

Программируемые контроллеры SIMATIC S7

1-й уровень профессиональной подготовки

S7- PROF1

**Учебник
(раздаточный материал)**



Предисловие

Данный учебник предназначен для слушателей курса S7-PROF1, а также может использоваться в качестве справочного материала для специалистов по разработке и обслуживанию систем управления на базе контроллеров SIMATIC S7-300/400.

Содержание учебника соответствует базовому уровню подготовки специалистов по работе с контроллерами SIMATIC S7.

Учебник содержит разделы по конфигурированию и обслуживанию систем SIMATIC S7-300/400, а также основам программирования на языке STEP7.

Большое количество иллюстративного материала позволит пользователю успешно освоить даже самые сложные темы.

При подготовке материалов использовалась версия STEP 7 Professional v5.5.

Автор курса: к.т.н. Альтерман И.З.

Об авторе:

Автор имеет 10-летний опыт преподавательской деятельности, работал старшим преподавателем в РГАТА. Автор 17 изобретений и 12 научных статей, в 1995 г. в Санкт-Петербургском государственном университете аэрокосмического приборостроения защитил кандидатскую диссертацию по системам управления.

С 1996 по 2000 г. работал в ООО «СИМЕНС» г. Москва мастер-тренером по системам автоматизации SIMATIC, провел более 130 учебных курсов.

Более 4 лет сотрудничал с компанией Siemens VAI ведущим консультантом. Успешно работал в российских предприятиях разработчиком и руководителем проектов по АСУТП.

С 2008 г. проводит авторские курсы по системам SIMATIC.

В 2013 г. основал учебный центр «СИМАТИК».

© Альтерман И.З., 2008-2013
igoraltreman@hotmail.com

Без полученного разрешения копирование, распространение и использование содержания данного документа или его части в коммерческих целях запрещается. Нарушители понесут ответственность за причиненные убытки.

Содержание

| | |
|---|----|
| Структура системы управления SIMATIC S7 | 7 |
| SIMATIC S7-300 Основные характеристики | 7 |
| Конструкция S7-300..... | 8 |
| Функциональные особенности | 9 |
| Набор поддерживаемых функций | 9 |
| Модификации систем S7-300 | 10 |
| Система ввода-вывода..... | 11 |
| SIMATIC S7-400. Обзор | 12 |
| Система ввода-вывода S7-400..... | 13 |
| Функциональные особенности S7-400 | 14 |
| Типы CPU систем S7-400 | 14 |
| SIMATIC S7-400H | 15 |
| Пример построения системы автоматизации..... | 15 |
| Подготовка SIMATIC S7 к запуску программы | 16 |
| Инструкция для монтажа S7-300 | 16 |
| Инструкция для монтажа S7-400 | 16 |
| Адресация модулей | 16 |
| Инструкция для ввода в эксплуатацию SIMATIC S7 | 18 |
| Сброс CPU | 18 |
| Выполнение старта | 18 |
| Режим RUN | 18 |
| Визуальный контроль индикаторов в S7-300 | 19 |
| Подключение программатора к CPU | 19 |
| Установление online связи | 19 |
| Выбор типа интерфейса для online соединения | 20 |
| Проверка соединения через окно "AccessibleNodes"..... | 20 |
| Тестирование сигналов ввода вывода..... | 21 |
| Загрузка программы в CPU | 22 |
| Основные определения | 22 |
| Время цикла..... | 22 |
| Время реакции..... | 22 |
| Управление проектом в SIMATIC Manager | 23 |
| Структура проекта | 23 |
| Настройки проекта | 23 |
| Вставка новой станции | 24 |
| Вставка папки "S7 Program"..... | 24 |
| Вставка блока | 24 |
| On-line проект | 25 |
| Выполнение online функций | 25 |
| Операции над блоками..... | 26 |
| Конфигурирование и параметрирование SIMATIC S7 | 27 |
| Конфигурирование | 27 |
| Параметрирование станции | 28 |
| Параметрирование CPU | 28 |
| Параметрирование модулей ввода/вывода..... | 30 |
| Сохранение и загрузка заданной конфигурации | 31 |
| Чтение текущей конфигурации и параметров из CPU..... | 32 |
| Диагностика аппаратуры | 32 |

| | |
|---|----|
| Структура памяти CPU | 33 |
| Адресное пространство CPU | 34 |
| Адресация ячеек системной памяти | 34 |
| Основы программирования на языке STEP7 | 35 |
| Принцип выполнения программы в S7-CPU. Время контроля цикла | 35 |
| Типы блоков | 35 |
| Типы программ | 36 |
| Глубина вложения | 36 |
| Вызовы блоков | 37 |
| Прерывание программы | 37 |
| Редактирование блоков | 38 |
| Запуск редактора | 38 |
| Раздел кода | 38 |
| Программные элементы | 38 |
| Таблица объявлений | 39 |
| Окно Details | 39 |
| Выбор языка программирования | 39 |
| Программирование в LAD | 40 |
| Программирование в FBD | 40 |
| Программирование в STL | 40 |
| Сохранение блока | 40 |
| Загрузка блоков в CPU | 40 |
| Открытие Online блока | 40 |
| Отладка программы | 41 |
| Изменение программы | 41 |
| Режимы вставки и замены | 41 |
| Двоичные операции | 42 |
| Операции: И, ИЛИ (AND, OR) | 42 |
| Исключающее ИЛИ (Exclusive OR) | 42 |
| Результат логической операции | 43 |
| Операции присвоения, установки и сброса | 43 |
| Первичный опрос (First check) | 43 |
| Правильный подход в программировании | 43 |
| Команда NOT | 43 |
| Триггер | 44 |
| Коннектор | 45 |
| Функции оценки (выделения) фронта RLO | 45 |
| Пример: делитель частоты | 45 |
| Загрузка и передача данных | 46 |
| Таймеры | 47 |
| Задание временных параметров таймера | 47 |
| Пример использования таймера (ON Delay) | 48 |
| Генераторы на таймерах | 48 |
| Таймеры: битовые инструкции | 49 |
| IEC-таймеры | 49 |
| Счетчики | 49 |
| IEC-счетчики | 50 |
| Форматы чисел | 50 |

| | |
|--|----|
| Функции сравнения..... | 51 |
| Арифметические функции | 51 |
| Функции преобразования..... | 52 |
| Операции цифровой логики..... | 53 |
| Операции сдвига..... | 53 |
| Обработка аналоговых сигналов..... | 54 |
| Адресация аналоговых модулей | 54 |
| Кодирование аналоговых величин | 54 |
| Масштабирование..... | 54 |
| Измерение температуры | 55 |
| Время преобразования..... | 55 |
| Символьная адресация..... | 56 |
| Абсолютная адресация | 56 |
| Символьная адресация | 56 |
| Символьная таблица | 56 |
| Структура таблицы..... | 56 |
| Функции Export/Import | 57 |
| Редактирование символов в редакторе..... | 57 |
| Символьная информация в редакторе | 57 |
| Установка адресных приоритетов | 58 |
| Приоритет имен | 58 |
| Приоритет адресов..... | 58 |
| Тестирование и отладка программы..... | 59 |
| Утилита "Monitor and Modifies Variables" | 59 |
| Управление переменными в режиме статуса программы..... | 60 |
| Функция "Go To Location" | 60 |
| Поиск и устранение ошибок..... | 61 |
| Концепция диагностики в SIMATIC S7 | 61 |
| Диагностика с использованием стеков I, V, L | 62 |
| Поиск логических ошибок | 62 |
| Перекрестные ссылки | 63 |
| Таблица используемых ячеек (план использования) | 63 |
| Структура программы | 64 |
| Блоки ошибок..... | 65 |
| Вывод сообщений(функция CPU Messages) | 65 |
| Документирование программы | 66 |
| Архивирование | 66 |
| Деархивирование | 66 |
| Как сохранить проект целиком в MMC | 67 |
| Чтение проекта из MMC..... | 68 |
| Заключение | 68 |

Структура системы управления SIMATIC S7

SIMATIC S7-300 – это модульный программируемый контроллер, предназначенный для построения систем автоматизации низкой и средней степени сложности.

Основные характеристики

Контроллеры SIMATIC S7-300 имеют модульную конструкцию и могут включать в свой состав:

Модуль центрального процессора (CPU). В зависимости от степени сложности решаемой задачи в контроллерах могут быть использованы различные типы центральных процессоров (20 типов), отличающихся производительностью, объемом памяти, наличием или отсутствием встроенных входов-выходов и специальных функций, количеством и видом встроенных коммуникационных интерфейсов и т.д.

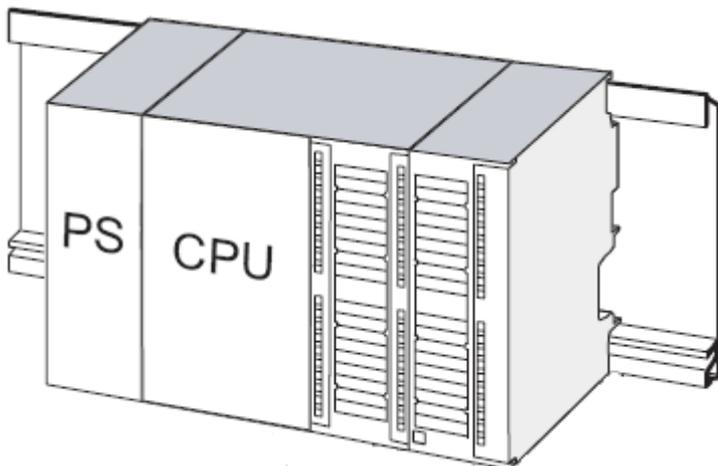
Модули блоков питания (PS), обеспечивающие возможность питания контроллера от сети переменного тока напряжением 120/230В или от источника постоянного тока напряжением 24/48/60/110В.

Сигнальные модули (SM), предназначенные для ввода-вывода дискретных и аналоговых сигналов с различными электрическими и временными параметрами.

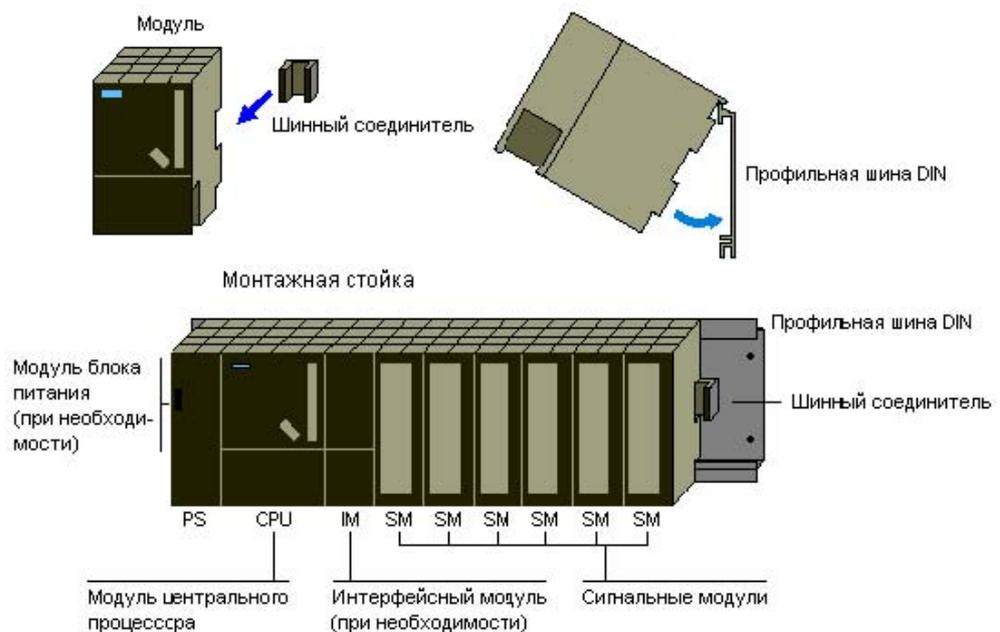
Коммуникационные процессоры (CP) для подключения к сетям PROFIBUS, Industrial Ethernet, AS-Interface или организации связи по PtP (point to point) интерфейсу.

Функциональные модули (FM), способные самостоятельно решать задачи автоматического регулирования, позиционирования, обработки сигналов.

Интерфейсные модули (IM), обеспечивающие возможность подключения к стойке с CPU стоек расширения ввода-вывода. Контроллеры SIMATIC S7-300 позволяют использовать в своем составе до 32 сигнальных и функциональных модулей, а также коммуникационных процессоров, распределенных по 4 монтажным стойкам. Все модули работают с естественным охлаждением.



Конструкция S7-300



Конструкция контроллера отличается высокой гибкостью и удобством обслуживания:

- Все модули легко устанавливаются на профильную рейку SIMATIC и фиксируются в рабочем положении винтом
- Во все модули (кроме модулей блоков питания) встроены участки внутренней шины контроллера
- Соединение этих участков выполняется шинными соединителями, устанавливаемыми на тыльной стороне корпуса
- Наличие фронтальных соединителей, позволяющих производить замену модулей без демонтажа внешних соединений и упрощающих выполнение операций подключения внешних цепей модулей
- Подключение внешних цепей через фронтальные соединители с контактами под винт или контактами-защелками. Механическое кодирование фронтальных соединителей, исключающее возможность возникновения ошибок при замене модулей
- Применение модульных и гибких соединителей SIMATIC TOP Connect, существенно упрощающих монтаж шкафов управления
- Единая для всех модулей глубина установки. Все кабели располагаются в монтажных каналах модулей и закрываются защитными дверцами
- Произвольный порядок размещения модулей в монтажных стойках. Фиксированные места должны занимать только блоки питания, центральные процессоры и интерфейсные модули.

Функциональные особенности

- Удобная для обслуживания конструкция и работа с естественным охлаждением.
- Свободное наращивание возможностей при модернизации системы.
- Встроенный переключатель режимов работы.
- Встроенный MPI интерфейс со скоростью передачи данных 187.5Кбит/с. Может быть использован для построения простейшей сети с подключением до 16 центральных процессоров и передачей глобальных данных.
- Загружаемая память в виде микро карты памяти NVFlash-EEPROM (MMC) емкостью до 8Мбайт: сохранение основной программы, данных конфигурации, всего проекта, также любых файлов, хранимых на внешних носителях.
- Необслуживаемое сохранение резервной копии данных: при перебоях в питании в микро карту памяти записываются состояния флагов, таймеров, счетчиков и содержимое блоков данных (работа без буферной батареи).
- Парольная защита: обеспечивает защиту программы от несанкционированного доступа, попыток копирования и модификации программы.
- Диагностический буфер: в буфере сохраняется 100 последних сообщений об отказах и прерываниях. Содержимое буфера используется для анализа причин, вызвавших остановку центрального процессора.
- Часы реального времени: все диагностические сообщения могут снабжаться отметками даты и времени.
- Встроенные коммуникационные функции: PG/OP функции связи, стандартные и расширенные (только сервер) S7-функции связи.
- Дополнительный встроенный интерфейс ведущего/ ведомого устройства PROFIBUSDP (в CPU 31xC-2 DP) со скоростью обмена данными до 12 Мбит/с.
- Возможность использования распределенных структур ввода-вывода.



Набор поддерживаемых функций

Контроллеры SIMATIC S7-300 оснащены широким набором функций, позволяющих в максимальной степени упростить процесс разработки программы, ее отладки, снизить затраты на обслуживание контроллера в процессе его эксплуатации:

- Высокое быстродействие и поддержка математики с плавающей запятой, позволяющие выполнять эффективную обработку данных.
- Удобный интерфейс для настройки параметров: для настройки параметров всех модулей используется единый набор инструментальных средств с общим интерфейсом.
- Человеко-машинный интерфейс. Функции обслуживания человеко-машинного интерфейса встроены в операционную систему контроллера. Эти функции позволяют существенно упростить программирование.
- Диагностические функции, встроенные в операционную систему контроллера. С их помощью осуществляется непрерывный контроль функционирования системы, и выявляются все возникающие отказы. Все диагностические сообщения с отметками даты и времени накапливаются в кольцевом буфере для последующего анализа.

Модификации систем S7-300

SIMATIC S7-300 - стандартная серия



- Модульный программируемый контроллер (CPU 312, CPU 314, CPU 315-2 DP, CPU 315-2 PN/DP, CPU 317-2 DP, CPU 317-2 PN/DP, CPU 319-3 PN/DP)
- Широкий спектр модулей с различными вычислительными функциями

SIMATIC S7-300C – компактная серия



- Модульный программируемый контроллер (CPU 312C, CPU 313C, CPU 313C-2 PtP, CPU 313C-2 DP, CPU 314C-2 PtP, CPU 314C-2 DP)
- Наличие встроенных входов и выходов (дискретных, аналоговых), а также встроенных функций (функции быстрого счета, измерения частоты, позиционирования и ПИД-регулирования)

SIMATIC S7-300F



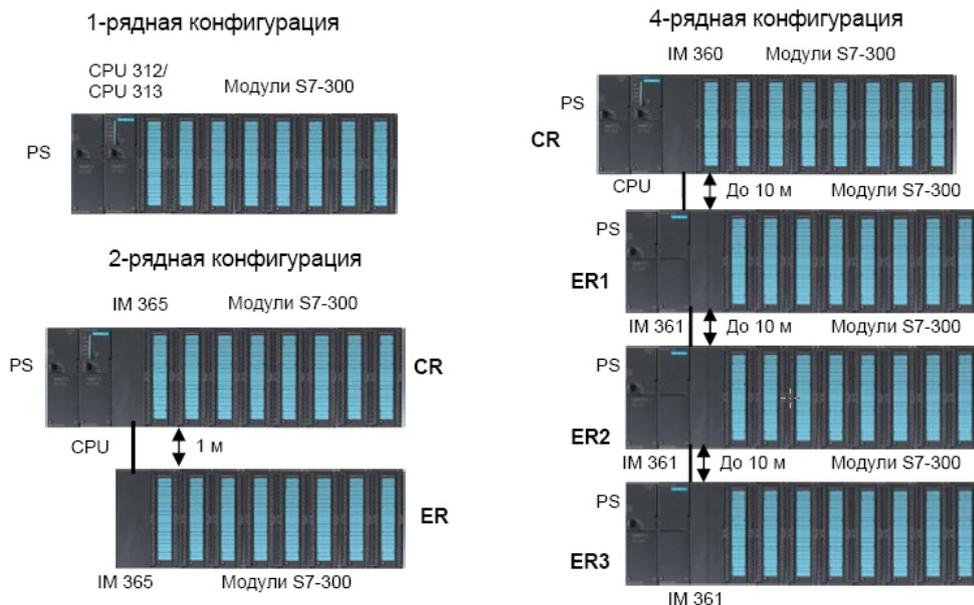
- Программируемый контроллер для построения распределенных систем автоматики безопасности (CPU 315F-2 DP, CPU 315F-2 PN/DP, CPU 317F-2 DP, CPU 317F-2 PN/DP, CPU 319F-3 PN/DP)
- Реализация стандартных функций управления с функциями автоматики безопасности в одной системе

Система ввода-вывода

Система ввода-вывода программируемого контроллера S7-300 может включать в свой состав две части: систему **локального** и систему **распределенного** ввода-вывода. Система локального ввода-вывода образуется модулями, устанавливаемыми непосредственно в монтажные стойки контроллера. Система распределенного ввода-вывода включает в свой состав станции распределенного ввода-вывода и приборы полевого уровня, подключаемые к контроллеру через сети PROFIBUS-DP или AS-Interface.

Системы локального ввода-вывода программируемых контроллеров S7-300 с CPU 313C/ CPU 314 или более мощными процессорами могут включать в свой состав до 32 сигнальных, функциональных и коммуникационных модулей. Все модули устанавливаются в монтажные стойки контроллера, функции которых выполняют профильные шины S7-300.

- В состав системы может входить одна базовая (CR) и до трех стоек расширения (ER). В каждой стойке может размещаться до 8 сигнальных, функциональных и коммуникационных модулей. В стойке CR устанавливается центральный процессор.
- Соединение стоек осуществляется с помощью интерфейсных модулей. Каждая стойка снабжается собственным интерфейсным модулем. Применение интерфейсных модулей IM 365 позволяет подключать к базовой стойке одну стойку расширения, удаленную на расстояние не более 1 м. Интерфейсные модули IM 360 и IM 361 позволяют подключать к базовой стойке до 3 стоек расширения. Расстояние между стойками в этом случае может достигать 10 м.



SIMATIC S7-400 – это модульный программируемый контроллер, предназначенный для построения систем автоматизации средней и высокой степени сложности.

Обзор

S7-400 имеет модульную конструкцию. Он может комплектоваться широким спектром модулей, устанавливаемых в монтажных стойках в любом порядке. Система включает в свой состав:

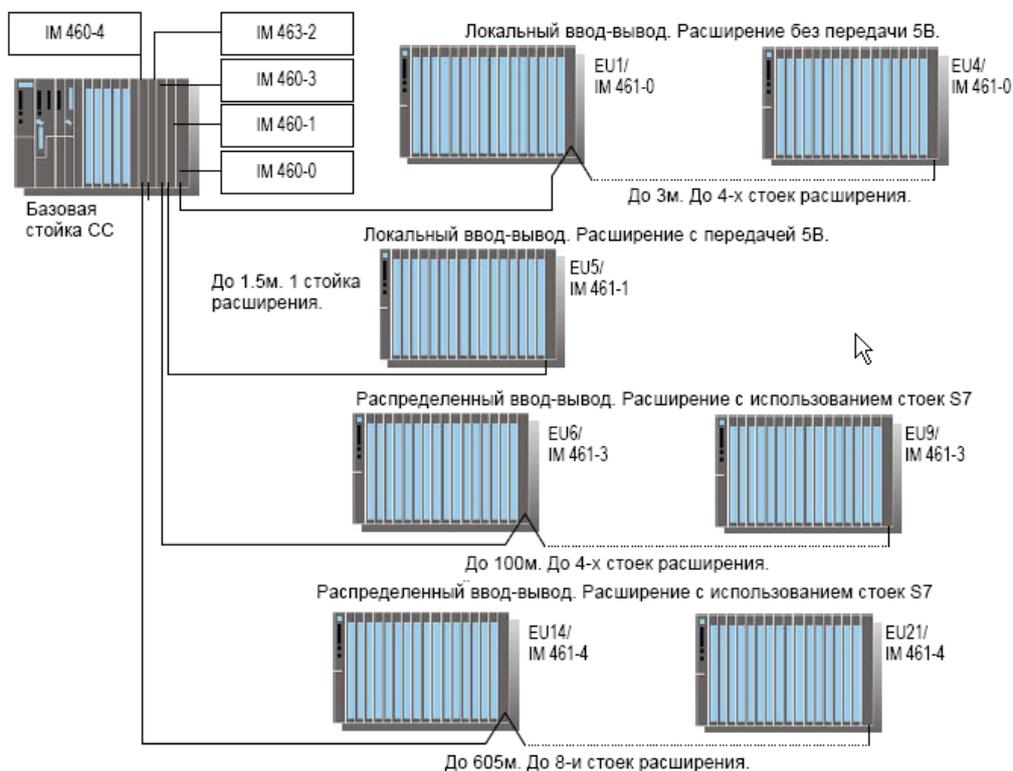
- Модули блоков питания (PS): используются для подключения SIMATIC S7-400 к источникам питания =24/ 48/ 60/120/ 230 В или ~120/ 230 В.
- Модули центральных процессоров (CPU): в составе контроллера могут использоваться центральные процессоры различной производительности. Все центральные процессоры оснащены встроенными интерфейсами MPI и PROFIBUS DP, некоторые модели – встроенным интерфейсом PROFINET. При необходимости, в базовом блоке контроллера может быть использовано до 4 центральных процессоров.
- Сигнальные модули (SM): для ввода-вывода дискретных и аналоговых сигналов.
- Коммуникационные модули (CP): для организации последовательной передачи данных через PtP интерфейс, построения систем распределенного ввода-вывода на основе PROFIBUSDP и PROFINET IO, обмена данными через промышленные сети PROFIBUS, PROFINET и Industrial Ethernet, а также через Internet.
- Функциональные модули (FM): для решения типовых задач управления (скоростной счет, позиционирование, автоматическое регулирование и т.д.)
- Интерфейсные модули (IM): для построения систем локального и распределенного ввода-вывода.
- Монтажные стойки: являются конструктивной основой контроллера и позволяют размещать от 4 до 18 модулей контроллера. Во все монтажные стойки встроена параллельная шина (P-шина) для скоростного обмена данными с сигнальными и функциональными модулями. Все стойки, за исключением ER1 и ER2 имеют последовательную коммуникационную шину (K-шину) для скоростного обмена большими объемами данных с функциональными модулями и коммуникационными процессорами.



