурса и, как следствие, к выходу памяти из строя.

ПРИМЕЧАНИЕ
Не используйте регистры энергонезависимой памяти для циклических операций записи. **Несоблюдение этих указаний может привести к повреждению оборудования.**

Переменные состояния

Пункт **Status variables (Переменные состояния)** позволяет определить переменные состояния, которые могут отображаться в меню устройства.

ПРИМЕЧАНИЕ. В случае M171O каждая переменная перекодируется на контроллере ввиду использования семисегментного дисплея на четыре знака. В окне **Display label (Отображаемая метка)** вы можете выбрать перекодирование и заранее просмотреть, что будет выводиться на дисплей, нажав на многоточие (…).
Некоторые буквы не отображаются (например, x и z), поэтому на дисплее остается пустое место.
Если в качестве отображаемой метки выбрано **SET**, то на дисплей выводится SEt:



В этой таблице представлены столбцы окна редактора.

Столбец	Описание
IPA	Заранее присвоенный индекс.
Address (Адрес)	Адрес ресурса Modbus.
Name (Имя)	Имя ресурса, которое может быть использовано разработчиком в приложении контроллера.
Display label (Отображаемая метка)	Имя, отображаемое в меню приложения целевого устройства M171O (семисегментный дисплей на четыре знака).
Device type (Тип устройства)	Тип данных, отображаемых на целевом устройстве.
Application type (Тип приложения)	Тип данных, используемых в приложении контроллера.
Size (Размер)	Имеет значение только в случае типа STRING. Размер (длина) строки. По умолчанию = 31 символ (макс.).
Read only (Только для чтения)	Включает/выключает редактирование переменных состояния.
Default value (Значение по умолчанию)	Значение объекта по умолчанию.
Min (Мин.)	Минимальное значение объекта.
Max (Макс.)	Максимальное значение объекта.
Scale (Масштаб)	Коэффициенты пересчета между Device type (Тип устройства) и Application type (Тип приложения). Application Type (Тип приложения) = Scale (Масштаб) x Device Type (Тип устройства) + Offset (Смещение).
Offset (Смещение)	

Столбец	Описание
Unit (Единица)	Единица измерения типа устройства, отображаемая на устройстве и, если доступна, в виде значка на целевом устройстве.
Format (Формат)	Формат отображения для значения по умолчанию/мин. значения/макс. значения. Например, отображение XXX.Y целого числа с десятичной точкой.
Access Level (Уровень доступа)	Этот столбец не применяется к M171P/M172. См. раздел «Видимость ресурсов меню» *(стр. 94)*.
Description (Описание)	Любой текст.
Note (Примечание)	

ПРИМЕЧАНИЕ. Столбцы можно скрыть или показать, щелкнув правой кнопкой мыши их название и выбрав команду **Hide column (Скрыть столбец)** или **Show columns (Показать столбцы)**.

Перечисления

Пункт **Enums (Перечисления)** позволяет определить элементы перечисления, которые можно использовать в столбце **Device Type (Тип устройства)** окна редактора.

Для получения дополнительной информации о перечислениях см. раздел «Перечисления» *(стр. 207)*.

Параметры BIOS

Пункт **BIOS Parameters (Параметры BIOS)** позволяет изменять параметры BIOS, установленные по умолчанию на заводе Schneider Electric.

Раздел 5.4
Меню целевого устройства

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Меню целевого устройства M171O	93
Меню целевого устройства M171P/M172	95

Меню целевого устройства M171O

Обзор

Меню целевого устройства состоит из меню BIOS и меню приложений.

Меню BIOS настраивается на заводе.

В таблице ниже приведены основные функции клавиш/светодиодов целевого устройства.

Клавиша	Нажатие	Описание
F5	Короткое	Переключение из меню BIOS в меню приложений и обратно.
F1 или F3	Короткое	Перемещение по папкам и редактирование значений.
F2	Короткое	Отмена операции (функция ESC).
F4	Короткое	Доступ к меню Set (Установить).
F2+F4	Короткое	Доступ к меню Prg (Программа).
F1/F2/F3/F4	Длительное	Настраивается разработчиком (с помощью целевой переменной `sysKeyFunctions[]`).

Управление светодиодами осуществляется разработчиком с помощью целевых переменных `sysLocalLeds`. Элементы, введенные в таблицу в этом разделе, отображаются на устройстве.

Меню Prg (Программа)

Меню **Prg** (Программа) может состоять из одной или нескольких папок, определенных разработчиком, в которые вставляются:

- параметры EEPROM (энергонезависимой памяти);
- переменные состояния;
- параметры BIOS;
- входы и выходы.

Меню Set (Установить)

Меню **Set (Установить)** создается так же, как меню **Prg (Программа)**.

Меню **Set (Установить)** содержит папку **AL**.

Видимость ресурсов меню

Видимость ресурсов, созданных разработчиком, описывается в следующей таблице.

Уровень доступа	Видимость на устройстве	Видимость на целевом устройстве	Примечание
Видно всегда	Да	Да	Объект, назначенный для меню **Prg** или **Set**
Уровень 1	Да	Да Уровень 1	
Уровень 2	Да	Да Уровень 2	
Не видно никогда	Да	Нет	Объект, НЕ назначенный для меню **Prg** или **Set**
Не видно никогда	Да Видно в папке ALL PARAMETERS (ВСЕ ПАРАМЕТРЫ)	Нет	

Меню целевого устройства M171P/M172

Обзор

Меню целевого устройства можно создать с помощью вкладки **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**. В окне **Resources (Ресурсы)** щелкните правой кнопкой мыши пункт **Menus (Меню)** и выберите команду **Add Menu (Добавить меню)**.

Меню BIOS настроено на заводе и отображается на экране устройства.

Основные функции клавиш/светодиодов целевого устройства можно запрограммировать с помощью вкладки **DISPLAY (ОТОБРАЖЕНИЕ)**. Светодиоды также можно программировать из окна **Operators and blocks (Операторы и блоки)** на вкладке **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**.

В этом разделе можно определить меню (не отображаемое на дисплее) и папки/переменные, из которых оно будет состоять.

Данное меню может состоять из одной или нескольких папок, определенных разработчиком, в которые вставляются:

- параметры EEPROM (энергонезависимой памяти);
- переменные состояния.

Раздел 5.5
Сопоставление входов/выходов

Сопоставление входов/выходов

Сопоставление входов/выходов M171O

Вы можете определить связи между переменными и физическими входами/выходами M171O:

- **Local (Локальные)** — локальные переменные модуля контроллера.
- **Extended (Расширенные)** — переменные модуля расширения.
- **Remote (Удаленные)** — переменные на дисплеях.

Сопоставление входов/выходов M171P/M172

Вы можете определить связи между переменными и физическими входами/выходами M171P/M172:

- **Local (Локальные)** — локальные переменные базового модуля M171P/M172.
- **Field (Полевые)** — переменные модуля расширения входов/выходов.

Чтобы сопоставить переменные с физическими входами/выходами, введите имя переменной ПЛК в поле **I/O Mapping (Сопоставление входов/выходов)** → **Local (Локальные)** → столбец **Variable (Переменная)**.

ПРИМЕЧАНИЕ. Переменные, правильно определенные в разделе **Resources (Ресурсы)**, автоматически размещаются в окне **Project (Проект)**: **Project (Проект)** → **Aux Variables (Вспом. переменные)** → **Global Shared (Общие глобальные)**. При сохранении проекта не должно быть ошибок в переменных.

Раздел 5.6
Сигналы тревоги

Сигналы тревоги

Сигналы тревоги M171O

Вы также можете определить переменные сигналов тревоги, статус которых должен обрабатываться разработчиком.
Переменные сигналов тревоги можно использовать в коде приложения. Они отображаются в окне **Project (Проект)**: **Project (Проект)** → **Aux Variables (Вспом. переменные)** → **Global Shared (Общие глобальные).**

Если переменная принимает значение, отличное от нуля, метка отображается в папке **Alarms (Сигналы тревоги)** или **AL** в меню установки M171O.

В окне **Resources (Ресурсы)** щелкните **Alarm (Сигнал тревоги)**, чтобы отобразить список переменных сигналов тревоги. В этой таблице представлены столбцы окна редактора.

Столбец	Описание
Name (Имя)	Имя ресурса, которое может быть использовано разработчиком в приложении контроллера.
Short name (Краткое имя)	Имя, отображаемое в меню приложения целевого устройства M171 (четырехзначный дисплей). **ПРИМЕЧАНИЕ.** Если не заполнено, то на контроллере будут отображаться четыре первых символа переменной.
Описание	Описание переменной.

ПРИМЕЧАНИЕ. Каждая переменная перекодируется на контроллере ввиду использования семисегментного дисплея на четыре знака. В окне **Short name (Краткое имя)** вы можете заранее просмотреть, что будет выводиться на дисплей, нажав на многоточие (…).
Некоторые буквы не отображаются (например, x и z), поэтому на дисплее остается пустое место.
Если введен текст **SET**, то на дисплей выводится SEt.

Сигналы тревоги M171P/M172

На целевом устройстве M171P/M172 используется только глобальное объявление типа USINT.
Сигналы тревоги для M171P/M172 определены только в целях обеспечения переносимости проекта M171O.

Раздел 5.7
Веб-сайт

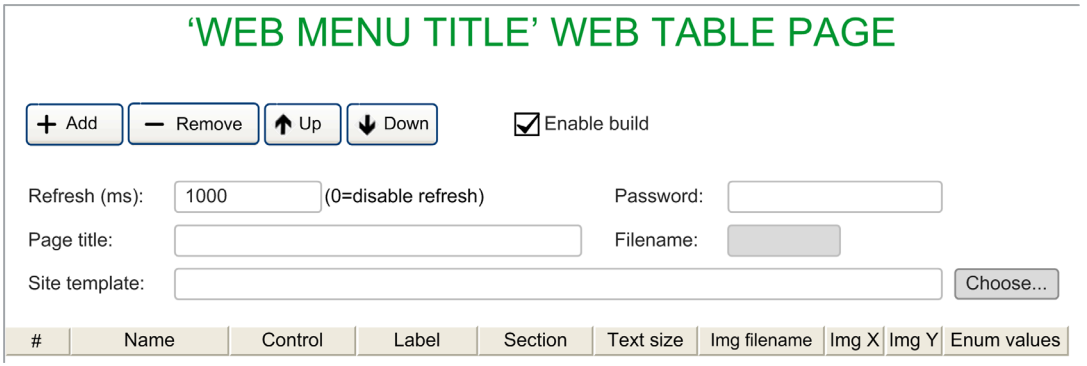
Веб-сайт

Веб-функции

Устройство M172 поддерживает доступ в Интернет, что позволяет изготовителям оборудования и системным интеграторам использовать удаленный доступ. Доступ к машинному оборудованию через Интернет позволяет уменьшить расходы на поддержку и техническое обслуживание за счет снижения затрат на вызовы специалистов. Конечные пользователи также получают пользу от такой возможности, поскольку могут контролировать свои системы с помощью графического интерфейса в любом браузере как локально, так и дистанционно.

Основные веб-функции:

- доступ через Интернет;
- дистанционное считывание показаний;
- локальное и дистанционное управление системой, включая управление сигналами тревоги;
- профилактическое и прогнозное обслуживание;
- сообщения о сигналах тревоги по электронной почте.



При использовании этого продукта в качестве устройства управления следует соблюдать осторожность и принимать соответствующие меры по устранению непреднамеренных последствий управляемой эксплуатации машины, изменения состояния контроллера или изменения данных памяти или эксплуатационных параметров машины.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Установите и настройте механизм, позволяющий использовать удаленный ЧМИ непосредственно на оборудовании для локального управления, независимо от команд дистанционного управления, отправляемых в приложение.
- Необходимо хорошо понимать принцип работы приложения и машинного оборудования, прежде чем пытаться управлять приложением дистанционно.
- Примите меры предосторожности, необходимые для обеспечения удаленной работы на соответствующей машине. Для этого требуется иметь однозначную идентифицирующую документацию внутри приложения и при удаленном подключении.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

ПРИМЕЧАНИЕ. Компания Schneider Electric придерживается лучших отраслевых практик при разработке и внедрении систем управления. Это предполагает реализацию концепции всеобъемлющей многоуровневой защиты в вопросах безопасности систем управления производством. В рамках этого подхода контроллеры размещаются за одним или несколькими брандмауэрами с целью предоставления доступа только авторизованному персоналу с использованием разрешенных протоколов.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕСАНКЦИОНИРОВАННЫЙ ДОСТУП И ПОСЛЕДУЮЩА Я НЕСАНКЦИОНИРОВАННАЯ ЭКСПЛУАТАЦИЯ ОБОРУДОВАНИЯ

- Оцените, подключена ли ваша среда или ваше оборудование к критически важной инфраструктуре, и если это так, перед подключением системы автоматизации к любой сети примите все необходимые меры безопасности, основанные на концепции всеобъемлющей многоуровневой защиты.
- Ограничьте количество устройств, подключенных к сети, до необходимого минимума.
- Изолируйте вашу промышленную сеть от других сетей внутри вашей компании.
- Защитите свои сети от несанкционированного доступа с помощью брандмауэров, VPN или других проверенных мер безопасности.
- Контролируйте действия в своих системах.
- Исключите возможность прямого доступа или несанкционированного подключения к защищаемым устройствам и выполнения неавторизованных операций.
- Подготовьте план восстановления, включая резервное копирование вашей системы и информации о процессах.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Создайте надежный пароль для использования веб-функций и не позволяйте неавторизованному или неквалифицированному персоналу использовать их.
- Убедитесь, что при удаленной работе с контроллером на месте присутствует компетентный и квалифицированный наблюдатель.
- Установите и настройте механизм, позволяющий использовать удаленный ЧМИ непосредственно на оборудовании для локального управления, независимо от команд дистанционного управления, отправляемых в приложение.
- Необходимо хорошо понимать принцип работы приложения и машинного оборудования, прежде чем пытаться управлять приложением дистанционно.
- Примите необходимые меры предосторожности, позволяющие убедиться при удаленной работе, что вы управляете предполагаемой машиной. Для этого требуется иметь однозначную идентифицирующую документацию внутри приложения и при удаленном подключении.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Раздел 5.8
Шина расширения CAN

Содержание этого раздела

В этом разделе описаны следующие темы.

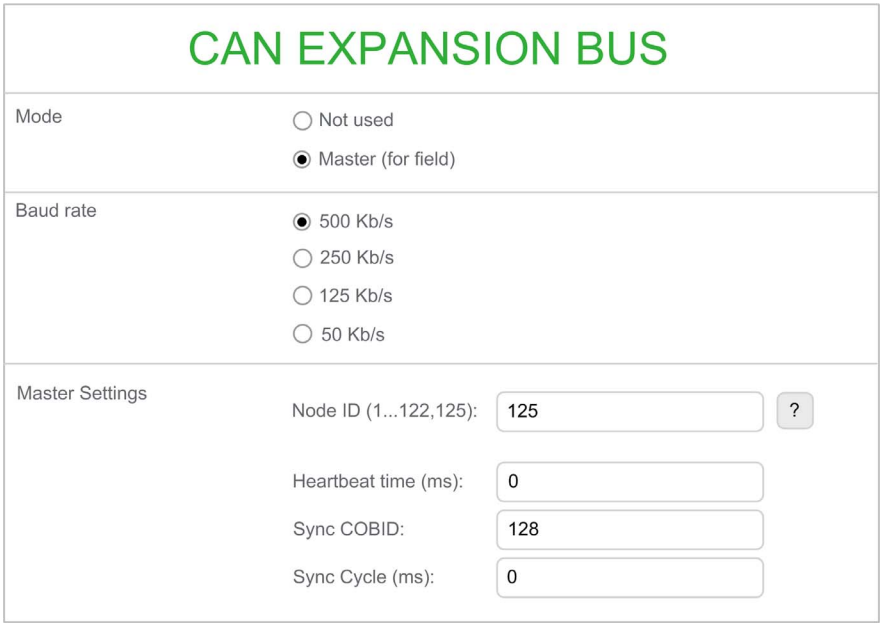
Тема	Страница
Шина расширения CAN	102
Использование модуля расширения в качестве ведомого устройства шины расширения CAN	104
Шина расширения CAN для M171P с креплением для скрытого монтажа	109
Пользовательское устройство CAN	110
Использование пользовательского устройства CAN	112
Шина расширения CAN (полевая) — виртуальные каналы ведущих устройств	116

Шина расширения CAN

Описание

M171P/M172 имеет один встроенный порт шины расширения CAN, еще один доступен в качестве внешнего подключаемого модуля. Каждый порт может быть сконфигурирован как **Not used (Не используется)**, тогда он будет отключен, или **Master (for field) (Ведущее устройство [для полевого режима])**.

Модуль M171P с креплением для скрытого монтажа можно подключить с помощью шины расширения CAN в полевом или сетевом режиме.



Ведущее устройство (для полевого режима)

При конфигурировании порта шины расширения CAN как **Master (for field) (Ведущее устройство [для полевого режима])** контроллер выступает в роли ведущего устройства шины расширения CAN для данного порта. Вы можете подключить ведомые устройства шины расширения CAN и обмениваться данными с удаленными входами/выходами.
Для порта ведущего устройства шины расширения CAN необходимо настроить:

- скорость передачи данных, используемую в этой сети шины расширения CAN (в кбит/с);
- ID узла для ведущего устройства (1–127), по умолчанию равен 125;
- интервал отправки контрольных сигналов, по умолчанию 0 (генератор контрольных сигналов отключен);
- используемый идентификатор коммуникационного объекта SYNC, по умолчанию 128;
- период цикла `SYNC` в мс, по умолчанию 0 (синхронизация отключена).

После добавления и настройки различных ведомых устройств шины расширения CAN вы можете связать находящиеся на расстоянии объекты ведомого устройства и внутренние переменные ПЛК для считывания или записи.

Набор объектов контроллера, доступных для чтения или записи:
- Переменные состояния, создаваемые на вкладке **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**.
- Полевые переменные, создаваемые на вкладке **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**.

Ведомое устройство (для привязки)

Режим привязки можно настроить с помощью программного обеспечения **Installer** *(см. стр. 67)* .

При конфигурировании порта шины расширения шины CAN как **Slave (Ведомое устройство)** порт шины M171P/M172P функционирует в качестве ведомого устройства шины расширения CAN. Используя привязку входов/выходов, вы можете обмениваться данными с другими устройствами в сети шины расширения CAN.

Настройка порта

Для порта ведомого устройства шины расширения CAN необходимо настроить:
- скорость передачи данных, используемую в этой сети шины расширения CAN (в кбит/с);
- ID узла для ведомого устройства (1–127) по умолчанию равен 1;
- «виртуальную сеть», к которой подключен дисплей M171P; в дереве отображается маленький цветной кружок, цвет которого совпадает с цветом выбранной сети (цвет соответствует определенной сети);
- можно привязать не более 10 устройств.

Объект **Binding (Привязка)**

После настройки порта шины расширения CAN как **Slave (Ведомое устройство)** устройство может отправлять (SEND) объекты по сети. Чтобы устройство могло читать (READ) объекты от других устройств, к порту необходимо добавить объект **Binding (Привязка)**.
Набор объектов ПЛК, доступных для отправки или получения:
- **Параметры EEPROM** (энергонезависимой памяти), создаваемые на вкладке **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**.
- **Переменные состояния**, создаваемые на вкладке **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**.

Если щелкнуть объект **Binding (Привязка)**, откроется его страница конфигурирования: сетка, в которую можно вставить удаленные объекты для чтения, а затем связать их с локальными местами назначения.

Для этого нажмите кнопку **Add (Добавить)**. Откроется окно, в котором отображаются «публичные» объекты других устройств в данной сети. Здесь вы можете применять фильтры поиска и выбирать, какие объекты считывать (множественный выбор также поддерживается).

После того как вы вставили удаленные объекты для чтения, необходимо назначить локальные места назначения для записи, выбрав их из списка в столбце **Dest parameter (Параметры назначения)** или вставив вручную **Address (Адрес)**.

ПРИМЕЧАНИЕ. Для обновления списка публичных объектов необходимо пересобрать проект ПЛК.
Пример:
- **M171 Perf Display_1** считывает с **M171 Perf Display_2** объект `M171_2_par1` и вставляет его в свой локальный объект `M171_1_par1`.
- **M171 Perf Display_1** считывает с **M171 Perf Display_2** объект `M171_2_par2` и вставляет его в свой локальный объект `M171_1_par2`.

В поле Period (Период) для каждого параметра можно настроить интервал обновления объекта в миллисекундах.

Использование модуля расширения в качестве ведомого устройства шины расширения CAN

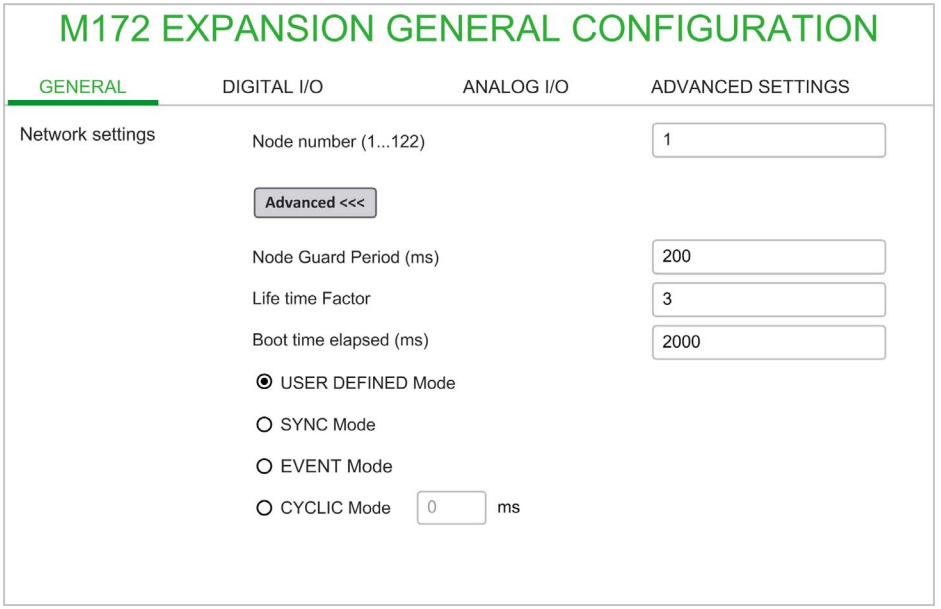
Шина расширения CAN с модулями расширения M172 и контроллерами M172

В этом примере конфигурации модуль расширения M172 на 28 входов/выходов используется для расширения возможностей устройства M172P. То же самое можно сделать и для остальных логических контроллеров.

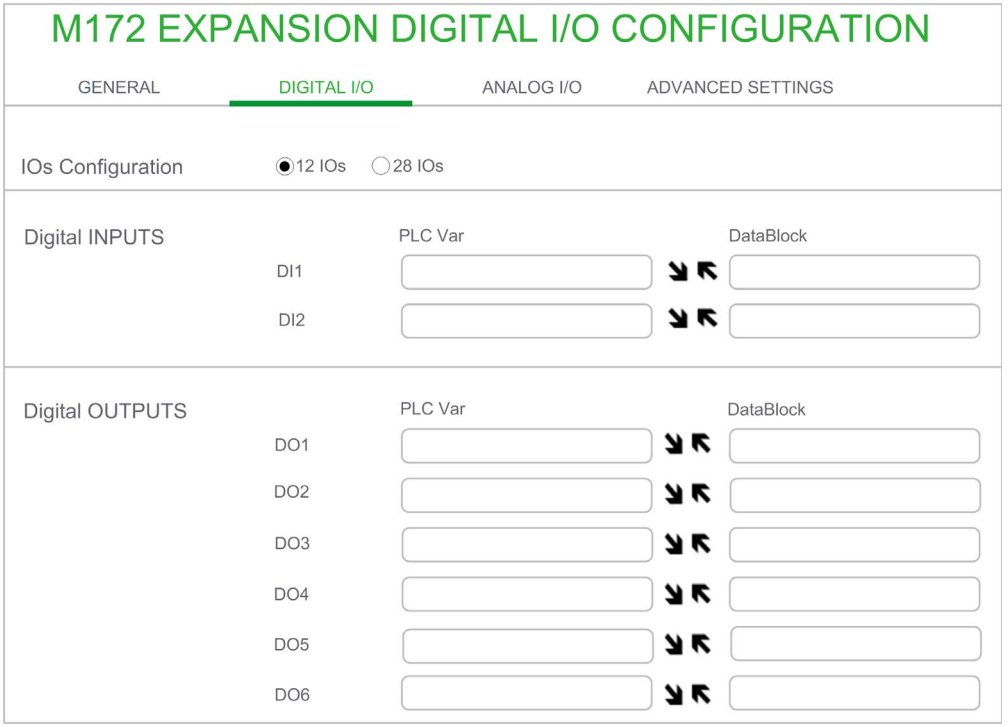
Настройте шину расширения CAN M172P в режиме **Master (for field) (Ведущее устройство [для полевого режима])**. Находясь в окне **Catalog (Каталог)**, выберите узел **модуля расширения M172 на 12 или 28 входов/выходов** и перетащите его на узел **шины расширения CAN**.
Окно конфигурирования **модуля расширения M172 на 12 или 28 входов/выходов** содержит четыре вкладки:

- **GENERAL (ОБЩЕЕ)**: для настройки параметров сети.
- **DIGITAL I/O (ЦИФРОВЫЕ ВХОДЫ/ВЫХОДЫ)**: для настройки цифровых входов/выходов.
- **ANALOG I/O (АНАЛОГОВЫЕ ВХОДЫ/ВЫХОДЫ)**: для настройки аналоговых входов/выходов.
- **ADVANCED SETTINGS (ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ)**: для добавления и удаления переменных.

Вкладка **GENERAL (ОБЩЕЕ) модуля расширения M172 на 12 или 28 входов/выходов:**

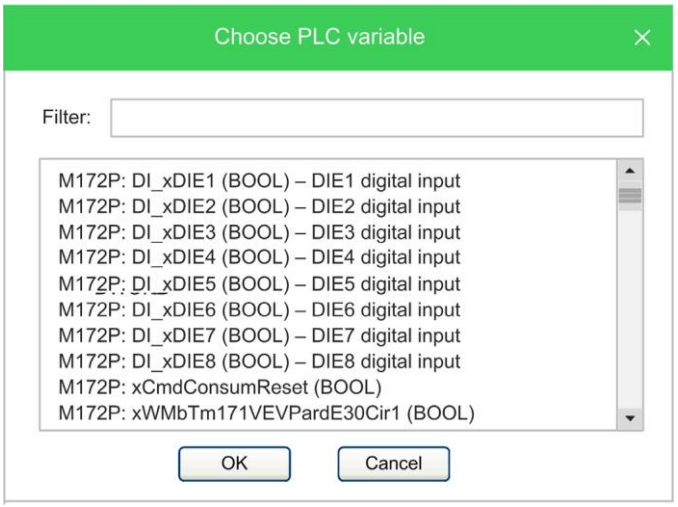


Вкладка **DIGITAL I/O (ЦИФРОВЫЕ ВХОДЫ/ВЫХОДЫ)** модуля расширения M172 на 12 или 28 входов/выходов:



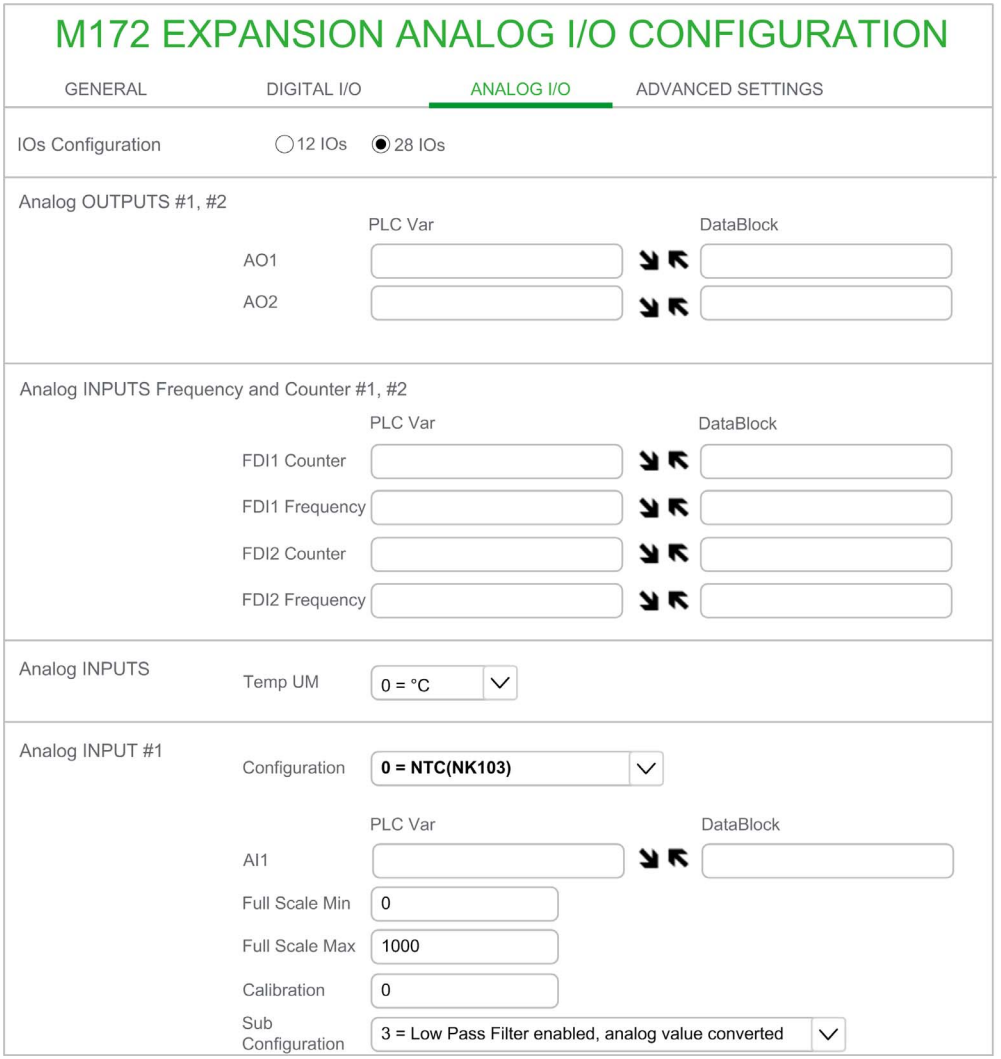
EcoStruxure Machine Expert - HVAC знает словарь **модуля расширения M172 на 12 или 28 входов/выходов**. Каждый объект можно присвоить **переменной ПЛК**.

Чтобы присвоить объект **переменной ПЛК**, выполните следующие действия.

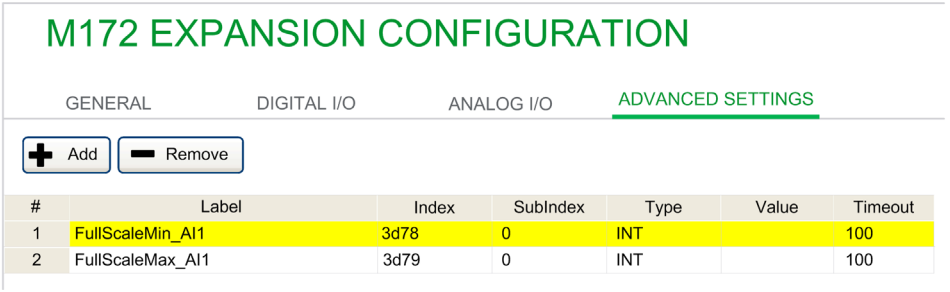
Шаг	Действие
1	Нажмите кнопку **Assign (Присвоить).**
2	Выберите **переменную ПЛК**, которую требуется назначить объекту ПЛК:  **ПРИМЕЧАНИЕ.** Можно применить фильтр к списку выбора, введя последовательность соответствующих символов.
3	Нажмите кнопку **OK** для подтверждения.
4	Переменная ПЛК с указанным именем добавляется, и ее адрес отображается в поле **DataBlock**.

ПРИМЕЧАНИЕ. Нажмите кнопку **Unassign (Отменить присвоение)**, чтобы отменить присвоение.

Вкладка **ANALOG I/O (АНАЛОГОВЫЕ ВХОДЫ/ВЫХОДЫ) модуля расширения M172 на 12 или 28 входов/выходов:**



Вкладка **ADVANCED SETTINGS (ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ) модуля расширения M172 на 12 или 28 входов/выходов:**



#	Label	Index	SubIndex	Type	Value	Timeout
1	FullScaleMin_AI1	3d78	0	INT		100
2	FullScaleMax_AI1	3d79	0	INT		100

Шина расширения CAN с модулем расширения M171P

В этом примере конфигурации модуль TM171EP27R используется для расширения возможностей устройства M171P. То же самое можно сделать и для остальных логических контроллеров.

Настройте шину расширения CAN M171P в режиме **Master (for field) (Ведущее устройство [для полевого режима])**. Находясь в окне **Catalog (Каталог)**, можно выбрать узел **модуля расширения M171P на 27 входов/выходов** и перетащить его на узел шины расширения CAN.

Процедура конфигурирования **модуля расширения M171P на 27 входов/выходов** аналогична пользовательской конфигурации CAN (с помощью пользовательского устройства CAN, (*см. стр. 112*) с отключенной функцией динамического сопоставления PDO. Доступные объекты входов/выходов, которые могут быть сопоставлены с переменными ПЛК через PDO, перечислены на вкладках **PDO TX — INPUT (PDO TX — ВХОДЫ)** и **PDO RX — OUTPUT (PDO RX — ВЫХОДЫ)**.

Шина расширения CAN для M171P с креплением для скрытого монтажа

Описание

Модуль M171P с креплением для скрытого монтажа можно подключить с помощью шины расширения CAN в полевом или сетевом режиме.

Полевой режим

Чтобы подключить в этом режиме модуль M171P с креплением для скрытого монтажа, выберите дисплей M171P или узел модуля расширения CAN **M171 Perf. Blind**, а затем опцию **Master (for field) (Ведущее устройство [для полевого режима])**, после этого перетащите устройство M171P с креплением для скрытого монтажа из вкладки **Catalog (Каталог)** на узел шины расширения CAN. Выберите дочерний узел **M171 Perf. Flush Mounting_1** и настройте параметры **Network (Сеть)**.

Датчики

M171 Perf. Display_1 может иметь доступ к встроенным датчикам **M171 Perf. Flush Mounting_1**. Для этого выберите вкладку **Probes — Input (Датчики — вход)**, после чего вы сможете сопоставить параметр **M171 Perf. Display_1** для получения значения встроенного датчика **дисплея M171P**.

Выберите один из датчиков и нажмите кнопку **Assign (Присвоить)**. Выберите один из параметров INT узла **M171 Perf. Display_1** и нажмите кнопку **OK**.

ЧМИ

С модулем M171P с креплением для скрытого монтажа (сконфигурированным в качестве ведомого устройства шины расширения CAN) можно связать проект ЧМИ с локальными страницами. Устройство M171P с креплением для скрытого монтажа сможет отобразить собственные целевые переменные и параметры ведущего устройства шины расширения CAN, к которому они относятся.

Сетевой режим

Режим удаленного управления и привязки терминала ЧМИ можно настроить с помощью программного обеспечения **Installer** *(см. стр. 67)* .

В этом режиме подключения M171P с креплением для скрытого монтажа можно подключить к одному из удаленных устройств, доступных в сети, для дистанционной навигации по страницам удаленного ЧМИ, предоставляемым другими устройствами.

Это можно сделать с помощью программы **Installer**, указав одно из доступных устройств в сети с поддержкой удаленного ЧМИ.

Например

Сеть шины расширения CAN уже содержит **M171 Perf. Display_1** и **M171 Perf. Blind_1**. Добавьте в сеть узел первого уровня **M171 Perf. Flush Mounting_1** с панели **Catalog (Каталог)**.

Нажмите на узел **CAN Exp bus**, относящийся к **M171 Perf. Flush Mounting_1**, и выберите узел **Master (Ведущее устройство)** (для удаленного доступа ЧМИ и привязки), присвойте уникальный **Node ID (ID узла)** и выберите сеть **CAN Exp bus1**.

Привязка переменных между **M171 Perf. Flush Mounting_1**, **M171 Perf. Blind_1** и **M171 Perf. Display_1** в сети этого типа разрешена. Более подробно — в главе «Шина расширения CAN — привязка» *(см. стр. 103)* .

Страницы удаленного ЧМИ

В сетевом режиме шины расширения CAN можно настроить M171P с креплением для скрытого монтажа для подключения к 0–10 удаленным устройствам, которые позволяют управлять страницами удаленного ЧМИ с клавиатуры.

Для добавления страниц удаленного ЧМИ выберите узел **M171 Perf. Flush Mounting_1**, затем нажмите кнопку **Add (Добавить)** в окне **HMI Remote pages (Страницы удаленного ЧМИ)**, после чего на экране отобразятся все доступные устройства, и пользователь сможет выбрать страницы для навигации.

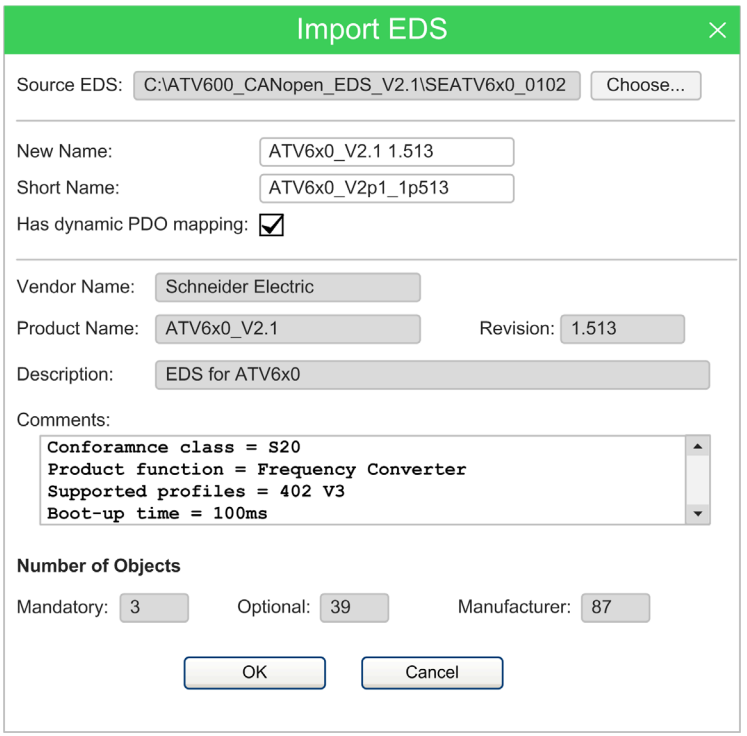
Пользовательское устройство CAN

Описание

Пользовательское устройство CAN можно создать и добавить в каталог путем импорта файла **EDS**. Поэтому в качестве ведомого можно использовать любое устройство шины расширения CAN стороннего производителя, если оно предоставляет стандартный файл **EDS**, соответствующий спецификации DS402 CiA.

Импорт нового пользовательского устройства CAN

Чтобы импортировать новое **пользовательское устройство CAN**, выберите команду **Developer (Разработчик) → Import EDS (Импорт EDS)**. Откроется окно **Import EDS (Импорт EDS)**:



Здесь необходимо выполнить следующие настройки:

- Выберите исходный файл **EDS** для импорта с помощью кнопки **Choose… (Выбрать…)**.
- Укажите полное имя устройства (по умолчанию **имя изделия** + **ревизия**).
- Короткое имя не должно содержать никаких специальных символов или пробелов.
- Динамическое сопоставление PDO: если активировать эту опцию, вы сможете вручную настраивать и изменять стандартное сопоставление PDO, считанное из EDS, приводя его в соответствие с актуальным сопоставлением ведомого устройства, в противном случае сопоставление PDO будет доступно только для чтения и будет определяться только стандартными значениями EDS.

После выбора файла **EDS** в окне отобразится обзор характеристик устройства и количество объектов, подробно описанных в полях Mandatory (Обязательные), Optional (Опциональные) и Manufacturer (Производитель).

Удаление пользовательского устройства CAN

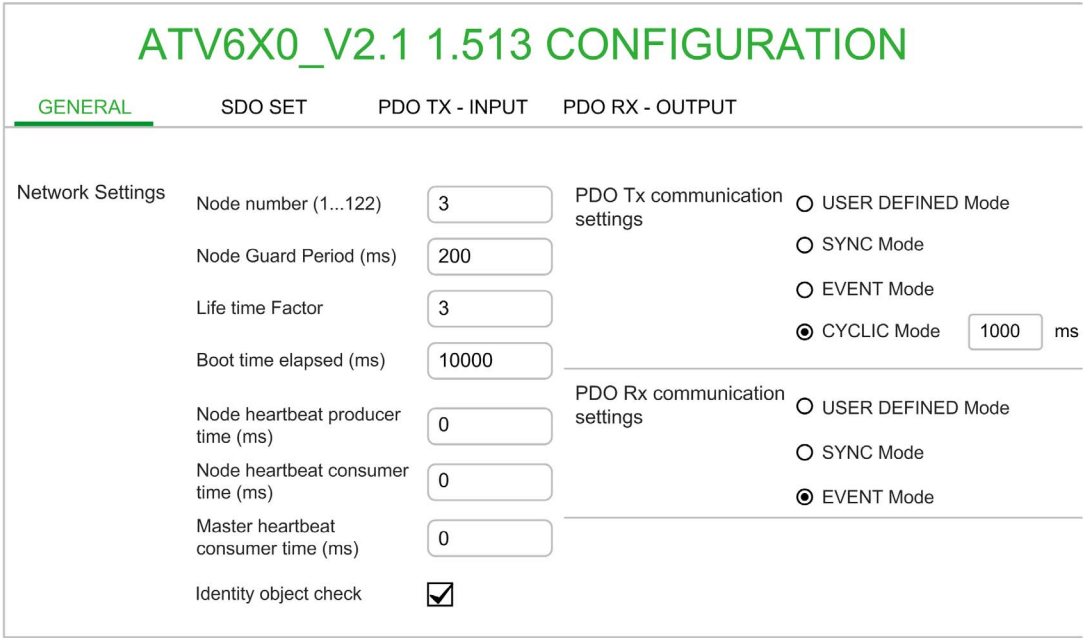
Если устройство, которое вы хотите удалить, отображается в окне **Catalog (Каталог)** (например, когда выбран порт шины расширения CAN и оно находится в режиме **ведущего устройства**), вы можете щелкнуть по нему правой кнопкой мыши и выбрать команду **Delete from catalog (Удалить из каталога)**.

Использование пользовательского устройства CAN

Описание

Если вставить **пользовательское устройство CAN** в качестве ведомого устройства шины расширения CAN (например, под портом шины расширения CAN ведомого устройства) и щелкнуть по нему в окне **Resources (Ресурсы)**, то в окне редактора отобразятся четыре вкладки.

Вкладка General (Общее)

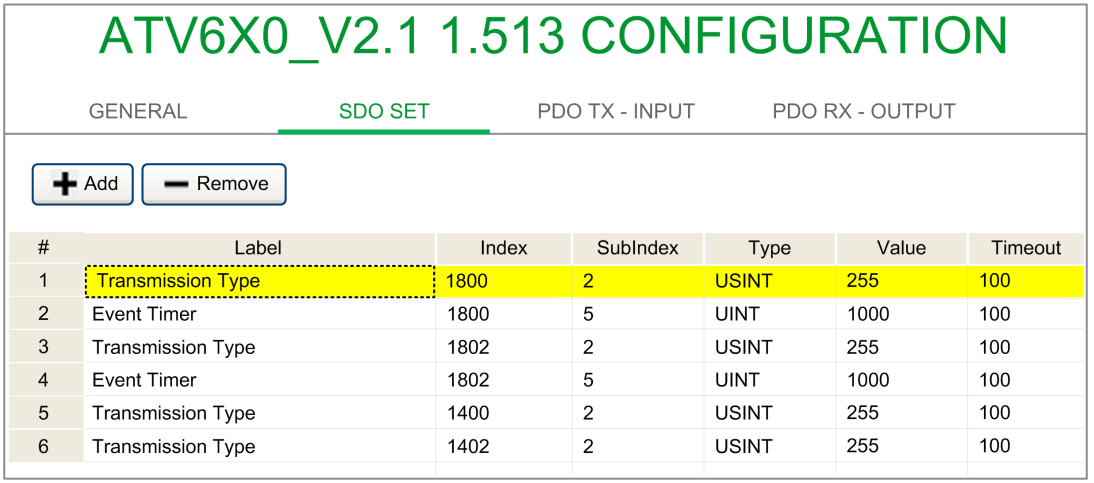


Во вкладке **General (Общее)** доступны следующие настройки:

- **Node number (Номер узла)** (1–122).
- **Node guard period (Период защиты узла),** в мс (по умолчанию 200 мс). Значение 0 отключает защиту для этого ведомого узла. Если значение не равно нулю, то оно соответствует интервалу пакетов защиты узла, посылаемых ведущим устройством ведомому устройству.
- **Life time factor (Коэффициент времени жизни)** (по умолчанию 3x). Значение 0 отключает защиту для этого ведомого узла. Если значение не равно нулю и умножается на **Node guarding period (Период защиты узла)**, то оно представляет собой максимальное время, в течение которого ведущее устройство ожидает ответа ведомого для защиты узла.
- **Boot time elapsed (Истекшее время загрузки)** – это максимальное время в мс, в течение которого ведущее устройство ожидает предпусковой готовности ведомого устройства в процессе загрузки (по умолчанию 10 с), прежде чем будет подан сигнал об ошибке.
- **Node heartbeat producer time (Время работы генератора контрольных сигналов для узла),** в мс, по умолчанию значение равно 0 (контрольные сигналы отключены). Если значение не равно нулю, то ведущее устройство позволяет обрабатывать ошибки контрольных сигналов для этого узла.

- **Node heartbeat consumer time (Время работы получателя контрольных сигналов для узла),** в мс, по умолчанию значение равно 0 (контрольные сигналы отключены). Если значение не равно нулю, то оно соответствует максимальному промежутку времени, в течение которого ведомое устройство ждет контрольного сигнала, генерируемого ведущим устройством, до истечения тайм-аута. Это значение должно быть больше, чем **Heartbeat time (Интервал отправки контрольных сигналов)** ведущего устройства.
- **Master heartbeat consumer time (Время работы получателя контрольных сигналов для ведущего устройства),** в мс, по умолчанию значение равно 0 (контрольные сигналы отключены); это максимальное время, в течение которого ведомое устройство ждет контрольного сигнала, генерируемого ведущим устройством, до истечения тайм-аута. Это значение должно быть больше, чем значение **Node heartbeat producer time (Время работы генератора контрольных сигналов для узла)**.
- **Identity object check (Проверка объекта идентификации)**: если эта опция включена (по умолчанию), ведущее устройство при загрузке проверяет ведомое устройство на его идентичность, сверяя поля объекта Identity (1018 HEX) со значениями EDS по умолчанию (ID поставщика, код изделия, ревизия, серийный номер). Если опция не включена, проверка не производится.
- **PDO Tx comm settings (Настройки связи PDO Tx)**: здесь можно настроить режим передачи для PDO Tx; в зависимости от характеристик устройства (определяемых на основании значений EDS) могут быть доступны не все опции.
- **PDO Rx comm settings (Настройки связи PDO Rx)**: здесь можно настроить режим передачи для PDO Rx; в зависимости от характеристик устройства (определяемых на основании значений EDS) могут быть доступны не все опции.

Вкладка SDO SET (НАСТРОЙКА SDO)



#	Label	Index	SubIndex	Type	Value	Timeout
1	Transmission Type	1800	2	USINT	255	100
2	Event Timer	1800	5	UINT	1000	100
3	Transmission Type	1802	2	USINT	255	100
4	Event Timer	1802	5	UINT	1000	100
5	Transmission Type	1400	2	USINT	255	100
6	Transmission Type	1402	2	USINT	255	100

На этой странице вы можете вставить список объектов и значений для отправки ведомому устройству при загрузке в целях конфигурирования, используя пакеты SDO.

Нажмите кнопку **Add (Добавить)**, выберите объекты для отправки, а затем вставьте их **Value (Значение)** в сетку.

Некоторые объекты обрабатываются автоматически: например, **Transmission type (Тип передачи)** и **Event timer (Таймер событий)** настраиваются автоматически в зависимости от настроек **PDO Tx comm settings (Настройки связи PDO Tx)** и **PDO Rx comm settings (Настройки связи PDO Rx)** на вкладке **General (Общее)**.

Вкладки PDO TX — INPUT (PDO TX — входы) и PDO RX — INPUT (PDO RX — ВХОДЫ)

На вкладке **PDO TX — INPUT (PDO TX — ВХОДЫ)** можно настроить объекты данных процесса (Process Data Object, PDO), передаваемые ведомым устройством и поступающие на вход ведущего устройства. На вкладке **PDO RX — OUTPUT (PDO RX — ВЫХОДЫ)** можно настроить PDO, принимаемые ведомым устройством и передаваемые ведущим устройством на выход.

Если **пользовательское устройство CAN** было импортировано с включенной функцией **динамического сопоставления PDO**, вы можете редактировать сопоставление PDO, добавляя и удаляя объекты, а также путем ручного изменения содержимого столбцов **PDO** и **Bit (Бит)**. В противном случае кнопки **Add (Добавить)** и **Remove (Удалить)** будут недоступны, и вам придется использовать конфигурацию PDO как есть.

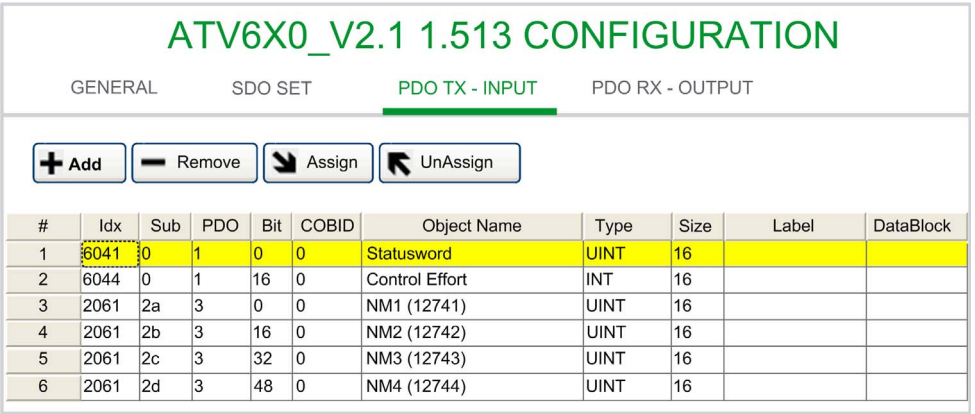
Если вы отметите опцию **Split single bits (Разделять одиночные биты)**, то выбранный вами объект будет вставлен как одиночный бит, связываемый с переменными BOOL (настройка по умолчанию для объектов цифровых входов/выходов в стандарте DS401).

ПРИМЕЧАНИЕ. Эта конфигурация сопоставления PDO не отправляется на устройство. Она нужна только для того, чтобы соответствовать настроенному сопоставлению PDO на устройстве.

Затем с помощью кнопки **Assign (Присвоить)** можно связать каждый объект CAN с **переменной ПЛК** для считывания (PDO Tx) или записи (PDO Rx).

ПРИМЕЧАНИЕ. Для обновления списка переменных ПЛК необходимо пересобрать проект ПЛК с помощью вкладки **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**.

Вкладка **PDO TX — INPUT (PDO TX — ВХОДЫ)** окна **ATV6X0_V2.1.1.513 CONFIGURATION**:



#	Idx	Sub	PDO	Bit	COBID	Object Name	Type	Size	Label	DataBlock
1	6041	0	1	0	0	Statusword	UINT	16		
2	6044	0	1	16	0	Control Effort	INT	16		
3	2061	2a	3	0	0	NM1 (12741)	UINT	16		
4	2061	2b	3	16	0	NM2 (12742)	UINT	16		
5	2061	2c	3	32	0	NM3 (12743)	UINT	16		
6	2061	2d	3	48	0	NM4 (12744)	UINT	16		

Вкладка **PDO RX — OUTPUT (PDO RX — ВЫХОДЫ)** окна **ATV6X0_V2.1.1.513 CONFIGURATION:**

ATV6X0_V2.1 1.513 CONFIGURATION

GENERAL | SDO SET | PDO TX - INPUT | PDO RX - OUTPUT

Add | Remove | Assign | UnAssign

#	Idx	Sub	PDO	Bit	COBID	Object Name	Type	Size	Label	Data Block
1	6040	0	1	0	0	Controlword	UINT	16		
2	6042	0	1	16	0	Target Velocity	INT	16		
3	2061	3e	3	0	0	NC1 (12761)	UINT	16		
4	2061	3f	3	16	0	NC2 (12762)	UINT	16		
5	2061	40	3	32	0	NC3 (12763)	UINT	16		
6	2061	41	3	48	0	NC4 (12764)	UINT	16		

Шина расширения CAN (полевая) — виртуальные каналы ведущих устройств

Обзор

В этом разделе описываются критерии, используемые на вкладке **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** для присвоения идентификаторов виртуальных узлов в зависимости от конфигурации сети.

Описание

При использовании шины расширения CAN на устройстве M171P в режиме **Master (field) (Ведущее устройство [для полевого режима])** открываются три канала ведущих устройств.

Первый канал ведущих устройств используется для обработки запросов, поступающих на его физический ID узла (ID, назначенный вами в окне конфигурирования). Персональный компьютер диспетчера должен подключаться с помощью этого ID узла. Физический ID узла шины расширения CAN addr должен быть в диапазоне от 1 до 122 или 125.

На этом устройстве открыты еще два виртуальных канала ведущих устройств, которые предназначены для связи с клавиатурами (не более двух на каждую сеть шины расширения CAN). ID узлов виртуальных ведущих устройств имеют фиксированные значения:

```
ch_1 = 123
ch_2 = 124
```

Пример: M171P и два графических дисплея M171P

Оба графических дисплея M171P подключены к узлу CAN.

Узел CAN по умолчанию содержит два виртуальных канала, которые могут использоваться для подключения максимум двух дисплеев M171.

Виртуальным каналом по умолчанию для первого дисплея является 124, а для второго дисплея M171 — 123.
Нажмите кнопку **?**, находясь в узле CAN, чтобы просмотреть значения.

Адрес по умолчанию для дисплея **M171 Perf_2** — 127, виртуальный канал по умолчанию — 124, скорость передачи данных по шине CAN по умолчанию 500 кбит/с.

Таким образом, при физическом подключении дисплея M171 к контроллеру M171P с настройками по умолчанию ЧМИ загружается из меню BIOS дисплея M171.

В других случаях, например для дисплея **M171 Perf_1** (с адресом 126), устанавливаются адре^с 126 и виртуальный канал 123 из меню BIOS дисплея M171.

Раздел 5.9
RS485

Содержание этого раздела

В этом разделе описаны следующие темы.

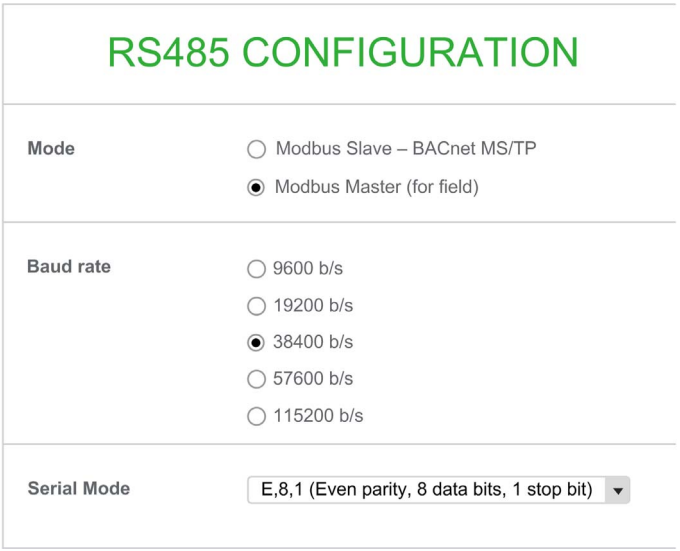
Тема	Страница
RS485	118
Использование TM171E27I/O в качестве ведомого устройства RS485	119
Обзор объекта типового устройства Modbus	120
Сообщения объекта типового устройства Modbus	122
Пользовательские устройства Modbus	124
Использование пользовательского устройства Modbus	126

RS485

Описание

Дисплей M171P/M171P с креплением для скрытого монтажа имеет один встроенный порт RS485, еще один доступен в качестве внешнего подключаемого модуля. Каждый порт может быть сконфигурирован как **Not used (Не используется)**, тогда он будет отключен, или **Modbus Master (Ведущее устройство Modbus)** для работы в полевом режиме.
Устройство M172 имеет два встроенных порта RS485.

Первый порт (RS485-1) используется для ведомого устройства Modbus — BACnet MS/TP. Порт RS485-2 можно настроить в качестве ведомого устройства **Modbus Slave — BACnet MS/TP (Ведомое устройство Modbus — BACnet MS/TP)** или **Modbus Master (for field) (Ведущее устройство Modbus [для полевого режима])**.



Полевой режим

При конфигурировании порта RS485 в качестве **Master (Ведущее устройство)** целевое устройство выступает в роли ведущего устройства Modbus RTU для данного порта. Таким образом, вы теперь можете подключать ведомые устройства Modbus.
Для порта ведущего устройства Modbus необходимо выполнить следующие настройки:

- Скорость передачи данных, используемая в этой сети Modbus (в бит/с).
- Последовательный режим (четность, биты данных, стоповые биты).

Для добавления ведомых устройств Modbus щелкните правой кнопкой мыши **RS485** и выберите команду **Add (Добавить)**. Кроме того, ведомое устройство Modbus можно перетащить из окна **Catalog (Каталог)** на элемент **RS485** в окне **Resources (Ресурсы)**.

После добавления и настройки различных ведомых устройств Modbus *(см. стр. 119)* вы можете связать находящиеся на расстоянии объекты ведомого устройства и внутренние переменные ПЛК для считывания или записи.

Набор объектов ПЛК, доступных для чтения или записи:

- переменные состояния.
- Полевые переменные, объявленные в **I/O Mapping (Сопоставление входов/выходов)** → **Field (Полевые)**.

Использование TM171E27I/O в качестве ведомого устройства RS485

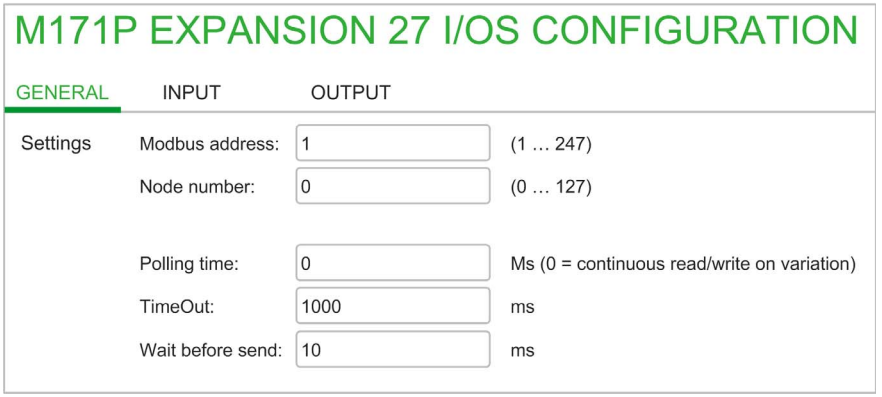
Описание

В этом примере конфигурации **модуль расширения Modicon M171 Performance на 27 входов/выходов** используется для расширения возможностей дисплея M171P. То же самое можно сделать и для остальных логических контроллеров.

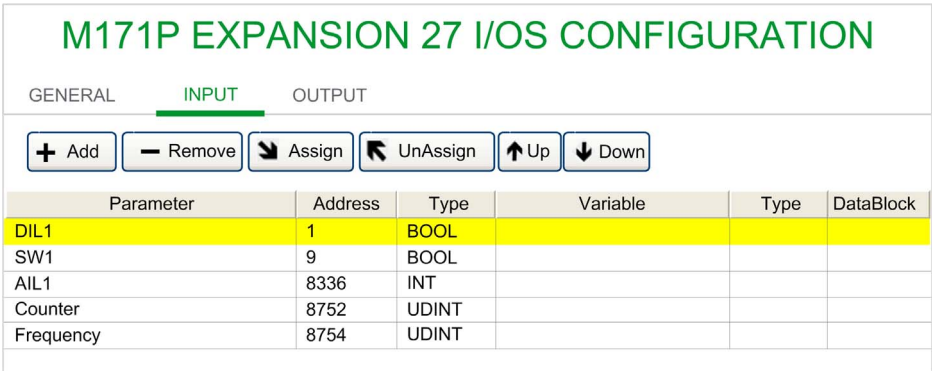
Настройте устройство M171P Display RS485 в режиме **Modbus Master (for field) (Ведущее устройство Modbus [для полевого режима])**. Находясь в окне **Catalog (Каталог)**, можно выбрать узел **модуля расширения Modicon M171 Performance на 27 входов/выходов** и перетащить его на узел **RS485**.
Процедура конфигурирования **модуля расширения Modicon M171 Performance на 27 входов/выходов** аналогична процедуре конфигурирования пользовательского устройства Modbus *(см. стр. 126)* . Вы можете присвоить доступный словарь объектов входов/выходов **модуля расширения Modicon M171 Performance на 27 входов/выходов** переменным ПЛК **дисплея M171 Performance**.

Модуль **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** знает словарь **модуля расширения Modicon M171 Performance на 27 входов/выходов**. Вы можете добавлять, удалять, назначать, отменять назначение или изменять позиции объектов `Input (Вход)` и `Output (Выход)`. Только назначенные объекты могут запрашиваться **дисплеем M171P**.

Вкладка **GENERAL (ОБЩЕЕ)** окна **конфигурирования модуля расширения M171P на 27 входов/выходов:**

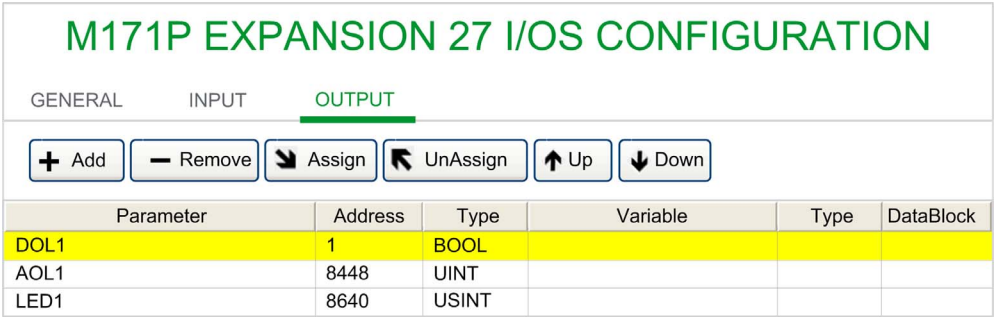


Вкладка **INPUT (ВХОД)** окна конфигурирования **модуля расширения M171P на 27 входов/выходов:**



Parameter	Address	Type	Variable	Type	DataBlock
DIL1	1	BOOL			
SW1	9	BOOL			
AIL1	8336	INT			
Counter	8752	UDINT			
Frequency	8754	UDINT			

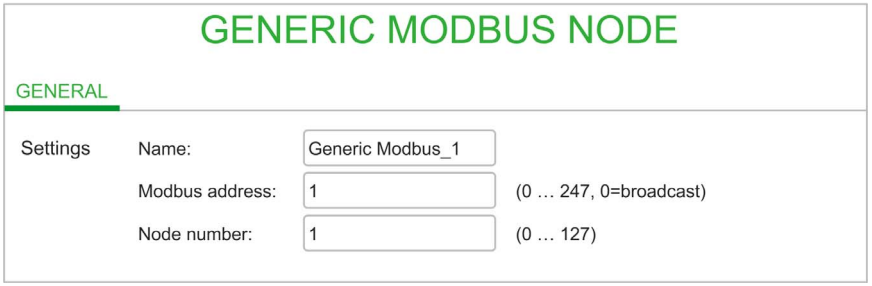
Вкладка **OUTPUT (ВЫХОД)** окна конфигурирования **модуля расширения M171P на 27 входов/выходов:**



Обзор объекта типового устройства Modbus

Описание

Объект **Generic Modbus (Типовое устройство Modbus)** — это универсальное ведомое устройство Modbus, которое при конфигурировании в качестве **Modbus Master (Ведущее устройство Modbus)** может быть подключено к порту RS485 логического контроллера.



Чтобы добавить объект **типового устройства Modbus**, выполните следующие действия.

Шаг	Действие
1	В окне **Resources (Ресурсы)** нажмите **RS485**.
2	• Перетащите **Generic Modbus (Типовое устройство Modbus)** из окна **Catalog (Каталог)** на элемент **RS485** в окне **Resources (Ресурсы)**. • Или щелкните правой кнопкой мыши по **RS485** и выберите команду **Add (Добавить)**. **Типовое устройство Modbus** отобразится под пунктом **RS485**.

Вы можете использовать **типовое устройство Modbus**, если хотите выполнить настройки вручную и иметь полный контроль над отдельными сообщениями Modbus, которые необходимо отправлять ведомому устройству.

Типовое устройство Modbus также часто применяется в ситуациях, когда сторонние устройства планируется использовать в проекте однократно, т. е. внесение их в каталог для повторного использования в будущем не требуется.

На главной странице **типового устройства Modbus** вам доступны следующие настройки:

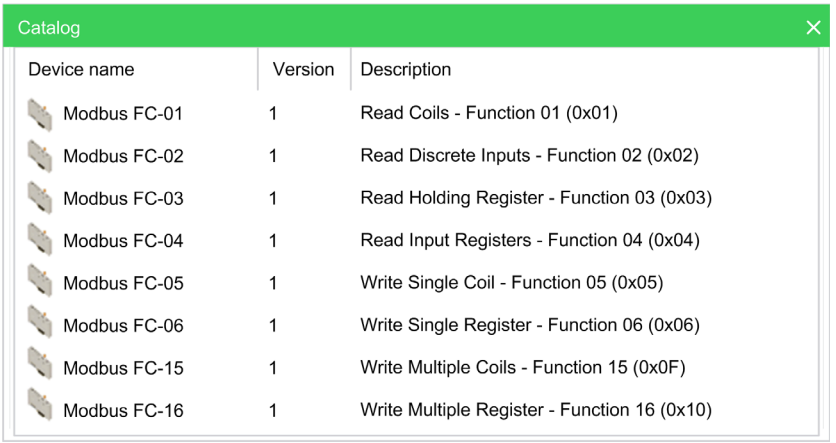
- Имя объекта в проекте.
- Его адрес Modbus (в диапазоне 1–247).
- Номер (ID) этого узла в диапазоне 0–127: это значение инкрементируется автоматически и может быть использовано в программе ПЛК для индексирования массива `SysMbMRtuNodeStatus[]`, содержащего диагностическую информацию о каждом ведомом узле.

Сообщения объекта типового устройства Modbus

Описание

Объект **Generic Modbus (Типовое устройство Modbus)** сам по себе ничего не делает. Вы должны добавить в него одно или несколько **сообщений Modbus**, которые являются специфическими запросами функций Modbus, посылаемыми по шине.

Чтобы добавить функцию к объекту **типового устройства Modbus**, перетащите функцию Modbus из окна **Catalog (Каталог)** на элемент **Generic Modbus (Типовое устройство Modbus)** в окне **Resources (Ресурсы)**. Вы также можете щелкнуть правой кнопкой мыши по **Generic Modbus (Типовое устройство Modbus)** и выбрать команду **Add (Добавить)**.

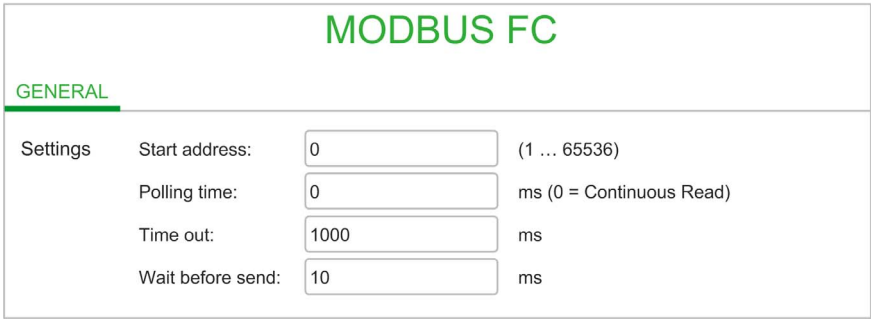


Функция		Описание	Подробная информация	Длина объектов
1	0x01	Чтение катушек	Читает одну или несколько дискретных выходных катушек, доступных только для чтения	1 бит
2	0x02	Чтение дискретных входов	Читает один или несколько цифровых входов, доступных только для чтения	1 бит
3	0x03	Чтение регистров хранения	Читает один или несколько регистров, доступных для чтения/записи	16 бит
4	0x04	Чтение входных регистров	Читает один или несколько регистров, доступных только для чтения	16 бит
5	0x05	Запись единичной катушки	Записывает одну дискретную выходную катушку	1 бит
6	0x06	Запись единичного регистра	Записывает один регистр хранения аналогового выхода	16 бит
15	0x0F	Запись нескольких катушек	Записывает один или несколько цифровых выходов	1 бит
16	0x10	Запись нескольких регистров	Пишет один или несколько регистров	16 бит

Сообщения обрабатываются в порядке их вставки в дерево.

Вкладка General (Общее)

Выберите функцию Modbus, добавленную в дерево, чтобы отобразить ее окно конфигурирования:

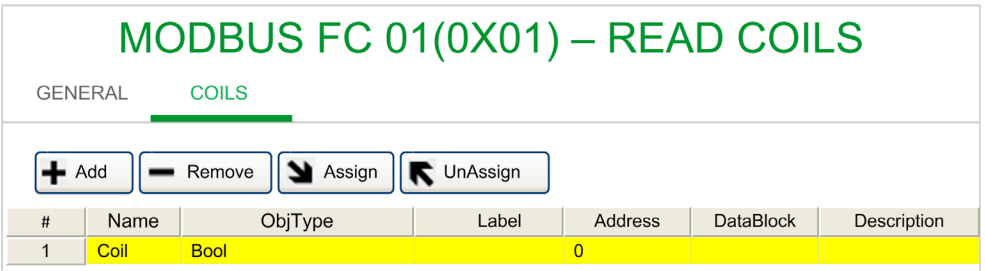


Для каждого сообщения в его вкладке **General (Общее)** можно выполнить следующие настройки:

- **Start address (Начальный адрес)** — адрес первого объекта Modbus для чтения или записи (1–65536).
- **Polling time (Интервал опроса)** — сообщение обрабатывается с указанной периодичностью (мс):
  - для операций записи 0 означает, что запись возможна только при изменении значения;
  - для операций чтения 0 означает максимальную скорость.
- **Time out (Тайм-аут)** — по истечении этого тайм-аута (мс) операция считается неудачной.
- **Wait before send (Ожидание перед отправкой)** — время ожидания перед отправкой очередного запроса на ведомое устройство.

Вкладка Coils (Катушки)

Помимо вкладки **General (Общее)** для каждого отдельного сообщения доступна вторая вкладка, где можно настроить список объектов для чтения или записи.



С помощью кнопки **Add (Добавить)** вставьте по одной строке для каждого объекта Modbus для чтения или записи. Список может содержать до 16 элементов. В первой строке указан адрес, настроенный в поле **Address (Адрес)** на вкладке **General (Общее)**, в остальных строках эти значения увеличиваются на единицу.

Для каждой строки нажмите кнопку **Assign (Присвоить)**, чтобы выбрать объект ПЛК, с которым необходимо установить связь и который должен быть прочитан или записан с помощью данного сообщения Modbus. В сообщении нельзя оставлять неприсвоенные строки.

ПРИМЕЧАНИЕ. Не забудьте пересобрать проект ПЛК, чтобы увидеть обновленный список переменных ПЛК.

Пользовательские устройства Modbus

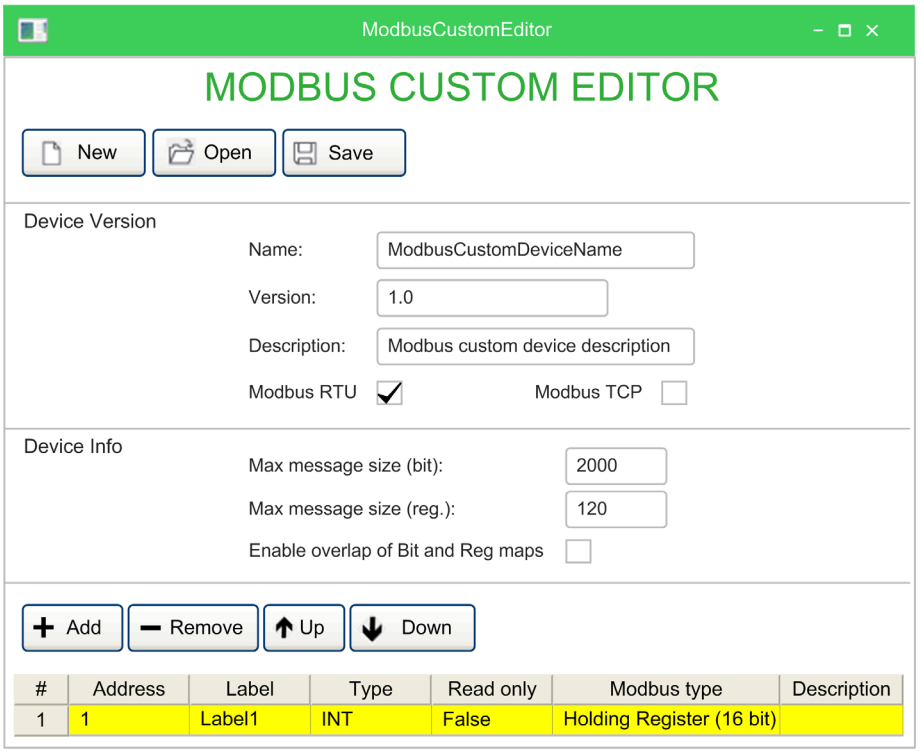
Описание

Вы можете напрямую создавать **пользовательские устройства Modbus.**

Таким образом, достаточно один раз настроить в проекте и добавить в каталог карту стороннего ведомого устройства Modbus, чтобы использовать его многократно и больше не думать о сообщениях и функциях Modbus.

Создание нового пользовательского устройства Modbus

Чтобы создать новое пользовательское устройство Modbus, выберите **Developer (Разработчик)** → **Run Modbus Custom Editor (Запустить редактор Modbus Custom Editor)**. Запустится внешний инструмент **Modbus Custom Editor** с новым пустым документом.



Пользователю доступны следующие настройки:

- Имя устройства.
- Подробное описание устройства.
- Номер версии.
- Перекрытие карт битов и регистров: установите флажок, если в устройстве имеется и регистр **0**, и бит **0** (другими словами, присутствует разная адресация 16-битных и 1-битных объектов). Снимите этот флажок, если адрес уникален и дублирование недопустимо, даже если тип отличается.
- Максимальный размер сообщения: укажите здесь максимальное количество регистров на сообщение, поддерживаемое устройством.

Затем с помощью кнопки **Add (Добавить)** добавьте по одной строке для каждого объекта Modbus устройства. Следует указать его адрес, имя, тип и, по желанию, подробное описание. Обратите внимание, что столбцы **Type (Тип)** и **Read only (Только для чтения)** связаны со столбцом **Modbus type (Тип Modbus)**.

По завершении сохраните текущее определение устройства; вам будет предложено указать имя файла с расширением **.PCT**, по умолчанию предлагается формат «текущее имя + версия».

Файл сохраняется в специальной папке **ModbusCustom** в каталоге, теперь **Modbus Custom Editor** можно закрыть и вернуться в раздел **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** для использования нового устройства.

Редактирование имеющегося пользовательского устройства Modbus

Для редактирования имеющегося пользовательского устройства Modbus можно выполнить следующие действия:

- Запустите редактор **Modbus Custom Editor** командой **Developer (Разработчик) → Run Modbus Custom Editor (Запустить редактор Modbus Custom Editor)**, а затем вручную откройте файл PCT стандартной командой **File (Файл) → Open (Открыть)**.
- Если устройство, которое необходимо отредактировать, отображается в окне **Catalog (Каталог)** (например, если выбран узел RS485 и находится в режиме **ведущего устройства)**, вы можете щелкнуть по нему правой кнопкой мыши и выбрать команду **Edit device (Редактировать устройство)**; запустится редактор **Modbus Custom Editor** и откроется выбранное устройство.

Удаление пользовательского устройства Modbus

Чтобы удалить существующее пользовательское устройство Modbus, отображаемое в окне **Catalog (Каталог),** щелкните по нему правой кнопкой мыши и выберите пункт **Delete from catalog (Удалить из каталога)**.

Использование пользовательского устройства Modbus

Описание

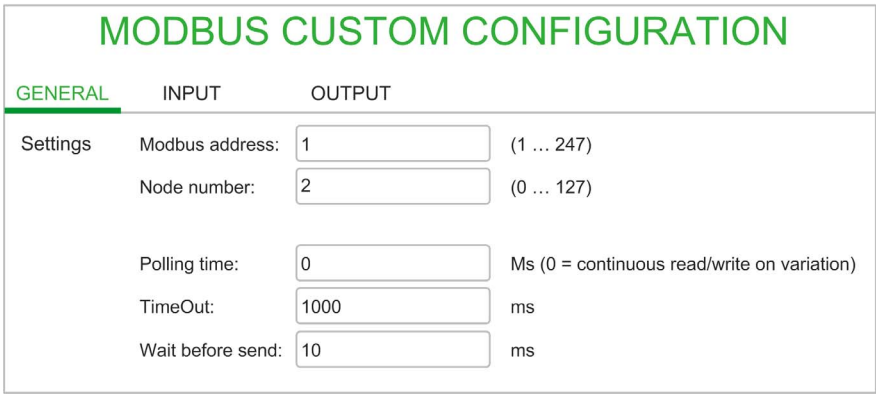
Если вставить **пользовательское устройство Modbus** в качестве ведомого устройства Modbus (например, под портом RS485) и щелкнуть по нему в окне **Resources (Ресурсы)**, то в окне редактора отобразятся три вкладки.

Вкладка General (Общее)

Во вкладке **General (Общее)** доступны следующие настройки:

- **Modbus address (Адрес Modbus)**: адрес Modbus устройства в диапазоне 1–247.
- **Node number (Номер узла)**: номер узла устройства в диапазоне 0–127. Это значение инкрементируется автоматически и может быть использовано в программе ПЛК для индексирования массива `SysMbMRtuNodeStatus[]`, содержащего диагностическую информацию о каждом ведомом узле.
- **Polling time (Интервал опроса):** сообщения Modbus обрабатываются с этой периодичностью (мс); в случае операций записи 0 означает выполнение записи только при изменении значения, в случае операций чтения 0 означает максимальную скорость.
- **Time out (Тайм-аут)**: по истечении этого тайм-аута (мс) операция считается неудачной.
- **Wait before send (Ожидание перед отправкой)**: это дополнительный тайм-аут (для использования с медленными ведомыми устройствами, которые не отвечают, если сообщения отправляются слишком быстро).