Система ввода-вывода

При построении систем локального ввода-вывода на основе монтажных стоек S7-400 должны выполняться следующие правила:

- Система ввода-вывода содержит один базовый блок и несколько стоек расширения.
- Базовым блоком является монтажная стойка, в которой установлен центральный процессор контроллера. Каждый базовый блок может содержать не более 6 передающих интерфейсных модулей (IM).
- Через передающие интерфейсные модули к одному базовому блоку можно подключить до 21 стойки расширения.
- В каждой стойке расширения устанавливается приемный интерфейсный модуль.
- Максимальное расстояние между базовым блоком и стойкой расширения равно 100 м.
- Все модули блоков питания устанавливаются в крайние левые разъемы соответствующих монтажных стоек базового блока и стоек расширения.

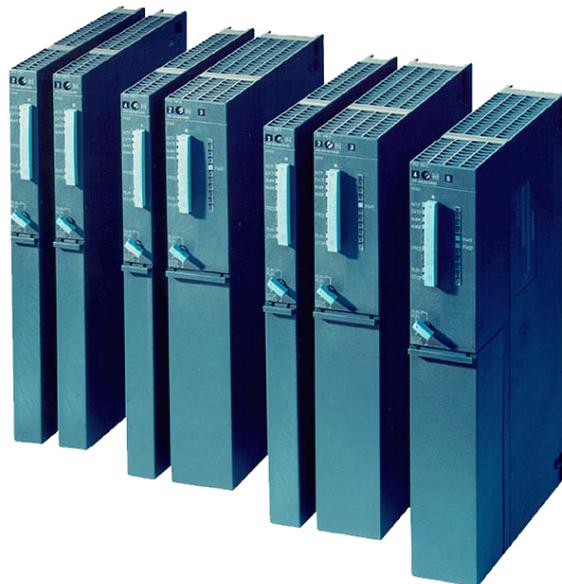


Функциональные особенности

Центральные процессоры S7-400 характеризуются следующими показателями:

- Большие объемы рабочей памяти: от 288 Кбайт до 30 Мбайт (в CPU 417-4).
- Встроенная загружаемая память не менее 512 Кбайт (RAM), расширяемая с помощью карты памяти до 64 Мбайт. Использование карты памяти (RAM или Flash-EEPROM) является обязательным.
- Параллельный доступ к памяти программ и данных, существенно повышающий производительность центрального процессора.
- Высокое быстродействие. Время выполнения:
 - логической операции с битами, операции со словами, арифметической операции с фиксированной точкой – от 75 нс в CPU 412 до 18 нс в CPU 417-4;
 - арифметической операции с плавающей запятой – от 225 нс в CPU 412-1 до 54 нс в CPU 417-4.
- Выбор режимов работы с помощью переключателя замкового типа. Переключение возможно только с помощью ключа.
- Запись программы и данных в карту памяти Flash-EEPROM через встроенный в центральный процессор интерфейс.
- Широкие коммуникационные возможности, поддержка от 32 до 64 активных коммуникационных соединений.
- Наличие буфера диагностических сообщений, сохраняющего 120 последних сообщений об ошибках, отказах и прерываниях. Возможность считывания и анализа диагностической информации.

Типы CPU систем S7-400



- CPU 412-1, CPU 412-2: для построения небольших систем управления и решения задач средней степени сложности.
- CPU 414-2, CPU 414-3: для построения систем управления средней степени сложности.
- CPU 416-2, CPU 416-3: для построения сложных систем автоматического управления со сложными алгоритмами обработки информации и интенсивным сетевым обменом данными.
- CPU 417-4: для построения наиболее мощных систем автоматического управления.



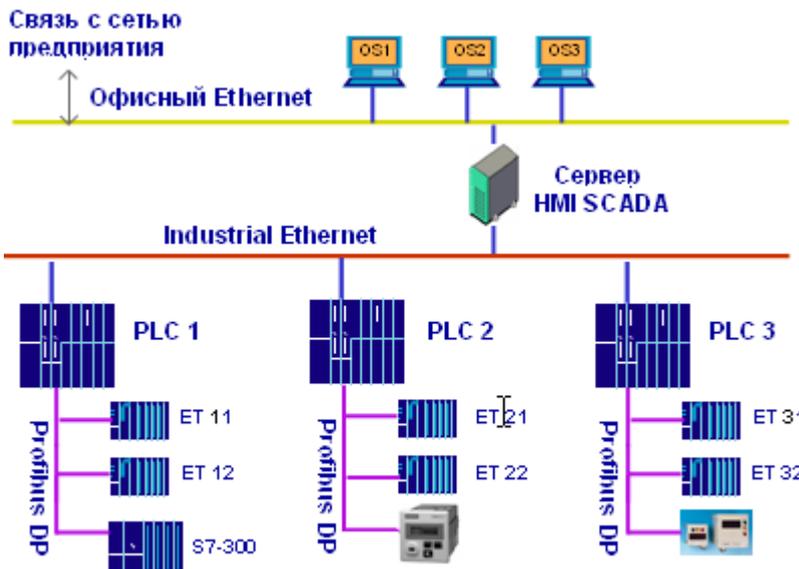
SIMATIC S7-400H

Отказоустойчивый программируемый контроллер SIMATIC S7-400H предназначен для построения систем управления, отказ которых может повлечь за собой гибель людей, большие экономические потери из-за простоя оборудования, угрозу окружающей природной среде. Каждый контроллер создается на основе двух центральных процессоров CPU 417-4H (CPU 414-4H), связанных между собой оптоволоконными линиями связи,

подключаемыми к submodule синхронизации. CPU работают по принципу "ведущий-ведомый". Основным принципом является принцип горячего резервирования с поддержкой безударного автоматического переключения на резервный базовый блок в случае отказа ведущего базового блока.

Пример построения системы автоматизации

Современные сложные системы автоматизации строятся с использованием нескольких контроллеров, объединенных в промышленную сеть (например, Industrial Ethernet). Система ввода-вывода часто имеет распределенную структуру с использованием сети PROFIBUS DP. Преимущества децентрализованного доступа: внешние сигналы поступают в модули ввода-вывода, расположенные в непосредственной близости к объекту (при этом удаленность до CPU может составлять до 10 км); использование различной техники передачи данных (оптика, витая пара); развитая диагностика отказов модулей и возможность их "горячей замены"; высокое быстродействие (цикл шины 2-5 мс); "прозрачное" программирование (нет отличий от программирования централизованной периферии); возможность использования "интеллектуальной периферии".



Подготовка SIMATIC S7 к запуску программы

Инструкция для монтажа S7-300

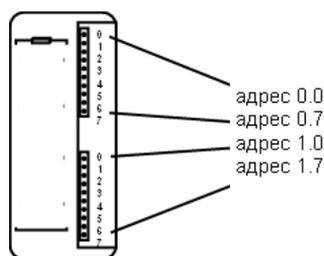
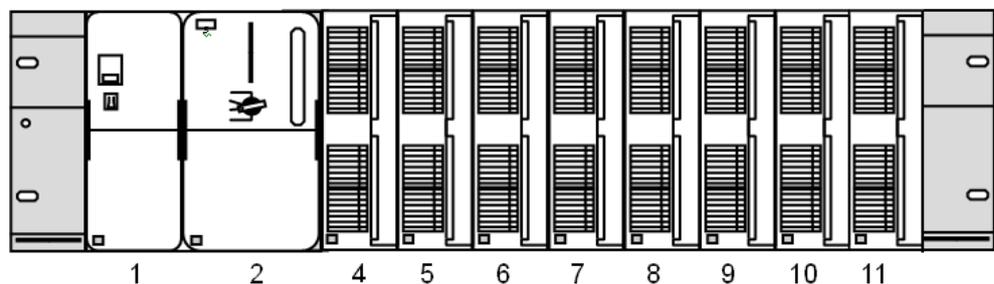
1. Монтируйте профильную шину.
2. Установите источник питания.
3. Установите шинный соединитель для CPU и монтируйте модуль.
4. Установите шинные соединители периферийных модулей и монтируйте модули.
5. Установите фронтштекер, описание и номера слотов.
6. Провести разводку модулей (ист. питания, CPU, периферийные модули).

Инструкция для монтажа S7-400

1. Монтируйте носитель модулей.
2. Монтируйте источник питания.
3. Удалите защитные крышки со слотов и монтируйте модули.
4. Установите фронтштекер и поместите надписи.
5. Выполните проводной монтаж модулей (ист. питания и периферийные модули).

Адресация модулей S7-300

Нумерация слотов (установочных мест) облегчает определение адресов модулей ввода-вывода.



Слоты в монтажной стойке S7-300 имеют логические номера. Слот для CPU имеет всегда номер 2. Номер 3 резервируется за модулем IM.

Размещение модулей DI/DO начинается со слота № 4 и до слота № 11. Для каждого слота под установку DI/DO резервируется 4 байта адресов (достаточно для отображения состояний всех дискретных входов или выходов 32-канального модуля).

Базовый (начальный) адрес модуля DI/DO определяется по формуле:

$$BA = [N_{\text{слота}} - 4] \times 4 + N_{\text{стойки}} \times 32,$$

где $N_{\text{слота}} \in 4..11$, $N_{\text{стойки}} \in 0..3$

Примеры адресации:

- Если в слот № 5 установлен модуль ввода-вывода дискретных сигналов (16DI/16DO), то его 8 первых входных каналов будут иметь адреса I4.0 ... I4.7, а первые 8 выходных каналов - адреса Q4.0 ... Q4.7.
- Первые 8 каналов модуля вывода дискретных сигналов, установленного в слот № 6, будут иметь адреса Q8.0 ... Q8.7.

Пример адресации модулей DI/DO (S7-300, 2-х стойки)

| Слот | | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|----|
| Источник питания | IM | 32.0 | 36.0 | 40.0 | 44.0 | 48.0 | 52.0 | 56.0 | 60.0 | |
| | (Приемник) | до 35.7 | до 39.7 | до 43.7 | до 47.7 | до 51.7 | до 55.7 | до 59.7 | до 63.7 | |

Стойка 1

| | | | | | | | | | | |
|------------------------|-------------|-----------|-----------|------------|------------|------------|------------|------------|------------|--|
| Источник питания и CPU | IM | 0.0 | 4.0 | 8.0 | 12.0 | 16.0 | 20.0 | 24.0 | 28.0 | |
| | (Передачик) | до 3.7 | до 7.7 | до 11.7 | до 15.7 | до 19.7 | до 23.7 | до 27.7 | до 31.7 | |

Стойка 0

Адресация в S7-400

Стойка центрального контроллера (стойка № 0)

| Слот | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---------------|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Базовый адрес | 0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 | 44 | 48 | 52 | 56 | 60 | 64 | 68 |

Стойка расширения (стойка № 1)

| Слот | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---------------|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Базовый адрес | 72 | 76 | 80 | 84 | 88 | 92 | 96 | 100 | 104 | 108 | 112 | 116 | 120 | 124 | 128 | 132 | 136 | 140 |

В программируемом контроллере S7-400 по умолчанию всем слотам монтажных стоек присваивается определенное адресное пространство. Задаваемые по умолчанию адреса определяются номером монтажной стойки и номером слота в монтажной стойке. Такая адресация называется географической.

В монтажной стойке центрального контроллера для вычисления адресов справедливы следующие формулы:

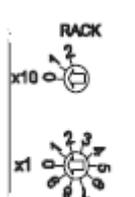
Начальный адрес цифрового блока = [№слота - 1] x 4

Начальный адрес аналогового блока = [№слота. - 1] x 64 + 512

В стойках расширения справедливы следующие формулы:

Начальный адрес цифрового блока = [№ стойки x 18 + №Слота - 1] x 4

Начальный адрес аналогового блока = [№ стойки x 18 + №Слота - 1] x 64 + 512



Номер стойки расширения задается в приемном интерфейсном модуле IM (1-21).

При выполнении операций конфигурирования и параметрирования с помощью STEP 7 географические адреса модулей для S7-300 и S7-400 могут быть заменены логическими адресами. Логический адрес модуля может выбираться свободно из доступного для CPU диапазона адресов.

Инструкция для ввода в эксплуатацию SIMATIC S7

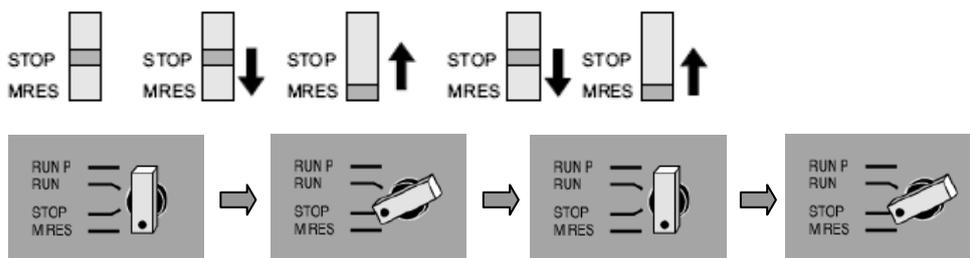
1. Включить питание.
2. Сбросить CPU.
3. Провести старт.
4. Выполнить визуальный контроль индикаторов на CPU, источнике питания и модулях.
5. Подключить программатор и запустить программу SIMATIC Manager.
6. Проверить входы/выходы с помощью утилиты HW Config.
7. Выполнить загрузку программы в CPU.

Сброс CPU

При сбросе CPU происходит очистка программы и данных в рабочей памяти (RAM) и области системной памяти. Затем проводится тест аппаратуры и, если модуль памяти (FLASH-память или MMC) вставлен в CPU и имеет программу, то программа и конфигурационные данные из модуля памяти копируются в рабочую память. При этом все сохраняемые данные очищаются, а все данные в блоках данных принимают начальное значение.

Выполнение сброса

Установите переключатель работы в режим "STOP". На CPU должен гореть желтый светодиод "STOP".



Переведите переключатель режимов работы в положение **MRES** и удерживайте его там не менее 3 секунд (пока желтый светодиод "STOP" не мигнет 2 раза). Отпустите переключатель (он перейдет в позицию "STOP") и сразу (максимум 2 секунды) снова поверните его в положение **MRES**. Если светодиод "STOP" замигает быстро (частота 2Гц), то CPU сброшен. Снова отпустите переключатель.

Выполнение старта

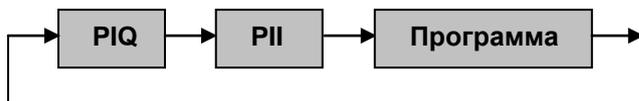
Поверните переключатель режимов работы в **RUN**. Загорится зеленый светодиод "RUN", а желтый светодиод "STOP" погаснет.

При новом старте удаляются области отображения процесса (PII, PIQ) и вся системная память, кроме тех областей памяти, которые назначены как сохраняемые, а затем начинается циклическая работа CPU.

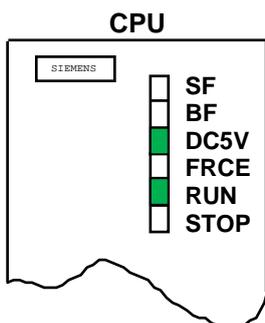
Примечание: В ранних моделях CPU изменение программы или любых данных без остановки CPU было возможно только в позиции RUN-P, а в позиции RUN – невозможно.

Режим RUN

В режиме RUN операционная система CPU выполняет циклический алгоритм, который предусматривает системные процедуры передачи данных из внутренней памяти PIQ (области отображения выходов) на модули вывода и передачи сигналов с модулей ввода в системную память PII (отображение входов). Затем управление передается на первую команду программы пользователя.



Визуальный контроль индикаторов в S7-300



SF - Суммарная ошибка, программная ошибка в CPU или ошибка от модуля с внутренней диагностикой.
BF - Ошибка сети PROFIBUS DP (для модулей с дополнительным интерфейсом DP). Светится постоянно - при коротком замыкании кабеля или неправильной конфигурации. Мигает - при отсутствии связи с каким-либо участником по PROFIBUS DP.
DC5V - Индикация о наличии 5V.
FRCE - Индикация об установленном режиме Force (по меньшей мере, один вход или выход управляется принудительно в соответствии с таблицей значений).
RUN - Run-режим: мигает при старте CPU, светится постоянно в Run-режиме.
STOP - Stop-режим: медленно мигает, если требуется сброс (например, при отсутствии модуля MMC); быстро мигает во время сброса, светится постоянно в Stop-режиме.

Подключение программатора к CPU



Если Вы имеете USB-адаптер (6ES7972-0CB20-0XA0), то в качестве устройства программирования может использоваться любой персональный компьютер или ноутбук, который удовлетворяет требованиям к установке пакета STEP7 (WIN2000 PROF/ XP PROF, 512Mb RAM). Начиная с версии STEP 7 V5.4 SP3, поддерживается работа с Windows Vista Ultimate/Business. Версия STEP 7 V5.5 SP1 работает также с MS Windows 7 32/64-Bit.

USB- адаптер поддерживает протоколы MPI (S7-300/400), PPI (S7-200) и PROFIBUS-DP. Максимальная скорость передачи 1.5Mbps.

Для связи с S7-CPU Вы можете также использовать специальные коммуникационные модули SIMATIC NET, например, CP 5512 (PC-CARD) или CP 5611 (PCI-CARD).

Установление online связи

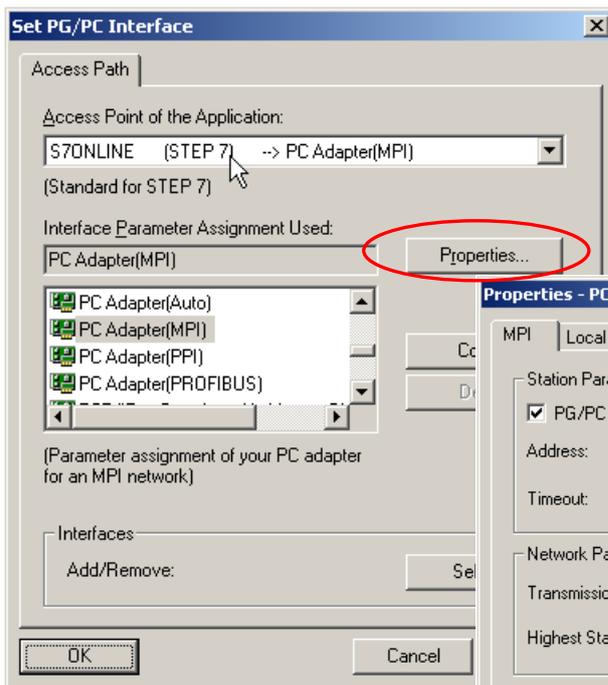
Online (онлайн) связь позволяет выполнить важные функции по работе с CPU, такие как чтение, загрузка программы, чтение/управление переменными, получение диагностической информации и т.д.



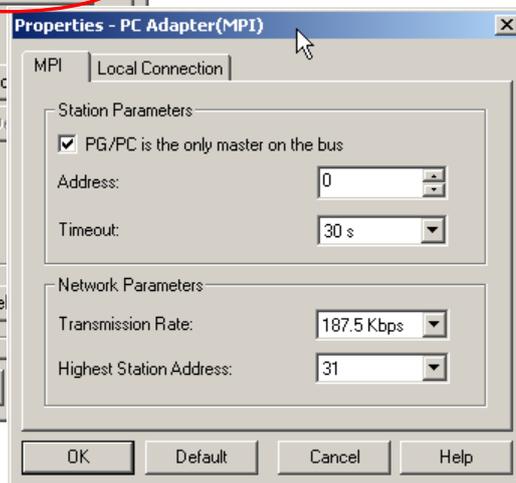
Параметрирование on-line соединения (выбор интерфейса, настройки) может быть выполнено:

- 1) через программу SIMATIC Manager (*Options-> Set PC/PG Interface*)
- 2) через панель управления вашего компьютера (*Пуск-> Настройка->Панель управления->Set PC/PG Interface*)

Выбор типа интерфейса для online соединения



По кнопке Properties Вы выбираете параметры MPI или PROFIBUS узла (сетевой адрес, скорость обмена) и тип узла (на вкладке Local Connection для USB – адаптера выбираем USB).



Проверка соединения через окно "Accessible Nodes"



Окно "Accessible Nodes" (Доступные узлы) открывается с помощью команды меню PLC -> Display Accessible Nodes или соответствующей кнопки панели инструментов. В окне "Accessible Nodes" отображаются все модули, доступные в сети, вместе с их адресами.

| Object name | Rack/Slot | Status | Module type |
|-------------|-----------|--------|-------------------|
| MPI = 2 | 0/2 | RUN | CPU 313C-2 DP |
| MPI = 3 | -/- | RUN | CP 343-1 Advanced |

| Object name | Rack/Slot | Status | Module type |
|-------------------------|-----------|--------|---------------|
| PROFIBUS = 2 (directly) | 0/2 | RUN | CPU 313C-2 DP |

| Object name | Rack/Slot | Status | Module type |
|----------------------|-----------|--------|-------------------|
| CP-343-1-Advanced-IT | -/- | RUN | CP 343-1 Advanced |

Тестирование сигналов ввода вывода

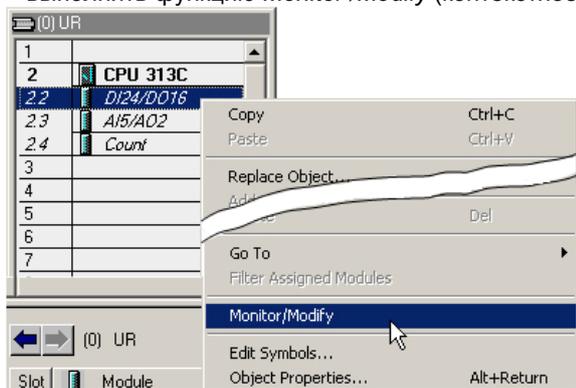
Если online соединение установлено, можно переходить к тестированию сигналов ввода/вывода из утилиты HW Config.

Для этого выполните следующие действия:

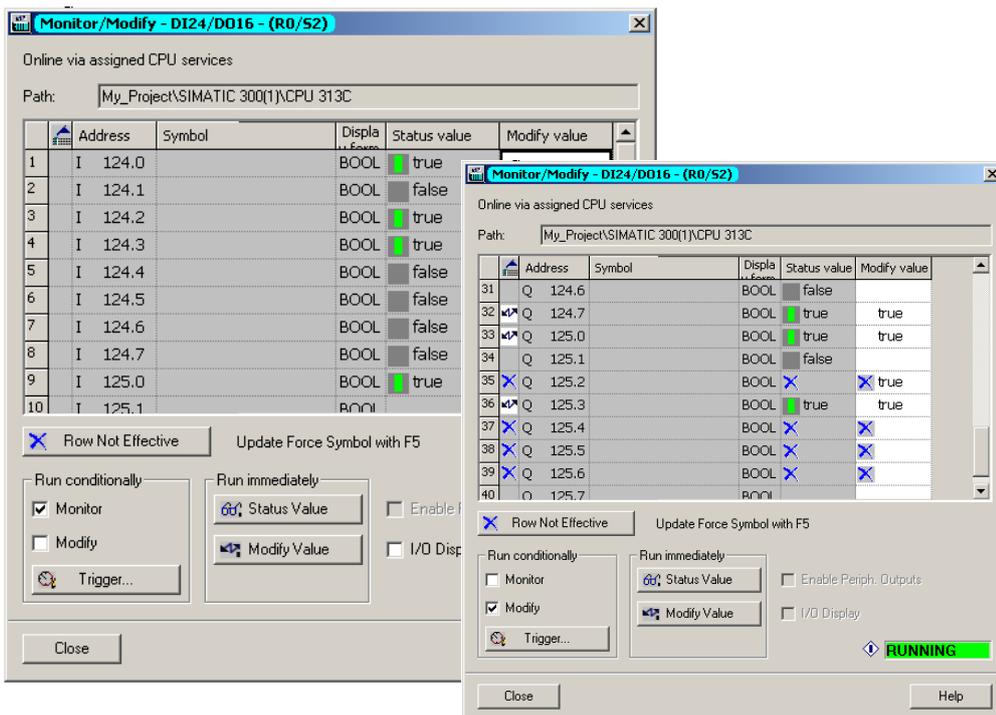
1. Создать новый проект (*File -> New...*) или открыть существующий (*File -> Open...*).
2. Для нового проекта выделить имя проекта и выполнить функцию *PLC -> Upload Station to PG...*
3. Выделить объект с именем станции и выполнить функцию *Open Object* (контекстное меню) или дважды кликнуть мышью на объекте Hardware.
4. В окне утилиты HW Config выделить модуль ввода-вывода и выполнить функцию *Monitor /Modify* (контекстное меню).



Hardware



5. В появившемся окне **Monitor /Modify** провести тестирование сигналов.



Загрузка программы в CPU



Загрузка программы в CPU выполняется с помощью команды *PLC -> Download* или соответствующей кнопки панели инструментов.

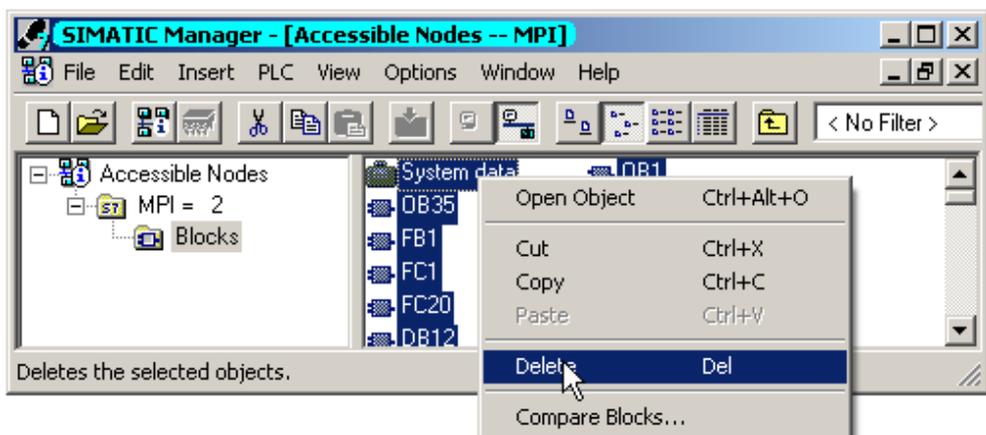
Загрузка всей программы (должна быть выделена папка "Blocks") или отдельного блока (выделяется блок/группа блоков) всегда проводится в загрузочную память CPU. По окончании загрузки операционная система CPU выполняет заключительную компиляцию и размещение блока в рабочей памяти.



Если загрузочной памятью является модуль MMC (NVFlash-EEPROM), то перед загрузкой новой программы модуль должен быть "чистым" (т.е. не содержать старой программы).

Для очистки MMC:

- 1) выбрать папку "Blocks" в окне "Accessible Nodes";
- 2) выделить все блоки в правом подокне (Ctrl+A) и нажать кнопку "Delete".



Основные определения

Время цикла

Время цикла - это время, которое необходимо операционной системе для обработки одного прогона программы, т.е. одного цикла ОВ 1, а также всех прерывающих этот цикл частей программы и системных операций. Это время контролируется.

Время цикла программы пользователя увеличивается за счет:

- обработки прерываний (по времени, от аппаратуры, обработки ошибок, т.д.)
- обмена данными с устройствами программирования, панелями оператора (OP) и подключенными CP (например, Ethernet, PROFIBUS-DP)
- функций тестирования (например, наблюдение и управление переменными, статус блоков)
- передачи и удаления блоков, сжатия памяти.

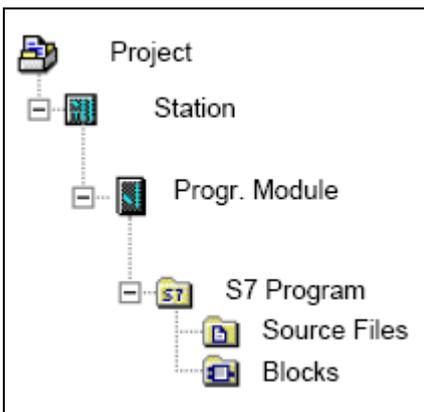
Время реакции

Время реакции - это время от обнаружения входного сигнала до изменения связанного с ним выходного сигнала.

Максимальное время реакции (когда изменение сигнала произошло после формирования области отображения PII) равно сумме удвоенного времени цикла и запаздывания входов и выходов. Более быстрой реакции можно достичь применением аппаратных прерываний.

Управление проектом в SIMATIC Manager

Структура проекта



Иерархия объектов для проектов и библиотек в STEP 7 отображается таким же образом, как Windows Explorer отображает структуру каталогов из папок и файлов.

Каждый проект представляет базу данных, в которой хранятся все актуальные данные.

Объекты **Station** хранят информацию о конфигурации аппаратуры и параметрах модулей SIMATIC S7 300/400.

Объекты **Progr. Module** - программируемые модули (CPU, FM, CP).

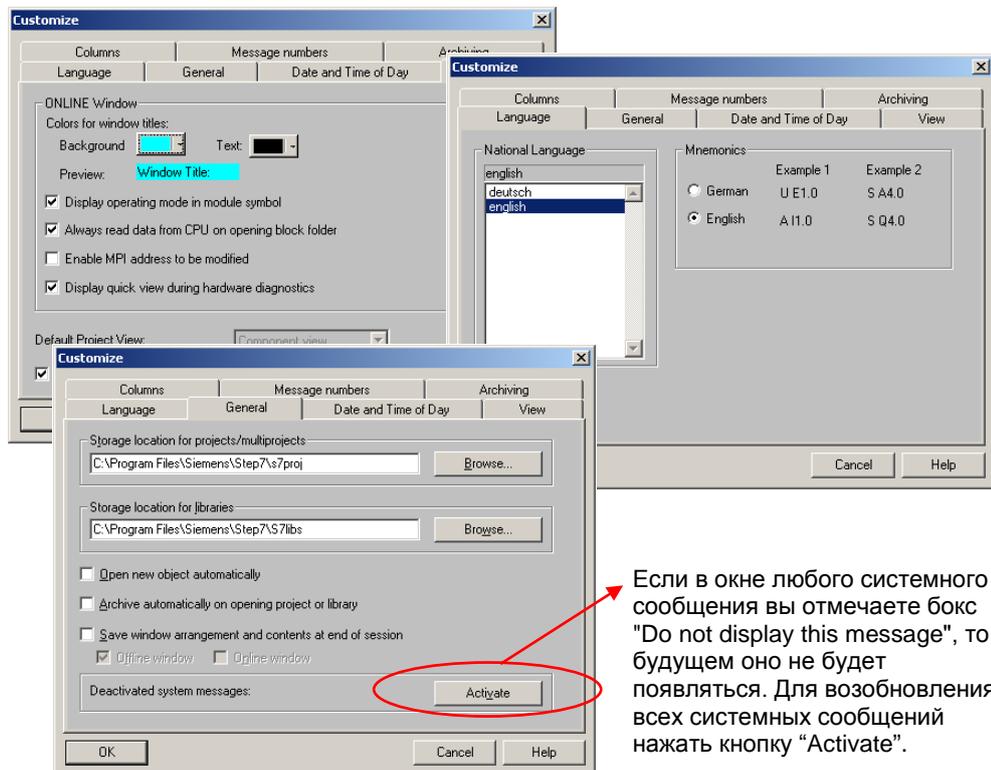
Объект **S7 Program** - программа для подчиненного объекта.

Объект **Source Files** - исходные тексты программ на языке STL или высокого уровня (SCL, GRAPH 7).

Объект **Blocks** - программные блоки для загрузки в CPU и для редактирования на языке STEP7.

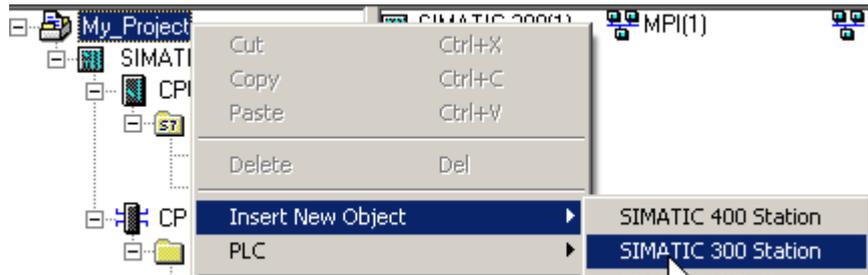
Настройки проекта

Пункт меню: **Options -> Customize**



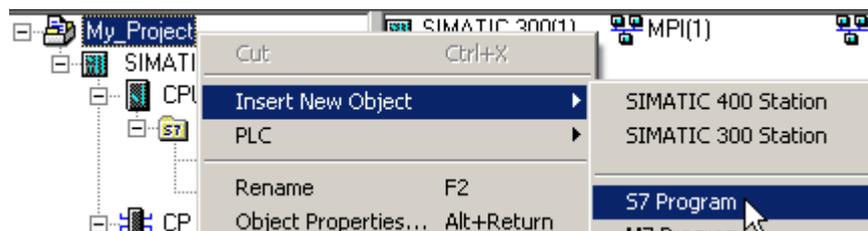
Вставка новой станции

Вставить станцию в проект для последующего конфигурирования можно через главное меню *Insert ->Station* либо через контекстное меню.



Вставка папки "S7 Program"

Папку с новой S7-программой можно вставить через главное меню *Insert ->Program* либо через контекстное меню.

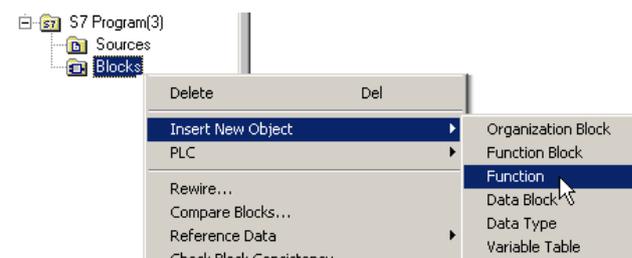


Папка "S7 Program", неподчиненная S7-станции, используется для:

- Создания программы раньше процедуры конфигурирования станции
- Хранения различных версий программы / блока
- Хранения различных Символьных таблиц (Symbol Table)
- Создания тестовой программы для последующей отладки на симуляторе (дополнительная программа S7-PLSIM).

Вставка блока

Новый блок можно вставить через главное меню *Insert ->S7 Block*, либо через контекстное меню.



Тип блока должен быть определен заранее. После выбора типа блока необходим ввод номера блока. Номер блоков ОВ задает связь блока с функцией операционной системы и не может выбираться произвольно.

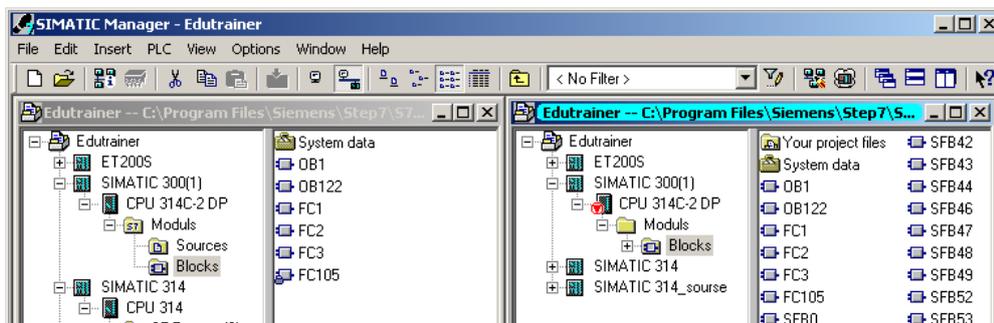
Номер блоков FC, FB, DB задается произвольно из диапазона, разрешенного для CPU (например, 0-255). Номера DB задается от 1 до максимально допустимого номера.

On-line проект



С помощью команды меню *View -> Online* или соответствующей кнопки Вы можете открыть окно проекта в режиме **online**.

Online проект отображает данные проекта, взятые из контроллера. Доступ через окно online объединяет данные в программируемой системе управления с соответствующими данными в устройстве программирования. Если, например, Вы открываете блок S7 из online проекта, то раздел кода берется из блока в CPU, а комментарии и символы из базы данных в устройстве программирования.



• Доступ с конфигурированием аппаратуры

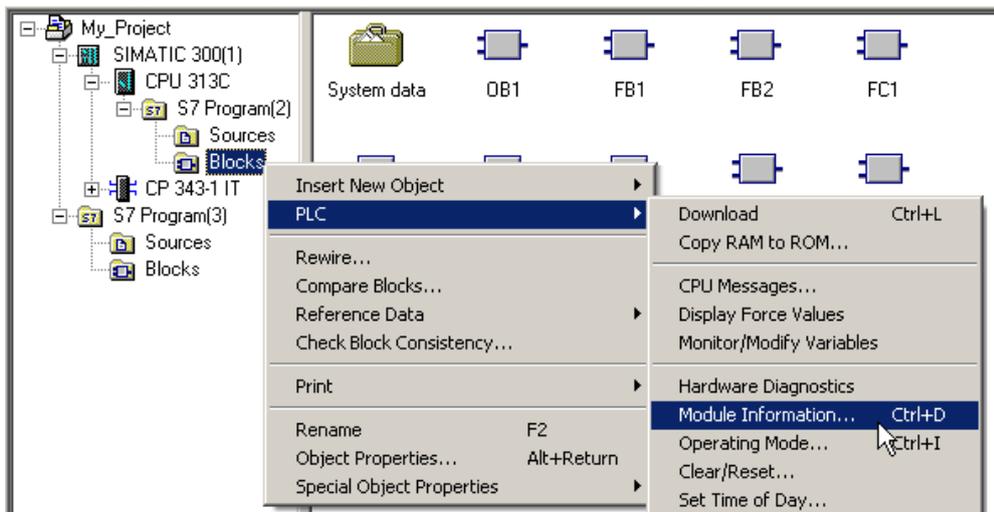
Вы можете обращаться только к станциям, которые были сконфигурированы в режиме offline, причем в точном соответствии с заданным адресом MPI.

• Доступ без конфигурирования аппаратуры

К какому конкретному CPU Вы можете получить доступ в режиме online, определяется указанием соответствующего адреса MPI при первой установке online связи.

Выполнение online функций

Все online функции доступны через главное меню PLC или через контекстное меню.



Основные online функции:

Monitor/Modify Variables – функции по наблюдению и управлению входами, выходами, флагами (ячейками памяти из системной области M), таймерами и т.д.

Hardware Diagnostics – функции чтения диагностической информации по аппаратуре.

Module Information – функции по запросу системной информации: время цикла, использования памяти, сведения о системных ресурсах CPU (количество блоков, таймеров, флагов) и об ошибках функционирования.

Operating Mode – функции управления запуском и остановом CPU.

Clear/Reset – функция очистки RAM-памяти и сброса CPU.

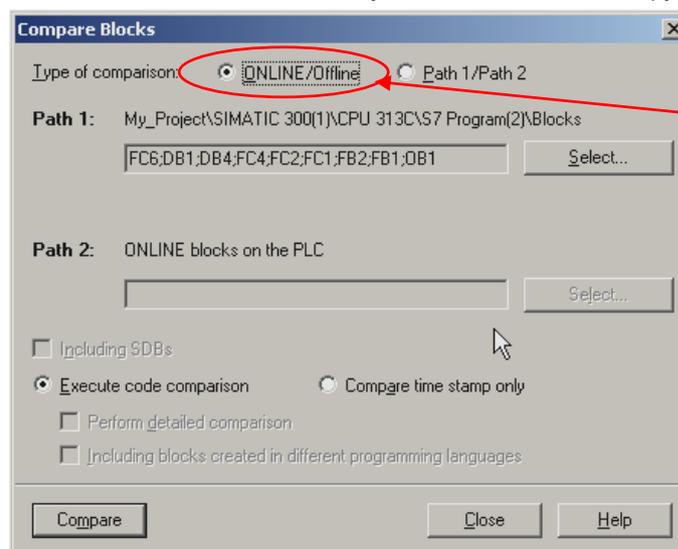
Set Date and Time – функция по установке даты и времени в CPU

Download user program to memory card – загрузка блоков программы в карту памяти (MEMORY CARD FLASH-EPROM), установленную в слот CPU.

Операции над блоками

Блоки программы (OB, FC, FB, DB) составляют основу программы пользователя. Выделяя блок(и) в окне проекта Вы можете выполнять следующие действия:

- Загрузка блока в контроллер (функция **Download**).
Загрузка блоков методом буксировки: 1) выделить блок(и) в offline проекте и 2) перетащить мышью в online окно проекта.
- Чтение блоков. Кликнув мышью (или по кнопке Enter), Вы открываете блоки с помощью редактора (LAD/ FBD/ STL).
- Копирование/удаление блоков. Вы можете копировать блоки как внутри одного проекта, так и разных проектов аналогично действиям при работе с файлами в Windows Explorer. Удаление блока в online проекте удаляет его из загрузочной и рабочей памяти CPU.
- Изменение номера блока. Для изменения номера блока достаточно выделить блок и кликнуть 1 раз мышью на объектном имени.
- Сравнение блоков (функция **Compare Blocks**). Вы можете сравнить блок(и) с аналогичными блоками доступного CPU или любой другой папки S7 Program.



Выбор режима сравнения с CPU.

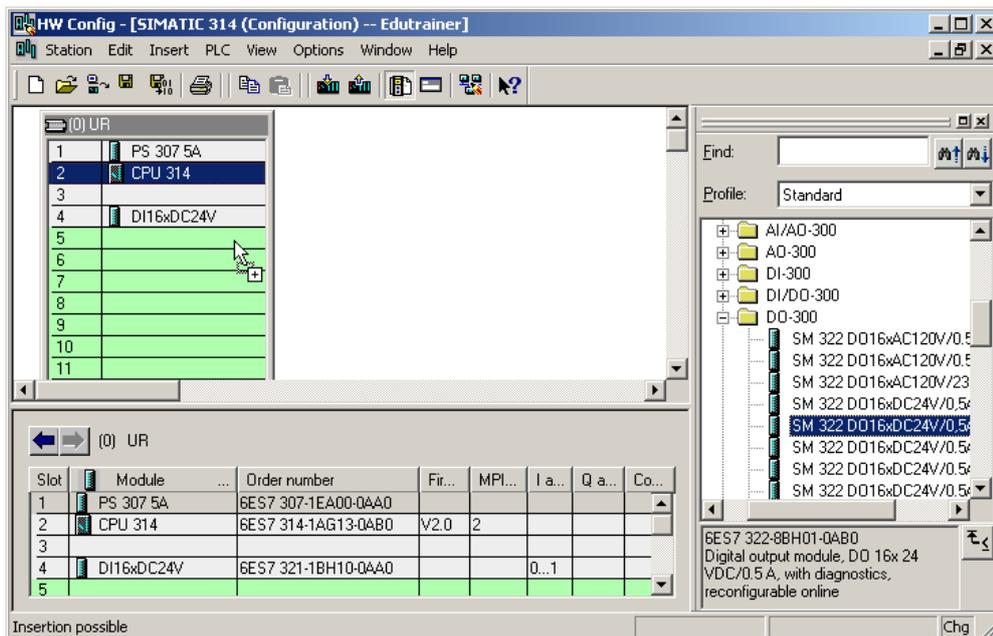
- Получение информации о свойствах блока (размер кода в памяти CPU, атрибутов, ввод символического имени, комментариев) – функция контекстного меню *Object Properties*.

Конфигурирование и параметрирование SIMATIC S7

Если Вы знаете состав модулей станции (в том числе заказные номера, расположенные на фронтальной панели модулей), то проектирование задачи автоматизации мы рекомендуем начать с конфигурирования станции. Преимущество этого подхода состоит в том, что редактор конфигурирования аппаратуры в STEP 7 отображает возможные адреса сигналов и позволяет сразу задать для них символьные адреса. При конфигурировании аппаратуры Вы не только можете определять адреса, но и можете также изменять параметры и свойства модулей. Например, если вы хотите работать с несколькими CPU, то вы должны согласовывать адреса MPI этих CPU.

Конфигурирование:

1. Выделить новую станцию. В правом подокне выделить объект "Hardware".
2. Открыть объект "Hardware". Откроется окно "HW Config".
3. В окне "HW Config" открыть каталог блоков *View -> Catalog*.
4. Сначала выбрать носитель модулей (rack) и разместить его в пустом окне конфигуратора (кликнуть мышью) Затем выбрать остальные модули и разместить их в соответствующие слоты корзины. Как минимум, укажите только CPU.



Примечание: при конфигурировании станции S7-300 слот №3 резервируется для интерфейсного модуля (IM).

Обновление каталога элементов

Для новых модулей, например CPU или I/O-модулей, которых нет в каталоге элементов, Вы можете выполнить команду обновления каталога элементов *Options -> Install HW Updates*. Обновление возможно через Internet (стандартная страница <http://www.siemens.com/automation/step7-hwconfig2>), либо через каталог файлов на жестком диске. При выборе Internet создается каталог C:\HWUpdates, который в дальнейшем можно использовать для обновления HW-библиотеки на компьютерах без подключения к Internet.

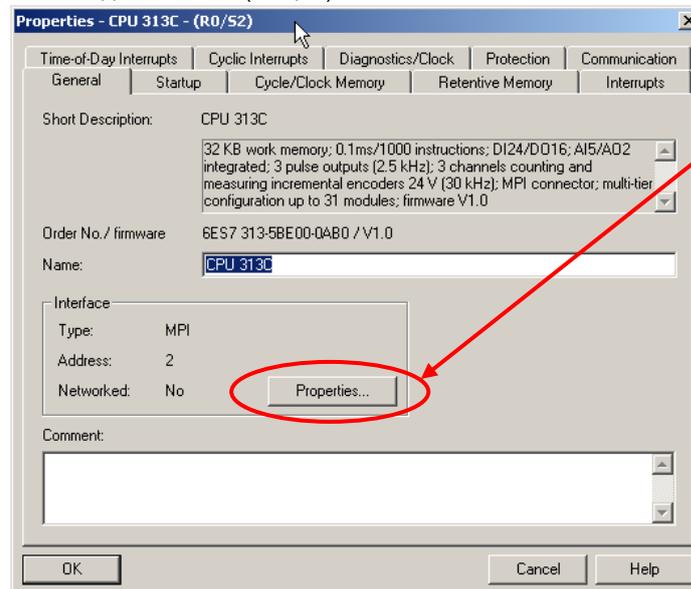
Параметрирование станции

Вы назначаете модулям параметры, чтобы отказаться от системных настроек модулей устанавливаемых при сбросе CPU (например, адресация модулей), или еще при производстве модуля (MPI адрес).

Параметрирование CPU

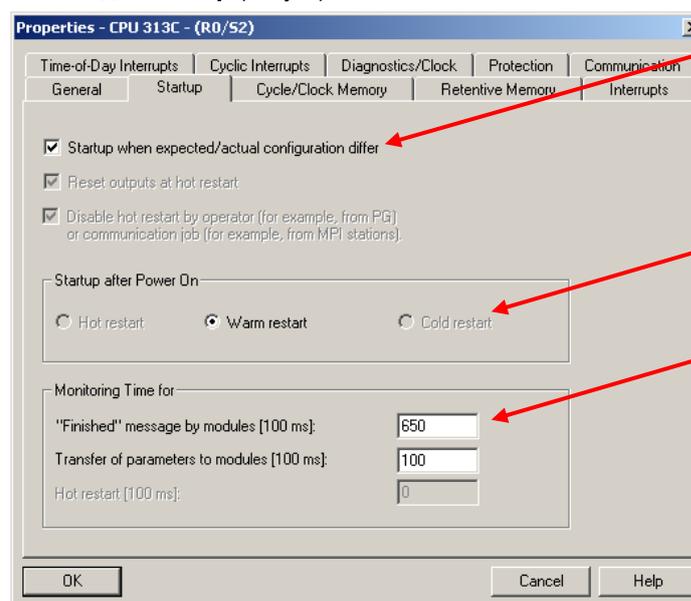
Выбрать модуль CPU и открыть окно свойств (Properties).

- Вкладка **General** (общие)



Кнопка для изменения MPI адреса.