Вкладка **GENERAL (ОБЩЕЕ)** окна **MODBUS CUSTOM CONFIGURATION (ПОЛЬЗОВАТЕЛЬСКАЯ КОНФИГУРАЦИЯ MODBUS)**:



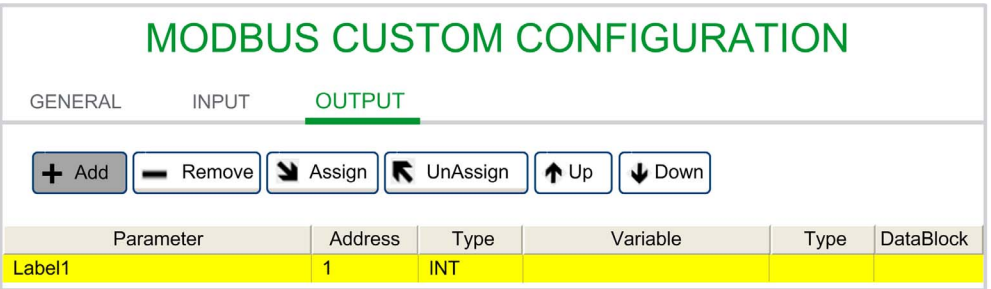
Здесь вы можете заметить, что настройки **Polling time (Интервал опроса)**, **Timeout (Тайм-аут)** и **Wait before send (Ожидание перед отправкой)** для пользовательского устройства Modbus являются общими для всего устройства, в то время как для **типового устройства Modbus** можно задать различные значения для каждого отдельного сообщения. Это связано с тем, что в случае **пользовательских устройств Modbus** низкоуровневые сообщения Modbus рассчитываются автоматически. Однако вы не можете настроить их под себя, потому что эти параметры являются глобальными.

Вкладки Input (Вход) и Output (Выход)

Во вкладках **Input (Вход)** и **Output (Выход)** вы можете вставить по строке для каждого объекта Modbus с целью чтения или записи. Нажмите кнопку **Add (Добавить)** и выберите параметры для обмена (поддерживается многократный выбор). Используйте кнопку **Assign (Присвоить)**, чтобы связать их со считываемым или записываемым объектом ПЛК.

Вкладка **INPUT (ВХОД)** окна **MODBUS CUSTOM CONFIGURATION (ПОЛЬЗОВАТЕЛЬСКАЯ КОНФИГУРАЦИЯ MODBUS)**:

Вкладка **OUTPUT (ВЫХОД)** окна **MODBUS CUSTOM CONFIGURATION (ПОЛЬЗОВАТЕЛЬСКАЯ КОНФИГУРАЦИЯ MODBUS)**:



Укажите во вкладке **Input (Вход)** объекты Modbus, предназначенные для чтения `(READ)` с ведомого устройства Modbus (и для вставки в переменные ПЛК). Укажите во вкладке **Output (Выход)** объекты Modbus, предназначенные для записи `(WRITE)` на ведомое устройство Modbus (и для получения из переменных ПЛК).

ПРИМЕЧАНИЕ. Не забудьте пересобрать проект ПЛК с приложением, чтобы увидеть обновленный список переменных ПЛК.

В окне **CONFIGURATION (КОНФИГУРИРОВАНИЕ)** создаются корректные сообщения Modbus посредством анализа последовательности адресов и типов. Если адреса следуют друг за другом, а типы однородны, то для оптимизации обмена данными в одном сообщении группируются различные объекты.

Наряду с максимальным количеством регистров на одно сообщение ведущего устройства (16 для M171P/M172) также учитывается максимальное количество регистров, сконфигурированных в **Modbus Custom Editor**.

Группировка и генерирование сообщений Modbus происходят автоматически, пересчет выполняется при каждой компиляции.

Раздел 5.10
Ethernet

Ethernet

Описание

Устройства M171P с креплением для скрытого монтажа / M172P оснащаются встроенным портом Ethernet. M171P/M172O может иметь один порт Ethernet, доступный в виде внешнего коммуникационного модуля. Порт Ethernet может быть сконфигурирован как Modbus TCP двумя способами:

- **Server only (Только сервер)**: контроллер поддерживает связь на основании запросов от других контроллеров.
- **Client/Server (Клиент/сервер)**: контроллер поддерживает связь по протоколу Modbus TCP на основании запросов других контроллеров, а также отправляет запросы на другие контроллеры.

ПРИМЕЧАНИЕ. Компания Schneider Electric придерживается лучших отраслевых практик при разработке и внедрении систем управления. Это предполагает реализацию концепции всеобъемлющей многоуровневой защиты в вопросах безопасности систем управления производством. В рамках этого подхода контроллеры размещаются за одним или несколькими брандмауэрами с целью предоставления доступа только авторизованному персоналу с использованием разрешенных протоколов.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕСАНКЦИОНИРОВАННЫЙ ДОСТУП И ПОСЛЕДУЮЩАЯ НЕСАНКЦИОНИРОВАННАЯ ЭКСПЛУАТАЦИЯ ОБОРУДОВАНИЯ

- Оцените, подключена ли ваша среда или ваше оборудование к критически важной инфраструктуре, и если это так, перед подключением системы автоматизации к любой сети примите все необходимые меры безопасности, основанные на концепции всеобъемлющей многоуровневой защиты.
- Ограничьте количество устройств, подключенных к сети, до необходимого минимума.
- Изолируйте вашу промышленную сеть от других сетей внутри вашей компании.
- Защитите свои сети от несанкционированного доступа с помощью брандмауэров, VPN или других проверенных мер безопасности.
- Контролируйте действия в своих системах.
- Исключите возможность прямого доступа или несанкционированного подключения к защищаемым устройствам и выполнения неавторизованных операций.
- Подготовьте план восстановления, включая резервное копирование вашей системы и информации о процессах.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Конфигурирование

Чтобы настроить порт Ethernet:

- выберите **Client/Server (Клиент/сервер)** или **Server only (Только сервер)**;
- укажите дополнительные сокеты Modbus TCP (по умолчанию 0).

IP-адрес сохраняется в **Modbus objects (Объекты Modbus)** → **BIOS Parameters (Параметры BIOS)**.

Клиент/сервер

Конфигурация **Client/Server (Клиент/сервер)** позволяет контроллеру отправлять запросы и считывать ответы при взаимодействии с другими устройствами в той же сети Ethernet. Вы можете подключать устройства Modbus и обмениваться с ними данными.

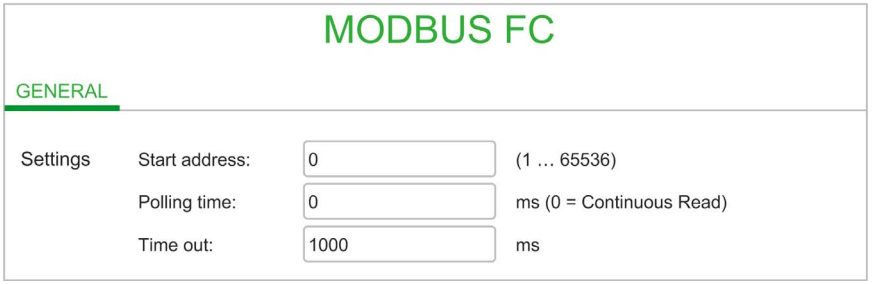
Вы можете добавлять типовые устройства Modbus *(см. стр. 120)* или пользовательские устройства, созданные с помощью редактора Modbus Custom Editor *(см. стр. 124)* таким же образом, как и в случае с RS485.

После добавления и настройки узлов Modbus можно добавить сообщения объектов типового устройства Modbus *(см. стр. 122)* для определения функций `READ (ЧТЕНИЕ)` или `WRITE (ЗАПИСЬ)`.

Набор объектов контроллеров, доступных для отправки или получения:

- **параметры EEPROM** (энергонезависимой памяти);
- **переменные состояния.**

Страница конфигурирования для объекта **Binding (Привязка)** в Modbus TCP такая же, как и для шины расширения CAN. Описание и использование этой страницы см. в главе «Шина расширения CAN — привязка» *(стр. 103)*.



Единственное отличие от привязки шины расширения CAN заключается в том, что здесь имеется еще один столбец под названием **Timeout (Тайм-аут)**, где вы можете настроить определенный тайм-аут в мс для каждого передаваемого объекта.

Раздел 5.11
Подключаемые модули

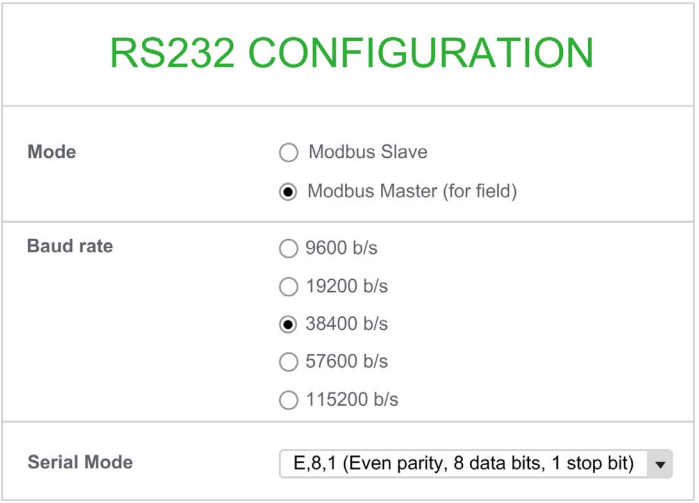
Обзор линейки коммуникационных модулей

Обзор

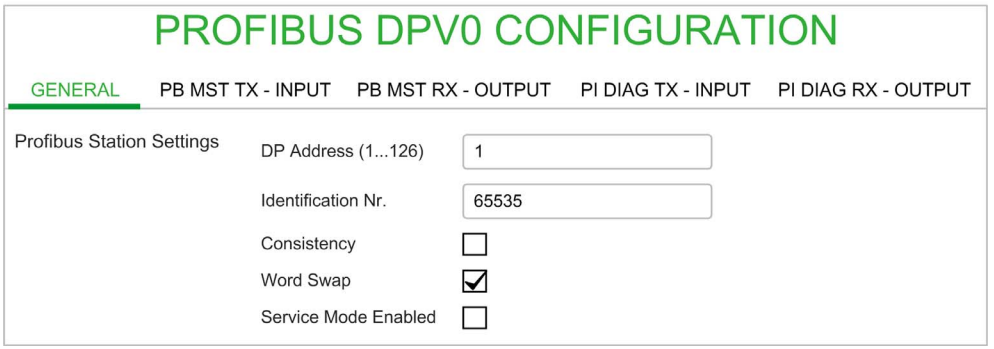
Коммуникационные модули (подключаемые модули), которые можно добавить, отображаются в окне **Catalog (Каталог)**.

Перетащите коммуникационный модуль в элемент **Plugins (Подключаемые модули)**.

Пример окна конфигурирования RS232:



Пример окна конфигурирования Profibus DP:



Номера для заказа коммуникационных модулей

Номер для заказа	Описание	Тип клеммы	Совместимые контроллеры
TM171ACAN	CAN	2 клеммные колодки	M172P M172O M171P(1)
TM171ALON	LonWorks	1 клеммная колодка	
TM171AMB	Modbus SL (RS-485)	2 клеммные колодки	
TM171ARS232	Последовательный канал RS232, выход реле	1 разъем SUB-D 9 1 клеммная колодка	
TM171ARS485	Modbus SL и BACnet MS/TP	2 клеммные колодки	
TM171AETH	Ethernet, Modbus TCP и BACnet/IP	1 разъем RJ45	M172O M171P(1)
TM171AETHRS485	Ethernet, Modbus TCP, BACnet/IP, Modbus SL и BACnet MS/TP	1 RJ45 2 клеммные колодки	
TM171APBUS	PROFIBUS	1 разъем SUB-D 9	M171P(1)
(1) Неприменимо к M171P с креплением для скрытого монтажа.			

Дополнительную информацию о коммуникационных модулях см. в инструкции Modicon M171A Communication Modules Instruction Sheet (Модули связи Modicon M171A), *EAV96007*.

Глава 6

Справочник по технической реализации

Раздел 6.1
Протокол Modbus

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	135
Типы данных	136
Коды функций	137

Обзор

Введение

Используется режим передачи RTU. Кадр не содержит байтов заголовка и конца сообщения.

Адрес ведомого устройства	Код запроса	Данные	CRC16

Данные передаются в двоичном формате.
CRC16 — контроль целостности при помощи циклического избыточного кода.
Конец кадра определяется по паузе, которая больше или равна трем символам.

Принцип

Одновременно передавать данные по линии может только одно устройство.
Обменом данными управляет ведущее устройство, которое при необходимости может брать инициативу в свои руки. Оно опрашивает каждое ведомое устройство по очереди.
Ни одно ведомое устройство не может отправлять сообщения, пока его об этом не попросят.

Ведущее устройство повторяет вопрос при некорректном обмене данными и объявляет опрашиваемое ведомое устройство недоступным, если ответ не поступает в течение заданного промежутка времени.

Если ведомое устройство не получает сообщения, оно посылает ведущему устройству ответ с исключением. Ведущее устройство может повторить или не повторить запрос.
Прямая связь между ведомыми устройствами не поддерживается.

Поэтому для обеспечения связи между ведомыми устройствами прикладное программное обеспечение должно быть спроектировано таким образом, чтобы опрашивать одно ведомое устройство и отправлять полученные данные другому ведомому устройству.
Между ведущим и ведомым устройствами возможны два типа диалога:

- Ведущее устройство посылает запрос ведомому и ждет от него ответа.
- Ведущее устройство посылает запрос всем ведомым устройствам и не ждет от них ответа (принцип широковещательной рассылки).

Адреса

Указание адреса:
- Адрес Modbus устройства можно настроить в диапазоне от 1 до 247.
- Адрес 0, кодируемый в запросе, отправляемом ведущим устройством, зарезервирован для широковещательной рассылки. Ведомые устройства принимают этот запрос, но не отвечают на него.

Типы данных

Описание

Информация, хранящаяся в ведомом устройстве, может быть четырех типов: первые два — дискретные значения вкл./выкл. (катушки и контакты), а два других — числовые значения (регистры).

- Контакты дискретного входа (только для чтения), 1 бит.
- Контакты дискретного выхода (чтение/запись), 1 бит.
- Регистры аналогового входа (только для чтения), 16 бит.
- Регистры хранения аналогового выхода (чтение/запись), 16 бит.

Для работы с более сложными типами данных (например, с 32-битными целыми числами или числами с плавающей точкой) необходимо использовать два или больше последовательных регистра и читать или записывать их вместе.

Коды функций

Описание

Протокол Modbus определяет различные «функциональные коды» для каждого сообщения Modbus:

- `01 (01 hex)` : считывание дискретных выходных катушек.
- `05 (05 hex)` : запись одной дискретной выходной катушки.
- `15 (0F hex)` : запись нескольких дискретных выходных катушек.
- `02 (02 hex)` : чтение контактов дискретных входов.
- `04 (04 hex)` : чтение регистров аналоговых входов.
- `03 (03 hex)` : чтение регистров хранения аналоговых выходов.
- `06 (06 hex)` : запись одного регистра хранения аналогового выхода.
- `16 (10 hex)` : запись нескольких регистров хранения аналогового выхода.

Часть 3
Программирование

Содержание данной части

Эта часть содержит следующие главы.

Глава	Название главы	Страница
7	Вкладка PROGRAMMING (ПРОГРАММИРОВАНИЕ)	141
8	Параметры проекта	163
9	Управление элементами проекта	181
10	Редактирование исходного кода	225
11	Компиляция	269
12	Запуск приложения	277
13	Отладка	293
14	Справочник по языку	371
15	Справочник по ошибкам	445

Глава 7
Вкладка PROGRAMMING (ПРОГРАММИРОВАНИЕ)

Раздел 7.1
Обзор

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор окна Programming (Программирование)	143
Панель меню	147
Панели инструментов	155
Строка состояния	162

Обзор окна Programming (Программирование)

Ввод в эксплуатацию и испытания

Перед началом эксплуатации электрических управляющих устройств и оборудования автоматизации в штатном режиме после завершения установки необходимо провести пусконаладочные испытания системы с привлечением специалистов, чтобы подтвердить правильность ее работы. Важно не просто организовать такую проверку, но и выделить достаточно времени на проведение полных и качественных испытаний.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Убедитесь, что процедуры по установке и настройке завершены.
- Перед выполнением проверки работоспособности удалите с компонентов оборудования все средства блокировки и временной фиксации, используемые для транспортировки.
- Убедитесь, что из оборудования убраны инструменты, измерительные приборы и посторонние предметы.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Общее описание

Рабочая среда **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** содержит различные окна для разработки приложений контроллера (например, программирование на совместимых языках МЭК 61131-3 [ГОСТ Р МЭК 61131-3]), тестирования, отладки и загрузки приложений контроллера на целевое устройство.

В разделе **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** вам доступны две возможности для загрузки:

- Загрузка только приложения контроллера по нажатию **On-line (Онлайн) → Download code (Загрузить код)**.
- Загрузка проекта (который включает приложение контроллера, параметры BIOS и ПЛК, а также их значения по умолчанию) по нажатию значка **Download all (Загрузить все)** на панели инструментов проекта.

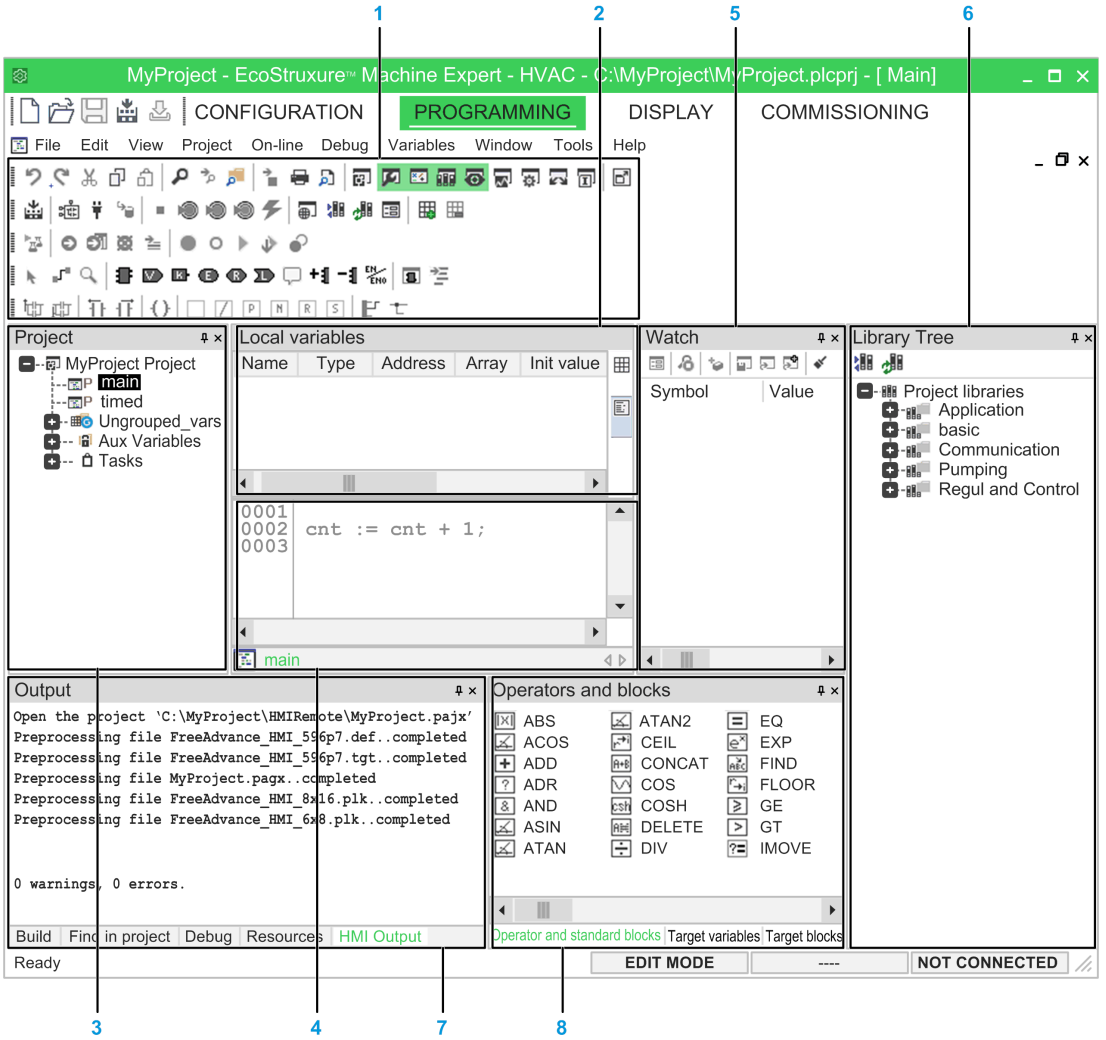
⚠ ПРЕДУПРЕЖДЕНИЕ

АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК КОНТРОЛЛЕРА

- Прежде чем загружать приложение, сначала проверьте состояние машины или процесса.
- Перед загрузкой приложения убедитесь в отсутствии опасностей, способных привести к травмированию людей, находящихся внутри или рядом с машиной или в непосредственной близости от технологического процесса.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Ниже показаны окна:



Позиция	Описание	
1	Панели инструментов	Многие функции, доступные в меню, отображаются здесь в виде значков на панелях инструментов. Эти значки помогут вам создать приложение. Наиболее часто используются панели **Main (Главная)** и **Project (Проект)**. Информацию по управлению панелями инструментов см. в разделе «Панели инструментов» *(стр. 43)*.
2	Окно **Local variables (Локальные переменные)**	Здесь выводятся глобальные и локальные переменные кода, отображаемые в редакторе исходного кода (программы, функциональные блоки и функции). Информацию по использованию окна **Local variables (Локальные переменные)** см. в разделе «Переменные» *(см. стр. 188)* .
3	Окно **Project (Проект)**	Это окно позволяет: ● управлять кодом приложения; ● определять комплексные пользовательские переменные и управлять ими; ● управлять меню целевого устройства. Информацию по использованию окна **Project (Проект)** см. в разделе «Окно Project (Проект)» *(см. стр. 182)*.
4	Редактор исходного кода	Это окно позволяет редактировать исходный код на любом из пяти языков программирования, определенных стандартом МЭК 61131-3 (ГОСТ Р МЭК 61131-3), а также управлять исходными файлами и печатать их содержимое. Для определения как глобальных, так и локальных переменных используются специальные редакторы, похожие на редакторы электронных таблиц. Информацию по работе с редактором исходного кода см. в разделе «Редактирование исходного кода» *(стр. 225)*.
5	Окно **Watch (Контрольные значения)**	Это окно позволяет управлять отладкой переменных путем отображения их состояния в числовом формате, если приложение запущено и подключено к целевому устройству. Информацию по использованию окна **Watch (Контрольные значения)** см. в разделе «Окно Watch (Контрольные значения)» *(см. стр. 295)*.
6	Окно **Library Tree (Дерево библиотеки)**	Окно **Library Tree (Дерево библиотеки)** содержит набор различных объектов. Каждый объект группируется в папку, к которой он принадлежит. Эти папки полезны для логической группировки элементов библиотеки. Чтобы отобразить свойства объекта, щелкните правой кнопкой мыши его имя и выберите команду **Object Properties (Свойства объекта)**. Откроется окно **Properties (Свойства)**, в котором отобразятся его свойства. Информацию по управлению библиотеками см. в разделе «Работа с библиотеками» *(см. стр. 174)*.
7	Окно **Output (Вывод)**	В этом окне инструментов отображаются сообщения, относящиеся к разработке проекта. Для получения более подробной информации см. описание окна Output (Вывод) *(см. стр. 146)*.
8	Окно **Operators and blocks (Операторы и блоки)**	Этот инструмент позволяет управлять библиотеками функций по умолчанию или пользовательскими библиотеками функций. Данное окно разделено на различные вкладки, по одной для каждой библиотеки. Всегда доступны следующие вкладки: ● **Operator and standard blocks (Оператор и стандартные блоки)**: операторы (AND, OR и пр.). ● **Target variables (Целевые переменные)**: отдельные переменные целевого устройства. ● **Target blocks (Целевые блоки)**: отдельные функции целевого устройства. Управление дополнительными вкладками осуществляется с помощью меню **Project (Проект) → Library manager (Диспетчер библиотек)**.

Окно Output (Вывод)

В этом окне инструментов отображаются сообщения, относящиеся к разработке проекта. Окно разделено на пять вкладок:

- **Build (Сборка)**: информация, связанная с открытием файла, ошибками компиляции и загрузкой кода в устройство.
- **Find in project (Найти в проекте)**: результат поиска в проекте.
- **Debug (Отладка)**: информация об отладке (например, точки останова).
- **Resources (Ресурсы)**: сообщения, относящиеся к целевому устройству.
- **HMI Output (Вывод ЧМИ)**: сообщения, относящиеся к процессу **DISPLAY (ОТОБРАЖЕНИЕ)**.

ПРИМЕЧАНИЕ. Подключение к целевому устройству также отображается в строке состояния *(см. стр. 162)*.

Панель меню

Обзор

В приведенных ниже таблицах представлен список команд раздела **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**. Однако поскольку интерфейс раздела **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** поддерживает открытие нескольких документов (MDI), вы можете обнаружить некоторые отключенные команды или даже целые меню — в зависимости от того, какой тип документа открыт в данный момент.

Меню File (Файл)

Команда	Значок	Клавиша	Описание
New project (Новый проект)		–	Создает новый проект *(см. стр. 53)*.
Open project (Открыть проект)		–	Открывает существующий проект *(см. стр. 59)*.
Import project from target (Импортировать проект с целевого устройства)	–	–	Импортирует проект источника с целевого устройства.
View project (read only) (Просмотр проекта [только для чтения])	–	–	Открывает существующий проект в режиме только для чтения.
Save project (Сохранить проект)		–	Сохраняет текущий открытый проект. *(см. стр. 57)*.
Save project as… (Сохранить проект как…)	–	–	Сохраняет текущий открытый проект, позволяя указать новое имя, местоположение и расширение *(см. стр. 57)*.
Close project (Закрыть проект)	–	–	Закрывает открытый проект *(см. стр. 59)*.
New text file (Новый текстовый файл)	–	–	Открывает новый пустой текстовый файл.
Open file (Открыть файл)	–	Ctrl+O	Открывает существующий файл независимо от его расширения. Файл отображается в текстовом редакторе. В любом случае, открывая файл проекта, вы открываете проект, к которому он относится.
Save (Сохранить)	–	Ctrl+S	Сохраняет документ, открытый в текущем активном окне.
Close (Закрыть)	–	–	Закрывает документ, открытый в текущем активном окне.
Options (Параметры)	–	–	Открывает диалоговое окно Program options (Параметры программы) *(см. стр. 47)*.
Print… (Печать…)		Ctrl+P	Распечатывает документ, открытый в текущем активном окне *(см. стр. 56)*.
Print preview (Предпросмотр перед печатью)		–	Создает предварительный просмотр документа, открытого в текущем активном окне, готовый к печати.
Print Project (Печать проекта)	–	–	Печатает документы, входящие в состав проекта.
Printer setup (Настройки принтера)	–	–	Открывает диалоговое окно Printer setup (Настройки принтера).
..recent.. (недавние)	–	–	Выводит список недавно открытых файлов проектов.
Exit (Выход)	–	–	Закрывает EcoStruxure Machine Expert - HVAC.

Меню Edit (Редактирование)

Команда	Значок	Клавиша	Описание
Undo (Отменить)		Ctrl+Z	Отменяет последнее действие.
Redo (Вернуть)		Ctrl+Y	Восстанавливает последнее действие, отмененное посредством команды Undo (Отменить).
Cut (Вырезать)		Ctrl+X	Удаляет выбранные элементы из активного документа и сохраняет их в системном буфере.
Copy (Копировать)		Ctrl+C	Копирует выбранные элементы в системный буфер.
Paste (Вставить)		Ctrl+V	Вставляет в активный документ содержимое системного буфера.
Delete line (Удалить строку)	–	Ctrl+E	Удаляет всю строку исходного кода.
Go to symbol (Перейти к символу)		Shift+F12	Переходит к объявлению переменной.
Find in project (Найти в проекте)		Ctrl+ Shift+F	Открывает диалоговое окно Find in project (Найти в проекте).
Bookmarks… (Закладки…)	–	–	
Add/toggle (Добавить/ переключить)	–	Ctrl+F2	Добавляет закладку для выделения строк. Если закладка уже установлена, удаляет ее.
Next (Следующая)	–	F2	Переходит к следующей установленной закладке.
Prev (Предыдущая)	–	Shift+F2	Переходит к предыдущей установленной закладке.
Remove all (Удалить все)	–	–	Удаляет установленные закладки.
Go to line (Перейти к строке)	–	Ctrl+G	Позволяет перейти к определенной строке в редакторе исходного кода.
Find… (Найти…)		Ctrl+F	Запрашивает ввод строки и ищет ее первый экземпляр в активном документе, начиная с текущего местоположения курсора.
Find next (Найти далее)		F3	Переход к следующему результату поиска, найденному командой **Find (Найти)**.
Replace (Заменить)	–	Ctrl+H	Позволяет автоматически заменять один или все экземпляры строки на другую строку.
Insert/Move mode (Режим вставки/перемещения)		–	Переключение между этими двумя режимами редактирования, используемыми для вставки или перемещения блоков.
Connection mode (Режим соединения)		–	Режим редактирования, позволяющий чертить логические линии для подключения выводов.
Watch mode (Режим контроля)		–	Режим редактирования, который позволяет добавлять переменные в любой инструмент отладки.

Меню View (Вид)

Команда	Значок	Клавиша	Описание
Панели инструментов	–	–	См. раздел «Панели инструментов» *(см. стр. 43)*.
Main (Главная)	–	–	Отображает или скрывает панель инструментов **Main (Главная)**.
Project (Проект)	–	–	Отображает или скрывает панель **Project (Проект)**.
Debug (Отладка)	–	–	Отображает или скрывает панель инструментов **Debug (Отладка)**.
FBD Bar (Панель FBD)	–	–	Отображает или скрывает панель инструментов **FBD**.
SFC Bar (Панель SFC)	–	–	Отображает или скрывает панель инструментов **SFC**.
LD Bar (Панель LD)	–	–	Отображает или скрывает панель инструментов **LD**.
Network (Сеть)	–	–	Отображает или скрывает панель инструментов **Network (Сеть)**.
Configuration (Конфигурирование)	–	–	Отображает или скрывает панель **Configuration (Конфигурирование)**.
HMI Page (Страница ЧМИ)	–	–	Отображает или скрывает панель **HMI Page (Страница ЧМИ)**.
HMI Project (Проект ЧМИ)	–	–	Отображает или скрывает панель **HMI Project (Проект ЧМИ)**.
HMI Profiles (Профили ЧМИ)	–	–	Отображает или скрывает панель **HMI Profiles (Профили ЧМИ)**.
Commissioning (Ввод в эксплуатацию)	–	–	Отображает или скрывает панель **Commissioning (Ввод в эксплуатацию)**.
Окна инструментов	–	–	См. раздел «Управление окнами инструментов» *(стр. 44)*.
Local variables (Локальные переменные)	–	–	Отображает или скрывает окно **Local variables (Локальные переменные)**.
Project (Проект)	–	–	Отображает или скрывает окно **Project (Проект)**.
Watch (Контрольные значения)	–	–	Отображает или скрывает окно **Watch (Контрольные значения)**.
Properties Window (Окно свойств)	–	–	Отображает или скрывает окно **Properties Window (Окно свойств)**.
Oscilloscope (Осциллограф)	–	–	Отображает или скрывает окно **Oscilloscope (Осциллограф)**.
PLC run-time status (Статус работы ПЛК)	–	–	Отображает или скрывает панель **PLC run-time status (Статус работы ПЛК)**.
Operators and blocks (Операторы и блоки)	–	–	Отображает или скрывает окно **Operators and blocks (Операторы и блоки)**.
Дерево библиотеки (Дерево библиотеки)	–	–	Отображает или скрывает окно **Library Tree (Дерево библиотеки)**.
Output (Выход)	–	–	Отображает или скрывает окно **Output (Вывод)**.
Cross Reference (Перекрестные ссылки)	–	–	Отображает или скрывает окно **Cross Reference (Перекрестные ссылки)**.
Resources (Ресурсы)	–	–	Отображает или скрывает окно **Resources (Ресурсы)**.
Catalog (Каталог)	–	–	Отображает или скрывает окно **Catalog (Каталог)**.
HMI Project (Проект ЧМИ)	–	–	Отображает или скрывает окно **HMI Project (Проект ЧМИ)**.
HMI Properties (Свойства ЧМИ)	–	–	Отображает или скрывает окно **HMI Properties (Свойства ЧМИ)**.
HMI Actions (Действия ЧМИ)	–	–	Отображает или скрывает окно **HMI Actions (Действия ЧМИ)**.
HMI Vars and Parameters (Переменные и параметры ЧМИ)	–	–	Отображает или скрывает окно **HMI Vars and Parameters (Переменные и параметры ЧМИ)**.
HMI Templates (Шаблоны ЧМИ)	–	–	Отображает или скрывает окно **HMI Templates (Шаблоны ЧМИ)**.
Commissioning (Ввод в эксплуатацию)	–	–	Отображает или скрывает окно **Commissioning (Ввод в эксплуатацию)**.

Команда	Значок	Клавиша	Описание
Full screen (Полный экран)		Ctrl+U	Разворачивает текущее активное окно документа на весь экран *(см. стр. 46)*. (Нажмите **Esc** для выхода из этого режима.)
Grid (Сетка)	–	–	Отображает или скрывает фоновую пунктирную сетку в графических редакторах исходного кода.
Show comments for objects (Показывать комментарии к объектам)	–	–	Отображает или скрывает комментарии для отдельных объектов, а не только для сетей. Только для редактора **LD**.

Меню Project (Проект)

Команда	Значок	Клавиша	Описание
New object (Новый объект)	–	–	
New program (New program)	–	–	Создает новую программу. Появляется диалоговое окно для указания свойств новой программы.
New function block (Новый функциональный блок)	–	–	Создает новый функциональный блок. Появляется диалоговое окно для указания свойств нового функционального блока.
New function (Новая функция)	–	–	Создает новую функцию. Появляется диалоговое окно для указания свойств новой функции.
New variable (Новая переменная)	–	–	
Automatic (Автоматическая)	–	–	Создает новую автоматическую переменную. Появляется диалоговое окно для указания свойств новых переменных.
Mapped variable (Сопоставленная переменная)	–	–	Создает новую сопоставленную переменную. Появляется диалоговое окно для указания свойств новых переменных.
Constant (Константа)	–	–	Создает новую константу. Появляется диалоговое окно для указания свойств новой константы.
Retain (Сохранить)	–	–	Создает новую сохраняемую переменную. Появляется диалоговое окно для указания свойств новой переменной.
New definition (Создать определение)	–	–	
Enumeration (Перечисление)	–	–	Создание нового перечисления.
Interface (Интерфейс)	–	–	Определение абстрактного интерфейса, который могут реализовывать функциональные блоки.
Structure (Структура)	–	–	Создание новой структуры.
Subrange (Поддиапазон)	–	–	Создание нового поддиапазона.
TypeDef (Определение типа)	–	–	Создание нового определения типа.
Copy object (Копировать объект)	–	–	Копирует объект, выделенный в данный момент в **рабочем пространстве.**
Paste object (Вставить объект)	–	–	Вставляет ранее скопированный объект.
Duplicate object (Дублировать объект)	–	–	Дублирует объект, выделенный в данный момент в **рабочем пространстве,** и предлагает указать имя копии.
Delete object (Удалить объект)	–	Delete	Удаляет текущий выбранный объект.

Команда	Значок	Клавиша	Описание
View PLC Object properties (Просмотр свойств объекта ПЛК)	–	Alt+Enter	Отображает свойства и описание выбранного объекта.
Object Browser (Обозреватель объектов)		–	Открывает **обозреватель объектов**, который позволяет осуществлять навигацию между объектами.
Compile (Компилировать)		F7	Запускает компилятор.
Recompile all (Перекомпилировать все)	–	Ctrl+ Alt+F7	Перекомпилирует проект.
Generate redistributable source module (RSM) (Сгенерировать распространяемый модуль источника [RSM])	–	–	Генерирует файл RSM.
Import objects (Импорт объектов)	–	–	Позволяет импортировать объект из библиотеки.
Export objects to library (Экспорт объектов в библиотеку)	–	–	Позволяет экспортировать объект в библиотеку.
Library manager (Диспетчер библиотек)		–	Открывает диспетчер библиотек.
Refresh all libraries (Обновить все библиотеки)		–	Перезагружает библиотеки, связанные с проектом.
Macros (Макросы)	–	–	
New macro (Новый макрос)	–	–	Создает новый макрос. Появляется диалоговое окно для указания свойств нового макроса.
Copy macro (Копировать макрос)	–	–	Создает копию выбранного макроса.
Delete macro (Удалить макрос)	–	–	Удаляет выбранный макрос.
Properties (Свойства)	–	–	Отображает свойства выбранного макроса.
Select target… (Выбрать целевое устройство…)	–	–	Позволяет выбрать новое целевое устройство для проекта.
Refresh current target (Обновить текущее целевое устройство)	–	–	Позволяет обновить целевой файл для той же версии целевого устройства.
Options… (Параметры…)	–	–	Открывает диалоговое окно параметров проекта.

Меню On-line (Онлайн)

Команда	Значок	Клавиша	Описание
Set up communication… (Настроить связь…)	–	–	Позволяет настроить параметры соединения с целевым устройством *(см. стр. 280)*.
Connect (Подключить)		–	Запускает обмен данными с устройством *(см. стр. 280)*.
Download code (Загрузить код)		F5	Загрузка конфигурации и приложения ПЛК для проекта на устройство *(см. стр. 289)*.
Download options… (Параметры загрузки…)	–	–	Позволяет настроить свойства исходного кода, загруженного на целевое устройство.

Команда	Значок	Клавиша	Описание
Force image upload (Принудительная загрузка образа)	–	–	Позволяет загрузить файл образа, если целевое устройство подключено.
Force debug symbols upload (Принудительная загрузка отладочных символов)	–	–	Позволяет загрузить файл отладочных символов, если целевое устройство подключено.
Halt (Остановка)		–	Останавливает работу ПЛК.
Cold restart (Холодный перезапуск)		–	Перезапускает работу ПЛК, при этом сбрасываются как сохраняемые, так и несохраняемые переменные.
Warm restart (Теплый перезапуск)		–	Перезапускает работу ПЛК, при этом сбрасываются несохраняемые переменные.
Hot restart (Горячий перезапуск)		–	Перезапускает работу ПЛК без сброса переменных.
Reboot target (Перезапуск целевого устройства)		–	Перезапускает целевое устройство.
Read all logs again (Повторно считать все журналы)	–	–	Удаленно перезагружает журналы с целевого устройства.

Меню Debug (Отладка)

Команда	Значок	Клавиша	Описание
Simulation mode (Режим моделирования)		–	Открытие/закрытие интегрированной среды моделирования.
Start/Stop watch value (Запуск/остановка контрольного значения)		–	Запускает или останавливает (переключает) оценку символов, добавленных в окно контрольных значений.
Add symbol to watch (Добавить символ в окно контрольных значений)	–	F 8	Добавляет символ в окно **Watch (Контрольные значения)**.
Insert new item into watch (Вставить новый элемент в окно контрольных значений)		Shift+F8	Вставляет новый элемент в окно **Watch (Контрольные значения)**.
Add symbol to debug (Добавить символ в окно отладки)	–	F10	Добавляет символ в окно **Debug (Отладка)**.
Inserts new item into a debug window (Вставить новый элемент в окно отладки)	–	Shift+F10	Вставляет новый элемент в окно **Debug (Отладка)**.
Quick watch symbol (Быстрый просмотр символа)	–	F11	Просмотр и принудительная активация значения выбранного символа.
Live debug mode (Режим отладки в режиме реального времени)		–	Если режим отладки работает, запускает или останавливает (переключает) режим отладки в режиме реального времени.
Add/remove text trigger (Добавить/удалить текстовый триггер)		F 9	Добавляет/удаляет текстовый триггер.
Add/remove graphic trigger (Добавить/удалить графический триггер)		Shift+F9	Добавляет/удаляет графический триггер.

Команда	Значок	Клавиша	Описание
Remove all triggers (Удалить все триггеры)		Ctrl+Shift+F9	Удаляет активные триггеры.
Trigger list (Список триггеров)		Ctrl+I	Выводит список активных триггеров.
Run (Выполнение)		–	Перезапускает программу после срабатывания точки останова.
Add/Remove breakpoint (Добавить/удалить точку останова)		F12	Добавляет или удаляет точку останова.
Remove all breakpoints (Удалить все точки останова)		–	Удаляет активные точки останова.
Breakpoint list (Список точек останова)		–	Выводит список активных точек останова.
Change current instance (Изменить текущий экземпляр)	–	–	Изменяет экземпляр текущего функционального блока (режим отладки в режиме реального времени).

Меню Variables (Переменные)

Команда	Значок	Клавиша	Описание
Add (Добавить)	–	–	
Automatic (Автоматически)	–	–	Создает новую автоматическую переменную. Появляется диалоговое окно для указания новой переменной.
Mapped variable (Сопоставленная переменная)	–	Ctrl+ Shift+M	Создает новую сопоставленную переменную. Появляется диалоговое окно для указания новой переменной.
Constant (Константа)	–	–	Создает новую константу. Появляется диалоговое окно для указания новой константы.
Retain (Сохранить)	–	–	Создает новую сохраняемую переменную. Появляется диалоговое окно для указания новой переменной.
Insert (Вставить)	–	Ctrl+ Shift+Ins	Добавляет новую строку в сетку в активном редакторе.
Delete (Удалить)	–	Delete	Удаляет переменную в выбранной строке текущей активной таблицы.
Create multiple… (Создать несколько…)	–	–	Позволяет создать набор из нескольких переменных.
Group (Группа)	–	–	Открывает диалоговое окно, позволяющее создавать и удалять группы переменных.

Меню Window (Окно)

Команда	Значок	Клавиша	Описание
Cascade (Каскад)	–	–	Отображает открытые документы в виде каскада так, чтобы они полностью перекрывались, за исключением заголовка.
Tile (Плитка)	–	–	Область редакторов ПЛК разделена на участки одинакового размера, в зависимости от количества открытых в данный момент документов. Каждый участок автоматически присваивается одному из таких документов.
Arrange Icons (Упорядочить значки)	–	–	Располагает значки свернутых документов в левом нижнем углу рабочей области редактора ПЛК.
Close all (Закрыть все)	–	–	Закрывает открытые документы.
Windows… (Окна…)	–	**Alt+W**	Открывает обозреватель **Windows List (Список окон)**.

Меню Tools (Инструменты)

Команда	Значок	Клавиша	Описание
Custom Tools (Пользовательские инструменты)	–	–	Отображает до 16 пользовательских команд *(см. стр. 48)*.

Меню Help (Справка)

Команда	Значок	Клавиша	Описание
Index (Указатель)	–	–	Выводит список **Help keywords (Ключевые слова справки)** и открывает соответствующий раздел.
Context (Контекст)	–	**F 1**	Контекстно-зависимая справка. Открывает раздел, посвященный открытому в настоящее время окну.
Registration (Регистрация)	–	–	Регистрация ПО *(см. стр. 26)*.
About… (О программе…)	–	–	Сведения об авторах и версии ПО.

Панели инструментов

Введение

Панель инструментов расположена в верхней части окна EcoStruxure Machine Expert - HVAC и обеспечивает доступ к часто используемым функциям.

Описание команд отображается в левом нижнем углу главного окна при наведении курсора мыши на значок команды.

Главная панель инструментов

Главная панель инструментов содержит следующие кнопки.

Значок	Описание	Сочетание клавиш
	Отменить Щелкните один раз, чтобы отменить последнее действие в редакторе программы. Щелкните стрелку вниз и выберите действие из списка, чтобы отменить все действия до выбранного действия включительно. Вы можете отменить до десяти действий.	Ctrl+Z
	Вернуть Нажмите один раз, чтобы отменить последнее действие отмены. Щелкните стрелку вниз и выберите действие из списка, чтобы вернуть все действия до выбранного действия включительно. Вы можете вернуть до десяти действий.	Ctrl+Y
	Вырезать	Ctrl+X
	Копировать	Ctrl+C
	Вставить	Ctrl+V
	Найти	Ctrl+F
	Найти далее	F3
	Найти в проекте	Ctrl+Shift+F
	Перейти к символу	Shift+F12
	Печать Распечатать содержимое текущего окна редактора.	Ctrl+P

Значок	Описание	Сочетание клавиш
	Предпросмотр перед печатью	–
	Рабочее пространство	Ctrl+W
	Выход	Ctrl+R
	Операторы и блоки	Ctrl+L
	Отображает или скрывает панель Library Tree (Дерево библиотеки)	
	Контрольные значения	Ctrl+T
	Осциллограф	Ctrl+K
	Панель статуса работы ПЛК	–
	Окно перекрестных ссылок	–
	Окно свойств объекта	–
	Полный экран	Ctrl+U

Панель инструментов проекта

Панель инструментов проекта содержит следующие кнопки.

Значок	Описание	Сочетание клавиш
	Компиляция	F7
	Режим моделирования	–
	Подключение к целевому устройству	–
	Загрузка кода	F5

Значок	Описание	Сочетание клавиш
	Остановка	–
	Холодный перезапуск	–
	Теплый перезапуск	–
	Горячий перезапуск	–
	Перезапуск целевого устройства	–
	Обозреватель объектов	–
	Диспетчер библиотек	–
	Обновить все библиотеки	–
	Свойства объекта	–
	Вставить запись	–
	Удалить запись	–

Панель инструментов Debug (Отладка)

Панель инструментов отладки содержит следующие кнопки.

Значок	Описание	Сочетание клавиш
	Режим отладки в режиме реального времени	–
	Добавить/удалить триггер	F9
	Добавить/удалить графический триггер	Shift+F9
	Удалить все триггеры	Ctrl+Shift+F9

Значок	Описание	Сочетание клавиш
	Список триггеров	Ctrl+I
	Добавить/удалить точку останова	F12
	Удалить все точки останова	–
	Выполнение	–
	Шаг	–
	Список точек останова	–

Панель инструментов FBD

Панель инструментов FBD содержит следующие кнопки.

Значок	Описание	Сочетание клавиш
	Переместить/вставить	–
	Подключение	–
	Контрольные значения	–
	Новый функциональный блок	–
	Переменная	–
	Константа	–
	Выражение	–
	Возврат	–
	Прыжок	–

Значок	Описание	Сочетание клавиш
	Комментарий	–
	Увеличить количество выводов	Ctrl++
	Уменьшить количество выводов	Ctrl+-
	Добавить/включить входные/выходные выводы	–
	Свойства FBD	–
	Просмотр исходника	–

Панель инструментов SFC

Панель инструментов **SFC** содержит следующие кнопки.

Значок	Описание	Сочетание клавиш
	Новый шаг	–
	Новый переход	–
	Новый прыжок	–
	Добавить шаг и переход над выбранным шагом	–
	Добавить шаг и переход под выбранным шагом	–
	Добавить вывод в позицию расхождения	–
	Удалить вывод из позиции расхождения	–
	Добавить вывод в позицию схождения	–
	Удалить вывод из позиции схождения	–
	Добавить вывод к одновременно расходящемуся переходу	–

Значок	Описание	Сочетание клавиш
	Удалить вывод из одновременно расходящегося перехода	–
	Добавить вывод к одновременно сходящемуся переходу	–
	Удалить вывод из одновременно сходящегося перехода	–
	Удалить пробел перед самым правым выводом	–
	Добавить пробел перед самым правым выводом	–
	Переместить переход вверх	–
	Переместить переход вниз	–
	Новое действие	–
	Новый код перехода	–

Панель инструментов LD

Панель инструментов LD содержит следующие кнопки.

Значок	Описание	Сочетание клавиш
	Параллельное соединение перед	–
	Параллельное соединение после	Shift+P
	Последовательное соединение перед	–
	Последовательное соединение после	Shift+C
	Катушка	Shift+O
	Открытый объект	O

Значок	Описание	Сочетание клавиш
	Отрицаемый объект	C
	Положительный объект	P
	Отрицательный объект	N
	Сбросить объект	R
	Установить объект	S
	Задать выходную линию	–
	Новое ответвление	–

Панель инструментов Network (Сеть)

Панель инструментов **Network (Сеть)** содержит следующие кнопки.

Значок	Описание	Сочетание клавиш
	Вставить в самом верху	–
	Вставить в самом низу	–
	Вставить после	–
	Вставить перед	–
	Показать сетку	–
	Автоматическое соединение	–

Строка состояния

Обзор

Строка состояния расположена в нижней правой части окна EcoStruxure Machine Expert - HVAC. Она служит для отображения состояния связи и статуса приложения, выполняемого в данный момент на целевом устройстве.

EDIT MODE	**SOURCE OK**	**CONNECTED**

Дополнительную информацию см. в следующих разделах:

- Режим редактирования и отладки *(см. стр. 321)*
- Статус соединения *(см. стр. 287)*
- Статус приложения *(см. стр. 288)*

Глава 8
Параметры проекта

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
8.1	Параметры проекта	164
8.2	Работа с библиотеками	174

Раздел 8.1
Параметры проекта

Общая информация

Вы можете редактировать основные параметры проекта, выбрав **Project (Проект)** → **Options… (Параметры…)**.

Содержание этого раздела

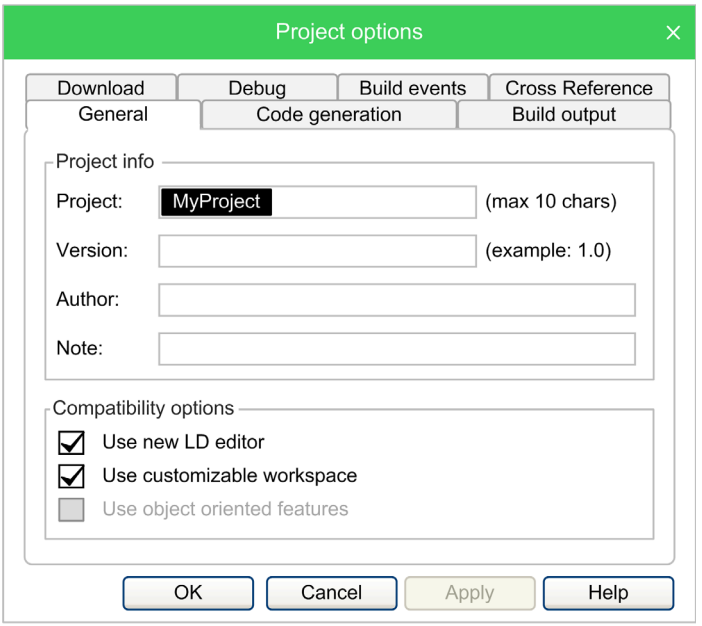
В этом разделе описаны следующие темы.

Тема	Страница
Вкладка General (Общее)	165
Вкладка Code Generation (Генерирование кода)	166
Вкладка Build Output (Вывод сборки)	168
Вкладка Debug (Отладка)	170
Вкладка Build Events (События сборки)	171
Вкладка Cross Reference (Перекрестные ссылки)	172

Вкладка General (Общее)

Обзор

Вкладка **General (Общее)** окна **Project options (Параметры проекта)**:



Здесь можно задать общую информацию о проекте:

- Название проекта
- Версия проекта
- Имя автора проекта
- Примечание к проекту

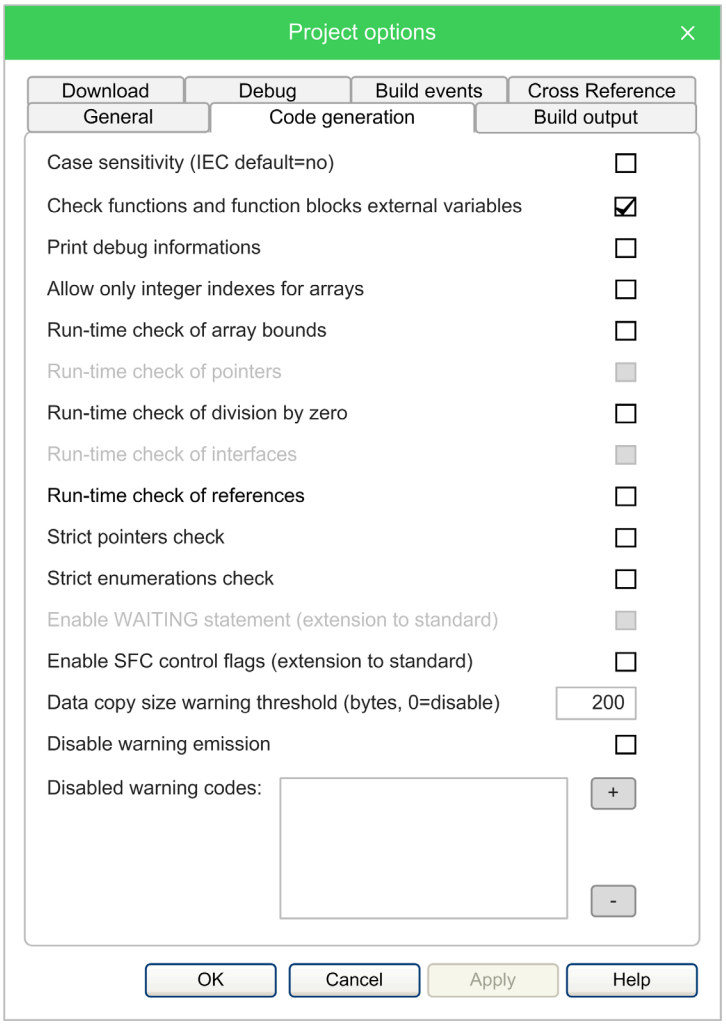
Кроме того, вы можете настроить параметры совместимости:

- **Use new LD editor (Использовать новый редактор LD)**: новый редактор релейных схем проще в использовании, облегчает рутинные операции и позволяет работать над схемой быстрее и эффективнее. По умолчанию эта опция активна. Для получения дополнительной информации см. раздел «Редактор релейных схем» *(стр. 236)*.
- **Use customizable workspace (Использовать настраиваемое рабочее пространство)**: позволяет управлять своим деревом проекта и более эффективно использовать рабочее пространство. По умолчанию эта опция активна. Для получения дополнительной информации см. раздел «Пользовательское рабочее пространство проекта» *(стр. 219)*.
- **Use object oriented features (Использовать возможности объектно-ориентированного программирования)**: не поддерживается.

Вкладка Code Generation (Генерирование кода)

Обзор

Вкладка **Code generation (Генерирование кода)** окна **Project options (Параметры проекта)**:



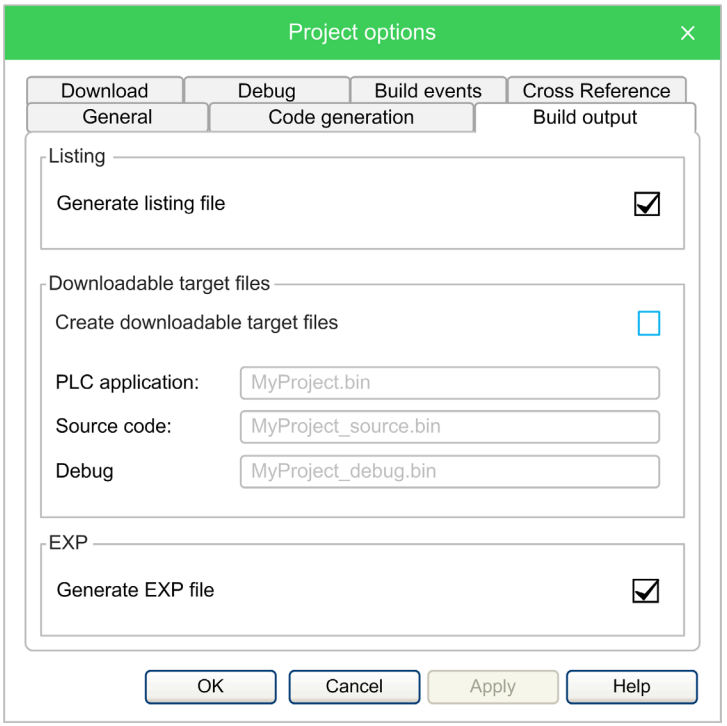
Вы можете редактировать некоторые свойства, относящиеся к генерированию кода:

- **Case sensitivity (Чувствительность к регистру)**: вы можете сделать проект чувствительным к регистру, установив этот флажок. **ПРИМЕЧАНИЕ.** По умолчанию эта опция неактивна.
- **Check function and function block external variables (Проверка внешних переменных функции и функционального блока)**: если эта опция отключена, то все функции и функциональные блоки могут обращаться к глобальным переменным, не объявляя их в качестве внешних переменных. **ПРИМЕЧАНИЕ.** По умолчанию эта опция включена в соответствии со стандартом МЭК 61131-3 (ГОСТ Р МЭК 61131-3).
- **Print debug information (Печать отладочной информации)**: отображает в окне вывода важную отладочную информацию.
- **Allow only integer indexes for arrays (Разрешать только целочисленные индексы для массивов)**: если эта опция выбрана, вы не можете использовать BYTE, WORD или DWORD в качестве индексов массива.
- **Run-time check of array bounds (Проверка границ массива в процессе выполнения)**: если эта опция выбрана, то вставляется дополнительный код для проверки во время выполнения того, что индексы массивов не выходят за допустимые границы. Эта опция доступна в зависимости от целевого устройства.
- **Run-time check of pointers (Проверка указателей в процессе выполнения)**: если эта опция выбрана, то указатели перед использованием проверяются на корректность, для чего на целевом устройстве вызывается пользовательская функция **checkptr**. Эта опция доступна в зависимости от целевого устройства.
- **Run-time check of division by zero (Проверка деления на ноль в процессе выполнения)**: если эта опция выбрана, то вставляется дополнительный код для проверки того, что во время выполнения программы над массивами не выполняется операция деления на ноль. Эта опция доступна в зависимости от целевого устройства.
- **Run-time check of interfaces (Проверка интерфейсов в процессе выполнения)**: не поддерживается.
- **Run-time check of references (Проверка ссылок в процессе выполнения)**: не поддерживается.
- **Strict pointers check (Строгая проверка указателей)**: если эта опция выбрана, то смешивать различные типы указателей и целочисленные значения невозможно.
- **Strict enumerations check (Строгая проверка перечислений)**: если эта опция выбрана, то смешивать перечисляемые переменные и целочисленные значения невозможно.
- **Enable WAITING statement (extension to standard) (Включить конструкцию WAITING [расширение стандарта])**: если эта опция выбрана, то в качестве расширения МЭК 61131-3 (ГОСТ Р МЭК 61131-3) для языка ST добавляется конструкция WAITING. Для получения дополнительной информации см. раздел «Инструкция ожидания» *(стр. 444)*.
- **Enable SFC control flags (extension to standard) (Включить управляющие флаги SFC [расширение стандарта])**: если эта опция отмечена, то для программного модуля SFC активируются флаги HOLD и RESET.
- **Data copy size warning threshold (bytes, 0=disable) (Порог предупреждения о размере копии данных [байты, 0=отключено])**: при копировании массивов или структур, размеры которых превышают указанный порог, выводится предупреждение о возможном снижении производительности ПЛК. Если порог установлен на 0, сообщения не выводятся.
- **Disable warning emission (Отключить вывод предупреждений)**: если эта опция выбрана, то сообщения не отображаются в окне вывода.
- **Disable warning codes (Отключить коды предупреждения)**: если эта опция выбрана, некоторые заданные сообщения не отображаются в окне вывода.

Вкладка Build Output (Вывод сборки)

Обзор

Вкладка **Build output (Вывод сборки)** окна **Project options (Параметры проекта)**:



Здесь можно редактировать некоторые важные свойства выходных файлов, генерируемых при компиляции.

Раздел Listing (Листинг):

- **Generate listing file (Генерировать файл листинга)**: если эта опция выбрана, компилятор генерирует файл листинга с именем **projectname.lst.**

Раздел Downloadable target files (Доступные для загрузки целевые файлы):

- **Create downloadable target files (Создать доступные для загрузки целевые файлы)**: если эта опция выбрана, компилятор генерирует бинарные файлы, которые могут быть загружены на целевое устройство. Вы можете указать собственные имена файлов или использовать имена по умолчанию.

Допускаются только имена файлов, действительные в ОС Windows.

- **PLC application (Приложение ПЛК)** (активно только при выбранной опции **Create downloadable target files [Создать доступные для загрузки целевые файлы]**): в этом поле задается имя бинарного файла приложения ПЛК. По умолчанию используется имя **projectname.bin**.
- **Source code (Исходный код)** (активно только при выбранной опции **Create downloadable target files [Создать доступные для загрузки целевые файлы]**): в данном поле задается имя бинарного файла с исходным кодом. По умолчанию используется имя **projectname._source.bin**.
- **Debug (Отладка)** (активно только при выбранной опции **Create downloadable target files [Создать доступные для загрузки целевые файлы]**): в этом поле задается имя бинарного файла с отладочными символами. По умолчанию используется имя **projectname._debug.bin**.

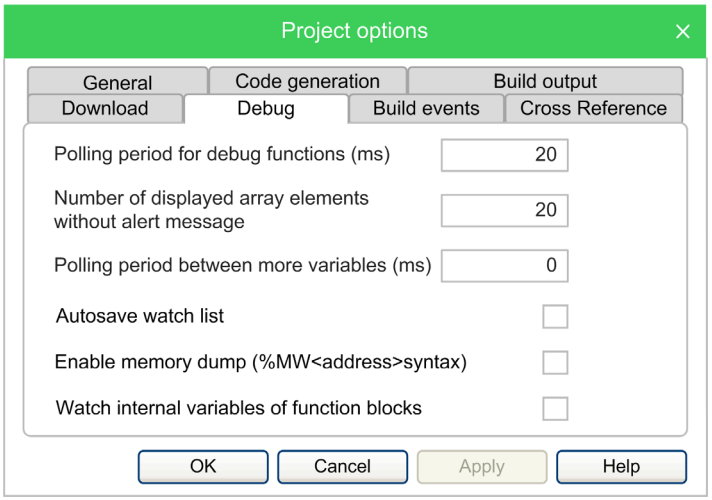
Раздел Generate EXP file (Генерировать файл EXP):

- **Generate EXP file (Генерировать файл EXP)**: если эта опция выбрана, компилятор генерирует файл EXP с именем **projectname.exp**.

Вкладка Debug (Отладка)

Обзор

Вкладка **Debug (Отладка)** окна **Project options (Параметры проекта)**:



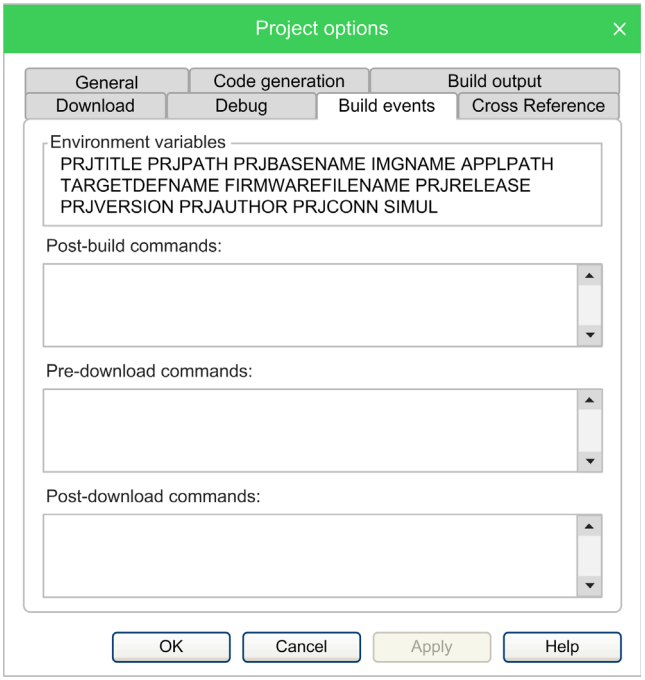
Здесь можно отредактировать некоторые важные свойства отладчика:

- **Polling period for debug function (Период опроса для функции отладки)**: устанавливает активный интервал выборки статуса отладки в миллисекундах.
- **Number of displayed array elements without alert message (Количество отображаемых элементов массива без предупреждающего сообщения)**: определяет максимальное количество элементов массива, добавляемых в окно контрольных значений без предупреждения.
- **Polling period between more variables (Период опроса для нескольких переменных)**: устанавливает задержку между выборкой переменных в миллисекундах.
- **Autosave watch list (Автосохранение списка контрольных значений)**: если опция выбрана, то при закрытии проекта статус списка контрольных значений сохраняется в файле.
- **Enable memory dump (Включить дамп памяти)**: включает функцию дампа памяти для расширенной отладки.
- **Watch internal variables of function blocks (Контроль внутренних переменных функциональных блоков)**: позволяет просматривать внутренние переменные класса VAR.

Вкладка Build Events (События сборки)

Обзор

Вкладка **Build events (События сборки)** окна **Project options (Параметры проекта)**:



Здесь можно указать команды, которые выполняются до начала сборки или после ее завершения. Вы также можете использовать набор определенных переменных окружения, перечисленных в верхней части окна.

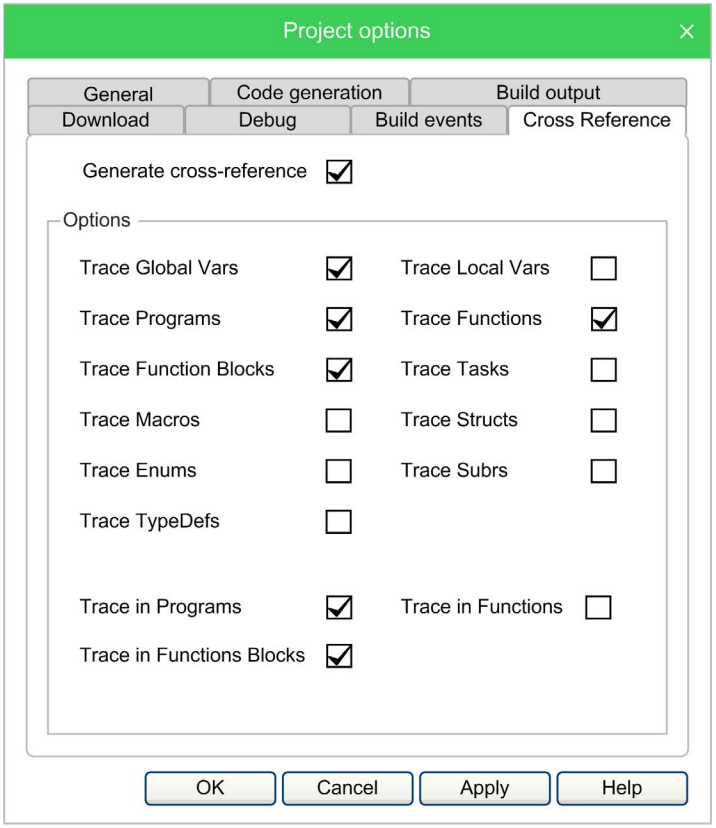
Переменные окружения:

- **PRJTITLE**: название проекта.
- **PRJPATH**: папка проекта.
- **PRJBASENAME**: полный путь к проекту без расширения.
- **PRJFULLNAME**: полный путь к проекту.
- **IMGNAME**: имя файла образа **.imgx**.
- **TARGETDEFNAME**: название целевого устройства проекта.
- **PRJRELEASE**: имя проекта, указанное на вкладке **General (Общее)** окна **Project options (Параметры проекта)**.
- **PRJVERSION**: версия проекта, указанная на вкладке **General (Общее)** окна **Project options (Параметры проекта)**.
- **PRJAUTHOR**: автор проекта, указанный на вкладке **General (Общее)** окна **Project options (Параметры проекта)**.
- **PRJCONN**: текущие настройки связи.
- **APPLPATH**: полный путь к приложению.
- **SIMUL**: равно 1, если активен режим моделирования, в противном случае — 0.

Вкладка Cross Reference (Перекрестные ссылки)

Обзор

Вкладка **Cross Reference (Перекрестные ссылки)** окна **Project options (Параметры проекта)**:



Здесь можно активировать функцию Generate cross-reference (Генерировать перекрестные ссылки) и установить соответствующие параметры.

Можно задать следующие параметры отслеживания перекрестных ссылок:

- **Global Vars** (Глобальные переменные)
- **Local Vars** (Локальные переменные)
- **Programs** (Программы)
- **Functions** (Функции)
- **Function Blocks** (Функциональные блоки)
- **Tasks** (Задачи)
- **Macros** (Макросы)
- **Structs** (Структуры)

- **Enums** (Перечисления)
- **Subrs** (Поддиапазоны)
- **TypeDefs** (Определения типа) для создания псевдонима для других типов данных

Параметры отслеживания перекрестных ссылок можно настроить в следующих элементах:

- **Programs** (Программы)
- **Functions** (Функции)
- **Function Blocks** (Функциональные блоки)

Раздел 8.2
Работа с библиотеками

Общая информация

Библиотеки являются мощным инструментом для совместного использования объектов в разных проектах. Библиотеки хранятся в специальном исходном файле с расширением **.pll**.

Содержание этого раздела

В этом разделе описаны следующие темы.

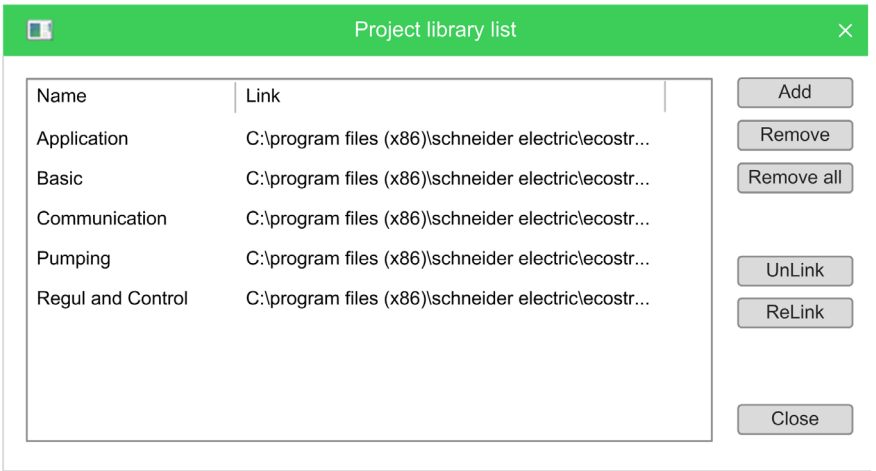
Тема	Страница
Диспетчер библиотек	175
Экспорт в библиотеку	177
Импорт из библиотеки или другого источника	178
Обновление имеющихся библиотек	180

Диспетчер библиотек

Обзор

Диспетчер библиотек выводит список библиотек, включенных в проект в настоящее время. Он также позволяет включать или удалять библиотеки.

Чтобы получить доступ к диспетчеру библиотек, выберите **Project (Проект)** → **Library manager (Диспетчер библиотек)**.



Подключение библиотеки

Ниже приводится процедура подключения библиотеки к проекту, в результате чего объекты библиотеки становятся доступными для текущего проекта.

Подключение библиотеки означает, что в текущий проект добавляется ссылка на файл библиотеки **.pll** и создается локальная копия библиотеки. Вы не можете редактировать элементы подключенной библиотеки, в отличие от импортированных объектов.

Чтобы скопировать или переместить проект, содержащий одну или несколько библиотек, убедитесь, что ссылки на эти библиотеки все еще действительны в новом месте.

Шаг	Действие
1	Нажмите **Project (Проект)** → **Library manager (Диспетчер библиотек)**, чтобы открыть диалоговое окно **Library manager (Диспетчер библиотек)**.
2	Нажмите кнопку **Add (Добавить)**. Откроется диалоговое окно проводника, позволяющее выбрать файл библиотеки **.pll**, которую необходимо открыть.
3	Когда вы найдете файл **.pll**, откройте его двойным щелчком или нажатием кнопки **Open (Открыть)**. Имя библиотеки и ее абсолютный путь теперь отображаются в новой строке внизу списка.
4	Повторите шаги 1, 2 и 3 для библиотек, которые необходимо подключить.
5	По завершении процедуры подключения библиотек нажмите кнопку **Close (Закрыть)**.

Удаление библиотеки

Удаление библиотеки не приводит к удалению самой библиотеки, а удаляет лишь ссылку на нее в проекте.

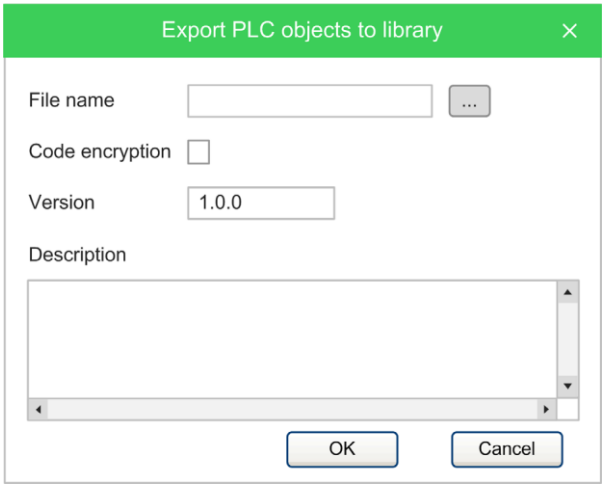
Чтобы удалить подключенную библиотеку из текущего проекта, выполните следующие действия.

Шаг	Действие
1	Выберите **Project (Проект)** → **Library manager (Диспетчер библиотек)** в главном окне **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**. Откроется диалоговое окно **Library manager (Диспетчер библиотек)**.
2	Щелкните имя библиотеки, которую нужно удалить. Кнопка **Remove (Удалить)** станет активной.
3	Нажмите кнопку **Remove (Удалить)**. Ссылка на выбранную библиотеку будет удалена из списка **Project library (Библиотека проекта)**.
4	Повторите эту процедуру для всех библиотек, которые необходимо удалить. Также можно удалить все библиотеки нажатием кнопки **Remove all (Удалить все)**.
5	По завершении процедуры удаления библиотек нажмите кнопку **Close (Закрыть)**.

Экспорт в библиотеку

Обзор

Вы можете экспортировать любой объект из открытого в данный момент проекта в библиотеку, чтобы сделать этот объект доступным для других проектов.
Чтобы экспортировать объекты в библиотеку, выполните следующие действия.

Шаг	Действие
1	Выберите в древовидной структуре вкладки проекта объект, который вы хотите экспортировать.
2	Нажмите **Project (Проект) → Export object to library (Экспорт объекта в библиотеку)**. Также можно щелкнуть объект правой кнопкой мыши и выбрать команду **Export object to library (Экспорт объекта в библиотеку)**. Откроется диалоговое окно: 
3	Укажите расположение файла **.pll** целевой библиотеки. Для этого: ● Введите полный путь в текстовое поле. **ПРИМЕЧАНИЕ.** Если ввести имя несуществующего файла **.pll**, EcoStruxure Machine Expert - HVAC создаст новую библиотеку. ● Нажмите кнопку ... **Browse (Обзор)**, чтобы открыть диалоговое окно проводника для выбора файла на диске или в сети.
4	Вы также можете зашифровать исходный код экспортируемого программного модуля в целях защиты вашей интеллектуальной собственности. Можно также изменить номер версии и добавить описание.
5	Нажмите **OK**, чтобы подтвердить операцию, или **Cancel (Отмена)**, чтобы ее завершить.

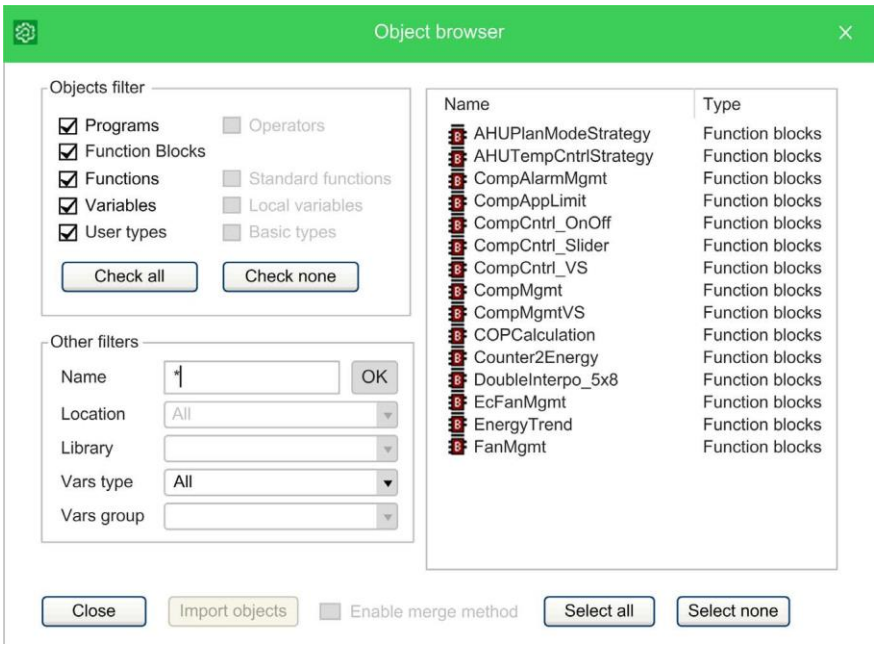
Импорт из библиотеки или другого источника

Обзор

Вы можете импортировать объект из библиотеки и использовать его в текущем проекте. При импорте объекта из библиотеки локальная копия объекта теряет ссылку на исходную библиотеку и теперь принадлежит исключительно текущему проекту. Вы можете редактировать импортированные объекты, в отличие от объектов подключенных библиотек.

Пример импорта

Чтобы импортировать объекты из библиотеки, выполните следующие действия.

Шаг	Действие
1	Нажмите **Project (Проект) → Import objects (Импорт объектов)**. Откроется диалоговое окно проводника, где вы сможете выбрать файл библиотеки **.pll**, которую хотите открыть.
2	Когда вы найдете файл **.pll** или **.plclib**, откройте его двойным щелчком или нажатием кнопки **Open (Открыть)**. Откроется диалоговое окно проводника библиотеки: 

Шаг	Действие
3	Выберите объекты, которые необходимо импортировать. Вы также можете выполнять простые запросы к объектам с помощью **Filters (Фильтры)**. Однако в настоящее время к библиотекам применяется только фильтр **Name (Имя)**. Чтобы воспользоваться им, введите имя нужного объекта, при необходимости используйте подстановочный символ *.
4	Выберите объекты, которые вы хотите импортировать, затем нажмите кнопку **Import objects (Импорт объектов)**.
5	По завершении импорта объектов нажмите **OK** или **Cancel (Отмена)**, чтобы закрыть обозреватель.

Отмена импорта из библиотеки

Импортируя объект в проект, вы создаете локальную копию этого объекта. Чтобы отменить импорт, необходимо удалить локальный объект.

Функция объединения

При импорте объектов в проект или вставке скопированной сопоставленной переменной вы можете столкнуться с ошибкой наложения адресов или дублирования имен.

Вы можете настроить поведение EcoStruxure Machine Expert - HVAC при возникновении таких проблем, установив соответствующие параметры среды *(см. стр. 49)*.

Ниже приведены возможные действия.

Поведение		Спрашивать	Автоматически	Брать из библиотеки	Не делать ничего
Поведение при конфликте имен	Если разные типы	✓	✓		✓
	Если одного типа, но не переменные	✓	✓	✓	
	Если обе переменные	✓	✓	✓	
Поведение при конфликте адресов	Если адреса накладываются	✓	✓	✓	
	Копировать/вставить сопоставленную переменную		✓		✓
Ask (Спрашивать): пользователь должен каждый раз принимать решение о том, какое действие необходимо выполнить (поведение по умолчанию). **Automatic (Автоматически):** EcoStruxure Machine Expert - HVAC автоматически генерирует действительное имя или адрес присваивает его импортируемому объекту. **Take from library (Брать из библиотеки):** имя или адрес берется из импортируемого объекта. **Do nothing (Ничего не делать):** имя или адрес объектов в проекте не изменяются.					