- Вкладка **Startup** (запуск)

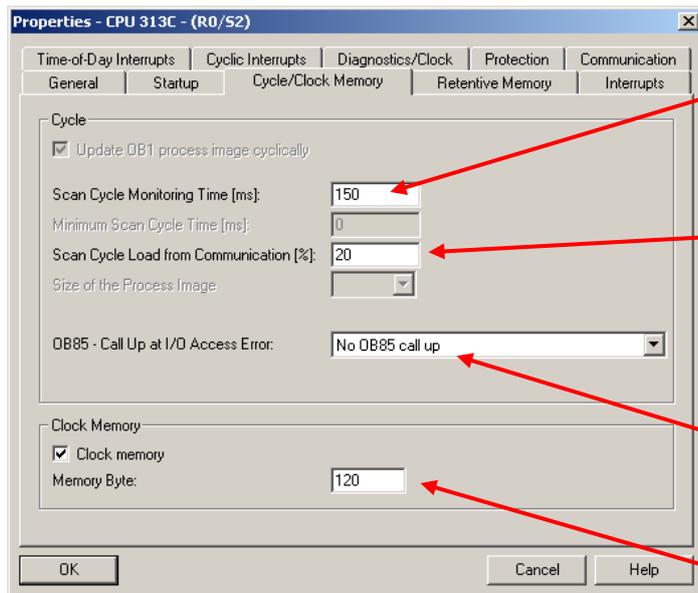


Разрешение старта CPU при различии текущей (определяемой ОС) и заданной конфигурации.

Вид старта после включения питания.

Максимальное время для получения сообщения о готовности модулей и передачи параметров в модули.

- Вкладка **Cycle/Clock Memory** (Цикл/ Байт синхронизации)



Максимальное время цикла (150 ms)/ Если это время превышено, CPU переходит в STOP (при отсутствии блока ошибки OB80).

Коэффициент (К) увеличения времени цикла за счет коммуникационной нагрузки (Н%)

$$K = \frac{100}{100 - N\%}$$

Реакция CPU на ошибки доступа при работе с областями отображения PII/PIQ.

Байт синхронизации.

Байт синхронизации (Clock Memory) задается в области М (Memory Byte) через указание его адреса. При задании синхробайта, операционная система CPU изменяет 8 бит этого байта с различной частотой (имеем 8 генераторов фиксированной частоты).

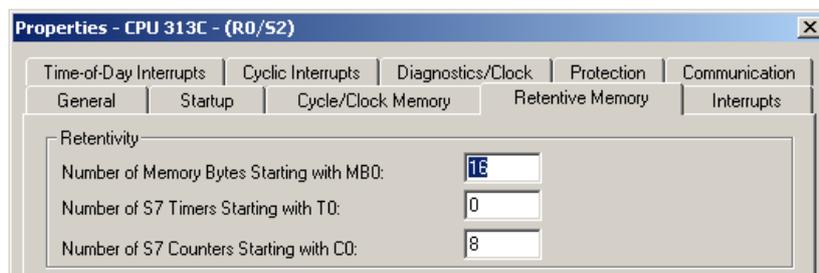
| Синхробит | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------|-----|------|---|------|-----|-----|-----|-----|
| Частота (Гц) | 0.5 | 0.62 | 1 | 1.25 | 2 | 2.5 | 5 | 10 |
| Период (с) | 2 | 1.6 | 1 | 0.8 | 0.5 | 0.4 | 0.2 | 0.1 |

Network 1: мигание лампочки (1Гц)



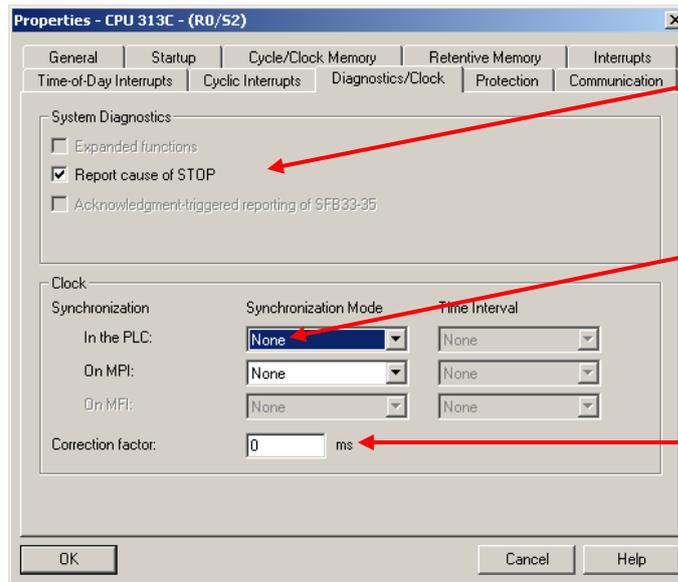
На рисунке слева вы видите пример управления лампочкой с помощью синхробита M120.5 с частотой 1 Гц. Для изменения частоты достаточно выбрать другой номер синхробита.

- Вкладка **Retentive Memory** (Сохраняемая память)



Вы определяете диапазон флагов, таймеров, счетчиков (начиная с 0), содержимое которых сохраняется при выключении питания и восстанавливается при старте CPU.

- Вкладка **Diagnostic/Clock** (Диагностика/Часы)



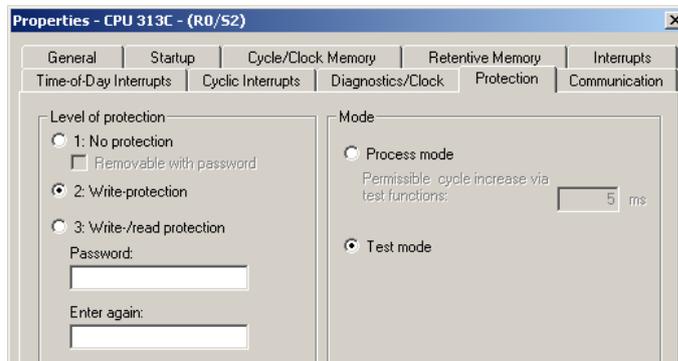
Разрешение CPU для передачи сообщения на PG/OP при переходе в STOP.

Настройки для синхронизации часов для нескольких CPU, работающих в сети MPI или между модулями внутри станции.

Показатель коррекции для учета погрешности часов.

Пример: Если часы за 24 часа убегают на 3 секунды, то это может быть скорректировано показателем "-3000ms".

- Вкладка **Protection** (Защита)



Вы можете сделать доступ к CPU на запись или чтение/запись, зависящим от ввода правильного пароля. С помощью команды меню *PLC* -> *Access Rights* Вы можете вызвать диалоговое окно "Enter Password (Введите пароль)".

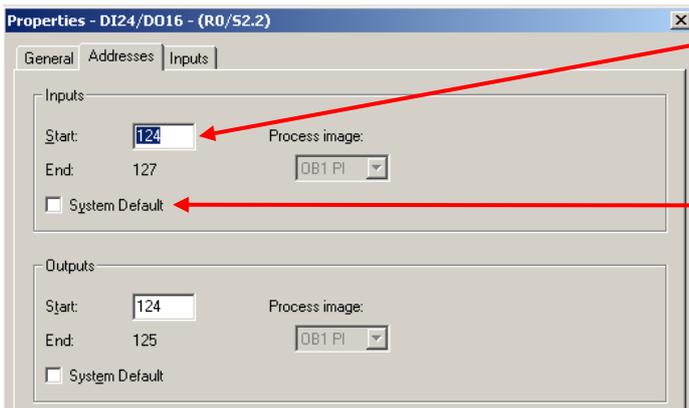
В режиме **Process mode** тестовые функции, такие как статус программы или управление/наблюдение переменных, ограничены временем, что не позволяет увеличить текущий цикл CPU на установленный предел.

В режиме **Test mode** все тестовые функции могут использоваться без ограничений, даже если они вызывают существенное увеличение времени цикла.

Параметрирование модулей ввода/вывода

Выбрать соответствующий модуль и открыть окно свойств (Properties). В открывшемся окне Вы можете задать адрес модуля и, если позволяет модуль, задать, например, условия аппаратных / диагностических прерываний.

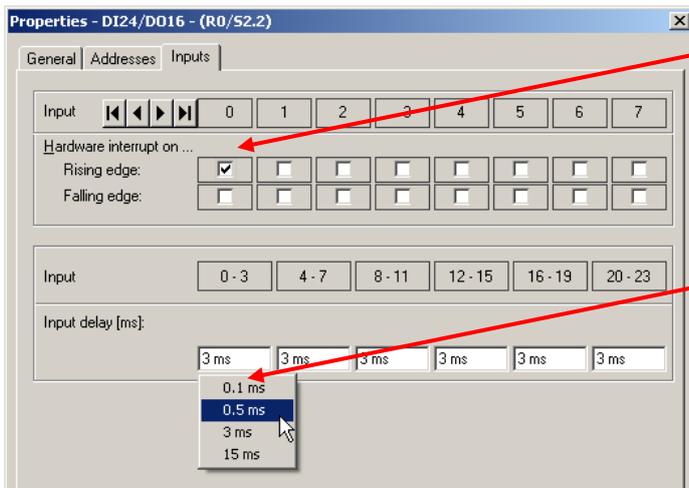
Параметрирование модуля DI/DO (вкладка Addresses)



Ввод нового адреса модуля.

Задание адреса в соответствии с системными настройками ("географический адрес").

Параметрирование модуля DI/DO (вкладка Inputs)



Ввод условия (+фронт сигнала) на формирование прерывания от сигнала на входе (адрес I124.0).

Программируемая задержка для оценки сигнала "лог. 1".

Сохранение и загрузка заданной конфигурации

С помощью команды меню *Station->Save* Вы можете сохранить текущую конфигурацию в текущем проекте (без генерации системных блоков данных).

Когда Вы выбираете команду *Station->Save and Compile* или нажимаете на соответствующую кнопку в панели инструментов, конфигурация и назначенные параметры сохраняются также в системных блоках данных (System Data).



Вы выбираете команду меню *PLC ->Download* или нажимаете на соответствующую кнопку в панели инструментов, чтобы загрузить выбранную конфигурацию в PLC (PLC должен быть в режиме "STOP").



Вы можете выбрать команду меню *Station -> Consistency Check*, чтобы проверить результат конфигурирования на ошибки.

Чтение текущей конфигурации и параметров из CPU



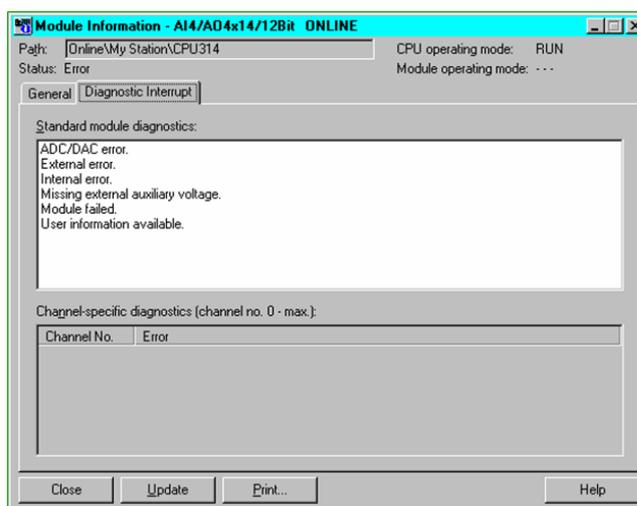
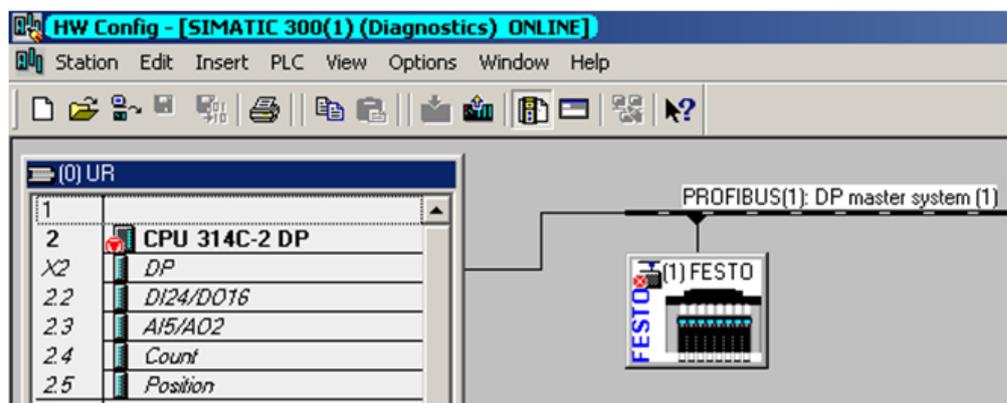
Через меню *PLC -> Upload* или с помощью соответствующей кнопки Вы можете прочитать текущую конфигурацию S7-станции. Фактическая конфигурация, считанная из контроллера, размещается в выбранном проекте как новая станция.

Примечание. При чтении реальной конфигурации модули могут быть определены не полностью. По этой причине Вы должны проверить конфигурацию и, если необходимо, указать точный тип существующих модулей. Для того чтобы сделать это, выберите модуль, а затем выберите команду *Options -> Specify Module*.

Диагностика аппаратуры



Для чтения диагностической информации об аппаратуре кликните на соответствующую кнопку. В результате откроется окно с online информацией. Вы получите тот же результат через online-функцию *PLC-> Hardware Diagnostics*.

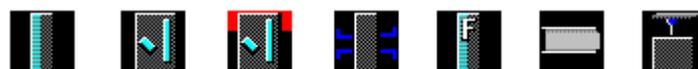


Если CPU находится в режиме STOP или если имеются проблемы при обслуживании модуля, то это показывается специальными символами.

Вы можете дважды щелкнуть на CPU или неисправном модуле, чтобы получить более подробную диагностическую информацию.

В приведенном окне диагностики для аналогового модуля отражается причина ошибки - отсутствие внешнего напряжения.

Информацию о символах, используемых при отображении статуса модулей, Вы можете получить через окно Help (стандартно, по F1).



Структура памяти CPU

Память S7-CPU можно разделить на три области: загрузочная память, рабочая память и системная память.

Загрузочная память предназначена для загрузки программы пользователя. Если она интегрирована в CPU (RAM), то она может быть расширена за счет использования внешних модулей памяти (RAM, EPROM). В компактной серии CPU 31xC (и других CPU серии S7-300) в качестве загрузочной памяти используются микрокарты памяти (MMC). В этом случае работа CPU без установки MMC не возможна. Микрокарты MMC выбираются из ряда 64 Kb, 128 Kb, 512 Kb, 2 Mb, 4 Mb, 8 Mb и заказываются отдельно.

Для обновления/сохранения версии операционной системы (firmware) необходимо иметь MMC объемом от 2 до 8MB (в зависимости от CPU). Файлы обновлений firmware находятся в Интернете по адресу <http://www.siemens.com/automation/service&support>.

Рабочая память (всегда RAM) встроена в CPU и не может быть расширена. Она служит для размещения исполняемого кода программы и данных. В CPU S7-400 рабочая память имеет 2 части: одна для размещения программы, другая - только для блоков данных (операции с данными быстрее). Доступ к рабочей памяти имеет только операционная система CPU.

Устройство программирования



Сжатие (функция *Module Information->Memory->Compress*)

В результате процессов загрузки и удаления блоков в рабочей памяти образуются неиспользуемые области памяти (например, при загрузке новой версии блока каждый раз выделяется новая область кода, а старая становится неиспользуемой). При сжатии происходит перераспределение блоков программы с высвобождением ранее занятых областей. Функция подобна процедуре дефрагментации жесткого диска.

Системная память встроена в CPU и не может быть расширена. Она содержит:

- образы процесса на входах и выходах - (RAM)
- области операндов: битов памяти (флагов/меркеров), таймеров и счетчиков - (RAM)
- локальные данные - (RAM)
- диагностический буфер, MPI адрес, счетчики времени работы - (сохраняемая память)

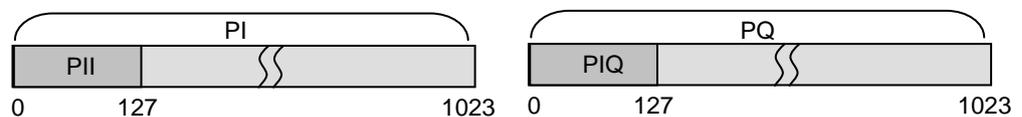
Адресное пространство CPU

Совокупность программно доступных адресов модулей ввода-вывода образуют адресное пространство CPU (I/O address area). В системе памяти S7-CPU это система адресов разделена на 2 равные области - для чтения и записи. Размеры адресного пространства зависят от типа CPU и занимают от 1 Kbyte до 16 Kbyte.

Следующая таблица показывает обозначения периферийных адресов в STEP7.

| Область адресов | Доступ | Запись в STEP7 |
|------------------------------|---|---|
| Периферийная область: входы | Периферийный входной байт Периферийное входное слово Периферийное входное двойное слово | PIBx PIWx PIDx |
| Периферийная область: выходы | Периферийный выходной байт Периферийное выходное слово Периферийное выходное двойное | PQBx PQWx PQDx |

Области отображения CPU представляют ячейки системной памяти, которые по диапазону адресов совпадают с адресным пространством только по младшим адресам. Например, CPU 314C-2DP имеет размер PII 128 байт, область PI – 1024 байта. Это означает, что вся считываемая информация из модулей ввода, имеющих адреса с 0 по 127 (от PIB0 до PIB127), размещается по адресам области PII с IB0 по IB127.

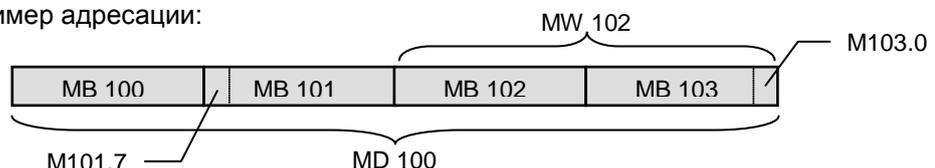


Адресация ячеек системной памяти

Следующая таблица показывает доступные области системной памяти.

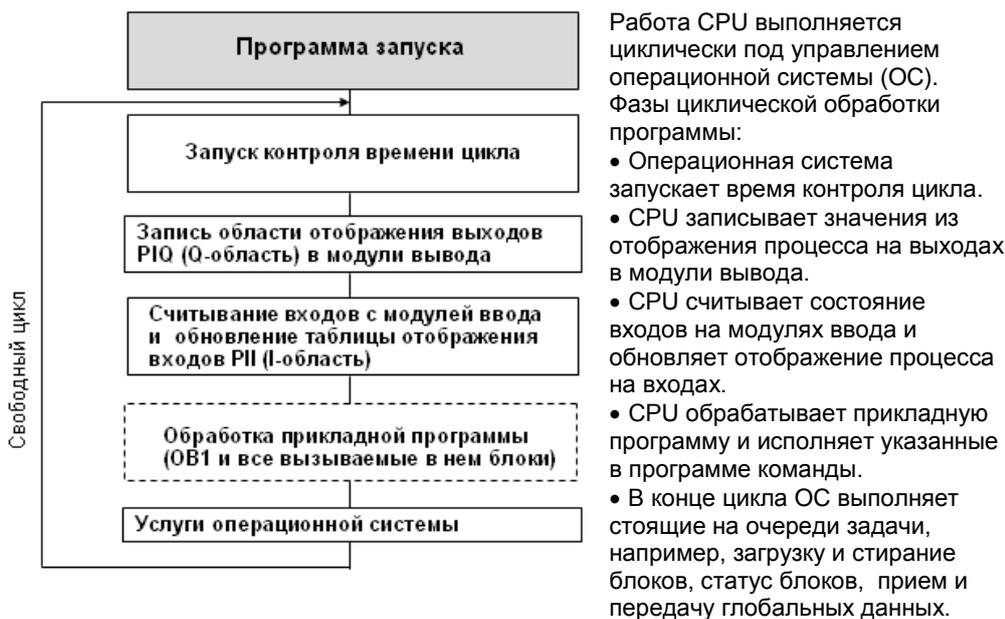
| Область адресов | Доступ | Запись в STEP7 |
|-------------------------------|---------------------------------------|--|
| Флаги | Бит Байт Слово Двойное слово | Mx.y MBx MWx MDx |
| Области отображения (PII/PIQ) | Бит Байт Слово Двойное слово | Ix.y / Qx.y IBx / QBx IWx / QWx IDx / QDx |
| Таймеры/ Счетчики | Таймер / Счетчик | Tx / Cx |
| Данные локального стека | Бит Байт Слово Двойное слово | Lx.y LBx LWx LDx |

Пример адресации:



Основы программирования на языке STEP7

Принцип выполнения программы в S7-CPU. Время контроля цикла.



Время, необходимое для выполнения всех перечисленных операций, называется циклом сканирования или временем цикла. Время цикла является величиной переменной и может меняться от цикла к циклу. В начале каждого цикла ОС перезапускает счетчик времени цикла.

Типы блоков.

Программа пользователя обычно представляет собой совокупность различных блоков, которая образует определенную программную структуру.

Организационные блоки (OB)

Организационные блоки вызываются ОС.

Они выступают в качестве интерфейса между ОС и программой пользователя. Они управляют:

- поведением PLC при старте
- циклическим выполнением программы
- обработкой прерываний
- обработкой ошибок

Функции и функциональные блоки (FC/FB)

Функции и функциональные блоки используются пользователем с целью построения программы из функционально законченных процедур, а также для реализации подпрограмм с параметрами.

Блоки данных (DB)

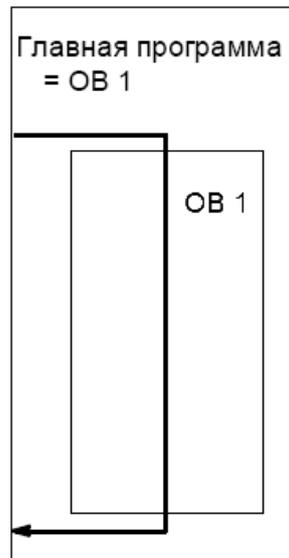
Блоки данных используются для хранения данных пользователя.

Системные функции и функциональные блоки (SFC / SFB)

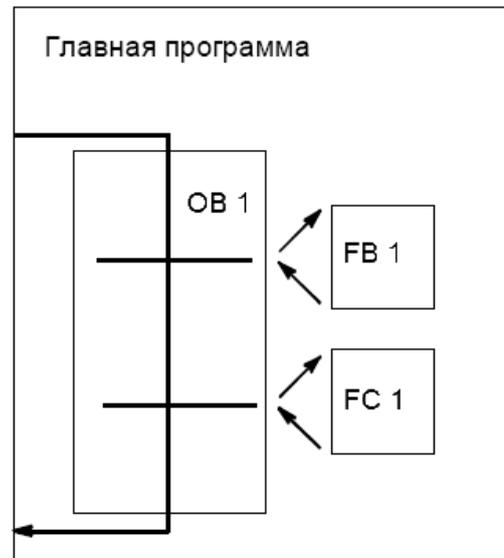
Эти блоки интегрированы в операционную систему CPU, и их код не доступен для пользователя.

Типы программ

Линейное программирование



Структурное программирование

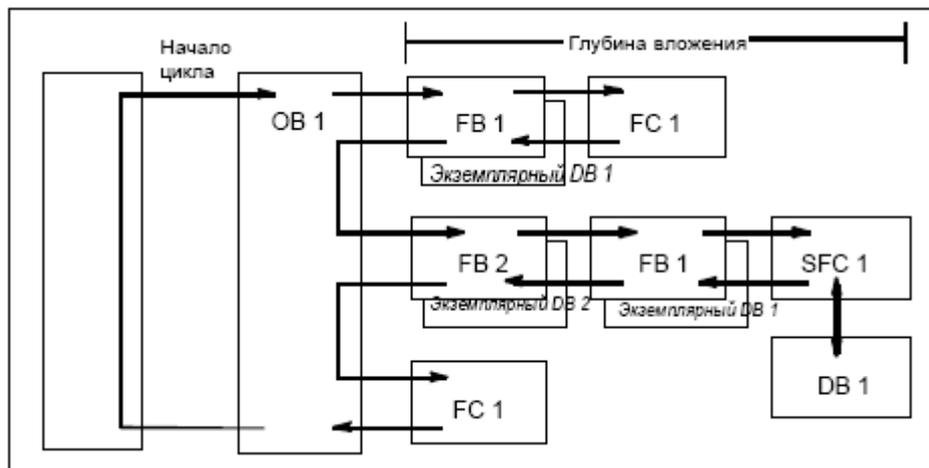


Вы можете записать всю свою программу в одном блоке OB1 (линейное программирование). Это целесообразно только в случае простых программ и требующих мало памяти.

Сложными задачами автоматизации проще управлять, если они разделены на более мелкие задачи, которые отражают технологические функции процесса и могут быть использованы неоднократно. Эти задачи представляются соответствующими программными блоками FC или FB (структурное программирование).

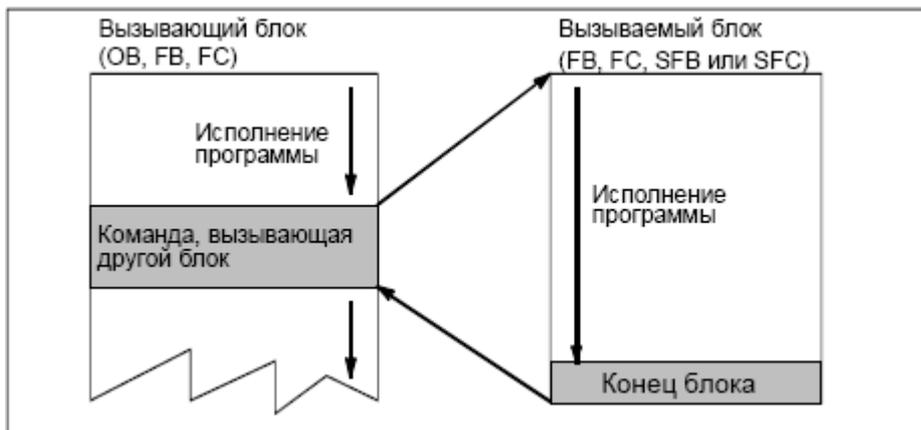
Глубина вложения

Глубина вложения, т.е. количество блоков, которые могут быть вложены друг в друга при выполнении задачи любого приоритетного класса (например, на основе OB1), зависит от конкретного CPU (8 в S7-300; 21 в S7-400).



Вызовы блоков

Вызовы блоков организует пользователь путем размещения команд передачи управления другому блоку (команд вызова). При выполнении команды вызова ОС сохраняет в стеке блоков данные для точки возврата в вызывающий блок (адрес следующей команды, состояние регистров) и передает управление на первую команду вызываемого блока. Как только вызываемый блок выполнен, исполнение прерванного блока возобновляется с команды, следующей за вызовом блока.



Команду “Конец блока” устанавливает редактор блока сразу при создании блока. Она не видна пользователю. Для завершения блока, например, в середине программы, пользователь может применить специальные команды (BE, BEC).

Прерывание программы



Прерывания программы всегда выполняет только ОС.

Если прерывающая программа имеет более высокий приоритет, то ОС прерывает текущую программу и передает управление на первую команду соответствующего блока OB.

На рисунке приведен пример программного прерывания по заданному времени (OB10). Блок OB1 (и программа, которую он вызывает) имеет самый низкий приоритет (=1) и поэтому он может быть прерван любым другим блоком OB.

Приоритеты для блоков OB в S7-300 CPU заданы операционной системой и не могут изменяться пользователем. В S7-400 пользователь может изменить приоритеты блоков прерываний (при задании свойств CPU в утилите HW Config).

Редактирование блоков

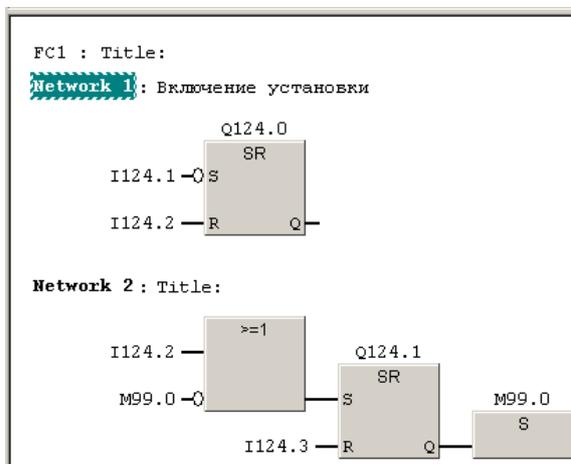
Запуск редактора

Вы можете запустить редактор LAD/STL/FBD через стартовое меню путем выбора *Start -> Simatic -> STEP 7 -> LAD, STL, FBD - Programming S7 Blocks*.

Рекомендуется запускать редактор более быстрым путем:

1. Выбрать объект "Blocks" в окне проекта из SIMATIC Manager.
2. Дважды кликнуть на любой блок для открытия редактора.

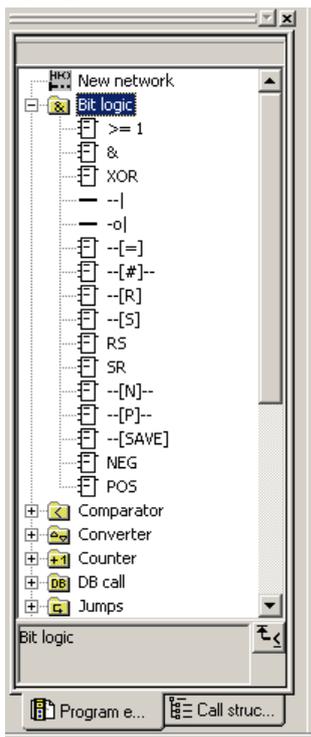
Раздел кода



Раздел кода содержит саму программу, разделенную на отдельные сегменты (network).

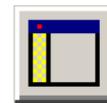
При вводе программы редактор выполняет проверку кода на соответствие синтаксису.

Программные элементы



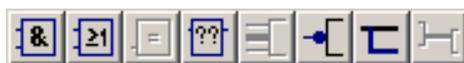
Содержимое окна "Program Elements" зависит от выбранного языка программирования. Для вставки элементов в программу (на позицию, отмеченную курсором) Вы можете дважды кликнуть на нужный объект из библиотеки.

Видимость окна программных элементов Вы можете управлять с помощью соответствующей кнопки инструментальной панели.



Программные элементы для основных двоичных операций размещены на панели инструментов.

Кнопки панели в FBD:



Кнопки панели LAD:



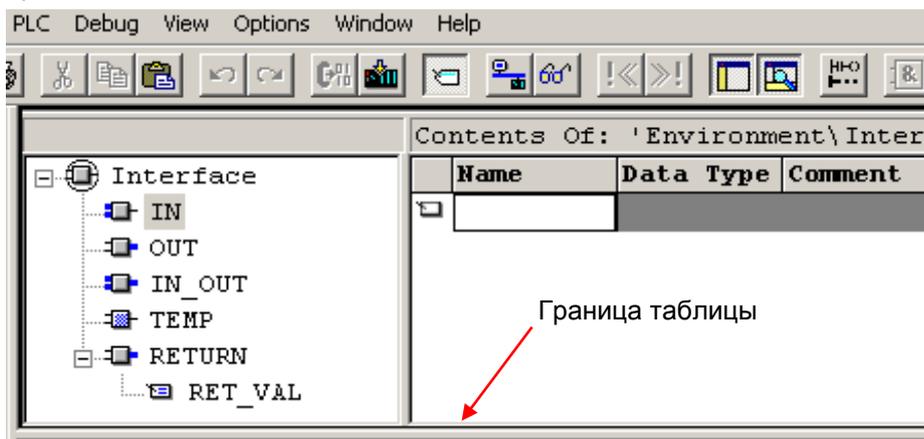
Добавление сегмента

Когда Вы кликните на соответствующую кнопку в панели инструментов, новый сегмент добавляется после текущего сегмента. Если нужно вставить сегмент перед первым, то вначале нужно выделить имя блока.



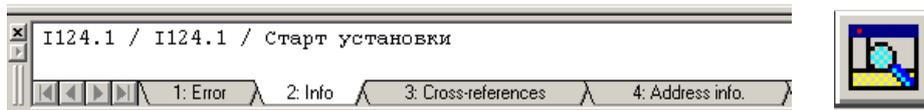
Таблица объявлений

Таблица используется для объявления параметров и локальных переменных. Вы можете управлять границей экранной области таблицы с помощью меню *Window->Move Split*.



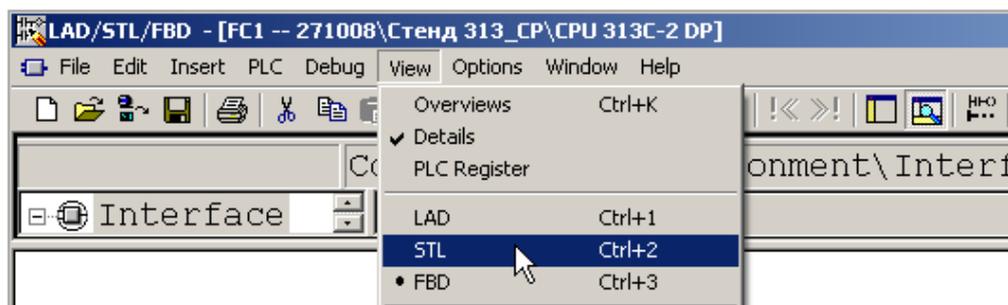
Окно Details

Окно Details позволяет получить дополнительную информацию в зависимости от выбранной закладки. Управление окном – по соответствующей кнопке меню.



Выбор языка программирования

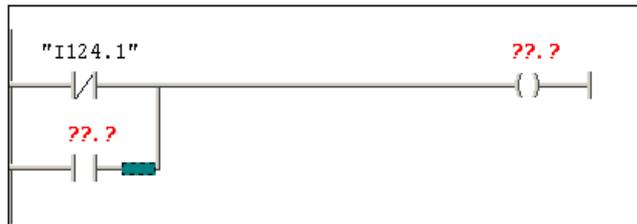
Редактор LAD/STL/FBD предлагает пользователю 3 стандартных для PLC языка программирования. Переключение между языками (формой представления программы) производится легко, как показано на рисунке.



Языки **LAD** (Ladder Logic) и **FBD** (Function Block Diagram) основаны на графическом способе представления программы в виде соответствующих элементов. Язык **STL** (Statement List) – это язык команд S7 CPU (ассемблер) и позволяет реализовать программу максимально эффективно, с использованием всех ресурсов CPU. Основной недостаток STL – необходимость помнить мнемоники команд и трудность восприятия программы. Сегменты программы, созданные на STL, не всегда преобразуются в LAD/FBD (причины: более одной логической операции в сегменте; отсутствие команд NOP 0 и т.д.).

Программирование в LAD

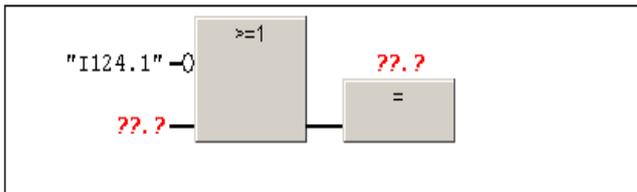
Network 1: Switch ON/OFF



Программирование в LAD подобно рисованию релейно-контактной схемы. Для логических операций, например, используются такие элементы, как контакты и катушки. Этот язык программирования рассчитан на тех, кто привык работать с электрическими схемами

Программирование в FBD

Network 1: Switch ON/OFF

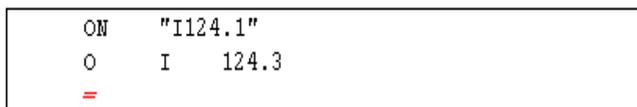


технологов.

Язык FBD представляет взаимосвязь элементов-прямоугольников. Символ в прямоугольнике указывает на функцию (например, & – логическая функция И). Этот язык рассчитан на программистов, инженеров - схемотехников и

Программирование в STL

Network 1: Switch ON/OFF



Каждая строка в программе на языке STL – это отдельная команда. Она содержит саму инструкцию и данные (операнд). В STL программе не требуется разбиение ее на сегменты.

Сохранение блока

Когда Вы закончили редактирование блока в активном окне, Вы можете сохранить его в проекте, выбирая пункт меню *File -> Save* или через кнопку.



Загрузка блоков в CPU

Вы можете загрузить блок из активного окна редактора непосредственно в контроллер. Для этого Вы можете использовать меню *File -> Download* или кнопку.



Открытие Online блока

Через кнопку меню Вы можете открыть соответствующий блок из CPU. Если один и тот же блок открыт в режиме Offline и в режиме Online, то редактировать можно только тот блок, который был открыт первым.



Отладка программы

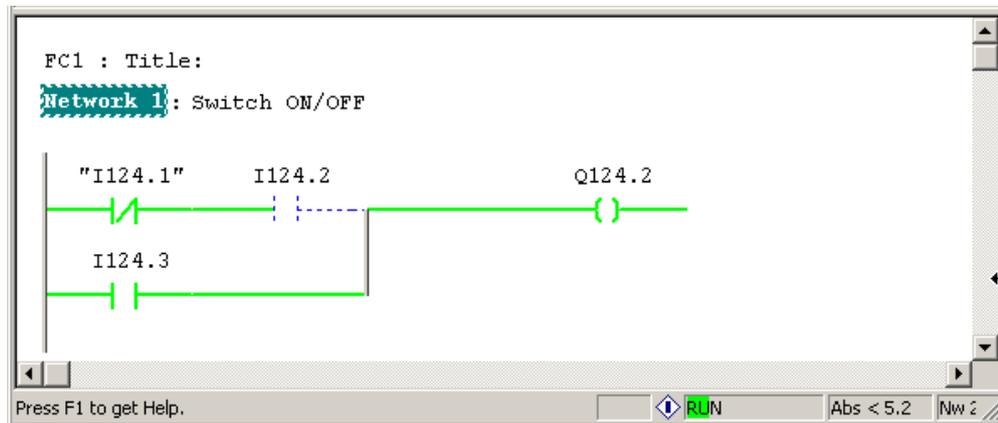
Имеется два варианта для активации/деактивации тестовой функции "Monitor": кликнуть на кнопку " Monitor " или выбрать пункт меню *Debug -> Monitor*.



Режим "Monitor" позволяет оценить выполнение фрагмента программы для выделенного сегмента, а также сегментов, расположенных ниже в активном окне. Статус программы отображается различно в зависимости от выбранного языка программирования (LAD/STL/FBD).

Вы можете изменить толщину и цвет линий путем выбора меню *Options -> Customize -> LAD/FBD*.

Когда активирована тестовая функция, Вы не можете изменить язык программирования (LAD/FBD/STL), в котором отображается блок.



Изменение программы

Вы можете корректировать блоки, открытые или в online, или в offline режимах. Если Вы не хотите тестировать программу сразу, Вы можете просто сохранить изменения на жестком диске, чтобы тестировать ее позже. Старая версия блока в этом случае стирается.

Если Вы делаете исправления и не хотите обновлять исходную версию программы до проверки ее работы, то Вы можете загрузить блоки только в CPU, не сохраняя их на жестком диске PG. При закрытии окна редактор предлагает сохранить блок в проект, если в тексте блока было что-нибудь изменено.



Режимы вставки и замены

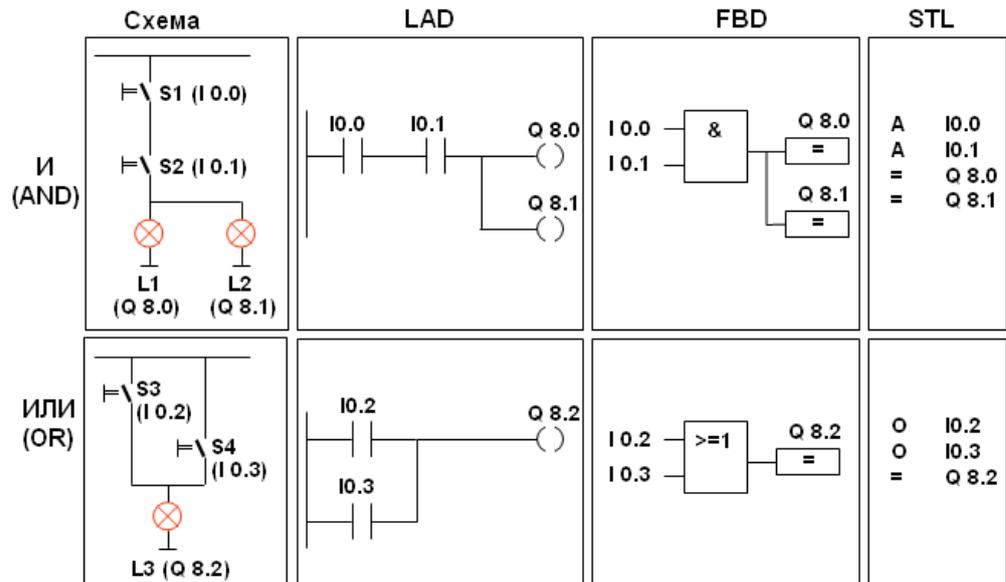
Нажимая клавишу "Insert", Вы переключаете между собой режимы вставки и замены. По умолчанию установлен режим вставки. В режиме замены удобно производить замену одного элемента на другой (без изменения входов и выходов).

В правом нижнем углу окна редактора, в строке статуса, Вы видите текущий режим вставки/замены.

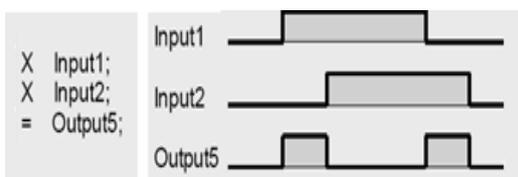
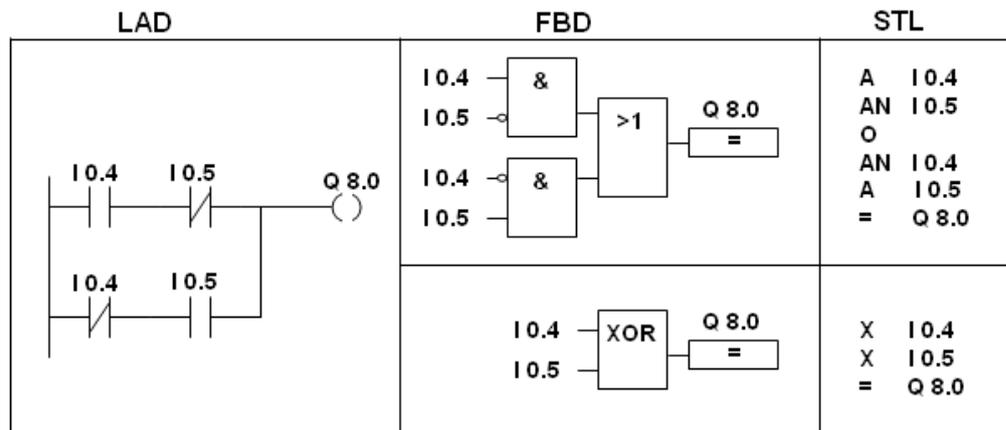


Двоичные операции

Операции: И, ИЛИ (AND, OR)



Исключающее ИЛИ (Exclusive OR)



Исключающее ИЛИ связывает два двоичных сигнала и возвращает результат RLO, равный "1", если оба эти сигнала имеют разные значения. Если оба сигнала имеют одинаковое значение, то RLO равен "0".

Если один из 2-х операндов элемента "Исключающее ИЛИ" равен 1, то выход равен инверсии 2-го операнда.

Результат логической операции

| FC1 : Title: | | | | RLO | STA |
|----------------------------------|----------|--|--|-----|-----|
| Network 1 : Switch ON/OFF | | | | | |
| AN | "I124.1" | | | 1 | 0 |
| A | I 124.2 | | | 0 | 0 |
| O | I 124.3 | | | 1 | 1 |
| = | Q 124.2 | | | 1 | 1 |

Результат логической операции (RLO) – это отдельный бит слова состояния, за статусом которого Вы можете наблюдать в режиме "Monitor" (STL-представление). Формирование RLO происходит при выполнении команд AND, OR, XOR в соответствии с состояниями операндов (STA) и правилами булевой алгебры (таблиц истинности).

Операции присвоения, установки и сброса

Команды присвоения, установки и сброса используют RLO для изменения операнда, но всегда оставляют RLO без изменения. Поскольку результат выполнения этих команд зависит от RLO, эти команды называются команды условий (conditional statement).

Команда присвоения (=) копирует текущее значение RLO в операнд.

Команда установки (S) записывает "лог. 1" в операнд, если RLO=1. Если RLO=0, операнд не изменяется.

Команда сброса (R) записывает "лог. 0" в операнд, если RLO=1. Если RLO=0, операнд не изменяется.

Первичный опрос (First check)

Первая команда опроса сигнала, следующая за командой условий, называется "первичным опросом" ("first check"). Первичный опрос всегда соответствует началу логической операции ("старое значение" RLO теряется).

Правильный подход в программировании

...

= Q 15.3

O I 18.5 первая AND
A I 21.7 функция

= Q 15.4

Результат логической операции (RLO) для команды первичного опроса берется непосредственно из результата опроса самого сигнала и не зависит от типа логической функции.

Правильный подход описания логической функции означает, что команда первичного опроса должна быть идентична требуемой функции.

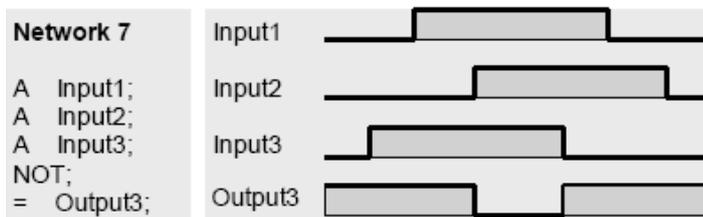
A I 18.4 вторая AND
A I 21.6 функция

= Q 15.5

Последовательность операторов, представленная на рисунке, представляет собой две AND-функции, из которых вторая функция (в которой обе проверки имеют логику AND) запрограммирована более правильным и более предпочтительным способом.

...

Инвертирование результата логической операции (команда NOT)



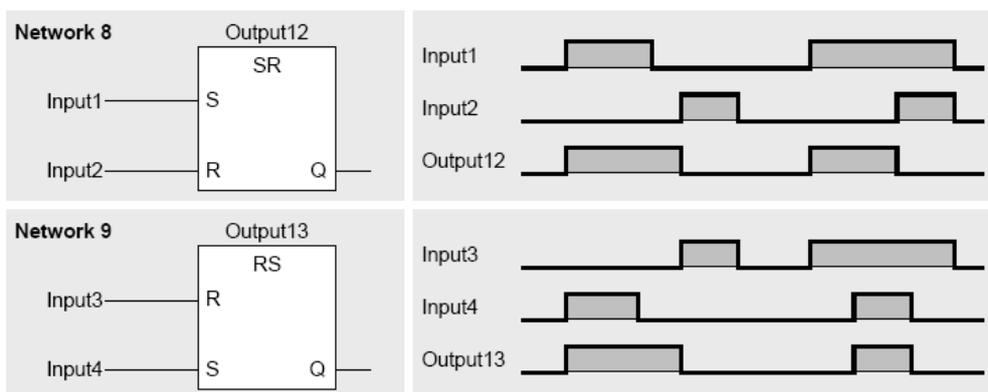
Вы можете использовать оператор **NOT** для инвертирования результата выполнения логической функции.

Триггер

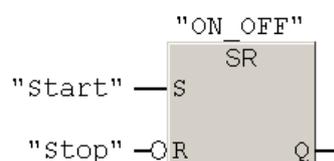
В LAD и FBD имеются 2 элемента для функций триггера с приоритетом установки и приоритетом сброса. Приоритет сброса (или установки) означает, что ячейка памяти сбрасывается в 0 (или устанавливается в 1), если RLO равен 1 одновременно на 2-х входах.

В STL приоритет имеет та инструкция, которая была запрограммирована последней.

| | LAD | FBD | STL |
|-------------------------|-----|-----|--|
| С приоритетом сброса | | | <pre>A I1.2 S M0.0 A I1.3 R M0.0 A M0.0 = Q9.3</pre> |
| С приоритетом установки | | | <pre>A I1.3 R M0.0 A I1.2 S M0.0 A M0.0 = Q9.3</pre> |



Функция памяти с приоритетом сброса является «нормальной» формой функции для работы с памятью, так как состояние сброса (сигнальное состояние «0») является обычно безопаснее или менее рискованным.



Пример: включение установки выполняется по кнопке "Start", а выключение по кнопке "Stop".

Аварийные сигналы обычно имеют активный "0" (при обрыве провода сигнал на входе модуля будет также "0"), поэтому вход сброса должен иметь приоритет и обрабатываться по "0".