После импорта объектов EcoStruxure Machine Expert - HVAC генерирует файл журнала в папке проекта с подробной информацией.

Обновление имеющихся библиотек

Обзор

Если вы редактируете файл подключенной библиотеки, можно обновить его содержимое в проекте, не закрывая EcoStruxure Machine Expert - HVAC, следующим образом.

Шаг	Действие
1	Нажмите **Project (Проект) → Refresh all libraries (Обновить все библиотеки)**.
2	Если файл правильный, EcoStruxure Machine Expert - HVAC обновляет содержимое подключенной библиотеки и отображает в окне вывода сообщение об успешном завершении операции, в противном случае никаких изменений в имеющейся подключенной библиотеке не производится.

Глава 9
Управление элементами проекта

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
9.1	Обзор	182
9.2	Программные модули	183
9.3	Переменные	188
9.4	Задачи	199
9.5	Производные типы данных	202
9.6	Навигация по проекту	211
9.7	Пользовательское рабочее пространство проекта	219

Раздел 9.1
Обзор

Окно Project (Проект)

Обзор

В этой главе описывается процедура управления элементами, из которых состоит проект, а именно: программными модулями, задачами, производными типами данных и переменными.

Окно **Project (Проект)** позволяет:

- управлять кодом приложения;
- определять комплексные пользовательские переменные и управлять ими;
- управлять задачами.

Содержимое окна Project (Проект)

По умолчанию окно **Project (Проект)** содержит следующие элементы.

Элемент	Значок	Описание
Project (Проект)		Имя проекта.
main *(см. стр. 183)*.		Первоначальная программа.
Var1 *(см. стр. 195)*.		Локальная переменная.
Ungrouped_vars *(см. стр. 188)*.		Группа глобальных переменных.
Aux Variables (Вспом. переменные)		Общие глобальные ресурсы отображаются в окне **Project (Проект)**, но определяются в окне **Resources (Ресурсы)**. Здесь отображаются **параметры EEPROM**, **переменные состояния** и **сопоставление входов/выходов** (поскольку они генерируются автоматически).
Tasks (Задачи) *(см. стр. 199)*.		Создает задачи.

ПРИМЕЧАНИЕ. Содержание окна **Project (Проект)** зависит от выбранного устройства.

Раздел 9.2
Программные модули

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	183
Создание программы	184
Создание функционального блока/функции	185
Редактирование программных модулей	186
Шифрование/дешифрование исходного кода	186

Обзор

Описание

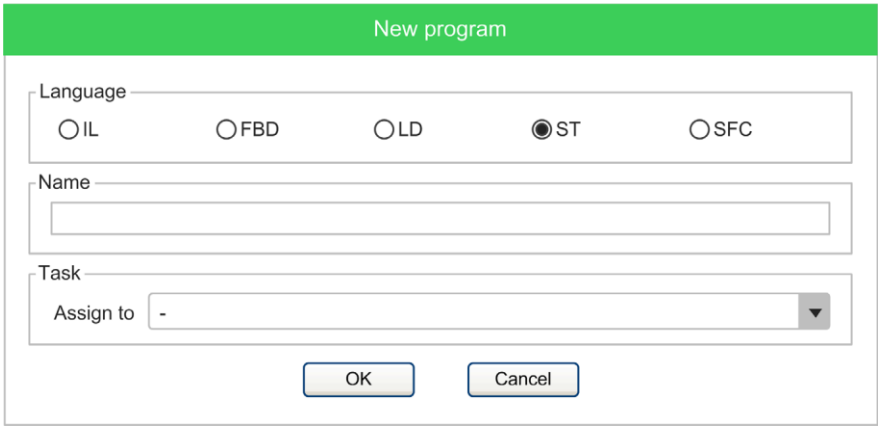
Программный модуль (Program Organization Unit, POU) — это программный блок типа «программа», «функция» или «функциональный блок».

В этом разделе рассказывается, как добавлять, редактировать и удалять программные модули в рамках проекта.

Создание программы

Описание

Чтобы создать программу, выполните следующие действия.

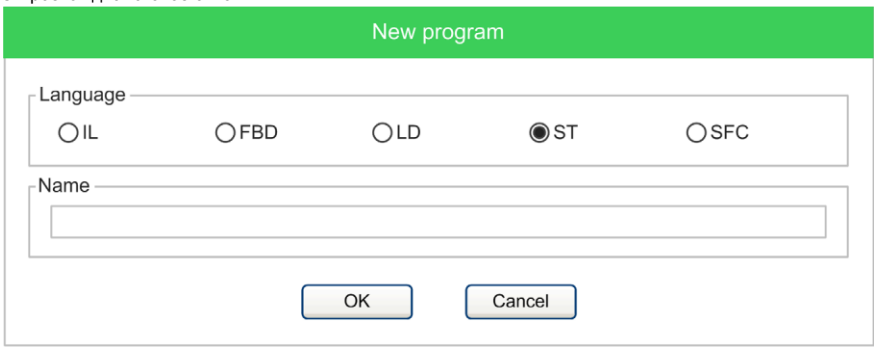
Шаг	Действие
1	Выберите целевое устройство в дереве окна **Project (Проект)**. Нажмите **Project (Проект)** → **New Object (Новый объект)** → **New program (Новая программа)**.
2	Откроется диалоговое окно:  Выберите нужный язык.
3	Введите имя.
4	При необходимости выберите задачу из списка, чтобы связать программу с выбранной задачей. **ПРИМЕЧАНИЕ.** Если на этом этапе вы не выберете задачу, под значком программы появится значок предупреждения, указывающий на то, что программа еще не ассоциирована с задачей. Процедура присвоения программы соответствующей задаче рассматривается в главе «Связывание программы с задачей» *(стр. 199)*.
5	Нажмите кнопку **OK** для подтверждения.

Кроме того, вы можете создать новый программный модуль с помощью контекстного меню, выбрав папку или корневой элемент проекта. Дополнительную информацию см. в разделе «Операции в пользовательском рабочем пространстве» *(см. стр. 222)*.

Создание функционального блока/функции

Описание

Чтобы создать функциональный блок/функцию, выполните следующие действия.

Шаг	Действие
1	Выберите целевое устройство в дереве окна **Project (Проект)**. Для функционального блока нажмите **Project (Проект)** → **New Object (Новый объект)** → **New function block (Новый функциональный блок)**. Для функции нажмите **Project (Проект)** → **New Object (Новый объект)** → **New function (Новая функция)**.
2	Откроется диалоговое окно:  Выберите нужный язык. **ПРИМЕЧАНИЕ.** Создание функций доступно для четырех языков программирования. Язык SFC не поддерживает функции.
3	Введите имя.
4	Нажмите кнопку **OK** для подтверждения.

Функция или функциональный блок — это (под)программа со входами и выходами:

- **Функция** требует **n** входов и один выход (**RESULT**) с тем же именем, что и функция. Локальная память функции инициализируется каждый раз при вызове функции.
  Функция используется внутри программы посредством передачи входных переменных.
- **Функциональный блок** требует **n** входов и **m** выходов. Локальная память каждого экземпляра функционального блока хранится между одним вызовом и следующим (статическая память).
  Функциональный блок используется внутри программы в качестве экземпляра так же, как и объявление переменной.

Каждую функцию или функциональный блок можно использовать внутри программы, перетаскивая значок в окно редактора программы.

Редактирование программных модулей

Обзор

Чтобы отредактировать программный модуль, откройте его, дважды щелкнув левой кнопкой мыши в дереве проекта. Откроется соответствующий редактор, который позволяет изменять исходный код программного модуля.

Изменение имени программного модуля

Выберите программный модуль в дереве проекта, затем щелкните его правой кнопкой мыши и выберите **Rename program (Переименовать программу)**, **Rename function block (Переименовать функциональный блок)** или **Rename function (Переименовать функцию)** в зависимости от программного модуля.

Дублирование программного модуля

Выберите программный модуль из дерева проекта, затем нажмите **Project (Проект)** → **Duplicate object (Дублировать объект)**. Введите имя нового дублированного программного модуля и подтвердите операцию.

Удаление программного модуля

Выберите программный модуль из дерева проекта, затем нажмите **Project (Проект)** → **Delete object (Удалить объект)**. Подтвердите операцию удаления программного модуля.

Шифрование/дешифрование исходного кода

Обзор

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** может шифровать программные модули и скрывать их исходный код, требуя ввода пароля.

Шифрование программного модуля

Выберите программный модуль из дерева проекта, щелкните правой кнопкой мыши и выберите в контекстном меню пункт **Crypt (Шифровать)**.
Дважды введите пароль и подтвердите операцию.

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** пометит в дереве проекта стандартный значок программного модуля специальным маркером, указывая на то, что программный модуль зашифрован.

Дешифрование программного модуля

Выберите программный модуль из дерева проекта, щелкните его правой кнопкой мыши и выберите **Decrypt (Дешифровать)** в контекстном меню.

Шифрование всех программных модулей

Выберите целевой объект из дерева проекта, щелкните его правой кнопкой мыши и выберите **Crypt all objects (Шифровать все объекты)** в контекстном меню. Все программные модули шифруются одним и тем же паролем.

Дешифрование всех программных модулей

Выберите целевой объект из дерева проекта, щелкните его правой кнопкой мыши и выберите **Decrypt all objects (Дешифровать все объекты)** в контекстном меню.

Раздел 9.3
Переменные

Обзор

В модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** существует два класса переменных: глобальные и локальные.

В этом разделе описываются процедуры добавления в проект, редактирования и удаления как глобальных, так и локальных переменных.

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Глобальные переменные	188
Локальные переменные	195
Создание нескольких переменных	198

Глобальные переменные

Описание

Глобальные переменные видны в любом программном модуле проекта, и на них можно ссылаться.

Классы глобальных переменных

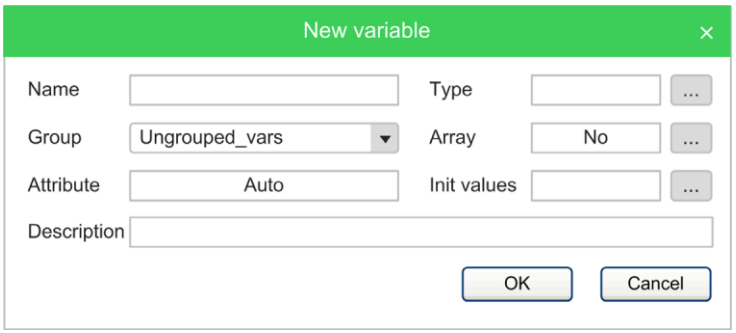
Глобальные переменные организованы в специальные папки в дереве проекта под названием **Global variables group (Группа глобальных переменных)**.

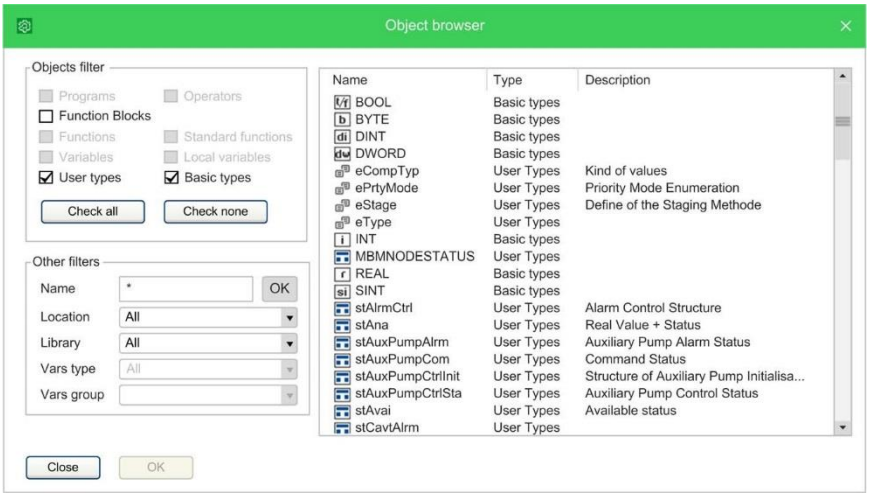
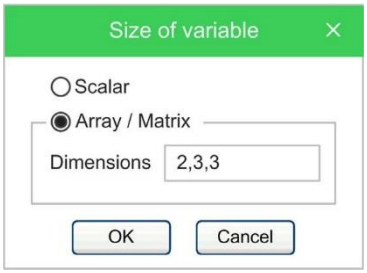
В соответствии с их свойствами эти переменные классифицируются следующим образом:

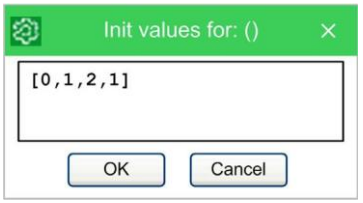
- Автоматические: компилятор автоматически размещает их в подходящем месте в памяти целевого устройства.
- Сопоставленные: они имеют заданный адрес в логической системе адресации целевого устройства, который вы указываете.
- Константы: объявляются с атрибутом `CONSTANT`; их значения не могут быть изменены логикой программы.
- Сохраняемые: они объявляются с атрибутом `RETAIN`; их значения хранятся в постоянной энергонезависимой области памяти целевого устройства.

Создание новой глобальной переменной

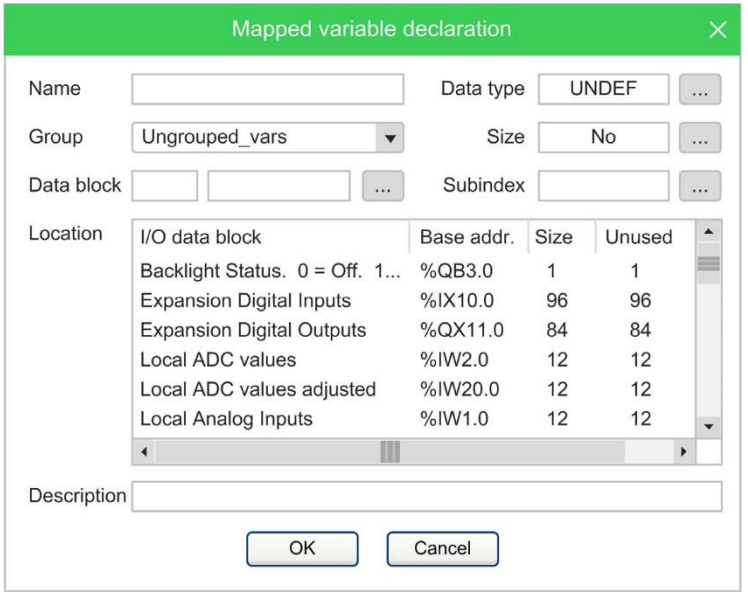
Чтобы создать новую глобальную переменную, выполните следующие действия.

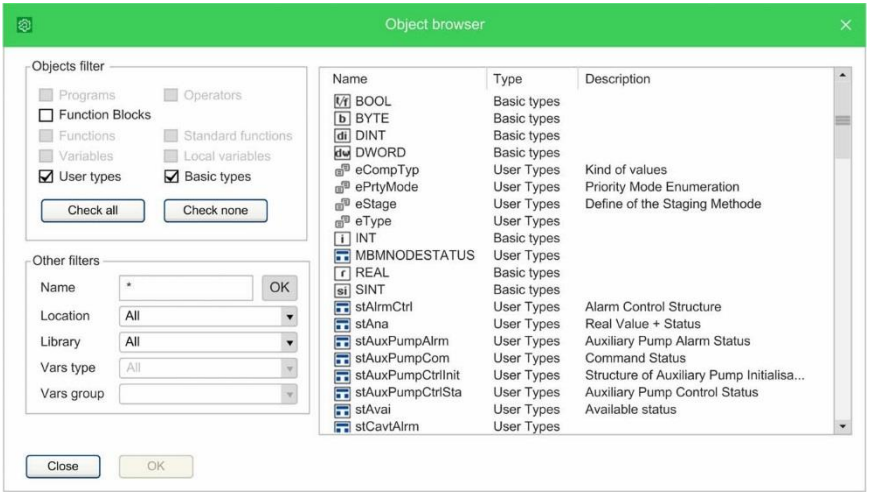
Шаг	Действие
1	Чтобы создать новую глобальную переменную, необходимо определить в проекте по крайней мере одну группу глобальных переменных. После этого выберите ее из дерева проекта, а затем выберите соответствующий пункт в меню **Project (Проект)** → **New Object (Новый объект)** → **New variable (Новая переменная)**. См. раздел «Операции в пользовательском рабочем пространстве» *(см. стр. 222)*. В модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** откроется диалоговое окно: 
2	Введите имя переменной. Имя переменной должно быть действительным идентификатором согласно стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3). Допустимые имена переменных могут состоять из любой комбинации букв, цифр и символов подчеркивания, но не могут начинаться с числа.

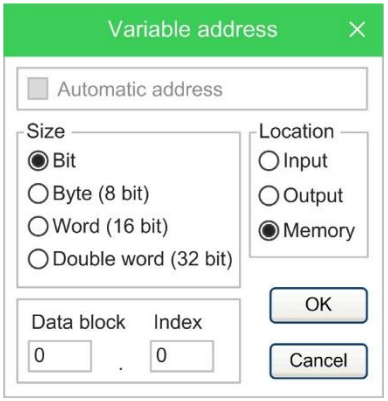
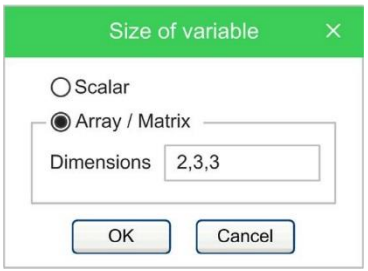
Шаг	Действие
3	Укажите тип переменной, введя его либо выбрав из списка, который отображается в модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** при нажатии кнопки ... **Browse (Обзор)**: 
4	Если вы хотите объявить массив, необходимо указать его размер, нажав кнопку ... **Browse (Обзор)** рядом с полем **Array (Массив)**:  Введите длину массива. Используйте запятую для отделения длины каждого измерения массива (до трех). Например: «2» или «2,3» или «2,3,3». **ПРИМЕЧАНИЕ.** Измерение должно быть больше единицы. Например, ввод «2,1» или «1,2» эквивалентен вводу «2».

Шаг	Действие
5	Вы также можете присвоить начальное значение переменной или отдельным элементам массива, нажав кнопку ... **Browse (Обзор)** рядом с полем **Init values (Начальные значения)**:  **ПРИМЕЧАНИЕ.** Начальные значения должны отделяться запятой.
6	Нажмите кнопку **OK** для подтверждения.

Чтобы создать новую глобальную сопоставленную переменную, выполните следующие действия.

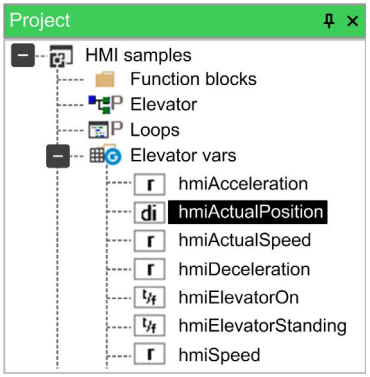
Шаг	Действие
1	Чтобы создать новую глобальную сопоставленную переменную, необходимо определить в проекте по крайней мере одну группу глобальных переменных. После этого выберите ее из дерева проекта, а затем выберите в меню **Project (Проект)** → **New Object (Новый объект)** → **New variable (Новая переменная)** → **Mapped Variable (Сопоставленная переменная)**. В модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** откроется диалоговое окно: 

Шаг	Действие
2	Введите имя переменной. Имя переменной должно быть действительным идентификатором согласно стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3). Допустимые имена переменных могут состоять из любой комбинации букв, цифр и символов подчеркивания, хотя они не могут начинаться с числа.
3	Укажите тип переменной, введя его либо выбрав из списка, который отображается в модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** при нажатии кнопки ... **Browse (Обзор)**: 
4	Вы можете выбрать группу в списке **Group (Группа)**.

Шаг	Действие
5	Вы должны указать адрес переменной, выполнив одну из следующих операций: ● Нажмите кнопку обзора ... **Data block (Блок данных)**, чтобы открыть редактор адреса; затем введите нужное значение:  Нажмите кнопку **OK**. ● Выберите из списка **Location (Местоположение)** область памяти, которую вы хотите использовать: инструмент автоматически вычислит адрес первой свободной ячейки памяти этой области.
6	В зависимости от выбранного местоположения вы можете указать его размер, нажав кнопку ... **Browse (Обзор)** рядом с полем **Size (Размер)**:  Введите длину массива. Используйте запятую для отделения длины каждого измерения массива (до трех). Например: «2» или «2,3» или «2,3,3». **ПРИМЕЧАНИЕ.** Измерение должно быть больше единицы. Например, ввод «2,1» или «1,2» эквивалентен вводу «2».
7	Вы можете ввести значение субиндекса.
8	Нажмите кнопку **OK** для подтверждения.

Редактирование глобальной переменной

Чтобы отредактировать определение существующей глобальной переменной, щелкните дважды по ней или по папке в дереве проекта, к которой она относится. Откроется редактор глобальных переменных, который позволяет изменять их определение.



	Name	Type	Address
1	hmiTargetPosition	DINT	%MB1.58
2	hmiActualPosition	DINT	%MD1.62
3	hmiHighSpeed	REAL	%MB1.66
4	hmiAcceleration	REAL	%MD1.70
5	hmiDeceleration	REAL	%MD1.74

Изменение имени переменной

Выберите в дереве проекта переменную, которую хотите переименовать, затем щелкните правой кнопкой мыши ее имя и выберите команду **Rename variable (Переименовать переменную)**. Также можно дважды щелкнуть по имени переменной и переименовать ее в окне редактора. Новое имя должно обновиться во всех местах, где используется переименованная переменная.

Дублирование переменной

Выберите в дереве проекта переменную, которую хотите дублировать, затем щелкните правой кнопкой мыши ее имя и выберите команду **Duplicate variable (Дублировать переменную)**. Введите имя новой дублированной переменной и подтвердите операцию.

Удаление глобальной переменной

Выберите в дереве проекта переменную, которую хотите удалить, затем щелкните правой кнопкой мыши ее имя и выберите команду **Delete variable (Удалить переменную)**.

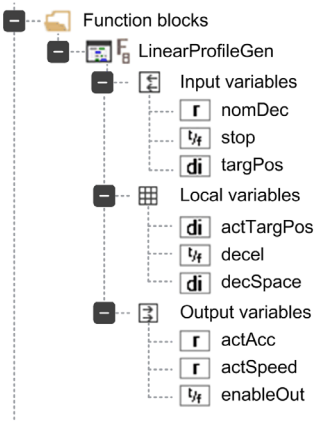
Подтвердите операцию удаления переменной.

Локальные переменные

Описание

Локальные переменные объявляются в программном модуле (программа, функция или функциональный блок), а сам программный модуль является единственным элементом проекта, который может обращаться к ним и иметь к ним доступ.

Локальные переменные перечисляются в дереве проекта под программным модулем, в котором они объявлены (только в том случае, если этот программный модуль открыт для редактирования). Локальные переменные далее подразделяются в соответствии с их классом (например, как входные или выходные переменные):



Для создания, редактирования и удаления локальных переменных необходимо открыть программный модуль в режиме редактирования и использовать редактор локальных переменных. Проект необходимо сохранить, чтобы обновить структуру программного модуля дерева проекта, включая изменения, применяемые к локальным переменным.

Для получения дополнительной информации см. раздел «Открытие редактора локальных переменных» *(стр. 263)*.

Создание локальных переменных

Выберите команду **Variables (Переменные)** → **Insert (Вставить)** или нажмите значок **Insert record (Вставить запись)** на панели инструментов **Project (Проект)**. Вы также можете создать несколько переменных *(см. стр. 198)*.

Переменная отображается желтым цветом в окне **Local variables (Локальные переменные)**, и вы можете определить ее характеристики, щелкнув по соответствующим ячейкам.

Созданные переменные можно отобразить в табличном формате, выбрав значок в правом верхнем углу окна:

Local variables

	Name	Type	Address	Array	Init value	Attribute
1	VAR1_OUT	INT	Auto	No		...
2	VAR2_OUT	INT	Auto	No		...
3	VAR3_OUT	INT	Auto	No		...
4	VAR4_OUT	INT	Auto	No		...
5	VAR5_OUT	INT	Auto	No		...
6	VAR6_OUT	INT	Auto	No		...
7	VAR7_OUT	INT	Auto	No		...
8	VAR8_OUT	INT	Auto	No		...
9	VAR9_OUT	INT	Auto	No		...
10	VAR10_OUT	INT	Auto	No		...

К характеристикам локальных переменных программы относятся:

- **Name (Имя)**: выбор имени переменной.
- **Type (Тип)**: выбор одной из предустановленных опций или переменных, определенных вами.
- **Address (Адрес)**: настройка по умолчанию **Auto (Автовыбор)**.
- **Array (Массив)**: определяет, является ли переменная массивом (если да, то указывается ее размерность) или нет.
- **Init value (Начальное значение)**: исходное значение переменной после каждого цикла подачи питания.
- **Attribute (Атрибут)**: для установки переменной в качестве **CONSTANT** (переменную нельзя перезаписать) или **RETAIN**.
- **Description (Описание)**: ввод описания для переменной.

ПРИМЕЧАНИЕ. Таблица локальных переменных имеет различные форматы для программных и функциональных блоков.

Созданные переменные можно отобразить в виде кода, выбрав второй значок в правом верхнем углу окна:

Local variables

```
0001 PROGRAM main
0002
0003 VAR
0004    VAR1_OUT : INT;
0005    VAR2_OUT : INT;
0006    VAR3_OUT : INT;
0007    VAR4_OUT : INT;
0008    VAR5_OUT : INT;
0009    VAR6_OUT : INT;
0010    VAR7_OUT : INT;
0011    VAR8_OUT : INT;
0012    VAR9_OUT : INT;
0013    VAR10_OUT : INT;
0014 END_VAR
0015
0016
```

Локальные переменные появляются в окне **Project (Проект)** под папкой программы, отмеченной значком.

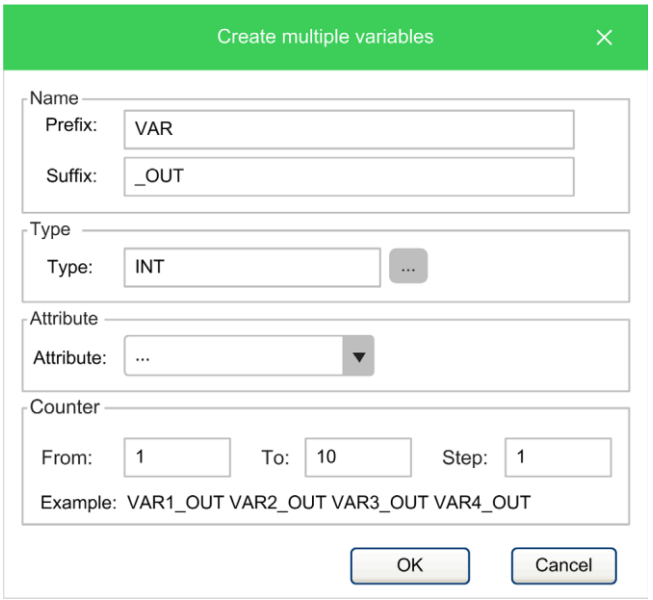
ПРИМЕЧАНИЕ. Локальные переменные инициализируются заново каждый раз при выполнении программного модуля.

Создание нескольких переменных

Описание

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** позволяет создавать несколько переменных одновременно.

Чтобы создать несколько переменных, выполните следующие действия.

Шаг	Действие
1	Откройте программный модуль для редактирования.
2	Выберите **Variables (Переменные)** → **Create multiple (Создать несколько)**. Откроется диалоговое окно: 
3	Укажите префикс и суффикс для имен новых переменных.
4	Укажите тип переменной, введя его либо выбрав из списка, который отображается при нажатии кнопки ... **Browse (Обзор)**.
5	При необходимости выберите в списке **Attribute (Атрибут)**.
6	Вставьте число переменных, которые вы хотите создать, указав начальный индекс, конечный индекс и значение шага. Пример сгенерированных имен переменных можно просмотреть в нижней части диалогового окна.

Раздел 9.4
Задачи

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Связывание программы с задачей	199
Настройка задач	201

Связывание программы с задачей

Обзор

Чтобы программа запускалась, она должна быть связана с задачей. Доступны следующие типы задач:

- **Boot (Загрузка)** — задача выполняется только один раз при запуске ПЛК.
- **Init (Инициализация)** — задача выполняется при каждой загрузке приложения и при запуске системы (после загрузочной задачи).

  ПРИМЕЧАНИЕ. Связанная программа инициализирует ведомые устройства и сообщения в соответствии с конфигурацией с помощью фиксированных значений, не зависящих от среды выполнения.

⚠ ПРЕДУПРЕЖДЕНИЕ

АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК КОНТРОЛЛЕРА

- Прежде чем загружать приложение, сначала проверьте состояние машины или процесса.
- Перед загрузкой приложения убедитесь в отсутствии опасностей, способных привести к травмированию людей, находящихся внутри или рядом с машиной или в непосредственной близости от технологического процесса.

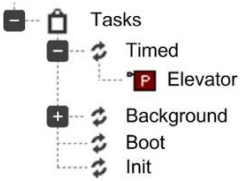
Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

- **Timed (Регулярная)** — эта задача выполняется через регулярные промежутки времени, которые можно задать. Установка по умолчанию зависит от типа целевого устройства.
  ПРИМЕЧАНИЕ. Сообщения Modbus не мешают выполнению этой задачи.
- **Background (Фоновая)** — эта задача запускается с низким приоритетом после выполнения **регулярных** задач.
- **Modbus** — эта задача выполняется для реализации ведущего устройства Modbus, вызова соответствующих функциональных блоков и отправки сообщений (только для M171O).

Связывание программы с типом задачи

Каждый новый проект содержит программу **main**, связанную с **фоновой** задачей. Программу **main** можно удалить и (или) связать с другими задачами.

Чтобы связать программу с задачей, выполните следующие действия.

Шаг	Действие
1	Щелкните правой кнопкой мыши задачу, в которую вы хотите добавить программу из дерева проекта, затем выберите команду **Add program (Добавить программу)**.
2	Выберите из открывшегося списка программу, которую вы хотите выполнить с помощью задачи, и подтвердите свой выбор.
3	Программа назначена задаче: 

Управление программой внутри задачи

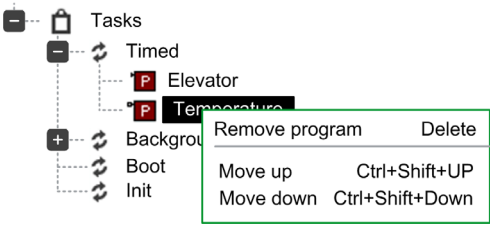
Вы можете назначить одной задаче несколько программ.

Программы выполняются последовательно, в порядке их назначения и отображения в дереве.

При щелчке правой кнопкой мыши по программе, связанной с задачей, доступны три действия:

- **Remove program (Удалить программу)** (**Delete**)
- **Move up (Переместить вверх)** (**Ctrl+Shift+Up**)
- **Move down (Переместить вниз)** (**Ctrl+Shift+Down**)

Команды **перемещения вверх и вниз** позволяют изменять порядок выполнения программ в рамках одной и той же задачи:



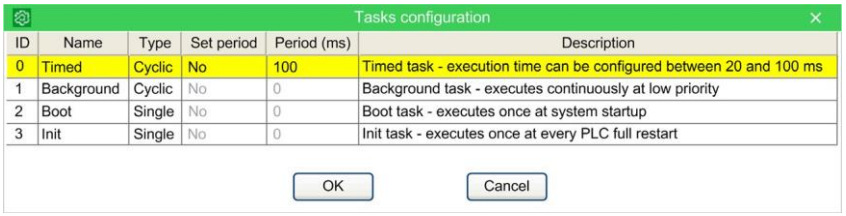
Настройка задач

Описание

В зависимости от целевого устройства можно изменять настройки задач контроллера.

Настройка задач

Чтобы настроить задачу, выполните следующие действия.

Шаг	Действие
1	Щелкните правой кнопкой мыши по элементу задачи в дереве проекта.
2	Выберите в контекстном меню пункт **Task configuration (Настройка задачи)**. **Результат:** отобразится окно **Task configuration (Настройка задачи)**.  Tasks configuration ID | Name | Type | Set period | Period (ms) | Description 0 | Timed | Cyclic | No | 100 | Timed task - execution time can be configured between 20 and 100 ms 1 | Background | Cyclic | No | 0 | Background task - executes continuously at low priority 2 | Boot | Single | No | 0 | Boot task - executes once at system startup 3 | Init | Single | No | 0 | Init task - executes once at every PLC full restart OK Cancel
3	Выберите **Yes (Да)** в списке **Set period (Установленный период)**.
4	Введите новое значение периода для задачи.
5	Нажмите **OK** для подтверждения.

ПРИМЕЧАНИЕ. Для всех других целевых устройств, кроме M172, продолжительность регулярной задачи можно установить в разделе **CONFIGURATION (КОНФИГУРИРОВАНИЕ)**, щелкнув по имени целевого устройства на дереве. В главном окне имеется флажок **Set time execution time (Установить время выполнения)**.

Раздел 9.5
Производные типы данных

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	202
Определение типа	203
Структуры	205
Перечисления	207
Поддиапазоны	209

Обзор

Описание

Раздел **Definitions (Определения)** окна **Workspace (Рабочее пространство)** позволяет определять производные типы данных.

Производный тип данных идентифицирует один или несколько типов данных и состоит из примитивных типов данных.

Вы можете гибко создавать специфические типы, обладающие расширенными свойствами, и использовать их в дополнение к примитивным типам данных.

В модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** вы можете управлять следующими элементами:

- Определения типов *(см. стр. 203)*
- Структуры *(см. стр. 205)*
- Перечисления *(см. стр. 207)*
- Поддиапазоны *(см. стр. 209)*

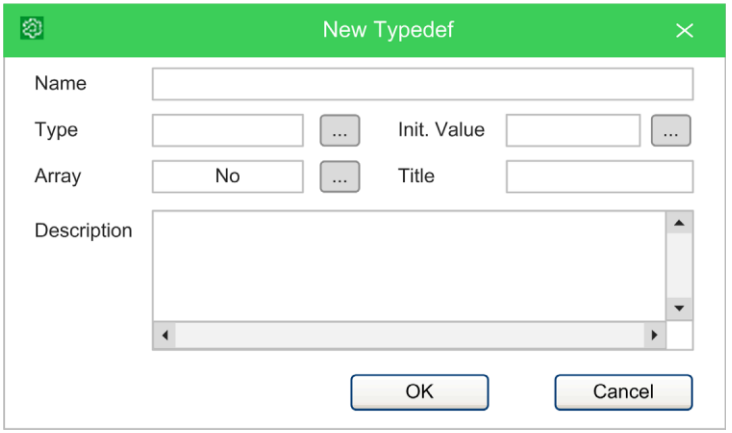
Определение типа

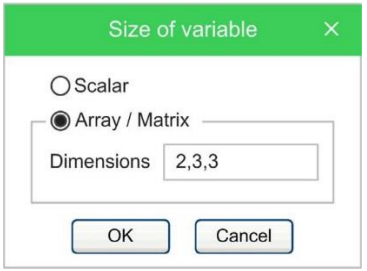
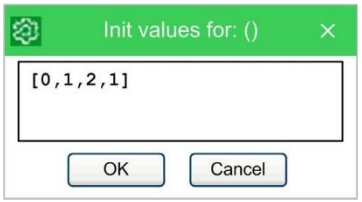
Описание

Ниже рассматривается использование определений типа.
Для получения дополнительной информации об определениях типа см. раздел «Описание определений типа» *(стр. 374)*.

Создание нового определения типа

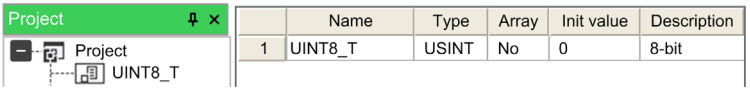
Чтобы создать новое **определение типа**, выполните следующие действия.

Шаг	Действие
1	Новое **определение типа** можно добавить в окне **Project (Проект)**: ● Щелкните правой кнопкой мыши по имени проекта. ● Нажмите **Add (Добавить)** → **New Definition (Новое определение)** → **TypeDef (Определение типа)**. Откроется диалоговое окно: 
2	Введите имя **определения типа**. Имя **определения типа** должно быть действительным идентификатором согласно стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3). Допустимые имена определений типа могут состоять из любой комбинации букв, цифр и символов подчеркивания, но не могут начинаться с числа.
3	Укажите **определение типа**, введя его либо выбрав из списка, который отображается при нажатии кнопки ... **Browse (Обзор).**

Шаг	Действие
4	Если вы хотите объявить массив, необходимо указать его размер, нажав кнопку ... **Browse (Обзор)** рядом с полем **Array (Массив)**:  Введите количество элементов в массиве. Используйте запятую для отделения количества элементов в каждом измерении (до трех). Например: «2» или «2,3» или «2,3,3». **ПРИМЕЧАНИЕ.** Чтобы учитывалось измерение, оно должно быть больше единицы. Например, ввод «2,1» или «1,2» эквивалентен значению «2».
5	Вы также можете присвоить начальное значение переменной или отдельным элементам массива, нажав кнопку ... **Browse (Обзор)** рядом с полем **Init values (Начальные значения)**:  **ПРИМЕЧАНИЕ.** Начальные значения должны отделяться запятой.
6	Вы можете указать: ● дополнительный заголовок; ● дополнительное описание.
7	Нажмите **OK** для подтверждения.

Редактирование определения типа

Чтобы отредактировать **определение типа**, дважды щелкните по нему в дереве окна **Project (Проект)**. Определение типа отображается в окне, в котором можно изменить значения:



Чтобы изменить значение, выберите его в таблице, а затем:

- введите новое значение;
- или нажмите кнопку обзора — появится окно для ввода нового значения.

Для редактирования свойств существующего **определения типа** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Edit properties (Редактировать свойства)**, чтобы открыть соответствующий редактор.

Чтобы изменить имя **определения типа** напрямую, щелкните по нему непосредственно в дереве окна **Project (Проект)**, а затем щелкните по нему еще раз, чтобы открыть поле имени. Введите новое имя и нажмите **Enter** для подтверждения.

Для редактирования свойств существующего **определения типа** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Edit properties (Редактировать свойства)**.

Для просмотра свойств существующего **определения типа** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **View properties (Просмотреть свойства)**, чтобы открыть соответствующее окно **Properties Window (Окно свойств)**.

Удаление определения типа

Для удаления существующего **определения типа** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Delete (Удалить)**.

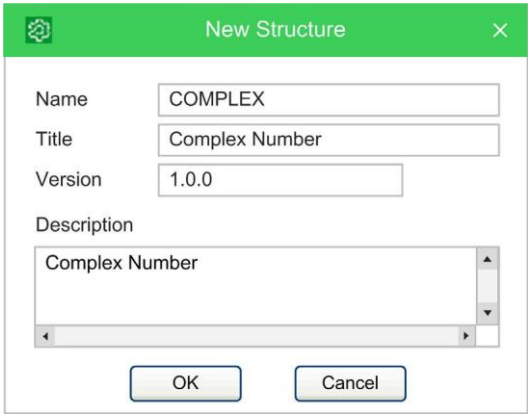
Структуры

Описание

Ниже рассматривается использование структур.
Для получения дополнительной информации о структурах см. раздел «Описание структур» *(стр. 375)*.

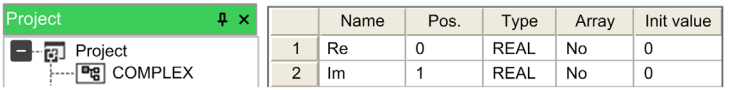
Создание новой структуры

Чтобы создать новую структуру в окне **Project (Проект)**, выполните следующие действия.

Шаг	Действие
1	Чтобы создать новую структуру, выполните одну из следующих операций: ● Щелкните правой кнопкой мыши имя проекта и нажмите **Add (Добавить)** → **New Definition (Новое определение)** → **Structure (Структура)**. ● Выберите проект в дереве окна **Project (Проект)**, а затем в меню выберите пункт **Project (Проект)** → **New object (Новый объект)** → **New Definition (Новое определение)** → **Structure (Структура).** Откроется диалоговое окно: 
2	Введите имя структуры.
3	Вы можете указать: ● дополнительный заголовок; ● дополнительный номер версии; ● дополнительное описание.
4	Нажмите **OK** для подтверждения.

Редактирование структуры

Чтобы отредактировать структуру, дважды щелкните по ней в дереве окна **Project (Проект).**



Щелкните правой кнопкой мыши, чтобы вставить или удалить элементы.

Чтобы изменить значение, выберите его в таблице, а затем:

- введите новое значение;
- или нажмите кнопку обзора — появится окно для ввода нового значения.

Для редактирования свойств существующей **структуры** щелкните по ней правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Edit properties (Редактировать свойства)**, чтобы открыть соответствующий редактор.

Чтобы изменить имя **структуры** напрямую, щелкните по ней непосредственно в дереве окна **Project (Проект)**, а затем щелкните по нему еще раз, чтобы открыть поле имени. Введите новое имя и нажмите **Enter** для подтверждения.

Для просмотра свойств существующей **структуры** щелкните по ней правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **View properties (Просмотреть свойства)**, чтобы открыть соответствующее окно **Properties Window (Окно свойств)**.

Удаление структуры

Для удаления существующей **структуры** щелкните по ней правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Delete (Удалить)**.

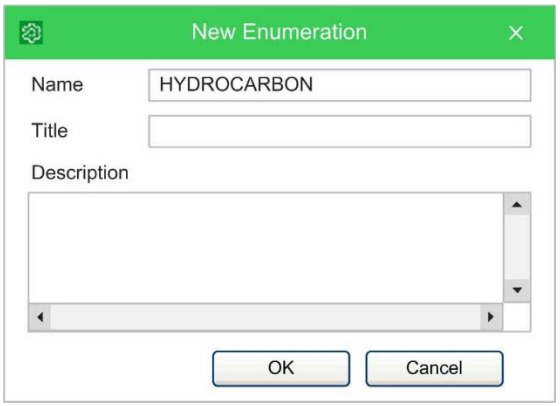
Перечисления

Описание

Ниже рассматривается использование перечислений.
Для получения дополнительной информации о **перечислениях** см. раздел «Перечисляемые типы данных» *(стр. 374)*.

Создание нового перечисления

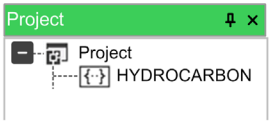
Чтобы создать новое **перечисление**, выполните следующие действия.

Шаг	Действие
1	Чтобы создать новое **перечисление**, выполните одну из следующих операций: ● Щелкните правой кнопкой мыши имя проекта и нажмите **Add (Добавить)** → **New Definition (Новое определение)** → **Enumeration (Перечисление)**. ● Выберите проект в дереве окна **Project (Проект)**, а затем в меню выберите пункт **Project (Проект)** → **New object (Новый объект)** → **New Definition (Новое определение)** → **Enumeration (Перечисление)**. Откроется диалоговое окно: 
2	Введите имя **перечисления.**

Шаг	Действие
3	Вы можете указать: ● дополнительный заголовок; ● дополнительное описание.
4	Нажмите **OK** для подтверждения.

Редактирование перечисления

Чтобы отредактировать **перечисление**, дважды щелкните по нему в дереве окна **Project (Проект)**.



	Name	Init value
1	Methane	1
2	Butane	4
3	Octane	8

Щелкните правой кнопкой мыши, чтобы вставить или удалить элемент.

Чтобы изменить значение, выберите его в таблице, а затем:

- введите новое значение;
- или нажмите кнопку обзора — появится окно для ввода нового значения.

Для редактирования свойств существующего **перечисления** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Edit properties (Редактировать свойства)**, чтобы открыть соответствующий редактор.

Чтобы изменить имя **перечисления** напрямую, щелкните по нему непосредственно в дереве окна **Project (Проект)**, а затем щелкните по нему еще раз, чтобы открыть поле имени. Введите новое имя и нажмите **Enter** для подтверждения.

Для просмотра свойств существующего **перечисления** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **View properties (Просмотреть свойства)**, чтобы открыть соответствующее окно **Properties Window (Окно свойств)**.

Удаление перечисления

Для удаления существующего **перечисления** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Delete (Удалить)**.

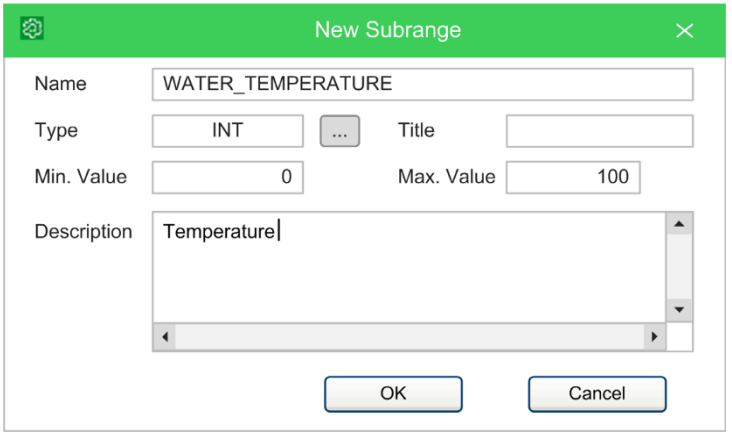
Поддиапазоны

Описание

Ниже рассматривается использование поддиапазонов.
Для получения дополнительной информации о **поддиапазонах** см. раздел «Описание поддиапазонов» *(стр. 375)*.

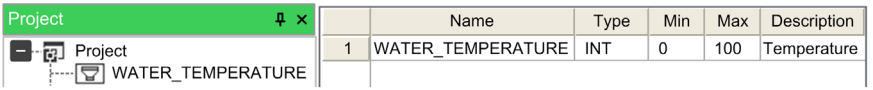
Создание нового поддиапазона

Чтобы создать новый **поддиапазон,** выполните следующие действия.

Шаг	Действие
1	Чтобы создать новый **поддиапазон**, выполните одну из следующих операций: ● Щелкните правой кнопкой мыши имя проекта и нажмите **Add (Добавить)** → **New Definition (Новое определение)** → **Subrange (Поддиапазон)**. ● Выберите проект в дереве окна **Project (Проект)**, а затем в меню выберите пункт **Project (Проект)** → **New object (Новый объект)** → **New Definition (Новое определение)** → **Subrange (Поддиапазон).** Откроется диалоговое окно: 
2	Введите имя **поддиапазона.**
3	Укажите тип **поддиапазона**, введя его либо выбрав из списка, который отображается при нажатии кнопки ... **Browse (Обзор)**.
4	Вы можете указать: ● минимальное значение; ● максимальное значение.
5	Вы можете указать: ● дополнительный заголовок; ● дополнительное описание.
6	Нажмите **OK** для подтверждения.

Редактирование поддиапазона

Чтобы отредактировать **поддиапазон**, дважды щелкните по нему в дереве окна **Project (Проект)**.



Чтобы изменить значение, выберите его в таблице, а затем:

- введите новое значение;
- или нажмите кнопку обзора — появится окно для ввода нового значения.

Для редактирования свойств существующего **поддиапазона** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Edit properties (Редактировать свойства)**, чтобы открыть соответствующий редактор.

Чтобы изменить имя **поддиапазона** напрямую, щелкните по нему непосредственно в дереве окна **Project (Проект)**, а затем щелкните по нему еще раз, чтобы открыть поле имени. Введите новое имя и нажмите **Enter** для подтверждения.

Для просмотра свойств существующего **поддиапазона** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **View properties (Просмотреть свойства)**, чтобы открыть соответствующее окно **Properties Window (Окно свойств)**.

Удаление поддиапазона

Для удаления существующего **поддиапазона** щелкните по нему правой кнопкой мыши в дереве окна **Project (Проект)** и выберите пункт **Delete (Удалить)**.

Раздел 9.6
Навигация по проекту

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	211
Обозреватель объектов	212
Поиск с помощью команды Find in Project (Найти в проекте)	217

Обзор

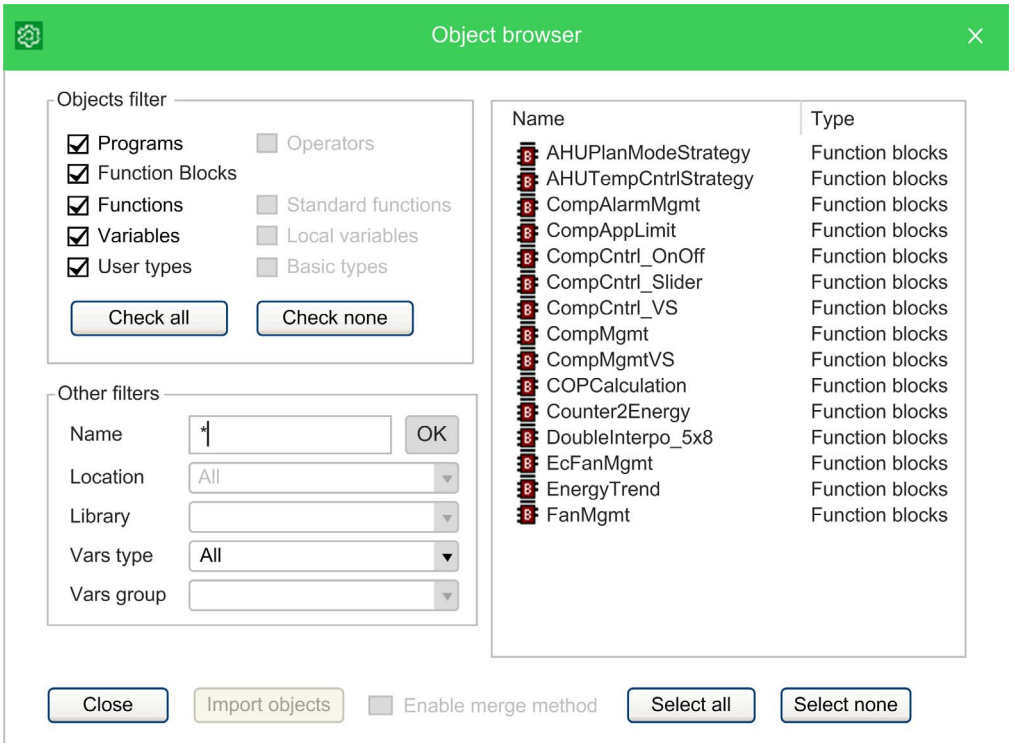
Описание

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** предоставляет два инструмента для поиска объектов в проекте: **Object browser (Обозреватель объектов)** и **Find in project (Найти в проекте)**.

Обозреватель объектов

Описание

В модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** для просмотра объектов используется инструмент **Object Browser (Обозреватель объектов)**.



Этот инструмент является контекстно-зависимым, т. е. вид выбираемых объектов и доступные операции с объектами зависят от контекста.

Обозреватель объектов можно открыть следующим образом:

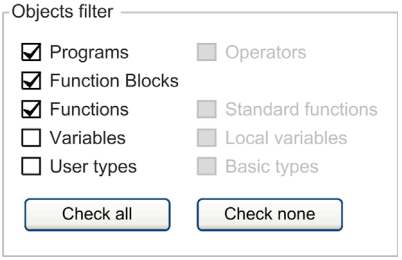
- Режим **Browser (Обозреватель)**:
  В модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** выберите команду **Project (Проект)** → **Object Browser (Обозреватель объектов).**
- Режим **Import object (Импорт объекта)**:
  Щелкните правой кнопкой мыши имя проекта в окне **Project (Проект)** и выберите **Import objects (Импорт объектов)**, в обозревателе объектов откроется выбранный объект.
- Режим **Select object (Выбор объекта)**:
  Например, чтобы добавить программу в задачу, щелкните правой кнопкой мыши элемент задачи в окне **Project (Проект)** и выберите команду **Add program (Добавить программу)**. Откроется окно **Object browser (Обозреватель объектов)**.

Взаимодействие пользователя с **обозревателем объектов** осуществляется аналогичным образом для всех трех режимов и описывается ниже.

Общие функции и использование обозревателя объектов

В этом разделе рассматриваются функции и использование **обозревателя объектов**.

Objects filter (Фильтр объектов)

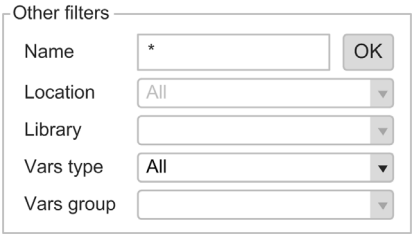


Это основной фильтр **обозревателя объектов.** Вы можете выбрать один из доступных (включенных) объектов.

В данном примере выбраны **Programs (Программы)**, **Function Blocks (Функциональные блоки)** и **Functions (Функции)**, поэтому объекты данного типа отображаются в списке объектов. Вы можете выбрать объекты **Variables (Переменные)** и **User types (Пользовательские типы)**, но объекты этого типа в настоящее время не отображаются в списке объектов.

Вы также можете нажать кнопку **Check all (Выбрать все)**, чтобы выбрать сразу все доступные объекты, или нажать кнопку **Check none (Отмена выбора)**, чтобы отменить выделение сразу всех объектов.

Другие фильтры



Выбранные объекты также можно отфильтровать по имени, расположению значков, конкретной библиотеке, типу переменной и группе переменных.

Все фильтры дополняют друг друга и применяются сразу же после настройки.

Name (Имя)	
Функция	Фильтрует объекты по имени.
Допустимые значения	Все строки символов.
Использование	Введите строку, чтобы отобразить объект, имя которого совпадает с этой строкой. Используйте подстановочный символ *, если необходимо отобразить все объекты, имя которых содержит строку, введенную в текстовое поле **Name (Имя)**. Введите *, если хотите отключить этот фильтр. Нажмите **Enter** при нахождении курсора в поле редактирования или нажмите кнопку **OK**, чтобы применить фильтр.
Охват	Все типы объектов.

Symbol location (Расположение символа)	
Функция	Фильтрует объекты по их расположению.
Допустимые значения	All (Все), Project (Проект), Target (Целевое устройство), Library (Библиотека), Aux Sources (Вспом. источники).
Использование	All = отключает этот фильтр. Project = объекты, объявленные в проекте **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**. Target = объекты микропрограммы. Library = объекты, содержащиеся в библиотеке. Используется вместе с фильтром **Library (Библиотека)**. Aux Sources = отображает только вспомогательные источники.
Охват	Все типы объектов.

Library (Библиотека)	
Функция	Фильтрует объекты, содержащиеся в библиотеке. Значение этого фильтра имеет значение только в том случае, если для фильтра **Symbol location (Расположение символа)** установлено значение **Library (Библиотека)**.
Допустимые значения	All (Все), библиотека1, библиотека2 и т. д.
Использование	All = отображает объекты, содержащиеся в любой библиотеке. библиотекаN = отображает только объекты, содержащиеся в библиотеке с именем библиотекаN.
Охват	Все типы объектов.

Vars Type (Тип переменной)	
Функция	Фильтрует глобальные и системные переменные (также известные как переменные микропрограммы) в соответствии с их типом.
Допустимые значения	All (Все), Normal (Стандартные), Constant (Константы), Retain (Сохраняемые).
Использование	All = отображает все глобальные и системные переменные. Normal = отображает только стандартные переменные. Constant = отображает только константы. Retain = отображает только сохраняемые переменные.
Охват	Переменные.

Vars Group (Группа переменных)	
Функция	Фильтрует переменные в соответствии с их группой.
Допустимые значения	Analog_inputs, Analog_outputs и пр.
Использование	Отображает переменные, относящиеся к выбранной группе.
Охват	Переменные.

Список объектов

В списке объектов **Object list** отображаются все отфильтрованные объекты. Содержимое данного списка можно упорядочить по возрастанию или убыванию, щелкнув по заголовку столбца. Вы можете упорядочить элементы по имени (**Name**), типу (**Type**) или описанию (**Description**).

Двойной щелчок по элементу позволяет выполнить связанную с ним операцию по умолчанию (действие совпадает с действием кнопок **OK**, **Import object [Импорт объекта]** или **Open source [Открыть исходный код]**).
Если разрешен множественный выбор элементов, отображаются кнопки **Select all (Выбрать все)** и **Select none (Отменить выбор)**.
Можно выделить все объекты, нажав кнопку **Select all (Выбрать все)**. Кнопка **Select none (Отменить выбор)** позволяет отменить выбор всех объектов. Если в списке выбран хотя бы один пункт, то кнопки становятся активны.

Изменение размера

Размер окна **Object browser (Обозреватель объектов)** можно изменять — при наведении на рамку окна курсор изменяется, позволяя менять размер окна. При повторном открытии окно **обозревателя объектов** сохраняет тот же размер и положение, что и при предыдущем использовании.

Использование обозревателя объектов

Чтобы использовать обозреватель объектов для просмотра элементов проекта, выберите пункт меню **Project (Проект)** → **Object Browser (Обозреватель объектов)**.

Доступные объекты

В этом режиме можно вывести список объектов следующих типов:

- программы;
- функциональные блоки;
- функции;
- переменные;
- пользовательские типы.

Напротив этих элементов в разделе **Objects filter (Фильтр объектов)** можно установить или снять флажок, что позволяет отобразить или скрыть объекты выбранного типа в списке.

Другие типы объектов (операторы, стандартные функции, локальные переменные, базовые типы) недоступны для просмотра в этом контексте, поэтому они не отмечены флажком и отключены.

Доступные операции

Open source (Открыть исходный код) — операция по умолчанию при двойном щелчке на элементе	
Функция	Открывает редактор, с помощью которого был создан выбранный объект, и отображает соответствующий исходный код.
Использование	Если объектом является программа, функция или функциональный блок, данная кнопка открывает соответствующий редактор исходного кода. Если объект является переменной, то данная кнопка открывает редактор переменных. Выберите объект, чей редактор вы хотите открыть, затем нажмите кнопку **Open source (Открыть исходный код)**.

Export to library (Экспорт в библиотеку)	
Функция	Позволяет экспортировать объект в библиотеку.
Использование	Выберите объекты, которые вы хотите экспортировать, затем нажмите кнопку **Export to library (Экспорт в библиотеку)**.

Delete objects (Удаление объектов)	
Функция	Позволяет удалить объект.
Использование	Выберите объекты, которые вы хотите удалить, затем нажмите кнопку **Delete object (Удалить объект)**.

Импорт через обозреватель объектов

Обозреватель объектов также используется для импорта объектов в проект из внешней библиотеки.

Чтобы использовать обозреватель объектов для импорта внешней библиотеки в проект, выберите пункт меню **Project (Проект)** → **Object Browser (Обозреватель объектов)**.

Доступные объекты

В этом режиме можно вывести список объектов следующих типов:

- программы;
- функциональные блоки;
- функции;
- переменные;
- пользовательские типы.

Напротив этих элементов в разделе **Objects filter (Фильтр объектов)** можно установить или снять флажок, что позволяет отобразить или скрыть объекты выбранного типа в списке.

Другие типы объектов (операторы, стандартные функции, локальные переменные, базовые типы) недоступны для импорта, поэтому они не отмечены флажком и отключены.

Доступные операции

Import objects (Импорт объектов) — единственная операция, доступная в этом режиме. Импортировать выделенные объекты можно нажатием кнопки **Import objects (Импорт объектов)** или двойным щелчком мыши по одному из объектов в списке.

Выбор объектов через обозреватель объектов

Диалоговое окно обозревателя объектов можно использовать для множества операций, требующих выбора одного объекта ПЛК. Обозреватель объектов можно использовать для выбора программы, добавляемой в задачу, для выбора типа переменной, для выбора элемента, для поиска в проекте и пр.

Доступные объекты

Доступные объекты зависят от контекста. Например, в случае назначения программы тому или иному заданию единственными доступными объектами являются объекты типа «программа».

По умолчанию не все доступные объекты можно выбрать.

Доступные операции

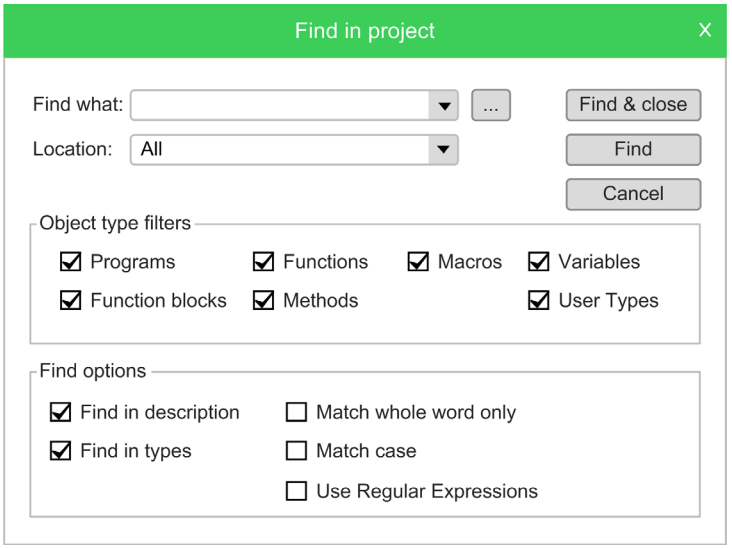
В этом режиме можно выбрать один объект двойным щелчком в списке или нажав кнопку **OK**, после чего диалоговое окно автоматически закроется.

Поиск с помощью команды Find in Project (Найти в проекте)

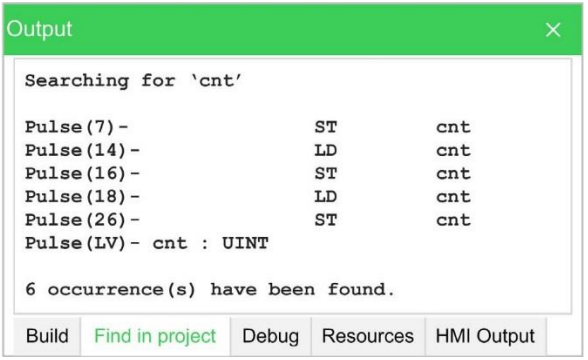
Описание

Команда **Find in project (Найти в проекте)** возвращает все экземпляры введенной строки символов, имеющиеся в проекте.
Чтобы воспользоваться этой функцией, выберите пункт меню **Edit (Редактировать)** → **Find in project (Найти в проекте)**.

В модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** откроется следующее диалоговое окно:



Шаг	Действие
1	В поле **Find what (Найти)** введите имя необходимого объекта. В противном случае нажмите кнопку ... **Browse (Обзор)** справа от поля и выберите имя объекта из списка всех существующих объектов.
2	Выберите одно из значений, перечисленных в поле **Location (Местоположение)**, чтобы ограничить поиск по местоположению отслеживаемых объектов.
3	В разделе с названием **Object type filters (Фильтры по типу объекта)** доступно семь флажков, каждый из которых позволяет активировать поиск строки среди соответствующих объектов.
4	Установите флажки для соответствующих опций в разделе **Find options (Опции поиска)**.

Шаг	Действие
5	Нажмите **Find (Поиск)**, чтобы начать поиск, или **Cancel (Отмена)**, чтобы завершить операцию. Результаты отображаются на вкладке **Find in project (Найти в проекте)** окна **Output (Вывод)**. 

Раздел 9.7

Пользовательское рабочее пространство проекта

Содержание этого раздела

В этом разделе описаны следующие темы.

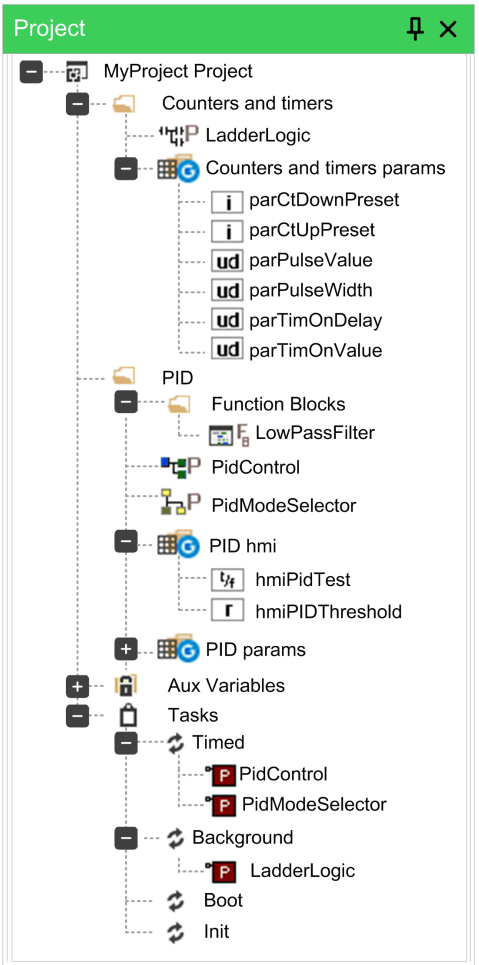
Тема	Страница
Обзор	220
Включение пользовательского рабочего пространства в существующий проект	221
Миграция между рабочими пространствами	221
Базовые блоки пользовательского рабочего пространства	221
Операции в пользовательском рабочем пространстве	222
Элементы рабочего пространства и их ограничения	223

Обзор

Описание

Функции пользовательского рабочего пространства позволяют организовать дерево окна **Project (Проект)** в соответствии с вашими потребностями и повысить эффективность управления проектом.

Организационные блоки пользовательского рабочего пространства являются логическими и не влияют на код ПЛК.



Включение пользовательского рабочего пространства в существующий проект

Описание

Чтобы включить эту функцию, установите флажок **Use customizable workspace (Использовать настраиваемое рабочее пространство)** в меню **Project (Проект)** → **Options… (Параметры…)** → вкладка **General (Общее)**. После включения проект необходимо перезагрузить.
Для получения дополнительной информации см. описание параметров, отвечающих за общую информацию о проекте *(стр. 165)*.
По умолчанию эта функция включена и настроена в соответствии с целевым устройством.

Миграция между рабочими пространствами

Описание

Всякий раз при активации функции пользовательского рабочего пространства модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** реорганизует рабочее пространство в соответствии с пользовательскими настройками, руководствуясь следующей логикой:

Static (old) workspace to custom (new) (Статическое [старое] рабочее пространство в пользовательское [новое])

Фиксированные логические блоки (например, папка функциональных блоков) преобразуются в новые динамические папки с теми же именами. Фиксированные блоки глобальных групп (например, сопоставленные переменные) преобразуются в новые глобальные динамические группы с теми же именами. Глобальные переменные, не принадлежащие ни одной из групп, объединяются в новую группу под названием **Ungrouped global vars (Несгруппированные глобальные переменные)**.

Custom (new) workspace to static (old) (Пользовательское [новое] рабочее пространство в статическое [старое])

Пользовательские блоки уничтожаются, а программные модули и глобальные переменные группируются в фиксированные блоки по умолчанию (например: папка функциональных блоков и сопоставленные переменные).

Базовые блоки пользовательского рабочего пространства

Описание

В новом пользовательском рабочем пространстве можно работать, используя два разных основных логических блока.

- **Folder (Папка)**: это опциональный логический блок, который может содержать программные модули, папки (вы можете вкладывать папки друг в друга), а также группу глобальных переменных.

- **Global variables group (Группа глобальных переменных)**: это обязательный логический блок, который может содержать только глобальные переменные. Чтобы создать глобальную переменную, в вашем пользовательском рабочем пространстве должна иметься хотя бы одна группа глобальных переменных.

Операции в пользовательском рабочем пространстве

Описание

В целях оптимизации структуры вашего проекта можно выполнять различные операции.

Создание папки

Для создания папки выберите корневой элемент дерева проекта или существующую папку, чтобы создать вложенную, а затем щелкните правой кнопкой мыши и выберите пункт **Add (Добавить) → New folder (Новая папка)**.

В результате этой операции добавляется новый настраиваемый блок папок, готовый к переименованию. Имя папки по умолчанию **New folder**.

Создание группы глобальных переменных

Для создания группы глобальных переменных выберите корневой элемент дерева проекта или существующую папку, чтобы создать вложенную группу, а затем щелкните правой кнопкой мыши и выберите пункт **Add (Добавить) → New global variables group (Новая группа глобальных переменных)**.

В результате этой операции добавляется новый настраиваемый блок папок, готовый к переименованию. Имя папки по умолчанию **New var group**.

Переименование блока (папка или группа глобальных переменных)

Чтобы переименовать группу глобальных переменных или папку, выберите ее, затем щелкните правой кнопкой мыши и выберите пункт **Rename (Переименовать)**. В результате этой операции имя блока будет подготовлено к переименованию.

Удаление блока (папка или группа глобальных переменных)

Чтобы удалить группу глобальных переменных или папку, выберите ее, затем щелкните правой кнопкой мыши и выберите пункт **Delete (Удалить)**. Если блоки содержат дочерние элементы, пользователю будет предложено три варианта.

Шаг	Действие
1	Удалить также все дочерние элементы (это может повлиять на ПЛК).
2	Не удалять дочерние элементы, они будут перемещены на уровень выше в дереве проекта.
3	Отменить операцию и ничего не делать.

Экспорт всех дочерних элементов в библиотеку

Чтобы экспортировать все дочерние элементы группы глобальных переменных или папки, выберите их, затем щелкните правой кнопкой мыши и выберите пункт **Export all children to library (Экспортировать все дочерние элементы в библиотеку)**.

Эта операция позволяет рекурсивно экспортировать в библиотеку все дочерние элементы выбранного объекта. Для получения дополнительной информации см. раздел «Экспорт в библиотеку» *(стр. 177)*.

Перемещение блока

Вы можете просто перетаскивать блоки в другое место на дереве, упорядочивая рабочее пространство вашего проекта. Все дочерние элементы перемещаются вместе с родительским элементом без нарушения первоначальной структуры.

Перемещать переменные также можно как из дерева проекта (одиночное выделение), так и из сетки переменных (одиночное и множественное выделение). Для получения дополнительной информации см. раздел «Редактор переменных» *(стр. 262)*.

Элементы рабочего пространства и их ограничения

Описание

Некоторые элементы рабочего пространства жестко фиксированы и не настраиваются. Они автоматически генерируются модулем **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** и не позволяют производить над ними какие-либо специальные операции.

- Корневой элемент проекта.
  Вы не можете перемещать, переименовывать или удалять этот элемент. Он может содержать настраиваемые блоки в качестве дочерних элементов.
- Дочерние элементы программных модулей.
  Эти элементы генерируются в соответствии со структурой программного модуля, к которому они относятся. Вы не можете перемещать, переименовывать или удалять эти элементы непосредственно из дерева. Для получения дополнительной информации о программных модулях см. раздел «Программные модули» *(стр. 183)* .
- Дочерние элементы SFC.
  Эти элементы подчиняются ранее упомянутым правилам, однако в случае дочерних узлов SFC операции по переименованию или удалению не разрешены также и для программных модулей, которые относятся к элементам Actions (Действия) или Transitions (Переходы). Для получения дополнительной информации о языке SFC см. раздел «Редактор последовательных функциональных схем» *(стр. 254)* .
- Элемент **Aux Variables (Вспом. переменные)**.
  Вы не можете перемещать, переименовывать или удалять этот элемент и его дочерние элементы. Они автоматически генерируются модулем **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**.
- Элемент **Tasks (Задачи)**.
  Вы не можете перемещать, переименовывать или удалять эти элементы. Они автоматически генерируются модулем **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**. Для получения дополнительной информации см. раздел «Задачи» *(стр. 199)* .

Глава 10

Редактирование исходного кода

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
10.1	Обзор	226
10.2	Редактор списков инструкций	227
10.3	Редактор функциональных блок-схем	230
10.4	Редактор релейных схем	236
10.5	Редактор «Структурного текста»	251
10.6	Редактор последовательных функциональных схем	254
10.7	Редактор переменных	262

Раздел 10.1
Обзор

Обзор

Редакторы ПЛК

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** включает в себя пять редакторов исходного кода, поддерживая тем самым весь спектр языков программирования МЭК 61131-3 (ГОСТ Р МЭК 61131-3):

- Список инструкций (Instruction List, IL) *(см. стр. 227)*.
- Язык функциональных блок-схем (Function Block Diagram, FBD) *(см. стр. 230)*.
- Язык релейных схем (Ladder Diagram, LD) *(см. стр. 236)*.
- «Структурный текст» (Structured Text, ST) *(см. стр. 251)*.
- Язык последовательных функциональных схем (Sequential Function Chart, SFC) *(см. стр. 254)*.

Как графические, так и текстовые редакторы поддерживают всплывающие подсказки. Если включить эту функцию *(см. стр. 47)*, модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** будет отображать информацию о символах при наведении на них курсора мыши.

Раздел 10.2
Редактор списков инструкций

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	227
Функции редактирования	228
Ссылка на объекты ПЛК	228
Автоматическое обнаружение ошибок	228
Закладки	229

Обзор

Описание

Редактор списка инструкций (IL) позволяет писать код и изменять программные модули на языке IL.

```
0001 MUL sysIq
0002 SHR 16#04
0003 ADD addIqSq
0004
0005 MUL sysIq
0006 SHR 16#04
0007 ADD addIqSq
```

Функции редактирования

Описание

Редактор IL предоставляет функции, характерные для большинства редакторов, работающих на платформе Windows, а именно:

- Выделение текста
- **Edit (Редактировать) → Cut (Вырезать)**
- **Edit (Редактировать) → Copy (Копировать)**
- **Edit (Редактировать) → Paste (Вставить)**
- **Edit (Редактировать) → Replace (Заменить)**
- Перетаскивание выделенного текста

Ссылка на объекты ПЛК

Описание

Если необходимо добавить ссылку на существующий объект ПЛК, у вас есть два варианта:

- Можно ввести непосредственно имя объекта ПЛК.
- Можно перетащить его в нужное место. Например, глобальные переменные можно перетащить из окна **Workspace (Рабочее пространство)**, стандартные операторы и встроенные функции — из окна **Libraries (Библиотеки)**, а локальные переменные можно выбрать в редакторе локальных переменных.

Автоматическое обнаружение ошибок

Описание

Редактор IL также автоматически отображает место расположения ошибок компиляции. Чтобы узнать, где произошла ошибка компиляции, дважды щелкните по соответствующей строке на панели **Output (Вывод)**.

Закладки

Описание

Вы можете установить закладки для часто используемых строк в исходном файле. Ненужные закладки можно удалить.

Установка закладки

Переместите курсор в строку, которую нужно пометить закладкой, затем нажмите **Ctrl+F2**.
Эта строка будет отмечена на полях голубым кружком.

Управление закладками осуществляется в разделе **Edit (Редактирование)** → **Bookmarks… (Закладки…)**. Доступные команды:

- **Add/toogle (Добавить/переключить)** (**Ctrl+F2**)
- **Next (Вперед)** (**F2**)
- **Prev (Назад)** (**Shift+F2**)
- **Remove all (Удалить все)**

Переход к следующей закладке

Нажимайте **F2**, пока не доберетесь до нужной строки.

Переход к предыдущей закладке

Нажимайте **Shift+F2**, пока не доберетесь до нужной строки.

Удаление закладки

Переместите курсор в любое место строки, содержащей закладку, а затем нажмите **Ctrl+F2**.

Раздел 10.3
Редактор функциональных блок-схем

Содержание этого раздела

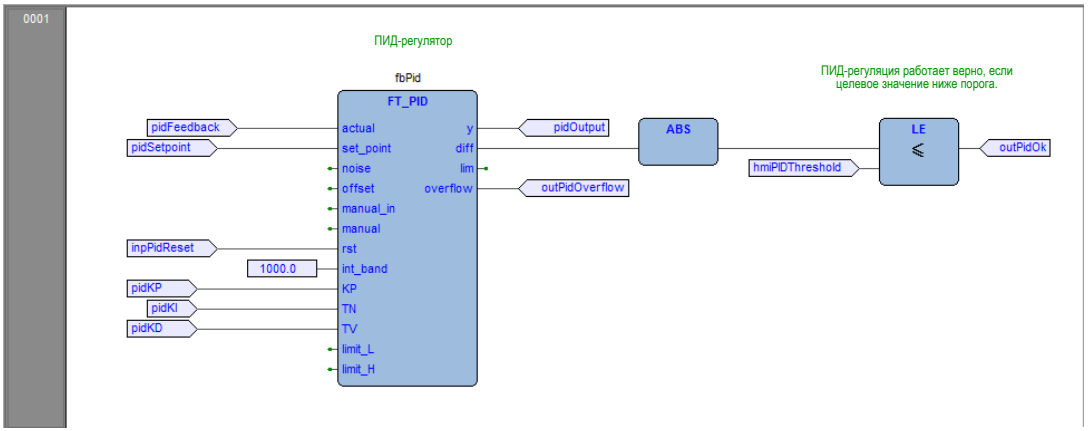
В этом разделе описаны следующие темы.

Тема	Страница
Обзор	230
Создание нового документа FBD	231
Добавление/удаление сетей	231
Маркировка сетей	232
Вставка и соединение блоков	233
Редактирование сетей	235
Изменение свойств блоков	235
Получение информации о блоке	235
Автоматическое обнаружение ошибок	235

Обзор

Описание

Редактор функциональных блок-схем (FBD) позволяет писать код и изменять программные модули на языке FBD.



Для получения более подробной информации см. справочник по языку функциональных блок-схем *(стр. 409)*.

Создание нового документа FBD

Описание

Создание и изменение документов FBD рассматривается в следующих разделах:

- Создание нового программного модуля *(см. стр. 184)*.
- Создание функционального блока / функции *(см. стр. 185)*.
- Редактирование программных модулей *(см. стр. 186)*

Добавление/удаление сетей

Описание

Каждый программный модуль, написанный на языке FBD, состоит из последовательности сетей. Под сетью понимается максимальный набор взаимосвязанных графических элементов. Верхняя и нижняя границы каждой сети обозначаются двумя прямыми линиями, при этом каждая сеть ограничена слева серой областью, в которой указан номер сети:

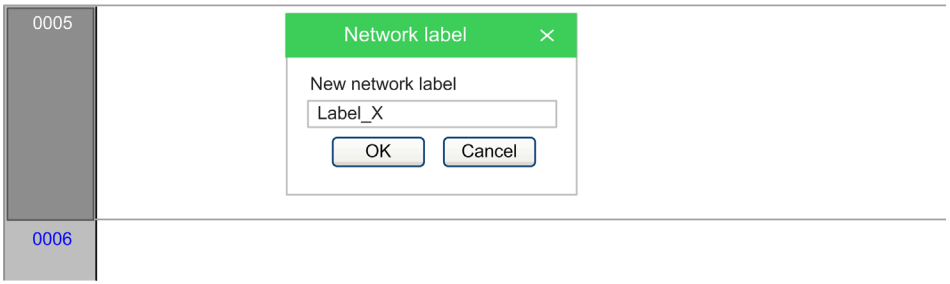
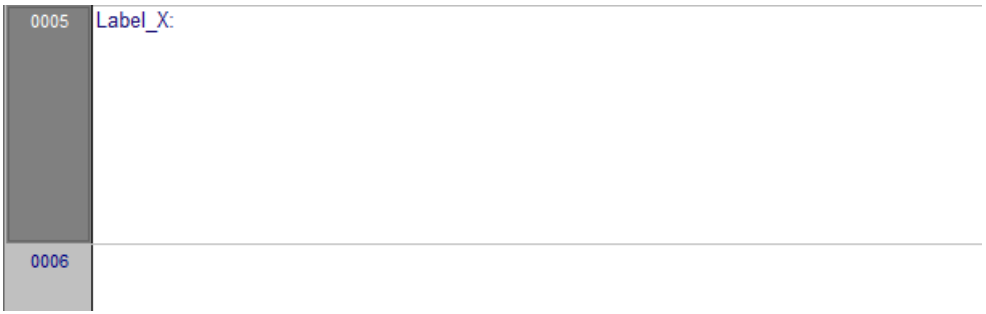
Вы можете выполнять следующие операции над сетями:

- Чтобы добавить новую пустую сеть, выберите пункт **Scheme (Схема)** → **Network (Сеть)** → **New (Создать)** и выберите позицию новой сети: **Top (Наверху)**, **Bottom (Внизу)**, **Before (До)** или **After (После)**.
- Чтобы удалить сеть, выберите ее и нажмите клавишу **Delete**.
- Чтобы отобразить фоновую сетку, помогающую при выравнивании объектов, выберите пункт **View (Вид)** → **Grid (Сетка)** .
- Чтобы добавить комментарий, выберите пункт **Scheme (Схема)** → **Object (Объект)** → **New (Создать)** → **Comment (Комментарий)**.

Маркировка сетей

Описание

Вы можете изменить стандартный порядок выполнения сетей, используя оператор прыжка, который передает управление программой в отмеченную сеть.
Чтобы назначить метку сети, выполните следующие действия.

Шаг	Действие
1	Выберите сеть.
2	Выполните одну из следующих операций: ● Выберите пункт **Scheme (Схема)** → **Network (Сеть)** → **Label (Метка)**. ● Дважды щелкните серую область, содержащую номер сети.
3	Откроется диалоговое окно, позволяющее ввести метку, которую необходимо связать с выбранной сетью: 
4	Нажмите кнопку **OK**. Метка отображается в верхнем левом углу выбранной сети. 

Вставка и соединение блоков

Обзор

В этом разделе рассматривается тема построения сети.

Вставка блоков

Добавьте блок в пустую сеть, выполнив одну из следующих операций:

- Откройте окно **Object browser (Обозреватель объектов)**.
  - Выберите в меню пункт **Scheme (Схема)** → **Object (Объект)** → **New (Создать)** → **Function Block (Функциональный блок)**.
  - Нажмите на панели инструментов FBD.

    Если блок является константой, оператором возврата или оператором прыжка, вы можете нажать непосредственно на соответствующие кнопки на панели инструментов **FBD** *(см. стр. 158)*.

    Затем выберите один пункт из списка и нажмите **OK**.
- Перетащите выделенный объект из соответствующего места. Например, глобальные переменные можно перетащить из окна **Workspace (Рабочее пространство)**, стандартные операторы и встроенные функции — из окна **Libraries (Библиотеки)**, а локальные переменные можно выбрать в редакторе локальных переменных.

Повторяйте эту процедуру до тех пор, пока не добавите все блоки в сеть.

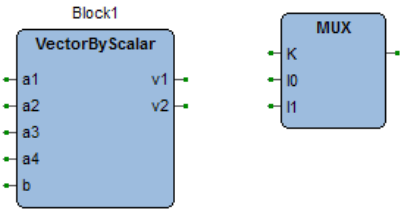
Соединение блоков

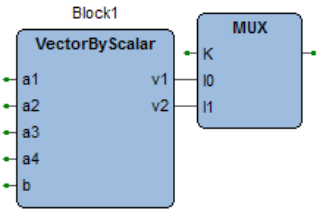
Для соединения блоков вручную выполните следующие действия.

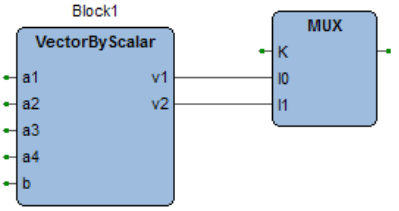
- Включите режим ручного соединения, выполнив одну из следующих операций:
  - Выберите в меню пункт **Edit (Редактировать)** → **Connection mode (Режим соединения)**.
  - Нажмите на панели инструментов FBD.
  - Нажмите клавишу **Пробел**.
- Щелкните один раз на выводе источника, затем переместите указатель мыши на вывод назначения: редактор FBD нарисует логическую линию от первого пункта ко второму.

Для соединения блоков автоматически выполните следующие действия.

- Включите режим автоматического соединения, выполнив одну из следующих операций:
  - Выберите в меню пункт **Scheme (Схема) → Auto connect (Автоматическое соединение)**.
  - В редакторе кода щелкните правой кнопкой мыши и нажмите кнопку **Auto connect (Автоматическое соединение)**.
- Затем выберите один блок, перетащите его к другому, чтобы соответствующие выводы совпали. Редактор FBD автоматически нарисует между ними логическую линию.







Удаление блоков

Чтобы удалить блок, выберите его и нажмите клавишу **Delete**.

При удалении блока его соединения не удаляются автоматически, но они становятся недействительными и перерисовываются красным цветом. Выберите пункт **Scheme (Схема) → Delete invalid connection (Удалить недействительное соединение)**.

Редактирование сетей

Описание

Редактор FBD предоставляет следующие функции, характерные для большинства графических приложений, работающих на платформе Windows.

- Выделение блока.
- Выделение набора блоков осуществляется сочетанием **Shift + левая кнопка мыши** и путем обведения необходимых блоков рамкой.
- Операции над одним или несколькими блоками: **Edit (Редактировать) → Cut (Вырезать)**, **Edit (Редактировать) → Copy (Копировать)** , **Edit (Редактировать) → Paste (Вставить)** .
- Перетаскивание.

Изменение свойств блоков

Описание

- Выберите пункт **Scheme (Схема) → Increment pins (Увеличить кол-во выводов)**, чтобы увеличить на единицу количество входных выводов для некоторых операторов и встроенных функций.
- Выберите пункт **Scheme (Схема) → Enable EN/ENO pins (Включить выводы EN/ENO)**, чтобы отобразить функцию включения входных и выходных выводов.
- Выберите в меню пункт **Scheme (Схема) → Object (Объект) → Instance name (Имя экземпляра)** или **Scheme (Схема) → Object properties (Свойства объекта)**, чтобы изменить имя экземпляра функционального блока.

Для получения дополнительной информации см. пункт «Изменение свойств блоков» *(стр. 245)* в разделе «Редактор релейных схем».

Получение информации о блоке

Описание

Вы всегда можете получить информацию о блоке, выбрав его и выполнив одну из следующих операций.

- Выберите пункт **Scheme (Схема) → Object (Объект) → Open source (Открыть исходный код)** , чтобы открыть исходный код блока.
- Выберите пункт **Scheme (Схема) → Object properties (Свойства объекта)**, чтобы просмотреть свойства и входные/выходные выводы выбранного блока.

Автоматическое обнаружение ошибок

Описание

Редактор FBD также автоматически отображает место расположения ошибок компиляции. Чтобы перейти к блоку, где произошла ошибка компиляции, дважды щелкните по соответствующей строке на панели **Output (Вывод)**.

Раздел 10.4
Редактор релейных схем

Содержание этого раздела

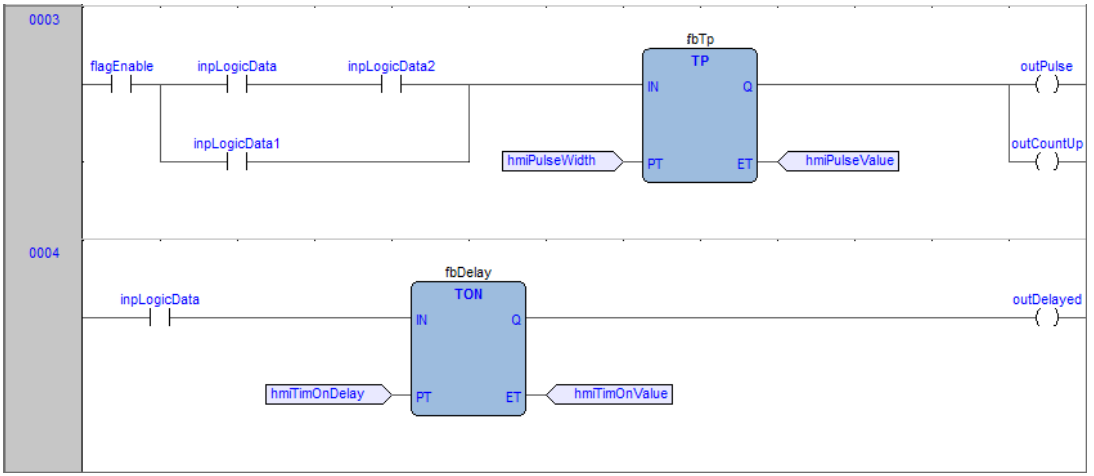
В этом разделе описаны следующие темы.

Тема	Страница
Обзор	237
Создание нового документа LD	237
Добавление/удаление сетей	238
Маркировка сетей	239
Вставка контактов	240
Вставка катушек	242
Вставка блоков	243
Редактирование свойств катушек и контактов	244
Редактирование сетей	244
Изменение свойств блоков	245
Получение информации о блоке	246
Автоматическое обнаружение ошибок	246
Вставка переменных	247
Вставка констант	247
Вставка выражения	248
Комментарии	249
Ветви	250

Обзор

Описание

Редактор релейных схем (LD) позволяет писать код и изменять программные модули на языке LD:



Для получения более подробной информации см. справочник по языку релейных схем *(стр. 414)*.

Создание нового документа LD

Описание

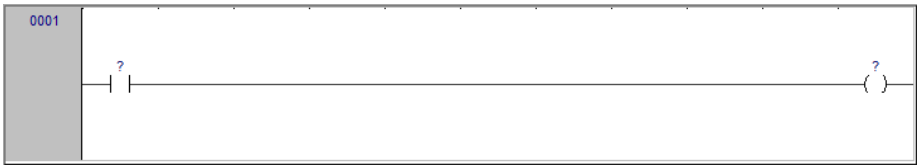
Создание и изменение документов FBD рассматривается в следующих разделах:

- Создание нового программного модуля *(см. стр. 184)*.
- Создание функционального блока / функции *(см. стр. 185)*.
- Редактирование программных модулей *(см. стр. 186)*

Добавление/удаление сетей

Описание

Каждый программный модуль, написанный на языке LD, состоит из последовательности сетей. Под сетью понимается набор взаимосвязанных графических элементов. Верхняя и нижняя границы каждой сети обозначаются двумя прямыми линиями, при этом каждая сеть ограничена слева серой областью, в которой указан номер сети.



В каждой сети LD в соответствии с обозначениями, принятыми в языке LD, имеются правая и левая шины питания (power rail).

В новой сети LD горизонтальная линия соединяет две шины питания. Она называется линией питания (power link). В этой линии питания размещены все элементы сети LD (контакты, катушки реле и блоки).

Вы можете выполнять следующие операции над сетями:

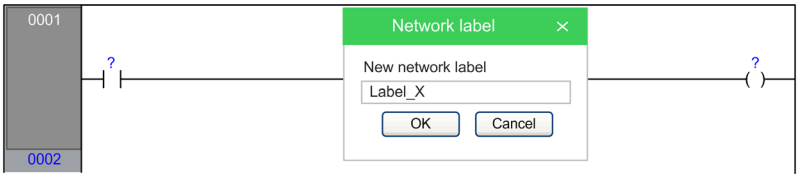
- Чтобы добавить новую пустую сеть, выберите пункт **Scheme (Схема)** → **Network (Сеть)** → **New (Создать)** или одну из кнопок на панели инструментов **Network (Сеть)**: .
- Чтобы отобразить фоновую сетку, помогающую при выравнивании объектов, выберите пункт **View (Вид)** → **Grid (Сетка)** .
- Чтобы добавить комментарий, выберите пункт **Scheme (Схема)** → **Object (Объект)** → **New Comment (Новый комментарий)** или нажмите **Shift+M**.

Маркировка сетей

Описание

Вы можете изменить стандартный порядок выполнения сетей, используя оператор прыжка, который передает управление программой в отмеченную сеть.

Чтобы назначить метку сети, выполните следующие действия.

Шаг	Действие
1	Выберите сеть.
2	Выполните одну из следующих операций: ● Выберите пункт **Scheme (Схема)** → **Network (Сеть)** → **Label (Метка)**. ● Дважды щелкните серую область, содержащую номер сети.
3	Откроется диалоговое окно, позволяющее ввести метку, которую необходимо связать с выбранной сетью: 
4	Нажмите кнопку **OK**. Метка отображается в верхнем левом углу выбранной сети:

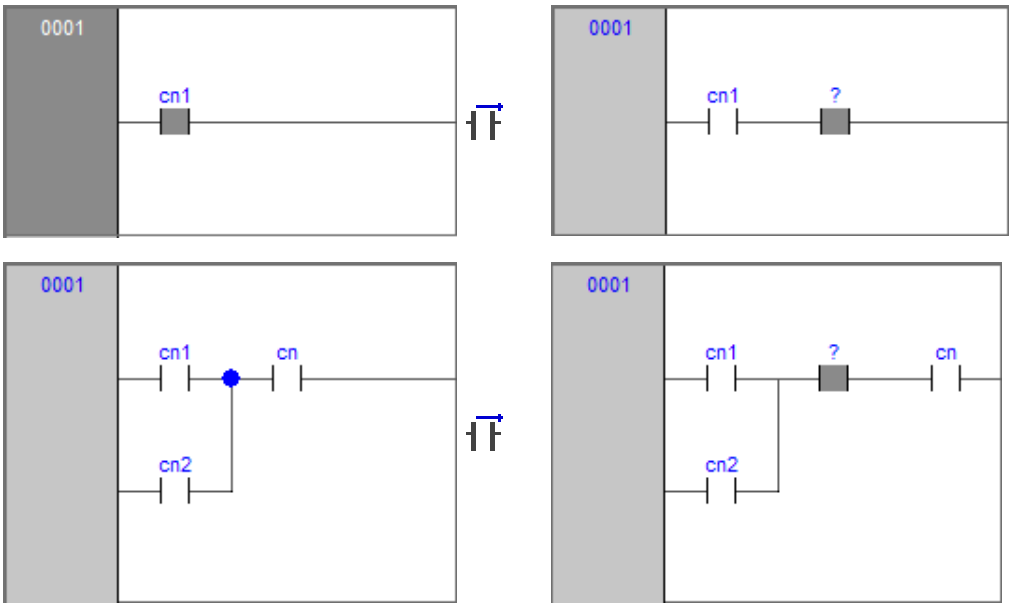
Вставка контактов

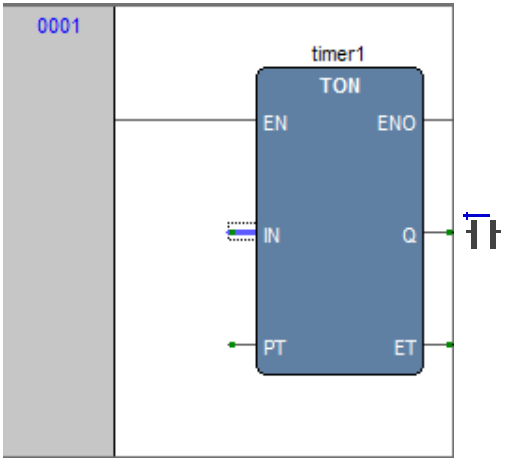
Описание

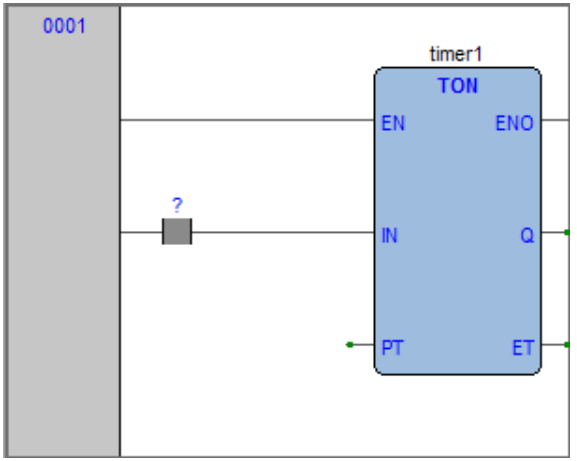
Чтобы вставить новые контакты в сеть, выполните одну из следующих процедур:

- Перетащите логическую переменную на нужный объект. Например, глобальные переменные можно перетащить из окна **Workspace (Рабочее пространство)**, а локальные переменные — выбрать в редакторе локальных переменных. Контакты, вставленные путем перетаскивания, всегда будут вставляться после объекта назначения.
- Выберите контакт, блок, вывод блока или точку соединения, выступающую в качестве точки вставки. Вставьте новый контакт, указав тип соединения (последовательное или параллельное) и позицию (до или после выделенного объекта) в меню **Scheme (Схема)** → **Object (Объект)** → **New (Создать)**.

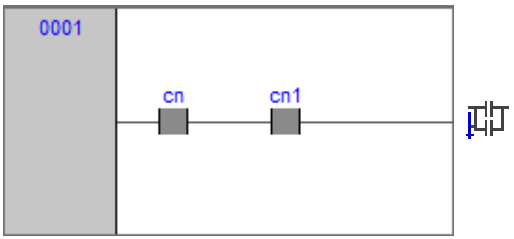
При последовательной вставке новый контакт вставляется слева или справа от выбранного контакта/блока или в середину выбранного соединения, в зависимости от элемента, выбранного перед вставкой. Примеры последовательной вставки приведены ниже.

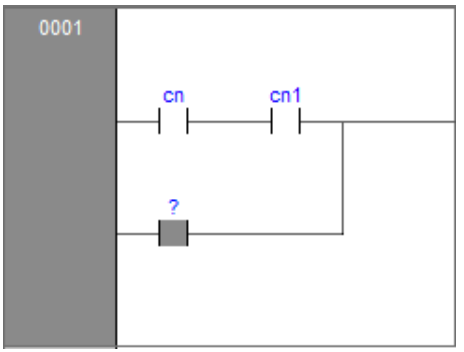


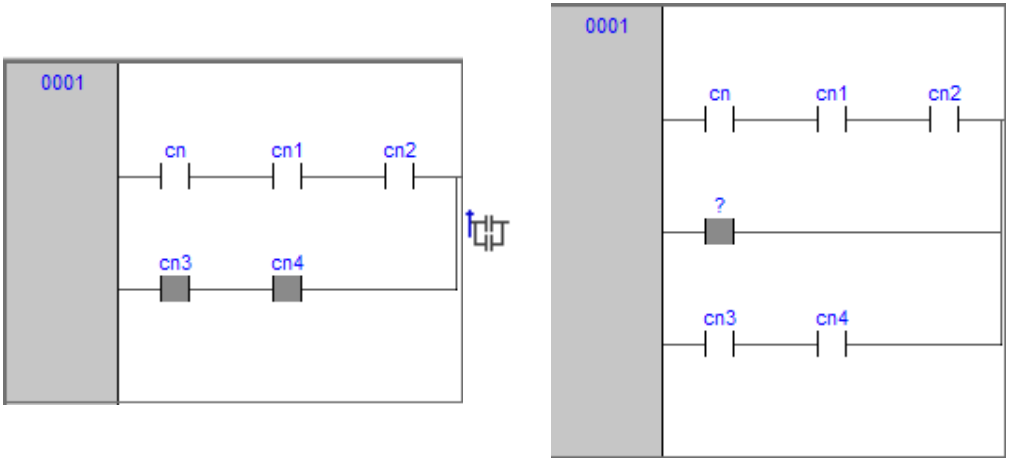




Для выполнения параллельной вставки можно предварительно выбрать несколько контактов; новый контакт вставляется выше или ниже группы выбранных контактов. Примеры параллельной вставки приведены ниже.





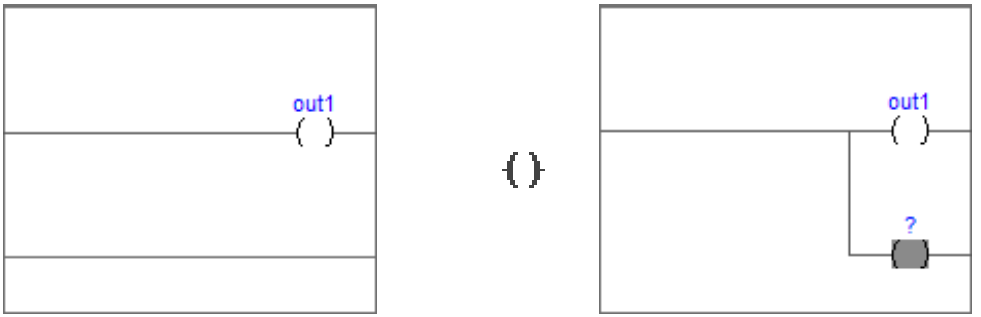


Вставка катушек

Описание

Чтобы вставить новые катушки в сеть, выполните одну из следующих процедур:

- Перетащите логическую переменную на имеющийся выход сети (катушка, возврат, прыжок). Например, глобальные переменные можно перетащить из окна **Workspace (Рабочее пространство)**, а локальные переменные — выбрать в редакторе локальных переменных.
- Выберите пункт **Scheme (Схема)** → **Object (Объект)** → **New (Создать)** → **Coil (Катушка)**. Новая катушка вставляется в правую шину питания. Если другие катушки, возврат или прыжки уже присутствуют в сети, новая катушка добавляется параллельно предыдущим элементам.



Вставка блоков

Описание

Чтобы вставить блоки в сеть, выполните одну из следующих операций:

- Выберите контакт, соединение или блок, затем выберите пункт **Scheme (Схема)** → **Object (Объект)** → **New (Создать)** → **Block (Блок)**, после чего откроется окно **Browser object (Обозреватель объектов)**. Выберите один пункт из списка.
- Перетащите выделенный объект из окна **Workspace (Рабочее пространство)**, окна **Libraries (Библиотеки)** или редактора локальных переменных на необходимое соединение.

Если объект имеет хотя бы один вход типа **BOOL** и один выход типа **BOOL**, то они подключены к линии питания (с помощью имеющейся команды можно будет впоследствии добавить выводы **EN/ENO**); в противном случае выводы **EN/ENO** добавляются автоматически.

Операторы, функции и функциональные блоки можно вставлять в сеть **LD** только на основную линию питания или на линию питания ветви (поэтому их нельзя вставлять параллельно контакту); также нельзя создавать контакт параллельно блоку.

Если блок имеет входной вывод типа **BOOL**, то перед ним можно создать другую логическую подсеть контактов и блоков; в противном случае к входным выводам, не являющимся **BOOL**, можно подключить только переменные, константы или выражения (которые тем не менее могут быть подключены к выводам **BOOL**).

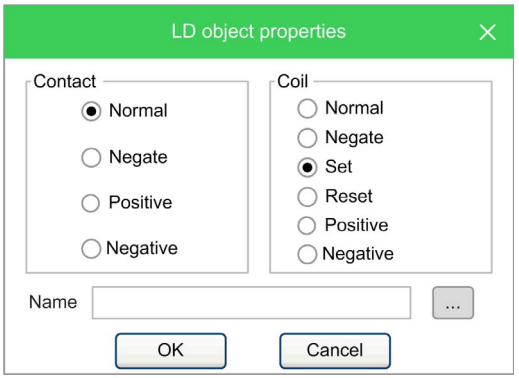
Редактирование свойств катушек и контактов

Описание

Тип контакта (нормальный, отрицаемый, положительный, отрицательный) или катушки (нормальный, отрицаемый, установленный, сброшенный, положительный, отрицательный) можно изменить с помощью одной из следующих операций:

- Дважды щелкните по элементу (контакту или катушке).
- Выберите элемент и нажмите клавишу **Enter**.
- Выберите элемент, активируйте всплывающее меню, затем выберите **Properties (Свойства)**.

Откроется соответствующее диалоговое окно. Выберите нужный тип элемента из отображаемого списка и нажмите **OK**.



В противном случае выберите нужный контакт или катушку и измените их тип с помощью шести кнопок на панели инструментов **LD** или шести команд в меню **Scheme (Схема)**.

Редактирование сетей

Описание

Редактор **LD** предоставляет следующие функции, характерные для большинства графических приложений, работающих на платформе Windows.

- Выделение блока.
- Выделение группы смежных контактов по нажатию **Ctrl + левая кнопка мыши** на каждом выбираемом контакте; если выделение распространяется на разные параллельные ветви, то в него автоматически добавляется больше контактов.
- Операции над одним или несколькими блоками: **Edit (Редактировать) → Cut (Вырезать)** , **Edit (Редактировать) → Copy (Копировать)** , **Edit (Редактировать) → Paste (Вставить)** .
- **Перетаскивание** выделенного объекта или группы внутри текущей сети или за ее пределы.

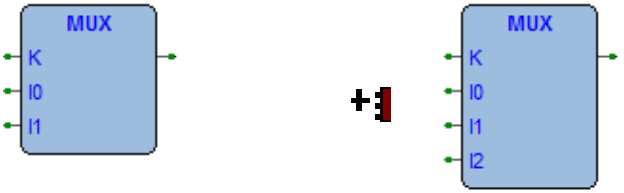
Добавление, перемещение, удаление или копирование/вставка объектов влечет за собой автоматический пересчет компоновки сетевых объектов; из-за этого невозможно вручную начертить соединительные линии или произвольно разместить объекты, не подсоединив их к сети.

Изменение свойств блоков

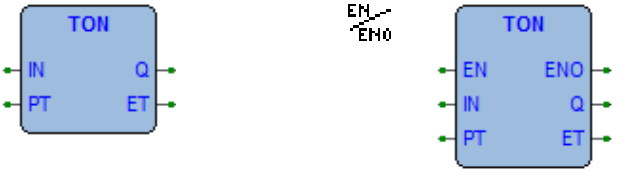
Описание

- Выберите пункт **Scheme (Схема) → Increment pins (Увеличить кол-во выводов)** , чтобы увеличить на единицу количество входных выводов для некоторых операторов и встроенных функций.

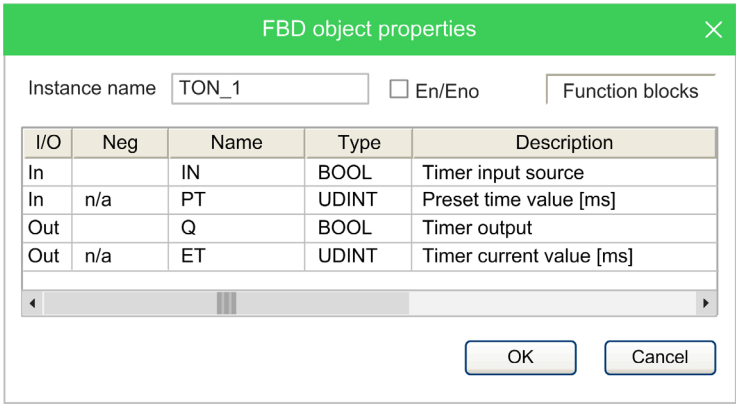
  ПРИМЕЧАНИЕ. Вы также можете удалить выводы, нажав кнопку **Decrease pins (Уменьшить кол-во выводов)**.



- Выберите пункт **Scheme (Схема) → Enable EN/ENO pins (Включить выводы EN/ENO)**, чтобы отобразить входные и выходные выводы. Выводы **EN/ENO** можно удалить только в том случае, если выбранный блок имеет хотя бы один вход типа **BOOL** и один выход типа **BOOL**; в противном случае они добавляются автоматически при создании блока, и удалить их будет невозможно (команда **Enable EN/ENO pins [Включить выводы EN/ENO]** отключена).
  Если блок имеет более одного выходного вывода типа **BOOL**, пользователь может выбрать, какой вывод будет выводить **сигнал** из блока и, таким образом, являться продолжением линии питания: выберите нужный выходной вывод и выберите в меню команду **Scheme (Схема) → Set output line (Задать выходную линию)** .



- Выберите пункт **Scheme (Схема) → Object properties (Свойства объекта)**, чтобы изменить имя экземпляра функционального блока.



Получение информации о блоке

Описание

Вы всегда можете получить информацию о блоке, добавленном в документ LD, выбрав его и выполнив одну из следующих операций:

- Выберите пункт **Scheme (Схема) → Object (Объект) → Open source (Открыть исходный код)** , чтобы открыть исходный код блока.
- Выберите в меню пункт **Scheme (Схема) → View PLC Object properties (Просмотр свойств объекта ПЛК)**, чтобы просмотреть свойства и входные/выходные выводы выбранного блока.

Автоматическое обнаружение ошибок

Описание

Редактор LD также автоматически отображает место расположения ошибок компиляции. Чтобы перейти к блоку, где произошла ошибка компиляции, дважды щелкните по соответствующей строке на панели **Output (Вывод)**.

Вставка переменных

Описание

Чтобы присоединить переменную к входному или выходному выводу блока, выполните одну из следующих процедур:

- Выберите вывод блока, а затем воспользуйтесь командой меню **Scheme (Схема) → Object (Объект) → New (Создать) → Variable (Переменная)**; затем дважды щелкните по новому объекту типа «переменная» (или нажмите клавишу **Enter**) и введите имя переменной.
- Перетащите выбранную переменную из окна **Workspace (Рабочее пространство)**, окна **Libraries (Библиотеки)** или редактора локальных переменных на нужный вывод блока.

Вставка констант

Описание

Чтобы присоединить числовую константу к входному выводу блока, укажите вывод и выберите пункт **Scheme (Схема) → Object (Объект) → New (Создать) → Constant (Константа)**, затем дважды щелкните новый объект типа «константа» (или нажмите клавишу **Enter**) и введите числовое значение константы.

Вставка выражения

Описание

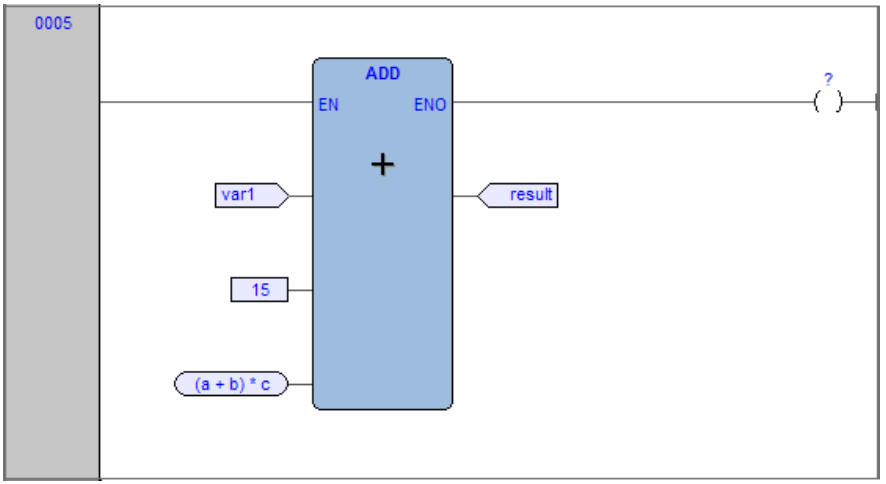
Чтобы присоединить сложное выражение к входному выводу блока, укажите вывод и выберите пункт **Scheme (Схема)** → **Object (Объект)** → **New (Создать)** → **Expression (Выражение)**, затем дважды щелкните новый объект типа «выражение» (или нажмите клавишу **Enter**) и введите выражение **ST**.

Например:

```
(a+b)*c
```

```
TO_INT(n)
```

```
ADR(x)
```

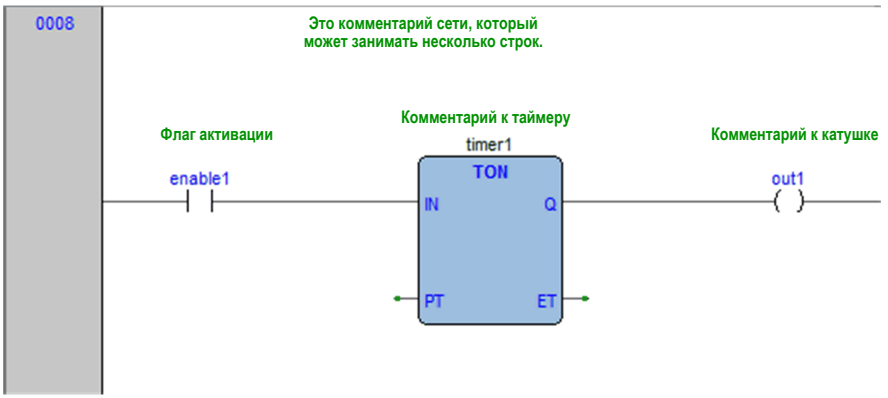


Комментарии

Описание

Предусмотрена возможность вставки двух типов комментариев:

- Комментарии к **сети**: активируйте сеть, щелкнув по заголовку слева или внутри сетки (но не выбирая ни одного объекта), а затем выберите пункт **Scheme (Схема) → Object (Объект) → New (Создать) → Comment (Комментарий)**. Комментарий к сети отображается в верхней части сети и при необходимости раскрывается для отображения всех строк.
- Комментарии к **объекту**: они активируются командой меню **View (Вид) → Show comments for objects (Показывать комментарии к объектам)**. Над любым контактом, функциональным блоком или катушкой изначально отображается описание связанной переменной ПЛК (если она присутствует). С помощью команды **Comment (Комментарий)** его можно изменить, введя соответствующий комментарий к объекту, заменив тем самым описание переменной ПЛК.



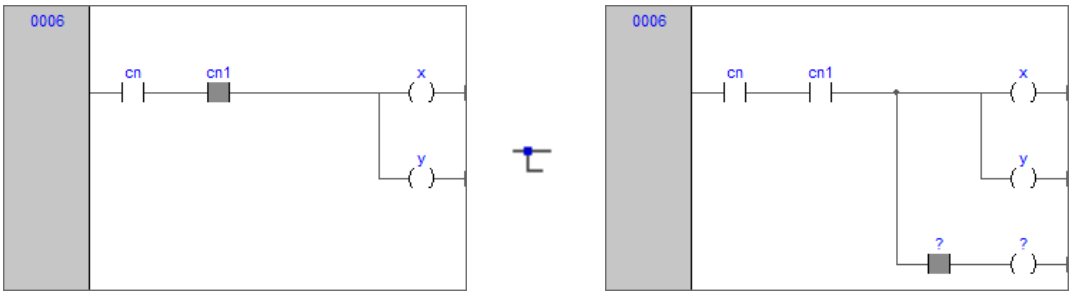
Ветви

Описание

Основная линия питания может разветвляться с целью создания подсетей, которые в дальнейшем могут ветвиться уже сами по себе. Чтобы добавить ветвь, выберите объект, после которого ее необходимо создать, а затем выберите в меню пункт **Scheme (Схема) → Object (Объект) → New (Создать) → Branch (Ветвь)**.

Начало новой ветви отмечается большой точкой в исходной строке; удаление объектов на ветви приводит к удалению самой ветви.

При выделении объекта на ветви выбирается и сама ветвь, поэтому, например, при выделении контакта на ветви и последующем выборе пункта **Scheme (Схему) → Object (Объект) → New (Создать) → Coil (Катушка)** катушка добавляется на эту ветвь, а не на основную линию питания.



Раздел 10.5
Редактор «Структурного текста»

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	251
Создание и редактирование объектов ST	252
Функции редактирования	252
Ссылка на объекты ПЛК	252
Автоматическое обнаружение ошибок	252
Закладки	253

Обзор

Описание

Редактор «Структурного текста» (ST) позволяет писать код и изменять программные модули на языке «Структурный текст»:

```
0001
0002 IqDW := sysIq * sysIq ;
0003 addIqSq := addIqSq + SHR( IqDW, 16#04 ) ;
0004
0005 IdDW := sysId * sysId ;
0006 addIdSq := addIdSq + SHR( IdDW, 16#04 ) ;
0007
0008 IF  a > b   THEN
0009     a := c;
0010     n := a * b * c;
0011 END_IF;
0012
```

Для получения более подробной информации см. справочник по языку «Структурный текст» *(стр. 419)*.

Создание и редактирование объектов ST

Описание

Создание и изменение документов FBD рассматривается в следующих разделах:

- Создание нового программного модуля *(см. стр. 184)*.
- Создание функционального блока / функции *(см. стр. 185)*.
- Редактирование программных модулей *(см. стр. 186)*.

Функции редактирования

Описание

Редактор ST предоставляет следующие функции, характерные для большинства редакторов, работающих на платформе Windows.

- Выделение текста.
- **Edit (Редактировать) → Cut (Вырезать)** .
- **Edit (Редактировать) → Copy (Копировать)** .
- **Edit (Редактировать) → Paste (Вставить)** .
- **Edit (Редактировать) → Replace (Заменить).**
- Перетаскивание выделенного текста.

Ссылка на объекты ПЛК

Описание

Если необходимо добавить в код ST ссылку на существующий объект ПЛК, у вас есть два варианта:

- Можно ввести непосредственно имя объекта ПЛК.
- Можно перетащить его в нужное место. Например, глобальные переменные можно перетащить из окна **Workspace (Рабочее пространство)**, встроенные функции — из окна **Libraries (Библиотеки)**, а локальные переменные можно выбрать в редакторе локальных переменных.

Автоматическое обнаружение ошибок

Описание

Редактор ST также автоматически отображает место расположения ошибок компиляции. Чтобы узнать, где произошла ошибка компиляции, дважды щелкните по соответствующей строке на панели **Output (Вывод)**.

Закладки

Описание

Вы можете установить закладки для часто используемых строк в исходном файле.
К установленным закладкам можно переходить с помощью клавиатуры. Ненужные закладки можно удалить.

Установка закладки

Переместите курсор в строку, которую нужно пометить закладкой, затем нажмите **Ctrl+F2**. Эта строка будет отмечена на полях голубым кружком:

Управление закладками осуществляется в разделе **Edit (Редактирование)** → **Bookmarks… (Закладки…)**. Доступные команды:

- **Add/toogle (Добавить/переключить)** (**Ctrl+F2**)
- **Next (Вперед)** (**F2**)
- **Prev (Назад)** (**Shift+F2**)
- **Remove all (Удалить все)**

Переход к следующей закладке

Нажимайте **F2**, пока не доберетесь до нужной строки.

Переход к предыдущей закладке

Нажимайте **Shift+F2**, пока не доберетесь до нужной строки.

Удаление закладки

Переместите курсор в любое место строки, содержащей закладку, а затем нажмите **Ctrl+F2**.

Раздел 10.6
Редактор последовательных функциональных схем

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	254
Создание нового документа SFC	254
Вставка нового элемента SFC	255
Присоединение элементов SFC	255
Назначение действия шагу	256
Указание условий перехода	258
Назначение переходу условного кода	259
Указание пункта назначения для прыжка	261
Редактирование сетей SFC	261

Обзор

Описание

Редактор последовательных функциональных схем (SFC) позволяет кодировать и изменять программные модули на языке SFC.

Для получения дополнительной информации о функциях редактора SFC см. раздел «Панель инструментов SFC» *(стр. 159)*.

Для получения более подробной информации см. справочник по последовательным функциональным схемам *(стр. 430)*.

Создание нового документа SFC

Описание

Создание и изменение документов FBD рассматривается в следующих разделах:

- Создание нового программного модуля *(см. стр. 184)*.
- Создание функционального блока / функции *(см. стр. 185)*.
- Редактирование программных модулей *(см. стр. 186)*.

Вставка нового элемента SFC

Описание

Вы можете вставить три типа элементов SFC:

- Выберите пункт **Scheme (Схема) → Object (Объект) → New (Создать) → Step (Шаг)**.
- Выберите пункт **Scheme (Схема) → Object (Объект) → New (Создать) → Transition (Переход)**.
- Выберите пункт **Scheme (Схема) → Object (Объект) → New (Создать) → Jump (Прыжок).**

В каждом отдельном случае курсор мыши меняется на:

для шагов;

для переходов;

для прыжков.

Присоединение элементов SFC

Описание

Следуйте этой процедуре для подключения блоков SFC:

- Выберите пункт **Edit (Редактировать) → Connection mode (Режим соединения)** или просто нажмите клавишу пробела на клавиатуре. Щелкните один раз на выводе источника, затем переместите указатель мыши на вывод назначения: редактор SFC нарисует логическую линию от первого пункта ко второму.
- Кроме того, вы можете включить режим автоматического соединения, выбрав **Scheme (Схема) → Auto connect (Автоматическое соединение)**. Затем возьмите два блока и перетащите их друг к другу, чтобы соответствующие выводы совпали. В результате редактор SFC автоматически нарисует между ними логическую линию.

Назначение действия шагу

Описание

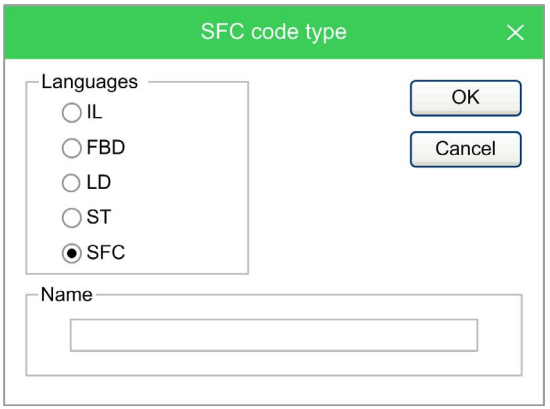
В этом разделе разъясняется, как реализовать то или иное действие и как назначить его тому или иному шагу.

Написание кода действия

Начните с открытия редактора, выполнив одну из следующих процедур:

- Выберите пункт **Scheme (Схема) → Code object (Объект кода) → New action (Новое действие)**.
- Щелкните правой кнопкой мыши имя программного модуля SFC в окне **Workspace (Рабочее пространство)**, выберите **New action (Новое действие)**.

В любом случае модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** отобразит следующее диалоговое окно:



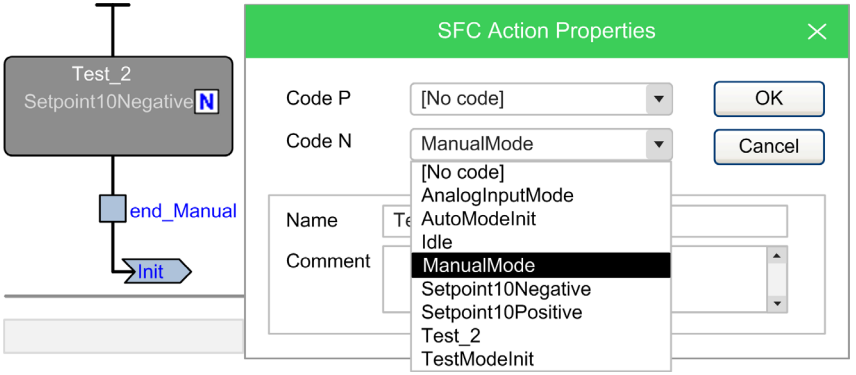
Выберите один из языков и введите имя нового действия в текстовом поле в нижней части диалогового окна. Затем либо подтвердите, нажав **OK**, либо отмените действие, нажав **Cancel (Отмена)**.

Если вы нажмете кнопку **OK**, модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** автоматически откроет редактор для языка, который вы выбрали в предыдущем диалоговом окне, и вам будет предложено ввести код для нового действия.

Вы не можете объявлять новые локальные переменные, так как модуль, который вы сейчас редактируете, является компонентом оригинального модуля SFC, т. е. программным модулем, в котором локальные переменные могут быть объявлены. Область действия локальных переменных распространяется на все действия и переходы, образующие схему SFC.

Назначение действия шагу

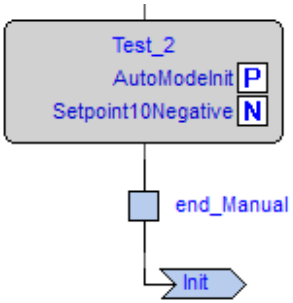
Закончив написание кода, дважды щелкните по шагу, которому необходимо назначить новое действие. В результате появится следующее диалоговое окно.



В списке, отображаемом в поле **Code N (Код N)**, выберите название действия, которое требуется выполнить, если шаг активен. Вы также можете выбрать из списка в поле **Code P (Код P)** (**Pulse** — импульс) название действия, которое необходимо выполнять каждый раз при активации шага (т. е. данное действие выполняется однократно при каждой активации шага, вне зависимости от количества циклов, в течение которых этот шаг активен). Подтвердите назначения, нажав **OK**.

В схеме SFC действия по назначению для шага представлены буквами на блоке шага:

- действие **N** с помощью буквы N;
- действие **P** с помощью буквы P.



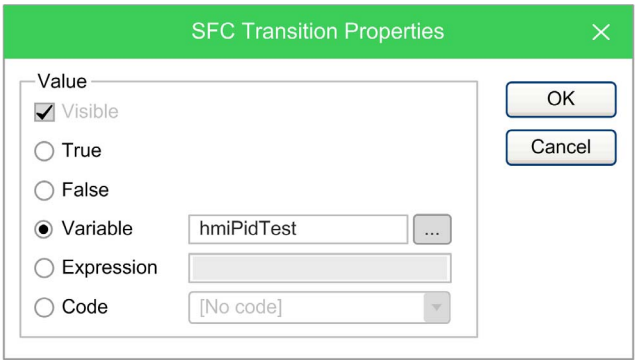
Если в дальнейшем необходимо будет отредактировать исходный код действия, то достаточно дважды щелкнуть по этим буквам. Кроме того, вы можете дважды щелкнуть по имени действия в папке **Actions (Действия)** окна **Workspace (Рабочее пространство)**.

Указание условий перехода

Описание

Условие перехода может определяется константой, переменной или фрагментом кода. В этом разделе рассматриваются первые два случая, а условный код обсуждается ниже.

Прежде всего дважды щелкните переход, которому вы хотите присвоить условие. В результате появится следующее диалоговое окно:



Выберите **True (Верно)**, если необходимо, чтобы этот переход постоянно очищался, или **False (Ложно)**, если хотите, чтобы программа ПЛК продолжала выполнять предыдущий блок.

Если же вы выберете **Variable (Переменная)**, то переход будет зависеть от значения логической переменной. Щелкните по соответствующему пункту, чтобы активировать текстовое поле справа и указать имя переменной.

Вы также можете воспользоваться обозревателем объектов, который можно открыть, нажав кнопку ... **Browse (Обзор)**.

Нажмите **OK** для подтверждения или **Cancel (Отмена)** для выхода без сохранения изменений.

Назначение переходу условного кода

Описание

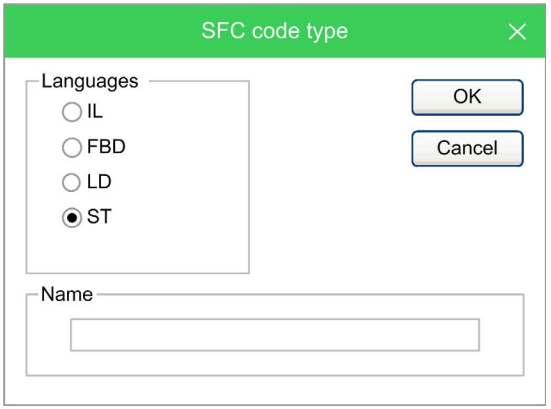
В этом разделе объясняется, как задать условие с помощью фрагмента кода и как присвоить его переходу.

Написание кода условия

Начните с открытия редактора, выполнив одну из следующих процедур:

- Нажмите **Scheme (Схема) → Code object (Объект кода) → New transition (Новый переход)**.
- Щелкните правой кнопкой мыши имя программного модуля SFC в окне **Workspace (Рабочее пространство)**, выберите **New transition (Новый переход)**.

В любом случае модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** отобразит следующее диалоговое окно:



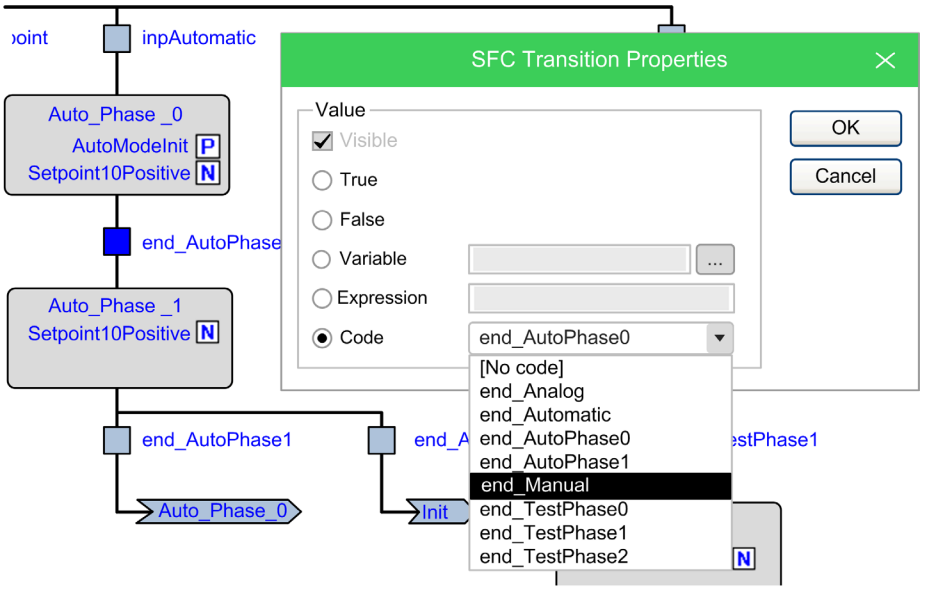
Выберите один из языков и введите имя нового условия в текстовом поле в нижней части диалогового окна. Затем либо подтвердите, нажав **OK**, либо отмените действие, нажав **Cancel (Отмена)**.

Если вы нажмете кнопку **OK**, модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** автоматически откроет редактор для языка, который вы выбрали в предыдущем диалоговом окне, где вы сможете ввести код нового условия.

Вы не можете объявлять новые локальные переменные, так как модуль, который вы сейчас редактируете, является компонентом оригинального модуля SFC, т. е. программным модулем, в котором локальные переменные могут быть объявлены. Область действия локальных переменных распространяется на все действия и переходы, образующие схему SFC.

Назначение условия переходу

Закончив написание кода, дважды щелкните по переходу, которому необходимо назначить новое условие. В результате появится следующее диалоговое окно:



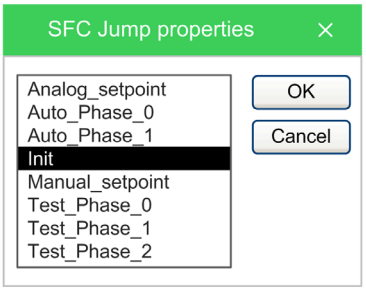
Выберите имя условия, которое вы хотите присвоить этому шагу. Затем подтвердите, нажав **OK**.

Если в дальнейшем вам понадобится отредактировать исходный код условия, вы можете дважды щелкнуть по имени перехода в папке **Transitions (Переходы)** окна **Workspace (Рабочее пространство)**.

Указание пункта назначения для прыжка

Описание

Чтобы указать пункт назначения для прыжка, дважды щелкните по блоку быстрого перехода в области **Chart (График)**. Откроется диалоговое окно, представленное ниже, в котором перечислены имена всех существующих шагов. Выберите шаг назначения, затем либо нажмите **OK** для подтверждения, либо **Cancel (Отмена)** для отмены действия.



Редактирование сетей SFC

Описание

Редактор SFC предоставляет следующие функции, характерные для большинства графических приложений, работающих на платформе Windows.

- Выделение блока.
- Выбор набора блоков комбинацией **Ctrl + левая кнопка мыши**.
- Операции над одним или несколькими блоками: **Edit (Редактировать) → Cut (Вырезать)** , **Edit (Редактировать) → Copy (Копировать)** , **Edit (Редактировать) → Paste (Вставить)** .
- Перетаскивание.

Раздел 10.7
Редактор переменных

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	262
Открытие редактора переменных	263
Создание новой переменной	264
Редактирование переменных	264
Удаление переменных	267
Сортировка переменных	267
Копирование переменных	267

Обзор

Описание

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** включает в себя графический редактор с интерфейсом для объявления и редактирования как глобальных, так и локальных переменных. Содержимое редактора преобразуется в синтаксически корректные исходные тексты в соответствии со стандартом МЭК 61131-3 (ГОСТ Р МЭК 61131-3).

В качестве примера рассмотрим содержимое редактора глобальных переменных, представленного на рисунке ниже:

	Name	Type	Address	Array	Init value	Attribute
1	gA	BOOL	Auto	No	TRUE	---
2	gB	REAL	Auto	[0...4]		---
3	gC	REAL	%MD60.20	No	1.0	---
4	gD	INT	Auto	No	-74	CONSTANT

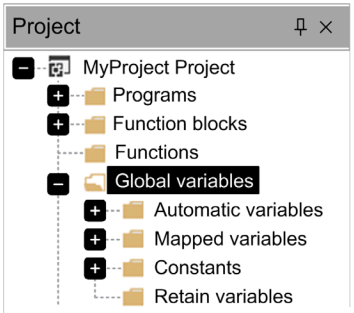
Соответствующий исходный код выглядит следующим образом:

```
VAR_GLOBAL
   gA : BOOL := TRUE;
   gB : ARRAY[ 0..4 ] OF REAL;
   gC AT %MD60.20 : REAL := 1.0;
END_VAR
VAR_GLOBAL CONSTANT
   gD : INT := -74;
END_VAR
```

Открытие редактора переменных

Открытие редактора глобальных переменных

Чтобы открыть редактор глобальных переменных, дважды щелкните левой кнопкой мыши папку **Global variables (Глобальные переменные)** в дереве проекта:

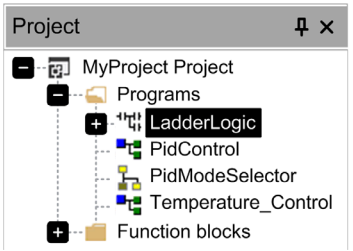


	Name	Type	Address
1	parCtDownPreset	UDINT	Auto
2	parCtUpPreset	UDINT	Auto
3	parPulseValue	UDINT	Auto
4	parPulseWidth	UDINT	Auto
5	parTimOnDelay	UDINT	Auto
6	parTimOnValue	UDINT	Auto
7	pidKD	REAL	Auto
8	pidKI	REAL	%MD1

ПРИМЕЧАНИЕ. При использовании настраиваемого рабочего пространства *(см. стр. 165)* глобальные переменные доступны с помощью значка группы **Global variables (Глобальные переменные)**.

Открытие редактора локальных переменных

Чтобы открыть редактор локальных переменных, дважды щелкните по программному модулю, где содержится локальные переменные, которые требуется отредактировать:



Local variables

	Name	Type	Address
1	fbDelay	TON	Auto
2	fbCtu	CTU_DINT	Auto
3	fbCtd	CTU_DINT	Auto
4	fbTp	TP	Auto
5	ettp	UDINT	Auto

ПРИМЕЧАНИЕ. При использовании настраиваемого рабочего пространства *(см. стр. 165)* локальные переменные доступны в программном модуле в группе **Local variables (Локальные переменные)**.

Создание новой переменной

Описание

Чтобы создать новую переменную, выполните одну из следующих операций:

- Выберите в меню пункт **Variables (Переменные)** → **Insert (Вставить)**.
- Нажмите на панели инструментов проекта.
- Нажмите сочетание **Ctrl+Shift+Insert**.

Редактирование переменных

Описание

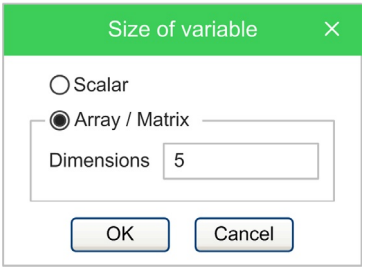
Выполните следующие действия, чтобы отредактировать объявление переменной в редакторе переменных (следующие шаги не обязательны, и при редактировании переменной большинство из них не актуальны).

<table>
<tr><th>Шаг</th><th>Действие</th></tr>
<tr><td>1</td><td>Отредактируйте имя переменной, введя новое имя в соответствующую ячейку:
<table>
<tr><th></th><th>Name</th><th>Type</th><th>Address</th><th>Group</th><th>Array</th></tr>
<tr><td>1</td><td>gA</td><td>BOOL</td><td>Auto</td><td>Ungrouped_vars</td><td>No</td></tr>
<tr><td>2</td><td>gB</td><td>REAL</td><td>Auto</td><td>Ungrouped_vars</td><td>[0...4]</td></tr>
<tr><td>3</td><td>gC</td><td>REAL</td><td>%MD60.20</td><td>MyMapped_vars</td><td>No</td></tr>
<tr><td>4</td><td>gD</td><td>INT</td><td>Auto</td><td>MyConstants_vars</td><td>No</td></tr>
</table></td></tr>
<tr><td>2</td><td>Измените тип переменной, либо отредактировав имя типа в соответствующей ячейке, либо щелкнув по кнопке в этой ячейке и выбрав нужный тип из списка:
<table>
<tr><th></th><th>Name</th><th>Type</th><th>Address</th><th>Group</th></tr>
<tr><td>1</td><td>gA</td><td>BOOL</td><td>Auto</td><td>Ungrouped_vars</td></tr>
<tr><td>2</td><td>gB</td><td>REAL</td><td>Auto</td><td>Ungrouped_vars</td></tr>
<tr><td>3</td><td>gC</td><td>REAL</td><td>%MD60.20</td><td>MyMapped_vars</td></tr>
<tr><td>4</td><td>gD</td><td>INT</td><td>Auto</td><td>MyConstants_vars</td></tr>
</table></td></tr>
</table>

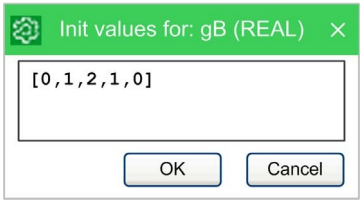
Шаг	Действие
3	Отредактируйте адрес переменной, нажав кнопку в соответствующей ячейке и введя необходимую информацию в появившемся окне. В случае глобальных переменных эта операция может привести к изменению положения переменной в дереве проекта: | | Name | Type | Address | Group | | 1 | gA | BOOL | Auto | Ungrouped_vars | | 2 | gB | REAL | Auto | Ungrouped_vars | | 3 | gC | REAL | %MD60.20 | MyMapped_vars | | 4 | gD | INT | Auto | MyConstants_vars | Variable address Automatic address Size: Bit / Byte (8 bit) / Word (16 bit) / Double word (32 bit) Location: Input / Output / Memory Data block 60 . Index 20 OK Cancel
4	В случае глобальных переменных можно назначить переменную группе, выбрав ее из списка, открывающегося при щелчке по соответствующей ячейке. Эта операция может привести к изменению положения переменной в дереве проекта: | | Name | Type | Address | Group | Array | | 1 | gA | BOOL | Auto | Ungrouped_vars | No | | 2 | gB | REAL | Auto | [redacted] | [0...4] | | 3 | gC | REAL | %MD60.20 | MyConstants_vars | No | | 4 | gD | INT | Auto | MyMapped_vars | No | Ungrouped_vars

Шаг	Действие
5	Выберите, является ли переменная массивом или нет. Если да, отредактируйте размер переменной:

	Name	Type	Address	Group	Array	Init value
1	gA	BOOL	Auto	Ungrouped_vars	No	TRUE
2	gB	REAL	Auto	Ungrouped_vars	[0...4]	
3	gC	REAL	%MD60.20	MyMapped_vars	No	1.0
4	gD	INT	Auto	MyConstants_vars	No	-74



6	Отредактируйте первоначальные значения переменной: нажмите кнопку в соответствующей ячейке и введите значения в появившееся окно:



7	Присвойте переменной атрибут (например, `CONSTANT` или `RETAIN`), выбрав его из списка, который открывается при щелчке по соответствующей ячейке:

	Name	Type	Address	Group	Array	Init value	Attribute
1	gA	BOOL	Auto	Ungrouped_vars	No	TRUE	---
2	gB	REAL	Auto	Ungrouped_vars	[0...4]	[0,1,2,1,0]	---
3	gC	REAL	%MD60.20	MyMapped_vars	No	1.0	CONSTANT
4	gD	INT	Auto	MyConstants_vars	No	-74	RETAIN

8	Введите описание переменной в соответствующую ячейку. В случае глобальных переменных эта операция может привести к изменению положения переменной в дереве проекта:

	Name	Type	Address	Group	Array	Init value	Attribute	Description
1	gA	BOOL	Auto	Ungrouped_vars	No	TRUE	---	Global variable A
2	gB	REAL	Auto	Ungrouped_vars	[0...4]	[0,1,2,1,0]	---	Global variable B
3	gC	REAL	%MD60.20	MyMapped_vars	No	1.0	---	Global variable C
4	gD	INT	Auto	MyConstants_vars	No	-74	CONSTANT	Global variable D

9	Сохраните проект, чтобы зафиксировать изменения, внесенные в объявление переменной.

Удаление переменных

Описание

Чтобы удалить одну или несколько переменных, выберите их в редакторе. Для выбора нескольких элементов можно использовать клавиши **Ctrl** или **Shift**:

	Name	Type	Address	Group	Array	Init value
1	G_I_iTe	INT	Auto	Ungrouped_vars	No	
2	G_I_iSe	INT	Auto	Ungrouped_vars	No	
3	G_I_iDi	INT	Auto	Ungrouped_vars	No	1
4	G_I_iC	BOOL	Auto	Ungrouped_vars	No	
5	G_I_iAI	BOOL	Auto	Ungrouped_vars	No	
6	gA	BOOL	Auto	Ungrouped_vars	No	TRUE
7	gB	REAL	Auto	Ungrouped_vars	[0...4]	[0,1,2,1,0]
8	gC	REAL	%MD60.20	MyMapped_vars	No	1.0
9	gD	INT	Auto	MyConstants_vars	No	-74

Удалите выбранные переменные, выполнив одну из следующих операций:

- Выберите в меню пункт **Variables (Переменные)** → **Delete (Удалить)**.
- Нажмите на панели инструментов проекта.
- Нажмите клавишу **Delete**.

Для `FUNCTION` невозможно удалить `RESULT` в соответствии со стандартом МЭК 61131-3 (ГОСТ Р МЭК 61131-3).

Сортировка переменных

Описание

Вы можете отсортировать переменные в редакторе, щелкнув по заголовку столбца поля, которое вы хотите использовать в качестве критерия сортировки.

Копирование переменных

Описание

Редактор переменных позволяет копировать и вставлять элементы. Вы можете использовать сочетания клавиш или команды в меню **Edit (Редактировать)** → **Copy (Копировать)**, **Edit (Редактировать)** → **Paste (Вставить)** .

ПРИМЕЧАНИЕ. При копировании сопоставленных переменных могут возникать проблемы с наложением адресов. Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** может автоматически присваивать вставленной переменной доступный адрес и устранять возможный конфликт. Чтобы активировать эту функцию, обратитесь к разделам «Параметры программного обеспечения» *(см. стр. 47)* и «Функция объединения» *(см. стр. 179)*.

Глава 11
Компиляция

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
11.1	Обзор	270
11.2	Компиляция проекта	271
11.3	Вывод компилятора	273
11.4	Компилятор с поддержкой командной строки	276

Раздел 11.1
Обзор

Обзор

Описание

Компиляция заключается в трансляции исходного кода ПЛК на другой язык программирования (целевой язык) — например, в двоичный код, который может выполнять процессор целевого устройства.

Раздел 11.2
Компиляция проекта

Содержание этого раздела

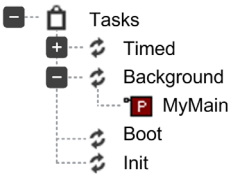
В этом разделе описаны следующие темы.

Тема	Страница
Обзор	271
Загрузка файла образа	272

Обзор

Обязательное условие

Перед началом компиляции убедитесь, что задаче назначена хотя бы одна программа:



В противном случае компиляция прерывается и выводится подробное сообщение об ошибке:

Компиляция проекта

Запустите компиляцию проекта, применив одну из следующих операций:

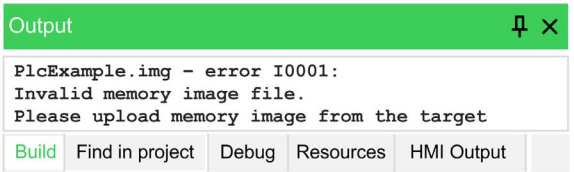
- Выберите в меню пункт **Project (Проект) → Compile (Компилировать)**.
- Нажмите на панели инструментов проекта.
- Нажмите клавишу **F7**.

ПРИМЕЧАНИЕ. Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** автоматически сохраняет изменения в проекте перед запуском компиляции.

Загрузка файла образа

Описание

Перед выполнением компиляции компилятору необходимо загрузить файл образа (файл ***.img**), содержащий карту памяти целевого устройства. Если целевое устройство подключено, то при запуске компиляции компилятор ищет файл образа непосредственно на целевом устройстве. В противном случае он загружает локальную копию файла образа из рабочей папки. Если целевое устройство отключено и локальная копия файла образа отсутствует, компиляция не производится: в этом случае необходимо подключиться к рабочему целевому устройству.



Раздел 11.3
Вывод компилятора

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	273
Ошибки компилятора	274

Обзор

Описание

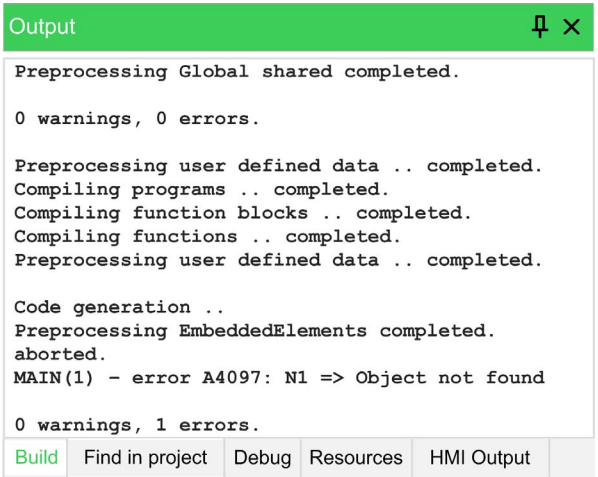
По завершении предыдущего шага компилятор выполняет компиляцию, а затем выводит отчет в окне **Output (Вывод)**. Последняя строка отчета имеет следующий формат:
`m warnings, n errors` (Предупреждений: m, ошибок: n)

Условие	Описание
`n > 0`	Ошибки компиляции. Код ПЛК содержит одну или несколько серьезных ошибок, которые нужно исправить. Исполняемый код не удалось сгенерировать.
`n = 0, m > 0`	Вывод предупреждений. В коде ПЛК содержится одна или несколько незначительных ошибок, которые компилятор обработал автоматически. Тем не менее вы проинформированы о том, что программа ПЛК может работать не так, как задумано. Такие незначительные ошибки следует исправлять, редактируя и перекомпилируя приложение до тех пор, пока компилятор не перестанет выводить соответствующие сообщения.
`n = m = 0`	Компиляция прошла успешно, так как в отчете отсутствуют ошибки и предупреждения.

Ошибки компилятора

Описание

Если в приложении имеется одна или несколько ошибок, то в окне **Output (Вывод)** отображается информация по каждой из обнаруженных ошибок.

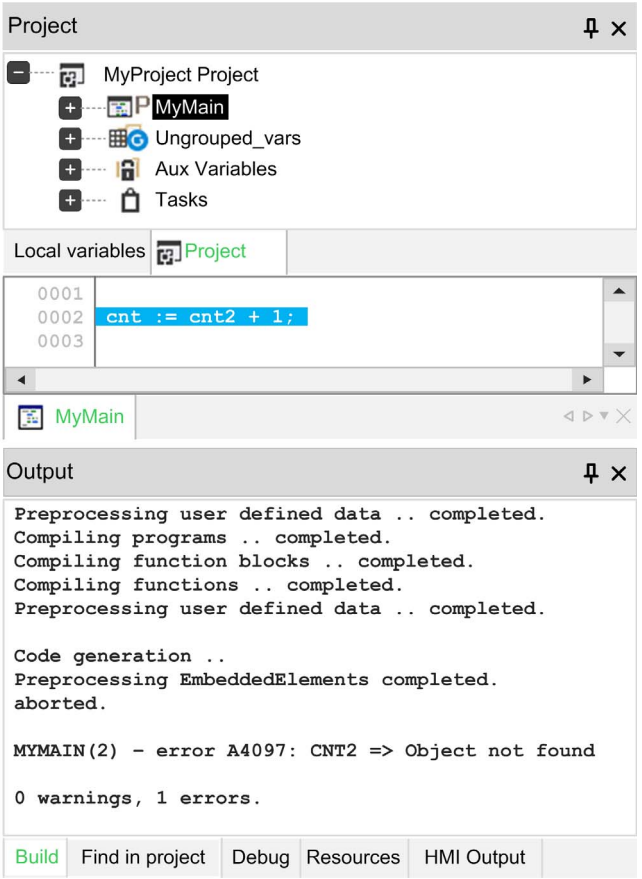


Для каждой обнаруженной ошибки выводится следующая информация:

- имя программного модуля, затронутого ошибкой;
- номер строки с исходным кодом, которая привела к ошибке;
- тип ошибки:
  - `error` — серьезная ошибка;
  - `warning` — незначительная ошибка;
- код ошибки;
- описание ошибки.

Для получения дополнительной информации см. раздел «Сообщения об ошибках компиляции» *(стр. 446)*.

Если дважды щелкнуть сообщение об ошибке в окне **Output (Вывод)**, модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** откроет исходный код и выделит строку, содержащую обнаруженную ошибку:



Раздел 11.4
Компилятор с поддержкой командной строки

Обзор

Описание

Компилятор можно использовать независимо от среды разработки: в каталоге EcoStruxure Machine Expert - HVAC имеется исполняемый файл **EWc.exe**, который можно запустить (например, с помощью пакетного файла) с рядом параметров.

Для получения информации о синтаксисе и настройках этого инструмента командной строки запустите исполняемый файл без параметров.

Глава 12
Запуск приложения

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
12.1	Обзор	278
12.2	Настройка связи	279
12.3	Онлайн-статус	287
12.4	Загрузка приложения	289
12.5	Моделирование	290
12.6	Управление выполнением приложения ПЛК	291

Раздел 12.1
Обзор

Обзор

Описание

Для загрузки и отладки приложения необходимо установить соединение с целевым устройством. Эта глава посвящена операциям, необходимым для подключения к целевому устройству и для загрузки приложения. Отдельная глава посвящена отладке *(см. стр. 293)*.

Раздел 12.2
Настройка связи

Содержание этого раздела

В этом разделе описаны следующие темы.

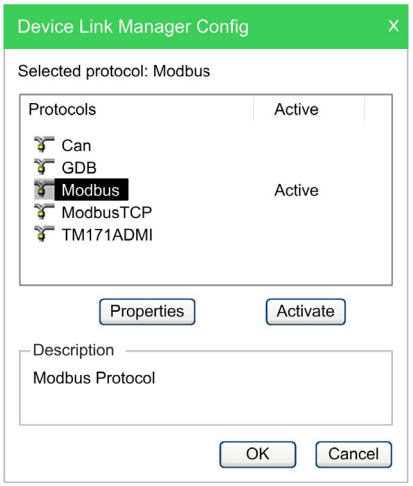
Тема	Страница
Обзор	280
Сохранение последнего использованного порта связи	286

Обзор

Описание

Для установления соединения с целевым устройством проверьте все соединения и связь в сети.

Чтобы настроить и установить соединение с целевым устройством, выполните следующие действия.

Шаг	Действие
1	Перейдите в меню **On-line (Онлайн) → Set up communication… (Настройка связи…)** на вкладке **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**. В результате появится следующее диалоговое окно. 
2	Выберите подходящий протокол: ● CAN *(см. стр. 281)* ● GDB *(см. стр. 281)* ● Modbus *(см. стр. 281)* ● Modbus TCP *(см. стр. 282)* ● TM171ADMI *(см. стр. 284)*
3	Нажмите кнопку **Activate (Активировать)**, чтобы активировать протокол.
4	Нажмите кнопку **Properties (Свойства)**, чтобы изменить свойства активированного протокола.
5	Установите все необходимые параметры для данного протокола. Например, адрес или тайм-аут связи (как долго модуль **PROGRAMMING [ПРОГРАММИРОВАНИЕ]** должен ждать ответа от целевого устройства, прежде чем отобразится сообщение об ошибке связи).
6	Нажмите кнопку **OK**, чтобы применить изменения, внесенные в настройки связи.

Теперь можно установить связь, выбрав в меню пункт **On-line (Онлайн) → Connect (Подключить)**.

CAN

Выберите **CAN** в случае соединения CAN.

Параметр	Описание	Значения	По умолчанию	Примечание
Baud_CAN_OB	Скорость передачи данных встроенного интерфейса CAN	2 = 500 кбод 3 = 250 кбод 4 = 125 кбод 5 = 125 кбод 6 = 50 кбод	2	–
Addr_CAN_OB	Последовательный адрес встроенного интерфейса CAN	1–127	1	Адрес определяется суммой этого значения и значений DIP-переключателей.

GDB

В контроллерах M171/M172 протокол GDB не реализован.

Протокол GDB зарезервирован для внутреннего использования программным обеспечением.

Modbus

Выберите **Modbus** в случае соединения USB/RS485.

Параметр	Описание	Значения	По умолчанию	Примечание
Baud_RS485_OB	Скорость передачи данных встроенного интерфейса Modbus	0 = 9600 бод 1 = 19 2 00 бод 2 = 38 400 бод 3 = 57 600 бод 4 = 76 800 бод 5 = 115 200 бод	2	–
Addr_RS485_OB	Последовательный адрес встроенного интерфейса RS485	1–255	1	Адрес определяется суммой этого значения и значений DIP-переключателей.
Proto_RS485_OB	Выбор протокола встроенного интерфейса RS485	2 = uNET 3 = Modbus/RTU 4 = BACnet MS/TP	3	–
Databit_RS485_OB	Номер бита данных встроенного интерфейса RS485	8	8	Постоянный: 8
Stopbit_RS485_OB	Номер стопового бита встроенного интерфейса RS485	1 = 1 стоповый бит 2 = 2 стоповых бита	1	–
Parity_RS485_OB	Бит четности встроенного интерфейса RS485	0 = NULL 1 = ODD 2 = EVEN	2	–

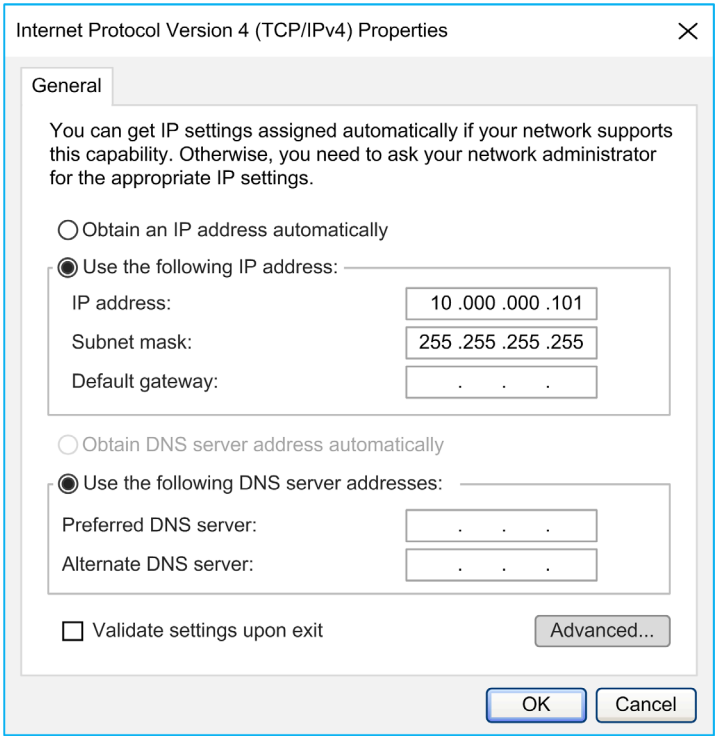
Modbus TCP

Выберите протокол **Modbus TCP** в случае Ethernet-соединения, при необходимости используя соответствующий коммуникационный модуль.
В окне свойств протокола:

- поле **IP or hostname (IP-адрес или имя хоста)** предназначено для ввода либо IP-адреса (по умолчанию для M171P/M172P установлен 10.0.0.100), либо имени хоста в локальной сети;
- по умолчанию в поле настройки порта связи **TCP/IP** установлено значение 502.

Подключите Ethernet-кабель компьютера к M171P/M172P.

Настройте TCP/IPv4-соединение в свойствах порта Ethernet вашего компьютера, указав адрес (10.0.0.101):



ПРИМЕЧАНИЕ. По умолчанию для M171P/M172P используется адрес 10.0.0.100, поэтому для порта Ethernet на персональном компьютере следует указать адрес, отличный от адреса по умолчанию (например, 10.0.0.101, т. е. первые три октета должны быть одинаковыми, а четвертый — отличаться).

Нажмите кнопку **OK**: персональный компьютер настроен на обмен данными с M171P/M172P через порт Ethernet.

M171P/M172P имеет ряд параметров BIOS для управления соединением между целевым устройством и EcoStruxure Machine Expert - HVAC, но в отличие от M171O, здесь нет меню по умолчанию, отображаемого на встроенном или выносном дисплее.

Подключаемый модуль пассивного адаптера Ethernet

Параметры конфигурации пассивного адаптера Ethernet включают в себя следующие настройки: порт связи TCP/IP (например, 502), IP-адрес, шлюз и маска подсети.

Параметры шлюза по умолчанию не актуальны в локальной сети «точка — точка».

Для подключения через роутер параметры шлюза по умолчанию должны быть установлены в соответствии с конфигурацией сети, как в следующем примере.

Параметр	Описание	Значение	Параметр	Описание	Значение
Ip_1_ETH_PI	Пассивный адаптер Ethernet, IP-адрес (первая часть)	192	DefGtwy_1_ETH_PI	Шлюз по умолчанию (первая часть)	192
Ip_2_ETH_PI	Пассивный адаптер Ethernet, IP-адрес (вторая часть)	168	DefGtwy_2_ETH_PI	Шлюз по умолчанию (вторая часть)	168
Ip_3_ETH_PI	Пассивный адаптер Ethernet, IP-адрес (третья часть)	0	DefGtwy_3_ETH_PI	Шлюз по умолчанию (третья часть)	0
Ip_4_ETH_PI	Пассивный адаптер Ethernet, IP-адрес (четвертая часть)	100	DefGtwy_4_ETH_PI	Шлюз по умолчанию (четвертая часть)	1

Специальные настройки ЧМИ (M171P с креплением для скрытого монтажа)

Помимо настроек BIOS, в M171P с креплением для скрытого монтажа имеется дополнительное меню ЧМИ.

Параметр	Описание	Значения	По умолчанию	Примечание
Hmi_language	Язык дисплея (локального/ удаленного)	0–65 535	0	–
HMIList_current	Текущий ЧМИ	0 = удаленный ЧМИ 1 / 1 = удаленный ЧМИ 2 2 = удаленный ЧМИ 3 / 3 = удаленный ЧМИ 4 4 = удаленный ЧМИ 5 / 5 = удаленный ЧМИ 6 6 = удаленный ЧМИ 7 / 7 = удаленный ЧМИ 8 8 = удаленный ЧМИ 9 / 9 = удаленный ЧМИ 10 10 = не используется 11 = локальный ЧМИ	11	Локальный ЧМИ идентифицируется на дисплее в виде сети с подключением как HMI (ЧМИ), при этом удаленный ЧМИ идентифицируется в подключении как Remote HMI (Удаленный ЧМИ).

Доступно десять удаленных меню. Ниже перечислены параметры первого меню. Остальные меню имеют схожую структуру.

Параметр	Описание	Значения	По умолчанию	Примечание
HmiList_ID_1	Удаленный ЧМИ 1, список ID для навигации	0–254	0	–
HmiList_Res_1	Удаленный ЧМИ 1, тип ресурса для навигации	1 = RTU (RS485 Modbus RTU) 2 = TCP (Modbus TCP) 3 = CAN (CAN)	3 = CAN	–
HmiList_Addr_1	Удаленный ЧМИ 1, адрес ресурса для навигации для CAN, RTU и TCP (IP-адрес, часть 1)	0–255	0	Например: CAN: 2,500000 RS485: 1.38400.P81 Modbus TCP: 010.000.000.100
HmiList_Addr_2	Удаленный ЧМИ 1, адрес ресурса для навигации для TCP (IP-адрес, часть 2)	0–255	0	
HmiList_Addr_3	Удаленный ЧМИ 1, адрес ресурса для навигации для TCP (IP-адрес, часть 3)	0–255	0	
HmiList_Addr_4	Удаленный ЧМИ 1, адрес ресурса для навигации для TCP (IP-адрес, часть 4)	0–255	0	
HmiList_File_1	Удаленный ЧМИ, файл навигации 1 (DOS 8.3, в верхнем регистре)	Буквенно-числовая строка, 8 символов	********	По умолчанию используется имя HMIREM.KBD.