Коннектор

Коннекторы (midline outputs) являются промежуточными буферами в контактном или функциональном планах. RLO, действительный для коннектора, хранится в операнде над этим коннектором. Этот операнд может быть снова опрошен в другой точке программы.

Для использования коннектора только внутри блока желательно использовать биты временных локальных данных (предварительно объявляются в таблице объявлений блока).

| | | STL |
|-----|--|--|
| LAD | | <pre>A I 1.0 A I 1.1 = M 0.0 A M 0.0 A I 2.0 A I 2.1 NOT = M 1.1 A M 1.1 = Q 4.0</pre> |
| FBD | | |

Функции оценки (выделения) фронта RLO

Фронт является **положительным** (возрастающим), когда сигнал меняется с «0» на «1». В противоположном случае фронт считается **отрицательным** (падает).

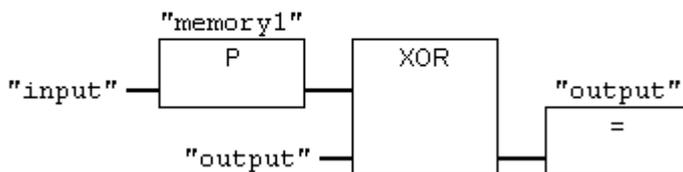
Чтобы обнаружить фронт сигнала, команды FP и FN используют бит памяти (M, DB) для сохранения текущего RLO. При каждой новой проверке наличия фронта сигнала CPU сравнивает текущее значение RLO с состоянием FP/FN бита. Фронт сигнала будет обнаружен, если эти два сигнала будут иметь различные состояния.

| LAD | FBD | STL |
|-----|-----|---|
| | | <pre>A I 1.0 A I 1.1 FP M1.0 = M8.0</pre> |
| | | <pre>A I 1.0 A I 1.1 FN M1.1 = M8.1</pre> |

Пример: делитель частоты.

Делитель частоты, или счетчик, имеет один вход (input) и один выход (output).

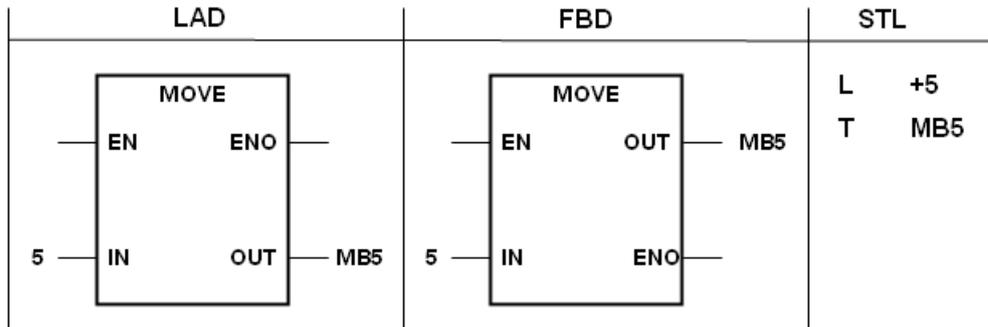
Network 1: Делитель частоты



На выходе делителя частоты входная частота уменьшается в 2 раза.

Загрузка и передача данных.

Пример показывает пересылку константы (десятичное число 5) в ячейку MB5.

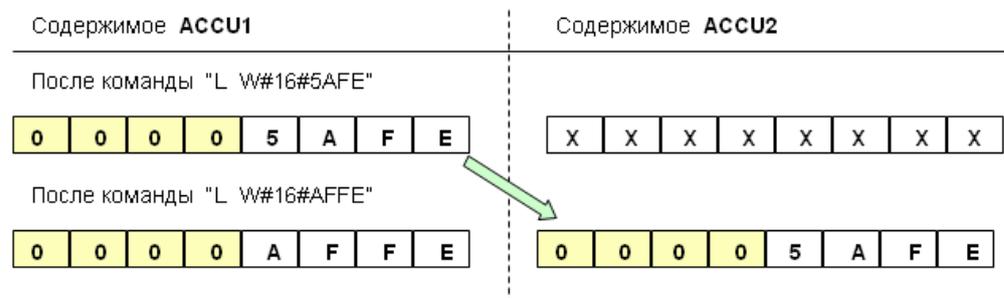


Примечание: вход "EN" (Enable) отображается только в LAD / FBD и может быть использован в качестве логического условия для передачи данных. Выход "ENO" (Enable Output) сигнализирует о выполнении пересылки и может быть использован как входной операнд при составлении логических функций.

Данные пересылаются через **ACCU 1** (аккумулятор 1). **ACCU 1** является основным регистром CPU.

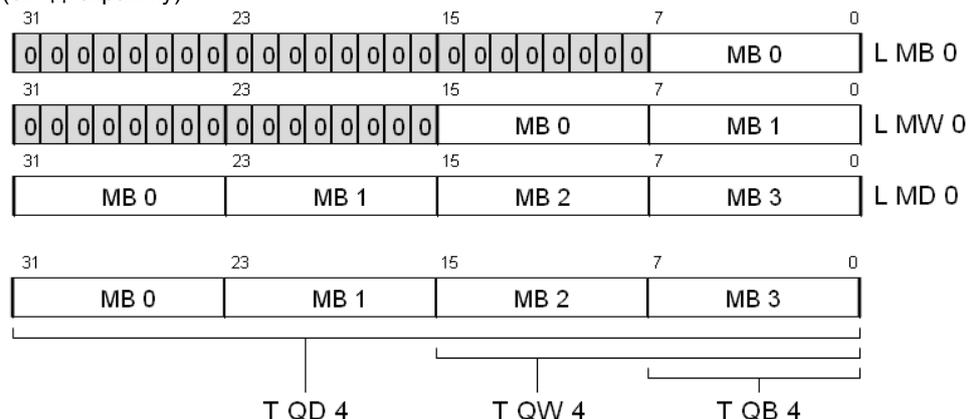
Когда выполняется инструкция **Load**, старое значение ACCU 1 сначала сдвигается в **ACCU 2**, а затем новое значение записывается в **ACCU 1**.

Все аккумуляторы имеют по 32 разряда.



При загрузке в ACCU 1 байта или слова старшие разряды ACCU 1 обнуляются.

Когда выполняется инструкция **Transfer**, содержимое ACCU 1 сохраняется. Одна и та же информация, таким образом, может быть передана по разным адресам. Если передается только байт, то используются только восемь младших разрядов ACCU 1 (см. диаграмму).



Таймеры

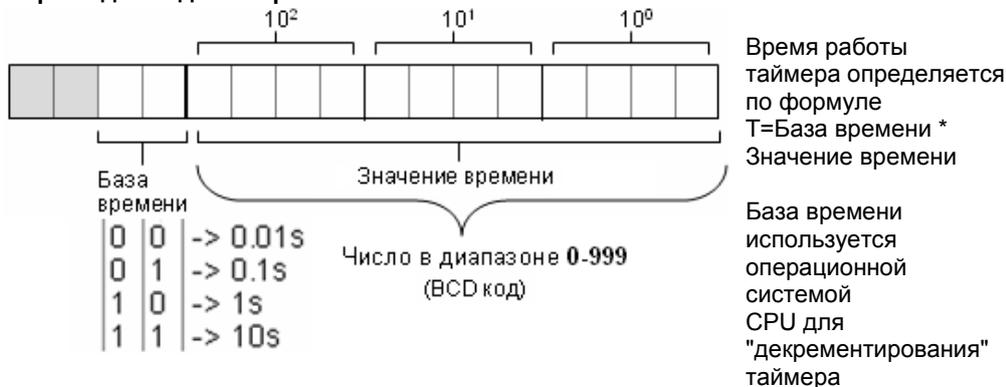
Таймеры представляют системную область памяти CPU, имеющую адресацию T0 – Txx, где xx- максимальный номер таймера (определенный для каждого CPU). Каждый таймер занимает 16-битное слово. Язык STEP7 (как и STEP5) имеет 5 инструкций для различных режимов работы таймеров.

| | | | | |
|---|-----------|-------------------|----------------|--|
| Таймер запускается как: | Ha STL | Ha LAD/ FBD | Сигнал запуска | |
| Таймер с управляемым импульсом (импульсный) (Pulse timer) | SP | S_PULSE | | |
| Таймер с расширенным импульсом (Extended pulse timer) | SE | S_PEXT | | |
| Таймер с задержкой включения (On-delay timer) | SD | S_ODT | | |
| Таймер с задержкой включения с памятью (Retentive on-delay timer) | SS | S_ODTS | | |
| Таймер с задержкой выключения (Off-delay timer) | SF | S_OFFDT | | |

Задание временных параметров таймера

При запуске таймера из **ACCU1** выбирается время запуска t (running time). Задание длительности работы таймера с помощью константы или переменной задается в формате S5T# (S5TIME#), которые должны быть загружены в **ACCU1**. Диапазон времени задается от S5T#10ms до S5T#2h46m30s (что соответствует диапазону времени, равному 9990 с).

Формат для задания времени

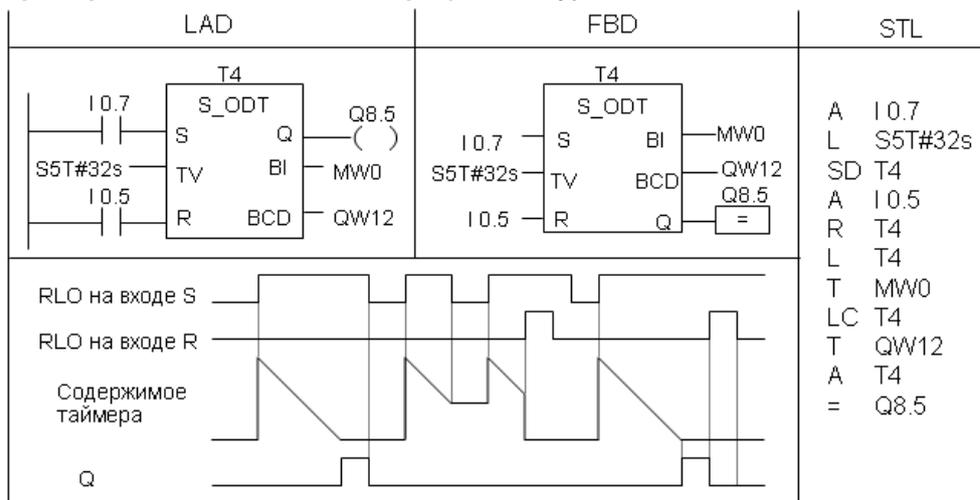


Наименьшее возможное значение для "временной базы" ("time base") обеспечивает более точное исчисление промежутков времени с помощью таймера.

Пример: необходимо задать для таймера отрезок времени, равный 1 с. Можно использовать одно из трех значений: с базой времени 1с (2001hex), с базой времени 100 мс (1010 hex) и 10 мс (0100 hex).

Последний из трех вариантов является предпочтительным.

Пример использования таймера (ON Delay)



Запуск

Таймер (T4) запускается, когда признак RLO на входе "S" (сигнал I0.7) изменяется из "0" в "1". Если состояние сигнала на входе S равно 1, таймер выполняет обратный отсчет времени с величины, задаваемой на входе TV.

Сброс

Когда RLO на входе сброса "R" равно "1", текущее значение времени обнуляется, а выход Q сбрасывается.

Цифровой выход

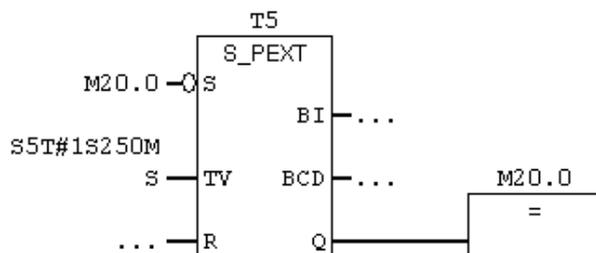
Текущее значение времени может быть прочитано как двоичное число на выходе BI и как BCD -число на выходе BCD. Текущее значение времени равно начальному значению на входе TV минус величина времени, которое прошло с тех пор, как таймер был запущен.

Двоичный выход

Сигнал на выходе "Q" устанавливается в "1" когда таймер досчитает до 0, а сигнал на входе "S" будет равен "1".

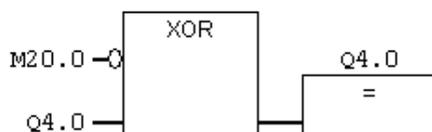
Генераторы на таймерах

Network 3 : Генератор



На рисунке приведен пример генератора импульсов (выход Q4.0) с периодом T=2,5с. Ячейка M20.0 перезапускает таймер T5 через каждые 1s 250ms (половина от заданного периода). Функция XOR используется с целью получения меандра. В момент перезапуска таймера (когда M20.0=0) происходит инверсия выхода Q4.0.

Network 4 : Выравнивание импульсов (меандр)



Таймеры: битовые инструкции

Все таймеры могут также использоваться и с помощью компактных блочных элементов. Запуск и опрос таймеров происходит в разных частях блока и даже в разных блоках. Для получения текущего значения таймера можно применить элемент MOVE.

| LAD | FBD | STL |
|-------------------|-----|-----------------------------------|
| <p>Network 1:</p> | | <pre>A I 0.0 L S5T#5s SD T4</pre> |
| <p>Network 2:</p> | | <pre>A T4 = Q8.0</pre> |
| <p>Network 3:</p> | | <pre>A I 0.1 R T4</pre> |

IEC-таймеры

IEC-таймеры (таймеры, функционирование которых соответствует международному стандарту IEC 1131) встроены в операционную систему CPU как системные функциональные блоки (блоки SFB).

В S7- CPU доступны следующие таймеры: импульсный таймер (SFB 3), таймер задержки включения (SFB 4); таймер задержки выключения (SFB 5).

Применение IEC-таймеров рассматривается в учебном курсе S7-PROF2.

Счетчики

Счетчики представляют системную область памяти CPU, имеющую адресацию C0 – Cxx, где xx- максимальный номер счетчика (определенный для каждого CPU). Каждый счетчик занимает 16-битное слово.

| LAD | FBD | STL |
|-----|-----|---|
| | | <pre>A I 0.4 CU C5 A I 0.5 CD C5 A I 0.3 L C#20 S C5 A I 0.7 R C5 L C5 T MW4 LC C5 T QW12 A C5 = Q8.3</pre> |

Счет на увеличение

Когда RLO на входе **CU** изменяется из "0" в "1", текущее значение счетчика увеличивается на 1. При достижении значения 999 (верхняя граница) счетчик больше не будет изменять своего состояния.

Счет на уменьшение

Когда RLO на входе **CD** изменяется из "0" в "1", текущее значение счетчика уменьшается на 1. При достижении 0 (нижняя граница) счетчик больше не будет изменять своего значения.

Установка счетчика

Когда RLO на входе **S** изменяется из "0" в "1", счетчик принимает значение, заданное на входе **CV**.

Сброс счетчика

Когда на входе **R** RLO = 1 счетчик устанавливается в 0. Если условие сброса выполняется, то не возможна ни установка счетчика, ни процедура счета.

Значение для счетчика

Значение для счетчика (0...999) задается на входе **CV**: как константа (C#...) или как переменная в BCD - формате.

Сигнал состояния счетчика

Состояние счетчика может быть проверено через выход **Q**:

Q = 0, если счетчик = 0; **Q = 1**, если счетчик ≠ 0.

IEC-счетчики

IEC-счетчики (счетчики, функционирование которых соответствует международному стандарту IEC 1131) встроены в операционную систему CPU как системные функциональные блоки (блоки SFB).

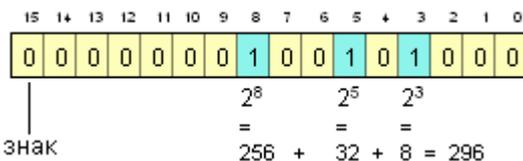
В S7-CPU доступны следующие счетчики: счетчик прямого счета (SFB 0), счетчик обратного счета (SFB 1); счетчик прямого/обратного счета (SFB 2).

Применение IEC-счетчиков рассматривается в учебном курсе S7-PROF2.

Функции для обработки чисел

Данные функции используются для обработки численных значений преимущественно типов данных INT, DINT и REAL.

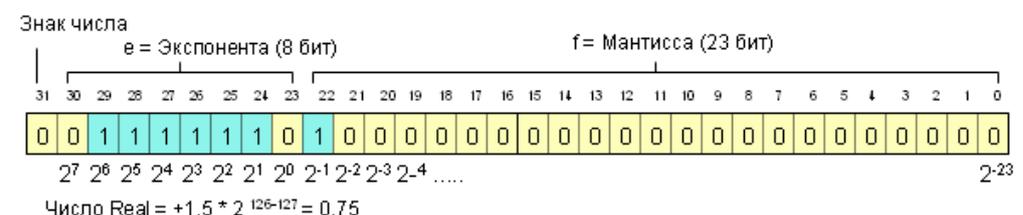
Форматы чисел



Integer (INT) – 16-разрядный формат представления чисел в диапазоне -32768 до 32767

Формат **Double Integer (DINT)** представляет собой 32-разрядное число в диапазоне -2147483648 – 2147483647.

Формат **REAL** представляет собой 32-разрядное число в диапазоне -1.175495e-38 - +3.402823e+38. Общий формат для числа Real = (Знак) • (1.f) • (2e-127)



Функции сравнения

Функции сравнения обеспечивают выполнение операции сравнения (см. таблицу ниже) двух численных значений, одно из которых находится в аккумуляторе 1 (**ACCU1**), а второе находится в аккумуляторе 2 (**ACCU2**). Сравнение производится методом вычитания **ACCU1** из **ACCU2** (при этом, содержимое аккумуляторов не изменяется).

| Операция сравнения | Типы данных | | |
|----------------------|-------------|------|------|
| | INT | DINT | REAL |
| Равно | ==I | ==D | ==R |
| Не равно | <>I | <>D | <>R |
| Больше чем | >I | >D | >R |
| Больше чем или равно | >=I | >=D | >=R |
| Меньше чем | <I | <D | <R |
| Меньше чем или равно | <=I | <=D | <=R |

Арифметические функции

Арифметические функции обеспечивают выполнение базовых арифметических операций с двумя числовыми значениями, одно из которых находится в аккумуляторе 1 (**ACCU1**), а второе находится в аккумуляторе 2 (**ACCU2**). Результат арифметической операции записывается в **ACCU1**.

| Арифметическая функция | Типы данных | | |
|--|-------------|------|------|
| | INT | DINT | REAL |
| Сложение (Addition) | +I | +D | +R |
| Вычитание (Subtraction) | -I | -D | -R |
| Умножение (Multiplication) | *I | *D | *R |
| Деление (Division with quotient as result) | /I | /D | /R |
| Остаток от деления (Division with remainder as result) | - | MOD | - |

Пример применения операции сложения:

| LAD | FBD | STL |
|-----|-----|--|
| | | <pre>L MW4 L MW10 +I T MW6</pre> |

Добавление констант к содержимому аккумулятора 1

Операция добавления константы более предпочтительна для вычислений с адресуемыми данными, потому что по сравнению с арифметическими функциями она не влияет ни на содержимое остальных аккумуляторов, ни на биты состояния.

Примеры:

- + **B#16#2A** // добавление байтовой константы (2A)
- + **25** // добавление константы формата Integer (25)
- + **-100** // вычитание константы формата Integer (-100)
- + **L#1000** // добавление константы формата DINT(1000)

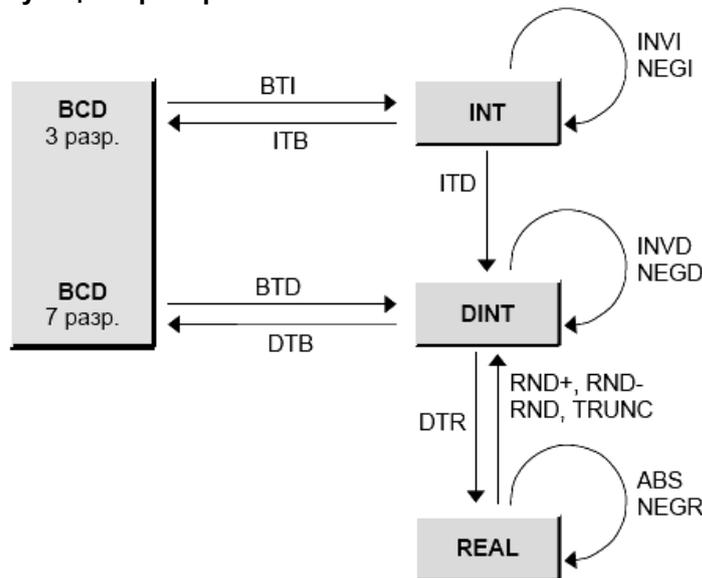
Операции декрементирования и инкрементирования

Операции декрементирования и инкрементирования меняют значение в **ACCU1**. Содержимое аккумулятора уменьшается (декрементируется) или увеличивается (инкрементируется) на число, определенное в параметре команды. Этот параметр может принимать значение в диапазоне от 0 до 255. При этом изменяется только значение в младшем байте аккумулятора.

Примеры:

- DEC 5 // уменьшение ACCU1 на 5
- INC 4 // увеличение ACCU1 на 4

Функции преобразования



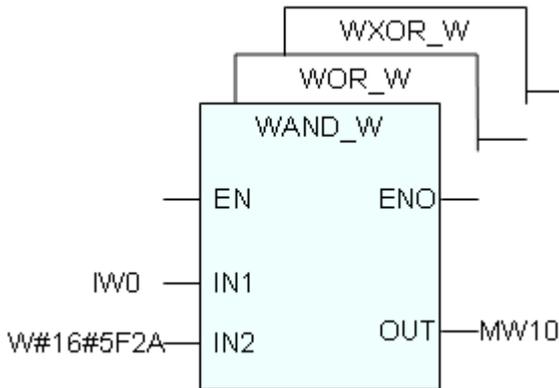
Функции преобразования конвертируют типы данных переменных. Исходное число находится в **ACCU1**. Результат преобразования записывается также в **ACCU1**.

При преобразовании, например, числа в BCD-коде в формат INT (DINT) при наличии псевдотетрады (код больше 9) возникает ошибка с вызовом организационного блока OB121.

| Входное значение . . . REAL | Результат | | | |
|--------------------------------|-----------|------|------|-------|
| | RND | RND+ | RND- | TRUNC |
| 1.00000001 | 1 | 2 | 1 | 1 |
| 1.00000000 | 1 | 1 | 1 | 1 |
| 0.99999995 | 1 | 1 | 0 | 0 |
| 0.50000005 | 1 | 1 | 0 | 0 |
| 0.50000000 | 0 | 1 | 0 | 0 |
| 0.49999996 | 0 | 1 | 0 | 0 |

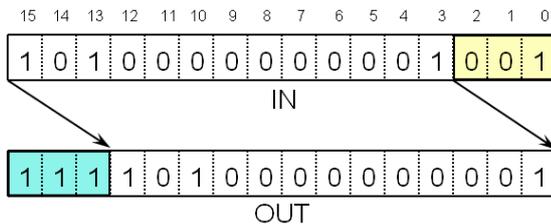
Таблица показывает различия (на примере положительных чисел) между командами преобразования из числа REAL в целое число формата DINT.

Операции цифровой логики



Логические функции для слов данных (Word Logic) позволяют побитно комбинировать значение, находящееся в **ACCU1** с константой или с содержимым ACCU2. Результат выполнения операции сохраняется в **ACCU1**. Логические функции могут выполняться как для данных формата Word, так и для данных формата Double Word.

Операции сдвига



Функции сдвига сдвигают влево или вправо содержимое **ACCU1** бит за битом. На рисунке поясняется выполнение операции SSI 3 (сдвиг вправо числа со знаком на 3 разряда).

Информация об основных операциях сдвига приведена в таблице.

| Функции сдвига | Word (Слово) | | Doubleword (Двойное слово) | |
|---|-------------------------|--------------------|----------------------------|--------------------|
| | Число поз. как параметр | Число поз. в ACCU2 | Число поз. как параметр | Число поз. в ACCU2 |
| Сдвиг влево (Shift left) | SLW n | SLW | SLD n | SLD |
| Сдвиг вправо (Shift right) | SRW n | SRW | SRD n | SRD |
| Сдвиг со знаком (Shift with sign) | SSI n | SSI | SSD n | SSD |
| Циклический сдвиг влево (Rotate left) | - | - | RLD n | RLD |
| Циклический сдвиг вправо (Rotate right) | - | - | RRD n | RRD |

Обработка аналоговых сигналов

Для обработки аналоговых величин в CPU, они должны быть преобразованы в цифровую форму. Это делается с помощью АЦП (аналого-цифровой преобразователь) в модуле аналоговых входов. Сигналы каждого входного канала, приходящего на один модуль преобразуются последовательно. Результат преобразования записывается в память результатов и остается там до тех пор, пока не переписывается новой величиной. Преобразованная аналоговая величина может быть прочитана с помощью инструкции загрузки: "L PIW x", где x – десятичный адрес аналогового канала. Инструкция передачи "T PQW x" используется для записи аналоговой величины, вычисленной программой, в модуль аналогового выхода, где ЦАП (цифро-аналоговый преобразователь) преобразует их в стандартные аналоговые сигналы.

Адресация аналоговых модулей

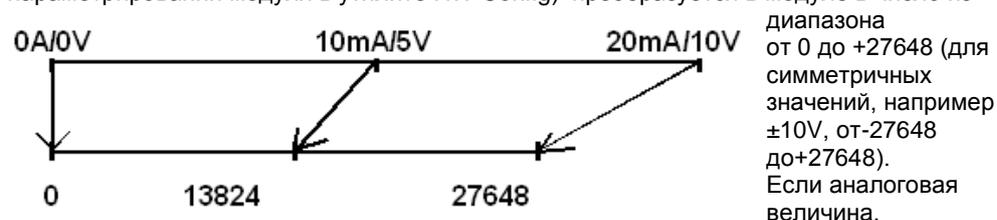
По умолчанию аналоговые модули получают адреса в соответствии с их размещением в монтажной стойке. На рисунке приведена адресация модулей для систем S7-300 (для 2-х стоек).

| Слот # | | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|--------|----------------------|--------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Rack 1 | Модуль блока питания | IM (Receive) | 384 по 398 | 400 по 414 | 416 по 430 | 432 по 446 | 448 по 462 | 464 по 478 | 480 по 494 | 496 по 510 |
| | CPU и блок питания | IM (Send) | 256 по 270 | 272 по 286 | 288 по 302 | 304 по 318 | 320 по 334 | 336 по 350 | 352 по 366 | 368 по 382 |

Каждый аналоговый канал занимает 2 байта. Таким образом, если аналоговый модуль имеет адрес 304, то 2-й канал этого модуля имеет адрес PIW306 (для выхода - PQW306).

Кодирование аналоговых величин

Аналоговая величина с номинальным диапазоном тока / напряжения (выбирается при параметрировании модуля в утилите HW Config) преобразуется в модуле в число из



превышает номинальное значение, то аналоговый модуль может выдать значение от 27648 до 32767. Считается, что при значении 32767 наступает переполнение.

Масштабирование

Масштабирование – математическое преобразование числовой величины в значение из физического диапазона. Для униполярных сигналов масштабирование выполняется по формуле:

$$OUT = [(FLOAT(IN) / 27648) * (HI_LIM - LO_LIM)] + LO_LIM,$$

где IN – входное числовое значение аналоговой величины

HI_LIM и LO_LIM – верхний и нижний предел для физического диапазона.

Для обеспечения точности все вычисления должны проводиться с числами в формате Real.

```

L   PIW 288
ITD
DTR
L   2.7648e+4
/R
L   9.000e+2
*R
L   1.000e+2
+R
T   MD10

```

Пример:

Аналоговый сигнал 0-10V с адресом PIW288 должен быть преобразован в физическую величину из диапазона 100-1000.

На рисунке слева приведено решение на STL.

Для управления выходной аналоговой величиной выполняются преобразования, обратные предыдущему случаю.

```

L   MD20
L   1.000e+2
-R
L   9.000e+2
/R
L   2.7648e+4
*R
RND
T   PQW 288

```

Пример:

Величина из диапазона 100-1000 хранится в ячейке MD20 в формате REAL. Необходимо произвести вывод данной величины в виде аналогового выхода 0-10V по адресу PQW 288.

На рисунке слева приведено решение на STL.

Измерение температуры

Температура измеряется с помощью термосопротивлений и термопар. Результатами для номинальной области являются числа из десятикратного температурного диапазона.

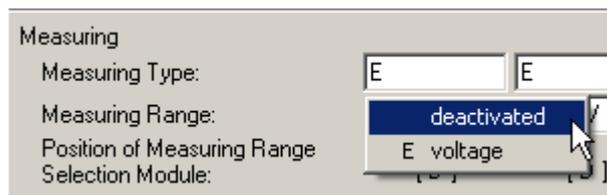
Например, для датчика Pt 100 и его температурного диапазона -200 до + 850 °C получаем (при чтении аналогового канала) величину из диапазона -2000 до + 8500.

Время преобразования

Время преобразования для различных модулей приводится в руководстве по модулям.

Например, для модуля SM334 время преобразования для всех каналов не более 5 ms.

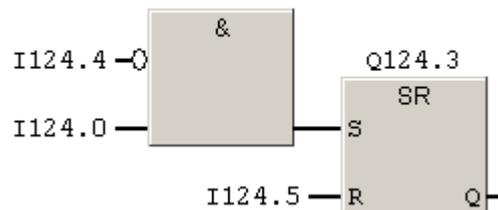
Если канал не используется, то при параметрировании модуля отметьте его как "Deactivated". Это позволит уменьшить время преобразования.



Символьная адресация

Абсолютная адресация

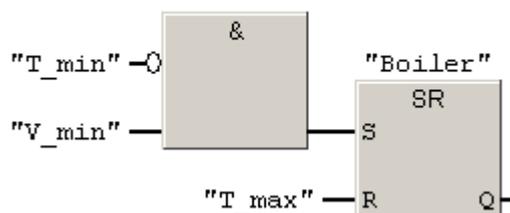
Network 4: Нагреватель



При абсолютной адресации Вы указываете фактический адрес операнда, например, I 124.0. Программа в этом случае не отражает физического смысла операции, и, как следствие, хуже понимается. Абсолютный адрес может быть заменен символьным именем по Вашему выбору.

Символьная адресация

Network 4: Нагреватель



При символьной адресации Вы используете вместо абсолютного адреса символическое имя (например, T_min). При вводе имен нет необходимости вписывать кавычки. Редактор делает это за Вас. Символьная адресация обеспечивает лучшую читаемость программы.

Символьная таблица

Вы открываете таблицу символов в LAD/STL/FBD редакторе с помощью команды меню *Options* -> *Symbol Table*. Вы можете также открыть символьную таблицу из SIMATIC Manager: выберите программу в левом подокне проекта и дважды щелкните на объекте "Symbols".



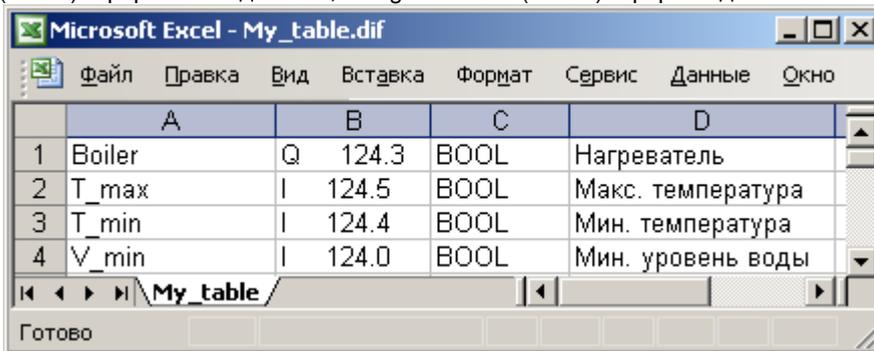
Структура таблицы

Символьная таблица содержит столбцы для символики (Symbol), адреса (Address), типа данных (Data type) и комментария (Comment). Символьные имена, объявленные в символьной таблице, применимы во всех частях программы и называются **глобальными** переменными. Вы можете присваивать имена (не более 24 символов) ячейкам памяти из областей I, Q, M, T, C, периферии (PI/PQ), а также имена программным блокам. Для облегчения программирования и улучшения документирования программы лучше использовать комбинацию кратких имен (до 10 знаков) и длинных комментариев.

| | Symbol | Address | Data type | Comment |
|---|----------------|---------|-----------|-------------------|
| 1 | Boiler | Q 124.3 | BOOL | Нагреватель |
| 2 | CYCL_EXC | OB 1 | OB 1 | Cycle Execution |
| 3 | Hot water tank | FC 1 | FC 1 | |
| 4 | T_max | I 124.5 | BOOL | Макс. температура |
| 5 | T_min | I 124.4 | BOOL | Мин. температура |
| 6 | V_min | I 124.0 | BOOL | Мин. уровень воды |

Функции Export/Import

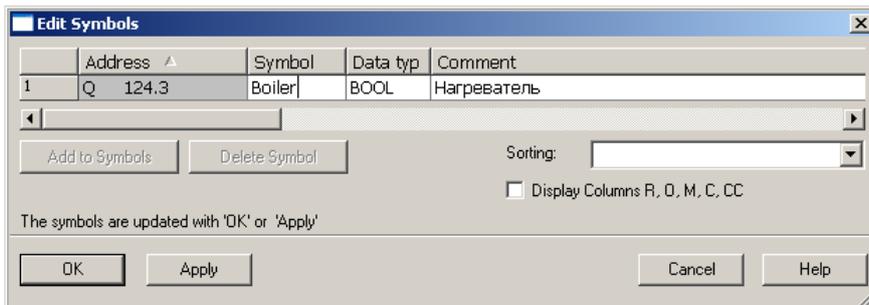
Команда меню *Symbol Table* -> *Export* позволяет Вам преобразовать символьную таблицу в различные файловые форматы: ASCII Format (*.ASC) - текстовый формат; Data Interchange Format (*.DIF) - формат электронных таблиц; System Data Format (*.SDF) - формат баз данных; Assignment List (*.SEQ) - формат для STEP 5.



Команда меню *Symbol Table* -> *Import* позволяет Вам импортировать символьные таблицы, подготовленные в других программах, например, в Excel.

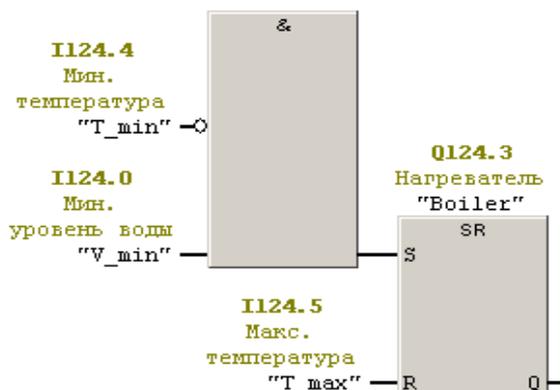
Редактирование символов в редакторе

Команда меню *Options* -> *Edit Symbols* или щелчок правой кнопки мыши на адресе с последующим выбором пункта выпадающего меню *Edit Symbol* позволяет Вам назначать символические имена для абсолютных адресов непосредственно при редактировании программы. Имена автоматически вводятся в символьную таблицу.



Символьная информация в редакторе

Network 4: Нагреватель



С помощью кнопки меню или команды *View* -> *Display with* -> *Symbolic Representation* Вы можете переключаться между абсолютным и символьным представлением адресов.

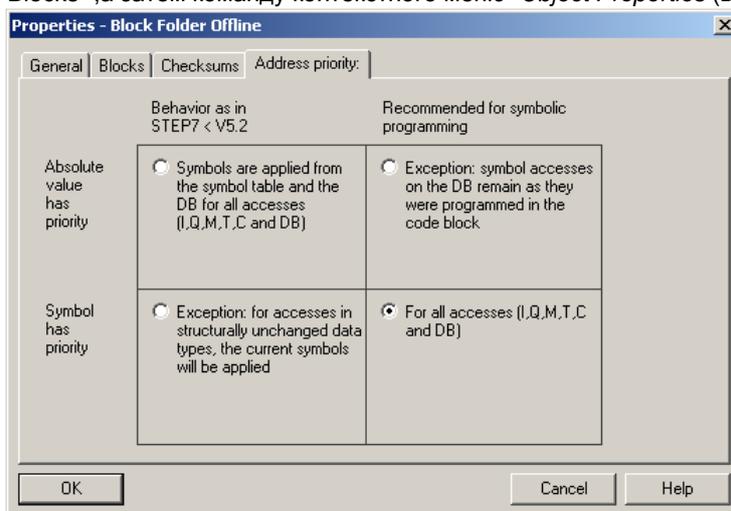


С помощью команды *View* -> *Display with* -> *Symbol Information* Вы можете отобразить одновременно абсолютный адрес, символическое имя и комментарий.

Установка адресных приоритетов

Приоритеты адресов позволяют Вам адаптировать программу после изменения адреса сигнала (например, после перемонтажа).

Для установки адресного приоритета войдите в SIMATIC Manager, выберите папку "Blocks", а затем команду контекстного меню *Object Properties* (вкладка "Address priority").



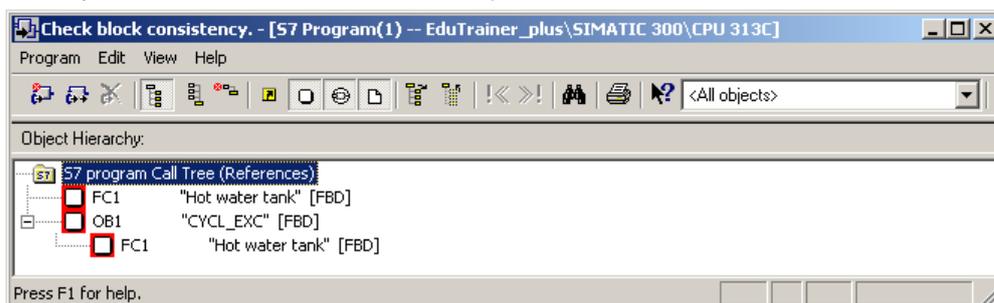
Приоритет имен

Если выбран приоритет имен (предпочтительный выбор), то замена адресов в программе выполняется за 2 шага:

- 1) замена соответствующих адресов операндов в символьной таблице;
- 2) вызов функции *Edit > Check Block Consistency*

Пример: выполнить замену адреса сигнала с именем "Boiler" с I124.3 на I0.3.

В символьной таблице Вы изменяете адрес для операнда "Boiler" с I124.3 на I0.3. Сохраните изменения. Далее, в SIMATIC Manager выберите папку "Blocks" и выполните команду меню *Edit > Check Block Consistency*.



Для автоматической замены адресов во всей программе необходимо выполнить компиляцию всей программы (меню *Program->Compile All*). Если нет ошибок компиляции, то скорректированные блоки программы будут храниться в папке "Blocks".

Приоритет адресов

Приоритет абсолютных адресов используется для быстрой замены адреса операнда или в случае, когда символьные имена операндов не используются.

Для замены адреса(ов) операнда во всей программе в SIMATIC Manager выберите папку "Blocks" (или группу блоков) и выполните функцию *Options-> Rewire*. В окне "Rewire" необходимо заполнить строки таблицы, в которых старому адресу (Old address) нужно сопоставить новый адрес (New address). После заполнения строк подтвердить "OK".

Тестирование и отладка программы

Для тестирования программного блока Вы используете функцию **Debug -> Monitor** или кнопку инструментального меню. Запуск данной функции означает перевести блок в режим **статуса**.



В режиме "Monitor" LAD / FBD элементы отображаются разными цветами. Вы можете изменить настройки цвета элементов с помощью пункта меню *Options -> Customize*.

Примечания

1. Когда активирована функция "Monitor", Вы не можете редактировать Вашу программу и изменить вид отображения (LAD, STL, FBD).
2. Состояние блока может проверяться только тогда, когда эта часть программы выполняется. Состояние блока не может проверяться, если CPU находится в режиме STOP или если данный блок не вызывается.

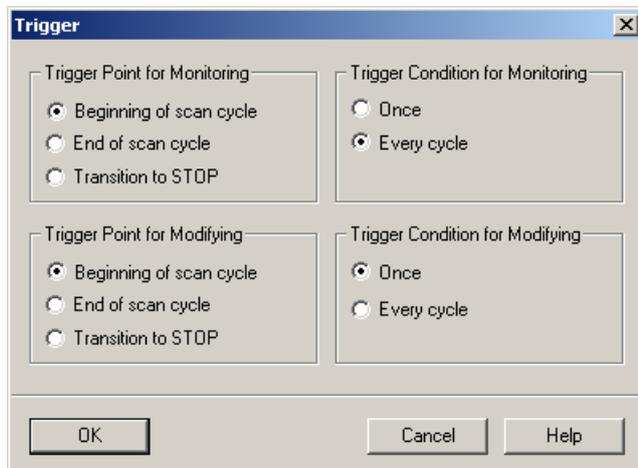
The statements are not being processed.



Monitor and Modifies Variables

Утилита "**Monitor and Modifies Variables**" позволяет наблюдать в выбранном формате состояние переменных вашей программы. Она также позволяет Вам модифицировать состояние переменных в CPU.

Наблюдение/управление переменными производится в соответствии с заданными триггерными точками. Вызов окна задания точек производится через меню *Variable -> Trigger* или по кнопке меню.

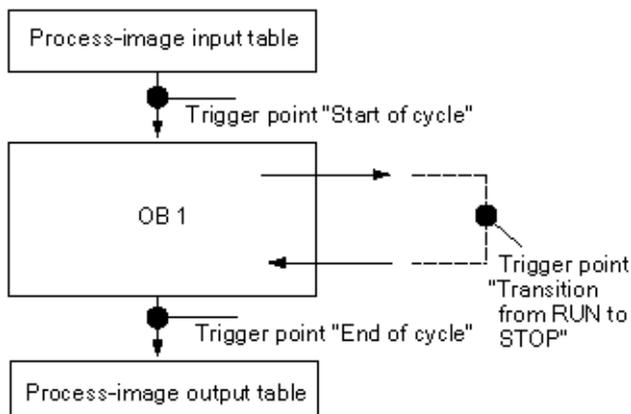


Опции функции "Monitor Trigger Point" определяют момент времени (по отношению к фазам основного цикла) для чтения состояний переменных.

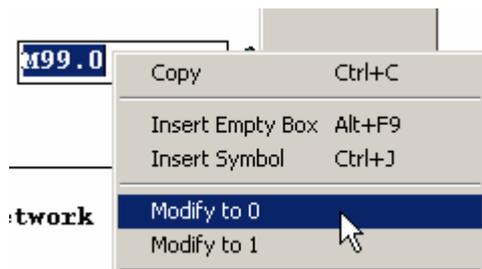
Опции функции "Modify Trigger Point" определяют момент времени, в который установленные значения будут записаны в соответствующие переменные.

Установки "Monitor Trigger Frequency" определяют, должны ли переменные читаться однократно (только по запросу, когда точка достигнута) или в каждом цикле.

Установки "Modify Trigger Frequency" определяют, должны ли новые значения передаваться в соответствующие переменные однократно или в каждом цикле.



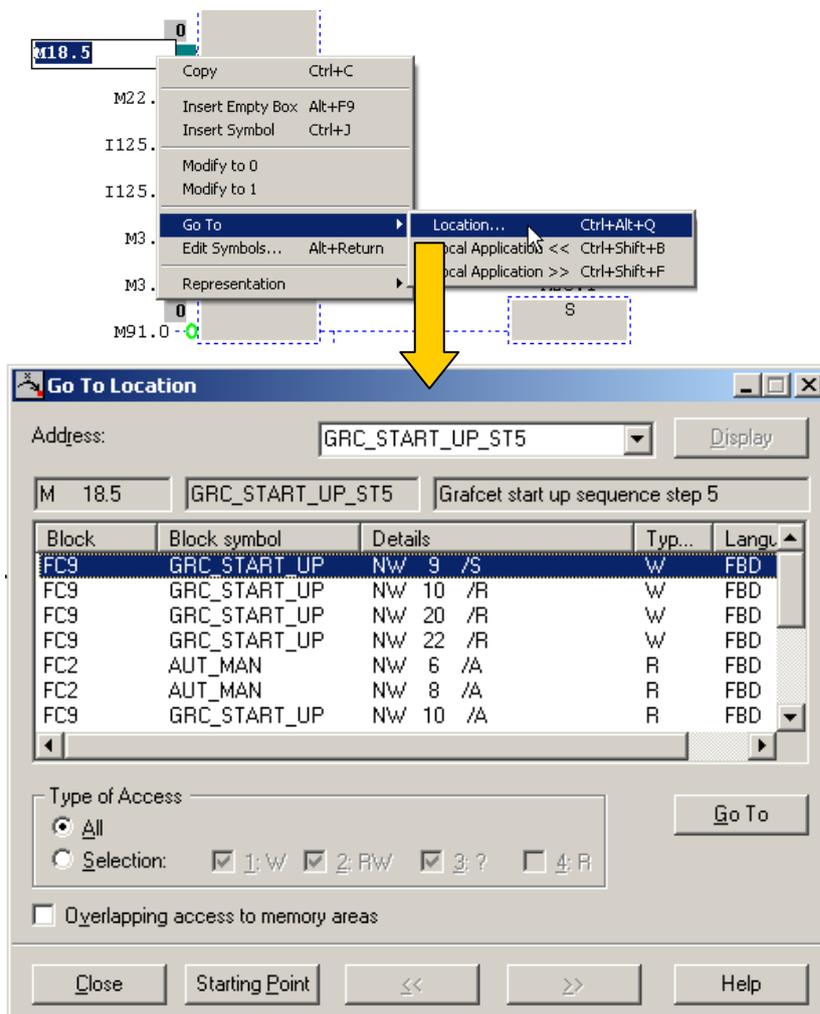
Управление переменными в режиме статуса программы



Вы можете оперативно изменить состояние переменных (ячейки из области M, а также блоки данных) в режиме статуса. Для этого, выделите операнд, и выполните функцию “Modify” из контекстного меню.

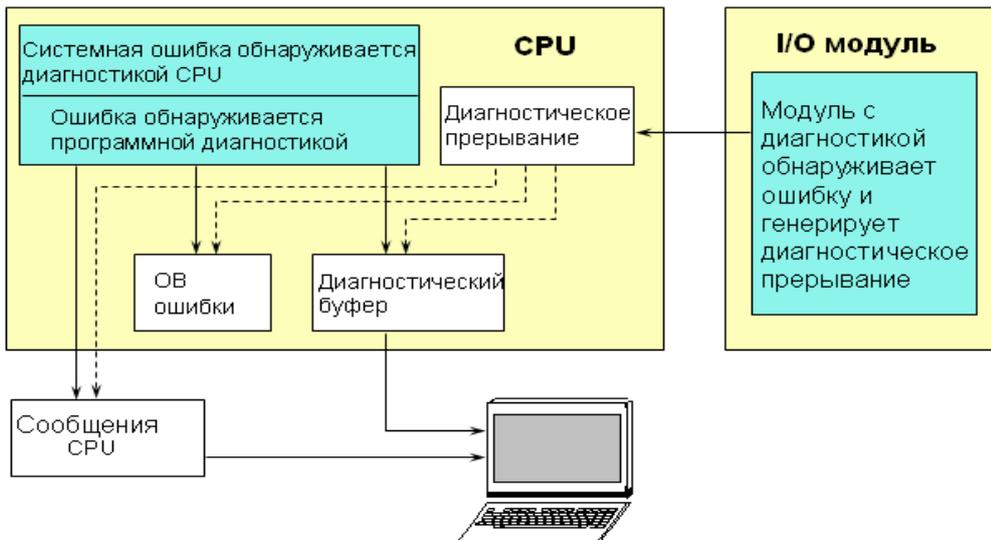
Функция “Go To Location”

Если выделить ячейку памяти и вызвать функцию “Go To Location”, то откроется соответствующее окно, в котором Вы сможете увидеть все случаи использования данной переменной. Выделив нужную строку и нажав кнопку “Go To”, Вы перейдете в новую точку для режима статуса или отображения.