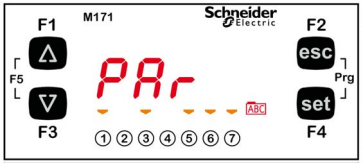
TM171ADMI

Выберите TM171ADMI в случае соединения с M171O с помощью кабеля для программирования TM171ADMI.

M171O содержит в папке **CF** в контроллере параметры для управления соединением между целевым устройством и EcoStruxure Machine Expert - HVAC.

Если целевое устройство не указано, например в устройстве нет приложения контроллера, то M171O выводит на дисплей сообщение FrEE . В остальных случаях при наличии приложения контроллера на M171O на дисплее выводится сообщение ПЛК. Для просмотра сообщения одновременно нажмите клавиши **ВВЕРХ** и **ВНИЗ**.

Чтобы изменить параметр, выполните следующие действия.

Шаг	Действие	Результат
1	Находясь на главном экране, одновременно нажмите кнопки **set** и **esc**, чтобы открыть меню программирования: 	Откроется меню программирования. Отобразится название первой подпапки (в данном случае **PAr**): 
2	Нажмите кнопку **set**, чтобы открыть меню Parameters (Параметры).	Отобразится название первой подпапки (**CL**).
3	Нажимайте кнопки **ВВЕРХ** и **ВНИЗ** для прокрутки к пункту **CF**.	–
4	Нажмите кнопку **set**, чтобы открыть папку.	Отобразится название первого параметра.
5	Нажимайте кнопки **ВВЕРХ** и **ВНИЗ** для прокрутки различных параметров и выбора параметров подключения.	–
6	Нажмите кнопку **set**, чтобы просмотреть значение параметра.	Отобразится значение параметра.
7	Нажимайте кнопки **ВВЕРХ** и **ВНИЗ** для изменения этого значения.	–
8	Нажмите кнопку **set**, чтобы подтвердить новое значение параметра. **ПРИМЕЧАНИЕ.** Нажмите **esc**, чтобы вернуться в предыдущую папку без сохранения введенного значения.	–
9	Нажмите кнопку **esc**, чтобы вернуться на главный экран.	–
10	С помощью кнопок **ВВЕРХ** и **ВНИЗ** выберите остальные параметры и повторите процедуру (шаги 5–9) для просмотра значений и (при необходимости) их редактирования.	–

Значения параметров, необходимые для правильного соединения между целевым устройством M171O и EcoStruxure Machine Expert - HVAC.

Параметр	Описание	Значения	По умолчанию	Уровень видимости параметра	Примечание
CF01 (1)	Выберите протокол COM1 (TTL)	0 = резерв 1 = Modbus	1	2	Необходимо установить на 1.
CF30	Адрес контроллера протокола Modbus	1–255	1	3	Убедитесь, что установленные значения соответствуют значениям, заданным на вкладке **On-line (Онлайн)** → **Set up (Настройка)** → **Communication (Связь)** → **Properties (Свойства)**.
CF31 (2)	Скорость передачи данных по протоколу Modbus	0, 1, 2 = не используется 3 = 9600 бод 4 = 19 200 бод 5 = 38 400 бод 6 = 57 600 бод 7 = 115 200 бод	3	3	
CF32	Бит четности контроллера протокола Modbus	1 = EVEN 2 = NONE 3 = ODD	1	3	
(1) COM1 = порты TTL и RS485 не могут использоваться одновременно.					
(2) CF31	5 = 38400 бод (RS485: не поддерживается) 6 = 57600 бод (RS485: не поддерживается) 7 = 115200 бод (RS485: не поддерживается)				

Для получения информации о других параметрах и об управлении уровнями видимости параметров см. *Руководство по работе с оборудованием для логического контроллера Modicon M171 Optimized*.

Сохранение последнего использованного порта связи

Описание

При подключении к целевым устройствам через последовательный порт (COM-порт) обычно используется один и тот же порт для всех устройств (на многих компьютерах имеется только один COM-порт). Вы можете сохранить последний использованный COM-порт и позволить модулю **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** использовать этот порт для переопределения настроек проекта: эта функция полезна, когда вы работаете над проектом совместно с другими разработчиками, которые могут использовать другой COM-порт для подключения к целевому устройству.

Чтобы сохранить настройки COM-порта, включите опцию **Use last port (Использовать последний порт)** в меню **File (Файл)** → **Options (Параметры)…** → вкладка **General (Общее)**.

Раздел 12.3
Онлайн-статус

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Статус соединения	287
Статус проекта	288

Статус соединения

Описание

Состояние соединения отображается в поле рядом с правой границей строки **состояния.**

Если вы еще не пытались подключиться к целевому устройству, то отображается статус **Not connected (Не подключено)**.

NOT CONNECTED

При подключении к целевому устройству статус соединения меняется на один из следующих:

- ERROR

  `Error (Ошибка)`: невозможно установить связь. Необходимо проверить физическое соединение и настройки связи *(см. стр. 279)*.

- CONNECTED

  `Connected (Подключено)`: связь установлена.

Статус проекта

Описание

Рядом со статусом связи расположено еще одно поле, в котором отображается состояние проекта, выполняемого в данный момент на целевом устройстве.

Если статус соединения `Connected (Подключено)`, то статус проекта принимает одно из следующих значений (в зависимости от исходного кода):

- **NO CODE**
  `No code (Без кода)`: на целевом устройстве не выполняется ни один проект.
- **DIFF. CODE**
  `Diff. code (Др. код)`: проект, запущенный в данный момент на целевом устройстве, отличается от проекта, открытого в настоящее время в программном обеспечении. Кроме того, для запущенного проекта недоступна отладочная информация: режим отладки недоступен и нельзя полагаться на показания окна контрольных значений или осциллографа.
- **DIFF. CODE (SYM)**
  `Diff. code, Symbols OK (Др. код, символы OK)`: проект, запущенный в данный момент на целевом устройстве, отличается от проекта, открытого в настоящее время в программном обеспечении. Однако некоторая отладочная информация для выполняющегося проекта доступна (например, проект ранее был загружен на целевое устройство с того же компьютера): можно доверять показаниям окна контрольных значений или осциллографа, но режим отладки все еще невозможно активировать.
- **SOURCE OK**
  `Source OK (Источник OK)`: проект, запущенный в данный момент на целевом устройстве, соответствует проекту, открытому в настоящее время в программном обеспечении, режим отладки может быть активирован.

Раздел 12.4
Загрузка приложения

Обзор

Описание

Скомпилированное приложение ПЛК необходимо загрузить на целевое устройство, чтобы процессор мог выполнить его. Ниже подробно описывается процедура загрузки кода приложения на целевое устройство.

Шаг	Действие
1	Подключите целевое устройство к персональному компьютеру, на котором запущена программа EcoStruxure Machine Expert - HVAC.
2	Выберите пункт **On-line (Онлайн) → Download code (Загрузить код)**. **Результат:** Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** проверит, имеются ли для данного проекта несохраненные изменения. Если это так, то автоматически запускается компиляция приложения. Бинарный код отправляется на целевое устройство. По завершении загрузки контроллер автоматически выполняет сброс. Затем загруженный код выполняется процессором на целевом устройстве.

⚠ ПРЕДУПРЕЖДЕНИЕ

АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК КОНТРОЛЛЕРА

- Прежде чем загружать приложение, сначала проверьте состояние машины или процесса.
- Перед загрузкой приложения убедитесь в отсутствии опасностей, способных привести к травмированию людей, находящихся внутри или рядом с машиной или в непосредственной близости от технологического процесса.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Раздел 12.5
Моделирование

Обзор

Описание

В зависимости от подключенного целевого устройства можно смоделировать выполнение приложения ПЛК с помощью встроенной среды моделирования в модуле **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**.

Чтобы запустить процесс моделирования, выберите пункт **Debug (Отладка) → Simulation (Режим моделирования)**.

Дополнительную информацию см. в *Руководстве по моделированию (Simulation manual)*.

Раздел 12.6
Управление выполнением приложения ПЛК

Управление выполнением приложения ПЛК

Обзор

Выполнение приложения ПЛК можно контролировать с помощью соответствующих функций.

Эти функции доступны на панели инструментов Project (Проект) *(см. стр. 156)* и в меню On-line (Онлайн) *(см. стр. 151)*.

Контроллер начнет выполнение программной логики при подаче питания на оборудование. Важно заранее знать, как выходные данные будут воздействовать на управляемый процесс или машину.

При запуске контроллер будет пытаться начать выполнение программной логики сразу после подачи питания на оборудование, независимо от причины, по которой ранее была прекращена работа. Важно заранее знать, как запуск в нетипичных условиях повлияет на управляемый процесс или машину.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕПРЕДНАМЕРЕННЫЙ ЗАПУСК МАШИНЫ

- Проведите тщательный анализ рисков, чтобы определить последствия запуска приложения при любых, в том числе нетипичных, условиях.
- Перед подачей питания на контроллер или перед запуском приложения с помощью программы EcoStruxure Machine Expert - HVAC убедитесь в безопасности вашего контроллера или технологической среды.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Никогда не исходите из того, что вам заранее известно состояние контроллера, выполняя такие операции, как переключение состояния, настройка параметров контроллера, загрузка программы или изменение физической конфигурации контроллера и подключенного к нему оборудования.
- Перед выполнением любой из этих операций необходимо оценить влияние на все подключенное оборудование.
- Прежде чем взаимодействовать с контроллером, всегда проверяйте его состояние, в том числе наличие принудительно установленных выходов и сведения о статусе контроллера, отображаемые в EcoStruxure Machine Expert - HVAC.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Остановка

Вы можете остановить работу ПЛК, выбрав пункт **On-line (Онлайн) → Halt (Остановка)**.

Холодный перезапуск

Приложение ПЛК будет перезапущено, а сохраняемые и несохраняемые переменные будут сброшены.

Вы можете выполнить холодный перезапуск ПЛК, выбрав пункт **On-line (Онлайн) → Cold restart (Холодный перезапуск)**.

Теплый перезапуск

Приложение ПЛК будет перезапущено, при этом будут сброшены только несохраняемые переменные.

Вы можете выполнить теплый перезапуск ПЛК, выбрав пункт **On-line (Онлайн) → Warm restart (Теплый перезапуск)**.

Горячий перезапуск

Приложение ПЛК будет перезапущено, при этом переменные не будут сброшены.

Вы можете выполнить горячий перезапуск ПЛК, выбрав пункт **On-line (Онлайн) → Hot restart (Горячий перезапуск)**.

Перезапуск целевого устройства

Вы можете перезапустить целевое устройство, выбрав пункт **On-line (Онлайн) → Reboot target (Перезапуск целевого устройства)**.

Глава 13
Отладка

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
13.1	Обзор	294
13.2	Окно контрольных значений	295
13.3	Осциллограф	303
13.4	Режим редактирования и отладки	321
13.5	Отладка в режиме реального времени	322
13.6	Триггеры	326
13.7	Графические триггеры	347

Раздел 13.1
Обзор

Обзор

Описание

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** предоставляет несколько инструментов отладки, которые помогут вам проверить, ведет ли приложение себя так, как предполагалось.

Все эти инструменты отладки, по сути, позволяют отслеживать значения выбранных переменных во время работы приложения ПЛК.

Инструменты отладки модуля **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** можно объединить в две группы:

- Асинхронные отладчики. Они считывают значения выбранных переменных, отправляя последовательные запросы на целевое устройство. Менеджер инструмента отладки (запущен на персональном компьютере) и задача, посылающая ответы на эти запросы (на целевом устройстве), работают независимо от приложения ПЛК. Значения двух различных переменных, взятые в один и тот же момент, не обязательно будут корректно соотноситься друг с другом, как это задумано кодом приложения ПЛК, поскольку взятые значения могут относиться к разным тактам выполнения. По этой же причине динамика изменения значения одной переменной не согласуется с ее фактическим значением в памяти контроллера, особенно когда она быстро обновляется внутри приложения.
- Синхронные отладчики. Для их использования необходимо определить триггер в коде приложения ПЛК. Они одновременно обновляют все указанные переменные при каждом считывании триггера процессором, приостанавливая выполнение дальнейших инструкций до обновления значений всех переменных. Благодаря этому синхронные отладчики избавлены от ограничений, присущих асинхронным отладчикам.

В этой главе рассматриваются вопросы отладки приложения с использованием как асинхронных, так и синхронных инструментов.

Раздел 13.2
Окно контрольных значений

Содержание этого раздела

В этом разделе описаны следующие темы.

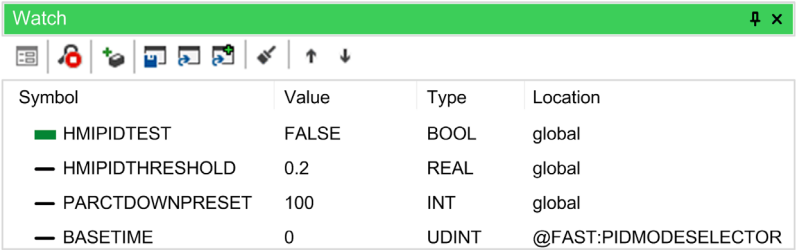
Тема	Страница
Обзор	295
Открытие и закрытие окна контрольных значений	296
Добавление элементов в окно контрольных значений	296
Удаление переменной	299
Обновление значений	299
Изменение формата данных	300
Работа со списками контрольных значений	301
Автосохранение списка контрольных значений	302

Обзор

Описание

Окно **Watch (Контрольные значения)** позволяет отслеживать значения набора переменных. Будучи асинхронным инструментом, окно **Watch (Контрольные значения)** не обеспечивает синхронизацию значений. Значения переменных в окне **Watch (Контрольные значения)** могут относиться к различным циклам выполнения соответствующей задачи.

Окно **Watch (Контрольные значения)** содержит элемент для каждой добавленной переменной. Информация в окне **Watch (Контрольные значения)** включает в себя имя переменной, ее значение, тип и расположение в приложении ПЛК.



Открытие и закрытие окна контрольных значений

Описание

Откройте/закройте окно **Watch (Контрольные значения)**, выполнив одну из следующих операций:

- Щелкните в меню **View (Вид)** → **Tool windows (Окна инструментов)** → **Watch (Контрольные значения)**.
- Нажмите на главной панели инструментов кнопку .
- Нажмите сочетание **Ctrl+T**.

Закрытие окна **Watch (Контрольные значения)** подразумевает его скрытие, а не сброс. Если вы закроете окно **Watch (Контрольные значения)**, а затем откроете его снова, то увидите, что оно все еще содержит все добавленные вами переменные.

Добавление элементов в окно контрольных значений

Описание

Чтобы отобразить переменную в окне **Watch (Контрольные значения)**, необходимо добавить ее в список контрольных значений.

ПРИМЕЧАНИЕ. В окно **Watch (Контрольные значения)** можно добавить любые переменные проекта, независимо от того, где они были объявлены.

Добавление переменной из текстового редактора исходного кода

Чтобы добавить переменную в окно **Watch (Контрольные значения)** из текстового редактора исходного кода (IL или ST), выполните следующие действия.

Шаг	Действие
1	Дважды щелкните по переменной, которую вы хотите отобразить в окне **Watch (Контрольные значения)**. 0001 x := x + hmiFrequency; 0002 0003 hmiSinVal := SIN(x) * hmiAmplitude; 0004 hmiCosVal := COS(x) * hmiAmplitude; 0005 0006 hmiStep := hmiStep + 1; 0007 0008
2	Перетащите выбранную переменную в окно **Watch (Контрольные значения)**: Watch Symbol | Value | Type | Location HMISINVAL | 0.982414 | REAL | @FAST:LOOPS

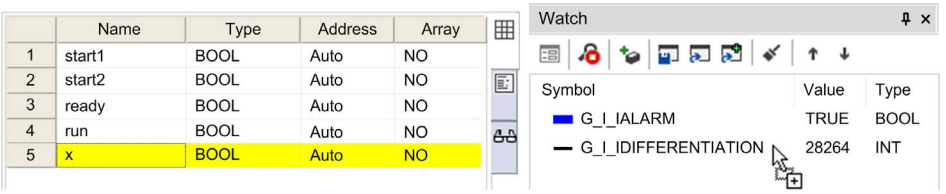
Добавление переменной из графического редактора исходного кода

Чтобы добавить переменную в окно **Watch (Контрольные значения)** из графического редактора исходного кода (LD, FBD и SFC), выполните следующие действия.

Шаг	Действие
1	Выберите пункт **Edit (Редактировать) → Watch mode (Режим контроля)**.
2	Щелкните по переменной, которую вы хотите отобразить в окне **Watch (Контрольные значения)**. Local variables Name / Type / Address / Array 1 start1 BOOL AUTO NO 2 start2 BOOL AUTO NO 3 ready BOOL AUTO NO 4 run BOOL AUTO NO 5 x BOOL AUTO NO 0001 start1 start2 run ready OR AND run
3	Появится диалоговое окно со списком всех существующих на данный момент экземпляров отладочных окон, при этом система спросит, в какое из них должен быть отправлен объект, по которому вы только что щелкнули. Debug windows list Symbol to add: start1 Debug windows Watch Oscilloscope OK Cancel Чтобы добавить переменную в окно **Watch (Контрольные значения)**, выберите **Watch (Контрольные значения)** и щелкните **OK**.
4	После добавления в окно **Watch (Контрольные значения)** всех переменных, которые необходимо отобразить, выберите **Edit (Редактировать) → Insert/Move mode (Режим вставки/перемещения)**, после чего курсор примет свою изначальную форму.

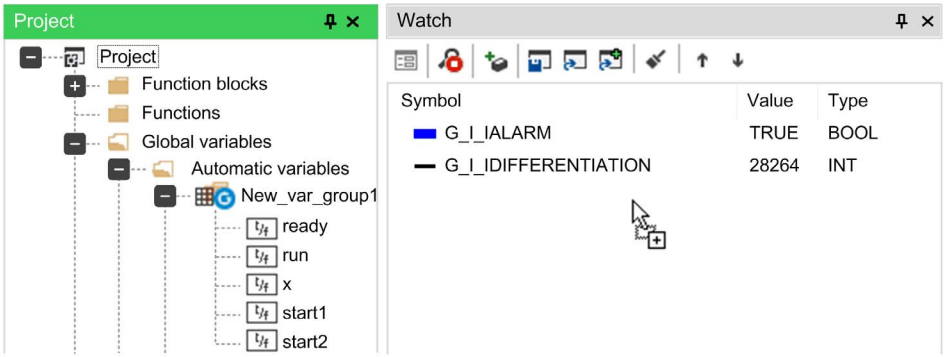
Добавление переменной из редактора переменных

Чтобы добавить переменную в окно **Watch (Контрольные значения)**, выберите соответствующую запись в редакторе переменных и перетащите ее в окно **Watch (Контрольные значения)**:



Добавление переменной из дерева проекта

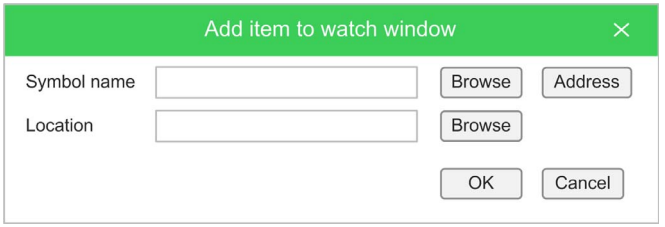
Чтобы добавить переменную в окно **Watch (Контрольные значения)**, выберите ее в дереве проекта и перетащите в окно **Watch (Контрольные значения)**:



Добавление переменной с панели инструментов окна контрольных значений

Чтобы добавить переменную в окно **Watch (Контрольные значения)**, нажмите кнопку **Insert new item (Вставить новый элемент)** на панели инструментов окна **Watch (Контрольные значения)**.

Необходимо ввести (или выбрать в обзоре символов проекта) имя переменной и ее местоположение (где она была объявлена).



Удаление переменной

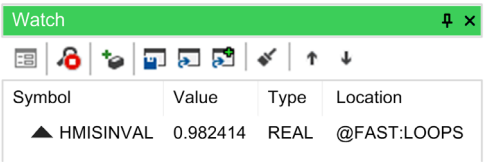
Описание

Если вы хотите удалить переменную из окна **Watch (Контрольные значения)**, выделите ее, щелкнув один раз по ее имени, затем нажмите клавишу **Delete**.

Обновление значений

Работа в обычном режиме

Менеджер окна контрольных значений периодически считывает из памяти значение переменных.



Однако это действие выполняется асинхронно. Может случиться так, что более приоритетная задача изменит значение некоторых переменных во время их считывания. В результате по окончании процесса обновления отображаемые в окне значения могут относиться к различным этапам выполнения кода приложения ПЛК.

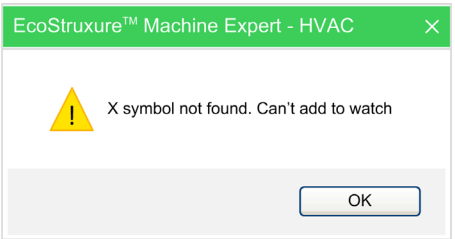
Целевое устройство отключено

Если целевое устройство отключено, в столбце **Value (Значение)** выводятся три точки.

Объект не найден

Если код приложения ПЛК изменяется, а модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** не может получить доступ к области памяти объекта в окне **Watch (Контрольные значения)**, то в столбце **Value (Значение)** выводятся три точки.

Если вы попытаетесь добавить в окно **Watch (Контрольные значения)** символ, для которого не была выделена память, модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** выдаст следующее сообщение об ошибке:



Изменение формата данных

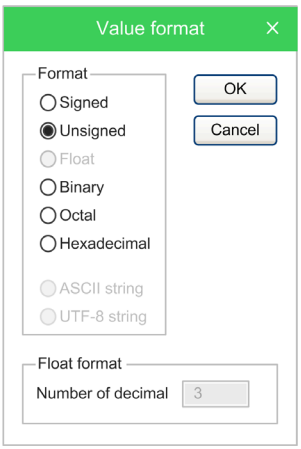
Описание

При добавлении переменной в окно **Watch (Контрольные значения)** модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** автоматически распознает ее тип (целое число без знака, целое число со знаком, число с плавающей точкой, шестнадцатеричное число) и последовательно отображает ее значение. Если переменная является числом с плавающей точкой, модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** присваивает ей количество десятичных знаков по умолчанию.

Однако вам может потребоваться, чтобы переменная отображалась в другом формате.

Чтобы применить формат, отличный от назначенного модулем **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**, выберите переменную и нажмите кнопку **Value format (Формат значения)** на панели инструментов.

Выберите формат и подтвердите свой выбор.



Работа со списками контрольных значений

Описание

Вы можете сохранить в файле набор всех элементов, содержащихся в окне **Watch (Контрольные значения)**, чтобы легко восстановить состояние этого отладочного инструмента в следующей рабочей сессии.

Для сохранения содержимого списка контрольных значений выполните следующие действия.

Шаг	Действие
1	Нажмите кнопку **Save watch list (Сохранить список контрольных значений)** на панели инструментов окна **Watch (Контрольные значения)**.
2	Введите имя файла и выберите папку для сохранения.

Вы можете загрузить список контрольных значений из файла, удалив открытый и выполнив следующие действия.

Шаг	Действие
1	Нажмите кнопку **Load (no appends) watch list (Загрузить список контрольных значений [без добавления])** на панели инструментов окна **Watch (Контрольные значения)**.
2	Выберите в окне проводника файл списка контрольных значений. Набор символов из списка контрольных значений добавляется в окно **Watch (Контрольные значения)**.

Вы можете загрузить список контрольных значений из файла, добавив его к уже открытому и выполнив следующие действия.

Шаг	Действие
1	Нажмите кнопку **Load watch list (Загрузить список контрольных значений)** на панели инструментов окна **Watch (Контрольные значения)**.
2	Выберите в окне проводника файл списка контрольных значений. Набор символов из списка контрольных значений добавляется в окно **Watch (Контрольные значения)**.

Вы можете очистить открытый в данный момент список контрольных значений, нажав кнопку **Remove all items from the watch list (Удалить все элементы из списка контрольных значений)**.

Вы можете изменить местоположение выбранного элемента в открытом списке контрольных значений, нажав кнопку **Move up item into the list (Переместить элемент списка вверх)** или **Move down item in the list (Переместить элемент списка вниз)**.

Автосохранение списка контрольных значений

Описание

Если выбрать соответствующую опцию в диалоговом окне параметров проекта (подробнее см. в разделе «Отладка» *(на стр. 170)*, список контрольных значений будет автоматически сохраняться при закрытии проекта.

Сохраненный список контрольных значений автоматически загружается во время первого подключения к целевому устройству при повторном открытии проекта.

Раздел 13.3
Осциллограф

Содержание этого раздела

В этом разделе описаны следующие темы.

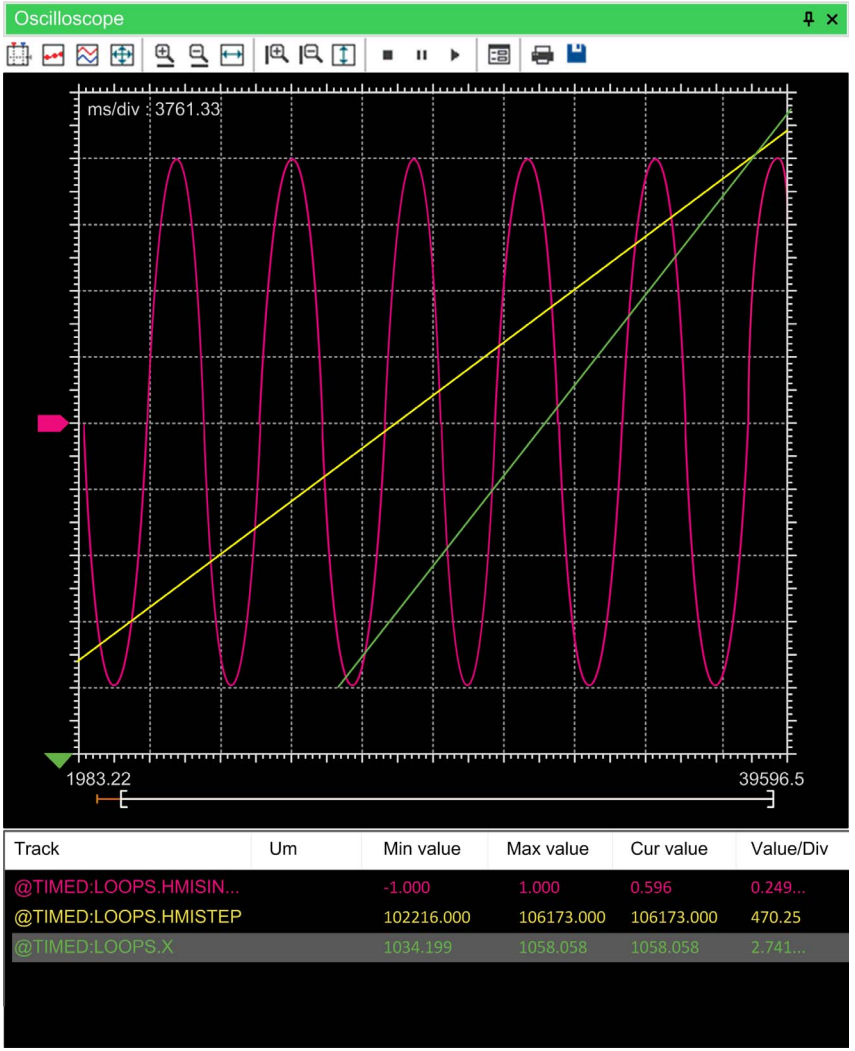
Тема	Страница
Обзор	304
Открытие и закрытие осциллографа	305
Добавление элементов в осциллограф	305
Удаление переменной	309
Выборка переменных	309
Управление сбором и отображением данных	310
Изменение частоты опроса	318
Сохранение и печать графика	318

Обзор

Описание

Осциллограф позволяет построить график изменения значений набора переменных. Будучи асинхронным инструментом, осциллограф не обеспечивает синхронизацию выборок.

Окно **Oscilloscope (Осциллограф)** — это интерфейс для доступа к функциям отладки, предоставляемым осциллографом:



Окно **Oscilloscope (Осциллограф)** состоит из трех областей:

- Панель инструментов позволяет управлять осциллографом. Подробное описание функции каждого элемента управления приводится ниже в этой главе.
- Область графика содержит несколько элементов:
  - График: в этой области отображается кривая значений переменных.
  - Вертикальные маркеры: эти маркеры позволяют настроить две вертикальные линии. Значения каждой переменной на пересечении с этими линиями отображаются в соответствующих столбцах.
  - Полоса прокрутки: если масштаб оси X слишком велик для отображения всех выборок в области графика, то полоса прокрутки позволяет перемещаться вперед и назад по горизонтальной оси.
- Нижняя область окна осциллографа представляет собой таблицу, каждая строка которой соответствует отдельной переменной.

Открытие и закрытие осциллографа

Описание

Чтобы открыть или закрыть осциллограф, выберите пункт **View (Вид) → Tool windows (Окна инструментов) → Oscilloscope (Осциллограф).**

Закрытие окна осциллографа подразумевает его скрытие, а не сброс. Если после закрытия окна осциллографа вы снова его откроете, то увидите, что построение кривой для всех добавленных переменных начнется заново.

Добавление элементов в осциллограф

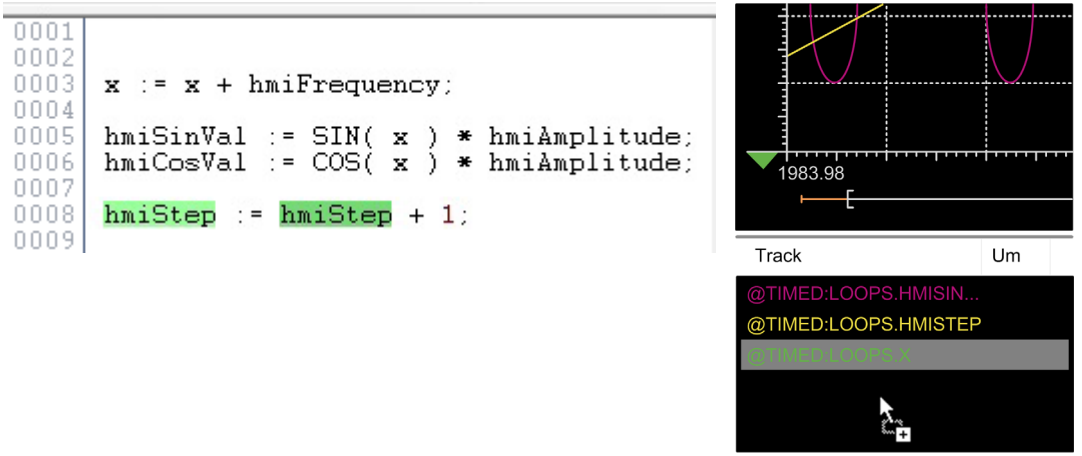
Описание

Чтобы построить график изменения значения переменной, необходимо добавить ее в осциллограф.

В отличие от окон триггеров и окна **Graphic trigger (Графический триггер)**, вы можете добавить в осциллограф любые переменные проекта независимо от того, где они были объявлены.

Добавление переменной из текстового редактора исходного кода

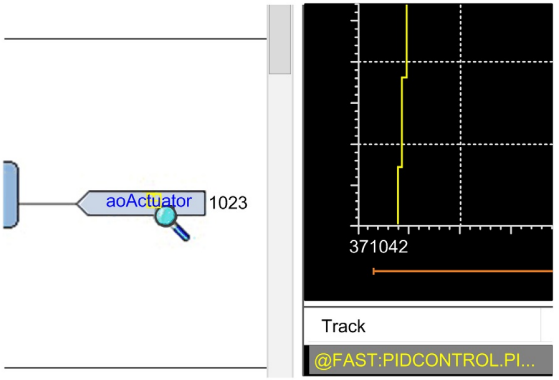
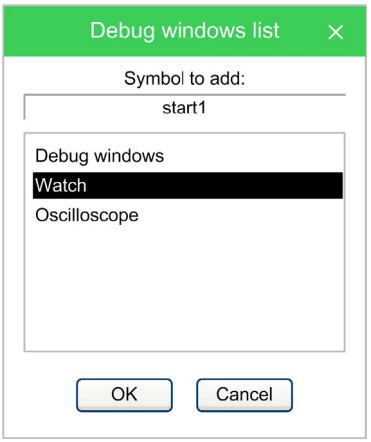
Чтобы добавить переменную в осциллограф из текстового (то есть IL или ST) редактора исходного кода, выделите переменную, дважды щелкнув по ней, а затем перетащите ее в окно **Oscilloscope (Осциллограф)**.



Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

Добавление переменной из графического редактора исходного кода

Чтобы добавить переменную в осциллограф из графического редактора исходного кода (LD, FBD и SFC), выполните следующие действия.

Шаг	Действие
1	Выберите пункт **Edit (Редактировать) → Watch mode (Режим контроля)**.
2	Щелкните блок с переменной, которую вы хотите отслеживать на осциллографе: 
3	Появится диалоговое окно со списком всех существующих на данный момент экземпляров отладочных окон, при этом система спросит, в какое из них должен быть отправлен объект, по которому вы только что щелкнули:  Выберите **Oscilloscope (Осциллограф)**, затем нажмите **OK**. Имя переменной теперь отображается в столбце **Track (Кривая)**.

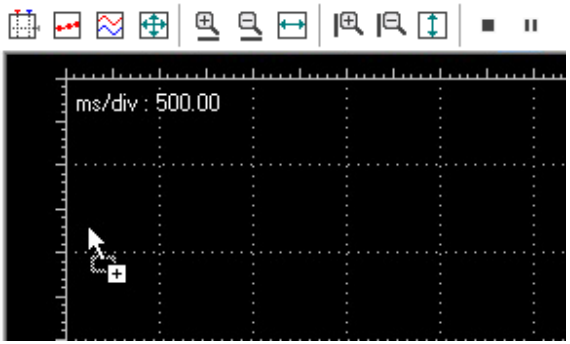
Та же процедура применяется ко всем переменным, которые необходимо отслеживать. После добавления в окно осциллографа всех переменных, которые необходимо отслеживать, выберите пункт **Edit (Редактировать) → Insert/Move mode (Режим вставки/перемещения)**, после чего курсор примет свою изначальную форму.

Добавление переменной из редактора переменных

Чтобы добавить переменную в осциллограф, можно выбрать соответствующую запись в редакторе переменных, а затем перетащить ее в окно осциллографа:

Local variables

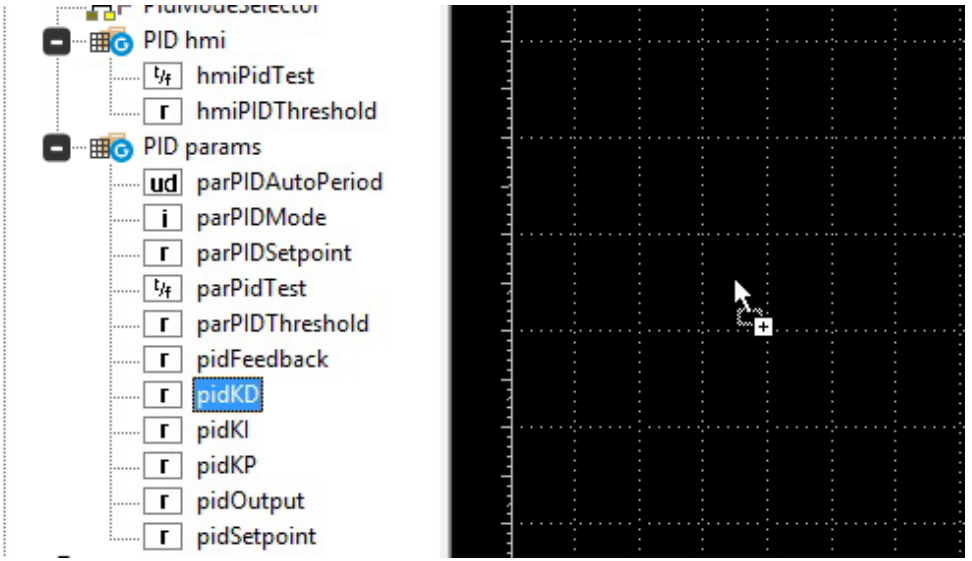
	Class	Pin	Name	Type	Array	Init value	Attribute
1	VAR		absSpeed	REAL	NO		..
2	VAR		T	REAL	NO		..
3	VAR		remSpace	DINT	NO		..
4	VAR		T2	REAL	NO		..
5	VAR		sign	REAL	NO		..
6	VAR		prevSpeed	REAL	NO		..



Также можно нажать клавишу **F10** и выбрать **Oscilloscope (Осциллограф)** из появившегося списка окон отладки.

Добавление переменной из дерева проекта

Чтобы добавить переменную в осциллограф, можно выбрать ее в дереве проекта, а затем перетащить в окно осциллографа:



Также можно нажать клавишу **F10** и выбрать **Oscilloscope (Осциллограф)** из появившегося списка окон отладки.

Удаление переменной

Описание

Если вы хотите удалить переменную из окна осциллографа, выделите ее, щелкнув один раз по ее имени, затем нажмите клавишу **Delete**.

Выборка переменных

Работа в обычном режиме

Менеджер окна осциллографа периодически считывает из памяти значение этих переменных.

Однако это действие выполняется асинхронно. Может случиться так, что более приоритетная задача изменит значение некоторых переменных во время их считывания. По окончании процесса выборки данные, ассоциированные с одним и тем же значением по оси X, могут относиться к различным этапам выполнения кода приложения ПЛК.

Целевое устройство отключено

Если целевое устройство отключается, кривые перетаскиваемых переменных застывают в одном положении до тех пор, пока связь не будет восстановлена.

Управление сбором и отображением данных

Описание

Окно осциллографа содержит панель инструментов с несколькими командами для управления процессом сбора данных и способом их отображения. Данный раздел посвящен этим командам. Команды на панели инструментов неактивны, если в осциллографе нет отслеживаемых переменных.

Запуск и остановка сбора данных

При добавлении переменной в осциллограф немедленно запускается сбор данных.

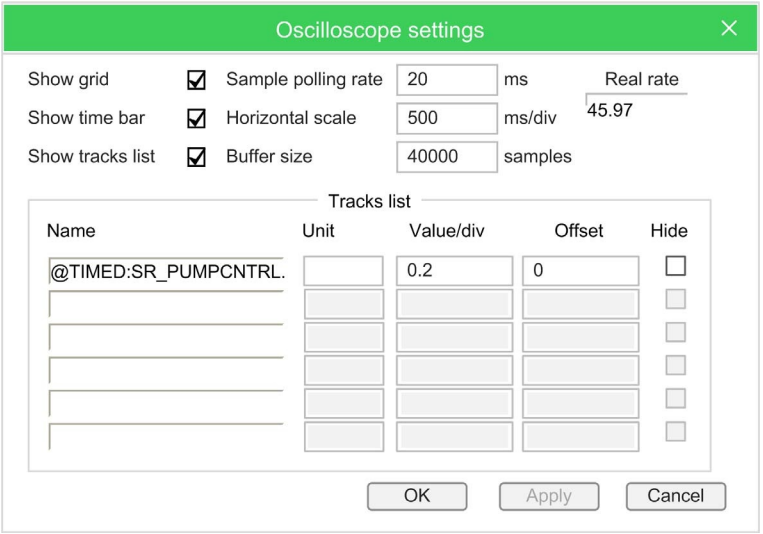
Однако вы можете приостановить сбор данных, нажав кнопку **Pause acquisition (Приостановить сбор данных)**.
Кривая застывает (при этом процесс сбора данных еще выполняется в фоновом режиме) до тех пор, пока вы не нажмете **Restart acquisition (Перезапустить сбор данных).**

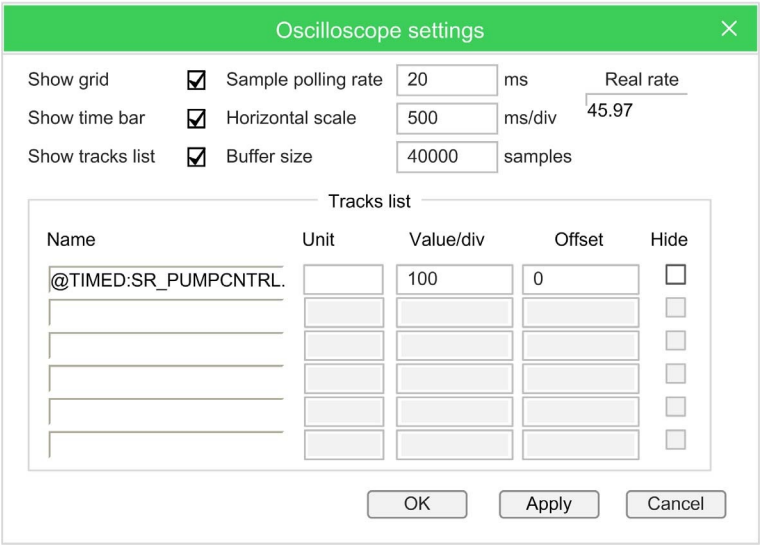
Чтобы остановить сбор данных, вы можете нажать кнопку **Stop acquisition (Остановить сбор данных).**
В этом случае при нажатии кнопки **Restart acquisition (Перезапустить сбор данных)** график сбрасывается и перезапускается с фактическим значением переменной.

Настройка масштаба осей

При открытии окна осциллографа модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** применяет к осям масштаб по умолчанию.

Чтобы изменить значение масштабирования, выполните следующие действия.

Шаг	Действие
1	Откройте **Oscilloscope settings (Настройки осциллографа)**, нажав кнопку **Graph properties (Свойства графика)** на панели инструментов.
2	Задайте масштаб для горизонтальной оси, который будет общим для всех кривых: 

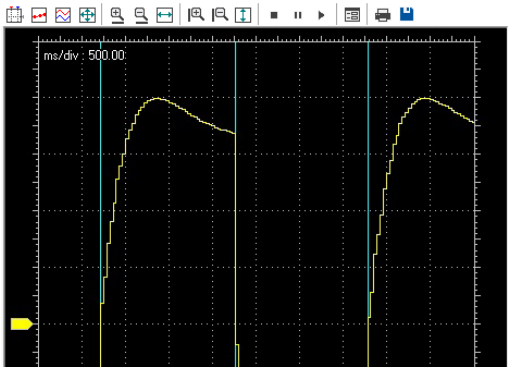
Шаг	Действие
3	Для каждой переменной можно задать свой масштаб по вертикальной оси: 
4	Подтвердите настройки. График адаптируется в соответствии с новым масштабом.

Также можно выполнять масштабирование как по горизонтальной, так и по вертикальной оси:

Наконец, вы также можете адаптировать масштаб горизонтальной оси, вертикальной оси или обеих таким образом, чтобы те охватывали все выборки. Для этого нажмите на соответствующий пункт панели инструментов:

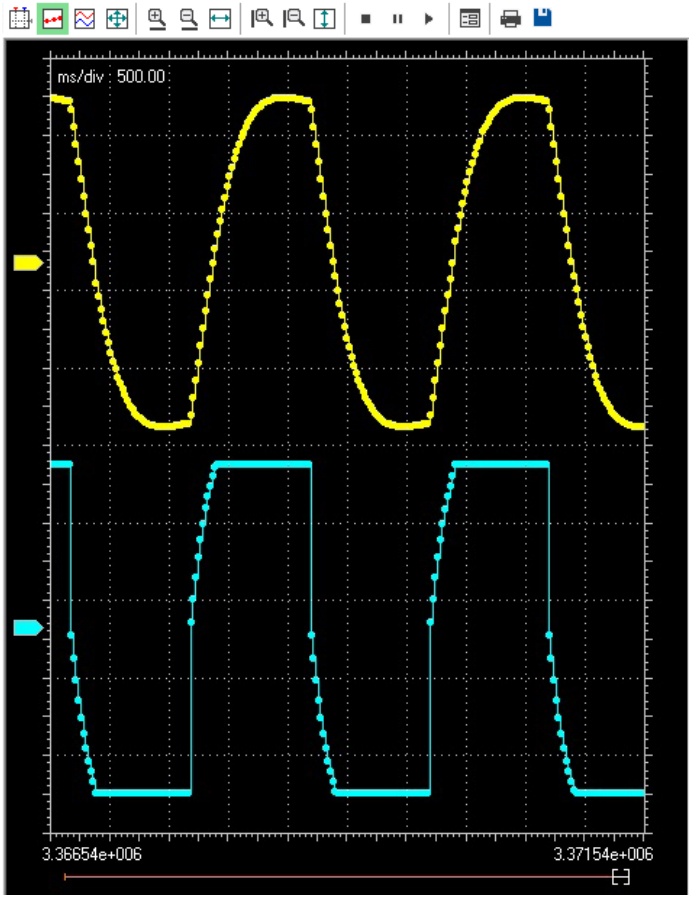
Вертикальное разделение

Если вы отслеживаете изменение двух или более переменных, можно при желании отобразить соответствующие кривые по отдельности. Для этого нажмите значок **Vertical split (Вертикальное разделение)** на панели инструментов окна **Oscilloscope (Осциллограф)**:



Просмотр выборок

Если нажать кнопку **Show samples (Показать выборки)** на панели инструментов **Oscilloscope (Осциллограф)**, инструмент выделит отдельные значения, обнаруженные во время сбора данных:

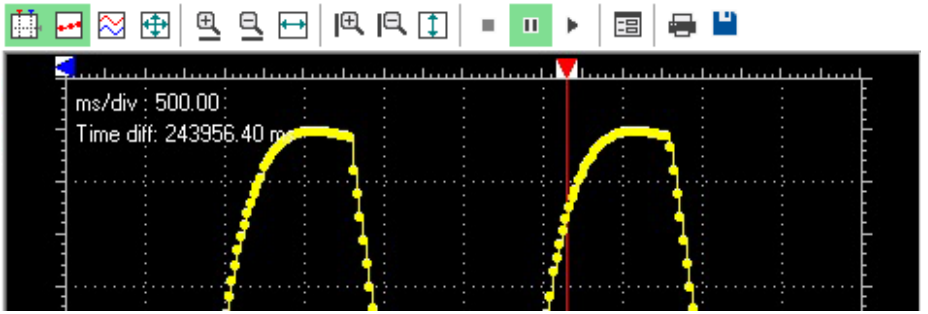


Вы можете щелкнуть по тому же пункту еще раз, чтобы вернуться в режим просмотра по умолчанию.

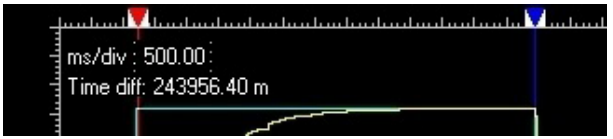
Выполнение измерений

Окно осциллографа содержит две измерительные направляющие, с помощью которых можно выполнять определенные измерения на графике. Чтобы показать и скрыть их, нажмите значок **Show measure bars (Показать измерительные направляющие)** на панели инструментов **Oscilloscope (Осциллограф)**.

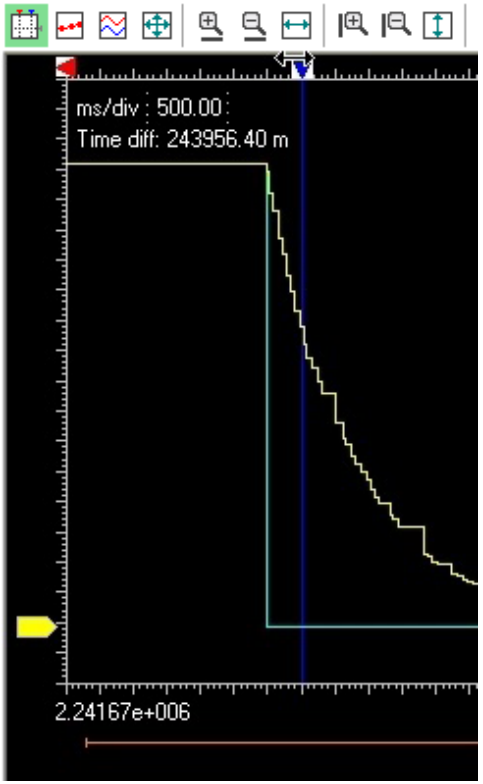
Чтобы измерить временной интервал между двумя событиями, необходимо переместить одну направляющую в точку на графике, соответствующую первому событию, а другую — в точку, соответствующую второму:



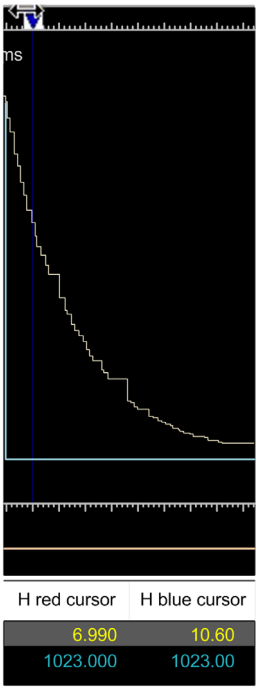
Временной интервал между двумя измерительными направляющими отображается в левом верхнем углу графика:



С помощью измерительной направляющей можно также считывать значение всех переменных осциллографа в определенный момент времени. Переместите направляющую в точку на графике, которая соответствует нужному моменту:



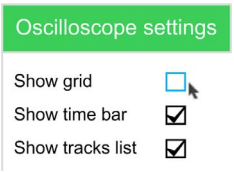
В таблице ниже (под графиком) теперь выводятся значения всех переменных в данный конкретный момент времени:



Настройки осциллографа

Внешний вид осциллографа можно настроить, нажав кнопку **Graph properties (Свойства графика)** на панели инструментов.

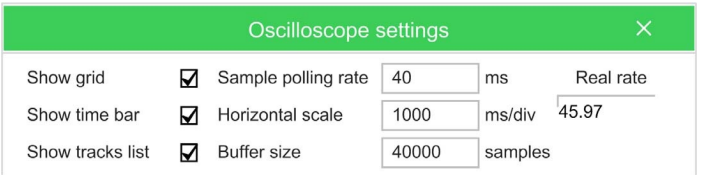
В открывшемся окне можно настроить отображение фоновой сетки (**Background grid**), ползунка времени (**Time slide bar**) и списка кривых (**Track list**):



Изменение частоты опроса

Описание

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** периодически посылает запросы на целевое устройство для считывания данных, которые должны выводиться на осциллограф. Чтобы настроить частоту опроса, выполните следующие действия.

Шаг	Действие
1	Нажмите кнопку **Graph properties (Свойства графика)** на панели инструментов.
2	В открывшемся окне отредактируйте значение **Sampling polling rate (Частота выборки)**: 
3	Подтвердите действие.

Частота зависит от производительности целевого устройства (в частности, от эффективности выполнения задачи по передаче данных). Значение частоты выводится в окне **Oscilloscope settings (Настройки осциллографа)**.

Сохранение и печать графика

Описание

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** позволяет сохранить данные в файл либо распечатать их в том виде, в каком они представлены на осциллографе.

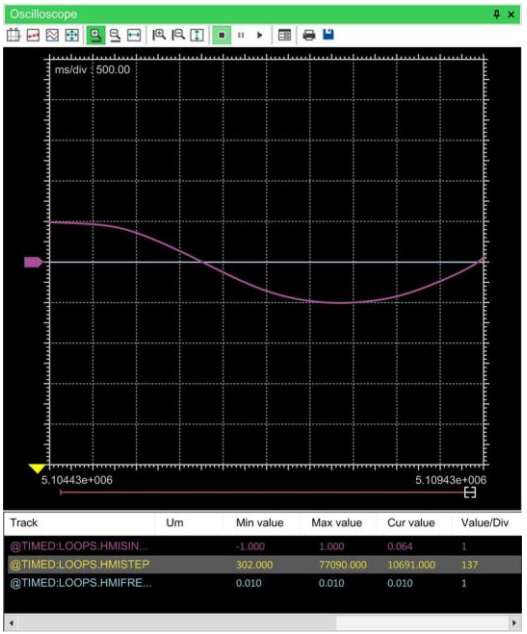
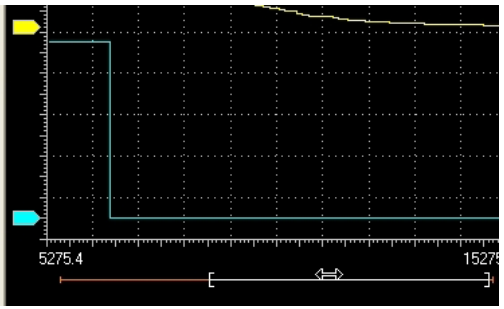
Сохранение данных в файл

Вы можете сохранить в файл выборки, выведенные на осциллограф, для последующего анализа с помощью других инструментов.

Шаг	Действие
1	Остановите сбор данных кнопкой ■ перед сохранением в файл.
2	Нажмите кнопку **Save tracks data into file (Сохранить данные кривых в файл)** на панели инструментов **Oscilloscope (Осциллограф)**.
3	Выберите один из доступных форматов вывода: ● OSC — это простой текстовый файл, содержащий время и значение для каждой выборки. ● OSCX — это XML-файл, содержащий более полную информацию, которую можно дополнительно проанализировать с помощью другой вкладки.
4	Выберите имя файла и директорию для сохранения, затем подтвердите операцию.

Печать графика

Чтобы распечатать данные, представленные на осциллографе, выполните следующие действия.

Шаг	Действие
1	Нажмите , чтобы временно приостановить, или ■ , чтобы полностью прекратить сбор данных.
2	Переместите ползунок времени и измените масштаб, чтобы охватить все элементы, которые необходимо распечатать:  
3	Нажмите кнопку **Print graph (Распечатать график)**.

Раздел 13.4
Режим редактирования и отладки

Обзор

Описание

Все инструменты отладки, кроме окон **Watch (Контрольные значения)** и Oscilloscope (Осциллограф), блокируют исходный код: при включенном режиме отладки появляется доступ к соответствующим инструментам, но внести изменения в исходный код не получится.

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** автоматически включает режим отладки, если соблюдается хотя бы одно из следующих условий:

- Минимум одна точка останова установлена правильно.
- Минимум один триггер (графический или текстовый) установлен правильно.
- Режим отладки в режиме реального времени включен.

Если все эти условия не выполняются, режим отладки автоматически отключается, а модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** переходит в режим редактирования.

В строке состояния отображается, активен ли режим отладки:

DEBUG MODE	SOURCE OK	CONNECTED

Вы не можете войти в режим отладки, если состояние соединения отличается от **Connected (Подключено)**, а приложения в программном обеспечении и контроллере совпадают, о чем свидетельствует индикатор **SOURCE OK**.

Раздел 13.5
Отладка в режиме реального времени

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	322
Моделирование SFC	323
Моделирование LD	324
Моделирование FBD	324
Моделирование IL и ST	325

Обзор

Описание

Модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** может отображать информативную анимацию текущего и динамического состояния исполнения программных модулей, описанных на любом языке программирования стандарта МЭК 61131-3 (ГОСТ Р МЭК 61131-3).

Чтобы включить или отключить режим отладки в режиме реального времени, нажмите кнопку **Debug (Отладка) → Live debug mode (Отладка в режиме реального времени)**.

Контроллер позволяет принудительно изменять состояние выбранных выходов и значения переменных в целях тестирования системы, ввода в эксплуатацию и технического обслуживания. Вы можете принудительно устанавливать значение для выхода и (или) переменной, если контроллер соединен с EcoStruxure Machine Expert - HVAC.

⚠ ПРЕДУПРЕЖДЕНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

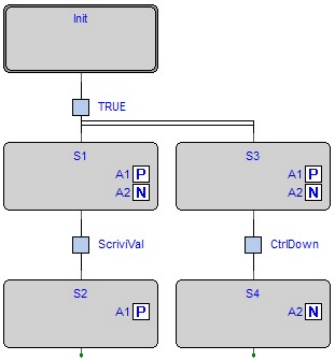
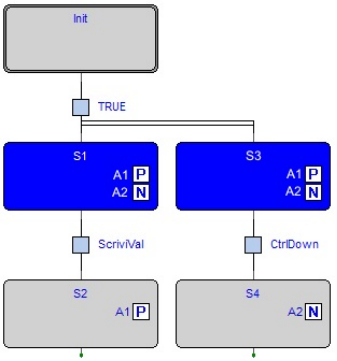
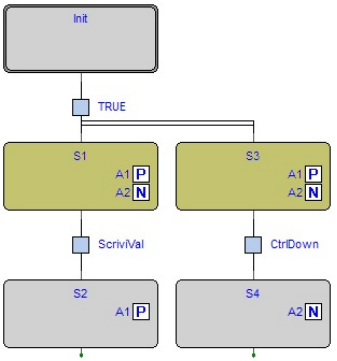
- Вы должны хорошо понимать, как подобная процедура будет влиять на выходы в зависимости от выполняемых задач.
- Не пытайтесь принудительно изменять значения для входов/выходов, задействованных в задачах, если вы не уверены, что те будут выполнены своевременно.
  Речь не идет о тех случаях, когда вы действительно хотите применить значения при следующем выполнении задачи.
- Даже если принудительное изменение значения выхода никак не повлияло на физический выход, не выходите из EcoStruxure Machine Expert - HVAC, не отключив принудительное изменение значений.

Несоблюдение этих указаний может привести к тяжелым или смертельным травмам либо к повреждению оборудования.

Моделирование SFC

Описание

Как описано в справке по языку, программный модуль SFC структурирован в виде набора шагов, каждый из которых может быть активным или неактивным в определенный момент времени. После запуска этот инструмент отладки для SFC анимирует документы SFC, подсвечивая активные шаги.

Анимация выключена	Анимация включена	Анимация включена, но в статусе удержания
		
Отображается часть сети SFC, анимация диаграммы отключена.	Та же часть сети отображается при активном режиме отладки в режиме реального времени. На рисунке показано, что шаги `S1` и `S3` в настоящее время активны, в то время как `Init, S2` и `S4` неактивны.	Та же часть сети отображается с шагами `S1` и `S3`, которые в данный момент активны, но находятся в состоянии удержания. Это может происходить в блоках SFC, когда они являются дочерними элементами блока в неактивном состоянии.

Менеджер анимации SFC периодически тестирует состояние всех шагов, при этом изменять период выборки запрещено. Шаг может активироваться на столь короткое время, что он не отображается на экране. Тот факт, что шаг никогда не подсвечивается, не означает, что его действие не выполняется. Возможно, исполнение шага не регистрируется из-за слишком малой частоты выборки.

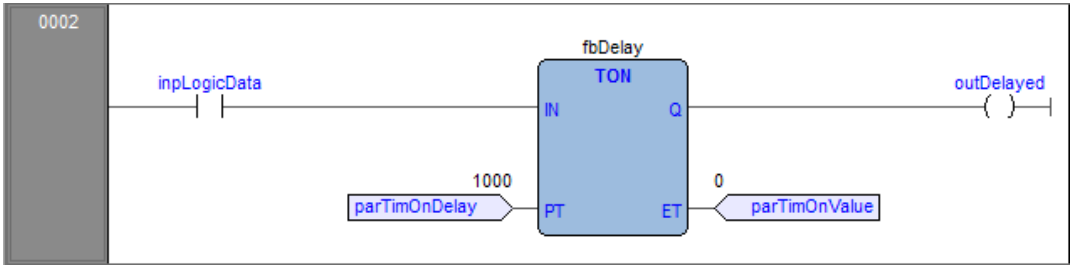
Действия и условия отладки

Как поясняется в справке по языку SFC, действие можно присвоить шагу, а переход можно связать с условием. Действия и условия могут быть описаны на любом из языков стандарта МЭК 61131-3 (ГОСТ Р МЭК 61131-3). Инструменты отладки общего назначения могут использоваться внутри каждого действия/условия, как если бы это был отдельный программный модуль.

Моделирование LD

Описание

В режиме отладки в режиме реального времени схемы релейной логики анимируются путем подсветки контактов и катушек, значение которых является истинным (в примере — i1 и i2):

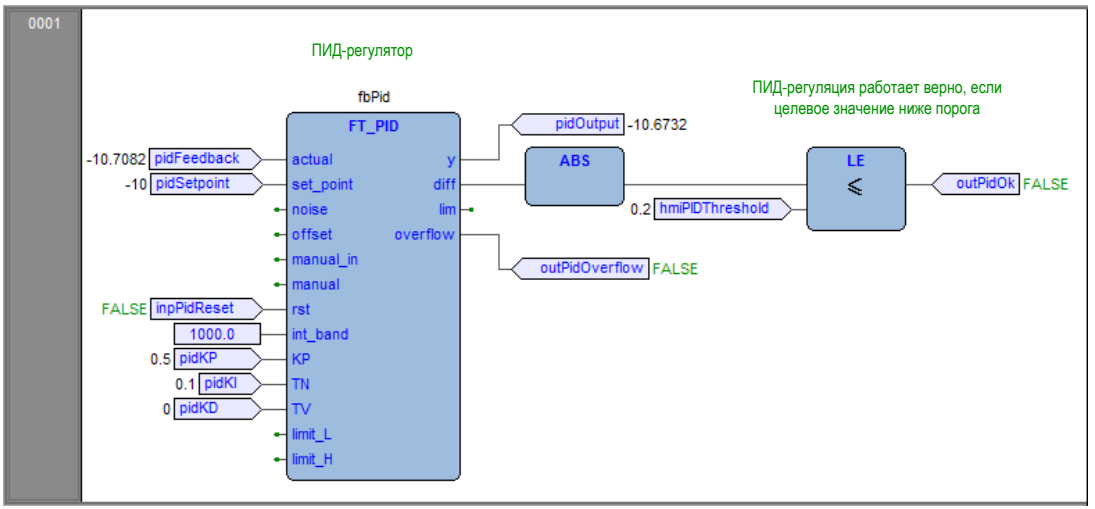


Менеджер анимации LD периодически проверяет состояние всех элементов. Может случиться так, что элемент остается истинным слишком короткое время и не успевает отобразиться. Тот факт, что элемент никогда не подсвечивается, не означает, что его значение никогда не становится истинным (частота выборки может быть слишком медленной).

Моделирование FBD

Описание

В режиме отладки в режиме реального времени модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** отображает значения всех видимых переменных непосредственно в графическом редакторе исходного кода:



Это касается как языка программирования FBD, так и языка LD.

Менеджер анимации FBD периодически проверяет состояние всех элементов. Может случиться так, что элемент остается истинным слишком короткое время и не успевает отобразиться. Тот факт, что элемент никогда не подсвечивается, не означает, что его значение никогда не становится истинным (частота выборки может быть слишком медленной).

Моделирование IL и ST

Описание

Режим отладки в режиме реального времени также распространяется на текстовые редакторы исходного кода (для IL и ST). Вы можете просматривать значения переменной, наведя на нее курсор мыши:

```
0001
0002
0003 (*  Analog output 0 = analog inp 0 + analog inp 1 *)
0004
0005 aout0 := ainp0 + ainp1;
0006
0007 (*  SFC state logic *)
0008
0009 fbStati( enab := inp10, run := inp11, stop := inp12 );
0010
0011 cnt := cnt + 1;
0012
0013      -29133
0014
```

Раздел 13.6
Триггеры

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Окно триггера	326
Отладка с использованием окон триггеров	332

Окно триггера

Описание

Окно **Trigger (Триггер)** — это инструмент, позволяющий выбрать набор переменных и синхронизировать их обновление в специальном всплывающем окне.

Предварительные условия для открытия окна триггера

Объем доступной памяти

Окно триггера занимает сегмент в секторе кода приложения, имеющий строго определенную длину. Очевидно, что для запуска окна триггера приложению необходимо достаточное количество памяти, иначе отобразится сообщение об ошибке.

Несовместимость с окнами графических триггеров

Окно графического триггера занимает все свободное пространство сектора кода приложения. После запуска такого инструмента отладки добавить окно триггера невозможно, а при попытке запустить новое окно появляется сообщение об ошибке. Если закрыть окно графического триггера, окна триггеров снова активируются.

Все окна триггеров, открытые до запуска окна графического триггера, продолжают работать в нормальном режиме. Добавлять новые нельзя.

Панель инструментов окна триггера

Значки окна триггера являются частью панели инструментов **Debug (Отладка)** и активируются только в том случае, если модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** находится в режиме отладки.

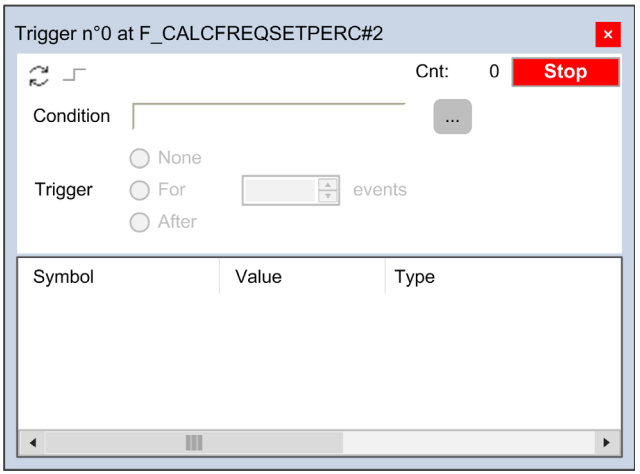
Значок	Команда	Описание
	Add/remove text trigger (Добавить/удалить текстовый триггер)	Для запуска окна триггера выберите точку в коде приложения ПЛК, куда следует вставить соответствующий триггер, а затем нажмите эту кнопку. Та же процедура применяется и для удаления окна триггера: чтобы окончательно закрыть окно отладки, нажмите один раз на инструкцию/блок, куда был вставлен триггер, затем снова нажмите эту кнопку. Горячие клавиши: **F9**
	Add/remove graphic trigger (Добавить/удалить графический триггер)	Эта кнопка работает точно так же, как и **Add/remove text trigger (Добавить/удалить текстовый триггер)**, за исключением того, что открывается окно графического триггера. Ее также можно использовать для закрытия окна графического триггера. Горячие клавиши: **Shift+F9**
	Remove all triggers (Удалить все триггеры)	Эта кнопка позволяет одновременно закрыть все имеющиеся окна триггеров и окно графического триггера. Горячие клавиши: **Ctrl+Shift+F9**
	Trigger list (Список триггеров)	Эта кнопка открывает диалоговое окно со списком всех имеющихся окон триггеров: Trigger list Type Module Line G System -1 T RMS -1 T Fast 14 T Init -1 T Slow -1 Open Remove Remove all OK Горячие клавиши: **Ctrl+I**

Каждая запись относится к окну графического или текстового триггера. В таблице ниже поясняется значение каждого поля.

Поле	Описание
Type (Тип)	**T** : окно триггера. **G**: окно графического триггера.
Module (Модуль)	Имя программы, функции или функционального блока, где установлен триггер. Если модуль является функциональным блоком, то в данном поле указывается его имя, а не имя его экземпляра, куда был установлен триггер.
Line (Строка)	В случае текстовых языков (IL и ST) указывает на строку, в которой находится триггер. Для других языков значение всегда равно **-1**.

Интерфейс окна триггера

Установка триггера приводит к появлению всплывающего окна **Interface (Интерфейс)**, представляющего собой интерфейс для доступа к функциям отладки окна триггера. Оно состоит из трех элементов, как показано ниже:



Строка заголовка

В строке **заголовка** всплывающего окна отображается информация о местоположении триггера, вызывающего обновление окна **Variables (Переменные)** в момент обработки процессором.

Текст в строке заголовка имеет следующий формат:

`Trigger n° X at НазваниеМодуля#Местоположение`

X	Идентификатор триггера.
НазваниеМодуля	Имя программы, функции или функционального блока, где установлен триггер.
Местоположение	Точное расположение триггера внутри модуля **НазваниеМодуля**. Если речь идет о **НазваниеМодуля** в коде на языке IL, то **Местоположение** имеет следующий формат: `N1` Если же **НазваниеМодуля** используется в коде на языке FBD, то используется следующий формат: `N2$BT:BID,` где: `N1` = номер строки с инструкцией; `N2` = номер сети; `BT` = тип блока (операнд, функция, функциональный блок и т. д.); `BID` = идентификатор блока.

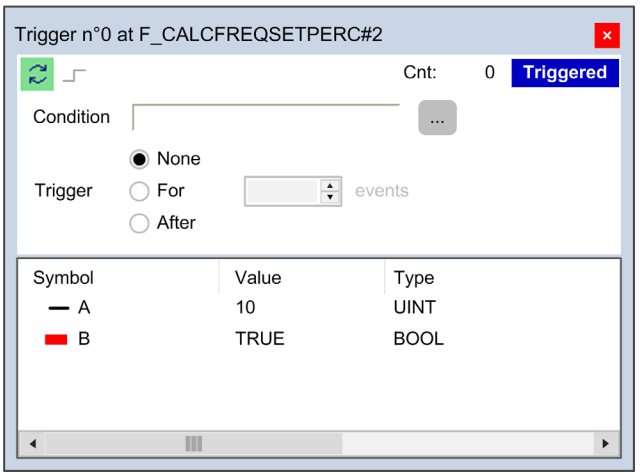
Область элементов управления

Это диалоговое окно позволяет управлять обновлением окна триггера для получения дополнительной информации о рассматриваемом коде. Подробное описание функции каждого элемента управления приводится в разделе с описанием элементов управления окна триггера. См. описание в разделе «Использование элементов управления» *(стр. 343)*.

Все элементы управления остаются недоступными до тех пор, пока вы не перетащите в окно отладки хотя бы одну переменную.

Область переменных

Эта нижняя область окна **Debug (Отладка)** представляет собой таблицу, каждая строка которой соответствует переменной, которую вы перетащили. Каждая строка имеет несколько полей: имя переменной, значение, тип, местоположение `(@task:НазваниеМодуля)`, а также описание, считанное из памяти при последнем обновлении:



Окно триггера: информация о перетаскивании

Чтобы посмотреть переменную, необходимо скопировать ее в нижнюю часть окна **Debug (Отладка)**.

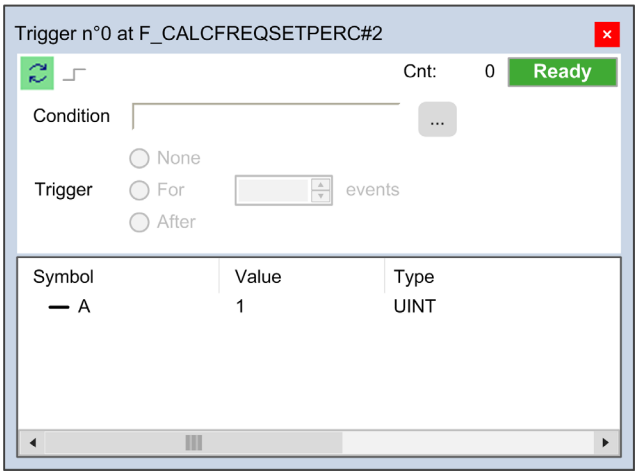
Эта область представляет собой таблицу, каждая строка которой соответствует переменной, которую вы перетащили. В окно триггера можно перетаскивать только переменные, являющиеся локальными для модуля, в котором установлен соответствующий триггер, или же глобальные переменные и параметры. Нельзя перетаскивать переменные, объявленные в другой программе, функции или функциональном блоке.

Обновление значений

Рассмотрим следующий пример:

```
0001    LD 1
0002    ST a
0003
0004    LD 2
0005    ST a
0006
0007    LD 3
0008    ST a
0009
```

Значение переменных обновляется каждый раз при вызове менеджера окна, т. е. каждый раз, когда процессор выполняет инструкцию, помеченную зеленой стрелкой. Однако вы можете настроить элементы управления таким образом, чтобы переменные обновлялись только тогда, когда триггеры удовлетворяют более строгим условиям, заданным лично вами.



Значение переменных в столбце **Symbol (Символ)** считывается из памяти непосредственно перед отмеченной инструкцией (в данном случае — инструкцией в пятой строке) и непосредственно после выполнения предыдущей инструкции (инструкции в четвертой строке).

Таким образом, в примере выше на момент считывания из памяти и отображения в окне триггера нового значения **a** вторая инструкция ST еще не была выполнена. Следовательно, результат выполнения второй инструкции ST **a** равен 1.

Элементы управления окна триггера

Элементы управления окна триггера позволяют контролировать работу этого инструмента отладки.

Элементы управления окна триггера воздействуют четко определенным образом на поведение окна, независимо от типа модуля (IL или FBD), куда был установлен соответствующий триггер.

Все элементы управления остаются недоступными до тех пор, пока вы не перетащите в окно **Variables (Переменные)** хотя бы одну переменную. Элементы управления окна становятся доступными в верхней половине окна отладки серого цвета.

Кнопка	Команда	Описание
	Пуск/Останов	Этот элемент управления используется для запуска сеанса отслеживания триггера. Во время отслеживания триггера можно нажать эту кнопку, чтобы остановить сеанс. В противном случае сеанс автоматически завершается при выполнении определенных условий. Этой же кнопкой можно возобновить сеанс отслеживания триггера.
	Установить триггер шага	Этот элемент управления используется для выполнения триггера одного шага. Он активируется только при отсутствии активного сеанса отслеживания триггера и выбранной опции **None (Отсутствует)**. Заданное условие, указанное в этом случае, является активным. После срабатывания триггера одного шага сеанс отслеживания триггера завершается автоматически.

Счетчик срабатываний триггера

Элемент управления **Cnt**, доступный только для чтения, подсчитывает, сколько раз менеджер окна отладки был вызван с момента установки окна.

Менеджер окна автоматически сбрасывает этот счетчик при каждом запуске нового сеанса отслеживания триггера.

Состояние триггера

Этот элемент управления, доступный только для чтения, отображает состояние окна **Debug (Отладка)**. Он может принимать следующие значения.

Ready	Триггер не срабатывал во время выполнения задачи.
Triggered	Триггер срабатывал во время выполнения задачи.
Stop	Система не отслеживает срабатывание триггера. Она была остановлена пользователем, или было достигнуто условие остановки.
Error	Связь с целевым устройством прервана, состояние окна триггера определить невозможно.

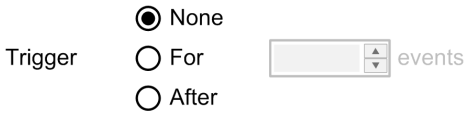
Заданное пользователем условие

Если вы определяете условие с помощью этого элемента управления, то значения в окне **Debug (Отладка)** будут обновляться каждый раз при вызове менеджера окна и соблюдении пользовательского условия.

После ввода условия элемент управления отображает выражение в упрощенном виде:

Счетчики

Эти элементы управления позволяют определять условия для счетчика срабатываний триггера:



Окно триггера может находиться в одном из следующих трех состояний.

- **None (Отсутствует)**: счетчик не был запущен, поэтому после срабатывания триггера не выполняются никакие условия.
- **For (До)**: допустим, в качестве предельного значения счетчика задано N. Менеджер окна будет прибавлять **1** к текущему значению счетчика и обновлять значение его переменных каждый раз при вызове окна отладки, а когда счетчик достигнет N, окно перестает обновлять значения и перейдет в состояние **Stop (Стоп)**.
- **After (После)**: допустим, в качестве предельного значения счетчика задано N. Менеджер окна сбрасывает счетчик и прибавляет **1** к текущему значению каждый раз при срабатывании триггера. Окно остается в состоянии **Ready (Готово)** и не обновляет значения своих переменных до тех пор, пока счетчик не достигнет N.

Отладка с использованием окон триггеров

Введение

Окно триггера позволяет выбрать набор переменных и синхронизировать отображение и обновление их значений во всплывающем окне. В отличие от окна **Watch (Контрольные значения)**, окна триггеров обновляют одновременно все содержащиеся в них переменные при каждом срабатывании триггера.

Открытие окна триггера из модуля IL

В данном примере мы исходим из того, что у вас есть модуль IL, содержащий следующие инструкции:

```
0001
0002 LD  a
0003 ADD b
0004 ST  a
0005
0006 LD  c
0007 ADD d
0008 ST  c
0009
0010 LD  k
0011 ADD 1
0012 ST  k
0013
```

Вы хотите проверить значения `b`, `d` и `k` непосредственно перед выполнением инструкции **ST k**.

Для этого наведите курсор на 12-ю строку:

```
0009
0010 LD  k
0011 ADD 1
0012 ST  k
0013
```

После этого выберите пункт **Debug (Отладка) → Add/remove text trigger (Добавить/удалить текстовый триггер)**.

В обоих случаях рядом с номером строки появляется зеленая стрелка и всплывает соответствующее окно триггера.

Не все инструкции IL поддерживают триггеры. Например, триггер невозможно поместить в начало строки, содержащей инструкцию `JMP`.

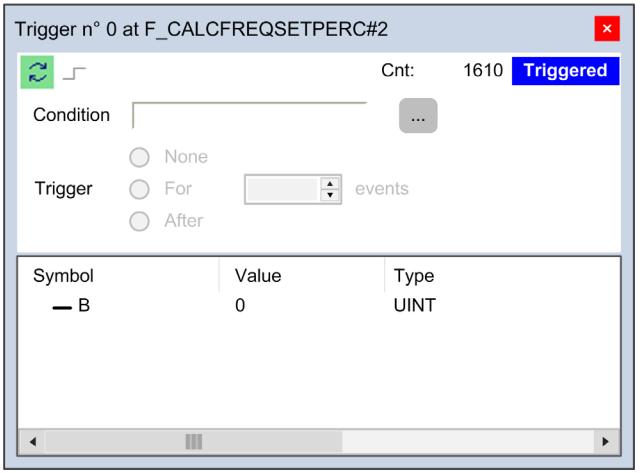
Добавление переменной в окно триггера из модуля IL

Чтобы отслеживать значение переменной, необходимо добавить ее в окно триггера.

Выберите переменную двойным щелчком мыши, а затем перетащите ее в окно **Variables (Переменные)**, представляющее собой нижнее белое поле во всплывающем окне:

```
0001
0002  LD   a
0003  ADD  b
0004  ST   a
0005
0006  LD   c
0007  ADD  d
0008  ST   c
0009
0010  LD   k
0011  ADD  1
0012  ST   k
0013
```

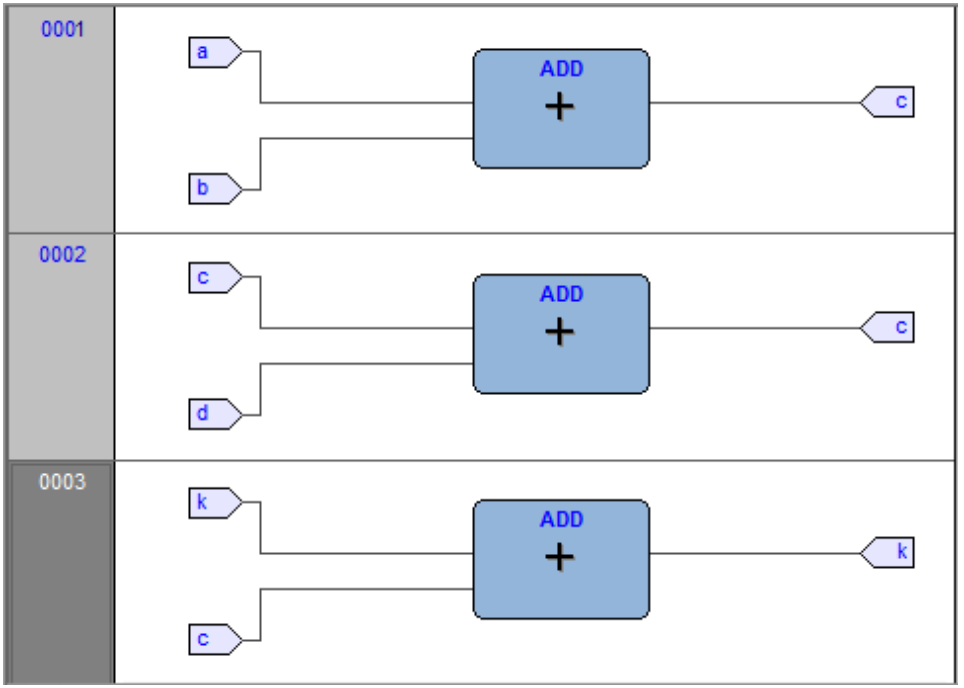
Имя переменной теперь отображается в столбце **Symbol (Символ)**:



Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

Открытие окна триггера из модуля FBD

В данном примере мы исходим из того, что у вас есть модуль FBD, содержащий следующие инструкции.

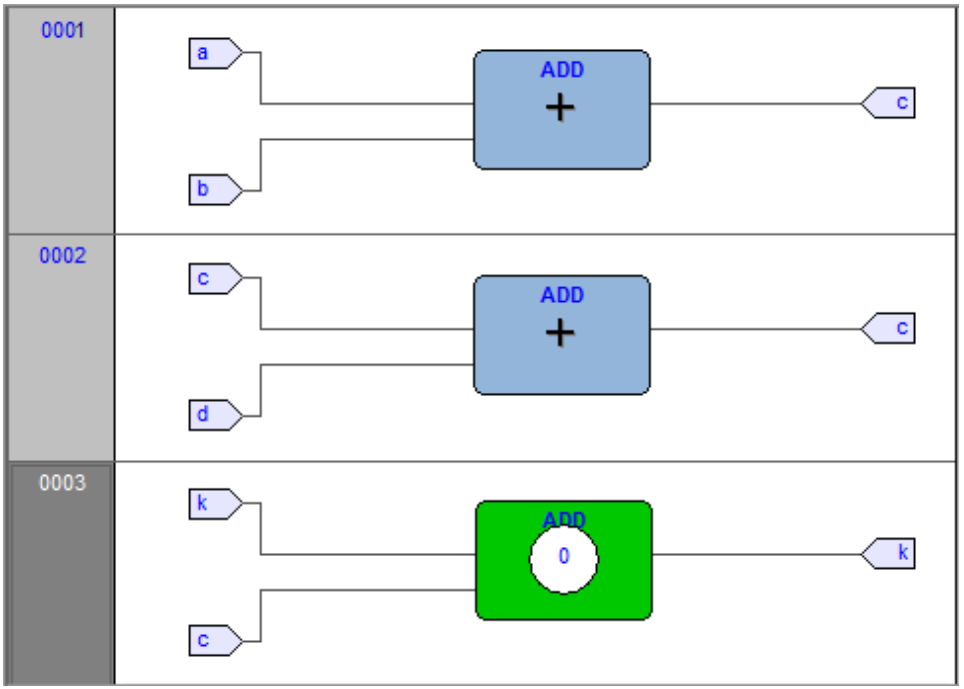


Вам необходимо проверять значения `C`, `D` и `K` непосредственно перед выполнением инструкции `ST k`.

Учитывая, что вам не удастся поместить триггер в блок, представляющий переменную, например , вы должны выбрать первый доступный блок, предшествующий выбранной переменной. В примере, изображенном на предыдущем рисунке, нужно переместить курсор на сеть № 3 и нажать на блок `ADD`.

Вы можете выбрать пункт **Debug (Отладка) → Add/remove text trigger (Добавить/удалить текстовый триггер)**.

В обоих случаях цвет выбранного блока меняется на зеленый, по центру блока появляется белый круг с номером внутри, при этом также всплывает соответствующее окно триггера:



В ходе предварительной обработки исходного кода на языке FBD компилятор транслирует его в инструкции на языке IL. Инструкция **ADD** в сети № 3 преобразуется в:

```
LD k
ADD 1
ST k
```

Когда вы добавляете триггер к блоку FBD, то помещаете этот триггер перед первой инструкцией в эквивалентном коде на языке IL.

Добавление переменной в окно триггера из модуля FBD

Чтобы отслеживать значение переменной, необходимо добавить ее в окно триггера. Например, вы хотите отслеживать значение переменной `k` в коде FBD.

Выберите пункт **Edit (Редактировать) → Watch mode (Режим контроля)**.

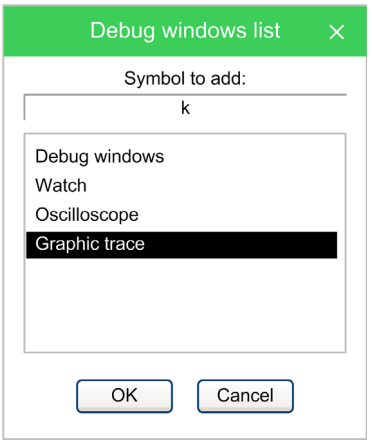
Курсор изменится следующим образом: .

Теперь вы можете щелкнуть блок, представляющий переменную, которую необходимо отобразить в окне триггера.

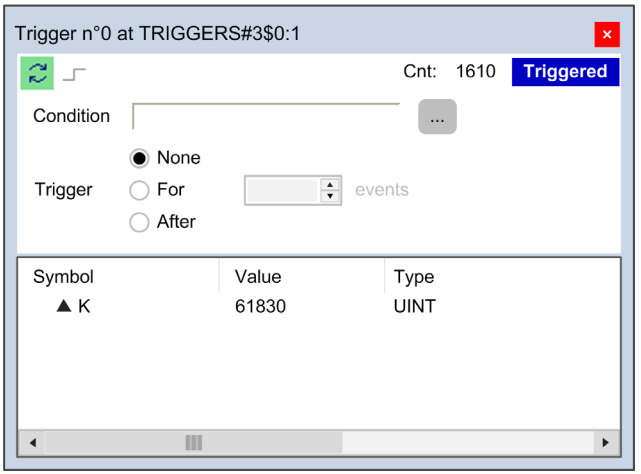
В данном примере нажмите на блок:

.

Появится диалоговое окно со списком всех существующих на данный момент экземпляров отладочных окон, при этом система спросит, в какое из них должен быть отправлен объект, по которому вы только что щелкнули:



Чтобы отобразить переменную `k` в окне триггера, выберите ее ссылку в столбце **Debug windows (Окна отладки)**, затем нажмите кнопку **OK**. Имя переменной теперь отображается в столбце **Symbol (Символ)**:

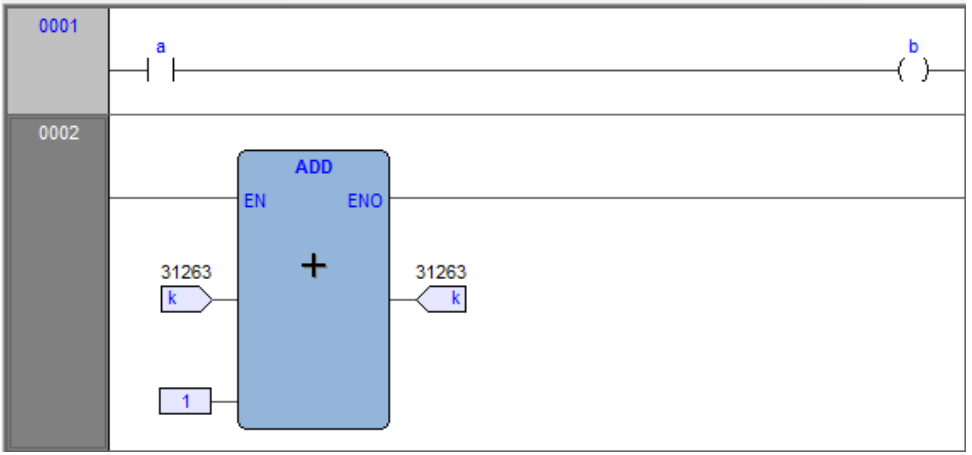


Та же процедура применяется ко всем переменным, которые необходимо отслеживать:

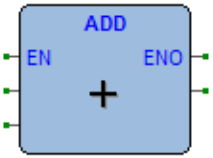
После добавления в окно **Graphic watch (Графические контрольные значения)** всех переменных, которые необходимо отобразить, выберите пункт **Edit (Редактировать) → Insert/Move mode (Режим вставки/перемещения)**, чтобы курсор принял свою изначальную форму.

Открытие окна триггера из модуля LD

В данном примере мы исходим из того, что у вас есть модуль LD, содержащий следующие инструкции:



Вы можете установить триггер на блок следующим образом:

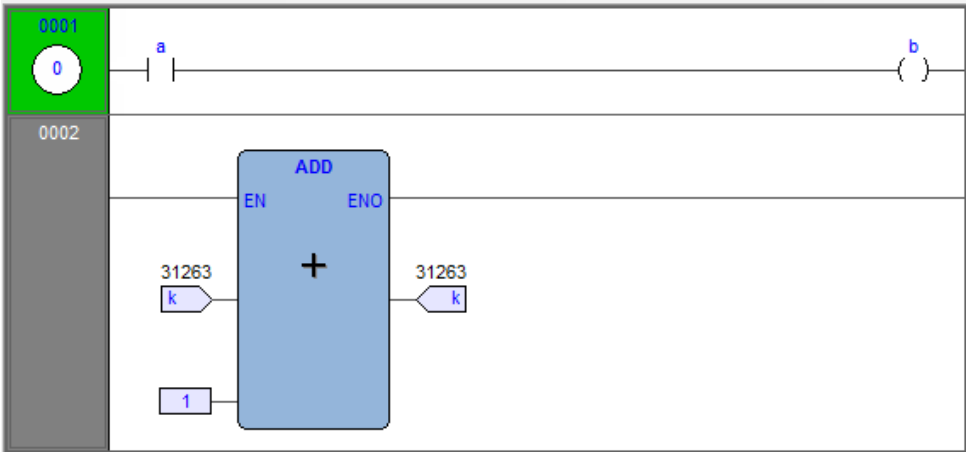


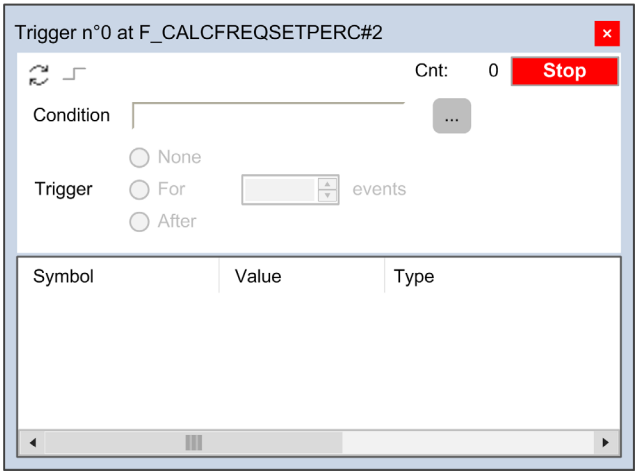
В этом случае действуют те же правила, что и при вставке триггера в модуль FBD на контакт или катушку.

В этом случае следуйте инструкциям SE. Допустим, что вы хотите проверять значения некоторых переменных каждый раз, когда процессор достигает сети № 1.

Сначала необходимо нажать на один из элементов, входящих в состав сети № 1. Теперь можно выбрать пункт **Debug (Отладка) → Add/remove text trigger (Добавить/удалить текстовый триггер).**

В обоих случаях серая область, содержащая номер сети, меняется на зеленый, по центру области появляется белый круг с номером триггера, при этом также всплывает соответствующее окно триггера:





В отличие от других языков, поддерживаемых модулем **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**, LD не позволяет вставить триггер в отдельный контакт или катушку — можно выбрать только целую сеть. Таким образом, переменные в окне триггера будут обновляться каждый раз, когда процессор достигает начала выбранной сети.

Добавление переменной в окно триггера из модуля LD

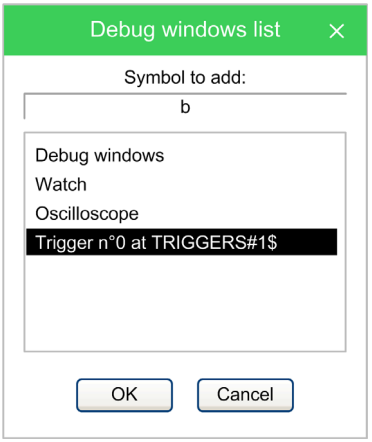
Чтобы отслеживать значение переменной, необходимо добавить ее в окно триггера. Например, вы хотите отслеживать значение переменной `b` в коде LD.

Выберите пункт **Edit (Редактировать) → Watch mode (Режим контроля)**.

Курсор изменится следующим образом: .

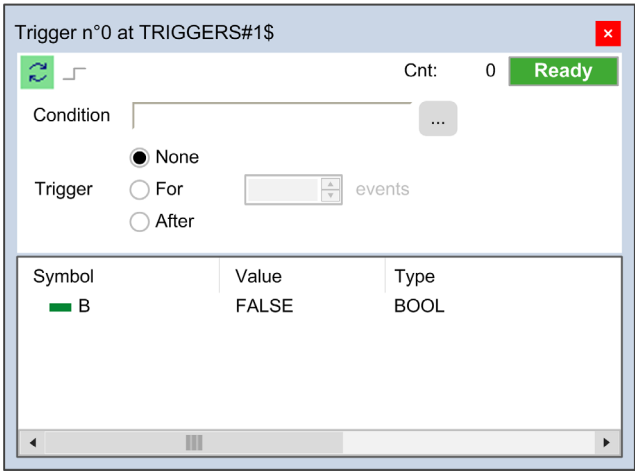
Теперь вы можете щелкнуть по элементу, представляющему переменную, которую необходимо отобразить в окне триггера.

Появится диалоговое окно со списком всех существующих на данный момент экземпляров отладочных окон, при этом система спросит, в какое из них должен быть отправлен объект, по которому вы только что щелкнули:



Чтобы отобразить переменную `B` в окне триггера, выберите ее ссылку в столбце **Debug windows (Окна отладки)**, затем нажмите кнопку **OK**.

Имя переменной теперь отображается в столбце **Symbol (Символ)**:



Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

Открытие окна триггера из модуля ST

В данном примере мы исходим из того, что у вас есть модуль ST, содержащий следующие инструкции:

```
0001
0002  a := b * b;
0003  c := c + SHR( a, 16#04 );
0004
0005  d := e * e;
0006  f := f + SHR( d, 16#04 );
0007
```

Предположим также, что вы хотите проверить значения `e`, `d` и `f` непосредственно перед выполнением инструкции `f := f+ SHR( d, 16#04 )`.

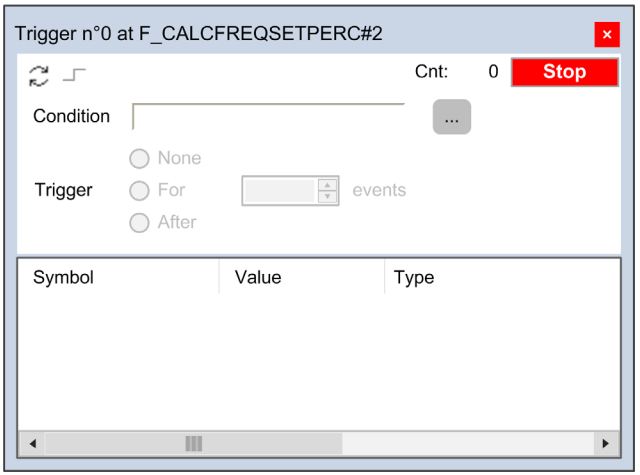
Для этого наведите курсор на шестую строку.

После этого выберите пункт **Debug (Отладка) → Add/remove text trigger (Добавить/удалить текстовый триггер)**.

В обоих случаях рядом с номером строки появляется зеленая стрелка:

```
0001
0002  a := b * b;
0003  c := c + SHR( a, 16#04 );
0004
0005  d := e * e;
0006  f := f + SHR( d, 16#04 );
0007
```

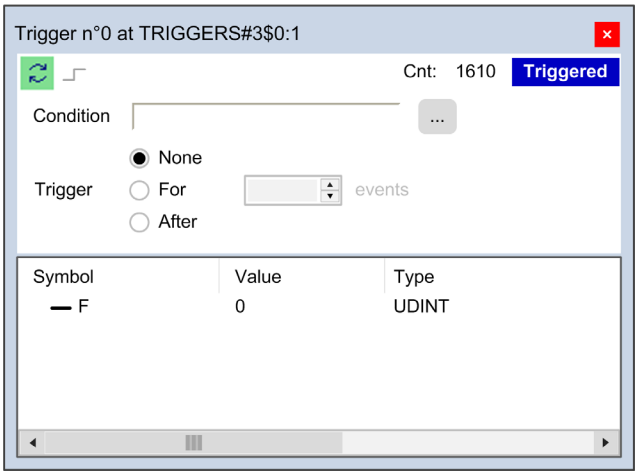
И всплывает соответствующее окно триггера:



Не все инструкции ST поддерживают триггеры. Например, невозможно поместить триггер на строку, содержащую такой закрывающий элемент, как `END_IF`, `END_FOR`, `END_WHILE` и так далее.

Добавление переменной в окно триггера из модуля ST

Чтобы отслеживать значение переменной, необходимо добавить ее в окно триггера. Выберите переменную двойным щелчком мыши, а затем перетащите ее в окно **Variables (Переменные)**, представляющее собой нижнее белое поле во всплывающем окне. Имя переменной теперь отображается в столбце **Symbol (Символ)**:



Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

Удаление переменной из окна триггера

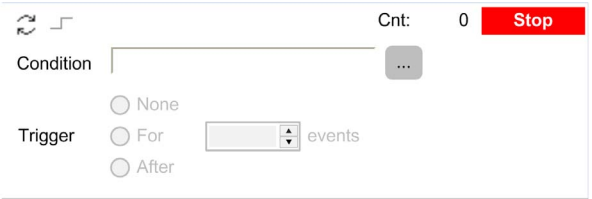
Если вы хотите удалить переменную из окна триггера, выделите ее, щелкнув один раз по ее имени, затем нажмите клавишу **Delete**.

Использование элементов управления

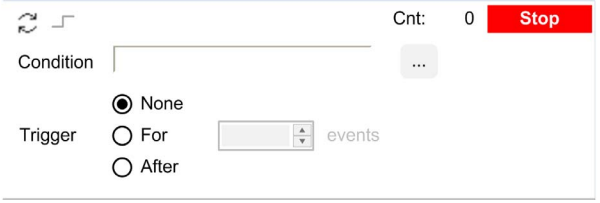
Элементы управления окна триггера позволяют контролировать работу этого инструмента отладки. Основная задача элементов управления окном триггера — дать возможность определить больше ограничивающих условий, при соблюдении которых переменные в окне **Variables (Переменные)** будут обновляться по достижении процессором положения триггера. Если вы не используете элементы управления, переменные обновляются каждый раз, когда процессор достигает соответствующий триггер.

Активация элементов управления

При установке триггера все элементы в окне **Controls (Элементы управления)** деактивируются:



Вы не сможете получить доступ ни к одному из элементов управления, пока не перетащите хотя бы одну переменную в окно **Debug (Отладка)**. После выполнения этого действия автоматически запускается отслеживание триггера, и окно **Controls (Элементы управления)** меняется следующим образом:



Отслеживание триггера можно запустить/остановить с помощью соответствующей кнопки: .

Исправление количества обновлений

Если вы хотите, чтобы значения обновлялись при первом вызове окна, выберите **None (Отсутствует)** и нажмите кнопку триггера одного шага , в противном случае выберите **For (До)** и установите для событий значение **1**.

Если вы хотите, чтобы значения в окне обновлялись при первых **X** срабатываниях триггера, выберите **For (До)** и установите для событий значение **X**.

Если вы хотите, чтобы значения в окне обновлялись после срабатывания триггера **Y** раз, выберите **After (После)** и установите для событий значение **Y**.

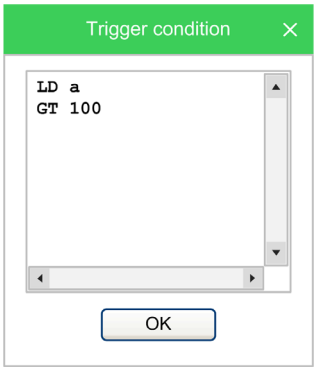
Настройки триггера и условия активируются при (повторном) запуске триггера.

Определение условия

Этот элемент управления позволяет задать условие срабатывания триггера. По умолчанию для этого условия установлено значение **TRUE**, а значения в окне отладки обновляются каждый раз при вызове менеджера окна.

Если вы хотите установить ограничение для механизма обновления, можете задать условие, нажав на кнопку ... .

Появится текстовое окно, в котором можно написать код IL, задающий следующее условие:



По завершении написания кода условия нажмите кнопку **OK**, чтобы установить его, или нажмите клавишу **Esc**, чтобы отменить. Если вы установите его, то значения в окне отладки будут обновляться каждый раз при вызове менеджера окна и соблюдении пользовательского условия.

Теперь элемент управления отобразит условие в упрощенном виде:

Для изменения снова нажмите кнопку .

Откроется окно условий срабатывания триггера, содержащее написанный вами код на языке IL, который теперь можно редактировать.

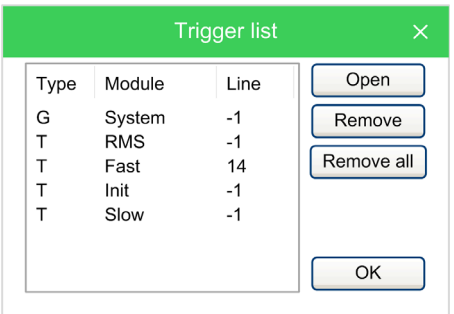
Чтобы полностью удалить условие, удалите весь код IL в окне, затем нажмите **OK**.

Результат кода условия должен быть логическим значением (**TRUE** или **FALSE**), в противном случае произойдет ошибка компилятора.

В коде условия могут использоваться только глобальные и перетаскиваемые переменные. Другими словами, все переменные, являющиеся локальными для программного модуля, куда изначально был установлен триггер, будут недействительны, если вы их не перетащили в окно отладки. В окне условий нельзя объявлять новые переменные.

Закрытие окна триггера и удаление триггера

Вы можете выполнить несколько действий по завершении сеанса отладки с помощью окна триггера:



Закрытие окна триггера

Если вы закончили отслеживание набора переменных с помощью окна триггера, при желании можно закрыть окно **Debug (Отладка)**, не удаляя триггер. Если нажать кнопку в правом верхнем углу, то окно интерфейса будет скрыто, а менеджер окна и соответствующий триггер будут продолжать работать.

Чтобы возобновить отладку с помощью ранее скрытого окна триггера, необходимо открыть окно **Trigger list (Список триггеров)**, выбрать запись, относящуюся к этому окну триггера, и нажать кнопку **Open (Открыть)**.

Окно интерфейса отображается с обновленным значением переменных и счетчиком триггеров, как если бы оно не было закрыто.

Удаление триггера

При выборе этой опции вы полностью удаляете код менеджера окна и его триггера. Откройте окно **Trigger list (Список триггеров)**, выберите запись, относящуюся к окну триггера, который вы хотите удалить, и нажмите кнопку **Remove (Удалить)**.

Кроме того, вы можете переместить курсор на строку (если модуль находится в режиме IL или ST) или щелкнуть блок (если модуль находится в режиме FBD или LD), куда вы поместили триггер.

Теперь нажмите кнопку **Add/remove text trigger (Добавить/удалить текстовый триггер)** на панели инструментов **Debug (Отладка)**.

Удаление всех триггеров

Также вы можете удалить все имеющиеся триггеры за один раз, независимо от того, какие записи выбраны. Для этого нажмите кнопку **Remove all triggers (Удалить все триггеры)**.

Раздел 13.7
Графические триггеры

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Окно графического триггера	347
Отладка с помощью окна графического триггера	354

Окно графического триггера

Описание

Окно графического триггера позволяет выбрать набор переменных, синхронизировать их выборку и отобразить их кривую в специальном всплывающем окне.

Выборка перетаскиваемых переменных происходит каждый раз, когда процессор достигает позиции (инструкции IL/ST или блока FBD/LD), куда был помещен триггер.

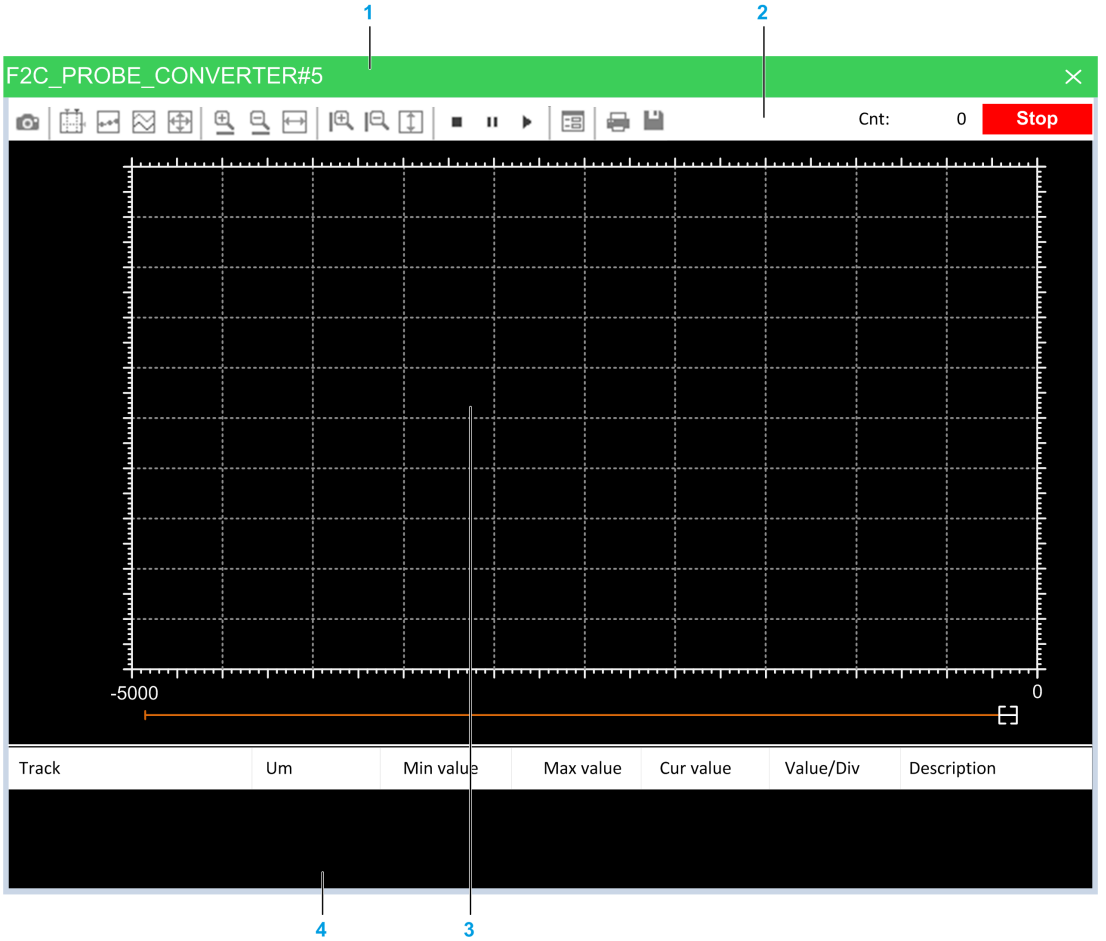
Предварительные условия для открытия окна графического триггера

Объем доступной памяти

Если свободной памяти в секторе кода приложения недостаточно, выводится соответствующее сообщение об ошибке. В этом случае необходимо удалить некоторые объекты из памяти, чтобы сделать эту функцию доступной.

Интерфейс окна графического триггера

При установке графического триггера открывается окно, которое называется окном **интерфейса**. Это основной интерфейс для доступа к отладочным функциям окна графического триггера. Окно состоит из нескольких элементов, как показано ниже:



1. Строка заголовка
2. Панель элементов управления
3. Область графика
4. Окно переменных

Строка заголовка

В строке **заголовка** в верхней части всплывающего окна отображается информация о местоположении триггера, активирующего выборку переменных, перечисленных в окне **Variables (Переменные)**.

Текст в заголовке имеет следующий формат:

`НазваниеМодуля#Местоположение`

`НазваниеМодуля`	Имя программы, функции или функционального блока, где установлен триггер.

`Местоположение`	Точное расположение триггера внутри модуля `НазваниеМодуля`.
	Если речь идет о `НазваниеМодуля` в коде на языке IL или ST, то `Местоположение` имеет следующий формат:
	`N1`
	Если же `НазваниеМодуля` используется в коде на языке FBD или LD, то применяется следующий формат:
	`N2$BT:BID`
	`N1` = номер строки с инструкцией;
	`N2` = номер сети;
	`BT` = тип блока (операнд, функция, функциональный блок и т. д.);
	`BID` = идентификатор блока.

Панель элементов управления

Панель элементов управления позволяет управлять работой окна графического триггера. Подробное описание функции каждого элемента управления приводится в разделе с описанием элементов управления окна **графического триггера**. См. описание в разделе «Элементы управления окна графического триггера» *(стр. 351)*.

Область графика

Область **графика** содержит шесть элементов:

- График: в этой области отображается график кривой изменения значений перетаскиваемых переменных.
- Выборки для сбора: количество выборок для сбора менеджером окна графического триггера.
- Горизонтальный курсор: курсор, определяющий горизонтальную линию. Значение каждой переменной в точке пересечения с этой линией выводится в столбце **horz cursor (гориз. курсор)**.
- Синий курсор: курсор, определяющий вертикальную линию. Значение каждой переменной в точке пересечения с этой линией выводится в столбце **left cursor (левый курсор)**.
- Красный курсор: курсор, определяющий вертикальную линию. Значение каждой переменной в точке пересечения с этой линией выводится в столбце **left cursor (левый курсор)**.
- Полоса прокрутки: если масштаб оси X слишком велик для отображения всех выборок в области **графика**, то полоса прокрутки позволяет перемещаться вперед и назад по горизонтальной оси.

Окно переменных

Эта нижняя часть окна **Debug (Отладка)** представляет собой таблицу, каждая строка которой соответствует переменной, которую вы перетащили.

Окно графического триггера: окно переменных

Чтобы посмотреть переменную, необходимо скопировать ее в нижнюю часть окна **Debug (Отладка)**:

Track	Um	Min value	Max value	Cur value	Value/Div
PIDOUTPUT		-11.963	11.964	-11.952	2.99089
PIDFEEDBACK		-10.710	10.710	-7.685	2.67756

Эта нижняя часть окна **Debug (Отладка)** представляет собой таблицу, где строки соответствуют переменным, которые вы перетащили. Каждая строка имеет несколько полей, как показано в следующей таблице.

Поле	Описание
Track (Кривая)	Имя переменной.
Um (Ед. измерения)	Единица измерения.
Min value (Мин. знач.)	Минимальное значение в наборе записей.
Max value (Макс. значение)	Максимальное значение в наборе записей.
Cur value (Тек. значение)	Последнее обновленное значение переменной.
Value/Div	Сколько инженерных единиц представлено единицей оси Y (т. е. пространство между двумя тактами на вертикальной оси).
V blue curs (Син. верт. курсор)	Значение переменной на пересечении с линией, обозначенной синим вертикальным курсором.
V red curs (Красн. верт. курсор)	Значение переменной на пересечении с линией, обозначенной красным вертикальным курсором.
H blue curs (Син. гор. курсор)	Значение переменной на пересечении с линией, обозначенной синим горизонтальным курсором.
H red curs (Красн. гор. курсор)	Значение переменной на пересечении с линией, обозначенной красным горизонтальным курсором.

В окно графического триггера можно перетащить только переменные, являющиеся локальными для модуля, где вы установили соответствующий триггер, или глобальные переменные и параметры. Нельзя перетаскивать переменные, объявленные в другой программе, функции или функциональном блоке.

Выборка переменных

Значение переменных обновляется каждый раз при вызове менеджера окна графического триггера, т. е. каждый раз, когда процессор выполняет инструкцию, помеченную зеленой стрелкой. Однако вы можете настроить элементы управления таким образом, чтобы выборка переменных производилась в тех случаях, когда триггеры также удовлетворяют дополнительным ограничительным условиям, заданным лично вами.

Значение переменных в столбце **Track (Кривая)** считывается из памяти непосредственно перед отмеченной инструкцией и непосредственно после предыдущей инструкции.

Элементы управления окна графического триггера

Элементы управления позволяют подробно указывать, когда модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** должен выполнять выборку переменных, добавленных в окно **Variables (Переменные)**.

Элементы управления окна графического триггера воздействуют на поведение окна независимо от типа модуля (IL, ST, FBD или LD), куда был установлен соответствующий триггер.

Элементы управления окна становятся доступными на панели **элементов управления** окна отладки.

Кнопка	Команда	Описание
	Запуск графического отслеживания	При нажатии этой кнопки начинается сбор данных. Если повторно нажать кнопку после запуска сбора данных, процесс выборки остановится, а все собранные данные будут сброшены.
	Показать измерительные направляющие	Пока нажата эта кнопка, отображаются два курсора (красный и синий), которые можно перемещать вдоль соответствующих осей. Нажмите кнопку еще раз, если хотите скрыть все курсоры.
	Показать выборки	Этот элемент управления используется для точного отображения точек, в которых произошло срабатывание триггера переменных при каждой выборке.
	Вертикальное разделение	Если этот элемент управления нажат, ось Y разбивается на сегменты, количество которых соответствует количеству перетаскиваемых переменных, так что диаграмма каждой переменной отрисовывается в отдельной области.
	Показать все значения	Используется для заполнения окна графика всеми значениями, отобранными для выбранных переменных в текущем наборе записей.
	Увеличение по горизонтали + и **Уменьшение по горизонтали –**	При увеличении кривые в области **графика** отображаются на экране крупнее, что позволяет рассмотреть их более подробно. При уменьшении кривые отображаются на экране в уменьшенном масштабе, что позволяет рассмотреть их целиком. Горизонтальное масштабирование действует только по горизонтальной оси.
	Показать все по горизонтали	Этот элемент управления используется для центрирования по горизонтали результатов выборки для набора записей. Таким образом, первый элемент выборки размещается на левом поле, а последний — на правом поле графического окна.
	Увеличение по вертикали + и **Уменьшение по вертикали –**	При увеличении кривые в области **графика** отображаются на экране крупнее, что позволяет рассмотреть их более подробно. При уменьшении кривые отображаются на экране в уменьшенном масштабе, что позволяет рассмотреть их целиком. Вертикальное масштабирование действует только по вертикальной оси.
	Показать все по вертикали	Этот элемент управления используется для центрирования по вертикали результатов выборки для набора записей. Таким образом, элемент выборки с максимальным значением располагается вблизи верхнего поля графического окна, а элемент выборки с минимальным значением — в нижнем.
	Остановить сбор данных	Этот элемент управления используется для остановки сбора данных.
	Временно приостановить сбор данных	Нажмите эту кнопку, чтобы временно приостановить сбор данных.
	Перезапустить сбор данных	Нажмите эту кнопку, чтобы перезапустить сбор данных.

Кнопка	Команда	Описание
	Свойства графика	При нажатии этой кнопки появляется диалоговое окно с вкладками, позволяющее установить общие пользовательские параметры, влияющие на работу окна графического триггера. Для получения дополнительной информации см. «Свойства окна графического триггера» *(см. стр. 353)*.
	Печать графика	Нажмите эту кнопку, чтобы распечатать **область графика** и окно **переменных**.
	Сохранить график	Нажмите эту кнопку, чтобы сохранить график.

Счетчик срабатываний триггера

Этот элемент управления, доступный только для чтения, отображает два числа в следующем формате: `Cnt: X/Y`.

- `X` означает, сколько раз менеджер окна отладки был вызван с момента установки графического триггера.
- `Y` соответствует количеству элементов выборки, которое графическое окно должно собрать до остановки сбора данных и отрисовки кривых.

Состояние триггера

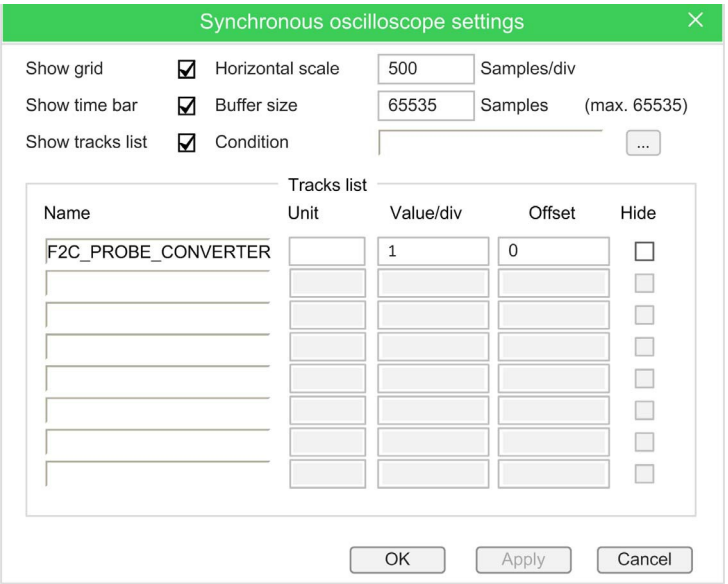
Этот элемент управления, доступный только для чтения, отображает состояние окна **Debug (Отладка)**. Он может принимать следующие значения.

Ready	Выборка не производилась, поскольку триггер не срабатывал во время выполнения текущей задачи.
Triggered	Выборка произведена, поскольку триггер срабатывал во время выполнения текущей задачи.
Stop	Счетчик триггера показывает, что было собрано несколько элементов выборки, удовлетворяющих запросу пользователя или ограничениям памяти, поэтому процесс сбора данных останавливается.
Error	Связь с целевым устройством прервана, состояние окна триггера определить невозможно.

Свойства окна графического триггера

Чтобы открыть окно свойств, необходимо нажать кнопку **Graph properties (Свойства графика)** на панели **элементов управления**. Откроется диалоговое окно **Synchronous oscilloscope settings** (**Параметры синхронного осциллографа)**.

Общие



Элементы управления

Элемент	Описание
Show grid (Показать сетку)	Установите этот флажок, чтобы отобразить фоновую сетку в **области графика**.
Show time bar (Показать панель времени)	Если установлен этот флажок, в нижней части **области графика** отображается полоса прокрутки.
Show tracks list (Показать список кривых)	Если установлен этот флажок, будет отображаться окно **Variables (Переменные)**, в противном случае **область графика** растягивается до нижней границы окна графического триггера.

Значения

Элемент	Описание
Horizontal scale (Горизонтальная шкала)	Количество элементов выборки на единицу оси X. Единицей оси X является пространство между двумя вертикальными линиями фоновой сетки.
Buffer size (Размер буфера)	Количество элементов выборки для сбора. Когда вы открываете окно **Synchronous oscilloscope settings (Настройки синхронного осциллографа)** после предварительного перетаскивания всех отслеживаемых переменных, в этом поле выводится число по умолчанию, представляющее максимальное количество элементов выборки, которое вы можете собрать для каждой переменной.

Заданное пользователем условие

Если вы определяете условие с помощью этого элемента управления, процесс выборки не начнется до тех пор, пока это условие не будет выполнено. В отличие от окон триггера, после начала сбора данных выборка производится каждый раз при вызове менеджера окна, даже если условие больше не соблюдается.

После ввода условия элемент управления отображает выражение в упрощенном виде:

Condition | A GT 100 | ...

Список кривых

Этот раздел позволяет определить некоторые графические свойства кривой каждой переменной. Чтобы выбрать переменную, щелкните ее имя в столбце **Name (Имя)**.

Элемент	Описание
Unit (Единица)	Единица измерения, отображаемая в таблице окна **Variables (Переменные)**.
Value/div (Знач./дел.)	Значения на единицу оси Y. Единицей оси Y является пространство между двумя горизонтальными линиями фоновой сетки.
Offset (Смещение)	Используется для применения значения смещения на графике.
Hide (Скрыть)	Установите этот флажок, чтобы скрыть выбранную кривую на графике.

Нажмите кнопку **Apply (Применить)**, чтобы изменения вступили в силу, или **OK**, чтобы применить изменения и закрыть окно **Synchronous oscilloscope settings (Настройки синхронного осциллографа)**.

Отладка с помощью окна графического триггера

Описание

Окно графического триггера позволяет выбрать набор переменных, синхронизировать их выборку и отобразить на графике в специальном всплывающем окне.

Открытие окна графического триггера из модуля IL

В данном примере мы исходим из того, что у вас есть модуль IL, содержащий следующие инструкции.

```
0001
0002 LD  a
0003 ADD b
0004 ST  a
0005
0006 LD  c
0007 ADD d
0008 ST  c
0009
0010 LD  k
0011 ADD 1
0012 ST  k
0013
```

Предположим, вы хотите проверить значения `b`, `d` и `k` непосредственно перед выполнением инструкции **ST k**. Для этого наведите курсор на 12-ю строку:

```
0009
0010 LD  k
0011 ADD 1
0012 ST  k
0013
```

После этого выберите пункт **Debug (Отладка) → Add/remove graphic trigger (Добавить/удалить графический триггер)**.

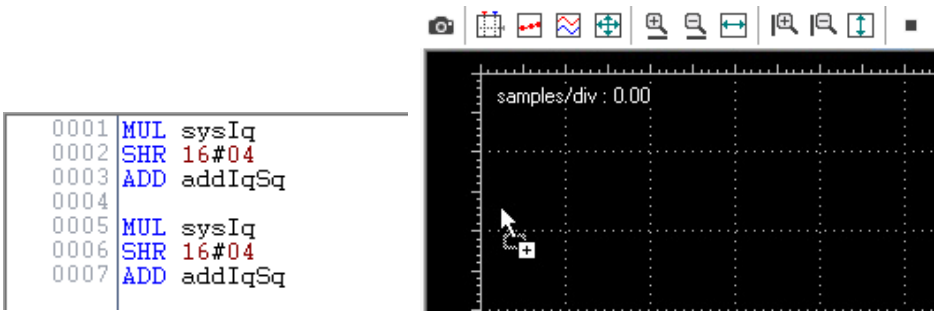
Рядом с номером строки появляется зеленая стрелка и всплывает окно графического триггера:

```
0001
0002 LD  a
0003 ADD b
0004 ST  a
0005
0006 LD  c
0007 ADD d
0008 ST  c
0009
0010 LD  k
0011 ADD 1
0012 ST  k
0013
```

Не все инструкции IL поддерживают триггеры. Например, триггер невозможно поместить в начало строки, содержащей инструкцию `JMP`.

Добавление переменной в окно графического триггера из модуля IL

Чтобы отобразить график переменной, необходимо добавить ее в окно графического триггера. Выберите переменную, дважды щелкнув по ней, а затем перетащите ее в раздел **Variables (Переменные)**. Имя переменной теперь отображается в столбце **Track (Кривая)**:

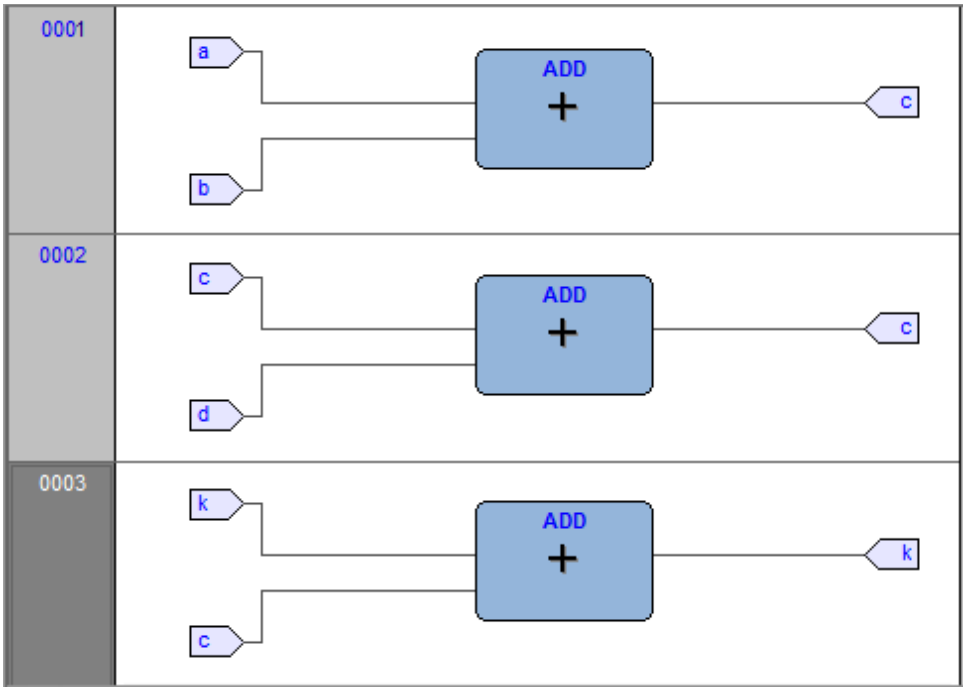


Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

Как только первая переменная попадает в область графического отслеживания, автоматически открывается окно **Graphic properties (Графические свойства)**, в котором можно настроить параметры выборки и визуализации.

Открытие окна графического триггера из модуля FBD

В данном примере мы исходим из того, что у вас есть модуль FBD, содержащий следующие инструкции:



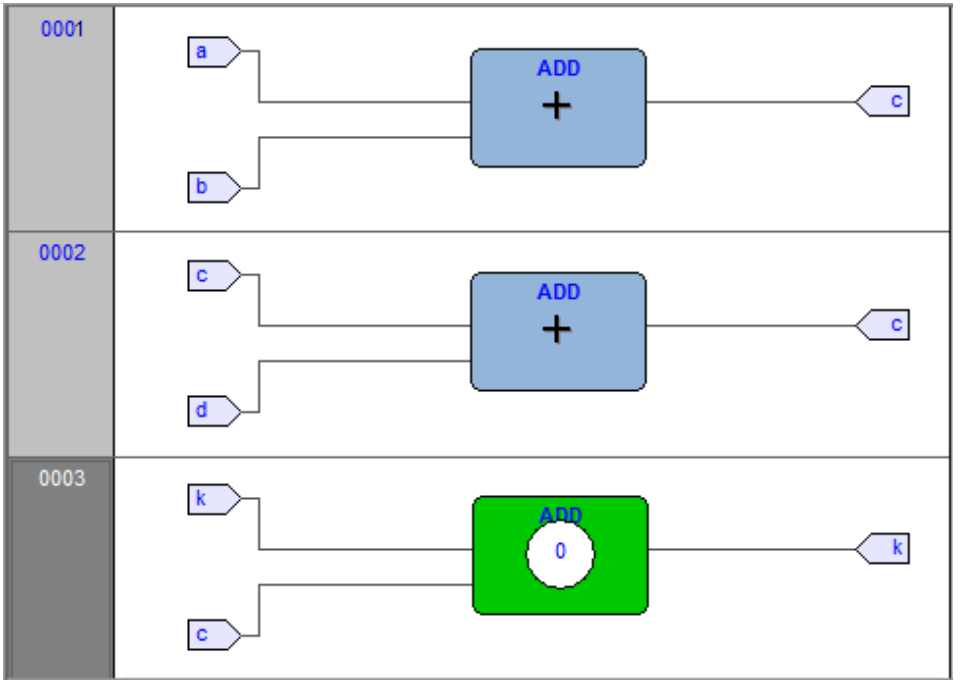
Предположим, вы хотите проверять значения `c`, `d` и `k` непосредственно перед выполнением инструкции `ST k`.

Вы не сможете установить триггер на блок, представляющий переменную, например

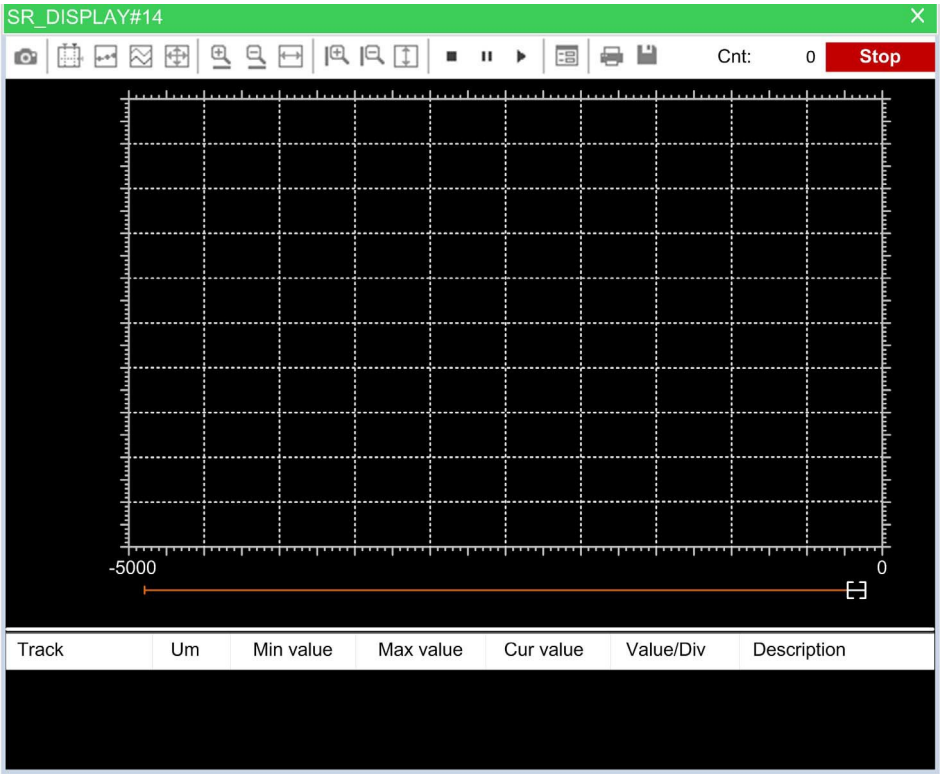
. Необходимо выбрать первый доступный блок, предшествующий выбранной переменной. В примере, изображенном на предыдущем рисунке, нужно переместить курсор на сеть № 3 и нажать на блок `ADD`.

После этого выберите пункт **Debug (Отладка) → Add/remove graphic trigger (Добавить/удалить графический триггер)**.

При этом цвет выбранного блока становится зеленым, по центру блока появляется белый круг с идентификатором триггера:



Также всплывает соответствующее окно триггера:



В ходе предварительной обработки исходного кода на языке FBD компилятор транслирует его в инструкции на языке IL. Инструкция **ADD** в сети № 3 преобразуется в:

```
LD k
ADD c
ST k
```

Когда вы добавляете триггер к блоку FBD, то помещаете этот триггер перед первой инструкцией в эквивалентном коде на языке IL.

Добавление переменной в окно графического триггера из модуля FBD

Чтобы отслеживать переменную, необходимо добавить ее в окно триггера. Предположим, вы хотите просмотреть график переменной **k** в коде FBD.

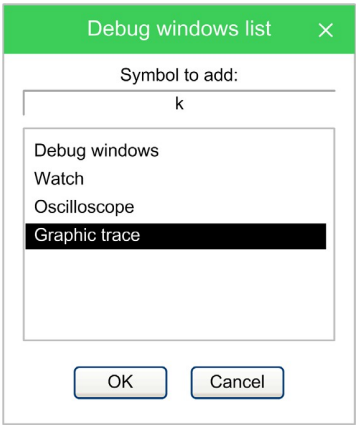
Выберите пункт **Edit (Редактировать)** → **Watch mode (Режим контроля)**.

Курсор изменится следующим образом: .

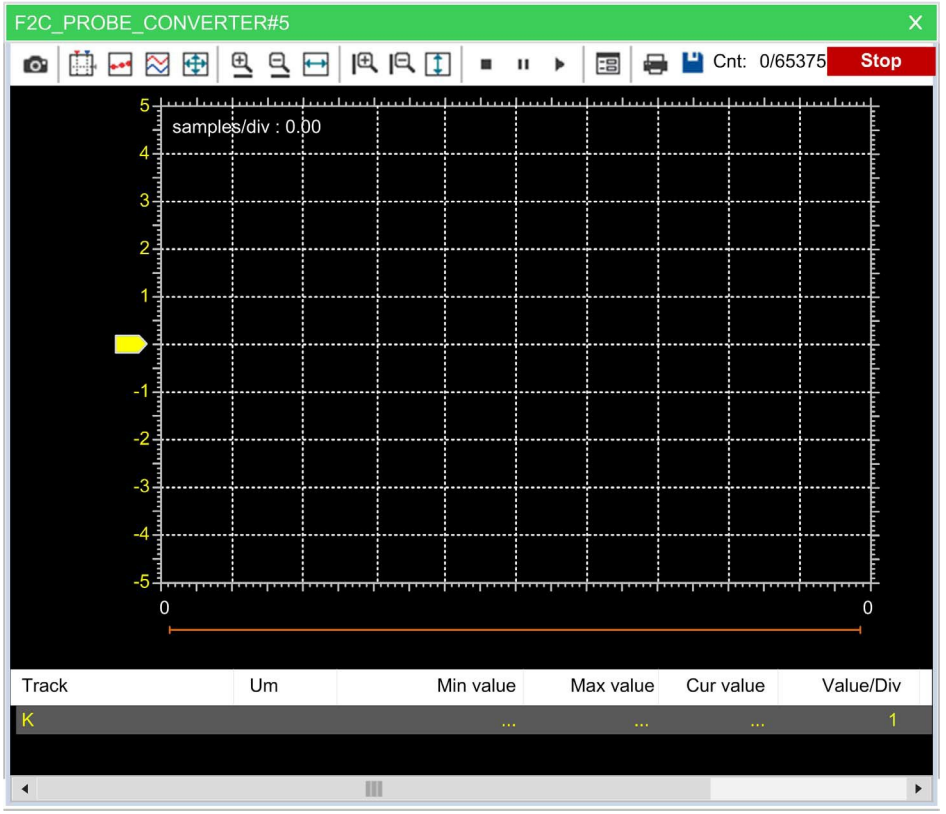
Теперь вы можете щелкнуть блок, представляющий переменную, которую необходимо отобразить в окне графического триггера.

В данном примере нажмите на блок .

Появится диалоговое окно со списком всех существующих на данный момент экземпляров отладочных окон, при этом система спросит, в какое из них должен быть отправлен объект, по которому вы только что щелкнули:



Чтобы построить кривую переменной **k**, в столбце **Debug windows (Окна отладки)** выберите **Graphic Trace (Графическое отслеживание)**, а затем нажмите **OK**. Имя переменной теперь отображается в столбце **Track (Кривая)**:



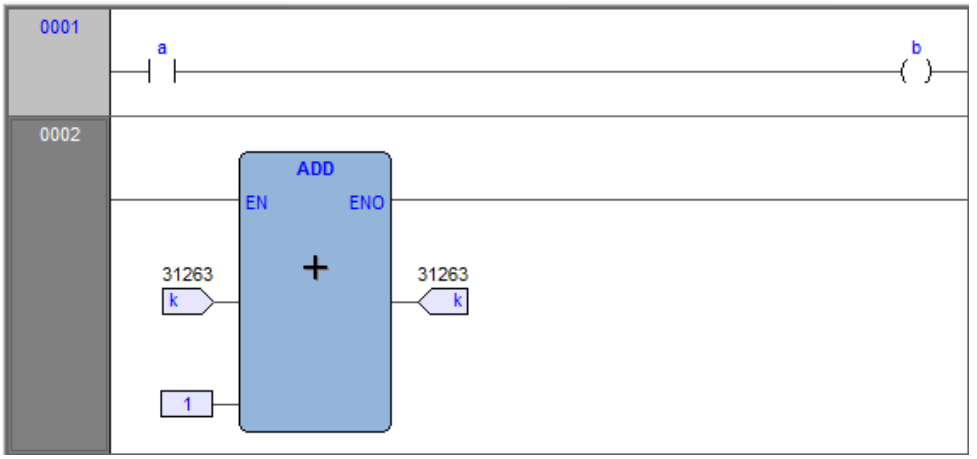
Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

После добавления в окно **Graphic watch (Графические контрольные значения)** всех переменных, которые необходимо отобразить, щелкните **Edit (Редактировать) → Insert/Move mode (Режим вставки/перемещения)**, чтобы курсор принял свою изначальную форму.

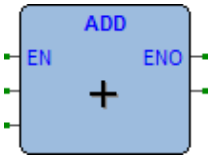
Как только первая переменная попадает в область графического отслеживания, автоматически открывается окно **Graphic properties (Графические свойства)**, в котором можно настроить параметры выборки и визуализации.

Открытие окна графического триггера из модуля LD

В данном примере мы исходим из того, что у вас есть модуль LD, содержащий следующие инструкции:



Вы можете установить триггер на блок следующим образом:

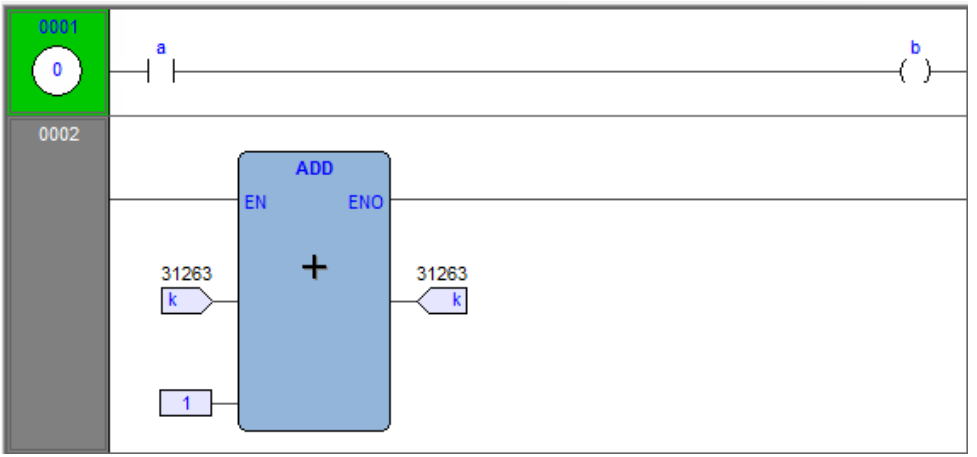


В этом случае действуют те же правила, что и при вставке графического триггера в модуль FBD.

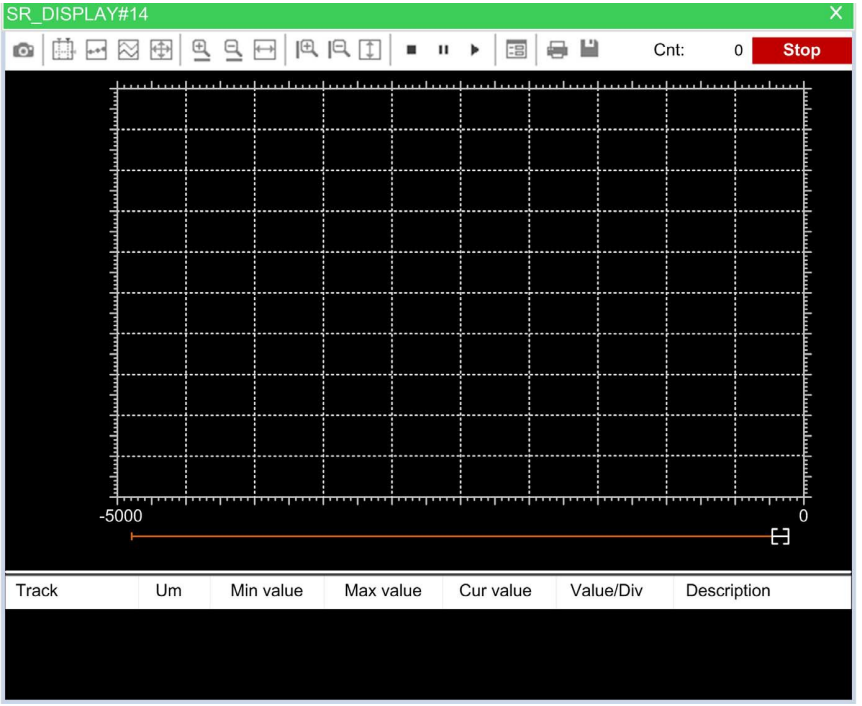
Допустим, что вы хотите проверять значения некоторых переменных каждый раз, когда процессор достигает сети № 1.

Щелкните по одному из элементов, составляющих сеть № 1, а затем выберите пункт **Debug (Отладка) → Add/remove graphic trigger (Вставить/удалить графический триггер)**.

При этом серая область с номером сети станет зеленой, по центру появится белый круг с номером внутри:



И всплывет окно графического триггера:



ПРИМЕЧАНИЕ. В отличие от других языков, поддерживаемых модулем **PROGRAMMING (ПРОГРАММИРОВАНИЕ)**, LD не позволяет вставить триггер перед отдельным контактом или катушкой — можно выбрать только целую сеть. Таким образом, выборка переменных в окне **Graphic trigger (Графический триггер)** будет производиться каждый раз, когда процессор достигает начала выбранной сети.

Добавление переменной в окно графического триггера из модуля LD

Чтобы просмотреть график переменной, необходимо добавить ее в окно **Graphic trigger (Графический триггер)**. Предположим, вы хотите просмотреть график переменной `b` в коде LD.

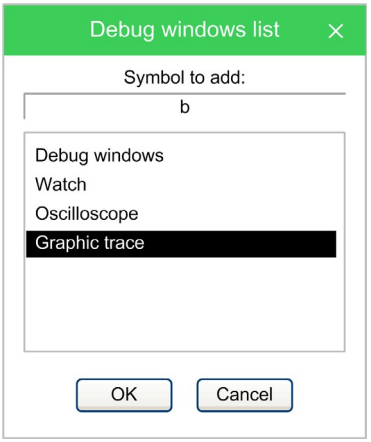
Выберите пункт **Edit (Редактировать) → Watch mode (Режим контроля)**.

Курсор изменится следующим образом:

Теперь вы можете щелкнуть по элементу, представляющему переменную, которую необходимо отобразить в окне **Graphic trigger (Графический триггер)**.

Появится диалоговое окно со списком всех существующих на данный момент экземпляров отладочных окон, при этом система спросит, в какое из них должен быть отправлен объект, по которому вы только что щелкнули.

Чтобы построить кривую переменной **b**, в столбце **Debug windows (Окна отладки)** выберите **Graphic Trace (Графическое отслеживание)**, а затем нажмите **OK**. Имя переменной теперь отображается в столбце **Track (Кривая)**:



Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

После добавления в окно **Graphic watch (Графические контрольные значения)** всех переменных, которые необходимо отобразить, щелкните **Edit (Редактировать) → Insert/Move mode (Режим вставки/перемещения)**, чтобы курсор принял свою изначальную форму.

Как только первая переменная попадает в область графического отслеживания, автоматически открывается окно **Graphic properties (Графические свойства)**, в котором можно настроить параметры выборки и визуализации.

Открытие окна графического триггера из модуля ST

В данном примере мы исходим из того, что у вас есть модуль ST, содержащий следующие инструкции:

```
0001
0002  a := b * b;
0003  c := c + SHR( a, 16#04 );
0004
0005  d := e * e;
0006  f := f + SHR( d, 16#04 );
0007
```

Предположим, вы хотите проверить значения `e`, `d` и `f` непосредственно перед выполнением следующей инструкции:

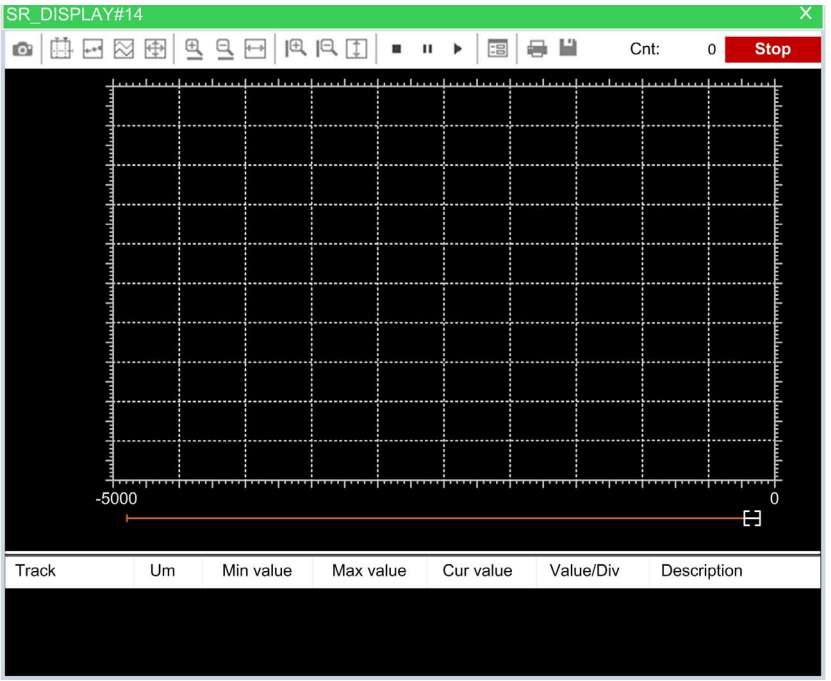
```
f := f+ SHR( d, 16#04 )
```

Для этого наведите курсор на шестую строку.

Для этого выберите пункт **Debug (Отладка) → Add/remove graphic trigger (Добавить/удалить графический триггер)**.

Рядом с номером строки появляется зеленая стрелка и всплывает окно **Graphic trigger (Графический триггер)**:

```
0001
0002  a := b * b;
0003  c := c + SHR( a, 16#04 );
0004
0005  d := e * e;
0006  f := f + SHR( d, 16#04 );
0007
```

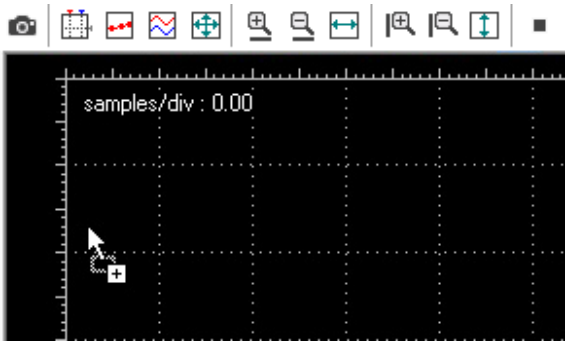


Не все инструкции ST поддерживают триггеры. Например, невозможно поместить триггер на строку, содержащую такой закрывающий элемент, как `END_IF`, `END_FOR`, `END_WHILE` и так далее.

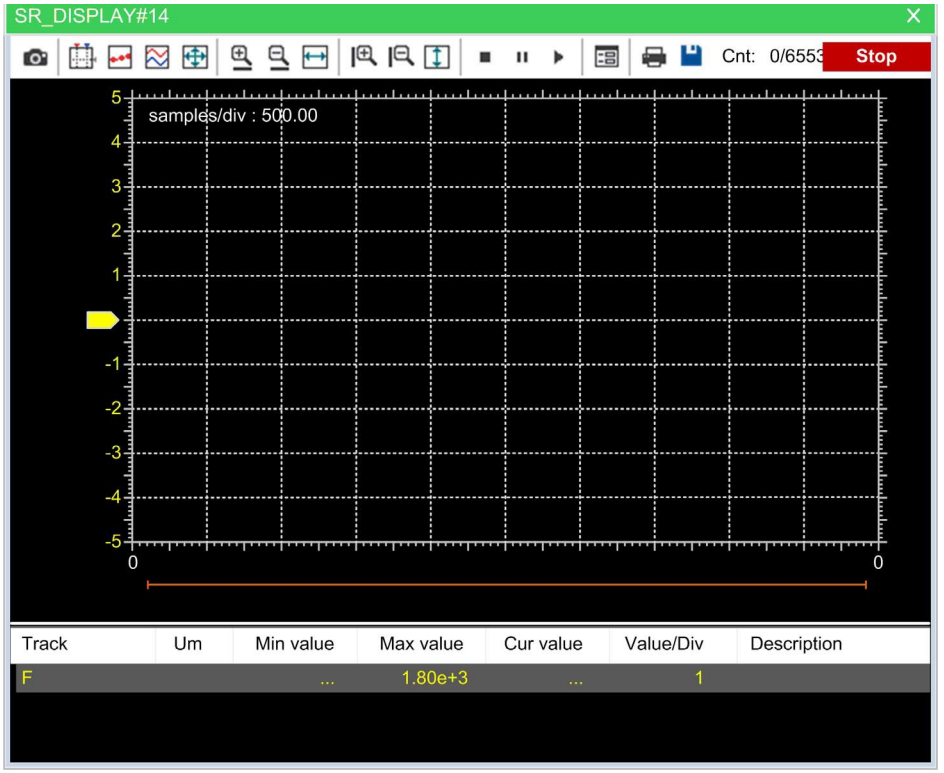
Добавление переменной в окно графического триггера из модуля ST

Чтобы отобразить график переменной, необходимо добавить ее в окно **Graphic trigger (Графический триггер)**.

Выберите переменную двойным щелчком мыши, а затем перетащите ее в окно **Variables (Переменные)**, представляющее собой нижнее белое поле во всплывающем окне:

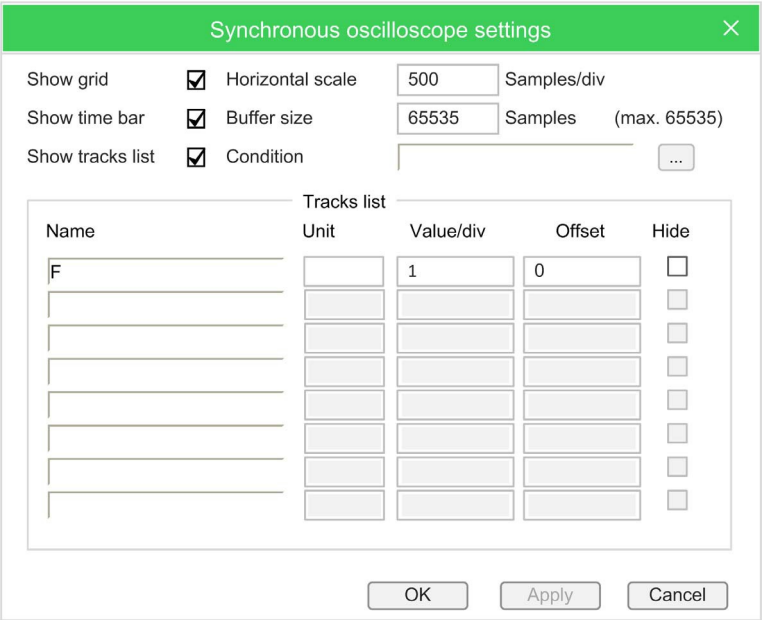


Имя переменной теперь отображается в столбце **Track (Кривая)**:



Та же процедура применяется ко всем переменным, которые необходимо отслеживать.

Как только первая переменная попадает в область графического отслеживания, автоматически открывается окно **Graphic properties (Графические свойства)**, в котором можно настроить параметры выборки и визуализации:



Удаление переменной из окна графического триггера

Если вы хотите удалить переменную из окна графического триггера, выделите ее, щелкнув один раз по ее имени, затем нажмите клавишу **Delete**.

Использование элементов управления

Элементы управления окна графического триггера позволяют контролировать работу этого инструмента отладки и получать больше информации о приложении.

Активация элементов управления

При установке триггера все элементы панели **Controls (Элементы управления)** активируются. Вы можете начать сбор данных, нажав на кнопку **Start graphic trace (Запуск графического отслеживания)**.

Если заданное пользователем условие в данные момент не соблюдается, сбор данных не запускается.

И наоборот, как только условие становится истинным, сбор данных запускается и продолжается до тех пор, пока не будет отпущена кнопка **Start graphic trace (Запуск графического отслеживания)**, даже если условие больше не соблюдается.

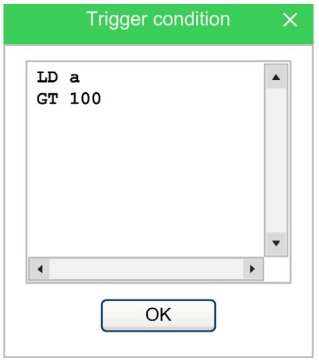
Если отпустить кнопку **Start graphic trace (Запуск графического отслеживания)** до того, как будут получены все необходимые элементы выборки, процесс сбора данных прекратится и все собранные данные будут потеряны.

Определение условия

Этот элемент управления позволяет установить условие для запуска сбора данных. По умолчанию для этого условия установлено значение TRUE, и сбор данных начинается сразу после нажатия кнопки **Enable/Disable acquisition (Включить/отключить сбор данных)**. С этого момента значение переменных в окне **Debug (Отладка)** обновляется каждый раз при срабатывании триггера.

Чтобы задать условие, откройте вкладку **Condition (Условие)** диалогового окна **Options (Параметры)**, затем нажмите соответствующую кнопку: ...

Откроется текстовое окно, в котором можно написать код IL, задающий условие:



По завершении написания кода условия нажмите кнопку **OK**, чтобы установить его, или нажмите клавишу **Esc**, чтобы отменить. Процедура выборки не запустится до тех пор, пока не будет нажата кнопка **Start graphic trace (Запуск графического отслеживания)** и не будет выполнено пользовательское условие. Теперь элемент управления отобразит условие в упрощенном виде:

Чтобы изменить его, нажмите еще раз на кнопку обозревателя: ...

Откроется текстовое окно, содержащее написанный вами текст, который теперь можно редактировать.

Чтобы удалить заданное пользователем условие, еще раз нажмите кнопку, удалите весь код IL в текстовом окне, затем нажмите кнопку **OK**.

Результат кода условия должен быть логическим значением (**TRUE** или **FALSE**), в противном случае произойдет ошибка компилятора.

В коде условия могут использоваться только глобальные и перетаскиваемые переменные. Другими словами, все переменные, являющиеся локальными для модуля, куда изначально был установлен триггер, будут вне области видимости, если вы их не перетащили в окно **Debug (Отладка)**. В окне условий также нельзя объявлять новые переменные.

Настройка масштаба осей

- Ось X

После завершения сбора данных модуль **PROGRAMMING (ПРОГРАММИРОВАНИЕ)** отображает кривую для перетаскиваемых переменных, настраивая ось X таким образом, чтобы все данные помещались в окно **графика**. Если вы хотите применить другой масштаб, откройте вкладку **General (Общее)** в диалоговом окне **Graph properties (Свойства графика)**, введите число в окне редактирования горизонтального масштаба, затем подтвердите выбор, нажав кнопку **Apply (Применить)**.

- Ось Y

Масштаб графика каждой переменной можно изменить с помощью вкладки **Tracks list (Список кривых)** диалогового окна **Graph properties (Свойства графика)**. В противном случае, если вам не нужно указывать точный масштаб, можно воспользоваться элементами управления **Zoom In (Увеличение)** и **Zoom Out (Уменьшение)**.

Закрытие окна графического триггера и удаление триггера

По завершении сеанса отладки с помощью окна графического триггера можно выбрать следующие опции:

- **Закрытие окна графического триггера**
  Если вы закончили построение графика для набора переменных с помощью окна **Graphic trigger (Графический триггер)**, можно при желании закрыть окно **Debug (Отладка)**, не удаляя триггер. Если нажать на кнопку в правом верхнем углу, то окно **интерфейса** будет скрыто, а менеджер окна и соответствующий триггер будут продолжать работать.
  Чтобы восстановить ранее скрытое окно **Graphic trigger (Графический триггер)**:
  - откройте окно **Trigger list (Список триггеров)**;
  - выберите запись (с типом **G**);
  - нажмите кнопку **Open (Открыть)**.

  Окно интерфейса отображается с обновленным счетчиком триггеров, как если бы оно не закрывалось.
- **Удаление триггера**
  При выборе этой опции вы полностью удаляете код менеджера окна и его триггера:
  - Откройте окно **Trigger list (Список триггеров)**.
  - Выберите запись (с типом G).
  - Нажмите кнопку **Remove (Удалить)**.

  Кроме того, вы можете переместить курсор на строку (если модуль находится в режиме IL) или щелкнуть блок (если модуль находится в режиме FBD), куда вы поместили триггер. Теперь нажмите кнопку **Graphic trace (Графическое отслеживание)** на панели инструментов **Debug (Отладка)**.

- **Удаление всех триггеров**

  Также вы можете удалить все имеющиеся триггеры за один раз, независимо от того, какие записи выбраны. Для этого нажмите кнопку **Remove all triggers (Удалить все триггеры)**.

Глава 14
Справочник по языку

Описание

Языки EcoStruxure Machine Expert - HVAC соответствуют стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3):

- Общие элементы
- Список инструкций (IL)
- Функциональная блок-схема (FBD)
- Релейная схема (LD)
- «Структурный текст» (ST)
- Последовательная функциональная схема (SFC)

Кроме того, в EcoStruxure Machine Expert - HVAC реализованы некоторые расширения:

- Указатели
- Макросы

Содержание настоящей главы

В этой главе обсуждаются следующие темы.

Раздел	Тема	Страница
14.1	Общие элементы	372
14.2	Список инструкций (IL)	404
14.3	Функциональная блок-схема (FBD)	408
14.4	Релейная схема (LD)	413
14.5	«Структурный текст» (ST)	418
14.6	Последовательная функциональная схема (SFC)	429
14.7	Языковые расширения EcoStruxure Machine Expert - HVAC	441

Раздел 14.1
Общие элементы

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	372
Базовые элементы	372
Элементарные типы данных	373
Производные типы данных	373
Литералы	376
Переменные	378
Программные модули	382
Операторы и стандартные блоки	385

Обзор

Описание

Общими являются текстовые и графические элементы, используемые во всех языках программирования программируемых контроллеров, указанных в стандарте МЭК 61131-3 (ГОСТ Р МЭК 61131-3).

ПРИМЕЧАНИЕ. Определение и редактирование большинства общих элементов (переменных, структурированных элементов, определений функциональных блоков и т. д.) производится в EcoStruxure Machine Expert - HVAC с помощью специальных редакторов, форм и таблиц. EcoStruxure Machine Expert - HVAC не позволяет напрямую редактировать исходный код, связанный с этими общими элементами.

ПРИМЕЧАНИЕ. Представленная далее информация получена непосредственно из стандартов МЭК, охраняемых авторским правом.

Базовые элементы

Кодировка

В текстовых документах и текстовых элементах графических языков используется стандартный набор символов ASCII.

Комментарии

Пользовательские комментарии отмечаются в начале и в конце специальными комбинациями символов: `(*` и `*)`. Комментарии можно оставлять в любом месте программы, и они не имеют синтаксического или семантического значения ни в одном из языков, определенных в этом стандарте.

Использование вложенных комментариев, например `(* (* ВЛОЖЕННЫЙ КОММЕНТАРИЙ *) *)`, приводит к ошибке.

Элементарные типы данных

Описание

EcoStruxure Machine Expert - HVAC предоставляет ряд элементарных (предопределенных) типов данных.

Элементарные типы данных, ключевое слово для каждого типа данных, количество бит на элемент данных и диапазон значений для каждого элементарного типа данных представлены в таблице ниже.

Ключевое слово	Тип данных	Кол-во бит	Диапазон
BOOL	Логическое значение	(1)	0–1
SINT	Короткое целое число	8	–128...127
USINT	Короткое целое число без знака	8	0–255
INT	Целое число	16	–32 768...32 767
UINT	Целое число без знака	16	0–65 536
DINT	Целое число двойной длины	32	$-2^{31}\ldots2^{31}-1$
UDINT	Длинное целое число без знака	32	$0\ldots2^{32}$
BYTE	Строка из 8 бит	8	–
WORD	Строка из 16 бит	16	–
DWORD	Строка из 32 бит	32	–
REAL	Вещественное число	32	–3,40E+38… +3,40E+38
STRING	Строка символов	–	–
(1) Реализация типа данных BOOL зависит от процессора целевого устройства — например, для устройств с побитовой адресацией размер составляет 1 бит.			