Поиск и устранение ошибок

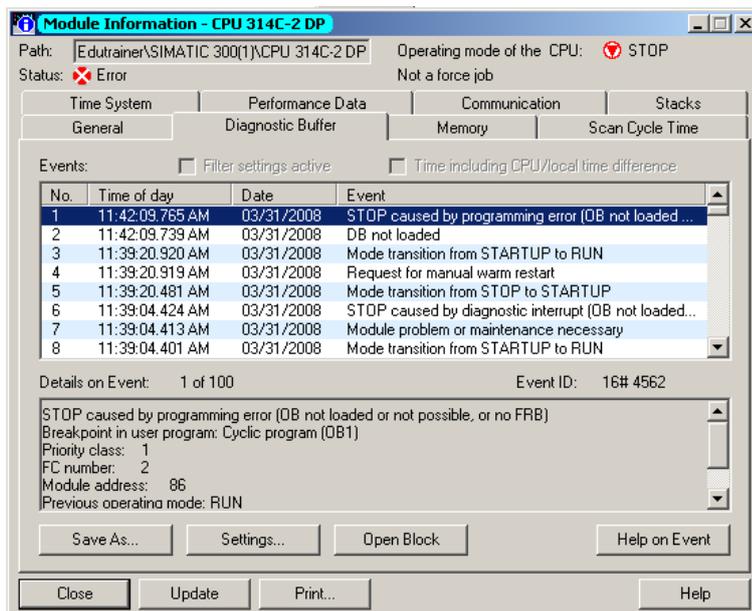
Концепция диагностики в SIMATIC S7



Когда происходит ошибка или событие, например, переход CPU в STOP, то выполняется следующее:

- В диагностический буфер вводится сообщение с датой, временем и кодом диагностического события. Самое последнее сообщение записывается в начале буфера (принцип FIFO). Когда буфер становится полным, самые ранние записи удаляются. Содержимое буфера находится в энергонезависимой памяти и не может быть удалено при сбросе CPU.
- Если предусмотрено, то событие активизирует соответствующий OB ошибки (организационный блок).

Вы открываете диагностический буфер командой меню *PLC -> Module Information*.



Если Вы нажмете на кнопку "Help on Event", то откроется помощь по выделенному событию.

Если Вы нажмете на кнопку "Open Block", то Вы откроете из CPU блок, который привел к ошибке (в примере FC2).

Диагностика с использованием стеков I, V, L

Для того чтобы получить дополнительную информацию о месте ошибки, Вы можете прочитать содержание стеков (I - стек, V - стек, L - стек). Вывод информации о стеках возможен в следующих случаях перехода CPU в режим STOP:

- вследствие программной ошибки,
- вследствие команды STOP,
- при достижении точки останова.

V стек

V - стек содержит список блоков, выполняемых при переходе в состояние Stop.

I стек I – стек содержит данные регистров в момент ошибки, а именно:

- содержание аккумуляторов и адресных регистров
- какие DB открыты
- слово состояния (все биты Status Word)
- прерванный блок с указанием сегмента и номера инструкции
- следующий блок, подлежащий выполнению

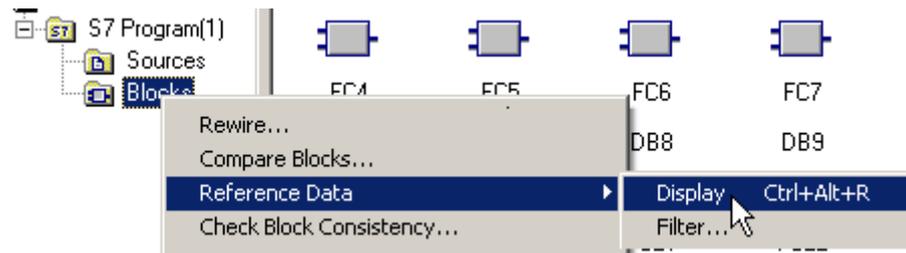
L стек

L – стек содержит значения временных переменных блока.

Поиск логических ошибок

Для поиска логических ошибок, Вы используете инструменты "Program Status" и "Reference Data" (ссылки).

Распространенная причина ошибок - повторное присваивание адресов. Это означает, что адрес используется в различных местах программы. Такие ошибки легко найти с помощью "Reference Data" (в SIMATIC Manager или LAD/STL/FBD редакторе через меню *Options -> Reference Data -> Display*).



Все ссылки могут находиться в 6 различных списках:

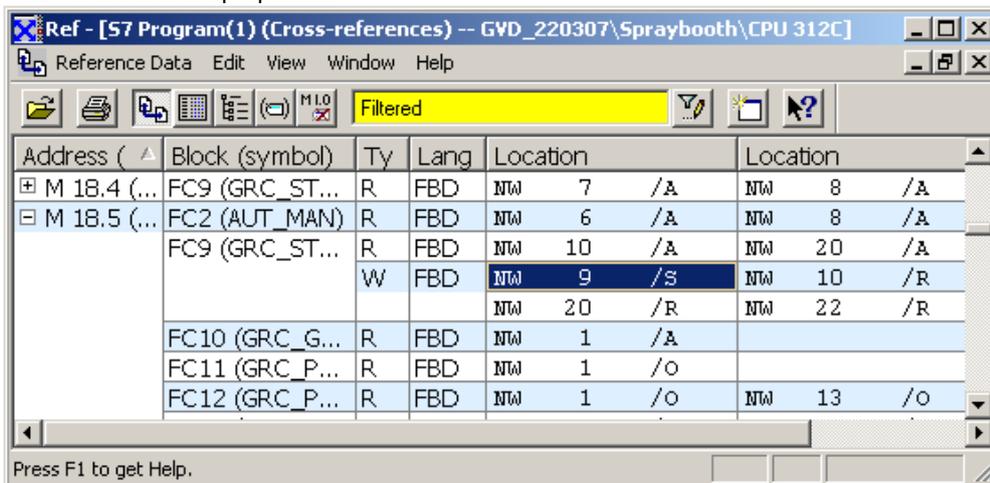


Перекрестные ссылки

Вы открываете список перекрестных ссылок, выбирая команду меню *View -> Cross References* или кликнув мышью на соответствующую кнопку.



Список перекрестных ссылок является списком адресов, использованных в конкретной пользовательской программе.

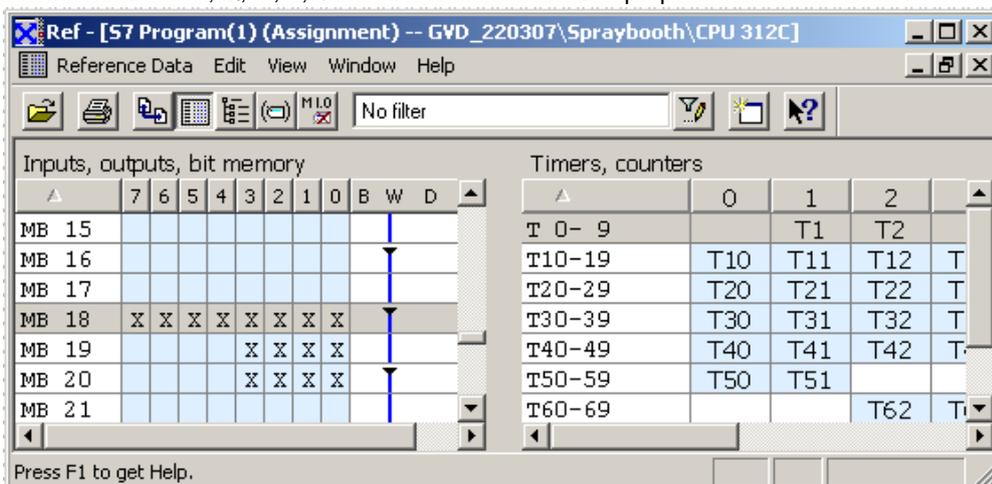


Кликнув на выбранное место использования операнда, Вы запускаете редактор с указанным блоком и сегментом.

Вы можете выбрать адрес в списке перекрестных ссылок, а далее выбрать меню *View -> Cross References for Address*. При этом откроется новое окно, содержащее перекрестные ссылки только для выбранного адреса.

Таблица используемых ячеек (план использования)

Этот список дает обзорный план, из которого видно, какие биты и в каких байтах областей памяти I, Q, M, T, S использованы в вашей программе.

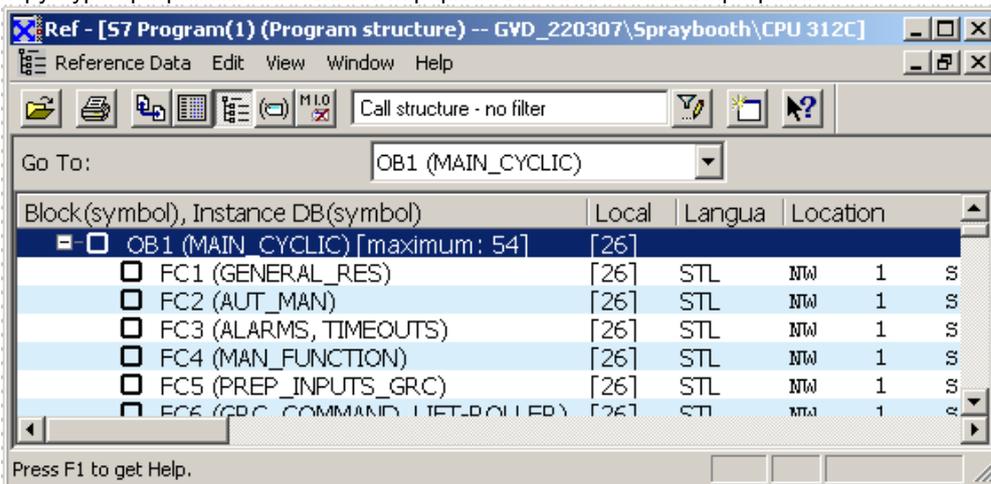


Символы, используемые для списка назначений I/Q/M:

- | | | | |
|-------------------------------------|--------------------------|-------------------------------------|--|
| <input type="checkbox"/> | Адрес еще не использован | <input checked="" type="checkbox"/> | Доступ только к биту, а также данный бит используется при обращении к байту, слову |
| <input checked="" type="checkbox"/> | Доступ только к биту | <input type="checkbox"/> | Доступ к байту, слову или двойному слову |

Структура программы

Структура программы показывает иерархию вызовов блоков в программе.



Используемые символы:



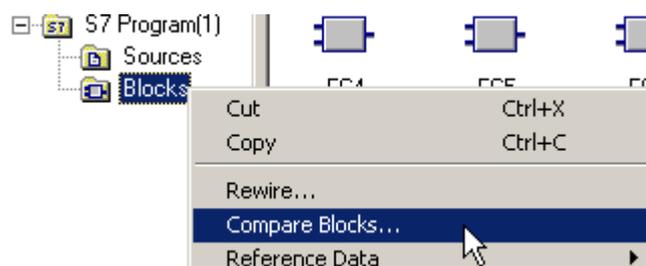
1 2 3 4 5 6

- 1 - Нормальный вызов блока (CALL FC1)
- 2 - Безусловный вызов блока (UC FC1)
- 3 - Условный вызов блока (CC FC1)
- 4 - Блок данных
- 5 - Рекурсия (вызов блока из самого себя)
- 6 - Блок не вызывается

В колонке Local показывается глубина использования локального стека для блоков OB и для вызываемых блоков.

Сравнение блоков

Одной из причин возникновения ошибок (обычно в случае нескольких пользователей) может быть различие между программами в проекте и в CPU. Поэтому поиск логических ошибок (нарушения в логике работы программы) проще начинать со сравнения блоков.



| Block | Result of comparison |
|-------|-------------------------------|
| FC9 | Path 1 contains newer version |
| FC19 | only exists in Path 1 |
| DB9 | Time stamps are identical |

Если блоки различны, то необходимо выделить строку с номером блока и нажать на кнопку "Go To..". Редактор блоков откроет блок в offline и online окнах и выделит сегмент, где имеется различие. Через закладку "Comparison" окна "Details" Вы можете просмотреть весь список различий.



Блоки ошибок

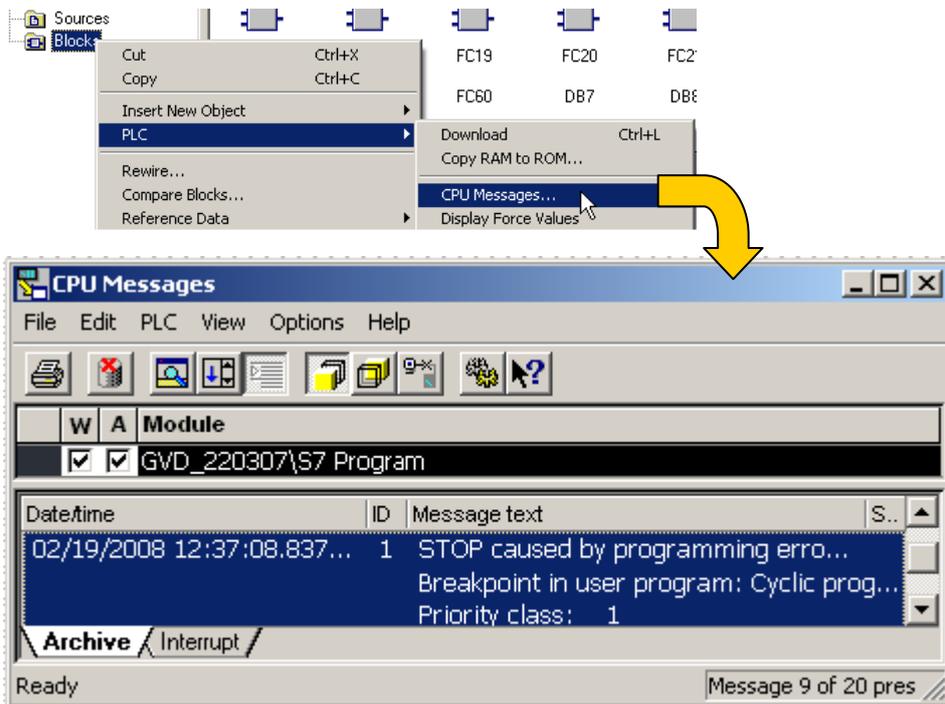
При возникновении большинства ошибок операционная система CPU вызывает различные организационные блоки (OB) в зависимости от типа ошибок. При отсутствии данных блоков CPU переходит в STOP (кроме OB81, OB84).
Ниже приведена таблица OB ошибок.

| Тип ошибки | Пример | Error OB |
|-----------------------------|---|----------|
| Ошибка программы | Вызываемого блока нет в CPU | OB 121 |
| Ошибка доступа | Прямой доступ к модулю, который неисправен или неправильно указан | OB 122 |
| Ошибка времени | Превышено максимального время цикла | OB 80 |
| Дефект блока питания | Неисправность резервной батареи | OB 81 |
| Диагностическое прерывание | Обрыв проводов на входах модуля, поддерживающего диагностику | OB 82 |
| Неисправность CPU | Ошибка в памяти операционной системы | OB 84 |
| Ошибка выполнения программы | Ошибка обновления области отображения (неисправность модуля) | OB 85 |
| Неисправность стойки | Неисправность удаленной стойки или DP Slave | OB 86 |
| Коммуникационная ошибка | Ошибка в передаче "Global Data" | OB 87 |

Блоки ошибок подготавливаются пользователем. В простейшем случае блок OB может быть "пустым", т.е. не содержать команд пользователя. Подробное рассмотрение вопросов программной обработки ошибок в блоках OB обсуждается в курсе S7-PROF2.

Вывод сообщений (функция CPU Messages)

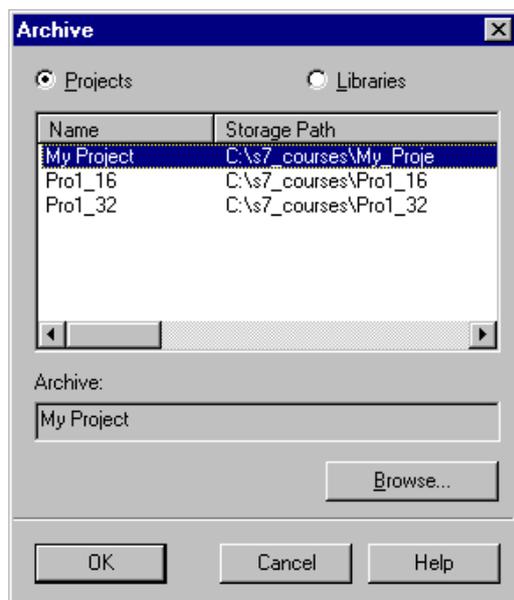
Запуск этой функции позволяет немедленно отобразить сообщение об ошибке на экране программатора (PG) или устройстве HMI. Окно сообщения выводится поверх других окон на PG или OP, как только CPU перейдет из-за ошибки в STOP. Используя в программе системные функции (SFC17, SFC18), Вы можете также выводить пользовательские сообщения, содержащие собственные тексты и, при необходимости, значения переменных.



Бокс "W" активируется для приема системных и пользовательских диагностические сообщения. Бокс "A", активируется для приема сообщения, передаваемых через системные функции (ALARM-функции).

Вы можете модифицировать размер архива (40 - 3000 сообщений) или очистить архив через функцию меню "Option -> Customize". Архив сообщений можно сохранить, например, в текстовом файле с помощью функции "File->Export Archive".

Документирование программы



Архивирование

Архивируемый проект должен быть закрыт в SIMATIC Manager.

1. Выбрать команду меню *File --> Archive*.
2. В диалоговом окне выберите проект, который должен архивироваться.
3. В следующем диалоговом окне введите имя архива и место размещения

Деархивирование

1. Выбрать команду меню *File -> Retrieve...*
2. Выбрать архивный файл.
3. Выбрать папку, в которую будет записан восстановленный проект.

Как сохранить проект целиком в MMC

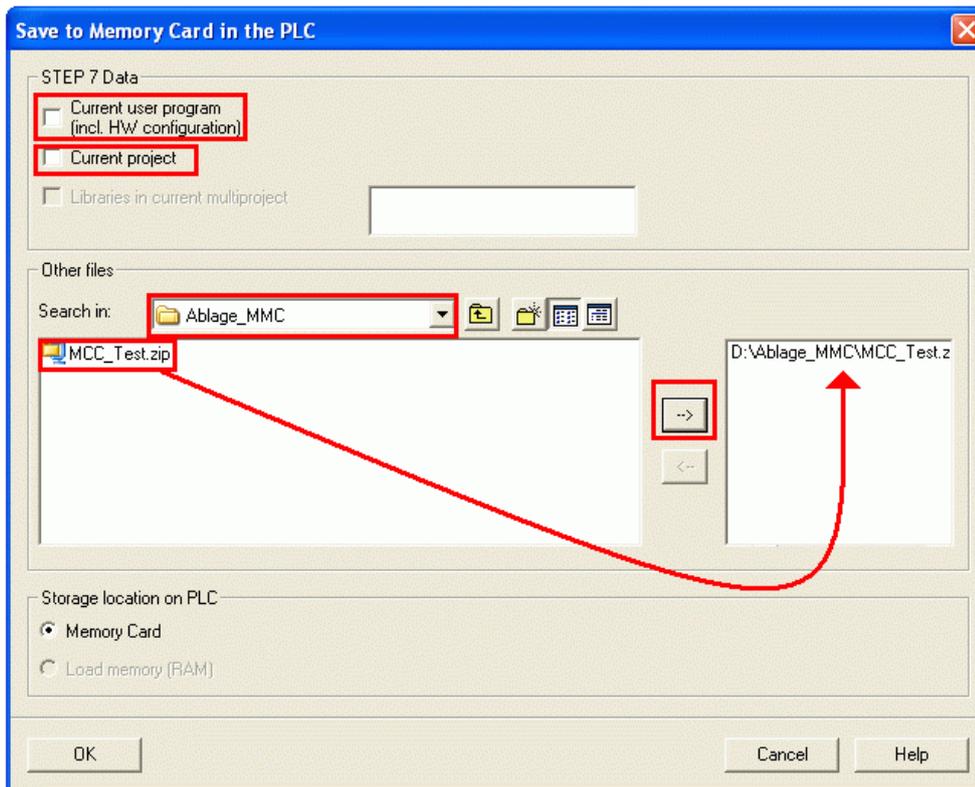
Хранить проект в модуле MMC удобно в случае, если несколько сотрудников заняты обслуживанием системы автоматизации SIMATIC. Тогда все последние изменения проектных данных будут всегда доступны в распоряжение сотрудникам.

Вы можете сохранить весь проект в модуле MMC, если

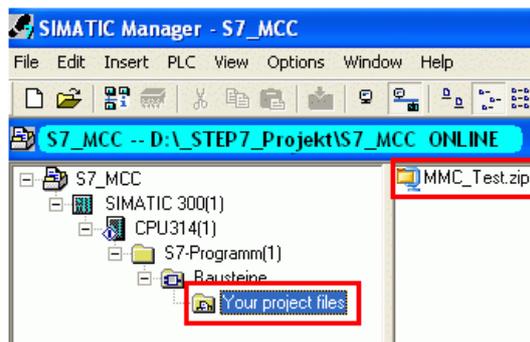
- размер MMC достаточен для записи данных (архива проекта)
- CPU поддерживает данную функцию

Порядок действий:

1. Подготовить архив проекта.
2. В SIMATIC Manager выделить объект "S7 Program" и выполнить функцию "PLC-> Save to Memory Card..."
3. В окне "Save to Memory Card", в подокне "Other file", выбрать архивный файл проекта.

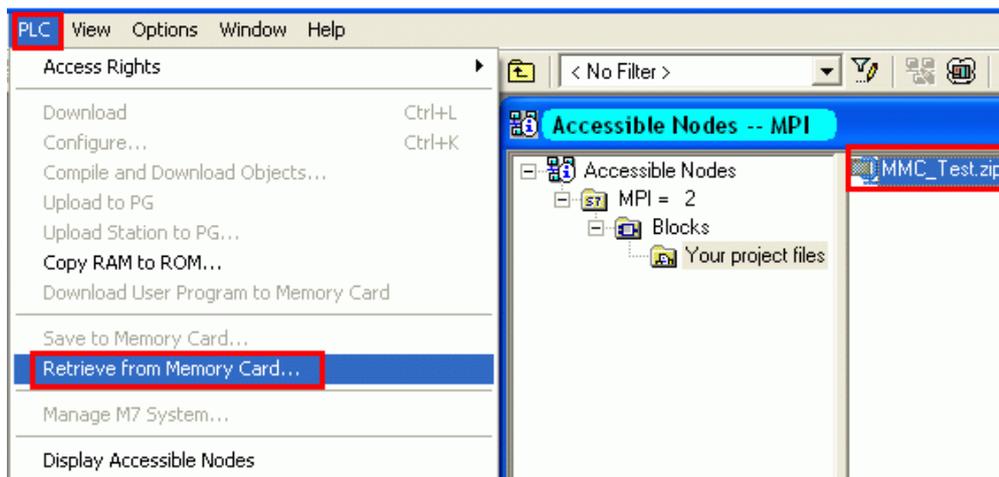


4. После пересылки архивного файла он будет отображаться в Online – проекте.



Чтение проекта из MMC

Для извлечения проекта из MMC откройте окно “Accessible Nodes” и выберите папку “Blocks”. Отметьте ZIP-файл и выполните команду “PLC->Retrieve from Memory Card...”.



Проект (архивный файл) будет сохранен в указанном Вами месте. Затем Вы сможете его открыть (*File -> Retrieve..*).

Заключение

Закончен обзор вопросов, включенных в программу учебного курса S7-PROF1. Как было отмечено ранее, слушатели данного курса получают базовый уровень знаний. Повысить свой уровень знаний по системам SIMATIC S7 позволит курс S7-PROF2, в котором участники подробно знакомятся со следующими вопросами:

- Структурное программирование. Использование блоков FC и FB
- Блоки данных (DB). Доступ к элементам DB. Комплексные типы данных
- Организационные блоки (OB). Программные и аппаратные прерывания
- Типы адресации данных. Косвенная адресация
- Расширенные средства тестирования программ. Контрольные точки
- Системные функции (SFC/SFB). Примеры использования
- Стандартные библиотеки и библиотеки пользователя

В курсе S7-TECHN слушатели знакомятся с методами формального проектирования программ. Рассмотренные в курсе S7-TECHN методы позволяют превратить искусство написания программ в простой набор правил. Участник получает технологию для "монтажа" программы из готовых и понятных "конструкций". Что дает пользователю знание и умение данной технологии?

- Простая реализация программы на основе алгоритма
- Встроенный контроль выполнения каждого шага алгоритма
- Сокращение издержек на разработку и отладку программ
- Программирование без ошибок

Желаем успехов в дальнейшем освоении “секретов” программирования и обслуживания систем SIMATIC S7!