Производные типы данных

Описание

Производные типы данных объявляются с помощью конструкции `TYPE… END_TYPE`. Они могут использоваться в объявлениях переменных в дополнение к элементарным типам данных.

Как одноэлементные переменные, так и элементы многоэлементной переменной, объявленные в качестве производных типов данных, могут использоваться везде, где допускается применение переменной родительского типа.

Определение типа

Определения типа позволяют присвоить альтернативные имена уже существующим типам. Нет никаких различий между определением типа и его родительским типом, кроме имени.

Определения типов объявляются с помощью следующего синтаксиса:

```
TYPE
   <имя перечисляемого типа данных> : <имя родительского типа>;
END_TYPE
```

Например, рассмотрим следующее объявление, где имя `LONGWORD` сопоставляется с определенным в стандарте МЭК 61131-3 (ГОСТ Р МЭК 61131-3) типом `DWORD`:

```
TYPE
   LONGWORD : DWORD;
END_TYPE
```

Перечисляемые типы данных

Объявление перечисляемого типа данных указывает, что значение любого элемента данных этого типа может быть только одним из значений, приведенных в соответствующем списке идентификаторов. Список перечисления представляет собой упорядоченный набор перечисляемых значений — начиная с первого идентификатора списка и заканчивая последним.

Перечисляемые типы данных объявляются с помощью следующего синтаксиса:

```
TYPE
   <имя перечисляемого типа данных> : ( <список перечисления> );
END_TYPE
```

Например, рассмотрим следующее объявление двух перечисляемых типов данных. При отсутствии явного значения у идентификатора в списке перечисления его значение равно значению, присвоенному предыдущему идентификатору, дополненному единицей.

```
TYPE
   enum1: (
      val1, (* значение val1 равно 0 *)
      val2, (* значение val2 равно 1 *)
      val3 (* значение val3 равно 2 *)
   );
   enum2: (
      k := -11,
      i := 0,
      j, (* значение j равно ( i + 1 ) = 1 *)
      l := 5
   );
END_TYPE
```

Различные перечисляемые типы данных могут использовать одни и те же идентификаторы для перечисляемых значений. Для однозначной идентификации в определенном контексте перечисляемые литералы можно дополнить префиксом, состоящим из имени связанного с ними типа данных и знака `#`.

Поддиапазоны

При объявлении поддиапазона указывается, что значение любого элемента данных этого типа ограничено верхним и нижним предельными значениями и включает их.

Поддиапазоны объявляются с помощью следующего синтаксиса:

```
TYPE
   <имя поддиапазона> : <имя родительского типа>
   ( <нижний предел>..<верхний предел> );
END_TYPE
```

Например, рассмотрим следующее объявление:

```
TYPE
   int_0_to_100 : INT (0..100);
END_TYPE
```

Структуры

При объявлении `STRUCT` указывается, что элементы данных этого типа должны содержать подэлементы указанных типов, доступ к которым возможен по указанным именам.

Структуры объявляются с помощью следующего синтаксиса:

```
TYPE
   <имя структурного типа> : STRUCT
      <Объявление элементов структуры>
   END_STRUCT;
END_TYPE
```

Например, рассмотрим следующее объявление:

```
TYPE
   structure1 : STRUCT
      elem1 : USINT;
      elem2 : USINT;
      elem3 : INT;
      elem3 : REAL;
   END_STRUCT;
END_TYPE
```

Литералы

Числовые литералы

Внешнее представление данных на различных языках программирования контроллеров состоит из числовых литералов.

Существует два класса числовых литералов: целочисленные и вещественные. Числовой литерал определяется как десятичное число или число с основанием.

Десятичные литералы представлены в общепринятой десятичной системе счисления. Вещественные литералы отличаются наличием десятичного разделителя. Экспонента означает целочисленную степень для числа десять, на которое следует умножить предшествующее число, чтобы получить представленное значение. Десятичные литералы и их экспоненты могут содержать знак (+ или –).

Целочисленные литералы также могут быть представлены с основанием 2, 8 или 16. По умолчанию используется десятичная система счисления. Для чисел с основанием 16 используется расширенный набор цифр, дополненный буквами от A до F, обозначающими цифры от 10 до 15. Числа, не основанные на десятичной системе, не содержат знака (+ или –).

Логические данные представлены ключевыми словами `FALSE` и `TRUE`.

Типы числовых литералов и примеры представлены в таблице ниже.

Описание типа	Примеры
Целочисленные литералы	–12 0 123 +986
Вещественные литералы	–12,0: 0.0 0,4560
Вещественные литералы с экспонентами	–1,34E–12 или –1,34e–12 1,0E+6 или 1,0e+6 1,234E6 или 1,234e6
Литералы с основанием 2	2#11111111 (256 десятичн.) 2#11100000 (240 десятичн.)
Литералы с основанием 8	8#377 (256 десятичн.) 8#340 (240 десятичн.)
Литералы с основанием 16	16#FF или 16#ff (256 десятичн.) 16#E0 или 16#e0 (240 десятичн.)
Логические литералы `FALSE` и `TRUE`	`FALSE` `TRUE`
Для получения дополнительной информации см. раздел «Двухсимвольные комбинации» *(стр. 377)*.	

Символьные литералы

Символьный литерал — это последовательность из нуля и более символов, заключенная в одинарные кавычки (').

Пример	Описание
`''`	Пустая последовательность (нулевая длина)
`'A'`	Последовательность длиной в один символ, содержащая один символ A
`' '`	Последовательность длиной в один символ, содержащая один символ **пробела**
`'$''`	Последовательность длиной в один символ, содержащая символ **одинарной кавычки**(1)
`'"'`	Последовательность длиной в один символ, содержащая символ **двойной кавычки**
`'$R$L'`	Последовательность длиной в два символа, содержащая символы `CR` и `LF`(1)
`'$0A'`	Последовательность длиной в один символ, содержащая символ `LF`(2)

(1) Для получения дополнительной информации см. раздел «Двухсимвольные комбинации» *(стр. 377)*.

(2) Трехсимвольная комбинация, состоящая из знака доллара ($) и двух шестнадцатеричных цифр, интерпретируется как шестнадцатеричное представление восьмиразрядного кода символов ASCII.

Двухсимвольные комбинации

В последовательности символов могут быть двухсимвольные комбинации, начинающиеся со знака доллара, которые интерпретируются в соответствии с приведенной ниже таблицей.

Комбинация	Интерпретация при отображении
`$$`	Знак доллара
`$'`	Одинарная кавычка
`$L` или `$l`	Символ перевода строки
`$N` или `$n`	Новая строка
`$P` или `$p`	Перевод формата (страницы)
`$R` или `$r`	Возврат каретки
`$T` или `$t`	Табуляция

Переменные

Предисловие

Переменные предназначены для идентификации объектов данных, содержимое которых может изменяться, например данных, связанных с входами, выходами или памятью программируемого контроллера. Переменная должна объявляться с указанием одного из элементарных типов. Переменные могут быть представлены символически или таким образом, чтобы непосредственно ассоциировать элемент данных с физическими или логическими точками на входе, выходе или в структуре памяти программируемого контроллера.

Каждый программный модуль (программа, функция или функциональный блок) содержит в начале минимум одну область объявления. Эта область объявления состоит из одного или нескольких структурных элементов, которые определяют типы (и, при необходимости, физическое или логическое расположение) переменных, используемых в модуле. Эта область объявления имеет текстовую форму и содержит одно из ключевых слов `VAR`, `VAR_INPUT` или `VAR_OUTPUT` (определены в разделе ключевых слов), за которым в случае `VAR` опционально следует один квалификатор `RETAIN`, `NON_RETAIN` или `CONSTANT`, а в случае `VAR_INPUT` или `VAR_OUTPUT` — опционально один квалификатор `RETAIN` или `NON_RETAIN`, за которым следует одно или несколько объявлений, разделенных точкой с запятой и завершаемых ключевым словом `END_VAR`. В объявлении можно также инициализировать объявленную переменную, если программируемый контроллер поддерживает объявление пользователем начальных значений для переменных.

Структурный элемент

Объявление переменной должно выполняться внутри следующего структурного элемента программы:

```
КЛЮЧСЛОВО [RETAIN] [CONSTANT]
   <Объявление 1>
   <Объявление 2>
   ...
   <Объявление N>
END_VAR
```

Ключевые слова и область видимости

Ключевое слово	Использование переменной
`VAR`	Внутри программного модуля.
`VAR_INPUT`	Предоставляется извне.
`VAR_OUTPUT`	Предоставляется программным модулем внешним сущностям.
`VAR_IN_OUT`	Предоставляется внешними сущностями, может изменяться внутри программного модуля.
`VAR_EXTERNAL`	Предоставляется конфигурацией через `VAR_GLOBAL`, может изменяться внутри программного модуля.
`VAR_GLOBAL`	Объявление глобальной переменной.

Область видимости (действия) объявлений, содержащихся в структурных элементах, ограничена программным модулем, в котором содержится эта область объявления. То есть объявленные переменные доступны другим программным модулям, за исключением явной передачи аргумента через переменные, которые были объявлены как входы или выходы этих модулей. Единственным исключением из этого правила являются переменные, объявленные глобально.

Глобальные переменные доступны программам в любом случае, либо посредством объявления VAR_EXTERNAL для функциональных блоков. Тип переменной, объявленной в блоке VAR_EXTERNAL, должен совпадать с типом, объявленным в блоке VAR_GLOBAL.

Чтобы предоставить доступ к этим переменным всем типам программных модулей без использования каких-либо ключевых слов, необходимо включить эту опцию на вкладке Code generation (Генерирование кода) окна Project Options (Параметры проекта) *(см. стр. 166)*.

Выводится ошибка:

- если какой-либо программный модуль попытается изменить значение переменной, которая была объявлена с помощью квалификатора CONSTANT;
- если переменная, объявленная как VAR_GLOBAL CONSTANT в элементе конфигурации или программном модуле («элемент-контейнер»), используется в объявлении VAR_EXTERNAL (без квалификатора CONSTANT) любого элемента, содержащегося в элементе-контейнере.

Квалификаторы

Квалификатор	Описание
CONST	Атрибут CONST указывает на то, что переменные внутри структурных элементов являются константами, т. е. они имеют постоянное значение, которое не может быть изменено после компиляции проекта ПЛК.
RETAIN	Атрибут RETAIN указывает на то, что переменные внутри структурных элементов сохраняются, т. е. они сохраняют свое значение даже после сброса или выключения целевого устройства.

Одноэлементные переменные и массивы

Одноэлементная переменная представляет собой один элемент данных одного из элементарных типов либо одного из производных типов.

Массив — это совокупность элементов данных одного типа; для доступа к одному элементу массива необходимо использовать индекс (идентификатор), заключаемый в квадратные скобки. Индексами могут быть как целочисленные литералы, так и одноэлементные переменные.

Для представления матриц данных массивы могут быть многомерными; в этом случае используется составной индекс, где каждый отдельный индекс соответствует одному измерению и отделяется от другого запятой. Максимальное количество измерений, допустимых при определении массива, равно трем.

Синтаксис объявления

Переменные должны объявляться внутри структурных элементов с использованием следующего синтаксиса:

```
VarName1 : ИмяТипа1 [ := НачальноеЗнач1 ];
ИмяПеременной2 AT Местоположение2 : ИмяТипа2 [ := НачальноеЗнач2 ];
ИмяПеременной3 : ARRAY [ 0..N ] OF ИмяТипа3;
```

Ключевое слово	Описание
ИмяПеременнойX	Идентификатор переменной, состоящий из последовательности буквенно-числовых символов (1 или более). Используется для символьного представления переменных.
ИмяТипаX	Тип данных переменной, выбранный из элементарных типов данных.
НачальноеЗначX	Значение, которое переменная принимает после сброса целевого устройства.
МестоположениеX	См. описание в разделе «Местоположение» *(стр. 380)*.
N	Индекс последнего элемента массива, имеющего длину N + 1.

Местоположение

Переменные могут быть представлены символически, т. е. доступны по идентификатору или таким образом, чтобы непосредственно ассоциировать элемент данных с физическими или логическими точками на входе, выходе или в структуре памяти программируемого контроллера.

Прямое представление одноэлементной переменной осуществляется с помощью специального символа, образованного путем соединения знака процента %, префикса местоположения и префикса размера, а также одного или двух беззнаковых целых чисел, разделенных точками (.).

```
%местоположение.размер.индекс.индекс
```

1) Местоположение

Для местоположения может использоваться один из следующих префиксов.

Префикс местоположения	Описание
I	Местоположение на входе
Q	Местоположение на выходе
M	Местоположение в памяти

2) Размер

Для размера может использоваться один из следующих префиксов.

Префикс размера	Описание
X	Один бит
B	Байт (8 бит)

Префикс размера	Описание
W	Слово (16 бит)
D	Двойное слово (32 бит)

3) индекс.индекс

Эта последовательность беззнаковых целых чисел, разделенных точками, задает положение переменной в области, заданной префиксом местоположения.

Пример.

Прямое представление	Описание
%MW4.6	Слово, начинающееся с первого байта седьмого элемента четвертого блока данных памяти.
%IX0.4	Первый бит первого байта пятого элемента входа установлен на 0.

Абсолютная позиция зависит от размера элементов блока данных, а не от префикса размера. %MW4.6 и %MD4.6 начинаются с одного и того же байта в памяти, но первый указывает на область, которая на 16 бит короче, чем у второго.

Только для опытных пользователей: если индекс состоит только из одного целого числа (без точек), то он теряет ссылку на блоки данных и указывает непосредственно на байт в памяти, имеющий в качестве абсолютного адреса значение индекса.

Прямое представление	Описание
%MW4.6	Слово, начинающееся с первого байта седьмого элемента четвертого блока данных памяти.
%MW4	Слово, начинающееся с четвертого байта памяти.

Пример:

```
VAR [RETAIN] [CONSTANT]
   XQuote : DINT;
   Enabling : BOOL := FALSE;
   TorqueCurrent AT %MW4.32 : INT;
   Counters : ARRAY [ 0 .. 9 ] OF UINT;
   Limits: ARRAY [0..3, 0..9]
END_VAR
```

- Переменная XQuote имеет длину 32 бита, и место под нее в памяти автоматически выделяется компилятором EcoStruxure Machine Expert - HVAC.
- Переменная Enabling инициализируется со значением FALSE после сброса целевого устройства.
- Место под переменную TorqueCurrent выделяется в области памяти целевого устройства и занимает 16 бит, начиная с первого байта 33-го элемента четвертого блока данных.
- Переменная Counters — это массив из десяти независимых переменных беззнаковых целых чисел.

Объявление переменных в EcoStruxure Machine Expert - HVAC

Среда EcoStruxure Machine Expert - HVAC позволяет вам не зависеть от используемого языка ПЛК и не учитывать предыдущий синтаксис, так как она предоставляет редактор локальных переменных, редактор глобальных переменных и редактор параметров, которые предоставляют интерфейсы для объявления всех типов переменных.

Программные модули

Описание

Программные модули — это функции, функциональные блоки и программы. Программные модули могут поставляться производителем или программироваться вами с помощью средств, определенных в этой части стандарта.

Программные модули не являются рекурсивными, т. е. вызов программного модуля не может приводить к вызову того же программного модуля.

Функции

Введение

В языках программирования контроллеров функция определяется как программный модуль, который при выполнении производит ровно один элемент данных, принимаемый за результат выполнения функции.

Функции не содержат информацию о внутреннем состоянии, то есть результатом вызова функции с одними и теми же аргументами (входные переменные `VAR_INPUT` и входные/выходные переменные `VAR_IN_OUT`) всегда являются одни и те же значения (выходные переменные `VAR_OUTPUT`, входные/выходные переменные `VAR_IN_OUT` и результат работы функции).

Синтаксис объявления

Объявление функции должно выполняться следующим образом:

```
FUNCTION ИмяФункции : ТипПолучДанных
   VAR_INPUT
      <Объявление входных переменных (см. соответствующий раздел)>
   END_VAR
   VAR
      <Объявление локальных переменных (см. соответствующий раздел)>
   END_VAR
   <Тело функции>
END_FUNCTION
```

Ключевое слово	Описание
`ИмяФункции`	Имя объявляемой функции.
`ТипПолучДанных`	Тип данных возвращаемого функцией значения.
`<Тело функции>`	Определяет операции, которые необходимо произвести над входными переменными. Конечный результат, зависящий от семантики функции, присваивается переменной с тем же именем, что и у функции. Содержимое может быть написано на любом из языков, поддерживаемых EcoStruxure Machine Expert - HVAC.

Объявление функций

Среда EcoStruxure Machine Expert - HVAC позволяет вам не зависеть от используемого языка ПЛК и не учитывать предыдущий синтаксис, так как она предоставляет удобный интерфейс для использования функций.

Функциональные блоки

Введение

Для целей языков программирования контроллеров функциональный блок определяется как программный модуль, который при выполнении производит одно или несколько значений. Можно создать несколько именованных экземпляров (копий) функционального блока. Каждый экземпляр имеет связанный с ним идентификатор (имя экземпляра), а также структуру данных, содержащую входные, выходные и внутренние переменные. Все значения выходных переменных и необходимых внутренних переменных этой структуры данных сохраняются при переходе от одного функционального блока к другому. Результатом вызова функционального блока с одними и теми же аргументами (входными переменными) не всегда являются одни и те же выходные значения.

Только входные и выходные переменные доступны снаружи экземпляра функционального блока, т. е. внутренние переменные функционального блока скрыты от пользователя функционального блока.

Для выполнения своих операций функциональный блок должен быть запущен другим программным модулем. Вызов зависит от языка модуля, вызывающего функциональный блок.

Область видимости экземпляра функционального блока ограничена программным модулем, в котором он создается.

Синтаксис объявления

Объявление функции должно выполняться следующим образом:

```
FUNCTION_BLOCK ИмяФункцБлока
   VAR_INPUT
      <Объявление входных переменных (см. соответствующий раздел)>
   END_VAR
   VAR_OUTPUT
      <Объявление выходных переменных>
   END_VAR
   VAR_EXTERNAL
      <Объявление внешних переменных>
   END_VAR
   VAR
      <Объявление локальных переменных>
   END_VAR
   <Тело функционального блока>
END_FUNCTION_BLOCK
```

Ключевое слово	Описание
`ИмяФункцБлока`	Имя объявляемого функционального блока (примечание: имя шаблона, а не его экземпляров).
`VAR_EXTERNAL .. END_VAR`	Функциональный блок может обращаться к глобальным переменным только в том случае, если они перечислены в структурном элементе `VAR_EXTERNAL`. Переменные, передаваемые в функциональный блок через конструкцию `VAR_EXTERNAL`, могут быть изменены изнутри функционального блока.
`<Тело функционального блока>`	Определяет операции, которые необходимо выполнить над входными переменными для присвоения значений выходным переменным — в зависимости от семантики функционального блока и от значения внутренних переменных. Содержимое может быть написано на любом из языков, поддерживаемых EcoStruxure Machine Expert — HVAC.

Объявление функций

Среда EcoStruxure Machine Expert - HVAC позволяет вам не зависеть от используемого языка ПЛК и не учитывать предыдущий синтаксис, так как она предоставляет удобный интерфейс для использования функциональных блоков.

Программы

Введение

В стандарте МЭК 61131-1 (ГОСТ Р МЭК 61131-1) программой называется «логическая совокупность всех элементов языка программирования и конструкций, предназначенных для обработки сигналов, необходимых для управления машиной или процессом с помощью ПЛК-системы».

Синтаксис объявления

Объявление программы должно выполняться следующим образом:

```
PROGRAM <Имя программы>
   <Объявление переменных (см. соответствующий раздел)>
   <Тело программы>
END_PROGRAM
```

Ключевое слово	Описание
`<Имя программы>`	Имя объявляемой программы.
`<Тело программы>`	Определяет операции, которые необходимо выполнить для получения необходимого результата обработки сигнала. Может быть описано на любом из языков, поддерживаемых EcoStruxure Machine Expert - HVAC.

Написание программ

Среда EcoStruxure Machine Expert - HVAC позволяет вам не зависеть от используемого языка ПЛК и не учитывать предыдущий синтаксис, так как она предоставляет удобный интерфейс для написания программ.

Операторы и стандартные блоки

Описание

Доступность следующих функций зависит от целевого устройства.

Эти функции являются общими для всего набора языков программирования и могут использоваться в любом программном модуле. Доступ к ним осуществляется из окна **Operators and blocks (Операторы и блоки)** на вкладке **Operator and standard blocks (Оператор и стандартные блоки)**.

Операторы и стандартные блоки разбиты на группы:

- Арифметические функции и операторы *(см. стр. 386)*
- Бистабильные операторы *(см. стр. 391)*
- Функции битового сдвига *(см. стр. 392)*
- Операторы сравнения *(см. стр. 393)*
- Функции приведения типов *(см. стр. 394)*
- Логические функции *(см. стр. 398)*
- Функции выбора *(см. стр. 399)*
- Строковые функции *(см. стр. 401)*

Арифметические функции и операторы

ABS	
Описание	Абсолютное значение. Вычисляет абсолютное значение входа № 0.
Количество операндов	1
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе

Примеры	`OUT := ABS(-5 );(* OUT = 5 *)`
	`OUT := ABS(-1.618 );(* OUT = 1.618 *)`
	`OUT := ABS(3.141592 );(* OUT = 3.141592 *)`

ACOS	
Описание	Арккосинус. Вычисляет главное значение арккосинуса входа № 0; результат выражается в радианах.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := ACOS( 1.0 ); (* OUT = 0.0 *)` `OUT := ACOS( -1.0 ); (* OUT = PI *)`

ADD	
Описание	Арифметическое сложение. Вычисляет сумму двух входов.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе
Примеры	`OUT := ADD( 20, 40 ); (* OUT = 60 *)`

ASIN	
Описание	Арксинус. Вычисляет главное значение арксинуса входа № 0; результат выражается в радианах.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := ASIN( 0.0 ); (* OUT = 0.0 *)` `OUT := ASIN( 1.0 ); (* OUT = PI / 2 *)`

ATAN	
Описание	Арктангенс. Вычисляет главное значение арктангенса входа № 0; результат выражается в радианах.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := ATAN( 0.0 ); (* OUT = 0.0 *)` `OUT := ATAN( 1.0 ); (* OUT = PI / 4 *)`

ATAN2*	
Описание	Арктангенс (с двумя параметрами). Вычисляет главное значение арктангенса для Y/X; результат выражается в радианах.
Количество операндов	2
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL

Примеры	`OUT := ATAN2( 0.0, 1.0 ); (* OUT = 0.0 *)`
	`OUT := ATAN2( 1.0, 1.0 ); (* OUT = PI / 4 *)`
	`OUT := ATAN2( -1.0, -1.0 ); (* OUT = ( -3/4 ) * PI *)`
	`OUT := ATAN2( 1.0, 0.0 ); (* OUT = PI / 2 *)`

CEIL*	
Описание	Округление до целого числа в большую сторону. Возвращает наименьшее целое число, которое больше или равно входу № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := CEIL( 1.95 ); (* OUT = 2.0 *)` `OUT := CEIL( -1.27 ); (* OUT = -1.0 *)`

COS	
Описание	Косинус. Вычисляет функцию косинуса входа № 0, выражается в радианах.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := COS( 0.0 ); (* OUT = 1.0 *)` `OUT := COS( -3.141592 ); (* OUT ~ -1.0 *)`

COSH*	
Описание	Гиперболический косинус. Вычисляет функцию гиперболического косинуса для входа № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := COSH( 0.0 ); (* OUT = 1.0 *)`

DIV	
Описание	Арифметическое деление. Делит значение на входе № 0 на значение на входе № 1.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе
Примеры	`OUT := DIV( 20, 2 ); (* OUT = 10 *)`

EXP	
Описание	Экспоненциальная функция. Вычисляет экспоненциальную функцию для входа № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := EXP( 1.0 ); (* OUT ~ 2.718281 *)`

FLOOR*	
Описание	Округление до целого числа в меньшую сторону. Возвращает наибольшее целое число, которое меньше или равно входу № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := FLOOR( 1.95 ); (* OUT = 1.0 *)` `OUT := FLOOR( -1.27 ); (* OUT = -2.0 *)`

LN	
Описание	Натуральный логарифм. Вычисляет логарифм с основанием e для входа № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL

LN	
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := LN( 2.718281 );(* OUT = 1.0 *)`

LOG	
Описание	Десятичный логарифм. Вычисляет логарифм с основанием 10 для входа № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := LOG( 100.0 );(* OUT = 2.0 *)`

MOD	
Описание	Деление по модулю. Вычисляет остаток от деления значения на входе № 0 на значение на входе № 1.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе
Примеры	`OUT := MOD( 10, 3 ); (* OUT = 1 *)`

MUL	
Описание	Арифметическое умножение. Умножает значения двух входов.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе
Примеры	`OUT := MUL( 10, 10 ); (* OUT = 100 *)`

POW	
Описание	Возведение в степень. Возводит основание в степень.
Количество операндов	2
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := POW( 2.0, 3.0 ); (* OUT = 8.0 *)` `OUT := POW( -1.0, 5.0 ); (* OUT = -1.0 *)`

SIN	
Описание	Синус. Вычисляет функцию синуса входа № 0, выражается в радианах.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := SIN( 0.0 ); (* OUT = 0.0 *)` `OUT := SIN( 2.5 * 3.141592 ); (* OUT ~ 1.0 *)`

SINH*	
Описание	Гиперболический синус. Вычисляет функцию гиперболического синуса для входа № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := SINH( 0.0 ); (* OUT = 0.0 *)`

SQRT	
Описание	Квадратный корень. Вычисляет квадратный корень значения входа № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := SQRT( 4.0 ); (* OUT = 2.0 *)`

SUB	
Описание	Арифметическое вычитание. Вычитает значение входа № 1 из значения входа № 0.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе
Примеры	`OUT := SUB( 10, 3 ); (* OUT = 7 *)`

TAN	
Описание	Тангенс. Вычисляет функцию тангенса для входа № 0, выражается в радианах.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL

TAN	
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := TAN( 0.0 ); (* OUT = 0.0 *)` `OUT := TAN( 3.141592 / 4.0 ); (* OUT ~ 1.0 *)`

TANH*	
Описание	Гиперболический тангенс. Вычисляет функцию гиперболического тангенса для входа № 0.
Количество операндов	1
Тип данных на входе	LREAL, если доступно, в противном случае REAL
Тип данных на выходе	LREAL, если доступно, в противном случае REAL
Примеры	`OUT := TANH( 0.0 ); (* OUT = 0.0 *)`

* Функция предоставляется в качестве расширения стандарта МЭК 61131-3 (ГОСТ Р МЭК 61131-3).

Бистабильные операторы

R	
Описание	Сброс логического значения.
Количество операндов	1
Тип данных на входе	BOOL
Тип данных на выходе	BOOL

Примеры	`LD x`
	`R y`
	`ST z`

S	
Описание	Установка логического значения.
Количество операндов	1
Тип данных на входе	BOOL
Тип данных на выходе	BOOL

Примеры	`LD x`
	`S y`
	`ST z`

Функции битового сдвига

ROL	
Описание	Значение входа № 0 сдвигается влево на число бит, указанное для входа № 1, циклически.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе № 0
Примеры	`OUT := ROL( IN := 16#1000CAFE, 4 ); (* OUT = 16#000CAFE1 *)`

ROR	
Описание	Значение входа № 0 сдвигается вправо на число бит, указанное для входа № 1, циклически.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе № 0
Примеры	`OUT := ROR( IN := 16#1000CAFE, 16 ); (* OUT = 16#CAFE1000 *)`

SHL	
Описание	Значение входа № 0 сдвигается влево на количество бит, указанное для входа № 1, освободившиеся биты справа заполняются нулями.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе № 0
Примеры	`OUT := SHL( IN := 16#1000CAFE, 16 ); (* OUT = 16#CAFE0000 *)`

SHR	
Описание	Значение входа № 0 сдвигается вправо на количество бит, указанное для входа № 1, освободившиеся биты слева заполняются нулями.
Количество операндов	2
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе № 0
Примеры	`OUT := SHR( IN := 16#1000CAFE, 24 ); (* OUT = 16#00000010 *)`

Операторы сравнения

Операторы сравнения могут также использоваться для сравнения строк, если эта функция поддерживается целевым устройством.

EQ	
Описание	Равно. Возвращает TRUE, если вход № 0 = вход № 1, в противном случае FALSE.
Количество операндов	2
Тип данных на входе	Любой
Тип данных на выходе	BOOL
Примеры	`OUT := EQ( TRUE, FALSE ); (* OUT = FALSE *)` `OUT := EQ( 'AZ', 'ABC' ); (* OUT = FALSE *)`

GE	
Описание	Больше или равно. Возвращает TRUE, если вход № 0 >= вход № 1, в противном случае FALSE.
Количество операндов	2
Тип данных на входе	Любой, кроме BOOL
Тип данных на выходе	BOOL
Примеры	`OUT := GE( 20, 20 ); (* OUT = TRUE *)` `OUT := GE( 'AZ', 'ABC' ); (* OUT = FALSE *)`

GT	
Описание	Больше. Возвращает TRUE, если вход № 0 > вход № 1, в противном случае FALSE.
Количество операндов	2
Тип данных на входе	Любой, кроме BOOL
Тип данных на выходе	BOOL
Примеры	`OUT := GT( 0, 20 ); (* OUT = FALSE *)` `OUT := GT( 'AZ', 'ABC' ); (* OUT = TRUE *)`

LE	
Описание	Меньше или равно. Возвращает TRUE, если вход № 0 <= вход № 1, в противном случае FALSE.
Количество операндов	2
Тип данных на входе	Любой, кроме BOOL
Тип данных на выходе	BOOL
Примеры	`OUT := LE( 20, 20 ); (* OUT = TRUE *)` `OUT := LE( 'AZ', 'ABC' ); (* OUT = FALSE *)`

LT	
Описание	Меньше. Возвращает TRUE, если вход № 0 < вход № 1, в противном случае FALSE.
Количество операндов	2
Тип данных на входе	Любой, кроме BOOL
Тип данных на выходе	BOOL
Примеры	`OUT := LT( 0, 20 ); (* OUT = TRUE *)` `OUT := LT( 'AZ', 'ABC' ); (* OUT = FALSE *)`

NE	
Описание	Не равно. Возвращает TRUE, если вход № 0 != вход № 1, в противном случае FALSE.
Количество операндов	2
Тип данных на входе	Любой
Тип данных на выходе	BOOL
Примеры	`OUT := NE( TRUE, FALSE ); (* OUT = TRUE *)` `OUT := NE( 'AZ', 'ABC' ); (* OUT = TRUE *)`

Функции приведения типов

Согласно стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3), функции приведения типов должны иметь форму `*_TO_**`, где `*` — тип входной переменной, а `**` — тип выходной переменной (например, `INT_TO_REAL`). EcoStruxure Machine Expert - HVAC предоставляет более удобный набор перегруженных функций приведения типов, освобождая вас от необходимости указывать тип входной переменной.

TO_BOOL	
Описание	Приведение к типу BOOL (логический).
Количество операндов	1
Тип данных на входе	Любой числовой тип
Тип данных на выходе	BOOL

Примеры	`out := TO_BOOL( 0 ); (* out = FALSE *)`
	`out := TO_BOOL( 1 ); (* out = TRUE *)`
	`out := TO_BOOL( 1000 ); (* out = TRUE *)`

TO_BYTE	
Описание	Приведение к типу BYTE (строка 8 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	BYTE
Примеры	`out := TO_BYTE( -1 ); (* out = 16#FF *)` `out := TO_BYTE( 16#100 ); (* out = 16#00 *)`

TO_DINT	
Описание	Приведение к типу DINT (целое число со знаком 32 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	DINT
Примеры	`out := TO_DINT( 10.0 ); (* out = 10 *)` `out := TO_DINT( 16#FFFFFFFF ); (* out = -1 *)`

TO_DWORD	
Описание	Приведение к типу DWORD (строка 32 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	DWORD
Примеры	`out := TO_DWORD( 10.0 ); (* out = 16#0000000A *)` `out := TO_DWORD( -1 ); (* out = 16#FFFFFFFF *)`

TO_INT	
Описание	Приведение к типу INT (целое число со знаком 16 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	INT
Примеры	`out := TO_INT( -1000.0 ); (* out = -1000 *)` `out := TO_INT( 16#8000 ); (* out = -32768 *)`

TO_LREAL	
Описание	Приведение к типу LREAL (число с плавающей точкой 64 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	LREAL
Примеры	`out := TO_LREAL( -1000 ); (* out = -1000.0 *)` `out := TO_LREAL( 16#8000 ); (* out = -32768.0 *)`

TO_REAL	
Описание	Приведение к типу REAL (число с плавающей точкой 32 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	REAL
Примеры	`out := TO_REAL( -1000 ); (* out = -1000.0 *)` `out := TO_REAL( 16#8000 ); (* out = -32768.0 *)`

TO_SINT	
Описание	Приведение к типу SINT (целое число со знаком 8 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	SINT
Примеры	`out := TO_SINT( -1 ); (* out = -1 *)` `out := TO_SINT( 16#100 ); (* out = 0 *)`

TO_STRING	
Описание	Приведение к типу STRING.
Количество операндов	1
Тип данных на входе	Любой числовой тип
Тип данных на выходе	STRING
Примеры	`str := TO_STRING( 10.0 ); (* str = '10,0' *)` `str := TO_STRING( -1 ); (* str = '-1' *)`

TO_STRINGFORMAT	
Описание	Приведение к типу STRING, с указателем формата.
Количество операндов	2
Тип данных на входе	Любой числовой тип, STRING
Тип данных на выходе	STRING
Примеры	`str := TO_STRINGFORMAT(10, '%04d'); (* str = '0010' *)`

TO_UDINT	
Описание	Приведение к типу UDINT (целое число без знака 32 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	UDINT
Примеры	`out := TO_UDINT( 10.0 ); (* out = 10 *)` `out := TO_UDINT( 16#FFFFFFFF ); (* out = 4294967295 *)`

TO_UINT	
Описание	Приведение к типу UINT (целое число без знака 16 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	UINT
Примеры	`out := TO_UINT( 1000.0 ); (* out = 1000 *)` `out := TO_UINT( 16#8000 ); (* out = 32768 *)`

TO_USINT	
Описание	Приведение к типу USINT (целое число без знака 8 бит).
Количество операндов	1
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	USINT
Примеры	`out := TO_USINT( -1 ); (* out = 255 *)` `out := TO_USINT( 16#100 ); (* out = 0 *)`

TO_WORD	
Описание	Приведение к типу WORD (строка 16 бит).
Количество операндов	1

TO_WORD	
Тип данных на входе	Любой числовой тип или STRING
Тип данных на выходе	WORD
Примеры	`out := TO_WORD( 1000.0 ); (* out = 16#03E8 *)` `out := TO_WORD( -32768 ); (* out = 16#8000 *)`

Логические функции

AND	
Описание	Логическая функция AND, если вход № 0 и вход № 1 являются BOOL, в противном случае — побитовая AND.
Количество операндов	2
Тип данных на входе	Любой, кроме STRING
Тип данных на выходе	Как на входе
Примеры	`OUT := TRUE AND FALSE;  (* OUT = FALSE *)` `OUT := 16#1234 AND 16#5678;  (* OUT = 16#1230 *)`

NOT	
Описание	Логическая функция NOT, если вход является BOOL, в противном случае — побитовая NOT.
Количество операндов	1
Тип данных на входе	Любой, кроме STRING
Тип данных на выходе	Как на входе
Примеры	`OUT := NOT FALSE; (* OUT = TRUE *)` `OUT := NOT 16#1234;(* OUT = 16#EDCB *)`

OR	
Описание	Логическая функция OR, если вход № 0 и вход № 1 являются BOOL, в противном случае — побитовая OR.
Количество операндов	2
Тип данных на входе	Любой, кроме STRING
Тип данных на выходе	Как на входе
Примеры	`OUT := TRUE OR FALSE; (* OUT = FALSE *)` `OUT := 16#1234 OR 16#5678;(* OUT = 16#567C *)`

XOR	
Описание	Логическая функция XOR, если вход № 0 и вход № 1 являются BOOL, в противном случае — побитовая XOR.
Количество операндов	2
Тип данных на входе	Любой, кроме STRING

XOR	
Тип данных на выходе	Как на входе
Примеры	`OUT := TRUE OR FALSE; (* OUT = TRUE *)` `OUT := 16#1234 OR 16#5678; (* OUT = 16#444C *)`

Функции выбора

LIMIT	
Описание	Устанавливает ограничение, при котором значение входа № 0 может быть только больше или равно значению входа № 1, а также меньше или равно значению входа № 2.
Количество операндов	3
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как на входе

Примеры	`OUT := LIMIT( IN := 4, MN := 0, MX := 5 ); (* OUT = 4 *)`
	`OUT := LIMIT( IN := 88, MN := 0, MX := 5 );(* OUT = 5 *)`
	`OUT := LIMIT( IN := -1, MN := 0, MX := 5 );(* OUT = 0 *)`

MAX	
Описание	Выбор максимального значения
Количество операндов	2–30
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как и макс. входное значение
Примеры	`OUT := MAX( -8, 120, -1000 ); (* OUT = 120 *)`

MIN	
Описание	Выбор минимального значения
Количество операндов	2–30
Тип данных на входе	Любой числовой тип
Тип данных на выходе	Как и мин. входное значение
Примеры	`OUT := MIN( -8, 120, -1000 ); (* OUT = -1000 *)`

MUX	
Описание	Мультиплексор. Выбирает один из N входов в зависимости от входа K.
Количество операндов	3–30
Тип данных на входе	Любой числовой тип

MUX	
Тип данных на выходе	Как и выбранное входное значение
Примеры	`OUT := MUX( 0, A, B, C ); (* OUT = A *)`

SEL	
Описание	Двоичный выбор
Количество операндов	3
Тип данных на входе	BOOL, любой, любой
Тип данных на выходе	Как и выбранное входное значение
Примеры	`OUT := SEL( G := FALSE, IN0 := X, IN1 := 5 );` `(* OUT = X *)`

Стандартные операторы

ADR	
Описание	Адрес
Количество операндов	1
Тип данных на входе	Любой тип
Тип данных на выходе	Указатель на тип
Примеры	`ptr_x := ADR(x)`

IMOVE	
Описание	Интерфейс запроса
Количество операндов	1
Тип данных на входе	Любой тип интерфейса
Тип данных на выходе	Любой тип интерфейса
Примеры	`intf1 ?= obj1;`

JMP	
Описание	Прыжок (условный/с отрицанием)
Количество операндов	0
Тип данных на входе	Вход
Тип данных на выходе	Имя метки
Примеры	`JMP mylabel;`

MOVE	
Описание	Перемещение
Количество операндов	1
Тип данных на входе	Любой тип
Тип данных на выходе	–
Примеры	`MOVE x, y;`

REF	
Описание	Ссылка
Количество операндов	0
Тип данных на входе	Любой тип
Тип данных на выходе	Ссылка на тип
Примеры	`ref_x = REF(x);`

RET	
Описание	Возврат (условный/с отрицанием)
Количество операндов	0
Тип данных на входе	–
Тип данных на выходе	–
Примеры	`RET;`

SIZEOF	
Описание	Вычисление размера
Количество операндов	1
Тип данных на входе	Любой тип
Тип данных на выходе	Числовое значение
Примеры	`mysize := SIZEOF(myvar);`

Строковые функции

CONCAT	
Описание	Конкатенация символов строк
Количество операндов	2
Тип данных на входе	STRING

CONCAT	
Тип данных на выходе	STRING
Примеры	`OUT := CONCAT( 'AB', 'CD' ); (* OUT = 'ABCD' *)`

DELETE	
Описание	Удаляет L символов из IN, начиная с символа в P-й позиции
Количество операндов	3
Тип данных на входе	STRING, UINT, UINT
Тип данных на выходе	STRING
Примеры	`OUT := DELETE( IN := 'ABXYC', L := 2, P := 3 );` `(* OUT = 'ABC' *)`

FIND	
Описание	Находит позицию символа в начале первого вхождения IN2 в IN1. Если вхождение IN2 не обнаруживается, то OUT := 0.
Количество операндов	2
Тип данных на входе	STRING
Тип данных на выходе	UINT
Примеры	`OUT := FIND( IN1 := 'ABCBC', IN2 := 'BC' ); (* OUT = 2 *)`

INSERT	
Описание	Вставляет IN2 в IN1 после символа в P-й позиции
Количество операндов	3
Тип данных на входе	STRING, STRING, UINT
Тип данных на выходе	STRING
Примеры	`OUT := INSERT( IN1 := 'ABC', IN2 := 'XY', P := 2 );` `(* OUT = 'ABXYC' *)`

LEFT	
Описание	Крайние слева L символов в IN
Количество операндов	2
Тип данных на входе	STRING, UINT
Тип данных на выходе	STRING
Примеры	`OUT := LEFT( IN := 'ASTR', L := 3 ); (* OUT = 'AST' *)`

LEN	
Описание	Функция определения длины строки
Количество операндов	1
Тип данных на входе	STRING
Тип данных на выходе	UINT
Примеры	`OUT := LEN( IN := 'ASTR' ); (* OUT = 4 *)`

MID	
Описание	L символов из IN, начиная с символа в P-й позиции
Количество операндов	3
Тип данных на входе	STRING, UINT, UINT
Тип данных на выходе	STRING
Примеры	`OUT := MID( IN := 'ASTR', L := 2, P := 2 );` `(* OUT = 'ST' *)`

REPLACE	
Описание	Заменяет L символов из IN1 на IN2, начиная с символа в P-й позиции
Количество операндов	4
Тип данных на входе	STRING, STRING, UINT, UINT
Тип данных на выходе	STRING
Примеры	`OUT := REPLACE( IN1 := 'ABCDE', IN2 := 'X', L := 2, P := 3 );` `(* OUT = 'ABXE' *)`

RIGHT	
Описание	Крайние справа L символов в IN
Количество операндов	2
Тип данных на входе	STRING, UINT
Тип данных на выходе	STRING
Примеры	`OUT := RIGHT( IN := 'ASTR', L := 3 ); (* OUT = 'STR' *)`

Раздел 14.2

Список инструкций (IL)

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	404
Синтаксис и семантика	404
Стандартные операторы	406
Вызов функций и функциональных блоков	407

Обзор

Описание

В этом разделе рассматривается семантика языка списка инструкций (IL).

Синтаксис и семантика

Синтаксис инструкций IL

Код IL состоит из последовательности инструкций. Каждая инструкция начинается с новой строки и содержит оператор с необязательными модификаторами и, если это необходимо для конкретной операции, один или несколько операндов, разделяемых запятыми. Операндами могут выступать любые представления данных для литералов и переменных.

Перед инструкцией может размещаться идентификационная метка с двоеточием (`:`). Между инструкциями можно вставить пустые строки.

Пример:

```
START:
   LD %IX1 (* Нажать кнопку *)
   ANDN %MX5.4 (* Не блокировано *)
   ST %QX2 (* Вентилятор выключен *)
```

Элементы, из которых состоит каждая инструкция, делятся на следующие группы.

Метка	Оператор [+ модификатор]	Операнд	Комментарий
START:	LD	%IX1	(* Нажать кнопку*)
	ANDN	%MX5.4	(* Не блокировано *)
	ST	%QX2	(* Вентилятор выключен *)

Семантика инструкций IL

- Аккумулятор
  Аккумулятор — это регистр, хранящий значение текущего результата.
- Операторы
  Если не указано иное, семантика операторов выглядит следующим образом:
  `аккумулятор := аккумулятор OP операнд`

  Таким образом, значение аккумулятора заменяется на результат, полученный в ходе выполнения операции OP, примененной к текущему значению аккумулятора и операнду.
  Например, инструкция `AND %IX1` интерпретируется как:

  `аккумулятор := аккумулятор AND %IX1`

  Результатом инструкции `GT %IW10` будет `TRUE`, если текущее значение аккумулятора больше значения входного слова 10, и `FALSE` в обратном случае:

  `аккумулятор := аккумулятор GT %IW10`
- Модификаторы
  Модификатор `N` обозначает побитовое отрицание операнда.
  Модификатор `C` указывает на то, что связанная с ним инструкция может быть выполнена только в том случае, если значение текущего результата оценки является логической 1 (или логическим 0, если оператор используется с модификатором `N`).
  Модификатор «левая скобка» «(» указывает на то, что обработку оператора следует отложить до тех пор, пока не встретится оператор правой скобки «)». Ниже приведена последовательность инструкций в скобках, на которую ссылается данная инструкция:

  `аккумулятор := аккумулятор AND (%MX1.3 OR %MX1.4)`

Стандартные операторы

Описание

Ниже перечислены стандартные операторы с допустимыми модификаторами и операндами.

Оператор	Модификаторы	Поддерживаемые типы операндов: Acc_type, Op_type	Семантика
LD	N	Любой, любой	Устанавливает аккумулятор равным операнду.
ST	N	Любой, любой	Сохраняет аккумулятор в область памяти, занимаемую операндом.
S		BOOL, BOOL	Устанавливает для операнда значение TRUE, если значение аккумулятора TRUE.
R		BOOL, BOOL	Устанавливает для операнда значение FALSE, если значение аккумулятора TRUE.
AND	N, (	Любой, кроме REAL; любой, кроме REAL	Логическое или побитовое AND
OR	N, (	Любой, кроме REAL; любой, кроме REAL	Логическое или побитовое OR
XOR	N, (	Любой, кроме REAL; любой, кроме REAL	Логическое или побитовое XOR
NOT		Любой, кроме REAL	Логическое или побитовое NOT
ADD	(	Любой, кроме BOOL	Сложение
SUB	(	Любой, кроме BOOL	Вычитание
MUL	(	Любой, кроме BOOL	Умножение
DIV	(	Любой, кроме BOOL	Деление
MOD	(	Любой, кроме BOOL	Деление по модулю
GT	(	Любой, кроме BOOL	Сравнение:
GE	(	Любой, кроме BOOL	Сравнение: =
EQ	(	Любой, кроме BOOL	Сравнение: =
NE	(	Любой, кроме BOOL	Сравнение:
LE	(	Любой, кроме BOOL	Сравнение:
LT	(	Любой, кроме BOOL	Сравнение:
JMP	C, N	Метка	Прыжок к метке
CAL	C, N	Имя экземпляра функционального блока	Вызов функционального блока
RET	C, N		Возвращает результат из вызванной программы, функции или функционального блока.
)			Оценивает отложенную операцию.

Вызов функций и функциональных блоков

Вызов функций

Функции (как было сказано в соответствующем разделе) вызываются путем помещения имени функции в поле оператора. Подобный вызов выглядит следующим образом:

```
LD 1
MUX 5, var0, -6.5, 3.14
ST vRES
```

Первый аргумент не содержится в списке входов, но в качестве первого аргумента функции используется аккумулятор. Дополнительные аргументы (начиная со второго) при необходимости указываются в поле операндов и разделяются запятыми в порядке их объявления. Например, оператор `MUX` в предыдущей таблице принимает пять операндов, первый из которых загружается в аккумулятор, а остальные четыре аргумента последовательно указываются после имени функции.

При вызове функции действуют следующие правила:

- Аргументам `VAR_INPUT` могут присваиваться пустые значения, константы или переменные.
- Выполнение функции завершается по достижении инструкции `RET` или физического конца функции. При этом выходная переменная функции копируется в аккумулятор.

Вызов функциональных блоков

Функциональные блоки (как было сказано в соответствующем разделе) могут вызываться условно и безусловно с помощью оператора `CAL`. Подобный вызов выглядит следующим образом:

```
LD A
ADD 5
ST INST5.IN1
LD 3.141592
ST INST5.IN2
CAL INST5
LD INST5.OUT1
ST vRES
LD INST5.OUT2
ST vVALID
```

Этот метод вызова эквивалентен `CAL` со списком аргументов, содержащим только одну переменную с именем экземпляра функционального блока.

Входные аргументы передаются в экземпляр функционального блока, а выходные аргументы считываются из него посредством операций `ST/LD`, выполняемых над операндами, следующим образом:

```
ИмяЭкземпляраФБ.ВходВыхПер
```

Ключевое слово	Описание
`ИмяЭкземпляраФБ`	Имя вызываемого экземпляра.
`ВходВыхПер`	Входная или выходная переменная для записи/чтения.

Раздел 14.3

Функциональная блок-схема (FBD)

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	408
Представление линий и блоков	408
Направление движения в сетях	409
Вычисление сетей	409
Элементы управления исполнением	411

Обзор

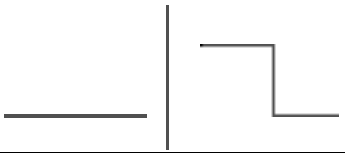
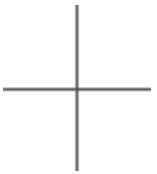
Описание

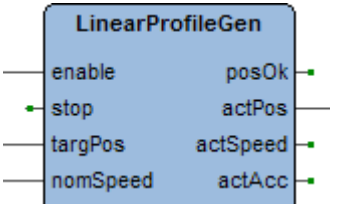
В этом разделе рассматривается семантика языка функциональных блок-схем (FBD).

Представление линий и блоков

Описание

Элементы графического языка отрисовываются с использованием графических или полуграфических компонентов, как показано в следующей таблице.

Функция	Пример
Линии	
Пересечение линии с соединением	

Функция	Пример
Блоки с соединительными линиями и неподсоединенными выводами	

Хранение данных или связь с элементами данных невозможно реализовать с использованием коннекторов. Поэтому во избежание неоднозначности коннекторам не может присваиваться никакой идентификатор.

Направление движения в сетях

Описание

Под сетью понимается максимальный набор взаимосвязанных графических элементов. Метка сети, ограниченная справа двоеточием (`:`), может быть ассоциирована с любой сетью или группой сетей. Область видимости сети и ее метки ограничивается программным модулем, в котором расположена сеть.

Графические языки используются для представления потока абстрактной величины через одну или несколько сетей, представляющих собой схему управления. В случае функциональных блок-схем обычно используется понятие «поток сигналов», аналогом которого является поток сигналов между элементами системы обработки сигналов. Поток сигналов на языке FBD поступает со стороны выхода (справа) функции или функционального блока на сторону входа (слева) подключенной функции или функционального блока.

Вычисление сетей

Порядок вычисления сетей

Порядок вычисления сетей и их элементов не обязательно совпадает с порядком, в котором они отмечены или показаны. Если тело программного модуля состоит из одной или нескольких сетей, результаты вычисления сети внутри вышеупомянутого тела в функциональном отношении эквивалентны соблюдению нижеследующих правил:

- Ни один элемент сети не вычисляется до тех пор, пока не будут вычислены состояния всех ее входов.
- Вычисление элемента сети не завершается до тех пор, пока не будут вычислены состояния всех его выходов.
- Как уже упоминалось при описании редактора FBD, каждой сети автоматически присваивается номер. Внутри программного модуля сети вычисляются в соответствии с последовательностью их номеров: сеть `N` вычисляется перед сетью `N+1`, если не указано иное с помощью элементов управления выполнением.

Комбинация элементов

Элементы языка FBD должны соединяться линиями потока сигналов.

Выходы блоков не должны соединяться между собой. В частности, конструкция `wired-OR` («монтажное ИЛИ») языка LD не допускается, так как необходимо использовать явный блок булевой логики `OR`.

Обратная связь

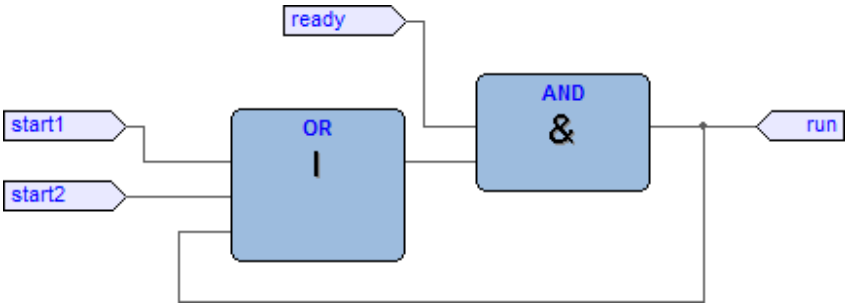
Считается, что путь обратной связи возникает в сети в том случае, если выход функции или функционального блока используется в качестве входа в предшествующую функцию или функциональный блок. Связанная с ним переменная называется переменной обратной связи.

Пути обратной связи могут использоваться в соответствии со следующими правилами:

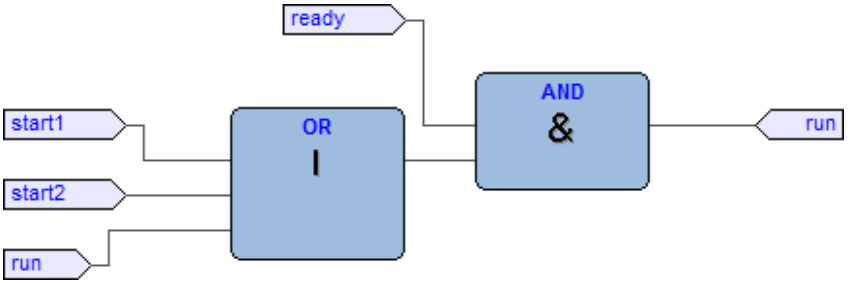
- Переменные обратной связи должны быть инициализированы, а их начальное значение используется во время первого вычисления сети. В редакторе глобальных переменных, редакторе локальных переменных или редакторе параметров можно узнать, как выполняется инициализация того или иного элемента.
- После вычисления элемента с переменной обратной связи на выходе новое значение переменной обратной связи используется вплоть до следующего вычисления элемента.

Например, логическая переменная `RUN` является переменной обратной связи в следующем примере.

Явный цикл:



Неявный цикл:



Элементы управления исполнением

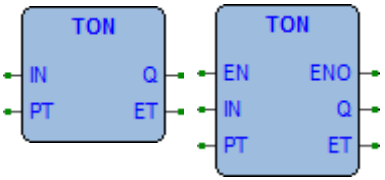
Сигналы EN/ENO

Блоки EcoStruxure Machine Expert - HVAC характеризуются дополнительным логическим входом EN (включение) и ENO (выход из состояния) и объявляются следующим образом.

EN	ENO

VAR_INPUT
  EN: BOOL := 1;
END_VAR
	VAR_OUTPUT
	  ENO: BOOL;
	END_VAR

Обратитесь к разделу «Изменение свойств блоков» *(см. стр. 235)*, чтобы узнать, как добавить эти выводы в блок:



При использовании этих переменных выполнение операций, определенных этим блоком, контролируется по следующим правилам:

- Если при вызове блока значение EN равно FALSE, то операции, определенные телом функции, не выполняются, а значение ENO сбрасывается ПЛК-системой на FALSE.
- В противном случае ПЛК-система устанавливает для ENO значение TRUE, при этом выполняются операции, заданные в теле блока.

Прыжки

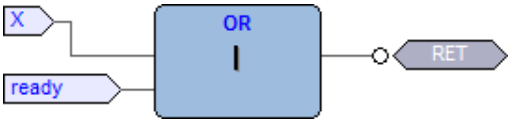
Прыжки изображаются логической сигнальной линией, заканчивающейся двойной стрелкой. Сигнальная линия для состояния прыжка формируется в логической переменной или на логическом выходе функции или функционального блока. Передача управления программой соответствующей метке сети происходит, когда логическим значением сигнальной линии является TRUE. Таким образом, безусловный переход можно рассматривать как особый случай условного перехода.

Целью перехода является метка сети в программном модуле, внутри которого происходит переход.

Символ/пример	Описание
	Безусловный прыжок
	Условный прыжок
	Пример: Сеть условий прыжка

Условные возвраты

- Условные возвраты из функций и функциональных блоков реализуются с помощью конструкции `RETURN`, как показано в следующей таблице. Выполнение программы передается обратно в вызывающую сущность, когда логический вход имеет значение `TRUE`, и продолжается в обычном режиме, когда логический вход имеет значение `FALSE`.
- Безусловные возвраты обеспечиваются за счет физического конца функции или функционального блока.

Символ/пример	Описание
	Условный возврат
	Пример: Сеть условий возврата

Раздел 14.4

Релейная схема (LD)

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	413
Шины питания	414
Соединительные элементы и состояния	414
Контакты	415
Катушки	416
Операторы, функции и функциональные блоки	417

Обзор

Описание

В этом разделе рассматривается семантика языка релейных схем (LD).

Шины питания

Описание

Сеть LD слева и справа ограничена вертикальными линиями, которые называются левой шиной питания и, соответственно, правой шиной питания. Правая шина питания может быть явно реализована в EcoStruxure Machine Expert - HVAC и отображается всегда.

Обе шины питания всегда соединены горизонтальной связью — сигнальной связью. Элементы LD должны размещаться и подключаться к сигнальной связи.

Описание	Символ
Левая шина питания (с подключенной горизонтальной связью)	
Правая шина питания (с подключенной горизонтальной связью)	
Шины питания, соединенные сигнальной связью	

Соединительные элементы и состояния

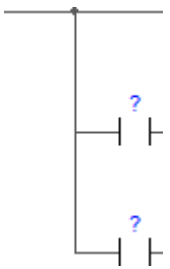
Описание

Соединительные элементы могут быть горизонтальными или вертикальными. Состояние соединительных элементов обозначается как `ВКЛ.` или `ВЫКЛ.`, что соответствует логическим значениям 1 или 0. Термин «состояние связи» является синонимом термина «поток мощности».

Следующие свойства относятся к соединительным элементам:

- Состояние левой шины всегда считается `ВКЛ.` Для правой шины состояние не определено.
- Горизонтальный соединительный элемент обозначается при помощи горизонтальной линии. Горизонтальный соединительный элемент передает состояние элемента слева элементу, расположенному справа от него.
- Вертикальный соединительный элемент состоит из вертикальной линии, пересекающейся с одним или несколькими горизонтальными соединительными элементами с каждой стороны. Состояние вертикальной связи представляет собой включающее ИЛИ для состояний ВКЛ. горизонтальных связей с левой стороны, т. е. состояние вертикальной связи соответствует:
  - `ВЫКЛ.`, если состояния всех подключенных горизонтальных связей слева `ВЫКЛ.`
  - `ВКЛ.`, если состояние одной или более подключенных горизонтальных связей слева `ВКЛ.`

- Состояние вертикальной связи копируется на все подключенные горизонтальные связи справа от нее.
- Состояние вертикальной связи не копируется на любые подключенные горизонтальные связи слева от нее.

Описание	Символ
Вертикальная связь с подключенными горизонтальными связями	

Контакты

Описание

Контакт — это элемент, который передает на горизонтальную связь справа состояние, эквивалентное логическому И состояния горизонтальной связи слева, вместе с соответствующей функцией связанного логического входа, выхода или переменной в памяти.

Контакт не изменяет значение связанной с ним логической переменной. Стандартные символы контактов приводятся в таблице ниже.

Тип	Описание	Символ
Нормально разомкнутый контакт	Состояние левой связи копируется в правую связь, если у связанной с ней логической переменной состояние ВКЛ. В противном случае состояние правой связи равно ВЫКЛ.	
Нормально замкнутый контакт	Состояние левой связи копируется в правую связь, если у связанной с ней логической переменной состояние ВЫКЛ. В противном случае состояние правой связи равно ВЫКЛ.	
Контакт обнаружения нарастания фронта	Состояние правой связи соответствует ВКЛ. после завершения оценки данного элемента и перехода к следующему, когда регистрируется переход связанной переменной из состояния ВЫКЛ. в состояние ВКЛ., если состояние левой связи ВКЛ. Состояние правой связи должно быть ВЫКЛ. все оставшееся время.	
Контакт обнаружения падения фронта	Состояние правой связи соответствует ВКЛ. после завершения оценки данного элемента и перехода к следующему, когда регистрируется переход связанной переменной из состояния ВКЛ. в состояние ВЫКЛ., если состояние левой связи ВКЛ. Состояние правой связи должно быть ВЫКЛ. все оставшееся время.	

Катушки

Описание

Катушка копирует состояние связи слева от нее на связь справа от нее без изменения и сохраняет соответствующую функцию состояния или перехода левой связи в соответствующей логической переменной.

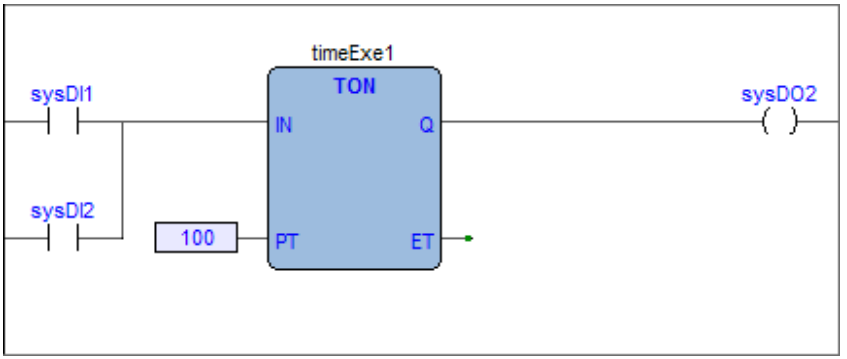
Стандартные символы катушки приводятся в таблице ниже.

Тип	Описание	Символ
Катушка	Состояние левой связи копируется в связанную с ней логическую переменную.	-( )-
Инвертирующая катушка	Инвертированное значение состояния левой связи копируется в связанную логическую переменную, т. е. если состояние левой связи ВЫКЛ., то состояние связанной переменной ВКЛ., и наоборот.	-(/)-
Катушка установки	Связанная с ней логическая переменная устанавливается в состояние ВКЛ., когда левая связь находится в состоянии ВКЛ., и остается установленной до тех пор, пока не будет сброшена катушкой сброса.	-(S)-
Катушка сброса	Связанная с ней логическая переменная сбрасывается в состояние ВЫКЛ., когда левая связь находится в состоянии ВКЛ., и остается сброшенной до тех пор, пока не будет установлена катушкой установки.	-(R)-
Катушка обнаружения нарастания фронта	Если регистрируется переход левой связи из состояния ВЫКЛ. в состояние ВКЛ., после завершения вычисления данного элемента и перехода к следующему значение состояния связанной логической переменной соответствует ВКЛ.	-(P)-
Катушка обнаружения падения фронта	Если регистрируется переход левой связи из состояния ВКЛ. в состояние ВЫКЛ., после завершения вычисления данного элемента и перехода к следующему значение состояния связанной логической переменной соответствует ВКЛ.	-(N)-

Операторы, функции и функциональные блоки

Описание

Функции и функциональные блоки на языке LD представляются таким же образом, что и в языке FBD. На каждом блоке должен отображаться минимум один логический вход и один логический выход, чтобы обеспечить прохождение потока мощности через блок, как показано на рисунке ниже:



Раздел 14.5

«Структурный текст» (ST)

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	418
Выражения	418
Инструкции в языке ST	420
Присвоения	421
Инструкции вызова функций и функциональных блоков	422
Инструкции выбора	424
Инструкции итерирования	426

Обзор

Описание

В этом разделе рассматривается семантика языка «Структурный текст» (ST).

«Структурный текст» — это текстовый язык программирования высокого уровня, схожий с PASCAL или C. Код программы представляет собой набор выражений и инструкций. В отличие от языка IL, здесь можно использовать многочисленные конструкции для написания циклов, что позволяет реализовывать сложные алгоритмы.

Выражения

Описание

Выражение — это конструкция, которая при выполнении возвращает значение, соответствующее одному из типов данных, перечисленных в таблице элементарных типов данных *(см. стр. 373)*. EcoStruxure Machine Expert — HVAC не ограничивает выражения по длине.

Выражения формируются из операторов, операндов и (или) присвоений. Операнд может быть константой, переменной, вызовом функции или другим выражением.

Операнды

Операнд может быть литералом, переменной, вызовом функции или другим выражением.

Операторы

Откройте таблицу операторов, чтобы посмотреть список всех операторов, поддерживаемых языком ST. Выполнение выражения заключается в применении операторов к операндам в последовательности, определенной правилами приоритета операторов.

Правила приоритета операторов

Операторы имеют различные уровни приоритета, которые указаны в таблице операторов. В выражении сначала применяется оператор с наибольшим приоритетом, затем оператор со следующим по значимости приоритетом, и так далее до завершения выполнения. Операторы с одинаковым приоритетом применяются слева направо — в том виде, в каком они записаны в выражении.

Например, если A, B, C и D имеют тип INT со значениями 1, 2, 3 и 4 соответственно, то результатом следующего выражение будет –9:

```
A + B - C * ABS(D)
```

При этом следующее выражение будет равно 0:

```
(A + B - C) * ABS(D)
```

Если оператор имеет два операнда, сначала вычисляется самый левый операнд. Например, в выражении `SIN(A)*COS(B)` сначала вычисляется значение `SIN(A)`, затем `COS(B)`, после чего вычисляется их произведение.

Функции вызываются в виде элементов выражений, состоящих из имени функции, за которым следует список аргументов в скобках, как было указано в соответствующем разделе.

Операторы языка ST

Операция	Символ	Приоритет
Скобки	(<выражение>)	НАИВЫСШИЙ
Выполнение функции	<имяфункции> (<списокаргументов>)	.
Отрицание	- NOT	.
Возведение в степень	**	.
Умножение Деление Остаток от деления	* / MOD	.
Сложение Вычитание	+ -	.
Сравнение	<, >, <=, >=	.
Равенство Неравенство	= <>	.
Логическое И	AND	.
Логическое исключающее ИЛИ	XOR	.
Логическое ИЛИ	OR	НАИНИЗШИЙ

Инструкции в языке ST

Описание

Все инструкции должны удовлетворять следующим правилам:

- они заканчиваются точкой с запятой;
- в отличие от языка IL, возврат каретки или символ новой строки интерпретируется так же, как и символ пробела;
- EcoStruxure Machine Expert - HVAC не ограничивает инструкции по максимальной длине.

Инструкции ST можно разделить на классы в соответствии с их семантикой.

Присвоения

Семантика

Инструкция присваивания заменяет текущее значение одиночной или многоэлементной переменной на результат вычисления выражения.

Инструкция присваивания используется также для присваивания значения, возвращаемого функцией, путем помещения имени функции слева от оператора присваивания в теле объявления функции. Возвращаемое функцией значение является результатом последнего вычисления такого присваивания.

Синтаксис

Инструкция присваивания состоит из ссылки на переменную с левой стороны, за которой следует оператор присваивания `:=`, за которым следует вычисляемое выражение. Например, инструкцию `A := B ;` можно использовать для замены единственного значения данных переменной A на текущее значение переменной B, если они оба имеют тип `INT`.

Примеры

Присваивание:

```
a := b ;
```

Присваивание:

```
pCV := pCV + 1 ;
```

Присваивание с вызовом функции:

```
c := SIN( x );
```

Присваивание выходного значения функции:

```
FUNCTION SIMPLE_FUN : REAL
   <Объявление переменных>
   ...
   <Тело функции>
   ...
   SIMPLE_FUN := a * b - c ;
END_FUNCTION
```

Инструкции вызова функций и функциональных блоков

Семантика

- Функции вызываются в виде элементов выражений, состоящих из имени функции, за которым следует список аргументов в скобках. Каждый аргумент может быть литералом, переменной или выражением любой сложности.
- Функциональные блоки вызываются с помощью инструкции, состоящей из имени экземпляра функционального блока, за которым в скобках указывается список аргументов. Поддерживаются вызовы как с формальным списком аргументов, так и с присваиванием аргументов.
- RETURN: управляющие инструкции функций и функциональных блоков состоят из механизмов для вызова функциональных блоков и возврата управления вызывающей сущности до достижения физического конца функции или функционального блока. Инструкция RETURN обеспечивает досрочный выход из функции или функционального блока (например, в результате вычисления инструкции IF).

Синтаксис

Функция:

```
dst_var := function_name( arg1, arg2 , … , argN );
```

Функциональный блок с формальным списком аргументов:

```
instance_name(var_in1 := arg1 ,
              var_in2 := arg2 ,
              ... ,
              var_inN := argN );
```

Функциональный блок с присваиванием аргументов:

```
instance_name.var_in1 := arg1;
...
instance_name.var_inN := argN;
instance_name();
```

Управляющая инструкция для функций и функциональных блоков:

```
RETURN ;
```

Примеры

Вызов функционального блока с формальным списком аргументов:

```
CMD_TMR( IN := %IX5,PT:= 300 ) ;
```

Вызов функционального блока с присваиванием аргументов:

```
IN := %IX5 ;
PT:= 300 ;
CMD_TMR() ;
```

Использование результата вывода функционального блока:

```
a := CMD_TMR.Q;
```

Досрочный выход из функции или функционального блока:

```
RETURN ;
```

Инструкции выбора

Семантика

К инструкциям выбора относятся инструкции IF и CASE. Инструкция выбора выбирает для выполнения одну (или несколько) из инструкций, входящих в ее состав, в зависимости от заданного условия.

- IF: инструкция IF указывает, что группа инструкций должна выполняться только в том случае, если связанное с ней логическое выражение вычисляется со значением TRUE. Если условие ложно, то не выполняется ни одна инструкция либо выполняется группа инструкций, указанная после ключевого слова ELSE (или ключевого слова ELSIF, если связанное с ним логическое условие истинно).
- CASE: инструкция CASE состоит из выражения, которое вычисляет переменную типа DINT (т. н. «селектор»), и списка групп инструкций, где каждая группа обозначается одним или несколькими целыми числами или диапазонами целочисленных значений, в зависимости от ситуации. Она указывает, что должна быть выполнена первая группа инструкций, один из диапазонов которых содержит вычисленное значение селектора. Если значение селектора не встречается ни в одном из диапазонов, выполняется последовательность инструкций после ключевого слова ELSE (если оно встречается в инструкции CASE). В противном случае не выполняется ни одна из последовательностей инструкций.

EcoStruxure Machine Expert - HVAC не ограничивает количество вариантов выбора в инструкции CASE.

Синтаксис

В квадратных скобках указывается необязательный код, а в фигурных скобках — повторяемые фрагменты кода.

IF:

```
IF <выражение 1> THEN
   <список инструкций>
[ { ELSIF <выражение 2> THEN
   <список инструкций> } ]
ELSE
   <список инструкций>
END_IF ;
```

CASE:

```
CASE <выражение 1> OF
   intv [ {, intv } ] :
      <список инструкций>
   { intv [ {, intv } ] :
      <список инструкций> }
[ ELSE
   <список инструкций> ]
END_CASE ;
```

intv является либо константой, либо интервалом: a или a..b

Примеры

Инструкция `IF`:

```
IF d < 0.0 THEN
   nRoots := 0 ;
ELSIF d = 0.0 THEN
   nRoots := 1 ;
   x1 := -b / (2.0 * a) ; ELSE
   nRoots := 2 ;
   x1 := (-b + SQRT(d)) / (2.0 * a) ;
   x2 := (-b - SQRT(d)) / (2.0 * a) ;
END_IF ;
```

Инструкция `CASE`:

```
CASE tw OF
   1, 5:
      display := oven_temp ;
   2:
      display := motor_speed ;
   3:
      display := gross_tare;
   4, 6..10:
      display := status(tw - 4) ;
ELSE
   display := 0;
   tw_error := 1;
END_CASE ;
```

Инструкции итерирования

Семантика

Инструкции итерирования указывают, что группа связанных с ними инструкций выполняется многократно. Инструкция FOR используется, если количество итераций можно определить заранее, в противном случае используются конструкции WHILE или REPEAT.

- FOR: инструкция FOR указывает, что последовательность инструкций выполняется многократно вплоть до ключевого слова END_FOR, при этом изменяющееся значение присваивается управляющей переменной цикла FOR. Управляющая переменная, начальное и конечное значение являются выражениями одного и того же целочисленного типа (например, SINT, INT или DINT) и не могут изменяться в ходе выполнения какой-либо повторяющейся инструкции. Инструкция FOR инкрементирует/декрементирует управляющую переменную от начального значения до конечного значения с шагом, определяемым значением выражения. По умолчанию в качестве значения инкремента используется 1. Проверка условия завершения производится в начале каждой итерации таким образом, чтобы последовательность инструкций не выполнялась, если начальное значение превышает конечное.
- WHILE: инструкция WHILE инициирует повторение последовательности инструкций вплоть до ключевого слова END_WHILE до тех пор, пока связанное с ней логическое выражение не станет ложным. Если выражение изначально ложно, то группа инструкций вообще не выполняется.
- REPEAT: инструкция REPEAT инициирует повторение последовательности инструкций вплоть до ключевого слова UNTIL (минимум один раз), пока связанное с ней логическое условие не станет истинным.
- EXIT: инструкция EXIT используется для досрочного прекращения итераций до выполнения условия прекращения. Если инструкция EXIT используется внутри вложенных итеративных конструкций, то выход осуществляется из вложенного цикла, в котором присутствует инструкция EXIT, при этом управление передается инструкции, указанной непосредственно после конструкции, закрывающей тело цикла (END_FOR, END_WHILE или END_REPEAT).

ПРИМЕЧАНИЕ. Инструкции WHILE и REPEAT нельзя использовать для выполнения межпроцессной синхронизации, например в качестве «цикла ожидания» с определяемым извне условием завершения. Для этого должны использоваться соответствующие элементы SFC.

Синтаксис

В квадратных скобках указывается необязательный код, а в фигурных скобках — повторяемые фрагменты кода. Если условие завершения некорректно, то это может привести к бесконечному циклу.

ПРИМЕЧАНИЕ

НЕИСПРАВНАЯ РАБОТА ОБОРУДОВАНИЯ

- Убедитесь, что переменная, используемая в инструкциях FOR, обладает достаточной размерностью (имеет большой верхний предел), чтобы вместить величину <КОНЕЧНОЕ_ЗНАЧЕНИЕ> + 1.
- Убедитесь, что цикл WHILE будет завершен в рамках инструкций цикла, создав условие FALSE для логического выражения.
- Убедитесь, что цикл REPEAT будет завершен в рамках инструкций цикла, создав условие TRUE для логического выражения.

Несоблюдение этих указаний может привести к повреждению оборудования.

FOR:

```
FOR control_var := init_val TO end_val [ BY increm_val ] DO
   <список инструкций>
END_FOR ;
```

WHILE:

```
WHILE <выражение> DO
   <список инструкций>
END_WHILE ;
```

REPEAT:

```
REPEAT
   <список инструкций>
   UNTIL <выражение>
END_REPEAT ;
```

Примеры

Инструкция FOR:

```
j := 101 ;
FOR i := 1 TO 100 BY 2 DO
   IF arrvals[i] = 57 THEN
      j := i ;
      EXIT ;
   END_IF ;
END_FOR ;
```

Инструкция WHILE:

```
j := 1 ;
WHILE j <=100 AND arrvals[i] <> 57 DO
   j := j + 2 ;
END_WHILE ;
```

Инструкция REPEAT:

```
j := -1 ;
REPEAT
   j := j + 2 ;
   UNTIL j = 101 AND arrvals[i] = 57
END_REPEAT ;
```

Раздел 14.6

Последовательная функциональная схема (SFC)

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	429
Шаги	430
Переходы	432
Правила изменения	433
Управляющие флаги SFC	438
Проверка программного модуля SFC из других программ	439

Обзор

Описание

В данном разделе рассматриваются элементы последовательных функциональных схем (SFC), которые описывают внутреннюю структуру программного модуля, написанного на одном из стандартизированных языков и осуществляющего последовательное управление ПЛК. Определения в этом разделе взяты из стандарта МЭК 60848 [ГОСТ IEC 60848] с необходимыми изменениями для преобразования представлений из стандартной документации в набор элементов управления исполнением программного модуля ПЛК.

Поскольку элементам SFC требуется возможность сохранения информации о состоянии, то единственными программными модулями, которые могут быть структурированы с помощью этих элементов, являются функциональные блоки и программы.

Если какая-либо часть программного модуля разбивается на элементы SFC, то тем самым разбивается весь программный модуль. Если для программного модуля разбивка SFC не выполняется, то весь программный модуль рассматривается как единое действие, которое выполняется под контролем вызывающей сущности.

Элементы SFC

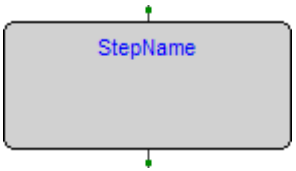
Элементы SFC обеспечивают возможность разделения программного модуля ПЛК на набор шагов и переходов, объединяемых между собой направленными связями. С каждым шагом связан соответствующий набор действий, а с каждым переходом связано условие.

Шаги

Определение

Шаг представляет собой ситуацию, когда поведение программного модуля по отношению к своим входам и выходам следует набору правил, определяемых соответствующими действиями в рамках шага. Шаг либо активен, либо неактивен. В любой момент времени состояние программного модуля определяется набором активных шагов и значениями его внутренних и выходных переменных.

Графически шаг представляется в виде блока, содержащего имя шага в форме идентификатора. Связи, ведущие к шагу, могут изображаться в виде вертикальной линии, присоединяемой к верхней части шага. Связи, ведущие от шага, могут изображаться в виде вертикальной линии, присоединяемой к нижней части шага.

Представление	Описание
	Шаг (графическое представление с направленными связями)

EcoStruxure Machine Expert - HVAC не устанавливает ограничения по максимальному количеству шагов для SFC в пределах доступной памяти.

Флаг шага

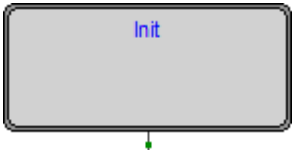
Флаг шага (активное или неактивное состояние шага) может быть представлен логическим значением булевой переменной `***_x`, где `***` — имя шага. Эта булева переменная имеет значение `TRUE`, когда соответствующий шаг активен, и `FALSE`, когда он неактивен. Область видимости имен и флагов шагов ограничивается программным модулем, в котором находятся эти шаги.

Представление	Описание
`STEP Name_x`	Флаг шага = `TRUE`, если `STEP Name_x` активно, в противном случае = `FALSE`

Начальный шаг

Начальное состояние программного модуля представлено начальными значениями его внутренних и выходных переменных, а также набором начальных шагов, т. е. шагов, которые активны изначально. Каждая сеть SFC или ее текстовый эквивалент имеет ровно один начальный шаг. Начальный шаг можно изобразить графически, воспользовавшись двойной линией для рамки, как показано ниже. Для проведения инициализации системы начальным состоянием по умолчанию является `FALSE` для обычных шагов и `TRUE` для начальных шагов.

EcoStruxure Machine Expert - HVAC не может скомпилировать сеть SFC, не содержащую ни одного начального шага.

Представление	Описание
	Начальный шаг (графическое представление с направленными связями)

Действия

Действием может быть:

- набор инструкций на языке IL;
- набор сетей на языке FBD;
- набор ступенек на языке LD;
- набор инструкций на языке ST;
- последовательная функциональная схема (SFC), организованная, как показано в данном разделе.

С каждым шагом может быть связано ноль или несколько действий. Действия объявляются с помощью одного из текстовых структурных элементов, перечисленных в таблице ниже.

Структурный элемент	Описание

`STEP StepName :`
`(* Тело шага *)`
`END_STEP`
	Шаг (текстовый вид)

`INITIAL_STEP StepName :`
`(* Тело шага *)`
`END_STEP`
	Начальный шаг (текстовый вид)

Такой структурный элемент присутствует в файле **Isc** для каждого шага, имеющего хотя бы одно связанное с ним действие.

Квалификаторы действия

Момент выполнения действия, связанного с шагом, зависит от квалификатора.

В EcoStruxure Machine Expert - HVAC используются следующие квалификаторы действий.

Квалификатор	Описание	Пояснение
N	Не сохраняется (нулевой квалификатор)	Действие выполняется до тех пор, пока шаг остается активным.
P	Импульс	Действие выполняется однократно за один шаг активации, независимо от количества циклов, в течение которых шаг остается активным.

Если с шагом не связано ни одно действие, то считается, что он имеет функцию WAIT, т. е. ожидает, когда условие перехода преемника станет верным.

Прыжки

Направленные связи ведут только вниз. Чтобы вернуться на шаг вверх с нижнего шага, нельзя просто протянуть логическую линию. Существует специальный тип блока, называемый прыжком, который позволяет реализовать такую задачу.

Блок прыжка логически эквивалентен шагу, так как оба всегда должны разделяться переходом. Единственной задачей прыжка является активация флага предыдущего шага и активация флага шага, на который тот указывает.

Представление	Описание
	Прыжок (логическая связь с шагом назначения)

Переходы

Определение

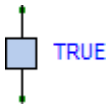
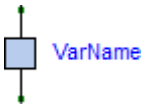
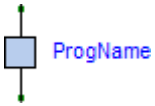
Переход определяет, при каких условиях управление передается от одного или нескольких шагов, предшествующих переходу, одному или нескольким шагам-преемникам по соответствующей направленной связи. Переход изображается в виде небольшого серого квадрата, пересекающего вертикальную направленную связь.

Направление изменения в соответствии с направленными связями — снизу вверх (от шагов-предшественников к шагам-преемникам).

Условие перехода

Каждый переход имеет связанное с ним условие, которое является результатом вычисления одного логического выражения. Условие перехода, которое всегда истинно, обозначается ключевым словом `TRUE`, а условие перехода, которое всегда ложно, обозначается ключевым словом `FALSE`.

Условие можно связать с переходом одним из следующих способов.

Представление	Описание
	Поместив соответствующую логическую константу {`TRUE, FALSE`} рядом с вертикальной направленной связью.
	Объявив логическую переменную, значение которой определяет, будет ли переход очищен.
	Написав фрагмент кода на любом из языков, поддерживаемых EcoStruxure Machine Expert - HVAC, за исключением SFC. Результат выполнения такого кода определяет условие перехода.

Область видимости имени перехода ограничивается программным модулем, в котором расположен переход.

Правила изменения

Введение

Начальная ситуация сети SFC определяется начальным шагом, который активируется после инициализации программы или функционального блока, содержащего сеть.

Изменение активных состояний шагов происходит по направленным связям в результате очистки одного или нескольких переходов.

Переход становится активен, если все предыдущие шаги, связанные с соответствующим символом перехода направленными связями, активны. Очистка перехода происходит в момент, когда переход активен и когда соответствующее условие перехода верно.

Очистка перехода приводит к деактивации (или «сбросу») всех предшествующих шагов, связанных с соответствующим символом перехода направленными связями, и активации всех следующих шагов.

Элементы SFC всегда соединяются путем чередования шаг/переход и переход/шаг, то есть:

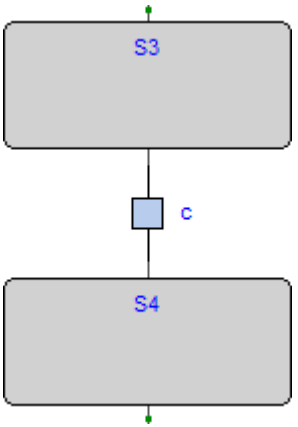
- два шага никогда не связываются напрямую, а всегда разделяются переходом;
- два перехода никогда не связываются напрямую, а всегда разделяются шагом.

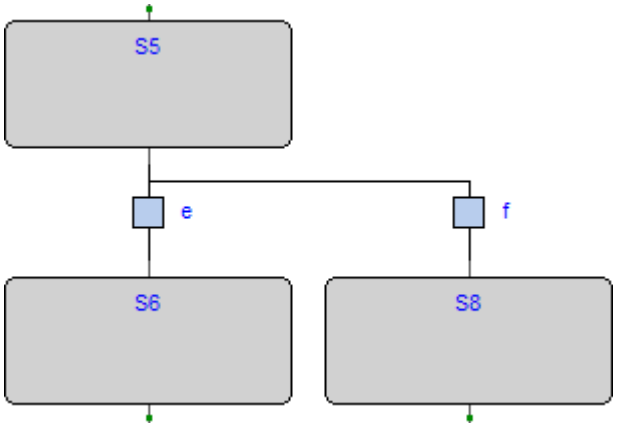
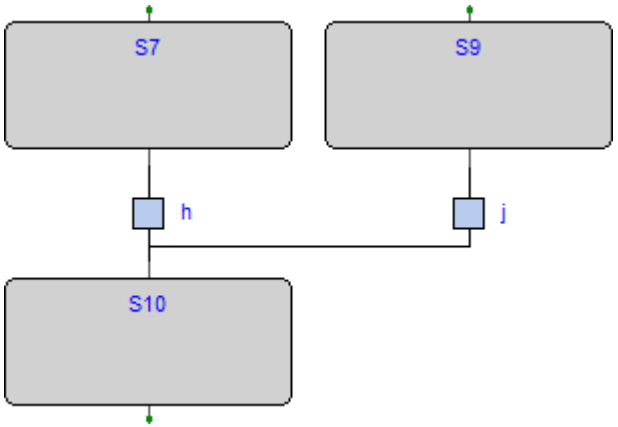
Если очистка перехода приводит к активации нескольких шагов одновременно, то последовательности, к которым эти шаги относятся, называются одновременными последовательностями. После их одновременной активации изменение каждой из этих последовательностей осуществляется автономно. Чтобы подчеркнуть особую природу таких конструкций, расхождение и схождение одновременных последовательностей обозначается двойной горизонтальной линией.

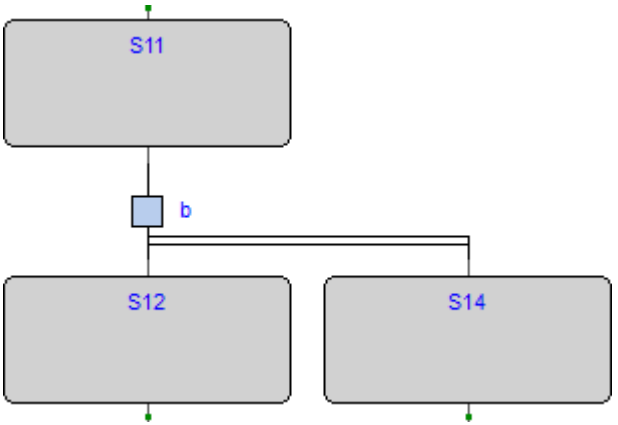
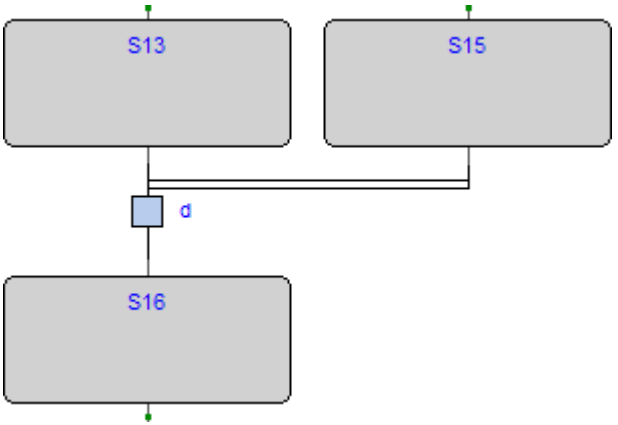
Время очистки перехода в теории может рассматриваться как минимальное, но оно никогда не может равняться нулю. На практике время, затрачиваемое на очистку, определяется реализацией ПЛК: несколько переходов, которые можно очистить одновременно, очищаются одновременно в рамках временных ограничений конкретной реализации ПЛК и ограничений по приоритету, определенных в таблице изменения последовательности. По этой же причине длительность действия шага никогда не может быть равной нулю. Проверка условий перехода преемника активного шага не должна выполняться до тех пор, пока эффект активации шага не распространится на весь программный модуль, в котором объявлен этот шаг.

Таблица изменения последовательности

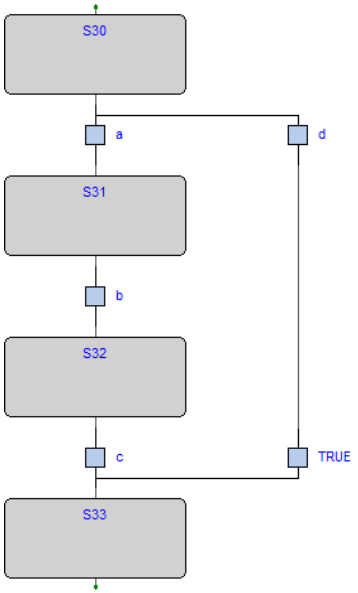
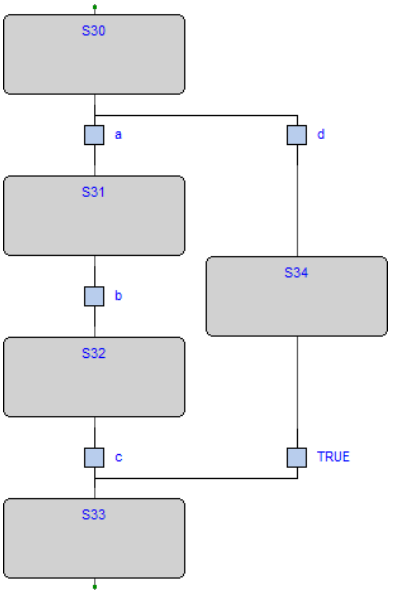
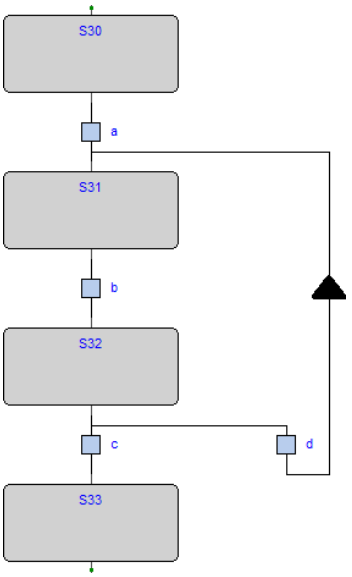
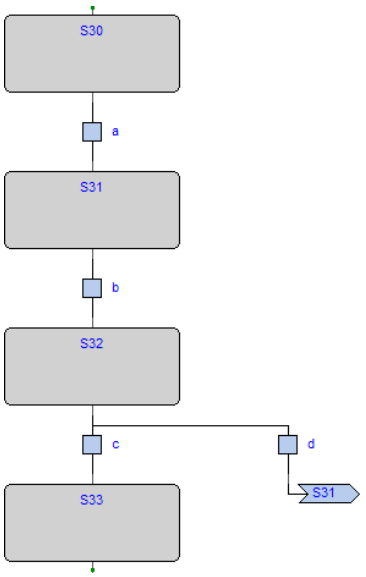
В этой таблице приводится синтаксис и семантика допустимых комбинаций шагов и переходов.

Пример	Правило
	**Обычный переход:** Продвижение с шага `S3` к шагу `S4` происходит только в том случае, если шаг `S3` находится в активном состоянии, а условие перехода `c` имеет значение `TRUE`.

Пример	Правило
	**Расходящийся переход:** Продвижение с шага `S5` к шагу `S6` происходит только в том случае, если шаг `S5` активен и условие перехода `e` имеет значение `TRUE`, а продвижение с шага `S5` к шагу `S8` — только если шаг `S5` активен, условие перехода `f` имеет значение `TRUE`, а `e` — `FALSE`.
	**Сходящийся переход:** Продвижение с шага `S7` к шагу `S10` происходит только в том случае, если шаг `S7` активен и условие перехода `h` имеет значение `TRUE`, а продвижение с шага `S9` к шагу `S10` — только если шаг `S9` активен, а условие перехода `j` имеет значение `TRUE`.

Пример	Правило
	**Одновременный расходящийся переход:** Продвижение с шага `S11` к шагам `S12`, `S14` и т. д. происходит, только если шаг `S11` активен и условие перехода `b`, связанное с общим переходом, имеет значение `TRUE`. После одновременной активации шагов `S12`, `S14` и т. д. изменение каждой последовательности продолжается автономно.
	**Одновременный сходящийся переход:** Продвижение с шагов `S13`, `S15` и т. д. к шагу `S16` происходит только в том случае, если все вышестоящие и соединенные с двойной горизонтальной линией шаги активны, а условие перехода `d`, связанное с общим переходом, имеет значение `TRUE`.

Примеры

Недействительная схема	Эквивалентная допустимая схема	Примечание
		Ожидаемое поведение: продвижение с шага `S30` к шагу `S33` происходит, если значением условия `a` является `FALSE`, а значением условия `d` является `TRUE`. Схема в крайнем левом столбце недействительна, так как условия `d` и `TRUE` связаны напрямую.
		Ожидаемое поведение: продвижение с шага `S32` к шагу `S31` происходит, если значением условия `c` является `FALSE`, а значением условия `d` является `TRUE`. Схема в крайнем левом столбце недействительна, так как направленные связи ведут только вниз. Переходы вверх могут выполняться с помощью блоков прыжка.

Управляющие флаги SFC

Описание

EcoStruxure Machine Expert - HVAC предоставляет управляющие флаги для программ или функциональных блоков SFC.

Чтобы включить эту функцию, см. раздел «Вкладка Code Generation (Генерирование кода)» *(стр. 166)*.

К эти флагам относятся:

- `<имя модуля>_HOLD_SFC` (тип `BOOL`);
- `<имя модуля>_RESET_SFC` (тип `BOOL`),

Где `<имя модуля>` соответствует программному модулю SFC (программе или функциональному блоку).

Например, если программный модуль SFC имеет имя Main, то управляющие флаги именуются `Main_HOLD_SFC` и `Main_RESET_SFC`.

Для каждого действия SFC доступно еще несколько действий, которые также содержатся в программном модуле SFC.

Например, если в программе **Main** содержится действие SFC с именем **Execute**, то управляющими флагами этого действия являются `Main_Execute_HOLD_SFC` и `Main_Execute_RESET_SFC`.

Флаг удержания

В соответствии с основными характеристиками флага `<имя модуля>_HOLD_SFC`:

- Значением по умолчанию является `FALSE`.
- Если устанавливается значение `TRUE`, блок SFC, на который выполняется ссылка (с тем же именем, что и `<имя модуля>`), сохраняется в текущем состоянии (удержание), и код не выполняется.
- При установке флага обратно в состояние `FALSE` выполнение блока SFC восстанавливается с той же точки, в которой было активировано удержание, посредством `<имя модуля>_HOLD_SFC := TRUE`.

Флаг сброса

В соответствии с основными характеристиками флага `<имя модуля>_RESET_SFC`:

- Значением по умолчанию является `FALSE`.
- Если устанавливается значение `TRUE`, блок SFC, на который выполняется ссылка (с тем же именем, что и `<имя модуля>`), возвращается в начальное состояние (выполнение начального действия).
- Это флаг автоматического сброса, т. е. если для него устанавливается значение `TRUE`, то после выполнения действия сброса его собственное состояние получает значение `FALSE`. Совсем не обязательно возвращать `<имя модуля>_RESET_SFC` в состояние `FALSE`.

Видимость флагов

Флаги `<имя модуля>_HOLD_SFC` и `<имя модуля>_RESET_SFC` автоматически генерируются компилятором кода и относятся к локальным переменным программного модуля, на которые ссылаются.

EcoStruxure Machine Expert - HVAC не показывает эти флаги в списке переменных программного модуля; они скрыты, но в любом случае их можно использовать в любом месте кода.

Проверка программного модуля SFC из других программ

Описание

Для управления программным модулем SFC из других программ EcoStruxure Machine Expert - HVAC предоставляет следующие функциональные возможности:

- Компилятор автоматически генерирует флаги `<имя модуля>_RESET_SFC` и `<имя модуля>_HOLD_SFC`.
- Если программный модуль SFC является функциональным блоком, пользователь имеет возможность объявить как `VAR_INPUT` с типом `BOOL` оба флага, имеющие имена управляющих флагов программного модуля SFC.
- Если программный модуль SFC является программой, пользователь имеет возможность объявить как `VAR_GLOBAL` с типом `BOOL` оба флага, имеющие имена управляющих флагов программного модуля SFC.
- В обоих случаях компилятор EcoStruxure Machine Expert - HVAC использует переменные, объявленные как `VAR_INPUT` или `VAR_GLOBAL`, а не автоматически сгенерированные (они не генерируются).

Используя эти методы, пользователь может управлять рабочим состоянием программного модуля SFC из другого программного модуля с помощью переменных `INPUT` программного модуля SFC.

Пример

```
FUNCTION_BLOCK test
   VAR_INPUT
      ...
      test_RESET_SFC : BOOL; (* Флаг управления объявлен в явном виде *)
   END_VAR
   ...
END_FUNCTION_BLOCK
PROGRAM Main
   VAR
      ...
      block : test; (* Экземпляр блока SFC *)
   END_VAR
   ...
   (* Сброс состояния блока SFC *)
   block.test_RESET_SFC := TRUE;
   ...
END_PROGRAM
```

Библиотека макросов SFC

EcoStruxure Machine Expert - HVAC предоставляет библиотеку **SFCControl.pll**, позволяющую управлять состояниями SFC, используя команды вместо установки переменных.

Эта библиотека состоит из макросов, используемых только в языке ST.

Пример использования флагов управления

Ниже приведены примеры использования управляющих флагов для программного модуля SFC с именем Main.

- Удержание (заморозка):
  ```
  Main_HOLD_SFC := TRUE;
  ```
- Перезапуск из состояния удержания:
  ```
  Main_HOLD_SFC := FALSE;
  ```
- Перезапуск из исходного состояния блока SFC в состоянии удержания:
  ```
  Main_RESET_SFC := TRUE;
  Main_HOLD_SFC := FALSE;
  ```
- Сброс в исходное состояние и мгновенный перезапуск блока SFC:
  ```
  Main_RESET_SFC := TRUE; (* Автоматический сброс из компилятора *).
  ```

Раздел 14.7

Языковые расширения EcoStruxure Machine Expert - HVAC

Содержание этого раздела

В этом разделе описаны следующие темы.

Тема	Страница
Обзор	441
Макросы	441
Указатели	442
Инструкция ожидания	443

Обзор

Описание

EcoStruxure Machine Expert - HVAC предлагает несколько расширений стандарта МЭК 61131-3 (ГОСТ Р МЭК 61131-3) с целью дальнейшего обогащения языка и адаптации к различным стилям программирования.

Макросы

Описание

EcoStruxure Machine Expert - HVAC реализует макросы так же, как это делает препроцессор языка программирования C.

Макросы могут создаваться с помощью следующего синтаксиса:

```
MACRO <Имя макроса>
   PAR_MACRO
      <Список параметров>
   END_PAR
   <Тело макроса>
END_MACRO
```

Список параметров может быть пустым, поэтому можно выделить два типа макросов: макросы-объекты, не принимающие параметры, и макросы-функции, принимающие параметры.

Ниже приводится пример определения макроса, принимающего два байта и составляющего 16-битное слово:

```
MACRO MAKEWORD
   PAR_MACRO
      lobyte;
      hibyte;
   END_PAR
   { CODE:ST }
   lobyte + SHL( TO_UINT( hibyte ), 8 )
END_MACRO
```

Когда имя макроса встречается в исходном коде, оно заменяется (вместе с текущим списком параметров, если речь идет о макросах-функциях) на тело макроса. Например, имя макроса `MAKEWORD` вставляется в следующий фрагмент кода на языке «Структурный текст»:

```
w := MAKEWORD( b1, b2 );
```

В этом случае препроцессор макросов преобразует его в:

```
w := b1 + SHL( TO_UINT( b2 ), 8 );
```

Указатели

Описание

Указатели — это специальные переменные, которые выступают в роли ссылки на другую переменную. Значение указателя по сути является адресом переменной, на которую выполняется ссылка. Чтобы получить доступ к данным, хранящимся по указанному адресу, указатели можно разыменовать.

Для объявления указателей используется тот же синтаксис, что и для объявления переменных, где имя типа — это имя типа переменной, на которую ссылается указатель, перед которым ставится знак `@`:

```
VAR
   <имя указателя> : @<имя типа переменной, на которую выполняется ссылка>;
END_VAR
```

Например, объявление указателя на переменную типа REAL будет выглядеть следующим образом:

```
VAR
   px : @REAL;
END_VAR
```

Указатель может быть связан с другим указателем или с адресом. Для получения адреса переменной предусмотрен специальный оператор `ADR`.

```
px := py; (* px и py являются указателями на REAL (переменные типа @REAL) *)
px := ADR( x ) (* x — переменная типа REAL *)
px := ?x (* ? альтернативный вариант записи для ADR *)
```

Оператор @ используется для разыменования указателя, т. е. для доступа к переменной, на которую тот ссылается.

```
px := ADR( x );
@px := 3.141592; (* x присваивается приближенное значение числа Пи *)
pn := ADR( n );
n := @pn + 1; (* n увеличивается на 1 *)
```

Помните, что необдуманное использование указателей является потенциальным источником серьезных ошибок в программах, которые при выполнении могут негативно влиять на состояние контроллера и (или) вашей машины или процесса.

⚠ ВНИМАНИЕ

НЕДОПУСТИМЫЙ УКАЗАТЕЛЬ

- Проверяйте корректность указателей на адреса до их применения.
- Всегда инициализируйте указатели действительным адресом памяти.

Несоблюдение этих указаний может привести к травмам или повреждению оборудования.

Инструкция ожидания

Описание

EcoStruxure Machine Expert - HVAC реализует инструкцию **WAITING**, которую можно использовать в коде ST, как показано в следующем примере:

```
...
WAITING <условие> DO
   <код, который будет выполнен, когда условие станет истинным>
END_WAITING;
...
```

Пока условие не проверено, код продолжает выполняться (не как в цикле, а возвращаясь к вызывающей стороне при каждом выполнении).

Инструкцию **WAITING** можно использовать только в том случае, если активирована соответствующая опция проекта. Для получения дополнительной информации см. раздел «Генерирование кода» *(стр. 166)*.

Глава 15

Справочник по ошибкам

Раздел 15.1

Сообщения об ошибках компиляции

Обзор

Описание

Код ошибки	Краткое описание	Описание
A4097	Объект не найден	Указанный объект (переменная или функциональный блок) не был определен в приложении.
A4098	Неподдерживаемый тип данных	Размер (в битах), запрашиваемый указанным типом данных, не поддерживается целевой системой.
A4099	Пространство для автоматических переменных исчерпано	Общее распределяемое пространство, запрашиваемое всеми локальными переменными, превышает пространство, доступное на целевой системе.
A4100	Пространство для энергонезависимых переменных исчерпано	Общее распределяемое пространство, запрашиваемое всеми локальными энергонезависимыми переменными, превышает пространство, доступное на целевой системе.
A4101	Пространство для битовых переменных исчерпано	Общее распределяемое пространство, запрашиваемое всеми локальными битовыми (логическими) переменными, превышает пространство, доступное на целевой системе.
A4102	Недопустимый ++ в блоке данных	Указанная переменная связана с индексом, который отсутствует в соответствующем блоке данных.
A4103	Блок данных не найден	Указанная переменная связана с блоком данных, который отсутствует (не определен) в целевой системе.
A4104	Пространство для кода израсходовано	Общий размер кода, используемого для программного модуля (программы, функции и функциональные блоки), превышает объем свободного места на целевой системе.
A4105	Недопустимое смещение бита	Указанная переменная связана с индексом бита, который отсутствует в соответствующем блоке данных.
A4106	Запрошена переменная образа	Код ошибки упразднен.
A4107	Целевая функция не найдена	Указанная функция недоступна в целевой системе.
A4108	Базовый объект не найден	Указанный экземпляр относится к функциональному блоку, который не был определен.
A4109	Недопустимый тип базового объекта	Указанная переменная связана с типом данных (включая определение функционального блока), который не определен.
A4110	Недопустимый тип данных	Тип данных, используемый в определении переменной, не существует.
A4111	Недопустимый тип операнда	Тип операнда недопустим для текущего оператора.

Код ошибки	Краткое описание	Описание
A4112	Функциональный блок совместно использует глобальные данные и задействуется большим количеством задач	Указанный функциональный блок вызывается более чем одной задачей, но использует глобальные переменные с образом процесса. По этой причине компилятор не может ссылаться на соответствующую переменную образа для каждого экземпляра функционального блока.
A4113	Ошибка распределения временных переменных	Внутренняя ошибка компилятора.
A4114	Встроенные функции не поддерживают массивы в качестве входных переменных	–
A4115	Слишком много параметров, передаваемых во встроенную функцию	–
A4116	Не удалось выполнить инкрементную сборку, выполните команду полной сборки	–
A4117	Менее 10 % свободного пространства для данных	–
A4118	Менее 10 % свободного пространства для сохраняемых данных	–
A4119	Менее 10 % свободного пространства для битовых данных	–
A4120	Переменная превышает объем, выделенный для блока данных	–
A4121	Элемент не найден	–
A4123	Ошибка доступа к закрытому члену	–
A4129	Не является структурированным типом	–
A4130	Не является экземпляром функционального блока	–
A4131	Несовместимое внешнее объявление	–
A4133	Не является переменной	–
A4134	Индекс превышает размер массива	–
A4135	Недопустимый тип данных индекса	–
A4136	Отсутствует один или несколько индексов	–
A4137	Требуется экземпляр функционального блока	–
A4138	Требуется простая переменная	–
A4139	Слишком много индексов	–
A4140	Не является экземпляром структуры	–
A4141	Не является массивом	–
A4143	Не является указателем	–
A4144	Разыменование двойного указателя не разрешено	–
A4145	Пока не реализовано	–
A4146	Битовый тип данных недопустим	–
A4147	Невозможно рассчитать смещение переменной	–
A4148	Комплексные переменные не могут иметь образ процесса	–

Код ошибки	Краткое описание	Описание
A4149	Невозможно использовать непосредственно представленные переменные с образом процесса в функциональных блоках (не реализовано)	–
A4150	Экземпляр функционального блока недопустим	–
A4151	Структура недопустима	–
A4152	16-битные переменные должны быть выровнены по границе 16 бит	–
A4153	32-битные переменные должны быть выровнены по границе 32 бит	–
A4154	Ошибка выделения памяти под временную строковую переменную, необходимо разбить инструкцию	–
A4155	Пространство для автоматических переменных EXT/AUX исчерпано	–
A4156	Неоднозначное значение перечисления, требуется префикс <enum>#	–
B0001	Блок данных не найден	Указанная переменная связана с блоком данных, который отсутствует (не определен) в целевой системе.
B0002	Ошибка при создании файла	Указанный файл не может быть создан из-за ошибки файловой системы или отсутствия исходного файла.
C0001	Парсер не инициализирован	Внутренняя ошибка компилятора.
C0002	Недопустимый токен	Недопустимое слово для синтаксиса используемого языка.
C0003	Недопустимая спецификация файла	Внутренняя ошибка компилятора.
C0004	Невозможно открыть файл	Указанный файл не может быть открыт из-за ошибки файловой системы или отсутствия исходного файла.
C0005	Ошибка таблицы парсера	Внутренняя ошибка компилятора.
C0006	Парсер не указан	Внутренняя ошибка компилятора.
C0007	Неожиданный конец файла	Указанный файл обрезан или синтаксис неполный.
C0009	Зарезервированное ключевое слово	Указанное слово не может быть использовано для целей объявления, поскольку является ключевым словом языка.
C0010	Недопустимый элемент	Указанное слово не является действительным для синтаксиса данного языка.
C0011	Отменено пользователем	–
C0032	Слишком много параметров в вызове макроса	–
C0033	Недопустимое количество параметров в вызове макроса	–
C0034	Слишком много вложенных вызовов макроса	–
C4097	Недопустимый тип переменной	Указанный тип данных недопустим.
C4098	Недопустимый префикс местоположения	Неправильная строка адреса указанной переменной, отсутствует символ «%».

Код ошибки	Краткое описание	Описание
C4099	Недопустимая спецификация местоположения	Неправильная строка адреса указанной переменной, индикатор типа доступа к данным не I, Q или M.
C4100	Недопустимый тип местоположения	Неправильная строка адреса указанной переменной, индикатор типа данных не X, B, W, D, R или L.
C4101	Недопустимая спецификация индекса местоположения	Неправильная строка адреса указанной переменной, недействительный индекс.
C4102	Дублирование имени переменной	Имя указанной переменной уже используется для другого объекта в проекте.
C4103	Здесь допускается только 0	Компилятор использует только массивы с нулевым индексом.
C4104	Недопустимая размерность массива	Размерность массива указана неверно (например: содержит недопустимые символы, отрицательные числа и т. д.).
C4105	Константа не инициализирована	Каждая константа должна иметь начальное значение.
C4106	Недопустимый размер строки	–
C4107	При инициализации превышается размер строки	–
C4108	Недопустимое повторение при инициализации	–
C4109	Недопустимый тип данных для инициализации	–
C4353	Дублирование метки	Указанная метка уже определена в текущем программном модуле (программе, функции или функциональном блоке).
C4354	Использование константы не допускается	Указанная операция не позволяет использовать константы (обычно это операции хранения или присвоения).
C4355	Адрес явной константы не определен	–
C4356	Превышено максимальное количество индексов	–
C4358	Недопустимая база массива	–
C4359	Недопустимый операнд	–
C4609	Недопустимая двоичная константа	Постоянное значение с префиксом 2# должно содержать только двоичные цифры (0 или 1).
C4610	Недопустимая восьмеричная константа	Постоянное значение с префиксом 8# должно содержать только восьмеричные цифры (в диапазоне от 0 до 7).
C4611	Недопустимая шестнадцатеричная константа	Постоянное значение с префиксом 16# должно содержать только шестнадцатеричные цифры (от 0 до 9 и от A до F).
C4612	Недопустимая десятичная константа	Десятичная константа должна содержать только цифры от 0 до 9, знак + или -, а также символ точки «.» в качестве десятичного разделителя. Или показатель экспоненты e или E.
C4613	Недопустимая временная константа	Постоянное значение с префиксом t# должно содержать только указание времени в десятичном формате записи и единицы времени ms, s или m.
C4614	Недопустимая постоянная строка	–

Код ошибки	Краткое описание	Описание
C4864	Дублирование имени функции	Указанное имя функции уже используется для другого объекта приложения.
C4865	Недопустимый тип функции	Тип данных, возвращаемый указанной функцией, неправильный.
C5120	Дублирование имени программы	Указанное имя программы уже используется для другого объекта приложения.
C5376	Дублирование имени функционального блока	Указанное имя функционального блока уже используется для другого объекта приложения.
C5632	Недопустимая директива компилятора	–
C5633	Недопустимое значение директивы компилятора	–
C5889	Дублирование имени макроса	–
C5890	Дублирование имени параметра макроса	–
C6144	Недопустимое определение ресурса: две или более задачи имеют одинаковый ID	–
C16385	Недопустимое начальное значение	–
C16386	Недопустимое определение инициализации	–
C16387	Недопустимые разделители массива (скобки)	–
C16388	Пустое начальное значение	–
C16389	Пустое начальное значение массива	–
C16390	Недопустимое дублированное начальное значение	–
C16391	Не реализовано	–
C16392	Отсутствующие разделители массива (скобки)	–
C16393	Отсутствующая запятая	–
C16394	Не реализовано	–
C16395	Недопустимая (неполная) строка	–
D12289	Невозможно выделить память под базу данных	Объем памяти, необходимый для базы данных параметров, превышает объем памяти, доступный на целевой системе. Если возможно, удалите неиспользуемые записи параметров, меню и т. д.
D12290	Невозможно выделить память под запись базы данных	Объем памяти, необходимый для базы данных параметров, превышает объем памяти, доступный на целевой системе. Если возможно, удалите неиспользуемые записи параметров, меню и т. д.
D12291	Переменная базы данных не найдена	Внутренняя ошибка компилятора.
D12292	Недопустимое выражение или неверный синтаксис выражения	Выражение базы данных, имеющее указанный результат, некорректно и содержит синтаксические ошибки или недопустимые операторы.

Код ошибки	Краткое описание	Описание
D12293	Недопустимая ссылка на параметр в выражении	Выражение базы данных, которое имеет указанный результат, содержит параметр (в качестве операнда), который не является тем, на который ссылается данное выражение. В выражении могут использоваться только переменные ПЛК (в т. ч. переменные, связанные с параметрами) и значение параметра, который передается в данный момент. Например, выражение pDELTA = DELTA / pRATIO + pOFFSET является правильным, потому что передаваемым параметром выступает DELTA, и это единственное значение параметра, используемое в выражении. Выражение pDELTA = DELTA / pRATIO + OFFSET является некорректным, так как параметр OFFSET, используемый в выражении, не передается.
D12294	Рекурсивное выражение	Выражение базы данных, имеющее указанный результат, вызывает само себя с помощью определенного операнда, содержащего результат текущего выражения.
D12295	Неразрешенная переменная в выражении	Выражение базы данных, имеющее указанный результат, использует операнд, который не определен в проекте ПЛК.
D12296	Неразрешенный результат выражения	Внутренняя ошибка компилятора.
D12297	Недопустимый тип результата для выражения	Параметр, являющийся результатом выражения, имеет недопустимый тип данных (например, перечисляемый) или не определен.
D12298	Недопустимый операнд в выражении	Выражение базы данных, имеющее указанный результат, использует недопустимый операнд.
D12299	Недопустимый тип переменной для выражения	Переменная, являющаяся результатом выражения, имеет недопустимый тип данных (например, перечисляемый) или не определена.
D12300	Ошибка сборщика	Внутренняя ошибка компилятора.
D12301	Невозможно выделить память под код базы данных	Свободное место, необходимое для кода выражения, исчерпано. Необходимо удалить некоторые выражения из базы данных параметра.
D12302	Недопустимая операция в выражении	Выражение базы данных, имеющее указанный результат, использует недопустимый операнд.
F1025	Недопустимая сеть	Указанная сеть FBD или LD содержит ошибку соединения (ошибки обычно обозначаются красными соединениями).
F1026	Неподключенный вывод	Указанный блок (оператор, функция, контакт или катушка) имеет неподключенный вывод.
F1027	Недопустимое соединение (неполное, более одного источника и т. д.)	Внутренняя ошибка компилятора.
F1028	Более одной сети на блок	Указанная сеть содержит несколько сетей блоков и переменных, не связанных между собой.
F1029	Неоднозначное выполнение сети	Компилятор не в состоянии найти однозначный порядок выполнения блоков.
F1030	Ошибка распределения временных переменных	Внутренняя ошибка компилятора.
F1031	Непоследовательная сеть	Указанная сеть не имеет входных или выходных переменных.

Код ошибки	Краткое описание	Описание
F1032	Недопустимый объект подключен к шине питания	–
F1033	Недопустимое использование отрицания вывода (оператор ADR не поддерживает отрицаемый ввод)	–
F1034	Недопустимое использование отрицания вывода (оператор SIZEOF не поддерживает отрицаемый ввод)	–
G0001	Недопустимое число операндов	Количество операндов не соответствует операнду или указанной функции.
G0002	Переменная не определена	Переменная не определена в локальном или глобальном контексте.
G0003	Метка не определена	Метка, указанная для операнда JMP, не определена в текущем программном модуле (программе, функции или функциональном блоке).
G0004	Функциональный блок не определен	Указанный экземпляр относится к функциональному блоку, который не был определен в проекте.
G0005	Ссылка на объект не определена	Указанный экземпляр относится к объекту, не определенному в проекте.
G0006	Использование константы не допускается	Указанная операция не позволяет использовать константы (обычно это операции хранения или присвоения).
G0007	Переполнение буфера кода	Общий размер кода, используемого для программного модуля (программы, функции и функциональные блоки), превышает объем свободного места на целевой системе.
G0008	Ошибка доступа к переменной	Доступ к указанной переменной запрещен. Была предпринята попытка записать переменную, доступную только для чтения, или прочитать переменную, доступную только для записи.
G0009	Программа не найдена	Указанная программа не существует в текущем проекте.
G0010	Программа уже назначена задаче	Указанная программа назначена нескольким задачам целевой системы.
G0011	Невозможно выделить память под буфер кода	На персональном компьютере недостаточно памяти для создания образа кода целевой системы.
G0012	Функция не определена	Указанная функция не существует в текущем проекте.
G0013	Циклическое объявление функциональных блоков	Указанный функциональный блок вызывает сам себя напрямую или с помощью других функций.
G0014	Несовместимое внешнее объявление	Объявление внешней переменной текущего функционального блока не совпадает с определением глобальной переменной, на которую та ссылается (с тем же именем). Обычно это происходит в случае несоответствия типов.
G0015	Расширение аккумулятора	–
G0016	Внешняя переменная не найдена	Внешняя переменная не ссылается ни на одну из глобальных переменных проекта (например, отсутствует глобальная переменная с таким же именем).
G0017	Программа не назначена задаче	Указанная программа не назначена задаче в целевой системе.

Код ошибки	Краткое описание	Описание
G0018	Задача не найдена в ресурсах	Указанная задача не определена в целевой системе.
G0019	Для приложения не определена ни одна задача	Для целевой системы не существует определений задач. Файл определения цели (*.TAR) отсутствует или неполный. Свяжитесь с разработчиком целевой системы.
G0020	Дальние данные разрешены только для операций загрузки/сохранения в блоках PROGRAM	Доступ к большому объему памяти в случае функциональных блоков не предусмотрен — только для программ (код ошибки действителен только для некоторых целевых систем с доступом к данным типа NEAR/FAR).
G0021	Недопустимый тип процессора	Процессор, указанный в файле определения цели (*.TAR), некорректен или не поддерживается компилятором.
G0022	Функциональный блок с переменными образа процесса нельзя использовать в задачах обработки событий	–
G0023	Переменные образа процесса нельзя использовать в задачах обработки событий	–
G0024	Аккумулятор не определен	–
G0025	Недопустимый индекс	–
G0026	Разрешается использовать только постоянный индекс	–
G0027	Недопустимая ссылка на адрес регистра	–
G0028	Менее 10 % свободного пространства для кода	–
G0029	Индекс превышает размер массива	–
G0030	Доступ к массиву как к скаляру — при наличии индекса 0	–
G0031	Количество индексов не соответствует размеру переменных	–
G0032	Многомерные переменные не поддерживаются	–
G0033	Недопустимый тип данных	–
G0034	Недопустимый тип операнда	–
G0035	Ошибка сборщика	–
G0036	Отменено пользователем	–
G0037	Элемент не определен	–
G0038	Циклическое объявление структур	–
G0039	Циклическое объявление определений типов	–
G0040	Неразрешенное определение типа	–
G0041	Превышение размерности при определении типов	–
G0042	Невозможно выделить память под внутренние данные компилятора	–

Код ошибки	Краткое описание	Описание
G0043	ВНУТРЕННЯЯ ОШИБКА ГЕНЕРАТОРА КОДА	–
G0044	Данные типа REAL не поддерживаются	–
G0045	Данные типа LONG REAL не поддерживаются	–
G0046	Данные типа LONG не поддерживаются	–
G0047	Операция не реализована	–
G0048	Недопустимый оператор	–
G0049	Недопустимое значение оператора	–
G0050	Незакрытые скобки	–
G0051	Преобразование данных	–
G0052	Пока не реализовано	–
G0053	Недопустимый тип данных индекса	–
G0054	Отрицание без условия	–
G0055	Операция не может быть проведена над логическими операндами	–
G0056	Отрицание операнда с типом, отличным от логического	–
G0057	Требуется операнд логического типа	–
G0058	Параметр с плавающей точкой недопустим	–
G0059	Расширение операнда	–
G0060	Деление на ноль	–
G0061	Недопустимое сравнение	–
G0062	Необходимо создать экземпляр функционального блока	–
G0063	Не допускается использование строкового операнда	–
G0064	Операция не может быть проведена над указателями	–
G0065	Область назначения может быть слишком мала для хранения текущего результата	–
G0066	Невозможно использовать функциональный блок, содержащий внешние переменные с образом процесса, более чем в одной задаче	–
G0067	Невозможно загрузить адрес явной константы	–
G0068	Запись значения типа REAL в целочисленную переменную	–
G0069	Невозможно использовать комплексные переменные в функциях (не реализовано)	–
G0070	Ошибка при использовании типов со знаком и без знака	–
G0071	Ошибка приведения типов, возможная потеря данных	–

Код ошибки	Краткое описание	Описание
G0072	Неявное приведение логических значений к целочисленному типу	–
G0073	Неявное приведение логических значений к значениям типа REAL	–
G0074	Неявное приведение целочисленных значений к логическому типу	–
G0075	Неявное приведение целочисленных значений к логическому типу	–
G0076	Неявное приведение значений типа REAL к логическому типу	–
G0077	Неявное приведение значений типа REAL к целочисленному типу	–
G0078	Для проведения арифметических операций требуются числовые операнды	–
G0079	Для проведения побитовых логических операций требуются битовые строки / целочисленные операнды	–
G0080	Для проведения операций сравнения требуются элементарные (т. е. не определяемые пользователем) операнды	–
G0081	Невозможно получить адрес битовой переменной	–
G0082	Запись значения со знаком в переменную без знака	–
G0083	Запись значения без знака в переменную со знаком	–
G0084	Неявное преобразование числа с бычной точностью в число с двойной точностью	–
G0085	Неявное преобразование числа с двойной точностью в число с обычной точностью	–
G0086	Расширение параметров функции	–
G0087	Приведение к тому же типу не действует	–
G0088	Неправильное количество параметров функции	–
G0089	Встроенная целевая функция не найдена	–
G0090	Рекурсивное объявление типа	–
G0091	Неверное начальное значение, ошибка при использовании типов со знаком и без знака	–
G0092	Неверное начальное значение. Ошибка приведения типов, возможная потеря данных	–
G0093	Строка будет обрезана	–

Код ошибки	Краткое описание	Описание
G0094	Несоответствие типа начального значения	–
G0095	Некорректное начальное значение	–
G0096	Объект начального значения не найден	–
G0097	Недопустимое присваивание указателю	–
G0513	Недопустимый оператор	Указанный оператор нельзя использовать для данной операции.
G0514	Операция не реализована	Указанный оператор не поддерживается целевой системой.
G0515	Данные типа REAL не поддерживаются	Используемая целевая система не поддерживает операции с плавающей точкой.
G0516	Область назначения может быть слишком мала для хранения текущего результата	Переменная назначения операции сохранения/присваивания имеет тип данных, размер которого меньше, чем тип данных аккумуляторов. В ходе операции могут быть потеряны данные. Например, если аккумулятор содержит 340, а операнд назначения имеет тип SINT, операция присваивания приведет к потере данных. Если операция находится под контролем программиста, для устранения предупреждающего сообщения можно использовать соответствующую функцию приведения типа (TO_SINT, TO_INT, TO_DINT и т. д.).
G0517	Данные типа LONG не поддерживаются	Используемая целевая система не поддерживает операции с данными типа LONG.
G0518	Расширение аккумулятора	Переменная назначения операции сохранения/присвоения имеет тип данных, размер которого больше, чем тип данных аккумуляторов. Операция расширения выполнена компилятором автоматически. Чтобы устранить это сообщение, используйте соответствующую функцию приведения типов (TO_SINT, TO_INT, TO_DINT и пр.).
G0519	Ошибка сборщика	Внутренняя ошибка компилятора.
G0520	Отрицание допускается только для логических значений	Модификатор N, используемый для некоторых операторов языка IL (LDN, STN, ANDN и пр.), не может использоваться с операторами, имеющими тип, отличный от логического.
G0521	Операция разрешена для значений логического типа	Указанный оператор языка IL (обычно S или R) не может быть использован, если аккумулятор имеет тип, отличный от BOOL.
G0522	Инструкция имеет постоянный результат	Указанная операция имеет постоянный результат (например, умножение на 0, операция AND с FALSE).
G0523	Инструкция NOP	Указанная операция не влияет на значение аккумулятора (умножение на 1, операция AND с TRUE).
G0524	Незакрытые скобки	Количество открывающих скобок не совпадает с количеством закрывающих скобок в указанном блоке кода.
G0525	Операция не может быть проведена над логическими операндами	Указанная операция не может быть выполнена над логическими операндами (например, арифметические операции).
G0526	Невозможно выполнить деление по модулю со значениями типа LONG	Текущая целевая система не позволяет выполнять операцию деления по модулю с типами данных LONG.

Код ошибки	Краткое описание	Описание
G0527	Деление на 0	Указанная операция деления имеет в качестве делителя постоянное значение 0.
G0528	Отрицание без условия	В указанной операции (JMP или RET) используется модификатор отрицания N без модификатора условного вычисления C. Использование JMPCN вместо JMPN или RETCN вместо RETN.
G0529	Начальное значение не определено	Внутренняя ошибка компилятора.
G0530	Недопустимое начальное значение	Указано неправильное начальное значение переменной.
G0531	Недопустимый тип аккумулятора	Аккумулятор имеет тип данных, недопустимый для указанной операции (например, оператор MUX с аккумулятором REAL).
G0532	Внутренняя ошибка генератора кода	Внутренняя ошибка компилятора.
G0533	Недопустимое значение оператора	Оператор имеет значение, недопустимое для указанной операции (например, SHL с постоянным значением больше 32).
G0534	Аккумулятор не определен	Операция выполняется без предварительно загруженного в аккумулятор значения.
G0535	Недопустимый индекс	Постоянное значение индекса, используемое в указанном выражении, слишком велико для размерности массива. См. строку объявления массива.
G0536	Разрешается использовать только постоянный индекс	Использование переменной в качестве индекса для указанного массива не поддерживается компилятором. Эта ошибка обычно возникает при работе с массивами логических (битовых) значений.
G0537	Индексирование логических констант не допускается	Использование переменной в качестве индекса для указанного массива не поддерживается компилятором. Эта ошибка обычно возникает при работе с массивами логических (битовых) значений.
G0538	Возврат из программ запрещен	Оператор RET не разрешен в блоках PROGRAM.
G0539	Необходимо создать экземпляр функционального блока	Функциональный блок невозможно вызвать напрямую с помощью инструкции CAL. Перед использованием необходимо создать его экземпляр — например, должна быть переменная с типом данных, соответствующим функциональному блоку.
G0540	Операция не разрешена для значений типа REAL	Указанная операция не может быть выполнена для типов данных REAL. Инструкции такого рода являются логическими и побитовыми операциями.
G0541	Приведение типа аккумулятора	Это сообщение информирует о том, что тип данных аккумулятора был автоматически приведен компилятором. Эта операция обычно выполняется, когда аккумулятор и операнд, используемый в арифметической операции, имеют разные типы данных.
G0542	Аккумулятор типа REAL должен быть перезагружен	Некоторые целевые реализации с программной эмуляцией плавающей точки требуют, чтобы каждой операции сохранения предшествовала новая операция загрузки или арифметическая последовательность.

Код ошибки	Краткое описание	Описание
G0543	Аккумулятор типа REAL не сохранен	Некоторые реализации для целевых систем с программной эмуляцией плавающей точки требуют, чтобы после загрузки стека плавающей точки тот выгружался в конце арифметической последовательности.
G0544	Данные типа LONG REAL не поддерживаются	Тип данных LREAL не поддерживается компилятором.
G0769	Недопустимый оператор	Указанный оператор нельзя использовать для данной операции.
G0770	Операция не реализована	Указанный оператор не поддерживается текущей целевой системой.
G0771	Ошибка сборщика	Внутренняя ошибка компилятора.
G0772	Данные типа LONG REAL не поддерживаются	Тип данных LREAL не поддерживается компилятором.
G0773	Данные типа LONG не поддерживаются	Тип данных LINT не поддерживается компилятором.
G0774	Отрицание параметра с типом, отличным от логического	Модификатор отрицания N не может быть использован в операциях с типами данных, отличными от логического.
G0775	Операция не может быть проведена над логическими операндами	Указанная операция не может быть выполнена над логическими операндами (например, арифметические операции).
G0776	Расширение аккумулятора	Переменная назначения операции сохранения/присвоения имеет тип данных, размер которого больше, чем тип данных аккумуляторов. Операция расширения выполнена компилятором автоматически. Чтобы устранить это сообщение, используйте соответствующую функцию приведения типов (TO_SINT, TO_INT, TO_DINT и пр.).
G0777	Аккумулятор не определен	Операция выполняется без предварительно загруженного в аккумулятор значения.
G0778	Область назначения может быть слишком мала для хранения текущего результата	Переменная назначения операции сохранения/присваивания имеет тип данных, размер которого меньше, чем тип данных аккумуляторов. В ходе операции могут быть потеряны данные. Например, если аккумулятор содержит 340, а операнд назначения имеет тип SINT, операция присваивания приведет к потере данных. Если операция находится под контролем программиста, для устранения предупреждающего сообщения можно использовать соответствующую функцию приведения типа (TO_SINT, TO_INT, TO_DINT и т. д.).
G0779	Деление на ноль	Указанная операция деления имеет в качестве делителя постоянное значение 0.
G0780	Операция разрешена только для параметров типа REAL	Указанная операция не может быть выполнена для типов данных REAL. Инструкции такого рода являются логическими и побитовыми операциями.
G0781	Недопустимое сравнение	Указанная операция сравнения выполняется над неоднородными типами данных.
G0782	Отрицание без условия	В указанной операции (JMP или RET) используется модификатор отрицания N без модификатора условного вычисления C. Использование JMPCN вместо JMPN или RETCN вместо RETN.

Код ошибки	Краткое описание	Описание
G0783	Требуется параметр логического типа	Указанный оператор языка IL (обычно S или R) не может быть использован, если аккумулятор имеет тип, отличный от BOOL.
G0784	Расширение операнда	Тип данных операнда расширен до типа данных аккумулятора. После этого операция выполняется. Расширение операнда происходит всякий раз, когда тип данных операнда меньше типа данных аккумулятора.
G0785	Не поддерживает аккумулятор данных с плавающей точкой	Аккумулятор имеет тип данных REAL, и его не разрешается использовать для указанной операции (обычно операции MUX).
G0786	Не поддерживает аккумулятор данных логического типа	Аккумулятор имеет логический тип данных, и его не разрешается использовать для указанной операции (например, операции MUX).
G0787	Сравнение беззнакового и знакового типа	Указанная операция сравнения выполняется с использованием операторов, имеющих знаковый и беззнаковый типы данных. Возможен нежелательный или неконтролируемый результат.
G0788	Недопустимое преобразование	Внутренняя ошибка компилятора.
G0789	Преобразование может привести к потере или повреждению данных	Код ошибки не используется.
G0790	Недопустимое отрицание параметра типа REAL	Код ошибки не используется.
G0791	Запись значения типа REAL в целочисленную переменную/параметр	Параметр, передаваемый в функцию, имеет тип REAL вместо целочисленного типа данных, как того требует определение входных переменных функции.
G0792	Запись целочисленного значения типа в переменную/параметр типа REAL	Параметр, передаваемый в функцию, имеет целочисленный тип вместо типа данных REAL, как того требует определение входных переменных функции.
G0793	Запись значения со знаком в переменную/параметр без знака	Операция присваивания выполняется над переменной беззнакового типа данных, но тип данных аккумулятора имеет знаковый тип данных. Возможно получение нежелательного результата.
G0794	Запись беззнакового значения типа в переменную/параметр со знаком	Операция присваивания выполняется над переменной беззнакового типа данных, но тип данных аккумулятора имеет знаковый тип данных. Возможно получение нежелательного результата.
G0795	Незакрытые скобки	Количество открывающих скобок не совпадает с количеством закрывающих скобок в указанном блоке кода.
G0796	Ошибка при расширении параметров	Внутренняя ошибка компилятора.
G0797	Недопустимый индекс	Постоянное значение индекса, используемое в указанном выражении, слишком велико для размерности массива. См. строку объявления массива.
G0798	Использование индекса логического типа для доступа к элементу массива	Указанный доступ к массиву неверен, поскольку используемая индексная переменная имеет логический тип данных.
G0799	Возврат из программ запрещен	Оператор RET не разрешен в блоках PROGRAM.
G0800	Требуется аккумулятор логического типа	Указанный оператор SEL требует, чтобы аккумулятор имел логический тип данных.
G0801	Операторы имеют несовпадающие типы данных	Выбор, выполняемый операторами MUX и SEL, должен осуществляться между элементами, имеющими однородные типы данных.

Код ошибки	Краткое описание	Описание
G0802	Необходимо создать экземпляр функционального блока	Функциональный блок невозможно вызвать напрямую с помощью инструкции CAL. Перед использованием необходимо создать его экземпляр — например, должна быть переменная с типом данных, соответствующим функциональному блоку.
G1537	Использование индекса логического типа для доступа к элементу массива	–
G1538	Не поддерживает аккумулятор данных логического типа	–
G1539	Не поддерживает аккумулятор данных с плавающей точкой	–
G1540	Ошибка при расширении одного или нескольких операндов	–
G1541	Запись значения со знаком в переменную без знака	–
G1542	Запись значения без знака в переменную со знаком	–
G1543	Запись значения типа REAL в целочисленную переменную	–
G1544	Запись целочисленного значения типа в переменную типа REAL	–
G1545	Преобразование строки в число	–
G1546	Преобразование числа в строку	–
G1547	Стек FPU заполнен	–
G1548	Стек FPU пуст	–
G1549	Ошибка размера стека FPU	–
G1550	Недопустимый доступ к переменной через функцию	–
G1551	Недопустимая ссылка на адрес переменной, доступной через функцию	–
G1552	Недопустимый доступ через функцию	–
G1553	Две переменные с одинаковым дескриптором	–
G1554	Неверный индекс для переменной, доступной через функцию	–
G1555	Недопустимая инструкция с непустым стеком FPU	–
G1556	Попытка сохранить строчный результат функции в переменной	–
G8193	Определение неизвестного типа данных	–
G8194	Определение типа содержит превышающие размерности массива	–
G8195	Циклическое определение типа данных	–

Код ошибки	Краткое описание	Описание
G8196	Двойные указатели не поддерживаются	–
G8197	Отсутствуют перечисляемые элементы	–
G8199	Недопустимая или неопределенная константа инициализации	–
G10241	Слишком много инициализаторов для переменной	–
G10242	Слишком мало инициализаторов для переменной	–
G10243	Константа без начальных значений	–
P2048	Невозможно открыть файл параметров	Исходный файл параметров (с расширением PPC) невозможно открыть, поскольку он отсутствует или заблокирован файловой системой компьютера.
P2049	Файл таблицы символов не создан	Файл аллокации символов (с расширением SYM) невозможно записать из-за установленной защиты от записи на диск или нехватки места на диске.
P2050	Невозможно создать файл параметров	Файл параметров (с расширением PAR) невозможно записать из-за установленной защиты от записи на диск или нехватки места на диске.
P2051	Невозможно создать директорию	Невозможно создать директорию для нового проекта. Проблема возникает при установленной защите от записи на диск или когда новая директория, указанная для проекта, расположена более чем на один уровень глубже, чем существующая директория на диске. Компилятор создает только один новый уровень директории (с именем проекта), начиная с уже имеющейся директории.
P2052	Невозможно открыть исходный проект	Исходный проект, указанный при создании нового проекта, не существует, является неполным или заблокирован файловой системой.
P2053	Ошибка при сохранении проекта	Новый проект невозможно сохранить из-за установленной защиты от записи на диск, несуществующей целевой директории или блокировки файловой системы.
P2054	Общая ошибка при работе с файлами	Ошибка неспецифического характера, произошедшая во время операций с файлами.
P2055	Невозможно скопировать файл	Указанный файл невозможно скопировать из-за отсутствия исходного файла, установленной защиты от записи на диск или из-за того, что файл назначения уже существует и защищен.
P2056	Невозможно сохранить файл	Указанный файл невозможно сохранить из-за установленной защиты от записи на диск или из-за того, что файл назначения уже существует и защищен.
P2057	Объект уже существует в проекте	Указанный объект (переменная, функция, функциональный блок или программа) содержится в последней загруженной библиотеке, но в текущем проекте уже есть другой объект с таким же именем.
P2058	Невозможно открыть файл библиотеки	Указанный файл библиотеки не закрывается или не открывается из-за блокировки со стороны файловой системы.
P2059	Файл листинга не создан	–
P2060	Невозможно создать бинарный файл модуля **PROGRAMMING**	–

Код ошибки	Краткое описание	Описание
P2061	Невозможно открыть проект-шаблон	–
P2062	Поддержка процессора недоступна	–
P2063	Менее 10 % свободного пространства для кода	–
P2064	Менее 10 % свободного пространства для данных	–
P2065	Менее 10 % свободного пространства для сохраняемых данных	–
P2066	Менее 10 % свободного пространства для битовых данных	–
P2067	Задача не найдена в ресурсах	–
P2068	Для приложения не определена ни одна задача	–
P2069	Проект выполнен в старом формате PPJ, он будет сохранен в современном формате PPJX	–
P2070	Невозможно открыть вспомогательный исходный файл	–
P2071	Невозможно прочитать файл	–
P2072	Имя приложения содержит более десяти символов: будут загружены только первые десять символов	–
P2073	Доступный для загрузки файл исходного кода не защищен паролем	–
P2074	Доступный для загрузки бинарный файл приложения ПЛК не создан	–
P2075	Менее 10 % свободного пространства для данных EXT/AUX	–
P2076	Копия этой библиотеки, принадлежащая проекту, отсутствовала и была заменена новой копией библиотеки (по исходному пути)	–
P2077	Невозможно загрузить библиотеку! Копия этой библиотеки, принадлежащая проекту, отсутствовала, а исходный путь к библиотеке недействителен: библиотека была исключена	–
P2078	Файл экспорта переменных ПЛК не создан	–
P2079	Пакет с отладочными символами (для последующей загрузки на целевое устройство) не создан	–
P2080	Пакет с исходным кодом (для последующей загрузки на целевое устройство) не создан	–
P2081	Недопустимое определение задачи	–
P2083	Недопустимый или непоследовательный период выполнения задачи	–
P2084	Нерабочая ссылка на библиотеку	–
S1281	Общая ошибка ST	–

Код ошибки	Краткое описание	Описание
S1282	Слишком много вложенных выражений	–
S1283	Отсутствует итерация для выхода	–
S1284	Отсутствует END_IF	–
S1285	Недопустимая инструкция ST	–
S1286	Недопустимое присваивание	–
S1287	Отсутствует символ «;»	–
S1288	Недопустимое выражение	–
S1289	Недопустимое выражение или отсутствует DO	–
S1290	Отсутствует END_WHILE	–
S1291	Отсутствует END_FOR	–
S1292	Отсутствует END_REPEAT	–
S1293	Недопустимое выражение или отсутствует THEN	–
S1294	Недопустимое выражение или отсутствует TO	–
S1295	Недопустимое выражение или отсутствует BY	–
S1296	Недопустимая инструкция или отсутствует UNTIL	–
S1297	Недопустимое присваивание, требуется :=	–
S1298	Недопустимый адрес выражения	–
S1299	Недопустимый размер выражения	–
S1300	Возвращаемое значение функции игнорируется	–
S1301	Передача недопустимого параметра	–
S1302	Параметр функции не определен	–
S1303	Бесполезное выражение	–
S1304	Незакрытые скобки	–
S1305	Неизвестная функция	–
S1306	Недопустимое указание одного или нескольких параметров функции	–
S1307	Параметр функции не существует	–
S1308	Множественное присваивание не допускается (в соответствии с МЭК 61131-3 [ГОСТ Р МЭК 61131-3])	–
S1309	Переполнение буфера препроцессора ST	–
S1310	Вызов функциональным блоком экземпляра, не являющегося функциональным блоком	–
S1311	Отсутствует END_WAITING	–
S1312	Синтаксическая ошибка	–
S1537	Общая ошибка SFC	–

Код ошибки	Краткое описание	Описание
S1538	Отсутствует начальный шаг	–
S1539	Отсутствует подключение к выходу	–
S1540	Выходной вывод должен быть подключен к переходу	–
S1541	Каждый выходной вывод перехода должен быть подключен к блоку шага/прыжка	–
S1542	Требуется переход	–
S1543	Требуется шаг или прыжок	–
S1544	Не удалось найти связанный код программы	–
S1545	Не удалось найти код условия	–
S1546	Переход неизвестного типа	–
S1547	Недопустимый пункт назначения	–
S1548	Дублирование действия, одно и то же действие SFC не может использоваться более чем в одном шаге	–
T8193	Тайм-аут связи	Не удалось установить связь с целевой системой, поскольку система не отвечает. Наиболее распространенными причинами этой проблемы являются неправильное подключение кабеля, неверный адрес целевого устройства в настройках связи, неверные настройки параметров связи (например, скорости передачи данных) или неисправная целевая система.
T8194	Несовместимая версия целевого устройства	Код ошибки не используется.
T8195	Недопустимый файл с кодом	Файл образа целевой системы (с расширением IMG) недействителен или поврежден. Попробуйте загрузить и создать новую версию файла образа с помощью пункта меню Communication (Связь) → Upload image file (Загрузить новый файл образа).
T8196	Недопустимый индекс блока данных	Файл образа (с расширением IMG) содержит блок данных, индекс которого превышает максимальный индекс, поддерживаемый целевой системой. Попробуйте загрузить и создать новую версию файла образа с помощью пункта меню Communication (Связь) → Upload image file (Загрузить новый файл образа). Если проблема с любым числовым типом сохраняется, обратитесь к разработчику целевой системы.
T8197	Неверный адрес с информацией о целевом устройстве	Внутренняя ошибка компилятора.
T8198	Ошибка при стирании флеш-памяти	Целевая система не смогла завершить процедуру стирания флеш-памяти. За подробностями обращайтесь к производителю целевой системы.
T8199	Ошибка при записи кода	Целевая система не смогла завершить процедуру записи во флеш-память. За подробностями обращайтесь к производителю целевой системы.

Код ошибки	Краткое описание	Описание
T8200	Устройство связи недоступно	Компилятор пытался установить связь с целевой системой, однако канал связи недоступен. Если проблема сохраняется и существуют другие приложения, которые взаимодействуют с целевой системой, отключите связь в других приложениях и повторите попытку.
T8201	Недопустимый индекс функции	Внутренняя ошибка компилятора.
T8202	Неверный адрес с информацией о базе данных	Адрес области памяти для базы данных параметра в целевой системе неправильный или недействительный. Попробуйте загрузить и создать новую версию файла образа с помощью пункта меню Communication (Связь) → Upload image file (Загрузить файл образа).
T8203	Неверная информация о целевом устройстве	–
T8204	Требуется пересборка	–
T8205	Недопустимая задача	–
T8206	Ошибка протокола связи на уровне приложения: среда выполнения ПЛК не смогла распознать полученную команду	–
T8209	Отсутствует место для исходного файла на целевом устройстве	–
T8210	Ошибка при загрузке исходного кода с целевого устройства	–
T8211	Отсутствует место для отладочных символов на целевом устройстве	–
T8212	Ошибка чтения из памяти	–
T8213	Ошибка записи в память	–
T8214	Недостаточно места на целевом устройстве для бинарного файла приложения ПЛК	–

Глоссарий

!

%

Согласно стандарту МЭК, символ % — это префикс для идентификации адресов внутренней памяти логического контроллера, предназначенной для хранения значений переменных программы, констант, ввода/вывода и т. д.

%I

Согласно стандарту МЭК, %I представляет собой входной бит (например, объект языка типа «цифровой вход»).

%IW

Согласно стандарту МЭК, %IW представляет собой регистр входного слова (например, объект языка типа «аналоговый вход»).

%MW

Согласно стандарту МЭК, %MW представляет собой регистр слова в памяти (например, объект языка типа «слово памяти»).

%Q

Согласно стандарту МЭК, %Q представляет собой выходной бит (например, объект языка типа «цифровой выход»).

A

ASCII

(*American Standard Code for Information Interchange*) Протокол для представления буквенно-числовых символов (букв, цифр, некоторых графических и управляющих символов).

B

BOOL

(*булев*) Базовый тип данных в программировании. Переменная `BOOL` может принимать одно из следующих значений: 0 (`FALSE`), 1 (`TRUE`). Извлеченный из слова бит имеет тип `BOOL`; например, `%MW10.4` — это пятый бит слова памяти с номером 10.

D

DINT

(*целое число двойной длины*) Кодируется в 32-битном формате.

DWORD

(*двойное слово*) Кодируется в 32-битном формате.

E

EDS

(*Electronic Data Sheet, электронное техническое описание*) Файл описания устройства полевой шины, который может содержать такие свойства устройства, как параметры и настройки.

EEPROM

(*Electrically Erasable Programmable Read-Only Memory, электрически стираемая программируемая постоянная память*) Тип энергонезависимой памяти для хранения важных данных после выключения электропитания.

Ethernet

Технология физического и канального уровня для создания локальных сетей, также известная как IEEE 802.3.

F

FBD

(*function block diagram, функциональные блок-схемы*) Один из пяти языков для построения логики или взаимодействия с системами управления, поддерживаемых стандартом МЭК 61131-3 (ГОСТ Р МЭК 61131-3). Функциональные блок-схемы — это графически ориентированный язык программирования. Он работает со списком сетей, где каждая сеть содержит графическую структуру из блоков и соединительных линий, представляющую собой логическое или арифметическое выражение, вызов функционального блока, прыжок или инструкцию возврата.

H

HEX

(*hexadecimal, шестнадцатеричное значение*)

I

I/O

(*input/output, вход/выход*)

IL

(*instruction list, список инструкций*) Программа, написанная на этом языке, состоит из ряда текстовых инструкций, последовательно выполняемых контроллером. Каждая инструкция содержит номер строки, код инструкции и операнд (см. МЭК 61131-3 [ГОСТ Р МЭК 61131-3]).

IP

(*Internet protocol, интернет-протокол*) Относится к семейству протоколов TCP/IP. Отвечает за адресацию устройств в сети Интернет, маршрутизацию исходящих сообщений и распознавание входящих сообщений.

M

Modbus

Протокол, обеспечивающий связь между несколькими устройствами, подключенными к одной и той же сети.

P

PDO

(*process data object, объект данных процесса*) Неподтвержденное широковещательное сообщение или сообщение, пересылаемое устройством-производителем на устройство-потребитель в сети на базе протокола CAN. Передаваемый устройством-производителем PDO имеет определенный идентификатор, который соответствует принимаемому PDO устройств-потребителей.

R

RJ45

Стандартный тип восьмиконтактного разъема для сетевых кабелей, предназначенных для Ethernet.

RPDO

(*receive process data object, принимаемый объект данных процесса*) Неподтвержденное широковещательное сообщение или сообщение, пересылаемое устройством-производителем на устройство-потребитель в сети на базе протокола CAN. Передаваемый устройством-производителем PDO имеет определенный идентификатор, который соответствует принимаемому PDO устройств-потребителей.

RS-232

Стандартный тип трехпроводной шины для последовательной связи (также называется EIA RS-232C или V.24).

RS-485

Стандартный тип двухпроводной шины для последовательной связи (также называется EIA RS-485).

RTU

(*remote terminal unit, удаленный терминал*) Устройство, обеспечивающее взаимодействие объектов физического мира с распределенной системой управления или системой SCADA путем передачи телеметрических данных в систему и (или) изменения состояния подключенных объектов на основе управляющих сообщений, полученных от системы.

S

SFC

(*sequential function chart, последовательная функциональная схема*) Язык, состоящий из шагов с назначенными действиями, переходов с назначенными логическими условиями и направленных связей между шагами и переходами. (Стандарт SFC определен в стандарте МЭК 60848 [ГОСТ IEC 60848]. Удовлетворяет требованиям МЭК 61131-3 [ГОСТ Р МЭК 61131-3].)

SINT

(*signed integer, целое число со знаком*) 15-битное значение плюс знак.

ST

(*structured text, структурный текст*) Язык, использующий сложные операторы и вложенные инструкции (например, циклы, выполнение при соблюдении условия или функции). ST соответствует стандарту МЭК 61131-3 (ГОСТ Р МЭК 61131-3).

T

TCP

(*transmission control protocol, протокол управления передачей данных*) Протокол соединений транспортного уровня, обеспечивающий одновременную двунаправленную передачу данных. TCP является частью набора протоколов TCP/IP.

TPDO

(*transmit process data object, отправляемый объект данных процесса*) Неподтвержденное широковещательное сообщение или сообщение, пересылаемое устройством-производителем на устройство-потребитель в сети на базе протокола CAN. Передаваемый устройством-производителем PDO имеет определенный идентификатор, который соответствует принимаемому PDO устройств-потребителей.

U

UDINT

(*unsigned double integer, целое число двойной длины без знака*) Кодируется 32 битами.

UINT

(*unsigned integer, целое число без знака*) Кодируется 16 битами.

W

WORD

Тип данных, кодируемый в 16-битном формате.

А

Аналоговый вход

Преобразует полученные уровни напряжения или силы тока в числовые значения. Преобразованные значения можно сохранять и обрабатывать в логическом контроллере.

Аналоговый выход

Преобразует числовые значения в логическом контроллере и передает на выход пропорциональные уровни напряжения или тока.

Б

Байт

Тип данных, который кодируется в 8-битном формате, от 00 до FF (HEX).

В

Ведущий/ведомый

Модель однонаправленного управления устройствами в сети по принципу «ведущий/ведомый».

К

Клеммная колодка

Компонент, устанавливаемый в электронный модуль и обеспечивающий электрическое соединение между контроллером и полевыми устройствами.

Контроллер

Автоматизирует промышленные процессы (также называется программируемым логическим контроллером или просто программируемым контроллером).

Конфигурирование

Размещение и подключение аппаратных компонентов в системе и настройка параметров аппаратного и программного обеспечения, определяющих рабочие характеристики системы.

М

Массив

(**ARRAY**) Массивом называют упорядоченное расположение объектов данных одного типа в виде таблицы, определенной в памяти логического контроллера. Используется следующий синтаксис: `ARRAY [<измерение>] OF <тип>`

Пример 1: `ARRAY [1..2] OF BOOL` — одномерная таблица с двумя элементами типа `BOOL`.

Пример 2: `ARRAY [1..10, 1..20] OF INT` — двумерная таблица с 10 x 20 элементами типа `INT`.

Машина

Состоит из нескольких *функций* и (или) *аппаратных узлов*.

Модуль расширения ввода/вывода

Цифровой или аналоговый модуль, который добавляет дополнительные входы/выходы к базовому контроллеру.

МЭК 61131-3 (ГОСТ Р МЭК 61131-3)

Третья, заключительная часть стандарта МЭК, применяемого к оборудованию промышленной автоматизации. Этот стандарт посвящен языкам программирования контроллеров и определяет требования к двум графическим и двум текстовым языкам программирования. Графическими языками программирования являются язык релейных схем и функциональных блок-схем. Текстовыми языками программирования являются «Структурный текст» (ST) и список инструкций (IL).

П

ПЛК

(*программируемый логический контроллер*) Промышленный компьютер, используемый для автоматизации производственных, промышленных и других электромеханических процессов. ПЛК отличаются от обычных компьютеров тем, что они оснащены несколькими массивами входов и выходов, а также имеют более высокие характеристики устойчивости, в частности к ударам, вибрации, воздействию температуры и электрических помех.

Пользовательская функция

Позволяет создавать собственные функции с одним или несколькими входными параметрами, локальными переменными и возвращаемым значением. После этого пользовательская функция может вызываться в операционных блоках. Пользовательская функция хранится как часть проекта и загружается в логический контроллер в составе приложения.

Приложение

Программа, включая данные конфигурации, символы и документацию.

Приложение-загрузчик

(*загрузчик*) Бинарный файл, содержащий приложение. Обычно он хранится в памяти контроллера и позволяет тому загружать приложение, созданное пользователем.

Присвоенная переменная

Переменная считается присвоенной, если известно ее местоположение в памяти логического контроллера.

Например, переменная `Water_pressure` будет считаться присвоенной вследствие установления связи с ячейкой памяти `%MW102`.

Программа

Компонент приложения, состоящий из скомпилированного исходного кода, который можно установить в память логического контроллера.

Программный модуль

Объявление переменной в исходном коде и соответствующий набор команд. Программные модули облегчают повторное использование программ, функций и функциональных блоков. После объявления программные модули становятся доступными друг для друга.

Протокол

Соглашение или стандартное определение, которое контролирует или обеспечивает соединение, связь и передачу данных между двумя вычислительными системами и устройствами.

С

Светодиод

Индикатор, который светится при воздействии электрического заряда низкого уровня.

Символ

Последовательность, состоящая максимум из 32 буквенно-числовых символов, при этом первый символ является буквой. Позволяет персонализировать объект контроллера для упрощения поддержки приложения.

Строка

Переменная, представляющая собой последовательность символов ASCII.

У

Узел

Адресуемое устройство в сети обмена данными.

Условный элемент

Позволяет реализовать условия в программе в автономном режиме.

Ф

Функция

Блок программы, который имеет один вход и возвращает один результат. Однако в отличие от функционального блока он вызывается непосредственно по своему имени (а не через экземпляр), не имеет постоянного состояния между вызовами и может использоваться в качестве операнда в других выражениях программирования.

Примеры: логические операторы (AND), вычисления, приведение (BYTE_TO_INT).

Функциональный блок

Блок программы, который имеет один или более входов и возвращает один или более результатов на выходе. Функциональные блоки вызываются через экземпляр (копию функционального блока с определенным именем и переменными), при этом каждый экземпляр имеет постоянное состояние (выходы и внутренние переменные) между вызовами.

Примеры: таймеры, счетчики.

Функциональный блок

(*function block, функциональный блок*) Удобный механизм программирования, позволяющий объединять в группу программные инструкции для выполнения определенного и стандартизированного действия, например управления скоростью, интервалом или подсчетом. Функциональный блок может включать в себя данные конфигурации, набор внутренних или внешних рабочих параметров и, как правило, один или более входов и выходов данных.

Функциональные блок-схемы

Один из пяти языков для построения логики или взаимодействия с системами управления, поддерживаемых стандартом МЭК 61131-3 (ГОСТ Р МЭК 61131-3). Функциональные блок-схемы — это графически ориентированный язык программирования. Он работает со списком сетей, где каждая сеть содержит графическую структуру из блоков и соединительных линий, представляющую собой логическое или арифметическое выражение, вызов функционального блока, прыжок или инструкцию возврата.

Ц

Цифровой вход/выход

Отдельное подключение цепи в электронном модуле, которое непосредственно соответствует биту таблицы данных. Бит таблицы данных хранит значение сигнала на схеме входов/выходов. Он предоставляет логике управления цифровой доступ к значениям входов/выходов.

Я

Язык релейных схем

Графическое представление инструкций программы контроллера с символами для контактов, катушек и блоков в виде ряда ступенек, последовательно выполняемых контроллером (см. МЭК 61131-3 [ГОСТ Р МЭК 61131-3]).

Язык списка инструкций

Программа, написанная на языке списка инструкций, состоит из ряда текстовых инструкций, последовательно выполняемых контроллером. Каждая инструкция содержит номер строки, код инструкции и операнд (см. МЭК 61131-3 [ГОСТ Р МЭК 61131-3]).

Указатель

I

IL
- закладка, *229*
- редактор, *227*

F

FBD
- редактор, *230*
- создать, *231*

L

LD
- вставить катушку, *242*
- вставить контакт, *240*
- редактор, *236*
- Свойства катушки/контакта, *244*
- создать, *237*

S

SFC
- вставить элемент, *255*
- редактор, *254*
- соединить элементы, *255*
- создать, *254*

ST
- редактор, *251*
- создать, *252*

Б

Блок
- информация, *235*, *246*
- свойства, *235*, *245*

Базовый блок пользовательского рабочего пространства, *221*
- включить, *221*
- операция, *222*
- элементы, *223*

Библиотека
- включить, *175*
- импортировать из, *178*
- обновить, *180*
- отменить импорт из, *179*
- удалить, *176*
- экспортировать, *177*

В

Ветви
- вставить, *250*

Выражение
- вставить, *248*

Г

Глобальная переменная
- редактировать, *194*

Д

Данные, не относящиеся к программе, *33*

Действие
- присвоить шагу, *256*

З

Задача
- назначить программу во время создания, *184*
- конфигурирование, *201*
- обзор, *199*
- связать с программой, *200*
- управление программой, *200*

Закладка, *253*
- IL, *229*

К

Катушка
вставить, *242*
Комментарии
вставить, *249*
Контакт
вставить, *240*

М

Меню
вид, *76*, *149*
инструменты, *154*
окно, *45*, *154*
онлайн, *77*, *151*
отладка, *152*
переменные, *153*
программа, *93*
проект, *77*, *150*
ресурсы, видимость, *94*
разработчик, *77*
редактировать, *148*
справка, *78*, *154*
установка, *93*
файл, *75*, *147*
целевое устройство, *95*
Минимальные требования к системе, *25*

О

Окно инструментов, *38*
управление, *44*
Окно контрольных значений
добавить элемент, *296*
обновить, *299*
список контрольных значений, *301*
формат данных, *300*
удалить элемент, *299*
Окно триггера, *326*, *327*
графический, *347*, *348*, *350*, *351*, *353*, *354*, *354*, *355*, *356*, *359*, *361*, *363*, *364*, *365*, *367*, *369*
текстовый, *328*, *330*, *331*, *333*, *334*, *335*, *336*, *338*, *340*, *341*, *343*, *343*, *345*
Оператор, *386*, *391*, *393*, *400*, *418*, *420*, *420*, *421*
Определение типа
редактировать, *205*
создать, *203*
удалить, *205*

П

Панель инструментов, *79*, *327*, *351*
FBD, *158*
LD, *160*
SFC, *159*
главная, *155*
Кнопки, *155*
конфигурирование, *79*
отладка, *157*
проект, *36*, *156*
сеть, *161*
Переход
условие, *258*
условный код, *259*
Переменная
вставить, *247*
выборка, *309*
глобальная, *188*
копировать, *267*
локальная, *195*
редактировать, *264*
редактор, *263*
создать, *264*
сортировать, *267*
удалить, *267*
Поддиапазон
создать, *209*
удалить, *210*
редактировать, *210*
Приложение
определение, *33*
Программа
компиляция, *155*
определение, *33*
связать с задачей, *200*
управление внутри задачи, *200*
Программный модуль
редактор, *186*

Проект
загрузить, *65*
закрыть, *59*
меню, *77*
найти в, *217*
окно, содержимое, *182*
определение, *33*
открыть, *59*
панель инструментов, *36*, *156*
печатать, *56*
пользовательское рабочее пространство, *221*
распределить, *60*
редактировать, *59*
создание, *33*
создать, *55*
сохранить, *57*
сохранить как, *57*
Прыжок, *261*

С

Сеть
вставить блок, *233*, *243*
добавить/удалить, *231*, *238*
метка, *232*, *239*
подключить блок, *233*
редактировать, *235*, *244*, *261*
Список контрольных значений, *301*
автосохранение, *302*
Структура
создать, *206*
удалить, *207*
редактировать, *206*

Р

Рабочее пространство
миграция, *221*
пользовательское, *221*
Разработка программ, этапы, *34*
Регистрация ПО EcoStruxure Machine Expert - HVAC, *26*
Редактировать
ST, *252*
определение типа, *205*
перечисление, *208*
поддиапазон, *210*
сеть, *235*, *244*, *261*
структура, *206*
функция, *228*, *252*
Редактор
IL, *227*
FBD, *230*
LD, *236*
PLC, *226*
POU, *186*
SFC, *254*
ST, *251*
добавление переменной, *306*, *307*, *308*
глобальная переменная, *194*
графика, *47*
переменная, *263*
текст, *47*
Режимы работы
моделирование, *39*
онлайн, *39*
офлайн, *39*

Т

Требования к системе, *25*

Ф

Функция, *382*, *386*, *392*, *394*, *398*, *399*, *401*, *407*, *418*, *423*
редактировать, *228*, *252*

Э

Элемент перечисления
редактировать, *208*
создать, *207*
удалить, *208*